applied sciences

MDPI

Article

SRP: A Microscopic Look at the Composition Mechanism of Website Fingerprinting

Yongxin Chen, Yongjun Wang * and Luming Yang

College of Computer, National University of Defence Technology, Changsha 410000, China
* Correspondence: Wangyongjun@nudt.edu.cn

Abstract: Tor serves better at protecting users' privacy than other anonymous communication tools. Even though it is resistant to deep packet inspection, Tor can be de-anonymized by the website fingerprinting (WF) attack, which aims to monitor the website users are browsing. WF attacks based on deep learning perform better than those using manually designed features and traditional machine learning. However, a deep learning model is data-hungry when simulating the mapping relations of traffic and the website it belongs to, which may not be practical in reality. In this paper, we focus on investigating the composition mechanism of website fingerprinting and try to solve data shortage with bionic traffic traces. More precisely, we propose a new concept called the send-and-receive pair (SRP) to deconstruct traffic traces and design SRP-based cumulative features. We further reconstruct and generate *bionic traces* (BionicT) based on the rearranged SRPs. The results show that our bionic traces can improve the performance of the state-of-the-artdeep-learning-based Var-CNN. The increment in accuracy reaches up to 50% in the five-shot setting, much more effective than the data augmentation method HDA. In the 15/20-shot setting, our method even defeated TF with more than 95% accuracy in closed-world scenarios and an F_1-score of over 90% in open-world scenarios. Moreover, expensive experiments show that our method can enhance the deep learning model's ability to combat concept drift. Overall, the SRP can serve as an effective tool for analyzing and describing website traffic traces.

Keywords: Tor; privacy; website fingerprinting; data augmentation; send-and-receive pair; bionic trace

Citation: Chen, Y.; Wang, Y.; Yang, L. SRP: A Microscopic Look at the Composition Mechanism of Website Fingerprinting. *Appl. Sci.* **2022**, *12*, 7937. https://doi.org/10.3390/app12157937

Academic Editor: Gianluca Lax

Received: 15 July 2022
Accepted: 3 August 2022
Published: 8 August 2022

1. Introduction

Tor has been proven better at protecting users' privacy than other anonymous communication tools. It establishes a random multi-hop communication circuit through relays provided by global volunteers. The user's browsing data is encrypted and contained in multiple layers in the process of transmitting, similar to an onion wrapped in layers. Random multi-hop communication circuits bring significant challenges to large-scale supervision. Onion encryption effectively prevents the attacker from analyzing the communication content to hide the user's identity and location, as well as the target website visited by the user. Nevertheless, website fingerprinting (WF) attacks have proven that there are characteristics explicitly or implicitly hidden in traffic patterns. The attacker can de-anonymize the target website a user is browsing by well-trained classifiers.

Early on, some WF attacks were based on machine learning, such as SVM [1,2], k-NN [3], and random forest [4]. They used handcrafted features to describe the difference between the traffic trace of different websites and achieved more than 90% accuracy in the closed-world setting. Manually designed features have been criticized for being vulnerable to protocol changes [5–8], while deep learning can make up for this deficiency through automated feature engineering. Moreover, existing works show that features extracted by unsupervised DNNs are more effective [9]. Deep-learning-based WF attacks, such as Var-CNN [10], show great advantages with accuracy and a true positive rate of over 98% in both closed-world and open-world settings.

Although deep learning methods demonstrate promising results, they often require massive data in the training phase. It would be costly for the attacker to collect sufficient traffic instances for each website. Moreover, the content of a website may change irregularly all the time causing the data distribution shift. The adversary must collect training data frequently and retrain new models to catch up with the variation. It is an unaffordable expense, especially when running into a large set of monitored websites. As a result, the data hunger problem severely restricts the application of WF attacks. To tackle this problem, many studies [11–13] were inspired by transfer learning, introducing pre-trained feature extractors in their attack process. The issue is that these attacks need a large amount of additional pre-training data. What is more, the effectiveness of the extractor will be lost when faced with target data with a different distribution than the pre-training data. Others [14] have tried to use data augmentation methods borrowed from the computer vision field to offer extra training data. However, website traffic and pictures are of a completely different nature. For example, the mixup can perturb the color and shading of a picture without changing the semantics. The same perturbation does not work for traffic traces which only contain the direction and timestamps of cells.

In this study, we reviewed the loading process of web pages and proposed the send-and-receive pair (SRP) as the basic unit to characterize website traffic. We observed that traffic traces could be bionic, quite different from entities with unique semantics and strict spatial structure,e.g., people's faces. The bionic traces generated by our proposed method can be used as a supplementary training dataset of WF attacks and further enhance their capability.

The main contributions of this paper are listed as follows:

- We proposed a new concept called send-and-receive pair (SRP), which provides a microscopic perspective for us to study website traffic.
- We demonstrated that website traffic could be bionic by reorganizing SRPs generated by web page loading. Furthermore, we proposed a bionic trace generation method based on the browser working mechanism and network state fluctuation simulation.
- We further investigate the concept drift problem of website traffic in closed-world and open-world scenarios. We reveal that bionic traces and SRP-based cumulative features can help mitigate the effects of concept drift.
- Expensive experiments show that bionic traces we generated can significantly alleviate the data hunger problem of deep learning-based WF attacks. It can achieve a nontrivial increase in performance when incorporating the state-of-the-art deep learning model.

The remainder of this paper is organized as follows. We first describe the threat model of the WF section in Section 2. In Section 3, we give a review of related work. We introduce the proposed SRP and bionic trace generation method in detail in Section 4. Section 5 describes our performance metrics and analyzes experimental results. Finally, we discuss the limitation of this study and draw a conclusion in Section 6.

2. Threat Model

We consider the attacker as a passive eavesdropper between the user and the Tor network entry node, as shown in Figure 1. A potential attacker could be the user's internet service provider, routers, autonomous services, etc. The attacker collects traffic traces and does not delay, modify, or drop any packets. After that, the attacker trains machine learning classifiers to identify whether users are visiting a particular website, thereby destroying anonymity.

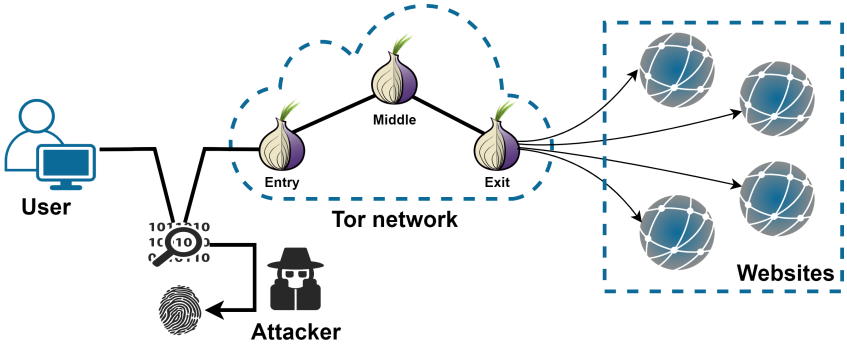


Figure 1. Threat model.

Evaluation Scenario Two basic WF attack evaluation scenarios are used in this study: closed-world and open-world.

