International Journal of Networking and Computing – www.ijnc.org, ISSN 2185-2847
Volume 14, Number 2, pages 157–185, July 2024

Method for Detecting DoH Communications from Non-Encrypted Information at a Middlebox

Yuya Takanashi
Graduate School of Science and Technology
University of Tsukuba, Tsukuba, Japan
yuya_tk@netlab.cs.tsukuba.ac.jp

and

Shigetomo Kimura
Institute of Systems and Information Engineering
University of Tsukuba, Tsukuba, Japan
0000-0001-7371-0407

Received: February 15, 2024
Revised: May 5, 2024
Accepted: May 30, 2024
Communicated by Tomoyuki Ohta

Abstract

DNS over HTTPS (DoH) enhances user privacy by encrypting DNS communications over HTTPS instead of plaintext. When all DNS messages are sent in plaintext, DNS queries can be examined and domain filtering applied if the queried domain name is identified as a phishing site or other such undesirable site. However, if DNS messages are encrypted over HTTPS, it can create many problems for network administrators. This paper proposes a method for detecting DoH communications from only non-encrypted information on a middlebox between user and resolvers by exploiting the fact that users always send a DNS query before they access a new domain. The middlebox can also identify the destination of the detected DoH traffic so that network administrators can recommend users to send DNS messages to a local DoH resolver with domain filtering instead of sending them to an open DoH resolver. In experiments to detect DoH communications during real communication from a web browser we achieved detection accuracy rates reaching 100% under certain parameters when the number of access IP addresses exceeded 350. To confirm the accuracy and generalizability of our experiments, the proposed method was also applied to captured HTTPS traffic data involving different web browsers and different DoH resolvers with an almost identical level of detection accuracy.

Keywords: DoH Detection Method, Non-Encrypted Information, DNS over HTTPS, DNS, HTTPS

1 Introduction

DNS (Domain Name System) is a decentralized database which links domain names (or host names) and their corresponding IP addresses, which allows users to access their favorite servers using familiar and memorable aliases. When communication between a client and a server starts, the client typically queries the IP address of the server specified by its domain name. At this time, the client sends a name resolution request to a server called a resolver. Both the request and its response are usually

sent in plaintext. If a third party can eavesdrop on the messages that the resolver sends or receives, it can observe the domain names that clients are trying to access. Because such domain name information is sensitive from a security standpoint, this situation is undesirable from the viewpoint of privacy protection. Therefore, some protocols such as DNS over TLS (DoT) [1] and DNS over HTTPS (DoH) [2] encrypt name resolution requests and responses exchanged between clients and resolvers. These technologies not only improve privacy protection, but also verify the legitimacy of the communication partners and prevent tampering with communication data. However, for network administrators, they present certain disadvantages. When all DNS messages are sent in plaintext, DNS queries can be examined and domain filtering applied if the queried domain name is identified as a phishing site or other such undesirable site. Even if clients encrypt DNS messages and send them to a resolver located on a local network, the resolver can still apply domain filtering, but most web browsers are implemented to send DNS messages to open resolvers located on the Internet.

To address these kinds of problems, this paper focuses on DoH as an encryption protocol and proposes a method for identifying DoH resolvers at a middlebox, such as a router, between users and resolvers when users browse Web pages [3, 4].

This method uses only non-encrypted information to detect DoH communications on a middlebox, exploiting the fact that clients always send a DNS query before they access a new domain. The proposed method does not involve any machine learning techniques to avoid the concomitant learning costs.

In experiments to detect DoH communications during real communication from a web browser we achieved detection accuracy rates reaching 100% with certain parameter settings when the number of access IP addresses exceeded 350. Today, a single web page is often filled with content such as advertisements, movies, scripts, and web fonts provided by multiple distributors. Therefore, it would be easy for a user to access 350 different IP addresses while browsing the web. It has also been suggested that highly accurate detection can be achieved with a small number of IP addresses by adjusting a parameter. To confirm the accuracy and generalizability of our experiments, the proposed method was also applied to captured HTTPS traffic data involving different web browsers and different DoH resolvers with an almost identical level of detection accuracy obtained.

This paper is organized as follows. Section 2 introduces DNS over HTTPS (DoH). Section 3 explains related research about overhead caused by DNS over HTTPS and detection methods using machine learning and other techniques. Section 4 proposes a method for detecting DoH communications, without the application of machine learning techniques, by exploiting non-encrypted information. Section 5 describes communication experiments for detecting DoH communications during real communication from a Firefox browser. The proposed method is also applied to captured HTTPS traffic data involving different web browsers and different DoH resolvers. Finally, section 6 concludes the paper and discusses future work.

2 DNS over HTTPS

Figure 1 shows an example of the name resolution flow in DNS. When a client resolves a domain name, it uses its own name resolution software (a stub resolver) to query a server (a full-service resolver) that will respond to the query. In Figure 1, The recursive resolver is the one that the client's ISP (Internet Service Provider) typically provides as a local DNS cache server. When the resolver does not know the corresponding IP address of a queried domain name, it recursively queries more authoritative servers that are more closely connected to the DNS root server. That is why the resolver is sometimes called a recursive resolver. When the recursive resolver finds the corresponding IP address, it replies to the client with the required IP address and caches the address for future queries. As well as the local DNS cache server, the client also selects a public DNS resolver that is accessible on the Internet.

DNS over HTTPS (DoH) standardized by RFC8484 [2] is a protocol that encrypts messages between a client and a resolver by HTTPS. Although DoH improves privacy protection for users, it presents many disadvantages for network administrators. For example, since it is difficult to distinguish normal HTTPS communications from DoH communications, network administrators

cannot block clients from accessing a public DoH recursive resolver. Since the DNS messages are not sent in plaintext, DNS queries cannot be examined and domain filtering cannot be applied even if the queried domain name is a phishing site or other undesirable site. DoH is now available in Firefox [5] and Chrome [6] and is being used more and more often. The number of public resolvers that support DoH is increasing. For example, public resolvers that now support it include Cloudflare (1.1.1.1) [7], Google (8.8.8.8, 8.8.4.4) [8] and Quad9 (9.9.9.9, 149.112.112.112) [9]. Since private non-public DoH resolvers are also available, it is difficult to statically filter all DoH resolvers by their IP addresses.

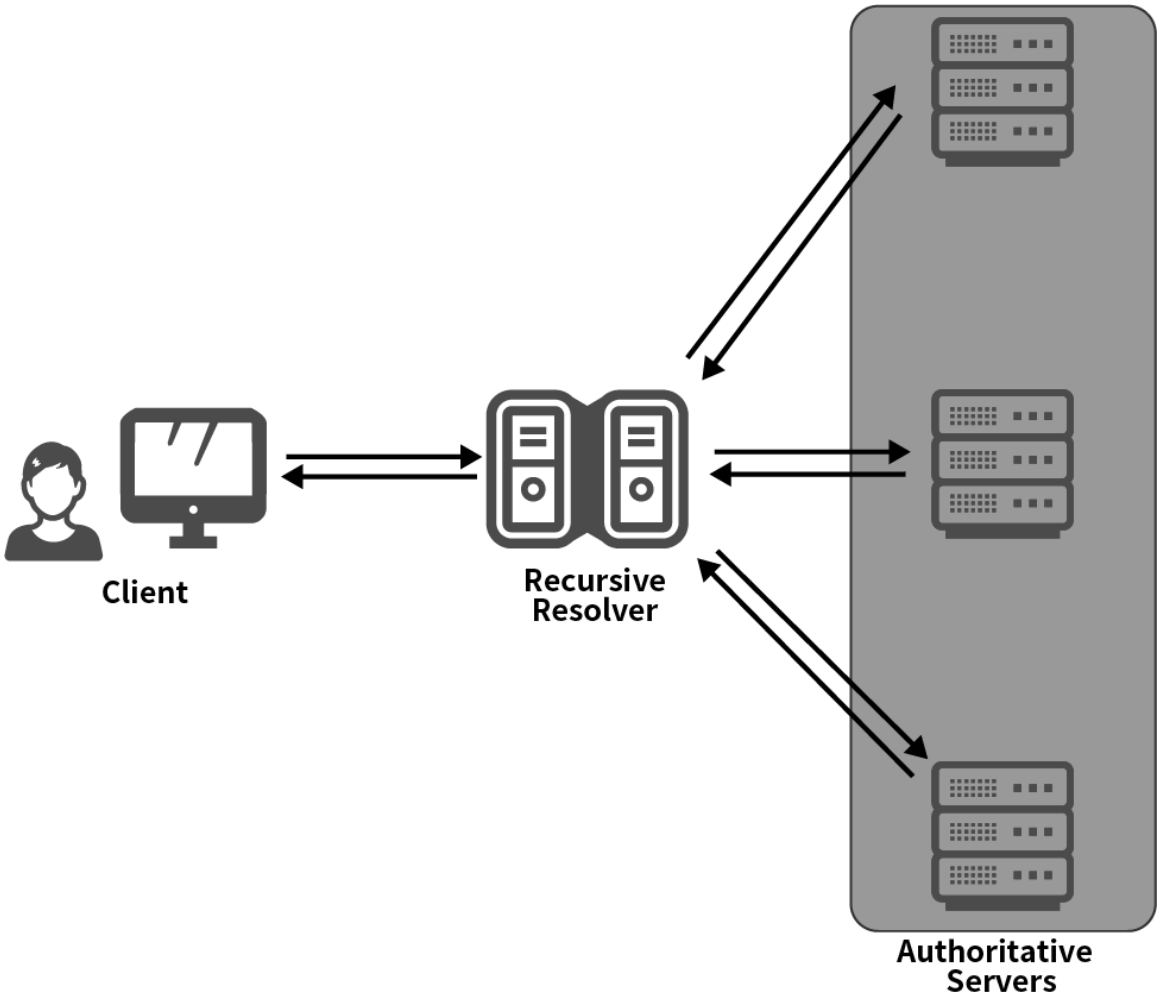

Figure 1: An example of name resolution flow in DNS.

3 Related Works

Compared to DNS, DoH is expected to have more overhead for the encryption process and longer processing latency. Böttger et al. [10] compared DNS and DoH name resolution times and web page load times for Google and Cloudflare public resolvers, where both DNS and DoH are available. The results confirmed that DoH took slightly longer to resolve names, but there was no significant difference in page load times. This means that users can benefit from the privacy provided by DoH without noticeable additional delays during web browsing.

Some studies have proposed systems for detecting DoH traffic by identifying DoH resolvers that clients communicate with for DoH, and other servers that are related with the identified DoH resolvers, and have correctly estimated that they are indeed DoH resolvers. For example, Wu et al. [11] proposed DOHUNTER that automatically identifies DoH resolvers. This system uses machine learning, trained in advance on DoH traffic patterns generated by web browser access, to detect actual DoH traffic. If a host has the identified DoH resolver's IP address but uses a different domain name from the DoH resolver, then the domain is estimated to be providing DoH services. Based on this idea, the proposed system also identifies other related DoH resolvers. However, the system relies on machine learning of DoH traffic patterns and therefore incur the concomitant learning costs and implementation challenges.

