Hindawi
Security and Communication Networks
Volume 2020, Article ID 8850472, 21 pages
https://doi.org/10.1155/2020/8850472

Research Article

Hidden Service Website Response Fingerprinting Attacks Based on Response Time Feature

Yitong Meng and Jinlong Fei

State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China

Correspondence should be addressed to Jinlong Fei; feijinlong@126.com

Received 20 August 2020; Revised 8 October 2020; Accepted 10 November 2020; Published 1 December 2020

Academic Editor: Zhe-Li Liu

It has been shown that website fingerprinting attacks are capable of destroying the anonymity of the communicator at the traffic level. This enables local attackers to infer the website contents of the encrypted traffic by using packet statistics. Previous researches on hidden service attacks tend to focus on active attacks; therefore, the reliability of attack conditions and validity of test results cannot be fully verified. Hence, it is necessary to reexamine hidden service attacks from the perspective of fingerprinting attacks. In this paper, we propose a novel Website Response Fingerprinting (WRFP) Attack based on response time feature and extremely randomized tree algorithm to analyze the hidden information of the response fingerprint. The objective is to monitor hidden service website pages, service types, and mounted servers. WRFP relies on the hidden service response fingerprinting dataset. In addition to simulated website mirroring, two different mounting modes are taken into account, the same-source server and multisource server. A total of 300,000 page instances within 30,000 domain sites are collected, and we comprehensively evaluate the classification performance of the proposed WRFP. Our results show that the TPR of webpages and server classification remain greater than 93% in the small-scale closed-world performance test, and it is capable of tolerating up to 10% fluctuations in response time. WRFP also provides a higher accuracy and computational efficiency than traditional website fingerprinting classifiers in the challenging open-world performance test. This also indicates the importance of response time feature. Our results also suggest that monitoring website types improves the judgment effect of the classifier on subpages.

1. Introduction

In the face of diversified attacks, anonymous communication platforms such as Tor [1], I2P [2], and Zeronet, take advantage of the protocol features to protect the data security and identity privacy for both parties involved in digital communications over Internet. For instance, Tor provides low-latency interaction to meet users' needs for anonymous access. On a daily basis, Tor embeds hidden service to protect anonymity of 150,000 websites. In order to monitor the hidden services that usually provide illegal services in violation of the user policy, regulators put forward attack technologies based on different levels of traffic and protocol as countermeasures. Instances of common legacy attack technologies include active association attack [3], information disclosure attack [4], and node attack [5]. Biryukov et al. [4] proposes to control the key nodes to send 50 padding packets and detect server's packet drop feedback. He did not however make any actual test and evaluation. Matic [6] uses service certificate vulnerabilities to detect the certificate chain of hidden services for tracking purpose; however, the achieved success rate is only 79%.

In order to avoid the problems associated with elevated conditions and low real-world test accuracy in conventional hidden service attacks, here we use mature website fingerprinting attacks [7–9]. This enables us to reexamine the website response and further achieve effective hidden service attacks based on analyzing the implicit information of hidden service response traffic. Website fingerprinting attacks are known for being capable of destroying the anonymity of transmission at the traffic level. The main goal of such an attack is to match the website contents (i.e., website

pages and its addresses) generated by the Tor users. Attackers are local observers, including local network administrators, Internet service providers (ISP), autonomous system administrators (AS), and other network backbones. The attacker's attack capability is also amplified. If the observer is near the monitoring server, the attacker can passively analyze the response encrypted traffic generated by the website and then record the two-way transmission features of data packet. These features are then matched with the self-built response traffic template library to restore the monitoring website content and track the website server.

In this paper, a novel hidden service attack technique, hidden service website response fingerprinting (WRFP), is proposed based on response time feature. We show, for the first time, that in a real communication environment, the WRFP attack is in fact capable of threatening hidden services. The attack scenario is shown in Figure 1. WRFP utilizes response fingerprinting dataset of hidden service, then designs website mirroring, and builds different mounting modes of the same-source or multisource servers, thus accurately replicating the website response traffic status. WRFP further analyzes 87 combined features including response time feature with an extremely randomized tree algorithm. It is shown that the proposed method successfully categorizes the index pages and subpages, websites, and service types of hidden services. The contributions of this paper are listed in the following:

(i) We propose a novel hidden service WRFP attack which is different from the traditional website fingerprinting attack. WRFP monitors the website pages, its types, and website's servers simultaneously. Furthermore, the higher the target, the better the classification effect will be.

(ii) We collect 300,000 pages of hidden service WRFP datasets to achieve the same benefit of data traffic of the real hidden service website. The datasets include the construction and mounting of two service scenarios of the same-source and multisource servers.

(iii) We propose a combination feature based on response time to select and optimize the traffic feature. We then show rationality and effectiveness of the response time feature using test evaluations.

(iv) We compare the recognition performance and computational efficiency of WRFP with other website fingerprinting classifiers and show that its classification performance is higher than that of k-NN [7], CUMUL [8], and k-FP [9]. We also test the advanced fingerprinting defense model and show that the lightweight website fingerprinting defense model is unable to effectively resist response fingerprinting attacks.

The rest of this paper is structured as follows. Section 2 introduces the deployment form of website fingerprinting attack and hidden service server and explains the attack hypothesis and targets. Section 3 summarizes the methods and effects of traditional website fingerprinting attacks and analyzes the results and deficiencies of various attacks on hidden services. Section 4 is to design website mirroring and website mounting methods and to collect the response fingerprinting dataset of hidden service websites. Section 5 is concerned with the response time measurement method and proposes the WRFP classifier based on the response time feature. Section 6 cross-evaluates the WRFP in the closed and open worlds and further conducts tests for the network-type subpage classification, TBB version changes, and defense models. Section 7 summarizes the paper and proposes the future work.

2. Background

2.1. Website Fingerprinting Attack. Evidences show that the success of the website fingerprinting attack technique is mainly due to the fact that attackers can capture and analyze the statistical feature of data packets exposed at the traffic level. This compromises the reliability of anonymous networks. This is shown in Figure 2(a) as the attacker analyzes the original encrypted communication data, extracts the transmission data at the cell, TLS, or TCP level [8], and then obtains features such as packet length, direction, and sequence. These features are then matched with the feature sequence of the monitoring website pages. In Figure 2(b), the data traffic deformed by fingerprinting defense is shown. The essence of website fingerprinting defense [10–13] against attacks lies in the real-time adjustment of the exposed features of the traffic, the quantitative changes in the statistical feature that attacks rely on, and obscuring the transmission trajectory of real websites. These are adjustments made to deceive the classifier. Conventional methods include analyzing the diversified traffic feature attack classifiers, improving the resistance of website fingerprinting defense models, reducing the accuracy of attack classifiers, and enhancing the security and anonymity of network transmission.

By evaluating the classifier performance of a website fingerprinting attack, the attacker tries to choose a closed world with a single target or a complicated open world based on the number of monitors [9]. In the closed world, restricting the monitoring quantity and accessing range of websites have the advantages of small training data, complete presupposition, and low monitoring cost. Therefore, it is the first choice for the attackers to evaluate the basic performance of the classifiers. However, in the open world, the real communication scenario is simulated and the number of unmonitored website pages in the big background is far larger than those of the monitored pages. This means that the classifier needs to comprehensively judge the feature information of unmonitored websites, to be able to pose a viable challenge to the performance of the classifier.

2.2. Website Server. We usually suppose that the service type of a website represents its service goal. To examine the content of monitoring hidden services, we classify the websites into eight categories, including store, search, e-mail, news, forum, social, porn, and others. We further

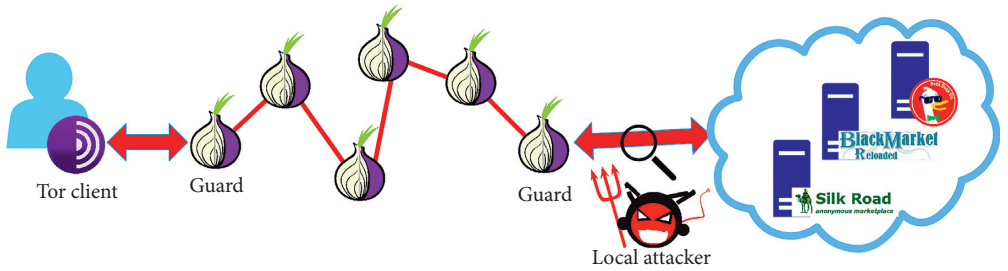

Figure 1: Hidden services response fingerprinting attack scenario.

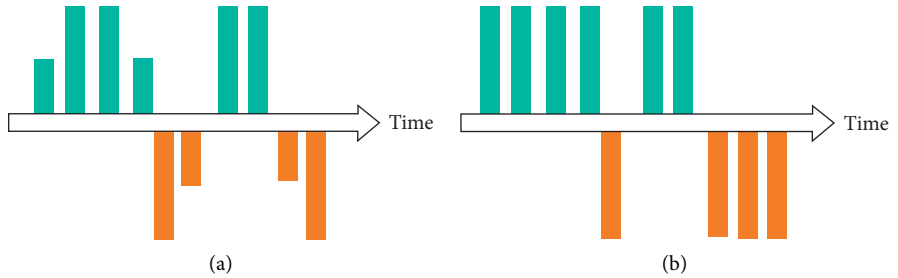

Figure 2: Website fingerprinting attack and defense data traffic. (a) Original encrypted communication data traffic. (b) Fingerprinting defense deformed data traffic.

note that a website relies on the carrying capacity of the server and responds to web data according to the internal scale of the server. Therefore, we consider two scales for the server configuration: same-source server and multisource server (Figure 3).

The same-source server is the basic configuration, where single or multiple website domain names are deployed in the same server space. To avoid issues such as insufficient server performance or data lost caused by multiuser responses, the two-way traffic of each website is concurrently processed on the service port. For instance, port 443 in Figure 3 is an external service port. By expanding the scale of the website or increasing the number of visitors, the website resources should be divided and deployed in a cluster server. In a cluster server, distributed servers, SQL servers, load balancing servers, and other functions cooperate with each other to provide the response to the needs of the website. Such a setting is collectively referred to as the multisource server. In a multisource server, the internal response traffic of the servers is then collected and provided through the service port for unified forwarding. This is different from that of the same-source servers in terms of traffic profile. In this paper, we focus on the response fingerprinting analysis of servers with two scales.

2.3. Attack Hypotheses. Juarez [14] pointed out that attacks on traffic fingerprints requires reliable attack hypotheses, and the success of website response attacks further depends on the hypotheses in the mirror mounting stage, fingerprint capturing, and response time measurement. In the mirror mounting stage of the monitoring website, it is assumed that the attacker has the ability to obtain the software and hardware configuration of the monitoring website server and copy and reconstruct the service content of the website page. It is further assumed that the attacker is capable of forwarding the response traffic according to the same-sized same-source or multisource server configured by software and hardware. This hypothesis is preliminarily verified for suitability in Section 4.2.

In the response fingerprint collection stage, it is assumed that the attacker, when it is monitoring locally, is able to judge the start and end of a single page request and response traffic and further filter out the remaining noise traffic. This enables the attacker to effectively identify and isolate the server responding to multiple response traffic of the same domain name. Wang's [15] research shows that the page parsing assumption is in fact a difficult task. In the stage of extracting the features of the response time, it was necessary to assume that the attacker is less disturbed by the environmental factors before and after the traffic is captured, and the overall time volatility is within the receiving range. Our experiments reported in Section 6.3 show that, by sliding the response time within 10% of the prediction range, the classifier keeps a normal judgment.