- Closed-world scenario. The closed-world scenario assumes that the user only visits a fixed collection of K websites, which is monitored by the attacker. We assume a weak attacker could only gather N instances for each monitored website. The attacker trains his classifier with the $K \times N$ training instances and the test instances belong to one of the K monitored websites. Despite many criticisms for being unrealistic, this scenario is widely used to evaluate the basic classification performance of WF attacks in previous studies.
- Open-world scenario. The open-world scenario is a realistic but challenging setting. It considers that users may visit a large number of websites that the attacker may not be interested in, to be called unmonitored websites. The attack gathers instances for the unmonitored websites to join with the monitored instances as the training set. In reality, the number of unmonitored websites is over the capability of an attacker can monitor. Therefore, the unmonitored instances in the test set would be from other never-seen websites. The attacker must identify whether an instance belongs to the unmonitored or monitored set. If it belongs to the latter, the attacker should further figure out which is a monitored website.

3. Related Work

The history of WF attacks can be traced back to decades ago when Wagner et al. [15], the pioneer of website fingerprint research, first studied the traffic encrypted by SSL. After that, reaserchers have studied the WF attack against different communication protocols and proxy tools [16–20]. As WF attacks increasingly threaten user privacy, high expectations are placed on anonymous communication tools like Tor. Once the data leaves the user's Tor client, it would be encrypted to hide the content and destination. This mechanism ensures that communication information is not leaked. As a contradictory pair, attack and defense have continuously been developed in confrontation.

Handcrafted features can represent traffic traces with meaningful statistics and traditional machine learning can work as classifiers to identify the monitored website. Some previous WF attacks focus on manually designing features through prior knowledge, e.g., brust [1], the sequence of packets, size of packets, and inter-packet timings. Herrmann et al. [21] first attempted to destroy Tor's anonymity with the distribution of IP packet sizes but only achieved 2.96% accuracy. After that, the attack effect gradually improved [1–4] with the progress of machine learning. Hayes and Danezis [4] proposed k-fingerprinting based on random decision forests, which can correctly determine one of 30 monitored hidden services a client is visiting with 85% TPR, an FPR as low as 0.02%. Moreover, their results show that simple features such as counting the number of packets in a sequence leak more information about a web page's identity than complex features such as packet ordering or packet inter-arrival time features. Panchenko et al. [22] extract cumulative behavioral representations of the page loading process along with other handcrafted features, e.g., packer ordering and burst behavior. Their CUMUL can attack with excellent computational efficiency, achieving

96.62% TPR in a sizeable open-world scenario. Nevertheless, attacks based on manually designed features have their performance limited by the quality of the feature set. For instance, simple WF defenses can easily perturb statistics between packets [4].

Recently, deep-learning-based WF attacks have gradually become the mainstream. They can automate the process of feature engineering [23,24] and extract implicit features by designing networks with more complex structures. Oh et al. [9] broadly study the applicability of deep learning to website fingerprinting. Their work proved that features extracted by unsupervised DNNs are more potent than the manual design features. Rimmer et al. [5] systematically explored three deep learning algorithms applied to WF, including stacked denoising autoencoder (SDAE), convolutional neural network (CNN), and long short-term memory (LSTM). Their work shows that deep learning methods are more effective when provided with more training data. Later, Bhat et al. [10] announced that Var-CNN was effective in a low data setting with 50 instances per class available.

Considering a more practical setting, websites on the Internet are far beyond an attacker's monitoring capability, which means the attacker would not be able to have enough training instances for all monitored websites. Therefore, the WF attacks are closer to a few-shot problem in the real world. In this case, DF [24] and Var-CNN both become weak in the few-shot setting, at the risk of overfitting the training data. Sirinam et al. [11] decompose the attack model into two parts: a k-NN classifier and a triplet-network-based feature extractor. Their TF outperforms other attacks in the few-shot setting with more than 90% accuracy. However, WF attacks [11–13] based on transfer learning are highly dependent on a large amount of pre-training data. Moreover, it is hard for these attacks to counteract the distribution changes of target data. As a typical data augmentation method, HDA [14] shows that virtual instances can help with the data hunger problem. Since HDA is not designed for the characteristics of website traffic, its enhancement effect is limited.

Another unavoidable question is whether the WF attacks are time sensitive for the reason web pages are constantly changing irregularly [25], namely the concept drift problem. A several minutes time gap would lead to the content of a news website to be update [26]. With only a 10 days time gap, the accuracy of WF attacks dropped by approximately 40%. Rimmer et al. [5] held that the classifier trained and evaluated at one moment in time might overlook the stable fingerprint and learn the temporary features instead. On the other hand, depending on the number of monitored websites an attacker aims to cover, the cost to catch up with the changes brought by time would be a significant burden. Guiding deep learning-based classifiers to learn deep abstract features will help combat concept drift in website fingerprinting.

4. Methodology

In this section, we propose the SRP and bionic trace generation method. Firstly, we cover the data representation of website traffic. Then, we analyze the website traffic trace with a microscopic look based on SRP and display it in an intuitive manner. After that, we explain the bionic trace generation method and introduce the base model used for this study.

4.1. Data Representation

Tor uses a 512-byte cell as the basic unit to encapsulate application layer data, ensuring that packet length analysis does not leak information. Nevertheless, researchers proposed transforming traffic traces from packets to cell sequence [26] to perform WF attacks. Some studies concerned the use of timing-related features [3,4,10,27] in their attacks. However, timing-related features are highly correlated with specific network states and thus are not general. Therefore, we only use the direction feature since the regularity of sending and receiving packets is more important for website traffic. Note that traces of different websites would have different lengths, depending on the resources a page is loading. We align them to 5000 cells by padding zeros to the shorter and truncating the longer. A trace is represented as a cell sequence vector $(d_1, d_2, \ldots, d_{5000})$, where $d \in \{\pm 1, 0\}$.

4.2. A Microscopic Look

We review the browsers' behavior of loading a web page, which offers us an intuitive understanding of website traffic. The browser first fetches the HTML template containing the layout and resource index when visiting a website. After that, it performs resource loading and page rendering synchronously. The browser classifies resources and performs security policy checks. In order to ensure user browsing experience, the resources are divided into different priority groups and downloaded in order, as shown in Figure 2.

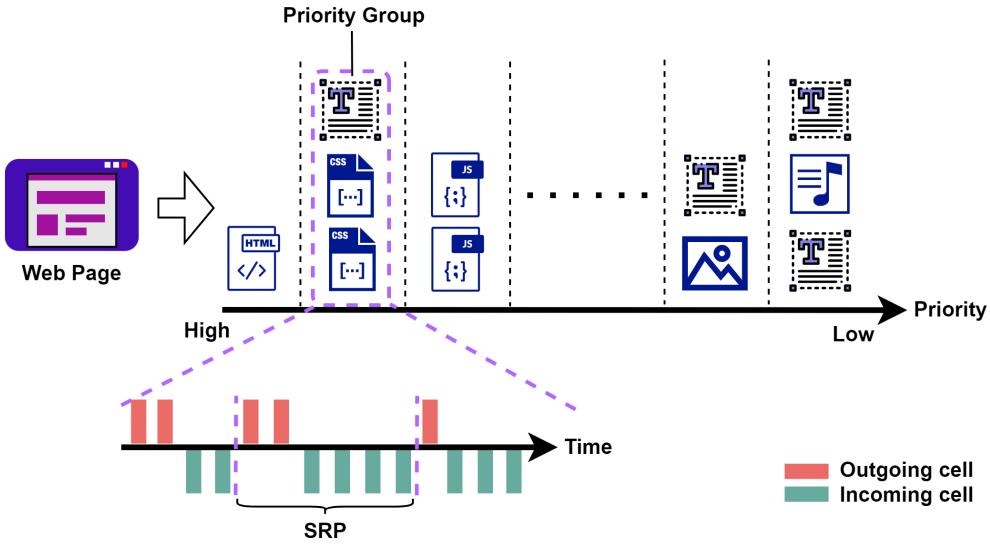


Figure 2. The loading process of a web page.