On the other hand, some DoH systems involve DoH tunneling as well as DNS tunneling. Therefore, a lot of research has advanced the idea of detecting DoH tunneling at a middlebox in the middle of the tunnel. For example, Kwan et al. [12] proposed a method for detecting DoH and DNS over TLS (DoT) tunneling traffic. When they analyzed characteristics such as throughput and average payload length of traffic sent through a prototype DNSTT tunneling tool, the authors found that tunneling traffic and normal DoH/DoT traffic can be distinguished by a threshold. It has been confirmed that some malware and other malicious entities used DNS tunneling to send out data to an external target. Zhan et al. [13] pointed out this fact and proposed a method for detecting data leakage by machine learning from characteristics of the TLS handshake used by both DoH tunneling and the DoH tunneling communication itself.

Mohammadreza et al. [14]. proposed a system using Random Forest and C4.5 as classifiers to detect malicious DoH traffic from HTTPS traffics with high accuracy. The evaluation used the CIRA-CIC-DoHBrw-2020 [15] dataset which includes not only the more than 268 M packets or 1 M flows captured for each DoH server such as Cloudflare and Quad9, for each web browser such as Chrome and Firefox, but also the feature data of HTTPS packets extracted by DoHlyzer [16]. Mitsuhashi et al. [17] proposed a multi-stage detection system using XGBoost, CatBoost, and LightGBM classifiers to detect DoH traffic and malicious and suspicious DoH tunneling traffic based on the names of the applied DNS tunneling tools. The system for example, used machine learning from 90 % dataset and evaluated from 10 % the CIRA-CIC-DoHBrw-2020 dataset so as to achieve more than 0.97 accuracy etc. for each stage.

Although we have introduced the DoH tunneling as our related works, the proposed method cannot detect the DoH tunneling, because their characteristics are different from those of normal DoH caused by web browsing. The goal of this study is to detect a DoH communication itself, which is used by general users while browsing the web, without using machine learning.

The computational complexity for the machine learning may be large but can be completed before the detection tasks. However so many related packets must be captured for learning, since such packets may depend on each region or age. The learning data may need to be updated and learn periodically, since DoH servers etc. may also change. The proposed method in the next section only needs to capture the real time communications through the middlebox so that the network managers can easily install the system.

4 Proposal Method

This section proposes a method for detecting DoH communication from non-encrypted information at a middlebox without incurring the learning cost of using machine learning techniques. Although machine learning is a powerful tool, in the future clients may change their access patterns to avoid detection of their DoH communication.

The proposed method assumes that a middlebox performs the analysis process for each client. If the IP address of a client is rewritten by NAT, it is difficult to identify the client. Therefore, in the proposed method, the middlebox analysis process must be performed in the local network before the IP address is rewritten by NAT, etc. Since DoH is hidden in HTTPS traffic, the target of the detection is a specific client that receives TCP/UDP (QUIC) packets with the source port number 443 used by HTTPS servers.

Figure 2 shows an IP address resolution flow as performed by a DoH recursive resolver whose IP address is *ip*, with the connection flowing from a server A to a server D. The proposed method is processed shown in the middlebox in Figure 2. Typically, a domain name is used to specify a connection point for web browsing. Therefore, as shown in Figure 2, the client repeats the following steps even when using DoH:

1. Communicates with the DoH recursive resolver whose IP address is *ip* to obtain the target server's IP address.
2. Connects to the server whose IP address is not *ip* in Figure 2 and performs the necessary communication.

Note that the client only knows the server's domain name and does not know their IP address before the resolution. Therefore, it can be assumed that the client communicates with a specific DoH resolver, whose IP address is *ip*, via HTTPS every time before starting a communication with a new destination. The proposed method detects DoH resolvers based on this idea.

4.1 CNNP

In the proposed method, the Connection Number between New Peers (CNNP) is defined for a client that connects to the server with the same IP address (*ip*) as the number of connections with the client between one communication with a new destination and any subsequent communication with another new destination.

For example, in Figure 2, the client connects with the DoH recursive resolver between a new communication *1 and the subsequent communication *2. Then, the CNNP is counted as 1, regardless of the number of packets. The CNNP is also counted before the first new communication (*1 in this figure). Thus, for the whole of Figure 2, the total number of connections to the DoH recursive resolver is 4, i.e., the total number of connection in the initial communication before *1 and all communications between *1-*2, *2-*3, and *3-*4.

If the CNNP of a server with the same IP address involved in every communication before connecting to a new server is high, then the server with the highest CNNP is considered to be a DoH recursive resolver.

Assuming that there may be multiple DoH recursive resolvers, and since the proposed method should not only detect the server with the maximum CNNP, but also detect other servers with very high CNNPs, outlier detection by IQR (InterQuartile Range) is applied. If the CNNP is higher than Equation (1), then the value is considered an outlier and the server is judged to be a DoH recursive resolver. Table 1 shows the meanings of the variables used in the equation.

$$IQR_c \times \alpha + Q3_c \tag{1}$$

α is a parameter used to identify outliers and a value of 1.5 or higher is generally used. When α is set to 1.5, it is possible to detect communications as DoH at an early stage of analysis, but it may also detect communications whose CNNP is not very high. It is expected that a larger α will increase the accuracy of the detection. In the next section, the experiments will change α from 10 to 50, but will also increase the detection time as α will increase. Section 5 will experimentally verify the difference in detection accuracy due to the parameter α. The advantage of identifying outliers by IQR is the lower computational cost as compared with identifying by standard deviations.

4.2 Influence from background traffics

When users browse web pages, they sometimes connect to multiple sites at the same time. For example, a user might watch a video or download a large file in the background while browsing. In addition to DoH communications, the CNNP also increases as these connections are maintained between successive new servers.

To address this problem, the proposed method calculates the average bytes for each HTTPS server by dividing the total bytes of HTTPS packets by the CNNP.

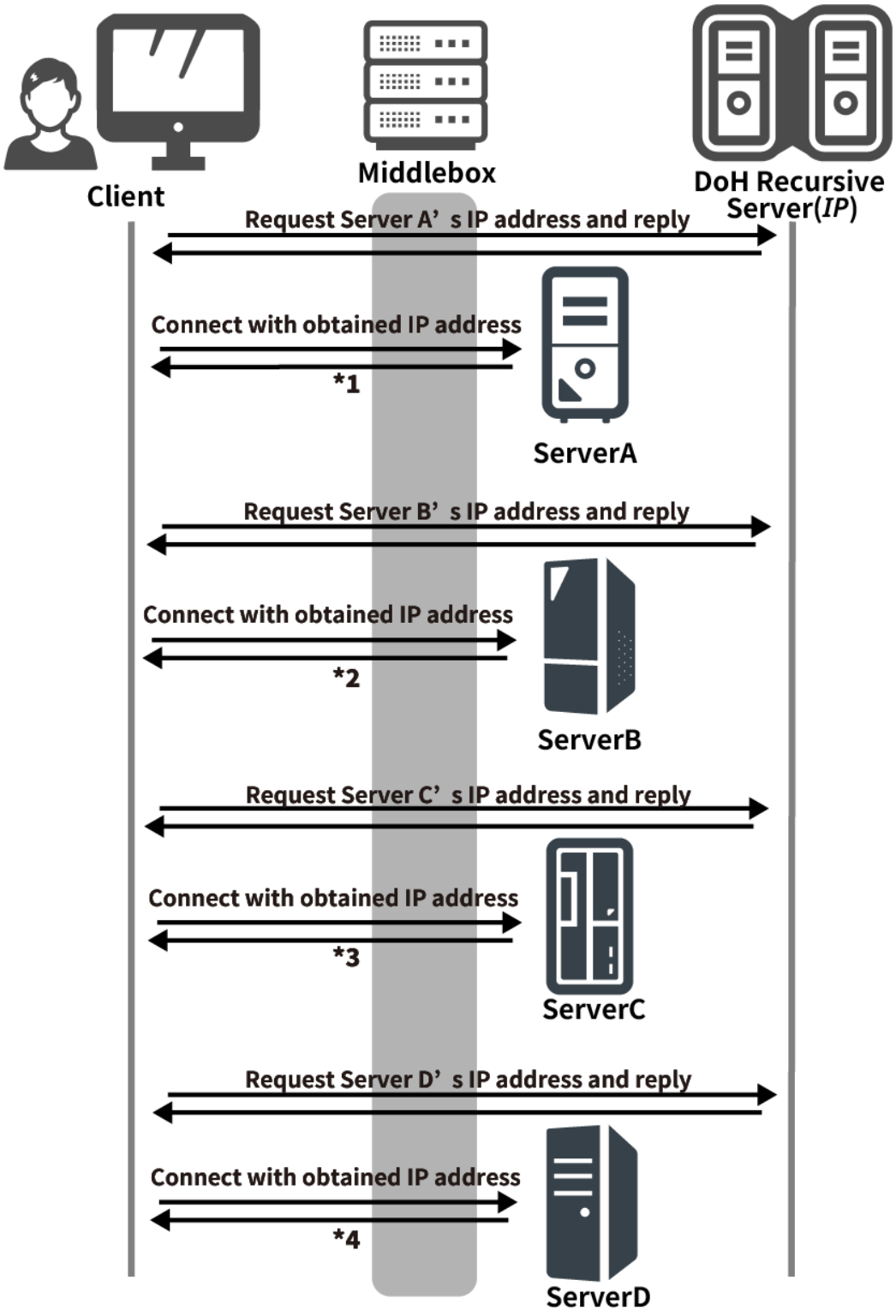

Figure 2: Flows for IP address resolution with the DoH recursive resolver and flows with new destinations.

Any destination with an extremely high average byte count is considered to be a non-DoH server. Such servers are also identified by the IQR method. That is, if the average byte count is greater than the value in Equation (2), the value is considered an outlier and the server is excluded from the list of candidate DoH servers. Table 1 also shows the meanings of the variables used in the equation.

$$IQR_l \times 1.5 + Q3_l \tag{2}$$

Since the DoH name resolution request and its response are text-based communications, the amount of data is assumed to be smaller than the amount of data involved in web browsing. Therefore, although the authors do not try alternate values for the coefficient in Equation (2), 1.5 is enough, since the value is commonly used in the IQR method [18].

Table 1: The meanings of the variables in Equations (1) and (2).

variables	meaning
IQR_c	Quartile range of CNNP.
$Q3_c$	3rd quartile of CNNP.
α	Parameter of detection accuracy.
IQR_l	Quartile range of average bytes.
$Q3_l$	3rd quartile of average bytes.

4.3 DoH destination detection algorithm

Algorithm 1 summarizes the procedures in the previous subsection. The meanings of the variables in the algorithm are shown in Table 2.

The middlebox executes Algorithm 1 every time it captures a packet. At line 1, it stores the packet if the source port number is 443 and the destination IP address is the client's. Then, at line 2, the source IP address is stored in *address*.

Lines 3 to 7 are performed when connecting to a new destination. *address_count_list* is a list of the lists of the server's IP addresses and the server's own CNNP.

At line 5, the CNNP in *address_count_list* of the IP address stored in *tmp_packet_list* is incremented by 1.

Then, line 6 sets *tmp_packet_list* to empty, and line 7 extracts the IP address of the presumed DoH communication destination.

Lines 9 through 11 are the process when connecting to a destination to which the client has previously connected. The IP addresses of the servers which the client has communicated with up until the next new destination are stored in *tmp_packet_list*. In this algorithm, the extraction is performed each time a new destination is detected. Since the algorithm has no loop and just count the number of addresses in packets, the computational complexity is $O(\log N)$, where N is the number of packets. But it does not need to capture for learning and only needs to capture the real time communications through the middlebox.