2.4. Attack Goal. Our goal is to attack the server mounted by the hidden service, that is, to monitor the server that provides the specified website at the traffic fingerprint level. The test in Section 6.3 shows that the recognition effect of the classifier can be improved by focusing the target on the server. The accuracy is also improved by focusing the target on the type of website service. At the same time, the classifier supports classification of the same-source and multisource servers and realizes the corresponding recognition effect in the face of different monitoring targets. In general, the response fingerprinting attack is different from the traditional

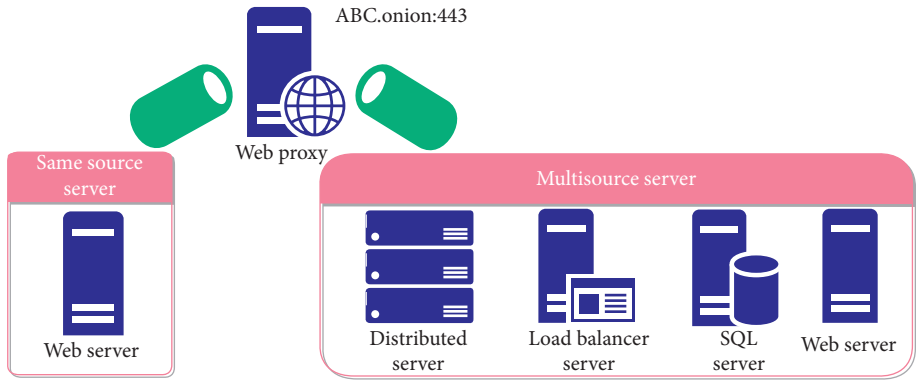

Figure 3: A schematic of the web server deployment.

fingerprinting attack, which is only a single target to restore the trajectory of user's behavior.

3. Related Works

3.1. Traditional Fingerprinting Attack. Website fingerprinting attacks are based on the passive traffic analysis in the initial stage where the tunnel traffic with data encryption techniques such as VPN, HTTPS, and SSH is identified. For the first time, Herrmann [16] uses the traditional data packet length feature to evaluate the protection performance of Tor website and obtained a modest webpage recognition accuracy rate of 2.95%. In fact, Herrmann's work turns the research goal in the field of website fingerprinting to Tor and tries to build a website fingerprinting model that is different from tunnel traffic recognition. Cai et al. [17] then optimizes Herrmann's recognition algorithm by using string-based edit distance algorithm and SVM model to characterize the communication data packet sequences. Their method achieves 86% recognition accuracy for 100 websites in the closed-world scenario. LashKari et al. [18] focus on time based traffic features, introduce 32 features, and show that the time features can be used to classify the type of communication type. The accuracy of the proposed method is not however comparable with that of the previous works.

Using deep learning techniques further increases the recognition accuracy of the classifiers. In 2016, Kota and Goto [19], instead of manual selection of the feature, use a SDAE autoencoder and achieve an accuracy of over 86% with simple input data. Later, Rimmer et al. [20] explore the automated recognition results provided by a variety of deep learning algorithms; however, to obtain a reasonable accuracy, their approach requires access to a rather large dataset including up to 2500 instances per page. In 2018, Sirinam et al. [21] propose a recognition model of deep fingerprinting (DF) which is trained and optimized based on CNN. This is the first classifier that is capable of producing effective attacks on WTF-PAD fingerprinting defense. Bhat et al. [22] further optimize the DF, to obtain an accuracy of 97.8%. They use a semiautomatic characteristic extraction to improve Var-CNN classifier understanding of the nature of the input data based on a small training dataset and short training time.

Although fingerprinting classifiers based on deep learning are in the mainstream of improving the recognition accuracy, they often lack characteristic interpretability and thus unable to express the differences between fingerprints compared with the traditional machine learning algorithms. In other words, deep learning weakens the statistical feature exposed by the encrypted traffic and prioritizes the accuracy of the classifier [23]. Therefore, in order to explore the degree of characteristic influence of response fingerprints, the following three representative machine learning classifiers are chosen for testing and comparison with the algorithm proposed in this paper.

3.1.1. k-NN Classifier. Wang et al. [7] propose a website fingerprinting classifier based on k-nearest neighbors (k-NN) and extract 3736 combined features such as the quantity, order, and single packet length of the load. The Euclidean-like distance function suitable for the recognition is then used to measure the distance similarity between features. Then, the weighted value of characteristic distance is adjusted to measure the efficiency of the classification. They show that k-NN is able to reach 85% TPR by monitoring 100 pages in the open world scenario.

3.1.2. CUMUL Classifier. Panchenko et al. [8] propose a CUMUL classifier based on the SVM algorithm which achieved 92% recognition accuracy for 100 pages in the closed-world scenario. CUMUL expresses the tracked cumulative sum of packets as a length cumulative vector to minimize the differences in bandwidth, congestion, and webpage loading time. A total of 300,000 real page fingerprint sets are collected to ensure the rationality of the classification results.

3.1.3. k-FP Classifier. Hayes and Danezis [9] propose a more advanced k-FP classifier in terms of web page recognition accuracy, and its recognition accuracy to Alexa pages and hidden service pages is both above 90%. k-FP is a set of webpage classifiers based on random forest and k-NN. It creatively uses random forest leaves to represent Hamming distance to achieve webpage classification and systematically analyzes the information benefits of 175 features. The test

result shows that using the first 40 combined features results in the highest classifier accuracy.

3.2. Hidden Service Attack. In 2008, Zander and Murdoch [24] propose an attack model based on hidden service request clock drift, and the results indicate that it had a certain effect, but the attack condition depended on the number and time of arrival of hidden service requests. Elices and Perez-Gonzalez [3] improved the request arrival time prediction, taking each request as an interval as a reference, thus improving the positioning effect of hidden services. Biryukov et al. [4] found that the server would make abnormal feedback after sending the padding packet to the key position of the joint rendezvous node and guard entry node. Ling et al. [5] manipulated the transmission load by using the similarity principle and confirmed the hidden service IP address by analyzing the protocol-level feature of the load at multiple intersection points. In 2015, Matic et al. [6] designed the detection tool CARONTE to locate the actual server IP address by detecting the unique string and certificate chain leaked in the content of the hidden service. Tan et al. [25] used the Eclipse attack to destroy the DHT structure of the hidden service directory server, which could cover any hidden service under a small amount of IP resources.

In terms of the research objectives, compared with the previous active attack methods, we tended to realize attacks based on passive traffic analysis theory. As early as 2015, when Kwon et al. [26] realized the classification of Hidden Service link fingerprints, they also initially explored the threat of website fingerprints to hidden services. Although the experiment obtained 88% TPR, for the lack of consideration of some objective factors such as actual server software and hardware configuration, environment, and internal relations, this research is just a reapplication of the website fingerprinting attack scenario, and it is not helpful to the reliability and effectiveness of response fingerprinting data analysis.

4. Dataset

During the hidden service response period, the confusing traffic between the server and the client is an unequal traffic; therefore, the previously disclosed website fingerprinting dataset cannot be used. This is because of objective reasons such as network disturbance, load encapsulation, and imbalance between the demand and transmission link capacity. Here, we collect a set of universal response fingerprinting dataset of hidden service websites combined with the actual server size, server mirroring, and mount mode. The dataset provides a reliable data basis for the classifier training.

4.1. Hidden Service Addresses. The addresses of hidden services are opaque and need to be collected through public crawling, network detection, and memory extraction. The public collection methods include summarizing addresses through Hidden Wiki, Real-World Onion Sites, Daniel's Hosting, and other means. The nonpublic collection methods consist of network detection and memory extraction. Network detection is based on Elasticsearch engine, combined with large search sites for large-scale hidden service link scanning. Memory extraction [27] is much more hidden and uses a self-built server that temporarily applies for and obtains the HSDir tags to extract memory from the short-term upload addresses.

Considering the rapid offline of some home pages and Tor privacy protection of Tor, only memory addresses within 96 hours are extracted. It should be also noted that servers with versions higher than Tor 0.3.5 give priority to v3 addresses. If the hidden service provides both v2 and v3 addresses, only v3 addresses are stored. During the collection, according to the HTTP status code and curl error code of the websites, the invalid, duplicated, and censored addresses are also deleted, and a total of 34,890 active website addresses are kept (Table 1).

4.2. Website Mirroring. As mentioned in Section 2, accurate simulation of the communication behavior in the real environment is one of the key steps in fingerprint collection. Taking the response traffic of DuckDuckGo as an example, when the response fingerprinting sets of hidden service are explored, the issue with the accuracy of image matching needs to be solved. In the present study, a self-controlled exit node is set to visit the normal site (http://duckduckgogo.com) and the mirror site (http://3g2upl4pq6kufc4m.onion). The response traffic (T and RT) is then recorded. The response packets are also arranged in equal proportions in time (Figure 4). The mean value of 50 subpages is also calculated. The traffic of mirror image (RT_1) is displayed on the top of Figure 4, and the combined traffic of mirror image and multisource server mount (RT_2) is shown in the bottom:

$$1 > \text{Array}\left(\sum \frac{RT_2}{T}\right) > \text{Array}\left(\sum \frac{RT_1}{T}\right). \quad (1)$$

A closer value of the cumulative sum of RT/T in mirror matching to 1 indicates a higher sensitivity of the packet can be maintained to time. Therefore, after evaluating the relationship between the site and server image, it is determined to build a combination mode of website image and website mounting to provide a real website response traffic.

The content of mirroring is stipulated. Public information such as website templates, service content, and addresses is included. Furthermore, server's architecture technology stack, file layout, storage, and other private information are detected with the modified OnionScan supporting tools. Public information refers to the main presentation form of the website service content. First, a website crawler tool is developed based on requests and selenium to obtain complete HTML, CSS, JavaScript, and other structural data. The internal and external links of JSON, JavaScript, and CSS language are then automatically rewritten, such that the style of the website pages is not lost and even the address links of dynamically generated images cannot be mirrored correctly. Secondly, we use Flask lightweight framework to build the site and Cchardet module to assist in confirming the coding method of the

Table 1: The number of collected hidden service website addresses.

Collection method	Addresses collected	Active addresses	
		v2	v3
Public crawling	24,591	13,130	8461
Network detection	16,345	10,328	4017
Memory extraction	53,452	36,234	16,218
Total		34,890	

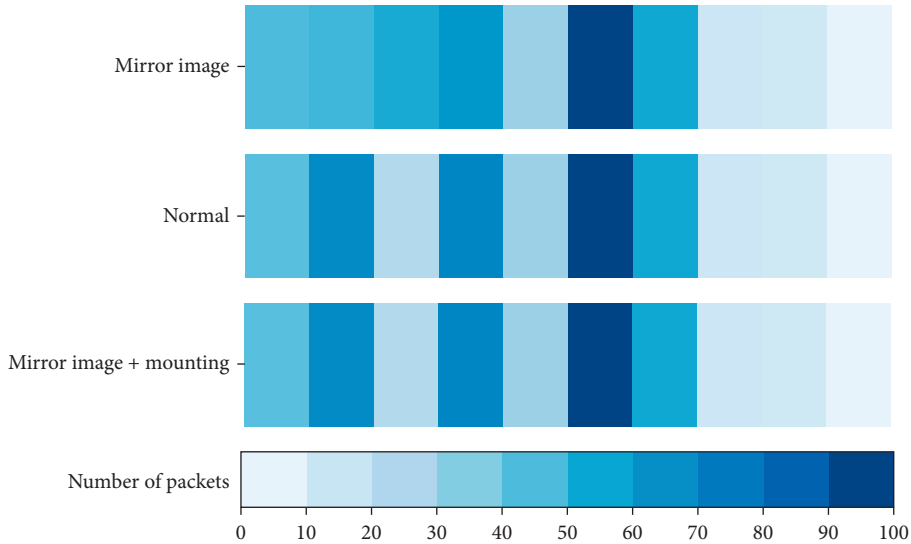

Figure 4: Comparison of response traffic of mirror image of DuckDuckGo.

website to ensure uniform mirror content. Finally, to ensure the balance between each type of website and the number of accepted subpages, the collection and processing of subsequent subpages of each website are determined according to the website type. The specific crawling depth of the website type is shown in Table 2. Websites of e-mail type need logging in and cannot be effectively crawled; therefore, only the login page is recorded as the index page.

Privacy information represents the response performance of the server; fortunately, information collection can be done without controlling the server itself. It is also found through scanning OnionScan that about 55% of hidden service websites are equipped with the classical technology stack combination of Nginx + PHP, and 38% of websites have a common file layout structure, such as /style, /images, /css, and other major directories. To optimize website hosting, the key exchange in the handshake phase of HTTPS is also ignored, and all HTTPS sites are simplified to HTTP sites. Different from the TLS protocol, in Tor transmission tunnel, the SSL certificate is only provided through simple handshake by HTTPS exchange protocol. Therefore, this cannot affect the subsequent page element transmissions. Optimization of the network bandwidth, server requirements, and internal links is conducted within the acceptable range. For example, timely deletion of access.log could effectively solve the problem of slow response.