Many elements make up a web page, such as text, pictures, cascading style sheets (CCS), JavaScript, audio, and video. Different elements generate different sequences of sending and receiving cells when loaded. Based on this observation, we propose that the send-and-receive pair (SRP) can be used to describe the correlation between traffic cells and web page elements. We present the results of an analysis of website traffic using SRP in Figure 3. We drew four heatmaps for each website, and each represents the statistics for 50 instances from the website. The y-axis of the heatmap represents the number of outgoing cells in the SRP, and the x-axis represents the number of incoming cells. The number of each type of SRP is reflected with a different color in the heatmap.

We extensively observe the top 200 Alexa websites, which represent users' browsing preferences to some extent. It can be seen intuitively from examples in Figure 3 that the heatmaps of the same website are highly similar, while the heatmaps of different websites show different styles. Therefore, the SRP is statistically significant and can be used to characterize website traffic. Note that we do not argue that the SRP can be strictly mapped to a specific web page resource. The SRP is a rough representation of the resource loading, which may be disturbed by network conditions or other factors. This feature can be reflected in the subtle differences in the heatmap of the same website.

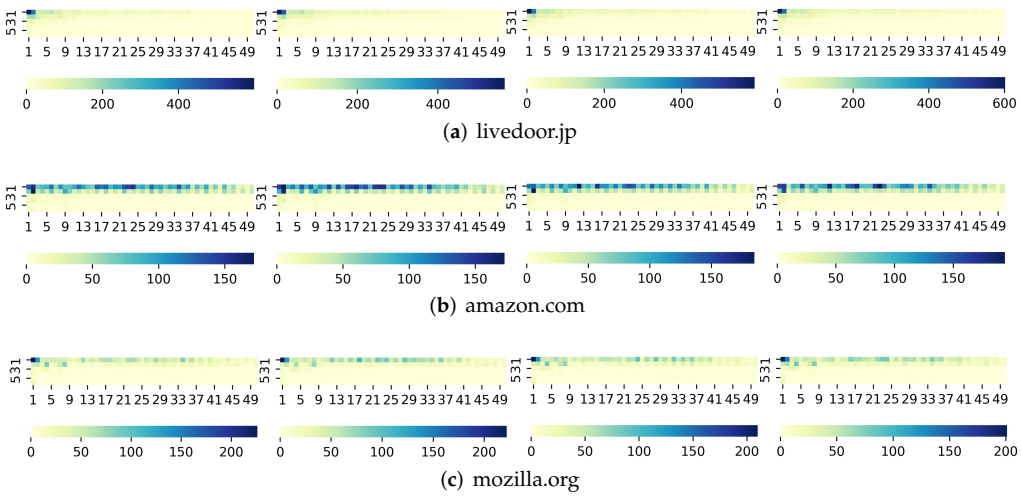


Figure 3. Heatmaps of website traffic traces based on SRPs statistics. (**a**) shows four heatmaps of the portal livedoor.jp. A large number of hyperlinks and other textual resources on this website result in brief responses (fewer incoming cells). (**b**) shows that numerous thumbnails and advertising animation results in lengthy responses to the shopping site amazon.com. (**c**) shows the statistics for the home page of the Mozilla project group. The mozilla.org shows some pictures and descriptions of their projects, resulting in both lengthy and brief responses. The limited resources on their page result in a limited number of SRPs.

4.3. Bionic Traffic Generation

We now explain the principle of generating bionic traffic trace and give more details about the operations.

The browser requests different resources through multiple threads simultaneously during the page loading process. For example, three resources in the priority group may be loaded simultaneously or successively, and cells generated by loading them will be included in an arrangement, as shown in Figure 2. The loading time of each resource depends on the state of the network link of the corresponding thread. Therefore, accessing the same website in a short time will also produce different packet sequences.

We simulate the impact of network state fluctuations during the resource loading phase by shuffling the order of SRPs. Figure 4 shows the generation process of bionic traces. We first pick two traces of the same website and compare them cell by cell. Then we mark the position where cells from two traces have different directions as the noise points. It is easy to infer that cells within a certain distance around the noise point have a higher probability of forming a priority group. We randomly selected noise points and set search intervals around them. The search interval length gives the priority group's approximate location. We simulate the priority groups by allowing the boundary of the search interval to be extended or shrunk to encompass a complete SRP. Finally, we generate bionic traces by reorganizing the SRPs within the simulation priority group. The process of searching and reorganizing the simulation priority group is described in Algorithm 1. The overall procedure of bionic traffic generation is described in Algorithm 2. The proposed generation algorithm can work at a low cost with $O(n)$ time complexity.

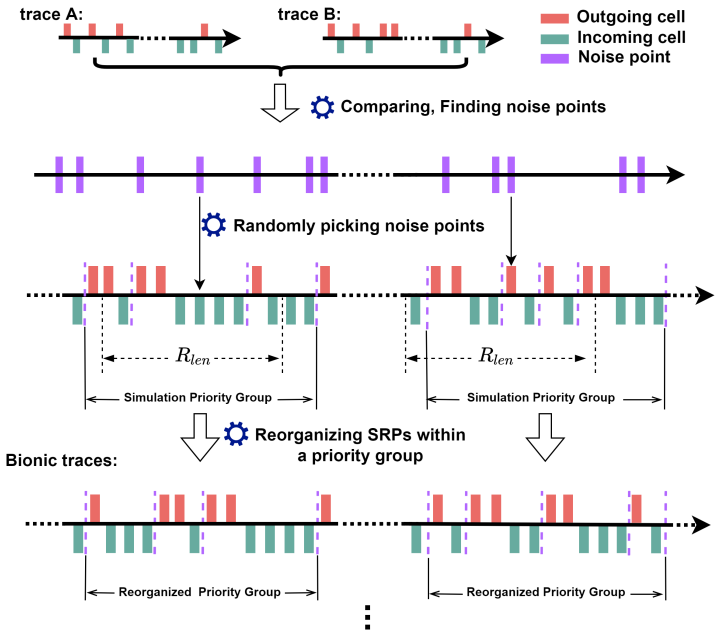


Figure 4. The process of bionic trace generation.

Algorithm 1 Search and reorganize simulation priority group.

Procedure SR_Simulation_ Priority_Group($\vec{A}$, i, L)
$start \leftarrow i - L/2$ // Searching backward for SRP
while $start > 0$ and $a_{start} > 0$ and $a_{start-1} > 0$ **do**
  $start \leftarrow start - 1$
end while
$end \leftarrow i + L/2$ // Searching forward for SRP
if $a_{end} - 1 > 0$ **then**
  while $a_{end} \neq 0$ and $a_{end-1} > 0$ and $a_{end} > 0$ **do**
    $end \leftarrow end + 1$
  end while
  $end \leftarrow end + 1$
  while $a_{end} \neq 0$ and $a_{end-1} < 0$ and $a_{end} < 0$ **do**
    $end \leftarrow end + 1$
  end while
else
  while $a_{end} \neq 0$ and $a_{end-1} < 0$ and $a_{end} < 0$ **do**
    $end \leftarrow end + 1$
  end while
end if
$\vec{P} \leftarrow \vec{A}[start : end]$ // Take the simulation priority group.
$\vec{P} \leftarrow operate\_SRPs(\vec{P})$ // Finding and shuffling SRPs.
$\vec{S} \leftarrow Merge(\vec{A}[: start], \vec{P}, \vec{A}[end :])$
return $\vec{S}$
End procedure