Table 2: The meanings of the variables in Algorithm 1

variables	meaning
packet_list	List of HTTPS server IP addresses which the client has communicated with
tmp_packet_list	The temporary list of HTTPS server IP addresses that the client communicated with up until the client connects with the next new server.
address_count_list	A list of the lists of HTTPS server IP addresses and CNNP to the address's server.

Algorithm 1 Detection algorithm of DoH destination

packet ← Packet sent from port 443 to the client.
address ← The source IP address of the packet.
if *address* ∉ *packet_list* **then**
  Add *address* into *packet_list*.
  CNNP in *address_count_list* of the IP address stored in *tmp_packet_list* is incremented by 1.
  Set *tmp_packet_list* to empty.
  Extract IP addresses whose CNNP in *address_count_list* satisfies $IQR_c \times \alpha + Q3_c < CNNP$ & $Averagebytes < IQR_l \times 1.5 + Q3_l$
else
  if *address* ∉ *tmp_packet_list* **then**
    Add *address* into *tmp_packet_list*.
  end if
end if

5 Experiments

This section describes the implementation of the scheme proposed in Section 4 via two experiments. The first experiment detects DoH communication in real online communication to confirm the effectiveness of the proposed method.

The second experiment applies the open captured CIRA-CIC-DoHBrw-2020 dataset [15] to the proposed method to show that the results of the first experiment are almost identical to those recorded in the captured dataset.

5.1 Experiments with actual communications

The experiment is conducted under two different scenarios. In the first scenario, a client browses web pages with DoH. In the second one, a client not only browses web pages with DoH, but also downloads files over HTTPS. In both scenarios, the client uses Firefox (v108.0) with enabled DoH by Selenium (version 4.7.2) and accesses 100 domains randomly selected from the top 10,000 domains obtained from Alexa Top Sites in December 2022. The downloaded files are placed on a VPS (Virtual Private Server) and Firefox downloads all the files one by one. The client also captures all packets using tshark.

The above process is repeated 50 times for each scenario. The captured data is then analyzed to detect DoH communication based on the proposed method with parameter α set to 10, 20, 30, 40, and 50.

Table 3: DoH configurations for Firefox

config	detail
network.trr.mode	3
network.trr.uri	https://mozilla.cloudflare-dns.com/dns-query

Table 3 shows the DoH configurations for Firefox. "network.trr.mode" is set to 3, since instead of the normal DNS, only DoH is used. In this experiment, Cloudflare's public resolver is adopted as the DoH resolver defined by "network.trr.uri". This public resolver is selected by default when DoH is enabled in Firefox. The domain name "mozilla.cloudflare-dns.com" of the Cloudflare public resolver has two A records, i.e., 104.16.249.249 and 104.16.248.249. Firefox may use these two different public resolvers during the experiment. The goal is that the proposed method can detect these two IP addresses as DoH resolvers. The detection accuracy is defined by Equation (3), i.e., the ratio of the CNNP of the correct DoH resolvers' IP addresses to the CNNP of all IP addresses detected as

possible DoH resolvers.

$$\frac{\text{CNNP of 104.16.249.249 and 104.16.248.249}}{\text{CNNP of all detected IP addresses}} \tag{3}$$

Equation (4) is an accuracy formula commonly used in the field of machine learning where TP, TN, FP, and FN indicate the numbers of True Positives, True Negatives, False Positives, and False Negatives, respectively. However, since there are at most two IP addresses of DoH recursive resolvers in this experiment, the accuracy defined by Equation (4) is greatly affected by the TN in the numerator. Therefore, we do not use Equation (4) because it is considered inappropriate for checking whether the proposed method can correctly detect DoH recursive resolvers, but we show the value in the table later as a reference.

$$\frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4}$$

5.1.1 The scenario of browsing web pages only

Figure 3 to Figure 6 show the results for the first scenario. The horizontal axis in these figures is the number of accessed IP addresses. Figure 3 shows the average detection accuracy for each parameter α. Figure 4 shows the same data as Figure 3 but as a single average of the detection accuracy with the 95% confidence interval, when α is set at 30. Figure 5 and Figure 6 show the number of experiments when the detection accuracy was 100% and 0%, respectively, across 50 experiments.

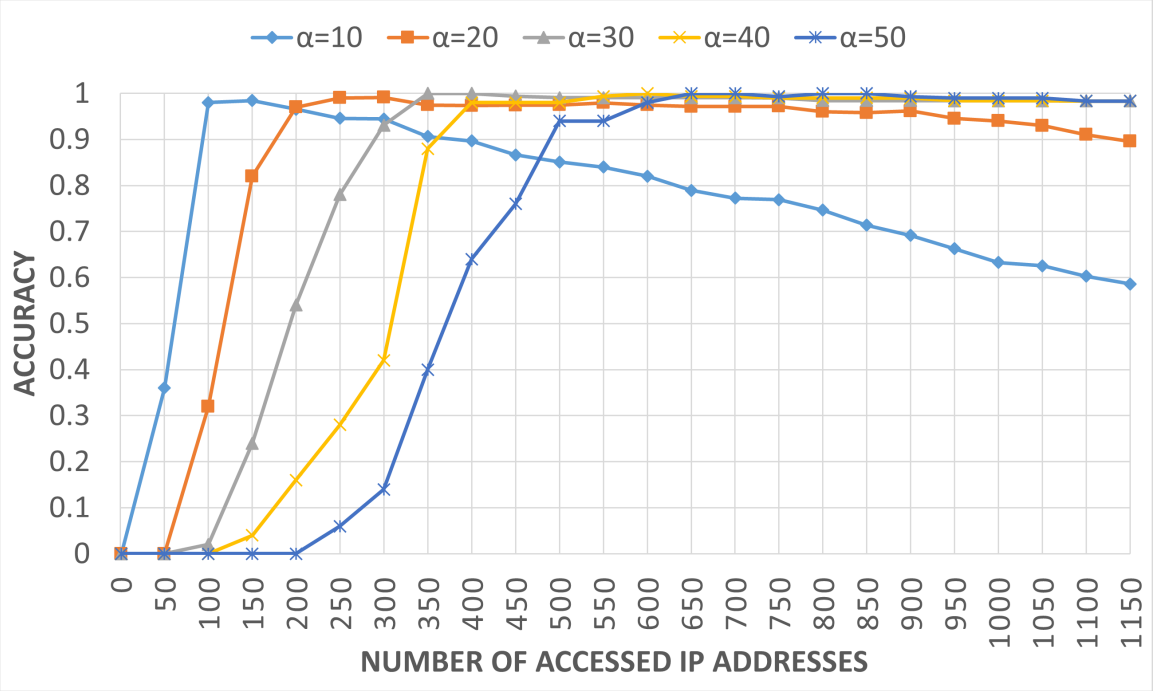

Figure 3: The average detection accuracy (web browsing only).

In Figure 3, the number of accessed IP addresses required for detection increases as α increases, because α raises the threshold for identifying outliers and increases the number of CNNP required for the detection. In Figure 4, the 95% confidence interval is larger, when the number of accessed IP addresses between 100 and 300, since the accuracy quickly increases. But the interval is very small when the number of accessed IP addresses exceeds 350. At this point, as shown in Figure 5, the majority of experiments achieve an accuracy of 100%. Since almost identical results are obtained, the 95% confidence interval is omitted in the other figures.

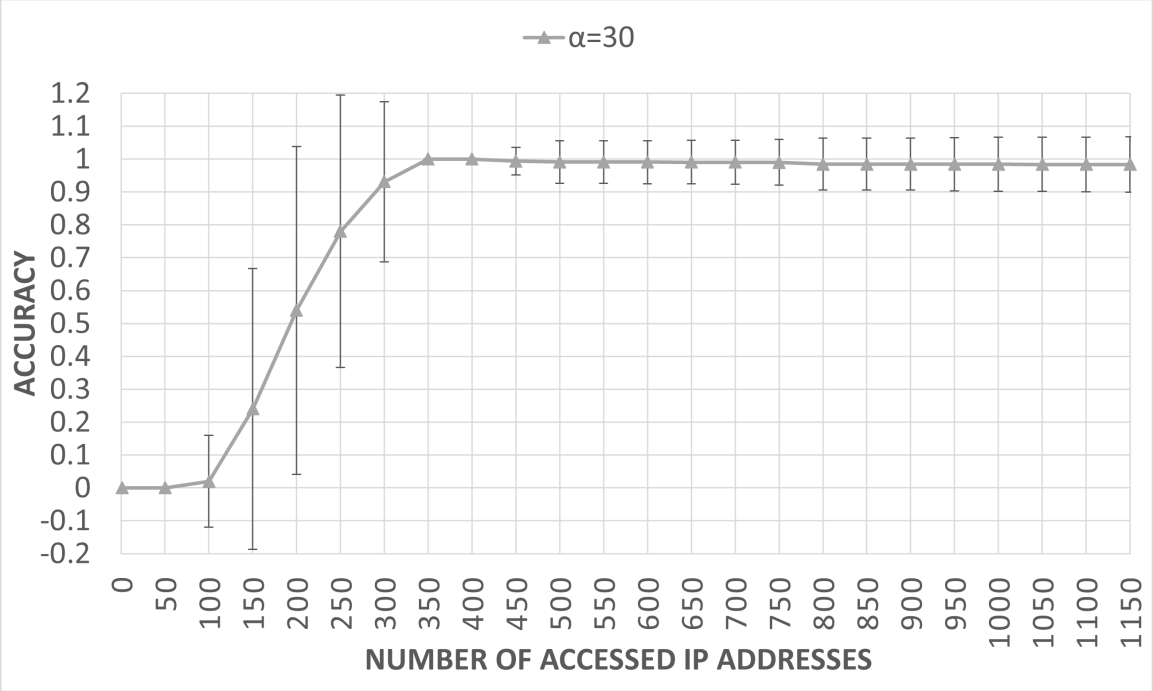

Figure 4: The average detection accuracy and the 95% confidence interval when $\alpha = 30$ (web browsing only).

For each α, the average accuracy is low until it exceeds 0.9 for the first time, because there were many experiments whose detection rate was 0% in that phase, as shown in Figure 6. Therefore, there are few false positive detections of DoH resolvers.

The lower the value of α, the smaller the number of accessed IP addresses required for DoH detection.

However, Figure 3 and Figure 5 show that as the number of accessed IP addresses increases, the accuracy decreases for $\alpha = 10, 20$, i.e., the number of false positives also increases. This indicates that the CNNP is high for some non-DoH connections.

Table 4 to Table 9 show the results analyzed in detail with parameter $\alpha = 1.5, 10, 20, 30, 40, 50$ with one experiment randomly selected from 50 experiments with web browsing only scenario.