The website relies on the server for page response. Further, its relationship with the server is usually one-to-one, i.e., one server mounts one domain name website. To meet the practical tactical needs, the relationship between website and server is also expanded to one-to-many and many-to-one cases. In this study, considering that the server configuration determines the quality of the site's response, based on the potential impact of the server size of the hidden service mount, the mounting scenarios of websites are classified into the three following categories:

4.2.1. Same Source-Single Website (SS-SW). In this scenario, the server only provides basic web services to the website. The response of the website can be optimized by using the performance of the server, and the server is very inclusive to the performance of the website. This scenario is mostly selected for temporary small websites.

4.2.2. Same Source-Multiple Websites (SS-MW). Many addresses in hidden services are attached to the reliable large server providers such as Daniel's Hosting, Ablative Hosting, and Kowloon Hosting. Each server of the server providers mounts multiple independent domain name websites. Here, the configuration mode of Daniel's Hosting is selected and copied in constructing scenarios in this study. To ensure the effective quarantine of agents between the websites, independent domain names are used for website addresses. The configuration of the hidden service hosting server is the basic configuration of Nginx + PHP + NTP, equipped with independent website registration function. Moreover, the server in the group passed the load performance stress test of the

Table 2: Selection of crawling depth for different types of websites.

Website type	Store, search, e-mail	News, porn	Others	Forum, social
Page properties	Index page	Index page + subpage	Index page + subpage	Index page + subpage
Crawling depth	Level 1	Level 2	Level 3	Level 4

Locust framework. It is found that if more than 235 domain name websites are mounted, the response failure rate will be gradually increased and the response time fluctuated frequently. Considering the bandwidth difference of the websites, 150 response ports are opened in the server to meet the load balance of multisite responses.

4.2.3. Multisource-Single Website (MS-SW). To improve the availability threshold of large-size websites, the load balancing server, web server, SQL server, and distributed server are combined into a server cluster. Among them, the distributed server provides several types of response elements, the SQL server mounts the background data, and the load balancing server balances the load of the website response traffic. By receiving large-scale request traffic, the server cluster automatically retransmits the traffic to the server members according to the response rules and responds to the specified resources.

4.3. Website Mounting. The mirror image of the website is automatically mounted according to the server size. The inconsistency of mirror response may occur between websites as well as servers. Therefore, in feasible mounting scenarios, the mounting is performed with reasonable rules according to the website configuration and data space. As shown in Algorithm 1, to avoid the intersection of website data, each domain name is only mounted in one server scenario.

Steps 5–11 of the algorithm: the website is placed in the SS-MW scenario if the server configuration matches with the configuration of classical technology stack. If the data space exceeds the maximum allotted space by the server, the MS-SW scenario is then imported.

Steps 12–15: the website is placed in the SS-SW scenario, if the underlying configuration of the current site is not in records. However, the response failure is likely, where the website is placed into the matching scenario. To exclude this situation, the HTTP access status code is detected and the abnormal website is manually debugged. In addition, the number of automatic control servers is much smaller than the total number of websites. The website mounting status is presented in Table 3, where the rotational mounting mechanism of the website is considered. The number of groups is the number of rotational mounting of the website:

4.4. Collection of the Response Traffic. To ensure the controllability of the response traffic and to prevent hidden service from being accessed by outsiders, we build a small Tor network based on the website image reserve. The collection of response traffic from resource deployment to traffic data processing is divided into the following five steps:

Step 1. Resource deployment: we build a small auto-control network composed of 40 relay nodes. It is equipped with necessary components such as hidden service directory server, authoritative directory server, and client agent. The release and update time of hidden service address are also revised to 15 seconds. Eight Amazon EC2 servers are mounted with the website content of different scenarios. Among them, one server was set as a large hidden service hosting server and another 4 formed a server cluster.

Step 2. Synchronization of client with server: inside the server is a rotational mounting mechanism. NFS is built to allow the client to share the address list and website resources with the server. Moreover, the client is permitted to modify the list. When the website is visited successfully, the header address is deleted, and the server automatically reads the most recent address for mirror image mounting.

Step 3. Traffic collection the request-response cooperation between the client and the server realizes the collection of traffic. The client traverses the list of hidden services in order and starts the tor-browser-crawler [14] self-running program. The interval between the visiting website and the internal page is set to 20 seconds, to have a sufficient traffic response and website replacement time. We then use Munin to monitor the traffic at the backend of website server. When the bandwidth of the service port is suddenly increased, it cooperates with tcpdump to collect the traffic. After completion, server sends destroy instruction packet to the link through the stem controller, and the client restarts the Tor program synchronously. This eliminates the impact of data retention and continuation while keeping the complete response packet. Finally, the server implements permuted access to the list of websites. Therefore, when the website response traffic is successfully collected, the current website is immediately gone offline and the mirror agent of the presorted website is turned on. Compared with the multiple visits to the website in a single round, the advantage is that the response of the website will not be affected by the internal environment of the server and will not miss the opportunity to get a fresh and correct response packet. The data are collected in a total of 50 rounds, and the same number of instances of page traffic is obtained from each website.

Step 4. Traffic processing: although the collection process reduced the possibility of traffic anomalies, there are potential interference packets. Packet filtering and packet retransmission operations are in place to ensure that the response traffic within each instance is similar to others. There were five types of packets at the TCP level which are involved in packet filtering including: missing, repeated response, hierarchical transmission, window

Input: Mounted website collection (Webm), website public information (Webopen), website privacy information (Webprivacy), website configuration (CF), and website data space (DS)
Output: Server scenario (Sce)
Steps:
(1) $(\text{Web}_m, \text{Web}_{\text{open}}, \text{Web}_{\text{privacy}}) \longleftarrow \text{GetImformation}(\text{Web}_m)$
(2) Records $\longleftarrow$ WebsiteConfig(Records)
(3) **for** each Website $w \in \text{Web}_m$ **do**
(4) $w.\text{CF} \longleftarrow \text{GetConfig}(\text{Web}_{\text{privacy}}), w.\text{DS} \longleftarrow \text{GetConfig}(\text{Web}_{\text{open}})$
(5) **if** $w.\text{CF} \subseteq$ Classical technology stack **then**
(6) **if** $w.\text{DS} \leq$ Maximum allotted space **then**
(7) **return** Sce $\longleftarrow w.\text{SS} - \text{MW}$
(8) **else**
(9) **return** Sce $\longleftarrow w.\text{MS} - \text{SW}$
(10) **end if**
(11) **end if**
(12) **if** $w.\text{CF}$ not in Records **then**
(13) **return** Sce $\longleftarrow w.\text{SS} - \text{SW}$
(14) Records $\longleftarrow w.\text{CF}$
(15) **end if**
(16) **end for**

Algorithm 1: Algorithm of website mounting.

Table 3: Same-source and multisource mounting mechanism.

Server size	Scenario	Website × server/group	Rounds
Same source	SS-SW	1 × 1	6457
	SS-MW	1 × 4	3579
Multisource	MS-SW	150 × 1	122

update, and ACK loss packets. The transmission packets with zero data length are also removed to directly filter packets that interfere with packet classification without providing reliable information. Packet retransmission improves the property value gap between the website instances. Winsorization [28], an effective method for routine detection of abnormal values in sample data, is also adopted to detect the site response duration property. The duration larger than 95 quantiles in the property is defined as an outlier, and revisiting is required to complete the instance. About 17% of the packets on the website are processed, and 6% of the page instances need to be retransmitted.

Step 5. Data conversion: to facilitate the data input of the instance which is interpreted by the classifier, the demand data are converted into a combination of symbols and values, including direction (d), time (t), and packet length (l) that are input in the form of $d = \{-, +\}$. The specific conversion formula is as follows:

$$\text{Dump}(\text{TCP}) = \left(d_1(t_1, l_1), d_2(t_2, l_2), \ldots, d_n(t_n, l_n)\right). \tag{2}$$

5. WRFP Attack

5.1. Response Time Features. Here, we investigate the effective function of the response packet of the website. The combination of response time features is then proposed to highlight the implicit information hidden in the response time and response time frequency features. The measurement of hidden service response time is illustrated in Figure 5. The attacker monitors encrypted TCP packets locally, records the standard time of each incoming or outgoing packet, and then calculates the time difference of the two-way packet group, that is, the difference between the start time of a set of outgoing server packets and the end time of the incoming packets, i.e., $\Delta\text{RT} = \text{IT} + \text{OT}$.

For the response time measurement specified in Algorithm 2, the attacker records the specific TCP packet time and calculates the response time frequency as well as the binning information. Specifically, steps 2–8 indicate that a set of continuous incoming packet time (IT) and continuous outgoing packet time (OT) is regarded as a response time unit, and the difference between the start time (T.start) and the end time (T.end) of the response is considered as the response time (ΔRT). The preset number of bins (RT.Bins) is included in step 9. Steps 10–16 indicate that the response time is looped into the response frequency bins according to the specified range (Range), and the output is the response time frequency array (ΔRT.F).

Figure 6 is a visual display of the response time frequency array. The internal response time of the two popular websites is calculated, and the global time is inserted into 20 bins. The upper layer of Figure 6 summarizes the response time of the forum website http://kbhpodhnfxl3clb4.onion. There is one bin with response time of 20 ms, and the response time

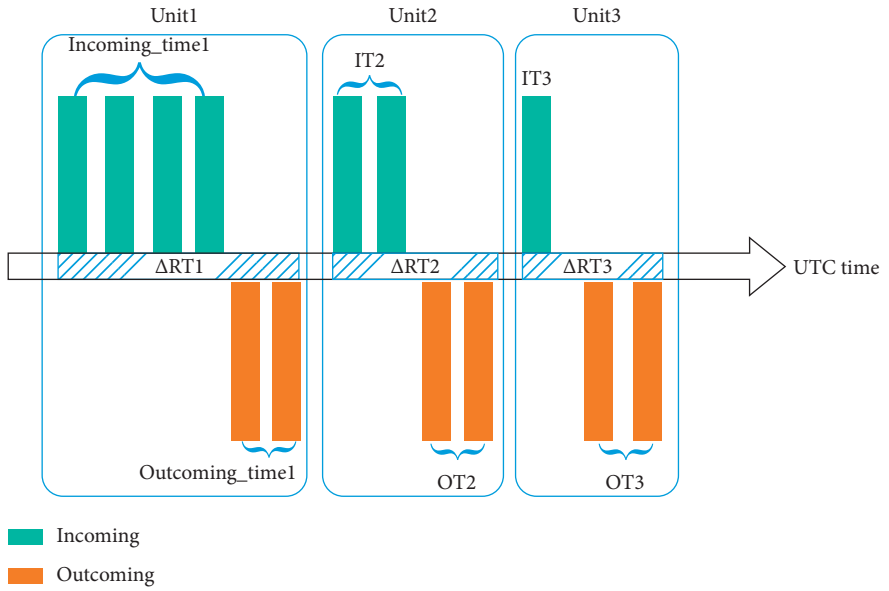

Figure 5: Measurement of the response time.

Input: the continuous incoming and outgoing TCP packet collection (T), response time frequency bins (RT_Bins)
Output: Response time (ΔRT), response time frequency array (ΔRT.F)
Steps:

(1) T.start ⟵ T_1.time, Range ⟵ 0, RT.bins ⟵ bins
(2) **While** $i \le$ Length (T) **do**
(3) **if** $T[i]$.in is outgoing packet and $T[i+1]$.out is incoming packet **then**
(4) T.end ⟵ $T[i]$.time
(5) **return** Add T (ΔRT) ⟵ T.end − T.start
(6) T.start ⟵ $T[i+1]$.time
(7) **end if**
(8) **end**
(9) Timerange ⟵ (Max (ΔRT) − Min (ΔRT))/RT.bins
(10) **for** $j \le$ RT.bins **do**
(11) **if** Range ≤ ΔRT ≤ Timerange + Range **then**
(12) ΔRT.F[j] + +
(13) **end if**
(14) Range ⟵ Timerange + Range
(15) **return** ΔRT.F
(16) **end for**

Algorithm 2: Response time measurement algorithm.