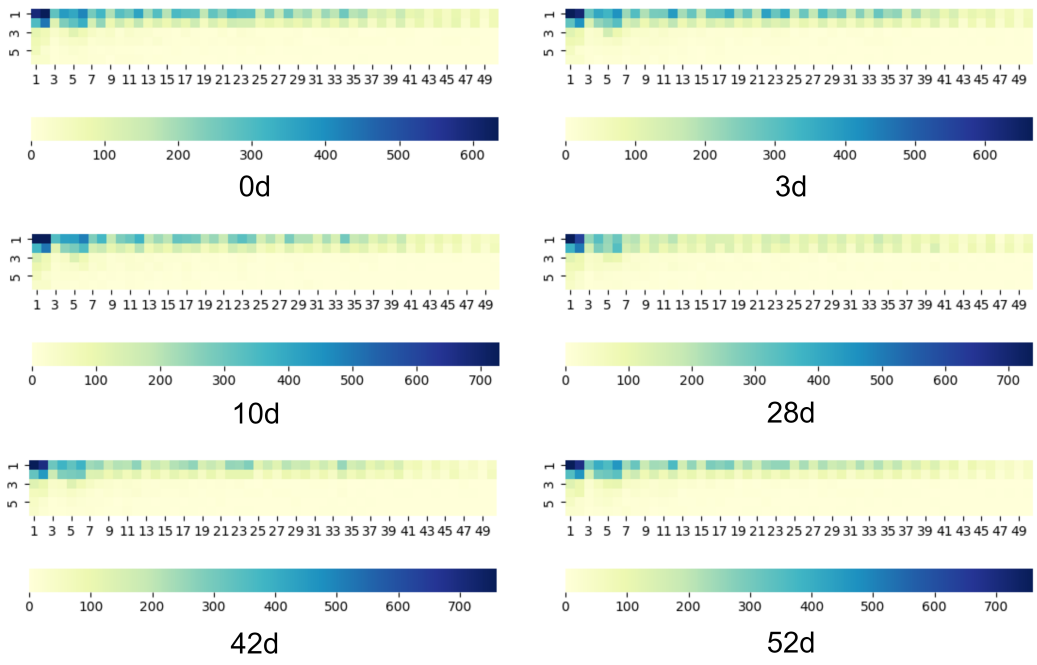
Algorithm 2 Bionic traffic generation.

Input: Traffic trace $\vec{A} = (a_1, a_2, \ldots, a_{5000})$, Traffic trace $\vec{B} = (b_1, b_2, \ldots, b_{5000})$
Output: Bionic traffic trace $\vec{S} = (s_1, s_2, \ldots, s_{5000})$

Setting $U_{noise} \leftarrow \emptyset$
for each $i \in [1, 5000]$ **do**
  if $a_i \neq b_i$ **then**
    $U_{noise} \leftarrow U_{noise} \cup \{i\}$
  end if
end for
$U_{noise} \leftarrow Compare\_Instances(\vec{A}, \vec{B})$
$U_{pick} \leftarrow Random\_Pick\_Noise\_Points(U_{noise}, R_{perc})$ //Randomly pick R_{perc} of the noise points.
$\vec{S} \leftarrow \vec{A}$ or $\vec{S} \leftarrow \vec{B}$
for $p \in U_{pick}$ **do**
  $\vec{S} \leftarrow SR\_Simulation\_Priority\_Group(\vec{S}, p, R_{len})$
  //R_{len} is the length of the search range with p as the midpoint.
end for
return $\vec{S}$

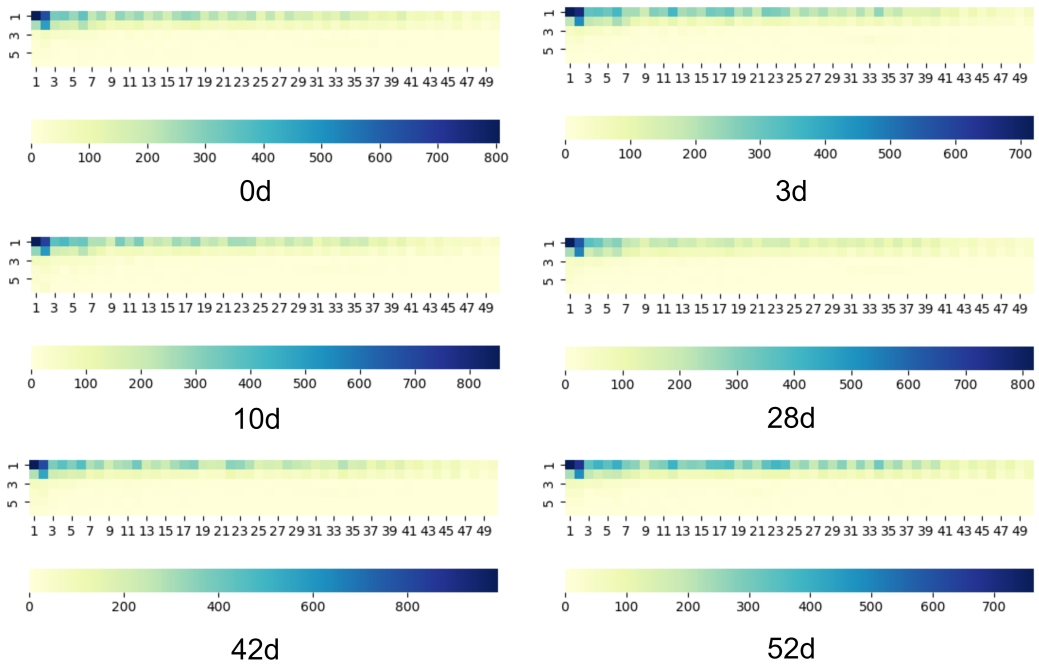
4.4. SRP-Based Cumulative Feature

We proposed a new cumulative feature based on the SRP for the attack model, which is meaningful in representing the traffic pattern. As shown in Figure 5, we plot heatmaps of traffic traces collected in different time. Heatmaps of the same website can be seen as much more similar even under the effect of concept drift. Based on this observation, we design the SRP-based cumulative feature, which consists of 336 values. Specifically, we separate the number of packets sent into six types, from one to five each is a type, and more than five is a type. Then, we separate the number of packets received into 56 types, from 1 to 55 each is a type, and more than 55 is a type. In this way, we can represent traffic traces with 336 SRP types. Finally, We calculate the statistics for each trace instance and convert them to normalized values to obtain the SRP-based cumulative feature.



(**a**)

Figure 5. *Cont.*



(**b**)

Figure 5. Heatmaps of website traffic traces under concept drift. (**a**,**b**) show heatmaps of alibaba.com and battle.net, respectively. Each heatmap is plotted based on traffic instances gather at different times over a two-month period: 3 days, 10 days, 28 days, 42 days, and 56 days after the end of the initial data collection (0 day).

4.5. Base Model

We test the effectiveness of bionic traffic trace with the Var-CNN model [10]. This model uses techniques from computer vision, namely dilated causal convolution [28] and ResNet [29]. It is robust to the trace length variations of website traffic with the wide receptive field offered by dilated causal convolution layers. The ResNet model architecture optimizes the learning task to fit residuals, making the deeper neural network available. Therefore, the deep semantics of traces can be extracted by the Var-CNN model. Both direction information and cumulative features are used in our experiments. We build the model by following the recommended hyperparameters [10].