IP indicates the number of accessed IP addresses so far at the time of analysis. ML_accuracy shown in the table is defined by Equation (4), which is used in general evaluations, and Equation (3), which is defined in this experiment, is listed as the Accuracy. Recall is defined by Equation (5) and Precision is defined by Equation (6). F-score is the harmonic mean of Recall and Precision as defined by Equation (7). Since these values are evaluated for machine learning in general, they should work well to compare with the other methods using machine learning.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

$$\text{F-score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{7}$$

CNNP_1 is the CNNP of 104.16.248.249 (DoH_1), one of the two DoH destinations, and CNNP_2 is the CNNP of 104.16.249.249 (DoH_2), the other DoH destination. In the table, "NaN" indicates

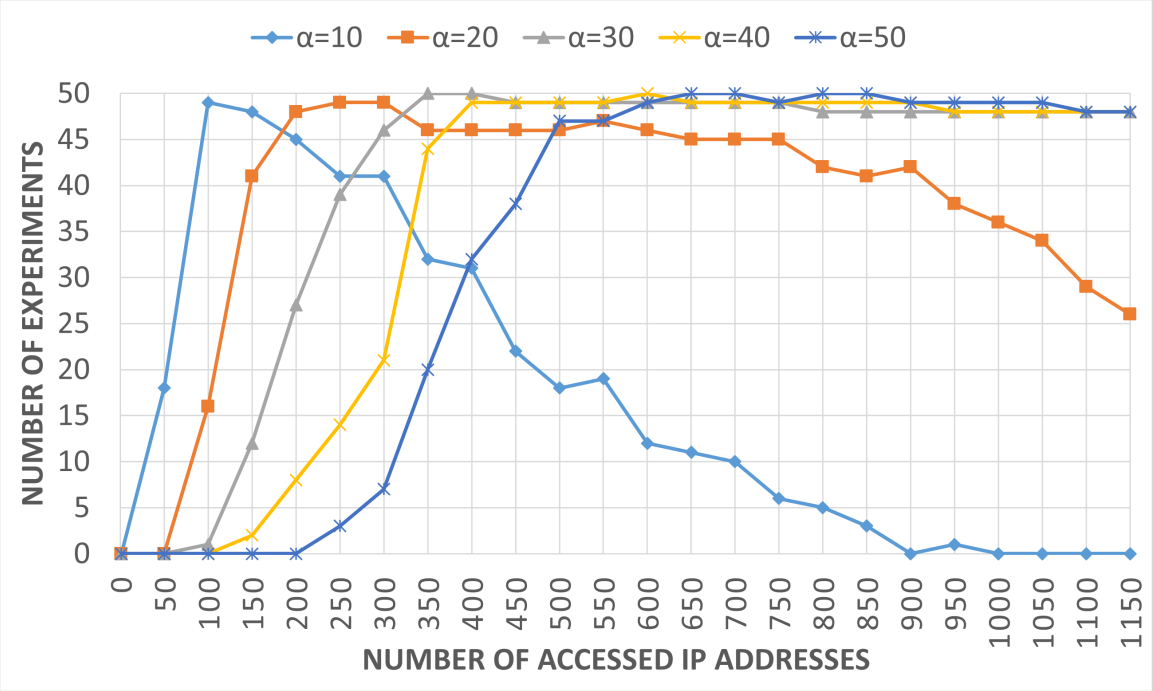

Figure 5: The number of experiments whose detection accuracy was 100% (web browsing only).

that the fractional denominator cannot be defined because it becomes zero during the calculation process. IQR_c is the Quartile range of CNNP and $Q3_c$ is the 3rd quartile of CNNP.

The analysis results for the parameter $\alpha = 1.5$ in Table 4 show that TP is 1 when the number of accessed IP addresses is 10–20, and Precision, Recall, ML_accuracy, and Accuracy are also 1, indicating that the DoH is detected correctly and quickly. However, FP appears when the number of accessed IP addresses exceeds 30, in which non-DoH destinations are detected as DoH destinations. The FP also increases as the number of accessed IP addresses increases, and Precision is 0.1667 when the number of accessed IP addresses is 100. After that, FP continues to increase and Precision reaches a very low value of 0.0135 when the number is 1150. Overall, since the F-score decreases as the number of accessed IP addresses increases, it is inappropriate to use the parameter α as a fixed value of 1.5, which is a value commonly used in interquartile outlier detection methods.

Focusing on the analysis results for parameter $\alpha = 10$ in Table 5, when the number of accessed IP addresses is 50, the CNNP of the IP address of the DoH destination does not exceed the detection threshold, so the DoH destination cannot be detected and the TP is 0. When the number of accessed IP addresses is 100–600, TP is 1 and FP is zero, indicating accurate detection. However, FP appears when the number of accessed IP addresses is around 650, and Precision, F-score, ML_accuracy and Accuracy decrease as the number of accessed IP addresses increases.

Focusing on the analysis results for parameter $\alpha = 20$ in Table 6, when the number of accessed IP addresses is 50, as with the result of parameter $\alpha = 10$, the DoH destination cannot be detected. However, there is no FP when the number of accessed IP addresses is between 100–1050, i.e., our proposed method correctly detects the DoH destination. On the other hand, false positives occur after 1100 addresses. This indicates that the parameter α should be increased to improve Precision, F-score, etc. as the number of accessed IP addresses increases.

Focusing on the analysis results for parameter $\alpha = 30, 40, 50$, in Table 7 to Table 9, there is no FP until the number of accessed IP addresses reaches 1150, i.e., no false positives have been detected. However, when the number of accessed IP addresses is small, TP is 0, indicating that the DoH destination cannot be detected quickly. Therefore, the larger the parameter is, the more difficult it is to detect the DoH destination quickly.

Table 4: Analysis results in parameter $\alpha = 1.5$ for web browsing only scenario

IP	TP	TN	FP	FN	Recall	Precision	F-score	ML_accuracy	Accuracy	$CNNP_1$	$CNNP_2$	IQR_c	$Q3_c$
10	1	9	0	0	1	1	1	1	1	0	9	3	4
20	1	19	0	0	1	1	1	1	1	0	14	4	5
30	1	28	1	0	1	0.5	0.6667	0.9667	0.5714	0	20	4	5
40	1	38	1	0	1	0.5	0.6667	0.975	0.5745	0	27	4	5
50	1	48	1	0	1	0.5	0.6667	0.98	0.6154	0	32	4	5
60	1	58	1	0	1	0.5	0.6667	0.9833	0.6491	0	37	4	5
70	1	67	2	0	1	0.3333	0.5	0.9714	0.5844	0	45	4	5
80	1	75	4	0	1	0.2	0.3333	0.95	0.4851	0	49	3	5
90	1	87	2	0	1	0.3333	0.5	0.9778	0.6322	0	55	4	5
100	1	94	5	0	1	0.1667	0.2857	0.95	0.4911	0	55	2	4
150	1	141	8	0	1	0.1111	0.2	0.9467	0.4056	0	73	3	5
200	1	189	10	0	1	0.0909	0.1667	0.95	0.36	0	99	4	6
250	1	233	16	0	1	0.0588	0.1111	0.936	0.2876	0	128	4	7
300	1	281	18	0	1	0.0526	0.1	0.94	0.307	0	163	5	7
350	1	329	20	0	1	0.0476	0.0909	0.9429	0.3099	0	198	5	7
400	1	376	23	0	1	0.0417	0.08	0.9425	0.3078	0	237	5	8
450	1	430	19	0	1	0.05	0.0952	0.9578	0.3354	0	277	6	9
500	1	481	18	0	1	0.0526	0.1	0.964	0.3484	0	301	7	10
550	1	526	23	0	1	0.0417	0.08	0.9582	0.3121	0	333	7	10
600	1	572	27	0	1	0.0357	0.069	0.955	0.2905	0	366	7	10
650	1	611	38	0	1	0.0256	0.05	0.9415	0.2366	0	406	7	11
700	1	651	48	0	1	0.0204	0.04	0.9314	0.2076	0	449	7	11
750	1	697	52	0	1	0.0189	0.037	0.9307	0.2007	0	483	7	11
800	1	744	55	0	1	0.0179	0.0351	0.9313	0.2034	0	524	7	11
850	1	786	63	0	1	0.0156	0.0308	0.9259	0.1916	0	559	7	11
900	1	835	64	0	1	0.0154	0.0303	0.9289	0.1957	0	595	7	11
950	1	882	67	0	1	0.0147	0.029	0.9295	0.1961	0	629	7	11
1000	1	942	57	0	1	0.0172	0.0339	0.943	0.2111	0	662	8	12
1050	1	988	61	0	1	0.0161	0.0317	0.9419	0.2072	0	700	8	12
1100	1	1029	70	0	1	0.0141	0.0278	0.9364	0.1949	0	743	8	12
1150	1	1076	73	0	1	0.0135	0.0267	0.9365	0.1904	0	782	8	12

Table 5: Analysis results in parameter $\alpha = 10$ for web browsing only scenario

IP	TP	TN	FP	FN	Recall	Precision	F-score	ML_accuracy	Accuracy	$CNNP_1$	$CNNP_2$	IQR_c	$Q3_c$
50	0	49	0	1	0	NaN	NaN	0.98	0	0	32	4	5
100	1	99	0	0	1	1	1	1	1	0	55	2	4
150	1	149	0	0	1	1	1	1	1	0	73	3	5
200	1	199	0	0	1	1	1	1	1	0	99	4	6
250	1	249	0	0	1	1	1	1	1	0	128	4	7
300	1	299	0	0	1	1	1	1	1	0	163	5	7
350	1	349	0	0	1	1	1	1	1	0	198	5	7
400	1	399	0	0	1	1	1	1	1	0	237	5	8
450	1	449	0	0	1	1	1	1	1	0	277	6	9
500	1	499	0	0	1	1	1	1	1	0	301	7	10
550	1	549	0	0	1	1	1	1	1	0	333	7	10
600	1	599	0	0	1	1	1	1	1	0	366	7	10
650	1	648	1	0	1	0.5	0.6667	0.9985	0.8104	0	406	7	11
700	1	698	1	0	1	0.5	0.6667	0.9986	0.8018	0	449	7	11
750	1	748	1	0	1	0.5	0.6667	0.9987	0.7997	0	483	7	11
800	1	797	2	0	1	0.3333	0.5	0.9975	0.7129	0	524	7	11
850	1	846	3	0	1	0.25	0.4	0.9965	0.644	0	559	7	11
900	1	896	3	0	1	0.25	0.4	0.9967	0.6467	0	595	7	11
950	1	945	4	0	1	0.2	0.3333	0.9958	0.5985	0	629	7	11
1000	1	997	2	0	1	0.3333	0.5	0.998	0.7065	0	662	8	12
1050	1	1046	3	0	1	0.25	0.4	0.9971	0.6434	0	700	8	12
1100	1	1095	4	0	1	0.2	0.3333	0.9964	0.5944	0	743	8	12
1150	1	1145	4	0	1	0.2	0.3333	0.9965	0.592	0	782	8	12

Table 6: Analysis results in parameter $\alpha = 20$ for web browsing only scenario

IP	TP	TN	FP	FN	Recall	Precision	F-score	ML_accuracy	Accuracy	$CNNP_1$	$CNNP_2$	IQR_c	$Q3_c$
50	0	49	0	1	0	NaN	NaN	0.98	0	0	32	4	5
100	1	99	0	0	1	1	1	1	1	0	55	2	4
150	1	149	0	0	1	1	1	1	1	0	73	3	5
200	1	199	0	0	1	1	1	1	1	0	99	4	6
250	1	249	0	0	1	1	1	1	1	0	128	4	7
300	1	299	0	0	1	1	1	1	1	0	163	5	7
350	1	349	0	0	1	1	1	1	1	0	198	5	7
400	1	399	0	0	1	1	1	1	1	0	237	5	8
450	1	449	0	0	1	1	1	1	1	0	277	6	9
500	1	499	0	0	1	1	1	1	1	0	301	7	10
550	1	549	0	0	1	1	1	1	1	0	333	7	10
600	1	599	0	0	1	1	1	1	1	0	366	7	10
650	1	649	0	0	1	1	1	1	1	0	406	7	11
700	1	699	0	0	1	1	1	1	1	0	449	7	11
750	1	749	0	0	1	1	1	1	1	0	483	7	11
800	1	799	0	0	1	1	1	1	1	0	524	7	11
850	1	849	0	0	1	1	1	1	1	0	559	7	11
900	1	899	0	0	1	1	1	1	1	0	595	7	11
950	1	949	0	0	1	1	1	1	1	0	629	7	11
1000	1	999	0	0	1	1	1	1	1	0	662	8	12
1050	1	1049	0	0	1	1	1	1	1	0	700	8	12
1100	1	1098	1	0	1	0.5	0.6667	0.9991	0.805	0	743	8	12
1150	1	1148	1	0	1	0.5	0.6667	0.9991	0.8004	0	782	8	12