220–240 ms appears three times. The lower layer of Figure 6 displays the response time of the news website http://3g2upl4pq6kufc4m.onion, for which the minimum response time is 53 ms, the maximum response time is 669 ms, and the frequency of the 3 bins is 4 times. Compared with what is in the upper layer of Figure 6, the response time density is more concentrated in the lower layer. This suggests that the response frequency of the website represents the difference between the response of the server and the load behavior of the website. It is therefore inferred that this kind of feature can be used to extract the response classification information of different websites.

5.2. Feature Selection. It is pointed out in a number of studies [7–9] that the accuracy and robustness of website fingerprinting attacks are strongly correlated with the demand features. It is also noted that with the development of website fingerprinting attack technology, statistical features such as incoming, outgoing, burst, and total number of packets are capable of improving the performance of the attack model. It is however suggested that time characteristics are fragile and no consensus has been reached about time [29]. Hence, it is an immediate need to verify the importance and inevitability of response time features in website response fingerprinting, according to

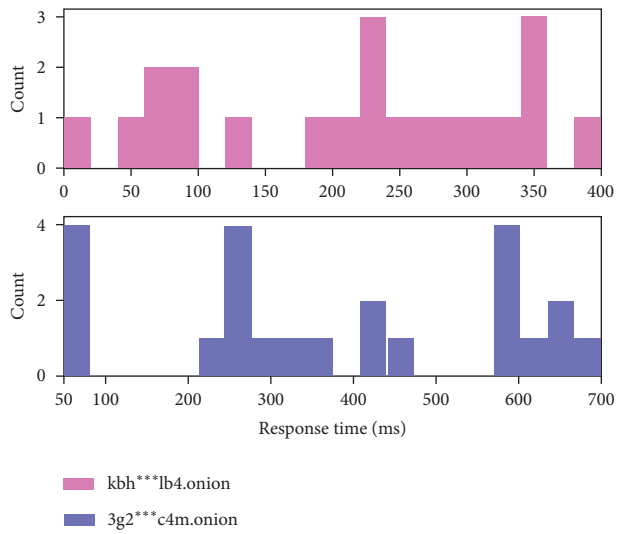

Figure 6: Visualization of the website response time frequency.

the ranking and recognition results of many kinds of features.

For feature weight ranking, we use an extremely randomized tree regression classifier on the response traffic label dataset. To obtain the best feature combination, the overall feature ranking is carried out according to the degree of influence of each preselected feature on the model prediction result.

The traditional feature classification standard is based on Gini or information entropy coefficient [30] which may ignore property overlap of some of the features. In a group, if a feature variable is set as a strong predictive variable, the importance of its associated feature variable is decreased. Note that most of fingerprinting traffic features are, in fact, combined features and tend to include multicategory variables. Therefore, in the feature ranking, the importance of excellent features might be reduced. To address this issue, we select R-square mean as the classification standard to ensure the importance of a single feature and the fair ranking of the excellent features. R-square mean measures the basic accuracy of the feature reduction model, which is regarded as the remeasurement of the model accuracy by inverting a single feature value. For instance, a feature score of 0.02 means that, after replacing the feature data, the accuracy of the attack model is reduced by 2%. As far as the results are concerned, for low-importance feature variables, an arbitrary change of the value has a slight impact on the accuracy of the model, while a disturbance on an important feature leads to a significant decrease in the model accuracy. In this study, the features are divided into main features and effective features according to their score, and the measurement range is the mean score as follows:

$$\text{main} \geq \left[\frac{\sum_{i=1}^{j} \text{scores}[i]}{j} \times 100\%\right] > \text{effective} \geq 0. \quad (3)$$

The main features have a range of score greater than 0.02, and the score of the effective features is in the range 0 and 0.02. For features with a score smaller than 0, the prediction of the attack model is biased towards the worse result.

Figure 7 shows the main feature ranking corresponding to the response size of multisource and same-source servers after 50 raking rounds of 87 features, with 25 main features for multisource server and 23 for same-source server. The importance of each feature is not exactly the same between the two sizes. The results show that in the two response sizes, the number of packets and response time play a leading role in judging the attack, and the number of packets, sequence information, and response time information is the primary judgment basis of the attack. Among the load key information of the total number of incoming and outgoing packets, the total number of outgoing packets ranked first, with mean scores of 0.18 and 0.16. The proportion of incoming packets ranked in the top 7 in both sizes, with scores greater than 0.1. Meanwhile, 15 burst packet features have the largest changes, which are ranked the 10^{th} with a score of 0.067 in the same-source server size, while they are ranked the 19^{th} in the multisource server size, with a score of 0.034.

(i) Response time features: in the multisource server size, the mean score of the total response time is 0.17, which is only lower than that of the total number of outgoing packets (rank 8) and the standard deviation (rank 11). The response time of the first 20 packets ranked between 12 and 19, and the total response time of the last 20 packets ranked 20 with a score of 0.031.

(ii) Response time frequency features: in the same-source server size, the total number of response time frequency is ranked 5 with a score of 0.13, in line with the expectation. The standard deviation of response time frequency is ranked 6 with a score of 0.1. The mean response frequency is ranked 23, with a score of only 0.02. The standard deviation feature of the 5 bins with minimum response time frequency is ranked 11 with a score of 0.058.

The first 32 valid features of the same-source server are shown in Figure 8. Burst packets and transmission time provide auxiliary support to attack classification. The total transmission time of outgoing packets is ranked 28, with a score of 0.017, which was 1/10 of that of the total response time. The main reason is that the website response time has a higher stability and reliability than the distance transmission time.

Figure 9 summarizes the data transmission process. The transmission intervals corresponding to response time ΔT_3 and ΔT_4 are short, and the processing is simpler than that of the distance transmission response time ΔT_1 and ΔT_2. This is consistent with the view of Hayes [9] that the statistical features of packet interarrival times slightly improve the attack accuracy, and the information disclosure ability of the response time is much higher than that of the packet interarrival times.

The number of selected features is an important factor. Hayes [9] shows that the first 30 of the 150 features ensure 90% accuracy of the classifier. Panchenko [8] optimizes the

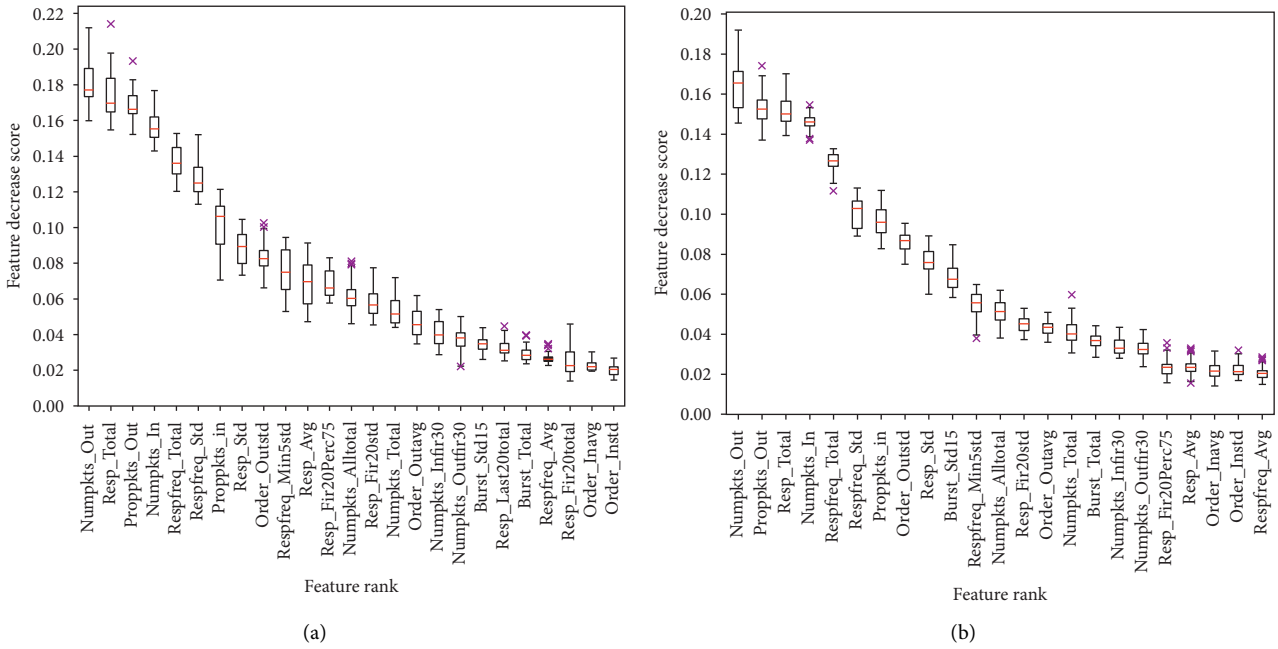

Figure 7: Rankings of main features of the multisource and same-source servers response sizes. (a) Main features of multisource server. (b) Main features of same-source server.

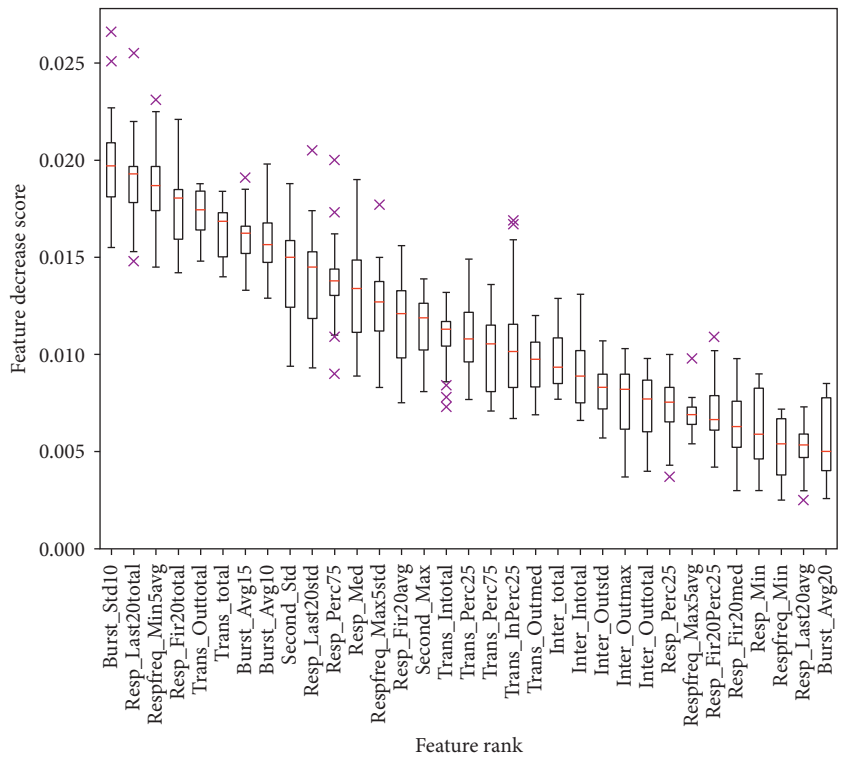

Figure 8: Ranking of the first 32 valid features of the same-source server (score >0.012).

number of cumulative features to 100 to improve the efficiency of CUMUL. These show that an excessive number of features are inversely proportional to the classification return, and further, the optimization of the number of features is able to maximize the accuracy of the model within the demand range. To determine the optimal number of features for the performance of the classifier, the changes in the number of features (started from 10) and the model accuracy variations of the two sizes are compared. The changes in and the relationship between the number of

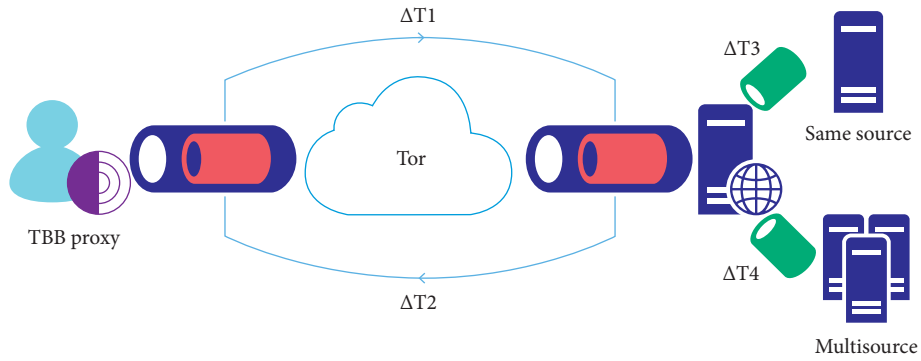

Figure 9: Data communication and transmission.

features and the accuracy are displayed in Figure 10. The accuracy of the first 24 features is increased rapidly, which is inline with the corresponding increase in the ranking degree of the main features. However, the accuracy is oscillated for the number of features larger than 70 in both same-source and multisource server sizes. This suggests that a reasonable choice for the number of the features is 73 or 82. Accuracy greater than 94% also meets the needs of the attack, and the optimized numbers prevailed in the subsequent tests.