5. Experiment

In this section, we design a series of experiments to prove that bionic traffic traces can be used to expand the training dataset as compensation for the data shortage of the WF problem. We simulate a tough, low data situation, where the attacker could only gather $N = 5, 10, 15, 20$ instances for each monitored website. Note that, for investigating the effectiveness of the proposed SRP-based cumulative feature, we report the results of the attack with or without using it (BionicT* represents the use of SRP-based cumulative feature while BionicT does not).

5.1. Dataset

We perform our experiment based on datasets used in the previous literature. These datasets are labeled as follows:

- AWF_{series} [5]: This dataset is the largest WF dataset collected in 2017 with Tor browser 6.5, We use three subsets in our study:
  - AWF_{100}. The set consists of the top 100 Alexa websites, with 2500 instances each.
  - AWF_{775}. The set consists of the other 775 websites, with 2500 traffic traces each.

  - $AWF_{400,000}$. The set consists of the top 400,000 Alexa websites, with one instance each.
  - AWF_{time}. The set consists of the top 200 Alexa websites, with 500 traffic traces each. 100 instances of these 200 sites were gathered at each point in time over a two-month period: 3 days, 10 days, 28 days, 42 days, and 56 days after the end of the initial data collection.
- $Wang_{100}$ [3]: This dataset was collected by using Tor Browser 3.5.1 in 2013, which contains 100 monitored websites. Each website has 90 instances available.
- $DF_{95,WTF-PAD}$ [24]. This dataset contains 95 monitored websites each with 1000 instances defended by WTF-PAD [7].

5.2. Metric

In order to reasonably evaluate our work, we use different metrics for the closed-world and open-world settings separately. We first define P as the total number of monitored traces. True positive (TP) represents a monitored trace classified as its category. Wrong positive (WP) indicates that it is classified as other monitored categories. Especially in the open-world scenario, a false negative (FN) indicates that it is classified as the unmonitored category. If an unmonitored trace is considered a monitored category, it is a false positive (FP). Therefore, we define accuracy for closed-world evaluation as:

$$Accuracy = \frac{TP}{P}. \tag{1}$$

The F_1-score used for open-world evaluation is defined as:

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2}$$

where:

$$Precision = \frac{TP}{TP + WP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + WP + FN} \tag{4}$$

We run each experiment 10 times and use the mean and standard deviation to report the performance. Moreover, we randomly sample instances for training and testing each time to ensure more reliable and representative results.

5.3. Hyperparameter Tuning

To develop the method of generating bionic traffic, we perform hyperparameter tuning for R_{perc} and R_{len}, which have been used in Algorithm 2. The search space of R_{perc} ranges from 10% to 100%, with a step of 10%. The search space of R_{len} is $[6, 8, 10, 12, 14, 16]$. We carried out the tuning process based on AWF_{100}.

5.3.1. Experimental Setting

We use 100 random-sampled examples for each website from AWF_{100} and divide them into three chunks with 20/10/70 examples, respectively. N training instances were collected from the first chunk to form the training datasets, and we generated 500 bionic traffic traces for each website based on the training dataset. Then, we join the bionic traces with original training samples to train the attack model. We train the model for 50 epochs and use the checkpoint to save the best performing one on the second chunk (validation dataset) and test on the third chunk (testing dataset). Note that, we apply these basic experimental settings for the rest of the experiments.

5.3.2. Results

Table 1 shows the effect of R_{perc} to bionic traffic when R_{len} was fixed to 10, the accuracy peaks at $R_{perc} = 40\%$. It is reasonable since the traffic is disturbed for various reasons while shuffling the order of SRPs can only simulate the disturbance to the priority group. If this disturbance is applied to more noise points, it will destroy the basic traffic pattern. If it is applied to fewer noise points, the disturbance will be limited, and the influence of R_{len} will become greater. As shown in Table 2, $R_{len} = 12$ is the most suitable value to help locate simulation priority group when R_{perc} was fixed to 40%.

Table 1. The effect of variation percentage on attack performance. Metrics: accuracy.

R_{perc}	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Var-CNN + BionicT	93.273	92.885	93.507	94.042	93.621	93.831	93.464	93.803	93.6	93.389

Table 2. The effect of seg_length on attack performance. Metrics: accuracy.

R_{len}	6	8	10	12	14	16
Var-CNN + BionicT	92.831	93.468	93.434	94.251	93.728	93.838

As the results, we use $R_{perc} = 40\%$ and $R_{len} = 12$ to generate bionic traffic traces for the rest of the experiments.

5.4. Closed-World Evaluation

We investigate the effectiveness of the proposed bionic traffic generation method in the closed-world setting. The experiments are carried out based on datasets AWF_{100} and $Wang_{100}$, which have different data distributions.

5.4.1. Experimental Setting

We use 100 random-sampled examples for each website from AWF_{100} and divide them into three chunks with 20/10/70 examples, respectively. Since $Wang_{100}$ only has 90 instances per website, the dataset could be split into three chunks with 20/10/60 instances each. N training instances were collected from the first chunk to form the n-shot training datasets. The second and third chunks serve as validation and testing datasets, respectively. We generated 1000 bionic traffic traces for each website.

5.4.2. Results

Table 3 shows the performance of WF attacks under n-shot settings. Traditional WF attacks [4,22] based on manually designed features performed better than deep learning-based attacks DF [29] and Var-CNN [10], which severely reveal the data hunger problem. With data augmentation methods applied to traces, HDA [14] can somewhat alleviate this problem. In contrast, our bionic traces show a significant advantage with 92.5% accuracy compared to 74.7% of HDA in the 10-shot setting. Moreover, our method performs better than TF [11] in both 15/20-shot settings with more than 95% accuracy. Since TF uses the AWF_{775} [5,11] for pre-training, the data distribution shift occurs when testing on $Wang_{100}$. As shown in Table 4, our advantage over TF expanded to the 10-shot setting in this case. Note that TF performs best in the five-shot setting because it learns extra knowledge from pre-training data. These two datasets were collected in different network environments with different browser versions, reflecting the feasibility and generality of the proposed SRP and bionic traces. By a fair comparison, the proposed SRP-based cumulative feature can enhance the performance of the attack further. The advantage is mainly shown when there are few origin samples, especially in the five-shot setting. This phenomenon can be reasonable since the manually designed features can help the model learn knowledge of traces lacking due to not having enough samples.

Table 3. Results of closed-world WF attack on AWF_{100}. Metrics: accuracy.

Method	5-Shot	10-Shot	15-Shot	20-Shot
CUMUL [22]	72.2 ± 1.7	79.7 ± 1.4	83.3 ± 2.0	85.9 ± 0.6
k-FP [4]	79.3 ± 1.0	83.9 ± 1.0	85.9 ± 0.6	87.5 ± 0.8
DF [24]	3.2 ± 0.6	66.4 ± 5.3	89.3 ± 1.3	90.3 ± 2.4
TF [11]	92.2 ± 0.6	93.9 ± 0.2	94.4 ± 0.3	94.5 ± 0.2
Var-CNN [10]	24.6 ± 3.0	61.9 ± 4.3	79.3 ± 2.6	87.9 ± 0.7
Var-CNN + HDA [14]	59.7 ±1.5	74.7 ± 2.6	86.4 ± 1.3	90.7 ± 0.8
Var-CNN + BionicT	76.3 ± 2.4	92.5 ± 0.3	95.1 ± 0.1	95.7 ± 0.2
Var-CNN + BionicT *	78.2 ± 0.6	93.1 ± 0.2	95.0 ± 0.2	96.1 ± 0.2

* Using the proposed SRP-based cumulative feature.