Table 7: Analysis results in parameter $\alpha = 30$ for web browsing only scenario

IP	TP	TN	FP	FN	Recall	Precision	F-score	ML_accuracy	Accuracy	$CNNP_1$	$CNNP_2$	IQR_c	$Q3_c$
50	0	49	0	1	0	NaN	NaN	0.98	0	0	32	4	5
100	0	99	0	1	0	NaN	NaN	0.99	0	0	55	2	4
150	0	149	0	1	0	NaN	NaN	0.9933	0	0	73	3	5
200	0	199	0	1	0	NaN	NaN	0.995	0	0	99	4	6
250	1	249	0	0	1	1	1	1	1	0	128	4	7
300	1	299	0	0	1	1	1	1	1	0	163	5	7
350	1	349	0	0	1	1	1	1	1	0	198	5	7
400	1	399	0	0	1	1	1	1	1	0	237	5	8
450	1	449	0	0	1	1	1	1	1	0	277	6	9
500	1	499	0	0	1	1	1	1	1	0	301	7	10
550	1	549	0	0	1	1	1	1	1	0	333	7	10
600	1	599	0	0	1	1	1	1	1	0	366	7	10
650	1	649	0	0	1	1	1	1	1	0	406	7	11
700	1	699	0	0	1	1	1	1	1	0	449	7	11
750	1	749	0	0	1	1	1	1	1	0	483	7	11
800	1	799	0	0	1	1	1	1	1	0	524	7	11
850	1	849	0	0	1	1	1	1	1	0	559	7	11
900	1	899	0	0	1	1	1	1	1	0	595	7	11
950	1	949	0	0	1	1	1	1	1	0	629	7	11
1000	1	999	0	0	1	1	1	1	1	0	662	8	12
1050	1	1049	0	0	1	1	1	1	1	0	700	8	12
1100	1	1099	0	0	1	1	1	1	1	0	743	8	12
1150	1	1149	0	0	1	1	1	1	1	0	782	8	12

Table 8: Analysis results in parameter $\alpha = 40$ for web browsing only scenario

IP	TP	TN	FP	FN	Recall	Precision	F-score	ML_accuracy	Accuracy	$CNNP_1$	$CNNP_2$	IQR_c	$Q3_c$
50	0	49	0	1	0	NaN	NaN	0.98	0	0	32	4	5
100	0	99	0	1	0	NaN	NaN	0.99	0	0	55	2	4
150	0	149	0	1	0	NaN	NaN	0.9933	0	0	73	3	5
200	0	199	0	1	0	NaN	NaN	0.995	0	0	99	4	6
250	0	249	0	1	0	NaN	NaN	0.996	0	0	128	4	7
300	0	299	0	1	0	NaN	NaN	0.9967	0	0	163	5	7
350	0	349	0	1	0	NaN	NaN	0.9971	0	0	198	5	7
400	1	399	0	0	1	1	1	1	1	0	237	5	8
450	1	449	0	0	1	1	1	1	1	0	277	6	9
500	1	499	0	0	1	1	1	1	1	0	301	7	10
550	1	549	0	0	1	1	1	1	1	0	333	7	10
600	1	599	0	0	1	1	1	1	1	0	366	7	10
650	1	649	0	0	1	1	1	1	1	0	406	7	11
700	1	699	0	0	1	1	1	1	1	0	449	7	11
750	1	749	0	0	1	1	1	1	1	0	483	7	11
800	1	799	0	0	1	1	1	1	1	0	524	7	11
850	1	849	0	0	1	1	1	1	1	0	559	7	11
900	1	899	0	0	1	1	1	1	1	0	595	7	11
950	1	949	0	0	1	1	1	1	1	0	629	7	11
1000	1	999	0	0	1	1	1	1	1	0	662	8	12
1050	1	1049	0	0	1	1	1	1	1	0	700	8	12
1100	1	1099	0	0	1	1	1	1	1	0	743	8	12
1150	1	1149	0	0	1	1	1	1	1	0	782	8	12

Table 9: Analysis results in parameter $\alpha = 50$ for web browsing only scenario

IP	TP	TN	FP	FN	Recall	Precision	F-score	ML_accuracy	Accuracy	$CNNP_1$	$CNNP_2$	IQR_c	$Q3_c$
50	0	49	0	1	0	NaN	NaN	0.98	0	0	32	4	5
100	0	99	0	1	0	NaN	NaN	0.99	0	0	55	2	4
150	0	149	0	1	0	NaN	NaN	0.9933	0	0	73	3	5
200	0	199	0	1	0	NaN	NaN	0.995	0	0	99	4	6
250	0	249	0	1	0	NaN	NaN	0.996	0	0	128	4	7
300	0	299	0	1	0	NaN	NaN	0.9967	0	0	163	5	7
350	0	349	0	1	0	NaN	NaN	0.9971	0	0	198	5	7
400	0	399	0	1	0	NaN	NaN	0.9975	0	0	237	5	8
450	0	449	0	1	0	NaN	NaN	0.9978	0	0	277	6	9
500	0	499	0	1	0	NaN	NaN	0.998	0	0	301	7	10
550	0	549	0	1	0	NaN	NaN	0.9982	0	0	333	7	10
600	1	599	0	0	1	1	1	1	1	0	366	7	10
650	1	649	0	0	1	1	1	1	1	0	406	7	11
700	1	699	0	0	1	1	1	1	1	0	449	7	11
750	1	749	0	0	1	1	1	1	1	0	483	7	11
800	1	799	0	0	1	1	1	1	1	0	524	7	11
850	1	849	0	0	1	1	1	1	1	0	559	7	11
900	1	899	0	0	1	1	1	1	1	0	595	7	11
950	1	949	0	0	1	1	1	1	1	0	629	7	11
1000	1	999	0	0	1	1	1	1	1	0	662	8	12
1050	1	1049	0	0	1	1	1	1	1	0	700	8	12
1100	1	1099	0	0	1	1	1	1	1	0	743	8	12
1150	1	1149	0	0	1	1	1	1	1	0	782	8	12

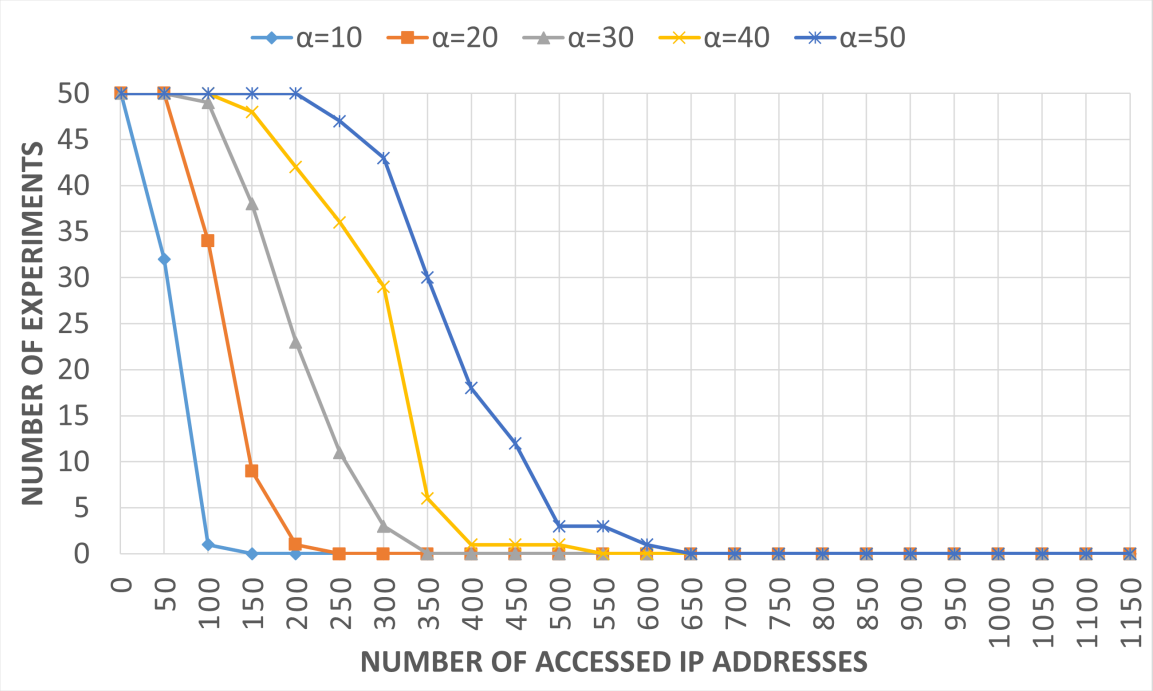

Figure 6: The number of experiments whose detection accuracy was 0% (web browsing only).

In summary, when the parameter α is small, it is possible to detect the DoH destination early and Precision will be 1. However, if the number of IP addresses is large, false positives will increase and Precision will decrease. On the other hand, when the parameter α is large, it is impossible to detect the DoH destination at an early stage, and Recall remains zero. As the number of accessed IP addresses increases, it becomes possible to detect with a high degree of accuracy. In addition, Precision can be high when it is detected. Therefore, it is considered that the proposed method can effectively detect DoH destinations by varying the parameter α according to the number of accessed IP addresses, rather than using a fixed value. In detail, α should set to a small value in the initial stage of the analysis and set to a larger value as the number of accessed IP addresses increases.

5.1.2 The scenario of browsing web pages and HTTPS file transfers in parallel

Next, Figure 7 to Figure 8 show the results for the second scenario. The horizontal axis in these figures is also the number of accessed IP addresses. Figure 7 shows the average of detection accuracy for each α. Figure 8 and Figure 9 show the number of experiments with a detection accuracy of 100% and 0%, respectively, across 50 experiments.

When a web browser downloads a file using HTTPS, the CNNP for the HTTPS server that provides the file increases. Therefore, if the IP address is incorrectly detected, the denominator of Equation (3) becomes very large, and the detection accuracy is expected to be very low. However, comparing these results with the results of the experiments performed without file downloading, both have similar trends. For example, in Figure 3 and Figure 7, the average detection accuracy rises up when the number of accessed IP addresses is between 0 and about 500 in relation to the value of α. After the average detection accuracy rises to 1, the average converges to 1 if $\alpha \geq 30$ or gradually decreases otherwise. Figure 5 and Figure 8 show similar graphs to Figure 3 and Figure 7. In Figure 6 and Figure 9, the average detection accuracy falls from 1 to 0 when the number of accessed IP addresses is between 0 and about 500 in relation to the value of α and the average converges to 0. This is because the proposed method excludes from the detection destinations that transfers extremely large numbers of bytes.