5.3. Process of Response Attack. Here, we propose a set of standard WRFP attack processes which follow the basic construction phases of fingerprinting classifier including response attack training phase and response attack matching phase. The response attack training phase is divided into the training and WRFP classifier performance testing. The training process is shown in Figure 11 and includes the following steps. Step 1: the response traffic of the website mount image is collected. Each mirror site repeatedly responds for 50 rounds to obtain the original TCP fingerprinting dataset. Step 2: features are extracted from the dataset, labeled, and bound to the website page or server to form a feature tag library that is used in the subsequent training steps, as well as the attack matching phase. Step 3: The extremely randomized tree classifier with a training to test ratio of 8 : 2 is then applied to the website response fingerprinting dataset to build a hidden service website response fingerprinting classifier. Step 4: Cross-validation of the test set is carried out. Step 3 is repeated until the verification classification result is at its highest value. The attack model is considered as the final WRFP classifier.

It is stipulated that the trained WRFP classifier is capable of conducting response fingerprinting attacks. This because the attacker has the ability to monitor the locally encrypted TCP traffic. The attack matching process is shown in Figure 12. Firstly, the attacker marks the locally encrypted TCP traffic. Secondly, the statistical information of the load packet is extracted and converted into the response eigenvalues. Finally, after accumulating the packets observing and monitoring the numbers, the extremely randomized tree classifier generates the matching results of the website server, including the website tags matched under the same-source and multisource server sizes, as well as the website server that provides mounting.

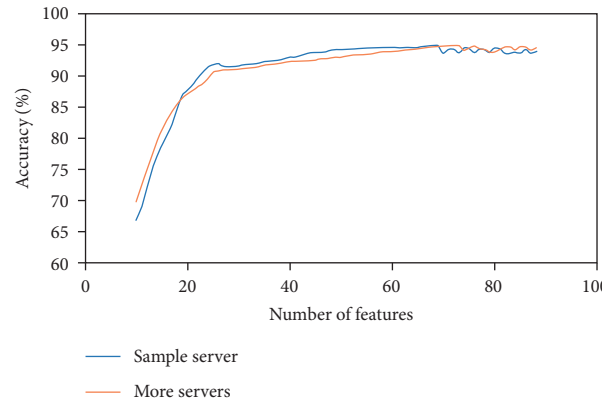

Figure 10: Changes in, and the relationship between the number of features and the accuracy.

Here, the extremely randomized tree [31] is used which is different from the random forest classifier in the way of selecting the division points in a single decision tree. The random feature splitting method is added to randomly obtain bifurcation values. This random feature selection greatly reduces the amount of training computation. The extremely randomized tree randomly selects the samples when constructing the data subset and further randomly extracts the features of the samples. In other words, when building the model, part of the features is used for training. Comparison of the classification performance between the extremely randomized tree and random forest is shown in the first two rows of Table 4. As it is seen, the test speed of extremely randomized tree is improved by 102%.

6. Experimental Results and Analysis

6.1. Evaluation of Dataset. To realize the rationality of evaluation, we use a hidden service response fingerprinting dataset collected in real network environment. The sizes and properties of the datasets are presented in Table 5, and the website size is expressed as the number of website pages multiplied by the number of fingerprint instances. In the results of dataset classification, the training effectiveness of the classifier is based on same-source and multisource server response fingerprints. The response fingerprints of 90,000 independent pages are collected, with 50 instances for each

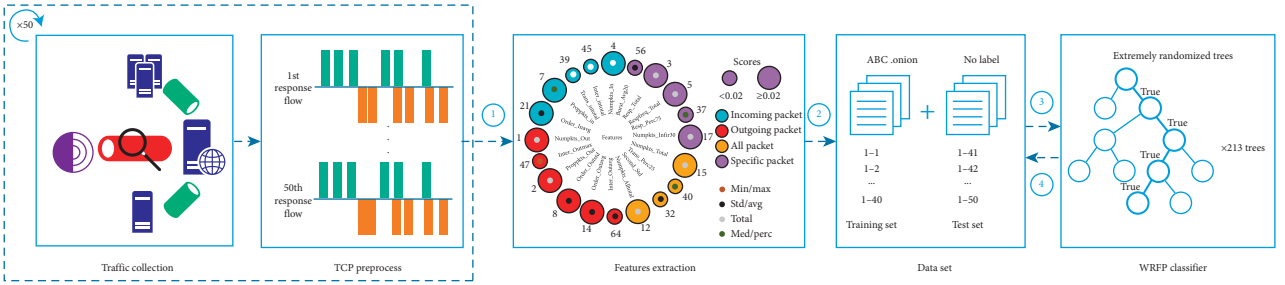

Figure 11: Training process of the WRFP classifier.

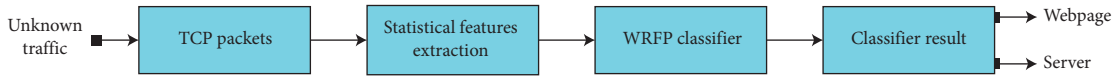

Figure 12: WRFP attack matching process.

Table 4: Performance comparison of different algorithms in the classifiers.

Classifier	Algorithm	Accuracy (%)	Test time* (hour)	Parameters	Features	Memory** (GB)	Page
WRFP	Extremely randomized tree	93.63	0.43	$n = 213$	87	8	10,000
	Random forest	91.82	0.87	$n = 487$	87	8	10,000
K-FP	Random forest + Nearest neighbors	92.35	1.14	$K = 5$, $n = 379$	186	65	10,000
K-NN	K-nearest neighbors	88.14	1.65	$K = 5$	1225	10	10,000
		88.06	4.88	$K = 5$	3736	16	10,000
CUMUL	SVM-RBF	90.78	0.55	$C = 4096$	150	30	10,000

*Test time refers to the instance feature extraction time + page classification time. **Memory denotes the minimum required memory size.

Table 5: Classification of datasets needed for evaluation.

Dataset	Data description	Data size	Website size	TBB version
Same-source server	Response fingerprinting of same-source server	$40{,}000 \times 50$	11,500	8.5
Multisource server	Response fingerprinting of multisource server	$50{,}000 \times 50$	15,000	8.5
Background	Response fingerprinting of big background	$200{,}000 \times 10$	25,000	8.5
TBB version	Response fingerprinting of multiple TBB versions	4000×50	100	6.5/7.5
Defense	Fingerprinting of multiple types of defenses	8000×50	100	8.5

page. Besides, a big background response fingerprinting is constructed to facilitate the understanding the recognition effect of the classifier in open world. Response fingerprints of a total of 200,000 pages are collected, covering 25,000 websites. To evaluate the impact of TBB version and website fingerprinting defense on the classifier, we collect 200,000 response fingerprints of version 6.5 and 7.5, with 2000 (pages) × 50 (instances) for each defense model.

6.2. Evaluation Indexes. The test evaluation indexes proposed in this study are aimed at the evaluation of the response fingerprinting classifier in multiple scenarios. In addition to the conventional accuracy of classification, other indexes provide practical insights on performance of the classifier.

6.2.1. Precision and Recall. Precision refers to the probability of recognizing correctly classified monitored pages. Recall refers to the probability that the monitored pages are classified as correctly monitored pages. Precision and recall affect each other. In tests and evaluation, a high precision is preferred to reflect the correct classification effect of the classifier under a big background.

6.2.2. True Positive Rate (TPR) and False Positive Rate (FPR). TPR is equivalent to recall, while FPR refers to the probability that unmonitored pages are misclassified as monitored pages. The combination of the two indexes shows the performance of the classifier; however, if they are affected by the fallacy of the basic rate, these indexes become bad references.

6.2.3. Bayesian Detection Rate (BDR). BDR refers to the probability that a given classifier can correctly judge a monitored page when it is recognized as a monitored page. This index includes the ratio of the monitored pages to the total pages, to a certain extent eliminating the classification influence caused by the basic rate fallacy in open world, and highlighting the correct classification results in the big

background. Juarez [14] and Hayes [9] apply BDR in their study for the first time and show that this index is feasible:

$$\begin{aligned} \mathrm{BDR} &= \frac{\mathrm{TPR} \times \Pr(M)}{\mathrm{TPR} \times \Pr(M) + \mathrm{FPR} \times \Pr(U)}, \\ \Pr(M) &= \frac{|\text{monitored}|}{|\text{total pages}|}, \\ \Pr(U) &= 1 - \frac{|\text{monitored}|}{|\text{total pages}|}. \end{aligned} \tag{4}$$

6.3. Closed-World Test. In this section, the performance of WRFP classifier is tested in closed world. The classification results of monitoring a small number of hidden service websites is simulated, and the response fingerprinting of same-source or multisource servers is judged. A total of 300, 600, and 900 website pages were monitored, with 40 instances for each page as the foreground training set and the other 10 instances as the test set. The background pages are fixed with $10{,}000 \times 10$ random pages, and ten-fold cross-validation is performed to ensure the rational use of the dataset.

The test results are shown in Table 6. The accuracy of monitoring 300 pages is 96.7%, with a TPR higher than 94%. By increasing the number of monitored pages to 600, the accuracy declined by 2%, indicating that if the attacker wants to get the best recognition results, the attacker needs to control the number of monitored pages. In monitoring 600 multisource pages, it is promising that the FPR reaches 0.4%. By comparison, it is seen that the accuracy of recognizing same-source pages is 2% higher than that of recognizing multisource pages. This however does not mean that the classifier has a weaker ability to classify multisource servers. Instead, it may be attributed to data jitter in multisource servers which may cause deviation in the correct classification of multisource pages.

The distribution of response time frequency is determined by the bin value, and the setting of bin value affects the granularity that divides time frequency. Therefore, the selection of bin value affects the accuracy of the classifier. In this study, the bin value is set as {5, 10, 15, 20, 30}. The results of the ROC test with 600 monitored pages are given in Figure 13. It is seen that for bin = 5, the accumulation of response frequency simplified the distinction between websites and the FPR is 43%. For bin = 30, the website response time is finely differentiated, and the frequency span is excessively lengthened such that the website classification features are mixed, resulting in a TPR of only 80%. The area under curve (AUC) for the case with bin = 10 and bin = 20 is 0.9505 and 0.949, respectively. This is a rough indication of similar performance of the two classifiers. Through specific measurement, it is also seen that, for bin >15, the AUC of the increased bin value shrinks and the classifier has a diminishing return. Based on the results of several candidate values, the bin value is finally set as 15.

In the present study, websites servers that provide the specified hidden services are monitored. In other words, the website information provided by the servers is monitored, which could alleviate the confusion from the website internal pages. Under the background of 10,000 websites, the number of websites to be recognized increased from 50 to 1000. As it is seen in Figure 14, the FPR of classifying 1000 websites is only 1.5%. For a website size of greater than 4%, the TPR is maintained above 93%. In the face of the growing number of monitored websites, the recognition performance of the classifier is excellent as it does not decline slightly as in the case of page recognition. The index of precision emphasizes more on the overall classification of monitored websites. Monitoring 100 websites, the precision is 69.4%, with 30 monitored websites unrecognized. Therefore, the effect of WRFP classifier for monitoring small-size websites is not satisfactory. Increasing the size of monitored websites by 10%, the precision is increased from 69.4% to 86.1%. The overall classification results manifest that when the attack target is turned to the server, the classification becomes easier and the effect of monitoring large-size website servers is excellent.