Table 4. Results of closed-world WF attack on $Wang_{100}$. Metrics: accuracy.

Method	5-Shot	10-Shot	15-Shot	20-Shot
DF [24]	1.2 ± 0.3	8.9 ± 3.4	58.6 ± 6.5	85.2 ± 2.3
TF [11]	84.5 ± 0.4	86.2 ± 0.4	86.6 ± 0.3	87.0 ± 0.3
Var-CNN [10]	37.4 ± 2.8	69.4 ± 1.8	79.9 ± 3.5	88.1 ± 0.2
Var-CNN + HDA [14]	76.9 ± 2.4	87.1 ± 0.6	89.8 ± 0.4	90.6 ± 0.4
Var-CNN + BionicT	78.8 ± 0.3	87.9 ± 0.7	90.2 ± 0.2	91.3 ± 0.1
Var-CNN + BionicT *	79.3 ± 0.3	88.1 ± 0.2	90.2 ± 0.3	91.0 ± 0.3

* Using the proposed SRP-based cumulative feature.

5.5. Open-World Evaluation

Next, we consider the more realistic open-world scenario to examine the practicality of the bionic trace generation method.

5.5.1. Experimental Setting

We include the unmonitored samples from $AWF_{400,000}$ with additional labels during the model training phase. We randomly pick 10,000 websites (each with one instance) from the dataset and split them into three disjoint chunks with 2000/1000/7000 instances, respectively. By doing this, we simulate the practical situation where the attack model must distinguish unmonitored websites that were never met in the training phase. To construct a balanced training dataset, we sample the same number of unmonitored samples from the first block as the monitored instances in the n-shot setting. We generate 1000 bionic traces for each monitored website.

5.5.2. Results

Figure 6 shows the results of WF attacks in the open-world scenario. Attackers may use high confidence thresholds to reduce false positives, resulting in higher precision but lower recall. It is also possible for the attacker to sacrifice precision to capture the user's access to the monitored websites as much as possible. Therefore, we use the F_1-score to balance these two metrics. Like the closed-world results, our bionic traffic is more effective than HDA. It significantly improves the detection ability of the Var-CNN model. To a certain extent, our method is more practical than TF. For example, its performance is more stable as the threshold changes in the 10/15/20-shot settings. Moreover, the model trained with the SRP-based cumulative feature performs the best when the threshold grows. This phenomenon indicates that our bionic traces and the SRP-based cumulative feature give more detailed information about monitored websites. It can help the model better understand the distinction between different websites.

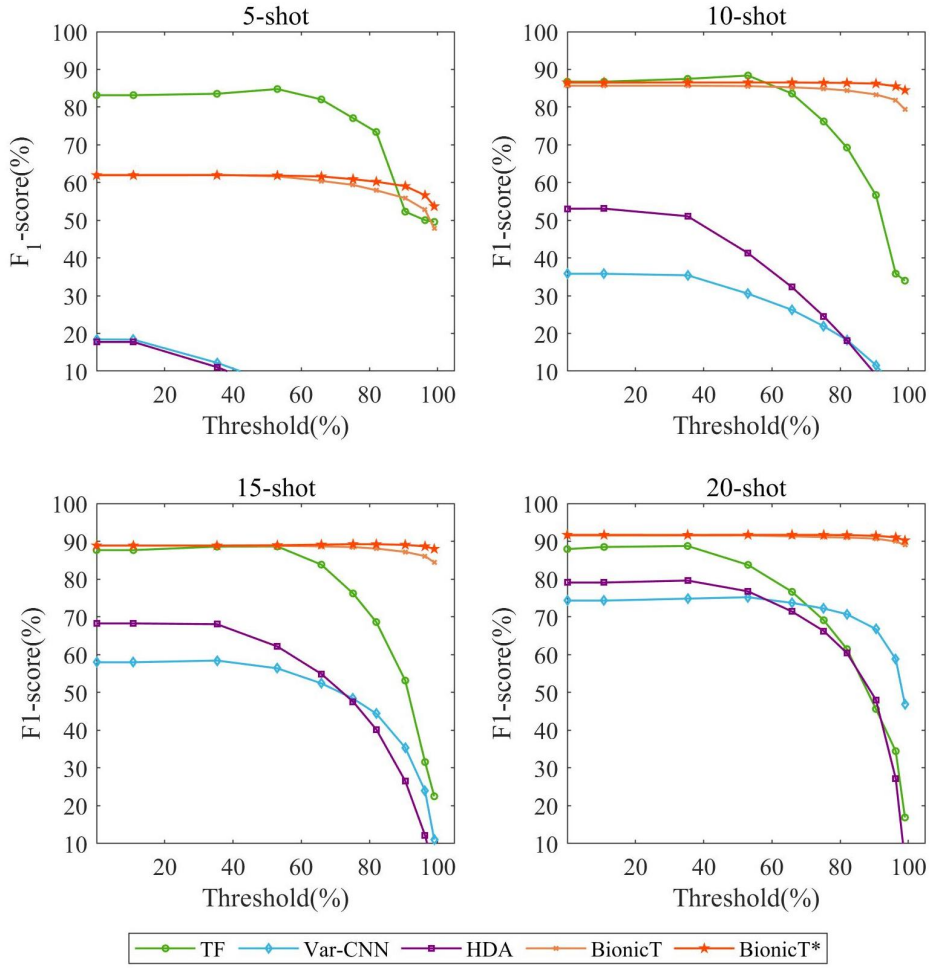


Figure 6. The performance of WF attacks in the open-world scenario.

5.6. Evaluation on Concept Drift

In this section, we compare the performance of attacks in dealing with concept drift. We carried out experiments in both closed-world and open-world scenarios with 20 instances available for each monitored website. We mimic a scenario where the attacker only trains models on the initial dataset and tests on datasets collected after a period of time. We set the confidence threshold to 50% to report the open-world scenario performance.

5.6.1. Experimental Setting

For the closed-world scenario, we use 100 random-sampled examples for each website from AWF_{100} and divide them into three chunks with 20/10/70 examples, respectively. We generated 1000 bionic traffic traces for each website base on the first chunk. We train the model by following the same way used in Section 5.4. Then, we test the performance of attacks on each AWF_{time} subset.

For the open-world scenario, we introduce the unmonitored samples from $AWF_{400,000}$. We randomly pick 10,000 websites (each with one instance) from the dataset and split them into three disjoint chunks with 2000/1000/7000 instances, respectively. We organize the balance training dataset and train the model by following the same way used in Section 5.5. We join the AWF_{time} subset with unmonitored samples from $AWF_{400,000}$ to form the test dataset at every point in time.

5.6.2. Results

The closed-world results are shown in Table 5, all attacks suffer the detrimental effect of concept drift. In a period of 56 days after training, the accuracy of DF dropped by nearly

25%, and the accuracy of Var-CNN dropped by nearly 22%. Our bionic trace can strongly weaken the influence of concept drift with the accuracy dropped by 18%. At every point in time, our method shows an advantage over HDA. The performance of TF is impressive to stay at high accuracy. However, the attack that applies our bionic traces and the SRP-based cumulative feature performs the best.

Table 5. Results of closed-world concept drift WF attack on awf_{100}. Metrics: accuracy.

Method	0-Day	3-Days	10-Days	28-Days	42-Days	56-Days
DF [24]	91.5	91.5	87.1	78.6	72.2	66.4
TF [11]	95.5	95.5	92.8	87.3	82.1	78.7
Var-CNN [10]	87.2	87.2	84.2	76.4	70.3	65.4
Var-CNN + HDA [14]	89.6	89.9	88.2	79.8	72.9	70.2
Var-CNN + BionicT	95.7	95.7	94.5	88.7	81.8	77.2
Var-CNN + BionicT *	96.3	96.5	94.7	88.2	83.1	78.9

* Using the proposed SRP-based cumulative feature.