Table 10 to Table 15 show the results analyzed in detail with parameter $\alpha = 1.5, 10, 20, 30, 40, 50$

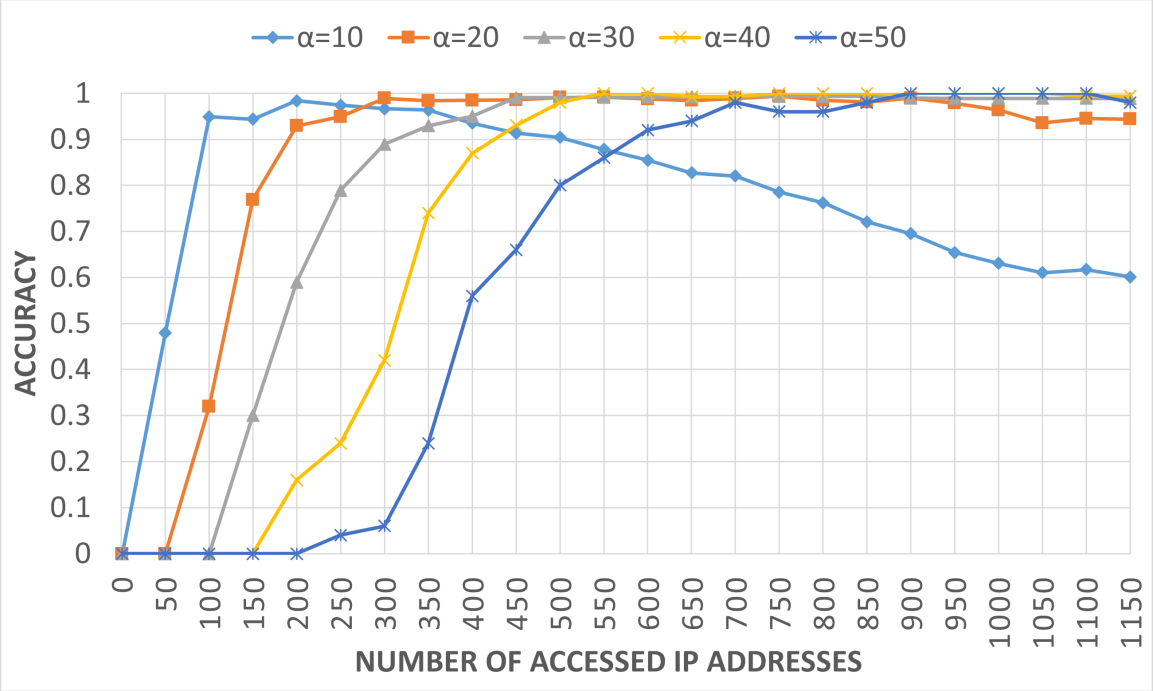

Figure 7: The average detection accuracy (web browsing and file downloading).

with one experiment randomly selected from 50 experiments with web browsing and background file transfer scenario. For each result of the value of parameter α, the trend is roughly the same as in the web browsing only experiment described above. When a web browser downloads a file via a HTTPS communication, the CNNP at the destination IP address becomes large. Therefore, it is assumed that the IP address of the file provider is always detected and FP shown in the table is always 1 or more if the exclusion process based on the average number of bytes is not performed. However, the fact that FP is always zero, especially in Table 13 to Table 15, indicates that the exclusion process of the proposed method works well. The above results suggest that the proposed method effectively excludes destinations whose average transferred bytes is extremely large.

As the number of accessed IP addresses for each parameter increases, Recall becomes 0.5 because TP is 1 and FN is 1. This is a trend that does not appear in Table 4 to Table 9, which show the results for web browsing only. If we look at CNNP_1 and CNNP_2 in this area, we can see that only CNNP_2 has been increasing continuously, but CNNP_1 has started to increase gradually. This is thought to be because Firefox continued to use one (104.16.249.249) of the two IP addresses associated with the domain that is set as the DoH communication destination for name resolution, but at some point Firefox started communicating with the other IP address (104.16.248.249). Therefore, it is assumed that file downloads did not directly affect the results of the proposed method. CNNP_1 has a much gradual increase compared to the rate of increase in CNNP_2, which is less than $Q3_c$. It is therefore assumed that 104.16.249.249 (DoH_2) will continue to be used for the main name resolution, and that there will be little communication with 104.16.248.249 (DoH_1). Even if we block the detected first DoH communication target, 104.16.249.249 (DoH_2), and then Firefox starts using 104.16.248.249 (DoH_1) for main name resolution, then we can detect the second target in the same way.

After the falsely detected IP addresses were analyzed, they were found to sometimes include IP addresses owned by Google Inc. Since many web pages use Google Analytics and Google AdSense and fonts and Javascript libraries are placed in CDNs (Content Delivery Networks), false positives may be detected when some content provided by the same provider from different sources are placed on each web page.

Table 10: Analysis results in parameter $\alpha = 1.5$ for web browsing and background file transfer scenario

IP	TP	TN	FP	FN	Recall	Precision	F-score	ML_accuracy	Accuracy	$CNNP_1$	$CNNP_2$	IQR_c	$Q3_c$
10	1	9	0	0	1	1	1	1	1	0	9	3	4
20	1	19	0	0	1	1	1	1	1	0	15	4	5.5
30	1	29	0	0	1	1	1	1	1	0	24	4	5
40	1	39	0	0	1	1	1	1	1	0	31	4	5
50	1	49	0	0	1	1	1	1	1	0	36	4	5
60	1	59	0	0	1	1	1	1	1	0	44	4	5
70	1	69	0	0	1	1	1	1	1	0	50	5	6
80	1	78	1	0	1	0.5	0.6667	0.9875	0.803	0	53	3.5	5
90	1	85	4	0	1	0.2	0.3333	0.9556	0.5825	0	60	3	5
100	1	95	4	0	1	0.2	0.3333	0.96	0.6106	0	69	3	4.5
150	1	141	8	0	1	0.1111	0.2	0.9467	0.505	0	102	3	5
200	1	185	14	0	1	0.0667	0.125	0.93	0.4379	0	134	3	5
250	1	240	9	0	1	0.1	0.1818	0.964	0.5189	0	165	4	6
300	1	287	12	0	1	0.0769	0.1429	0.96	0.4567	0	190	5	7
350	1	329	20	0	1	0.0476	0.0909	0.9429	0.3711	0	226	4	7
400	1	372	27	0	1	0.0357	0.069	0.9325	0.3359	0	266	4	7
450	1	424	25	0	1	0.0385	0.0741	0.9444	0.3639	0	302	5	8
500	1	464	35	0	1	0.0278	0.0541	0.93	0.3061	0	337	5	8
550	1	521	27	1	0.5	0.0357	0.0667	0.9491	0.3427	1	377	6	9
600	1	568	30	1	0.5	0.0323	0.0606	0.9483	0.3282	1	404	6	9
650	1	611	37	1	0.5	0.0263	0.05	0.9415	0.2912	1	433	6	9
700	1	653	45	1	0.5	0.0217	0.0417	0.9343	0.2663	2	478	6	9
750	1	698	50	1	0.5	0.0196	0.0377	0.932	0.2599	2	517	6	9
800	1	747	51	1	0.5	0.0192	0.037	0.935	0.2558	2	551	6	9
850	1	797	51	1	0.5	0.0192	0.037	0.9388	0.2568	2	593	6	10
900	1	845	53	1	0.5	0.0185	0.0357	0.94	0.2574	3	637	6	10
950	1	889	59	1	0.5	0.0167	0.0323	0.9368	0.2454	3	670	6	10
1000	1	933	65	1	0.5	0.0152	0.0294	0.934	0.2363	3	703	6	10
1050	1	974	74	1	0.5	0.0133	0.026	0.9286	0.228	4	742	6	10
1100	1	1015	83	1	0.5	0.0119	0.0233	0.9236	0.2161	4	784	6	10
1150	1	1073	75	1	0.5	0.0132	0.0256	0.9339	0.2246	5	820	7	11

Table 11: Analysis results in parameter $\alpha = 10$ for web browsing and background file transfer scenario

IP	TP	TN	FP	FN	Recall	Precision	F-score	ML_accuracy	Accuracy	$CNNP_1$	$CNNP_2$	IQR_c	$Q3_c$
50	0	49	0	1	0	NaN	NaN	0.98	0	0	36	4	5
100	1	99	0	0	1	1	1	1	1	0	69	3	4.5
150	1	149	0	0	1	1	1	1	1	0	102	3	5
200	1	199	0	0	1	1	1	1	1	0	134	3	5
250	1	249	0	0	1	1	1	1	1	0	165	4	6
300	1	299	0	0	1	1	1	1	1	0	190	5	7
350	1	349	0	0	1	1	1	1	1	0	226	4	7
400	1	399	0	0	1	1	1	1	1	0	266	4	7
450	1	449	0	0	1	1	1	1	1	0	302	5	8
500	1	499	0	0	1	1	1	1	1	0	337	5	8
550	1	548	0	1	0.5	1	0.6667	0.9982	1	1	377	6	9
600	1	598	0	1	0.5	1	0.6667	0.9983	1	1	404	6	9
650	1	647	1	1	0.5	0.5	0.5	0.9969	0.8474	1	433	6	9
700	1	697	1	1	0.5	0.5	0.5	0.9971	0.8415	2	478	6	9
750	1	746	2	1	0.5	0.3333	0.4	0.996	0.7547	2	517	6	9
800	1	795	3	1	0.5	0.25	0.3333	0.995	0.6888	2	551	6	9
850	1	845	3	1	0.5	0.25	0.3333	0.9953	0.6895	2	593	6	10
900	1	895	3	1	0.5	0.25	0.3333	0.9956	0.6947	3	637	6	10
950	1	943	5	1	0.5	0.1667	0.25	0.9937	0.6014	3	670	6	10
1000	1	993	5	1	0.5	0.1667	0.25	0.994	0.5927	3	703	6	10
1050	1	1043	5	1	0.5	0.1667	0.25	0.9943	0.6013	4	742	6	10
1100	1	1093	5	1	0.5	0.1667	0.25	0.9945	0.6049	4	784	6	10
1150	1	1143	5	1	0.5	0.1667	0.25	0.9948	0.6097	5	820	7	11

Table 12: Analysis results in parameter $\alpha = 20$ for web browsing and background file transfer scenario

IP	TP	TN	FP	FN	Recall	Precision	F-score	ML_accuracy	Accuracy	$CNNP_1$	$CNNP_2$	IQR_c	$Q3_c$
50	0	49	0	1	0	NaN	NaN	0.98	0	0	36	4	5
100	1	99	0	0	1	1	1	1	1	0	69	3	4.5
150	1	149	0	0	1	1	1	1	1	0	102	3	5
200	1	199	0	0	1	1	1	1	1	0	134	3	5
250	1	249	0	0	1	1	1	1	1	0	165	4	6
300	1	299	0	0	1	1	1	1	1	0	190	5	7
350	1	349	0	0	1	1	1	1	1	0	226	4	7
400	1	399	0	0	1	1	1	1	1	0	266	4	7
450	1	449	0	0	1	1	1	1	1	0	302	5	8
500	1	499	0	0	1	1	1	1	1	0	337	5	8
550	1	548	0	1	0.5	1	0.6667	0.9982	1	1	377	6	9
600	1	598	0	1	0.5	1	0.6667	0.9983	1	1	404	6	9
650	1	648	0	1	0.5	1	0.6667	0.9985	1	1	433	6	9
700	1	698	0	1	0.5	1	0.6667	0.9986	1	2	478	6	9
750	1	748	0	1	0.5	1	0.6667	0.9987	1	2	517	6	9
800	1	798	0	1	0.5	1	0.6667	0.9988	1	2	551	6	9
850	1	848	0	1	0.5	1	0.6667	0.9988	1	2	593	6	10
900	1	898	0	1	0.5	1	0.6667	0.9989	1	3	637	6	10
950	1	948	0	1	0.5	1	0.6667	0.9989	1	3	670	6	10
1000	1	997	1	1	0.5	0.5	0.5	0.998	0.8419	3	703	6	10
1050	1	1047	1	1	0.5	0.5	0.5	0.9981	0.849	4	742	6	10
1100	1	1097	1	1	0.5	0.5	0.5	0.9982	0.8531	4	784	6	10
1150	1	1148	0	1	0.5	1	0.6667	0.9991	1	5	820	7	11