A total of 40 page instances are used as the training set in the above test. According to the previous experiences, expansion of the training set can easily improve the performance of the classifier. Whether or not a small number of training sets can keep the performance of WRFP classifier within an acceptable range is investigated in this study. The background is 10,000 pages. The instances in each page is split into training sets and test sets as shown in Table 7, and ten-fold cross-validation is performed. For 10 training instances, the TPR is reduced to the lowest (90%), while the FPR is 2.5%. The recognition accuracy gaps of 20 and 30 instances with 40 instances is also acceptable (less than 1%). As expected, the classification result is affected by reducing the size of the training sets. However, WRFP classifier allows effective page recognition even for small training sets. Of course, 40 page instances should still be used for training to obtain the best WRFP classifier performance.

We also investigate robustness of WRFP classifier against time fluctuations. The response time is expanded proportionally, and the number of monitored pages to 600 and 20 rounds of tests are conducted. As it is shown in Figure 15, time error rates smaller than 5% have no real effect on the accuracy. As the error rate increased to 9%, the accuracy is gradually declined. For a time error rate of 24.5%, the recognition accuracy is dropped below 83.3%, which is the accuracy when the response time feature is removed. Such a decrease seriously affects the attack judgment. Hence, to ensure the balance between WRFP classifier accuracy and time error, time error rate needs to be kept below, and the accuracy rate is steadily higher than 90%, that is, the fluctuation of 2 bins within the response time frequency is acceptable.

6.4. Comparison in Open World. In this section, comparisons are made between WRFP classifier and the traditional website fingerprinting attack classifiers. Although the monitor target of the traditional classifiers and that of WRFP classifier are different, they are all based on fingerprinting

Table 6: Recognition effect of different pages in closed world.

Server information	Page	Accuracy	TPR	FPR
Same source	300	0.967 ± 0.023	0.947 ± 0.008	0.013 ± 0.005
	600	0.964 ± 0.041	0.944 ± 0.013	0.009 ± 0.007
	900	0.952 ± 0.025	0.936 ± 0.016	0.015 ± 0.013
Multisource	300	0.962 ± 0.011	0.936 ± 0.011	0.009 ± 0.009
	600	0.953 ± 0.016	0.935 ± 0.016	0.004 ± 0.008
	900	0.945 ± 0.021	0.930 ± 0.014	0.017 ± 0.012

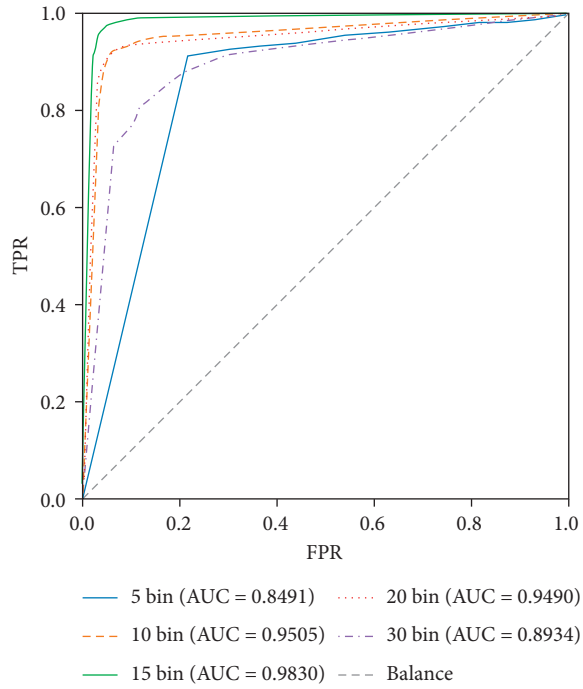

Figure 13: The effect of bin value on response time frequency.

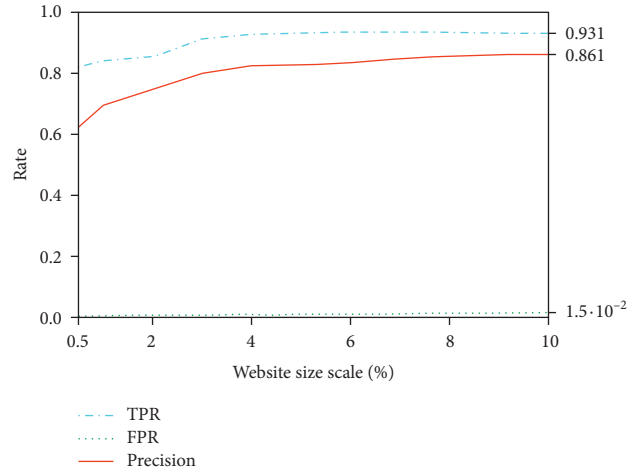

Figure 14: Classification results of monitoring website servers.

Table 7: Recognition result of the reduced training sets.

No. of training instances	No. of test instances	Accuracy	TPR	FPR
10	40	0.923 ± 0.023	0.921 ± 0.021	0.025 ± 0.015
20	30	0.957 ± 0.022	0.931 ± 0.013	0.016 ± 0.013
30	20	0.961 ± 0.035	0.934 ± 0.009	0.015 ± 0.009
40	10	0.961 ± 0.017	0.935 ± 0.009	0.008 ± 0.009

feature recognition at the traffic level and the results are comparable. The traditional models involved in the comparison are CUMUL [8], k-NN [7], and k-FP [9], and the response datasets are uniformly used for classification tests. To understand the actual effect of the classifiers, test data with unbalanced foreground and background are used to increase the challenge of classification. The background size is divided into [2000, 5000, 10,000, 50,000, 100,000, 200,000], the foreground is set as 1000 (pages) × 40 (instances), and 5 rounds of tests are carried out. In addition, the basic traffic is a response packet close to the server, and the data themselves are biased towards the response classifier. To judge the dependence between the result and the data themselves, the traditional website fingerprinting attack classifiers are divided into two states:

(i) Response time features unloaded: the basic classification function of classifiers is discussed based on the maintained original attention features of the classifier.

(ii) Response time features loaded: while keeping the original features, the response time and response time frequency features are added to test the degree of classifier judgment to page classification after loading response time.

The performance of the classifiers in open world is measured with reference to indexes of precision and recall. The classification results of original classifiers are shown in Figure 16. The performance of the classifiers is weakened by increasing background size. However, by increasing the background size from 2000 pages to 200,000 pages, the precision of WRFP classifier is kept at 86% and its recall is declined from 95.2% to 84.1%. This suggests that the WRFP classifier remains effective in recognizing monitored pages under a big background. Under a background of the same size, the precision and recall of k-NN classifier are 58.4% and 53.5%, respectively, and there are about 400 pages that are not correctly classified as monitored pages. The k-FP classifier performs well under a background of 50,000 pages, with precision of 83.8% and recall of 81.1%. However, for a background of 200,000 pages, the recall is 71.2%, and that of CUMUL is 66.4%, both of which are unable to guarantee

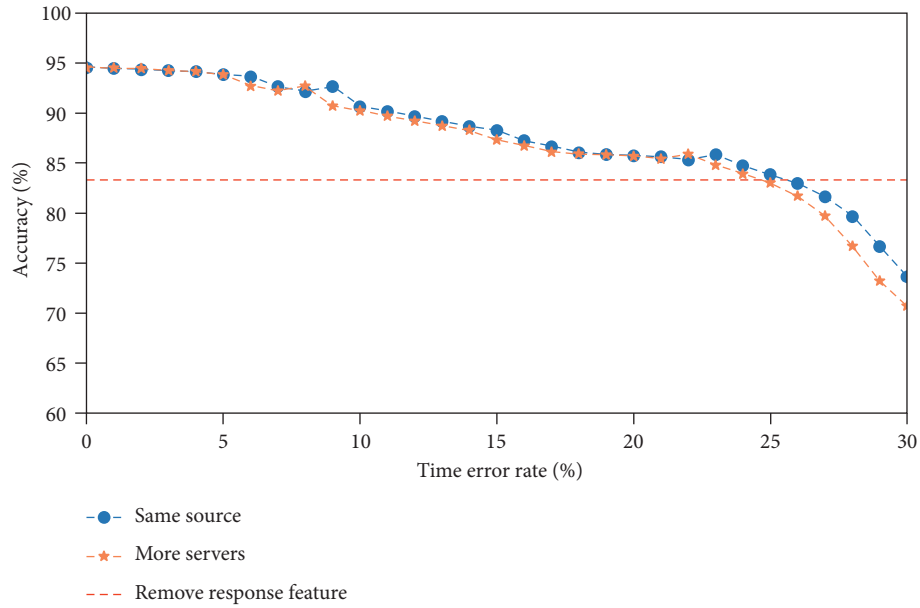

Figure 15: Response time error affected accuracy.

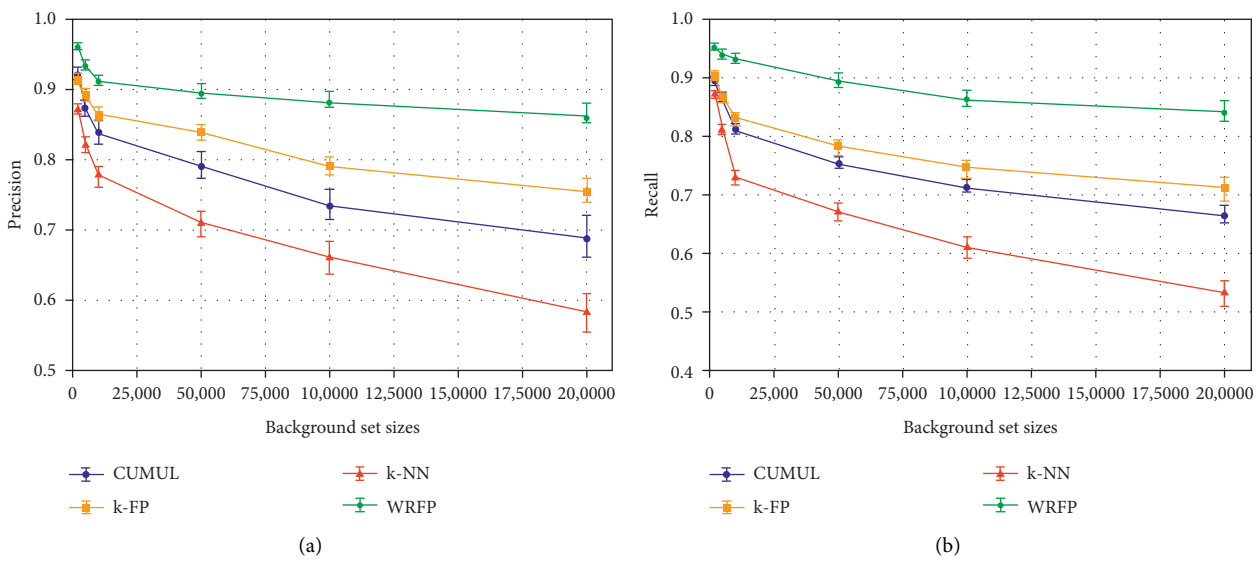

Figure 16: Comparison of test results of the original classifiers. (a) Precision. (b) Recall.

reasonable attack accuracy. The BDR results are shown in Table 8. For the background smaller than 5000 pages, all four classifiers maintain a good and correct page recognition effect. For the background of 50,000 pages, only the WRFP classifier maintained correct page recognition larger than 50%. In the face of unlimited increase of background pages, it is impossible to guarantee the classification accuracy. For example, for the background size of 200,000 pages, less than 20% of them are correctly classified.

The test results of classifiers loaded with response time features are displayed in Figure 17. In contrast with the original classifiers, the precision of k-NN classifier is increased from 58.4% to 72.4% under a background of 200,000 pages, while the recall is increased from 53.5% to 70.5%. The recognition performance of k-FP classifier is also increased slightly, with precision increasing from 75.4% to 82.1% and recall from 71.2% to 81.1%. On the contrary, the recall of CUMUL classifier is only increased by 6%, while the precision is decreased by 2%. This may be due to the fact that CUMUL is biased towards the accumulation of packets and cannot effectively utilize the loaded response time and response time frequency features, which make the recognition skew toward a worse result. It is concluded that the response time features can provide a reliable help to the classifier to judge the response fingerprint.