Figure 7 illustrates the performance of attacks in the open-world scenario. Our method leads the deep learning model to learn the deep abstract features of traffic traces, even though concept drift affects the data distribution of target datasets. Furthermore, as illustrated in Figure 8a, the proposed SRP-based cumulative features show a greater advantage at a time gap of 56 days. Figure 8b shows the enhancement of our method to the Var-CNN model. As the time gap grows, the enhancement of the F_1-score increases from 12% to 15%, which proves that our method can enhance the model's ability to combat concept drift.

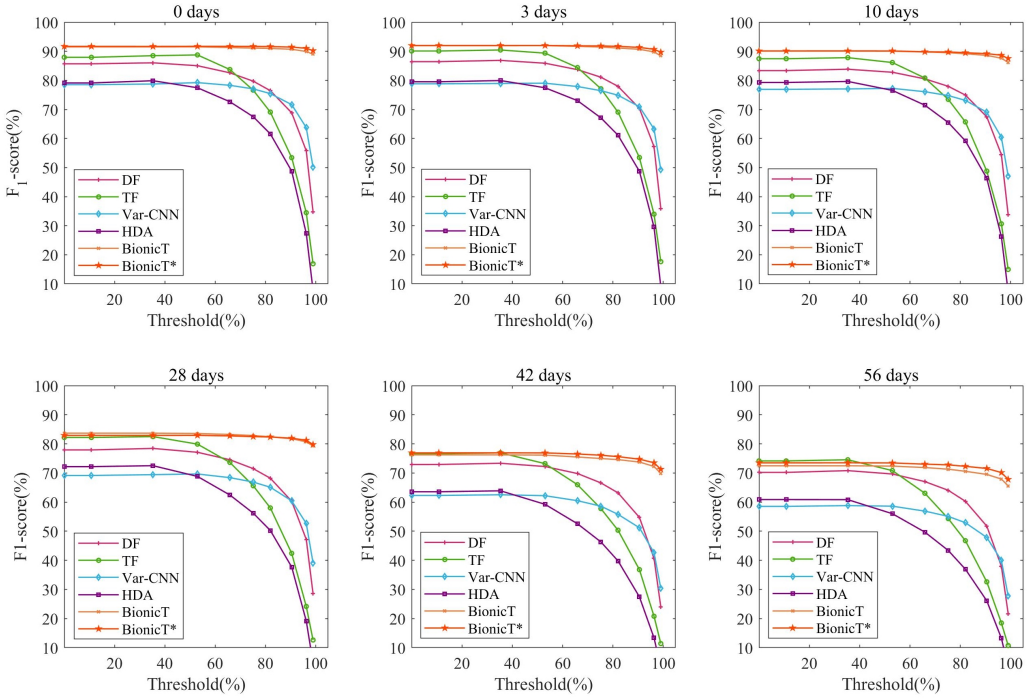
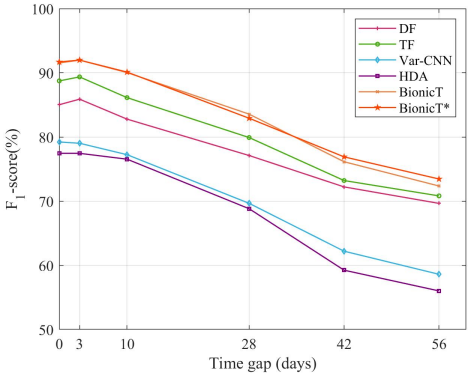
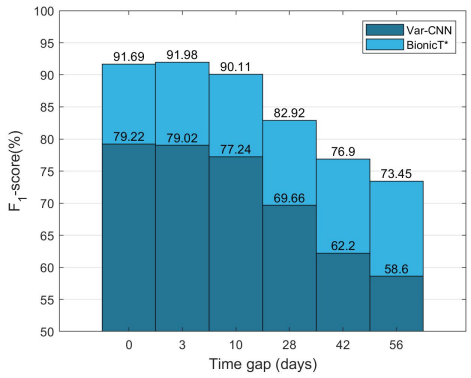


Figure 7. The performance of WF attacks at each point in time in the open-world scenario.



(**a**)



(**b**)

Figure 8. The performance of WF attacks under concept drift in the open-world scenario. (**a**) shows WF attacks' performance over a period of 56 days. (**b**) shows the enhancement of our method to Var-CNN.

5.7. Closed-World Evaluation on the Defended Dataset

We further investigate whether bionic traces are effective for website traffic defended by WTF-PAD [7], which is the main candidate to be deployed in Tor.

5.7.1. Experimental Setting

We randomly sample 100 instances for each website from the $DF_{95,WTF-PAD}$. Then, we form the *n*-shot training dataset by following the same procedure used in Section 5.4. We generate 1000 bionic traces for each website.

5.7.2. Results

Table 6 shows the performance of WF attacks against WTF-PAD defense. The accuracy of all attacks significantly dropped to nearly 60% in the 20-shot setting. DF also behaves less than ideally even though it is announced to be effective against WTF-PAD. This observation suggests that the data hunger is more severe when WF defenses are applied. On the other hand, our bionic traces can help the Var-CNN model increase its capabilities to a certain extent. However, it is not as effective as it would be on undefended traces. The SRP-based cumulative feature even has a negative effect. It could be inferred that WTF-PAD sent dummy packets by both ends of the communication, which results in dummy SRP insertion and confusing traffic patterns. Therefore, we believe that the proposed bionic generation method needs to be tuned with some other tricks, e.g., insertion or deletion at SRP granularity.

Table 6. Results of closed-world WF attack on $DF_{95,wtf-pad}$. Metrics: accuracy.

Method	5-Shot	10-Shot	15-Shot	20-Shot
DF [24]	1.1 ± 0.1	8.6 ± 1.5	28.0 ± 6.8	42.5 ± 4.1
TF [11]	54.1 ± 0.7	57.8 ± 0.6	60.2 ± 0.4	61.2 ± 0.4
Var-CNN [10]	6.4 ± 0.4	7.8 ± 0.4	12.4 ± 0.9	12.9 ± 0.9
Var-CNN + HDA [14]	25.3 ± 2.2	46.9 ± 1.9	48.7 ± 1.4	63.2 ± 1.8
Var-CNN + BionicT	27.3 ± 1.1	44.2 ± 1.0	53.1 ± 0.5	59.9 ± 1.0
Var-CNN + BionicT *	20.4 ± 0.9	39.9 ± 1.1	51.9 ± 0.5	56.3 ± 0.4

* Using the proposed SRP-based cumulative feature.

6. Conclusions and Future Work

In this study, we investigated the composition mechanism of website fingerprinting from a microscopic level by proposing the concept of the send-and-receive pair (SRP). We demonstrated that SRP is statistically significant and can be used to describe website traffic trace. Based on this finding, we further proposed the bionic trace generation method. Expensive experiments show that bionic traces successfully simulated the website traffic and relieved the data hunger problem. The proposed SRP-based cumulative feature can help classify under the concept drift circumstances. Both closed-world and open-world results demonstrate that our method is competitive with TF while reducing the burden of data collection. The promising results verify that the concept of SRP is valuable.