Table 13: Analysis results in parameter $\alpha = 30$ for web browsing and background file transfer scenario

IP	TP	TN	FP	FN	Recall	Precision	F-score	ML_accuracy	Accuracy	$CNNP_1$	$CNNP_2$	IQR_c	$Q3_c$
50	0	49	0	1	0	NaN	NaN	0.98	0	0	36	4	5
100	0	99	0	1	0	NaN	NaN	0.99	0	0	69	3	4.5
150	1	149	0	0	1	1	1	1	1	0	102	3	5
200	1	199	0	0	1	1	1	1	1	0	134	3	5
250	1	249	0	0	1	1	1	1	1	0	165	4	6
300	1	299	0	0	1	1	1	1	1	0	190	5	7
350	1	349	0	0	1	1	1	1	1	0	226	4	7
400	1	399	0	0	1	1	1	1	1	0	266	4	7
450	1	449	0	0	1	1	1	1	1	0	302	5	8
500	1	499	0	0	1	1	1	1	1	0	337	5	8
550	1	548	0	1	0.5	1	0.6667	0.9982	1	1	377	6	9
600	1	598	0	1	0.5	1	0.6667	0.9983	1	1	404	6	9
650	1	648	0	1	0.5	1	0.6667	0.9985	1	1	433	6	9
700	1	698	0	1	0.5	1	0.6667	0.9986	1	2	478	6	9
750	1	748	0	1	0.5	1	0.6667	0.9987	1	2	517	6	9
800	1	798	0	1	0.5	1	0.6667	0.9988	1	2	551	6	9
850	1	848	0	1	0.5	1	0.6667	0.9988	1	2	593	6	10
900	1	898	0	1	0.5	1	0.6667	0.9989	1	3	637	6	10
950	1	948	0	1	0.5	1	0.6667	0.9989	1	3	670	6	10
1000	1	998	0	1	0.5	1	0.6667	0.999	1	3	703	6	10
1050	1	1048	0	1	0.5	1	0.6667	0.999	1	4	742	6	10
1100	1	1098	0	1	0.5	1	0.6667	0.9991	1	4	784	6	10
1150	1	1148	0	1	0.5	1	0.6667	0.9991	1	5	820	7	11

Table 14: Analysis results in parameter $\alpha = 40$ for web browsing and background file transfer scenario

IP	TP	TN	FP	FN	Recall	Precision	F-score	ML_accuracy	Accuracy	$CNNP_1$	$CNNP_2$	IQR_c	$Q3_c$
50	0	49	0	1	0	NaN	NaN	0.98	0	0	36	4	5
100	0	99	0	1	0	NaN	NaN	0.99	0	0	69	3	4.5
150	0	149	0	1	0	NaN	NaN	0.9933	0	0	102	3	5
200	1	199	0	0	1	1	1	1	1	0	134	3	5
250	0	249	0	1	0	NaN	NaN	0.996	0	0	165	4	6
300	0	299	0	1	0	NaN	NaN	0.9967	0	0	190	5	7
350	1	349	0	0	1	1	1	1	1	0	226	4	7
400	1	399	0	0	1	1	1	1	1	0	266	4	7
450	1	449	0	0	1	1	1	1	1	0	302	5	8
500	1	499	0	0	1	1	1	1	1	0	337	5	8
550	1	548	0	1	0.5	1	0.6667	0.9982	1	1	377	6	9
600	1	598	0	1	0.5	1	0.6667	0.9983	1	1	404	6	9
650	1	648	0	1	0.5	1	0.6667	0.9985	1	1	433	6	9
700	1	698	0	1	0.5	1	0.6667	0.9986	1	2	478	6	9
750	1	748	0	1	0.5	1	0.6667	0.9987	1	2	517	6	9
800	1	798	0	1	0.5	1	0.6667	0.9988	1	2	551	6	9
850	1	848	0	1	0.5	1	0.6667	0.9988	1	2	593	6	10
900	1	898	0	1	0.5	1	0.6667	0.9989	1	3	637	6	10
950	1	948	0	1	0.5	1	0.6667	0.9989	1	3	670	6	10
1000	1	998	0	1	0.5	1	0.6667	0.999	1	3	703	6	10
1050	1	1048	0	1	0.5	1	0.6667	0.999	1	4	742	6	10
1100	1	1098	0	1	0.5	1	0.6667	0.9991	1	4	784	6	10
1150	1	1148	0	1	0.5	1	0.6667	0.9991	1	5	820	7	11

Table 15: Analysis results in parameter $\alpha = 50$ for web browsing and background file transfer scenario

IP	TP	TN	FP	FN	Recall	Precision	F-score	ML_accuracy	Accuracy	$CNNP_1$	$CNNP_2$	IQR_c	$Q3_c$
50	0	49	0	1	0	NaN	NaN	0.98	0	0	36	4	5
100	0	99	0	1	0	NaN	NaN	0.99	0	0	69	3	4.5
150	0	149	0	1	0	NaN	NaN	0.9933	0	0	102	3	5
200	0	199	0	1	0	NaN	NaN	0.995	0	0	134	3	5
250	0	249	0	1	0	NaN	NaN	0.996	0	0	165	4	6
300	0	299	0	1	0	NaN	NaN	0.9967	0	0	190	5	7
350	1	349	0	0	1	1	1	1	1	0	226	4	7
400	1	399	0	0	1	1	1	1	1	0	266	4	7
450	1	449	0	0	1	1	1	1	1	0	302	5	8
500	1	499	0	0	1	1	1	1	1	0	337	5	8
550	1	548	0	1	0.5	1	0.6667	0.9982	1	1	377	6	9
600	1	598	0	1	0.5	1	0.6667	0.9983	1	1	404	6	9
650	1	648	0	1	0.5	1	0.6667	0.9985	1	1	433	6	9
700	1	698	0	1	0.5	1	0.6667	0.9986	1	2	478	6	9
750	1	748	0	1	0.5	1	0.6667	0.9987	1	2	517	6	9
800	1	798	0	1	0.5	1	0.6667	0.9988	1	2	551	6	9
850	1	848	0	1	0.5	1	0.6667	0.9988	1	2	593	6	10
900	1	898	0	1	0.5	1	0.6667	0.9989	1	3	637	6	10
950	1	948	0	1	0.5	1	0.6667	0.9989	1	3	670	6	10
1000	1	998	0	1	0.5	1	0.6667	0.999	1	3	703	6	10
1050	1	1048	0	1	0.5	1	0.6667	0.999	1	4	742	6	10
1100	1	1098	0	1	0.5	1	0.6667	0.9991	1	4	784	6	10
1150	1	1148	0	1	0.5	1	0.6667	0.9991	1	5	820	7	11

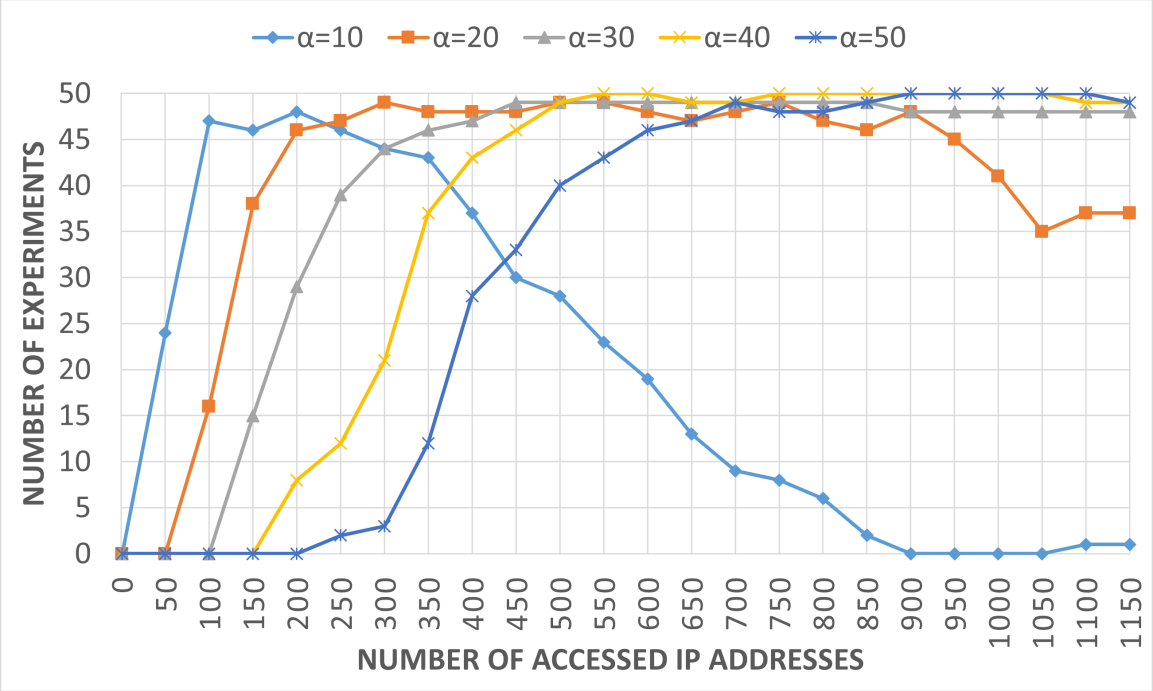

Figure 8: The number of experiments whose detection accuracy was 100% (web browsing and file downloading).

5.2 Experiments with datasets

Although the first experiment generated web browsing and file download traffic, such traffic may not be generated during actual communication. To show the accuracy and generalizability of the first experiment, this subsection uses CIRA-CIC-DoHBrw-2020 [15] open dataset. This dataset includes not only the data captured when DoH is used on browsers, but also the feature data of HTTPS packets extracted by DoHlyzer [16]. Based on this dataset, the experiment is conducted with six different scenarios using Chrome and Firefox as browsers, and Cloudflare, Google Chrome, and Quad9 as DoH recursive resolvers. The detection accuracy is defined in Equation (8). The accuracy is defined as the ratio of the CNNP of the correct DoH resolvers' IP addresses to the total CNNP of all IP addresses detected as possible DoH resolvers.

$$\frac{\text{CNNP of DoH recursive server's IP addresses shown by [15]}}{\text{CNNP of all extracted IP addresses}} \tag{8}$$

5.2.1 Google Chrome

There are a number of captured items of data in the dataset where Google Chrome was the browser. Therefore, we analyzed 5 each of these data items using Cloudflare, Google, and Quad9 as DoH recursive resolvers using the proposed method and obtained the average of the accuracy. Figure 10 to Figure 12 show the average detection accuracy for the 5 captured data items when a client used Google Chrome for the browser and selected Cloudflare, Google, and Quad9 for the DoH recursive resolvers. From these figures, larger α requires more accessed IP addresses. For smaller α, fewer accessed IP addresses are required for detection. But even when the number of accessed IP addresses increases, more false positives are detected. These results are similar to those for Firefox, which was used in the first experiments, although the detection accuracy does not converge to 1.