The results of WRFP classifier recognizing the website servers represented by the pages are shown in Figure 18. Compared with the classification result of a single page,

Table 8: BDRs of classifiers under different background sizes.

Classifier	Background sizes					
	2000	5000	10,000	50,000	100,000	200,000
WRFP	0.987	0.962	0.903	0.658	0.292	0.183
k-FP	0.984	0.944	0.863	0.456	0.219	0.093
k-NN	0.979	0.921	0.776	0.313	0.121	0.042
CUMUL	0.983	0.932	0.813	0.368	0.182	0.081

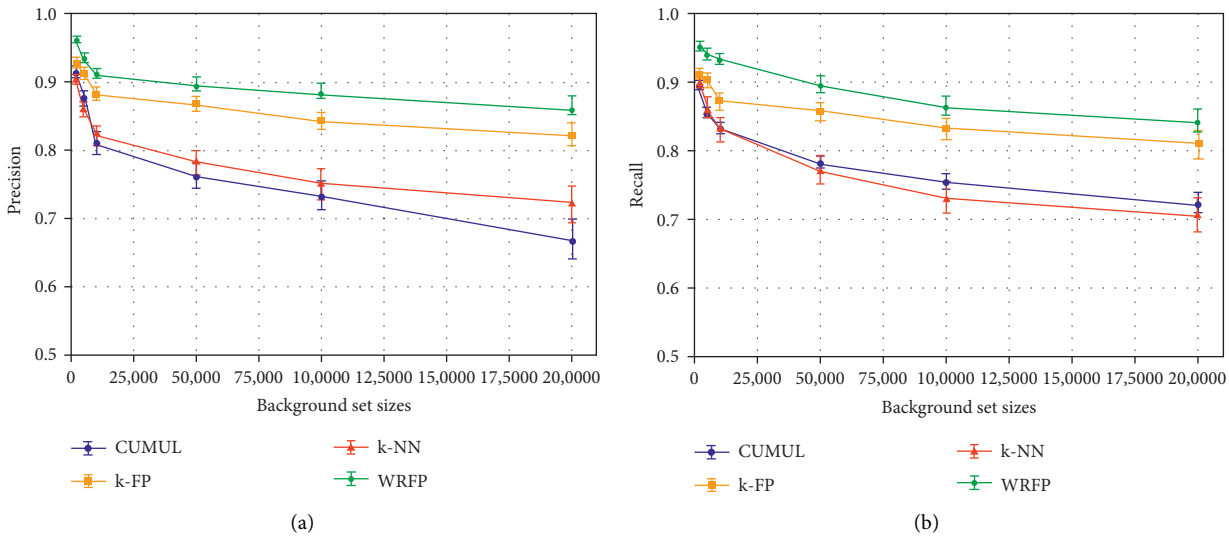

Figure 17: Test results of classifiers loaded with response time features. (a) Precision. (b) Recall.

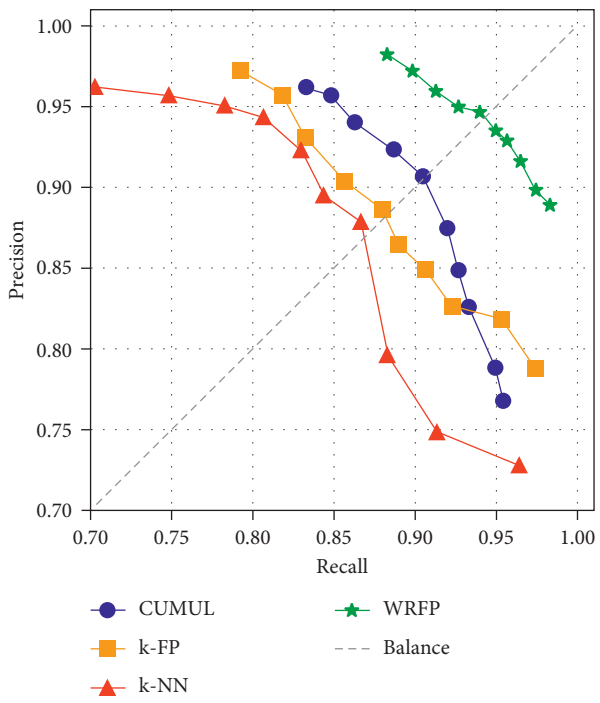

Figure 18: Precision-recall curves of recognizing website servers.

simplification of the scenarios increased the recognition probability. The test is conducted under a background of 26,000 web servers, and 1000 web servers are monitored. When an attacker looks for a balance between precision and recall, WRFP classifier can get the optimal effect of both precision (94.3%) and recall (93.7%). Although the balance value of k-FP classifier is lower than that of CUNUL, the attackers can highlight precision or recall getting the correct recognition effect. The k-NN classifier is not effective in recognizing server scenarios, and for a precision of 94.3%, the recall is only 80.6%.

In addition, the computational efficiency of the classifier is particularly important. The test results of the computational efficiency of the classifiers for classifying 10,000 pages are given in Table 4. Considering the high memory consumption by k-FP classifier, the background is set to 30,000 pages. An Intel Xeon E5-2696 v4 is used as the CPU core in the test. To obtain the optimal accuracy, the false positive pages are ignored and the corresponding optimal parameters of the classifier are selected. The test time includes the algorithm classification efficiency and the frequency of feature extraction and transformation, all of which are linked to the number of features. We test the k-NN classifier proposed by Wang [7]. The model performance is improved by optimizing the number of features (the packet length feature which is useless to the cell is deleted), and the computational efficiency is increased by 310%. However, its computing time is still longer than that of other classifiers. CUMUL classifier is the closest to WRFP classifier in terms of computing time, but its recognition accuracy is 90.7%. k-FP classifier needs a large memory in exchange for higher accuracy.

Our results suggest that WRFP classifier has the two following advantages. First, it needs smaller memory than that of the other classifiers, with a memory usage of 12.3% of that of k-FP. Second, it has a high computational efficiency which is 11 times faster than that of k-NN.

6.5. Subpages of Website Types. In the evaluation experiment in Section 6.4, the test set contained index pages and subpages of the hidden services. The results showed that separate classification of each page causes a great pressure on WRFP classifier, and too many subpages may confuse the judgment results. In reality, the subpages wrap the service content of the website, and there is a certain gap in the service content of each type of website, showing different contours at the traffic level. However, it is based the service profile of the website that the attack against the server determines the server to which the page belongs. Here, the results of WRFP classifier recognizing the subpages of website types of news, porn, forum, and social sites are investigated, and the corresponding recognition strategies are analyzed.

Firstly, the recognition scenario is set to divide the subpages within the website type, and evaluation was made on a small scale to get the most direct recognition effect. For each website type, 8 independent domain name websites each containing 30 different subpage instances are randomly selected. Figure 19 displays the subpage recognition heat map of the website types. The accuracy of the 32 websites is 86%. Among them, the first 8 are subpages of the forum sites, with an accuracy of 90.8%, which is higher than that of the news sites (82.1%). The accuracy of porn sites is also above the mean value. In addition, there exists an interesting observation when observing the incorrectly classified pages. Among 22 confusing page instances in the forum sites, 19 are recognized as forum sites, i.e., there is a high probability that the false positive pages are subpages of the forum sites. Page recognition is conducted in a small environment, and the test pages are independent of each other. It is also seen through specific analysis that the service patterns of forum sites are similar, and these similar patterns improved the possibility of recognition. Therefore, it can be assumed that the website type affects the classifier's judgment of the page.

To investigate the influence of website types on the recognition, the classification of website subpages is focused upward, and the types provided by the server are classified. Based on these, the subpages are recognized for the second time. The original dataset is used in the test, and ten-fold cross-validation is also performed. The classification data of the four types of focused websites are given in Table 9. The classification accuracy of the forum sites is improved to 98.7%, with a TPR of 97.6% and an FPR of 0.04%, and there are only 3 false positive pages. The news sites experience the largest increase, with an accuracy of 93.5%. Most of the main service contents in porn websites are pictures, which makes traffic fingerprinting special and different from other types of websites. The results indicate that, in monitoring servers providing specific website types, fixed-point training with more pages of relevant types makes up for the shortcoming of false positive page recognition in terms of attack strategies.

6.6. Client Traffic and Defense Confrontation. Different versions of Tor Browser Bundle (TBB) can be run at the client. In this section, we investigate whether or not the fluctuation of server response fingerprinting caused by different TBB versions can affect the classification efficiency. There are small differences between different versions of TBB. During the communication process of TBB v6.5 (core Tor 0.2.9), the client is equipped with multiple ingress nodes. TBB v7.5 (Tor 0.3.2) provides third-generation service response request. TBB v8.5 (Tor 0.4.0) supports traffic adaptive filling defense mechanism. (2000 (websites) $\times$ 20 (instances)) $\times$ 3 groups of website instances are used for evaluation, and 10,000 pages samples are used as the background. Considering that TBB v6.5 can only access v2 websites, v3 addresses are removed from the dataset. The defense mechanism is not added to TBB v8.5 at this stage. We consider separate training sets for each version; the other two versions are used as the test sets. We fix the basic properties of TBB, UseEntryGuards is set to disabled state, and new Guard nodes are enabled for each link communication to ensure the freshness of the link traffic.

The response recognition results generated under the 3 TBB versions are presented in Figure 20. Surprisingly, the TPR value fluctuates between 91% and 96%, and the peak of FPR is 1.6%, suggesting that the version difference is not reflected in the response fingerprint, and WRFP classifier can simply ignore the version change of Tor Browser. TBB v8.5 has the best classification effect, and the attacker can use the current highest version as a training set by default (version 9.0 or above has been released when this paper is published, but the core is still based on Tor 0.4), so there the TBB version has no impact of the previously analyses.

It can be seen from the above conclusion that the TBB change does not affect server recognition. Therefore, the test is extended to evaluate the impact of the client defense model on WRFP classifier. Comparison with the classical fingerprinting defense of CS-BuFLO [10], Tamaraw [11], WTF-PAD [12], and ALPaca [13] is helpful to understand the actual impact of the existing defense technology in resisting the server response fingerprinting attack. In this test, we process 2000×50 instances by each defense to obtain a defense fingerprinting set with its own tags. The defense test results of the classifier are presented in Table 10, which includes the overhead resources for statistical processing.