We recognize some limitations in our study. For example, our bionic trace generation does not work well when WF defenses are applied. The bionic traces generation method we proposed is random to some extent, which may cause it useless for defended traffic. To tackle this problem, we need to focus on the property of each defense and specifically design our method. Therefore, more qualitative and quantitative studies are worthwhile to explore its potential further. Moreover, we believe that the intra-relationship between SRPs is an essential factor in composing fingerprints. We will continue researching in this direction by referring to the successful experience of learning the contextual connection of text in the natural language processing field.

Author Contributions: Conceptualization, Y.C., Y.W. and L.Y.; methodology, Y.C., Y.W. and L.Y.; software, Y.C.; validation, Y.C.; formal analysis, Y.C. and L.Y.; investigation, Y.C.; resources, Y.C.; data curation, Y.C.; writing—original draft preparation, Y.C.; writing—review and editing, Y.C., Y.W. and L.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Key Research and Development Program of China (2018YFB0204301), National Natural Science Foundation of China (61472439).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The principal datasets used in this research can be downloaded from the websites (https://github.com/DistriNet/DLWF and https://www.cse.ust.hk/~taow/wf/data/).

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

References

1. Panchenko, A.; Niessen, L.; Zinnen, A.; Engel, T. Website fingerprinting in onion routing based anonymization networks. In Proceedings of the WPES'11: Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society, Chicago, IL, USA, 17 October 2011; pp. 103–114.
2. Cai, X.; Zhang, C.X.; Joshi, B.; Johnson, R. Touching from a distance: Website fingerprinting attacks and defenses. In Proceedings of the CCS'12: Proceedings of the 2012 ACM Conference on Computer and Communications Securit, Los Angeles, CA, USA, 16–18 October 2012.
3. Wang, T.; Cai, X.; Nithyanand, R.; Johnson, R.; Goldberg, I. Effective attacks and provable defenses for website fingerprinting. In Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, 20–22 August 2014; pp. 143–157.

4. Hayes, J.; Danezis, G. k-fingerprinting: A Robust Scalable Website Fingerprinting Technique. In Proceedings of the 25th USENIX Security Symposium (USENIX Security 16), Austin, TX, USA, 10–12 August 2016.
5. Rimmer, V.; Preuveneers, D.; Juárez, M.; Goethem, v.T.; Joosen, W. Automated Website Fingerprinting through Deep Learning. In Proceedings of the 25th Symposium on Network and Distributed System Security (NDSS 2018), San Diego, CA, USA, 18–21 February 2018.
6. Cai, X.; Nithyanand, R.; Wang, T.; Johnson, R.; Goldberg, I. A Systematic Approach to Developing and Evaluating Website Fingerprinting Defenses. In Proceedings of the 2014 ACM Conference on Computer and Communications Security, Scottsdale AZ, USA, 3–7 November 2014.; pp. 227–238.
7. Juárez, M.; Imani, M.; Perry, M.; Díaz, C.; Wright, M. Toward An Efficient Website Fingerprinting Defense. In Proceedings of the Computer Security—ESORICS 2016, Heraklion, Greece, 28–30 September 2016; pp. 27–46.
8. Cherubin, G.; Hayes, J.; Juárez, M. Website Fingerprinting Defenses at the Application Layer. *PoPETs* **2017**, *2017*, 186–203. [CrossRef]
9. Oh, E.S.; Sunkam, S.; Hopper, N. p1-FP: Extraction, Classification, and Prediction of Website Fingerprints with Deep Learning. In Proceedings of the Privacy Enhancing Technologies, Minneapolis, MN, USA, 18–21 July 2017.
10. Bhat, S.; Lu, D.; Kwon, A.; Devadas, S. Var-CNN: A Data-Efficient Website Fingerprinting Attack Based on Deep Learning. *PoPETs* **2019**, *4*, 292–310. [CrossRef]
11. Sirinam, P.; Mathews, N.; Rahman, S.M.; 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, London, UK, 11–15 November 2019; pp. 1131–1148.
12. Chen, M.; Wang, Y.; Xu, H.; Zhu, X. Few-shot website fingerprinting attack. *Comput. Networks* **2021**, *198*, 108298. [CrossRef]
13. Chen, M.; Wang, Y.; Zhu, X. Few-shot Website Fingerprinting Attack with Meta-Bias Learning. *Pattern Recognit.* **2022**, *130*, 108739. [CrossRef]
14. Chen, M.; Wang, Y.; Qin, Z.; Zhu, X. Few-Shot Website Fingerprinting Attack with Data Augmentation. *Secur. Commun. Netw.* **2021**, *2021*, 2840289. [CrossRef]
15. Wagner, D.; Schneier, B. Analysis of the SSL 3.0 protocol. In Proceedings of the WOEC'96 Proceedings of the 2nd conference on Proceedings of the Second USENIX Workshop on Electronic Commerce, Oakland, CA, USA, 18–21 November 1996; Volume 2, p. 4.
16. Sun, Q.; Simon, R.D.; Wang, Y.M.; Russell, W.; Padmanabhan, N.V.; Qiu, L. Statistical Identification of Encrypted Web Browsing Traffic. In Proceedings of the IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 12–15 May 2002; p. 19.
17. Hintz, A. Fingerprinting websites using traffic analysis. In *Privacy Enhancing Technologies*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 171–178.
18. Liberatore, M.; Levine, N.B. Inferring the source of encrypted HTTP connections. In Proceedings of the ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 30 October–3 November 2006; pp. 255–263.
19. Bissias, D.G.; Liberatore, M.; Jensen, D.; Levine, N.B. Privacy vulnerabilities in encrypted HTTP streams. In *Privacy Enhancing Technologies*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 1–11.
20. Lu, L.; Chang, E.C.; Chan, M.C. Website fingerprinting and identification using ordered feature sequences. In Proceedings of the European Symposium on Research in Computer Security, Athens, Greece, 20–22 September 2010.
21. Herrmann, D.; Wendolsky, R.; Federrath, H. Website fingerprinting: Attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier. In Proceedings of the CCSW, Chicago, IL, USA, 13 November 2009; pp. 31–42.
22. Panchenko, A.; Lanze, F.; Pennekamp, J.; Engel, T.; Zinnen, A.; Henze, M.; Wehrle, K. Website Fingerprinting at Internet Scale. In Proceedings of the Network and Distributed System Security Symposium, San Diego, CA, USA, 21–24 February 2016.
23. Abe, K.; Goto, S. Fingerprinting attack on tor anonymity using deep learning. In Proceedings of the Asia-Pacific Advanced Network, Pasay City, Philippines, 25–29 January 2016.
24. Sirinam, P.; Imani, M.; Juárez, M.; Wright, M. Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep Learning. In Proceedings of the ACM Conference on Computer and Communications Security, Toronto, Canada, 15–19 October 2018; pp. 1928–1943.
25. Juárez, M.; Afroz, S.; Acar, G.; Díaz, C.; Greenstadt, R. A Critical Evaluation of Website Fingerprinting Attacks. In Proceedings of the ACM Conference on Computer and Communications Security, Scottsdale, AZ, USA, 3–7 November 2014; pp. 263–274.
26. Wang, T.; Goldberg, I. Improved website fingerprinting on Tor. In Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society, Berlin, Germany, 4 November 2013; pp. 201–212.
27. Rahman, S.M.; Sirinam, P.; Matthews, N.; Gangadhara, G.K.; Wright, M. Tik-Tok: The Utility of Packet Timing in Website Fingerprinting Attacks. Cryptography and Security. *arXiv* **2019**, arXiv:1902.06421..
28. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. In Proceedings of the International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016.
29. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning For Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 May 2016; pp. 770–778.