It can be seen in Table 10 that CS-BuFLO and Tamaraw control the packet sending rate and disrupt the response time reception rhythm. These defense models consume more than 140% of the bandwidth and double the data delay, resulting in a TPR of only 4% and an FPR of 70% for WRFP classifier. For WTF-PAD, the classifier has a TPR of nearly 80% and an FRP of 7%, which is better than the actual effect of lightweight defense. It is also observed that WTF-PAD is unable to break the division of bin in response time frequency by padding the traffic gap. It is worth noting that, even when recognizing the ALPaca fingerprinting data specially provided for server defense, the recognition accuracy of TPR is 56.5% under same-source

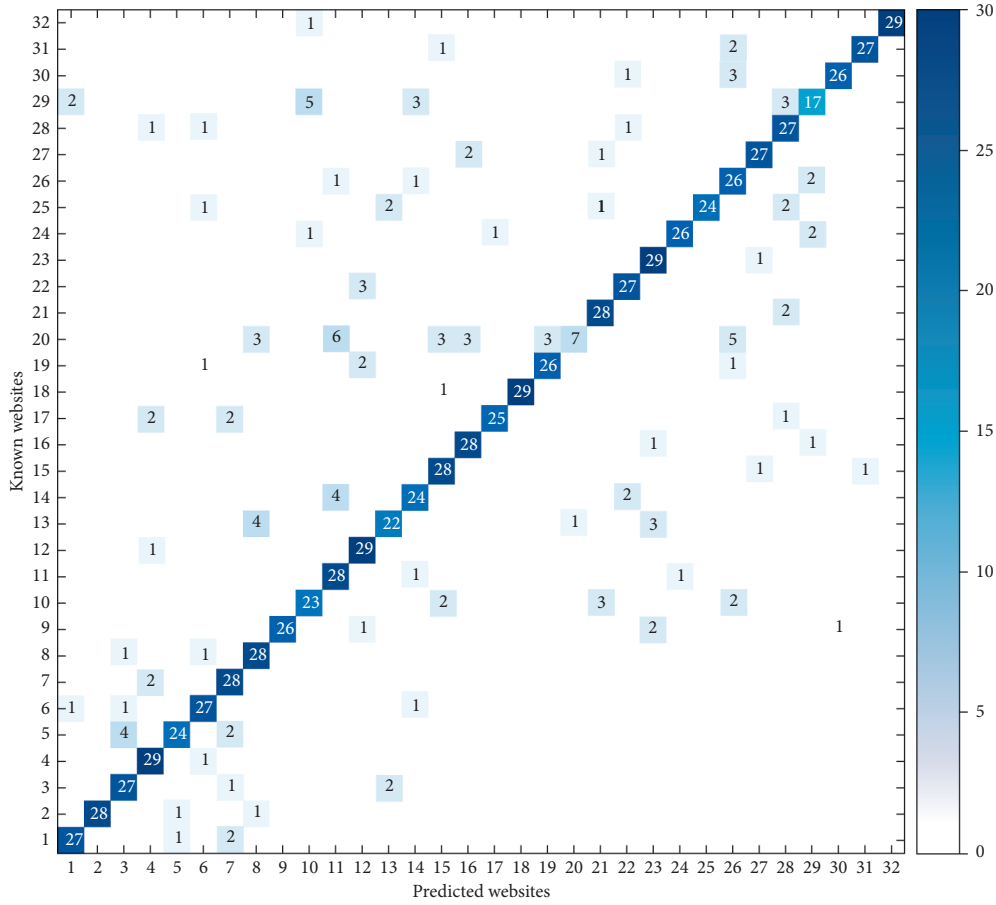

Figure 19: Subpage classification heat map of four types of websites.

Table 9: Classification results of subpages of focused website service types.

Website type	Accuracy	TPR	FPR
Forum	0.987 ± 0.005	0.976 ± 0.007	0.004 ± 0.005
News	0.935 ± 0.024	0.921 ± 0.023	0.013 ± 0.008
Social	0.948 ± 0.014	0.935 ± 0.016	0.021 ± 0.013
Porn	0.912 ± 0.016	0.905 ± 0.013	0.016 ± 0.004

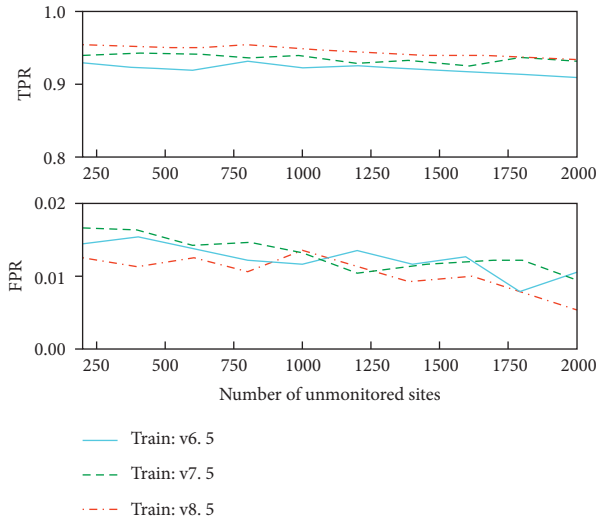

Figure 20: Fluctuation test of TBB versions to response fingerprinting attack.

Table 10: Comparison of test results among defense models.

Defense	Overhead		Same server		Multiserver	
	Bandwidth (%)	Delay (%)	TPR (%)	FPR (%)	TPR (%)	FPR (%)
CS-BuFLO	149	116	4.4	75.8	7.4	72.3
Tamaraw	144	102	4.1	69.8	6.6	66.7
WTF-PAD	22	17	79.5	7.8	81.3	8.3
ALPaca	56	45	56.5	23.3	59.3	19.6

server while consuming 56% of the bandwidth and 45% of the delay. Interestingly, the recognition effect of multisource server is always better than that of the same-source server. This indicates that multisource response fingerprinting is less affected by defense. Our results suggest that the defense model of traffic padding can compete with classifiers, but at the expense of a large bandwidth and an extended delay. In the face of lightweight defense model, it is promising that WRFP classifier is capable of maintaining recognition results with a high precision.

7. Conclusion and Future Works

In this paper, a WRFP attack technique based on response time features was proposed. A hidden service response fingerprinting dataset was constructed, and the basic performance of WRFP classifier was tested based on the extremely randomized tree and the response time measurement standard. The experimental evaluation revealed that, in closed world, both same-source server and multisource server achieve a better accuracy in traffic recognition, and even if the training set is reduced by half, the original accuracy will not be reduced. In the open world with a large size gap between the foreground and background, it was shown that the response fingerprinting classifier is more efficient in terms of accuracy and computational efficiency compared to the previous fingerprinting classifiers based on traditional manual features. In addition, the disturbance caused by factors of TBB versions and fingerprinting defense was considered and analyzed, and the stability and effectiveness of the classifier were confirmed. The test results of subpages showed that WRFP classifier is able to effectively focus on the classification of different website types, and with the increase of subpages, its recognition effect of subpages will not lag behind that of the index pages.

The traffic fingerprinting recognition in response to the hidden service which was proposed in this paper is different from the conventional website fingerprinting attack scenario, thus introducing new challenges in traffic analysis and attack standards. In the future, we will carry out in-depth research on website servers in different geographical locations and take steps to integrate a deep learning algorithm to improve the performance of the classifier in presence of extra noise interference.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key R&D Program of China (Nos. 2019QY1302 and 2019QY1305).

References

[1] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: the second-generation onion router," in *Proceedings of the 13th USENIX Security Symposium*, pp. 303–320, San Diego, CA, USA, August 2004.

[2] B. Zantout and R. Haraty, "I2P data communication system," in *Proceedings of the 10th International Conference on Network*, pp. 401–409, Valencia, Spain, May 2011.

[3] J. A. Elices and F. Pérez-González, "Locating Tor hidden services through an interval-based traffic-correlation attack," in *Proceedings of the IEEE Conference on Communications and Network Security*, pp. 385-386, Washington, DC, USA, October 2013.

[4] A. Biryukov, I. Pustogarov, and R. P. Weinmann, "Trawling for tor hidden services: detection, measurement, deanonymization," in *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, pp. 80–94, Westin St. Francis, San Francisco, CA, USA, May 2013.

[5] Z. Ling, J. Luo, K. Wu, and X. Fu, "Protocol-level hidden server discovery," in *Proceedings of the 2013 IEEE INFOCOM*, pp. 1043–1051, Turin, Italy, April 2013.

[6] S. Matic, P. Kotzias, and J. Caballero, "CARONTE: Detecting location leaks for deanonymizing tor hidden services," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1455–1466, Denver, CO, USA, October 2015.

[7] T. Wang, X. Cai, R. Nithyanand, R. Johnson, and I. Goldberg, "Effective attacks and provable defenses for website fingerprinting," in *Proceedings of the 23rd USENIX Conference on Security Symposium*, pp. 143–157, San Diego, CA, USA, August 2014.

[8] A. Panchenko, F. Lanze, A. Zinnen et al., "Website fingerprinting at internet scale," in *Proceedings of the 23th Annual Network and Distributed System Security Symposium*, pp. 1–15, San Diego, CA, USA, February 2016.

[9] J. Hayes and G. Danezis, "K-fingerprinting: a robust scalable website fingerprinting technique," in *Proceedings of the 25th USENIX Security Symposium*, pp. 1187–1203, Austin, TX, USA, August 2015.

[10] X. Cai, R. Nithyanand, and R. Johnson, "CS-BuFLO: a congestion sensitive website fingerprinting defense," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, pp. 121–130, Scottsdale, AZ, USA, November 2014.

[11] X. Cai, R. Nithyanand, T. Wang, R. Johnson, and I. Goldberg, "A systematic approach to developing and evaluating website fingerprinting defenses," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 227–238, Scottsdale, AZ, USA, November 2014.

[12] M. Juarez, M. Imani, M. Perry, C. Diaz, and M. Wright, "Toward an efficient website fingerprinting defense," *Computer Security*, pp. 27–46, Heraklion, Greece, September 2016.

[13] G. Cherubin, J. Hayes, and M. Juarez, "Website fingerprinting defenses at the application layer," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 2, pp. 186–203, 2017.

[14] M. Juarez, S. Afroz, G. Acar, C. Diaz, and R. Greenstadt, "A critical evaluation of website fingerprinting attacks," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 263–274, New York, NY, USA, November 2014.

[15] T. Wang and I. Goldberg, "On realistically attacking tor with website fingerprinting," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 4, pp. 21–36, 2016.

[16] D. Herrmann, R. Wendolsky, and H. Federrath, "Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial Naïve-Bayes classifier," in *Proceedings of the 2009 ACM Workshop on Cloud Computing Security*, pp. 31–42, Chicago, IL, USA, November 2009.

[17] X. Cai, X. C. Zhang, B. Joshi, and R. Johnson, "Touching from a distance: website fingerprinting attacks and defenses," *y*, in *Proceedings of the 2012 ACM Conference on Computer and Communications Securit*, pp. 605–616, Raleigh, NC, USA, October 2012.

[18] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy*, pp. 253–262, Porto, Portugal, February 2017.

[19] A. Kota and G. Shigeki, "Fingerprinting attack on tor anonymity using deep learning," *Proceedings of the Asia-Pacific Advanced Network*, pp. 15–20, 2016.

[20] V. Rimmer, D. Preuveneers, M. Juárez, T. V. Goethem, and W. Joosen, "Automated website fingerprinting through deep learning," in *Proceedings of the 25th Annual Network and Distributed System Security Symposium*, 2017, https://arxiv.org/abs/1708.06376v2.

[21] P. Sirinam, M. Imani, M. Juarez, and M. Wright, "Deep fingerprinting: undermining website fingerprinting defenses with deep learning," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1928–1943, Toronto, Canada, October 2018.

[22] S. Bhat, D. Lu, A. Kwon, and S. Devadas, "Var-CNN: a data-efficient website fingerprinting attack based on deep learning," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 4, pp. 292–310, 2019.

[23] T. Pulls and R. Dahlberg, "Website fingerprinting with website oracles," *Proceedings on Privacy Enhancing Technologies*, vol. 2020, no. 1, pp. 235–255, 2020.

[24] S. Zander and S. J. Murdoch, "An improved clockskew measurement technique for revealing hidden services," in *Proceedings of the 17th Conference on Security Symposium*, pp. 211–226, Tallinn, Estonia, November 2008.

[25] Q. Tan, Y. Gao, J. Shi, X. Wang, and B. Fang, "A closer look at Eclipse attacks against Tor hidden services," in *Proceedings of the IEEE International Conference on Communications*, pp. 1–6, 2017.

[26] A. Kwon, M. AlSabah, D. Lazar, M. Dacier, and S. Devadas, "Circuit fingerprinting attacks: passive deanonymization of tor hidden services," in *Proceedings of the 24th USENIX Conference on Security Symposium*, pp. 287–302, Berkeley, CA, USA, August 2015.

[27] J. Marques, L. Velasco, and R. V. Duijn, *Tor: Hidden Service Intelligence Extraction*, https://www.delaat.net/rp/2017-2018/p98/report.pdf, 2018.

[28] R. Louis-Paul, "Statistical properties of Winsorized means for skewed distributions," *Biometrika*, vol. 81, no. 2, pp. 373–383, 1994.

[29] J. Yan and J. Kaur, "Feature selection for website fingerprinting," *Proceedings on Privacy Enhancing Technologies*, vol. 2018, no. 4, pp. 200–219, 2018.

[30] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts, "Understanding variable importances in forests of randomized trees," in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, pp. 431–439, Lake Tahoe, NV, USA, December 2013.

[31] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.