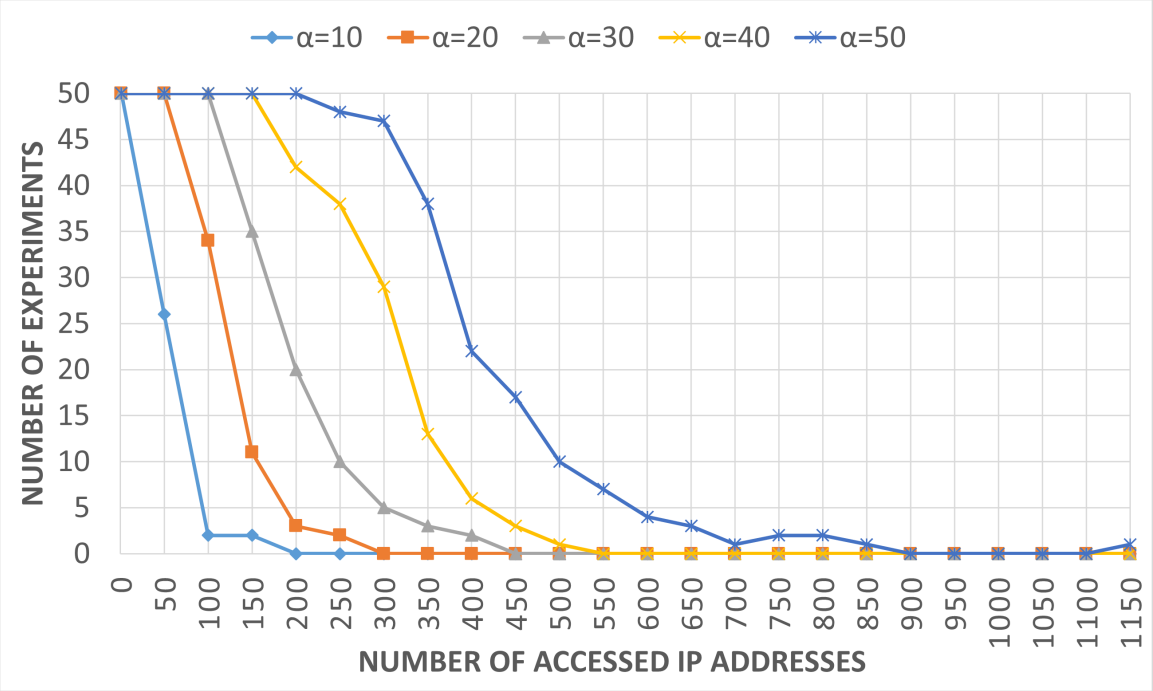

Figure 9: The number of experiments whose detection accuracy was 0% (web browsing and file downloading).

5.2.2 Firefox

Figure 13 to Figure 15 show the same results from the captured dataset analyzed using Cloudflare, Google Chrome, and Quad9 with Firefox as the browser. These graphs are steeper than in the other experiments because only one captured data item exists for each DoH recursive resolver.

Compared with actual communication experiments, the detection accuracy shows similar trends regardless of the DoH recursive resolver. In any case, α needs to be adjusted every time the number of accessed IP addresses increases by a certain amount.

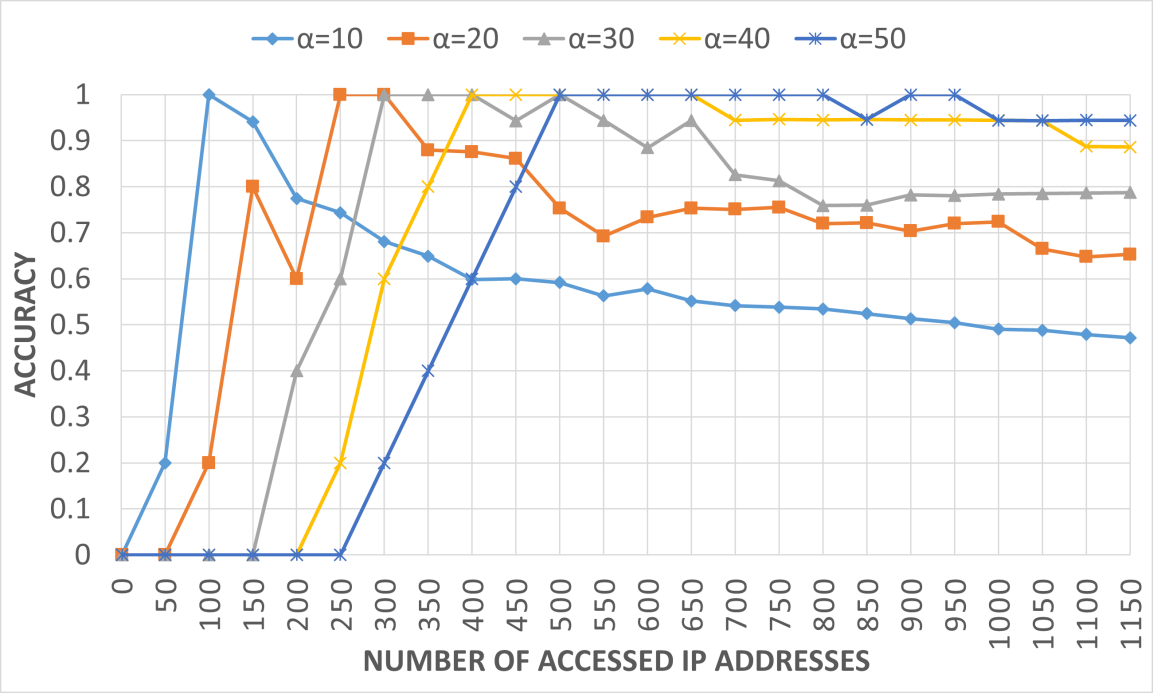

Figure 10: The average detection accuracy (Chrome-Cloudflare).

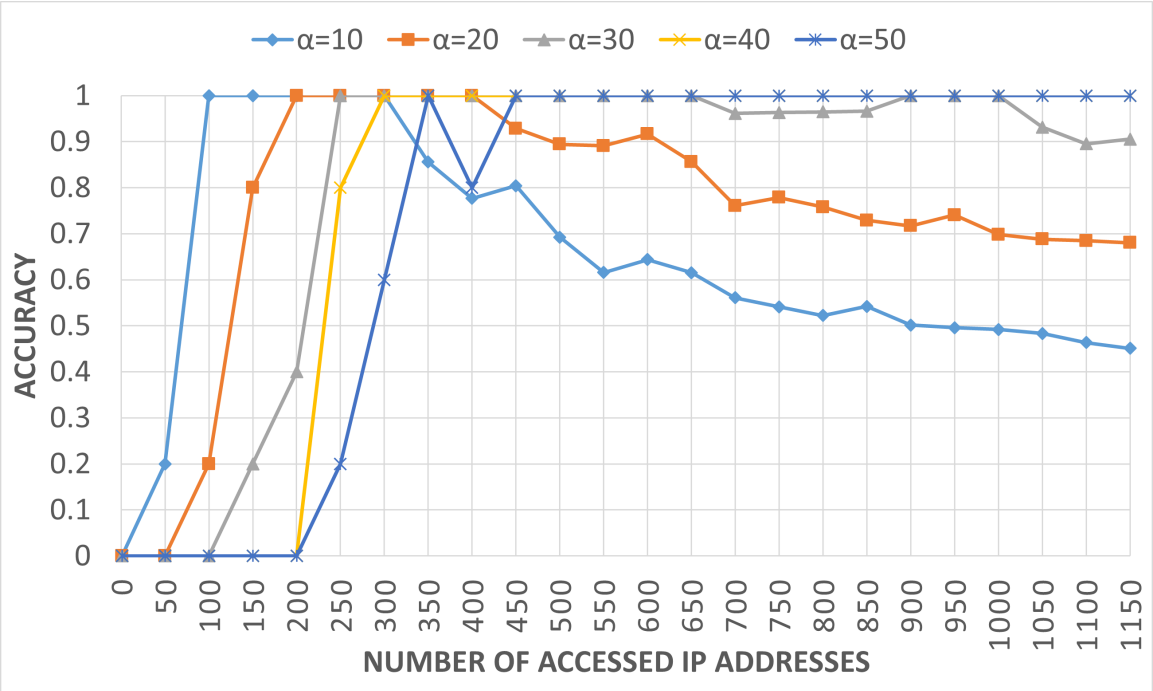

Figure 11: The average detection accuracy (Chrome-Google).

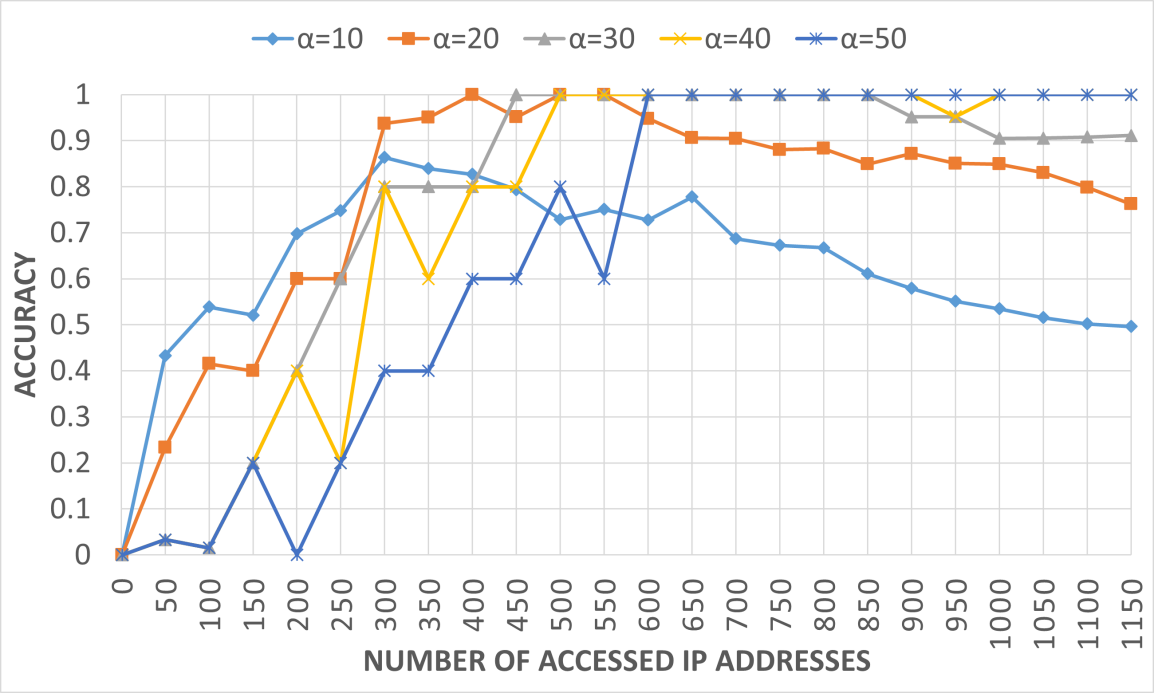

Figure 12: The average detection accuracy (Chrome-Quad9).

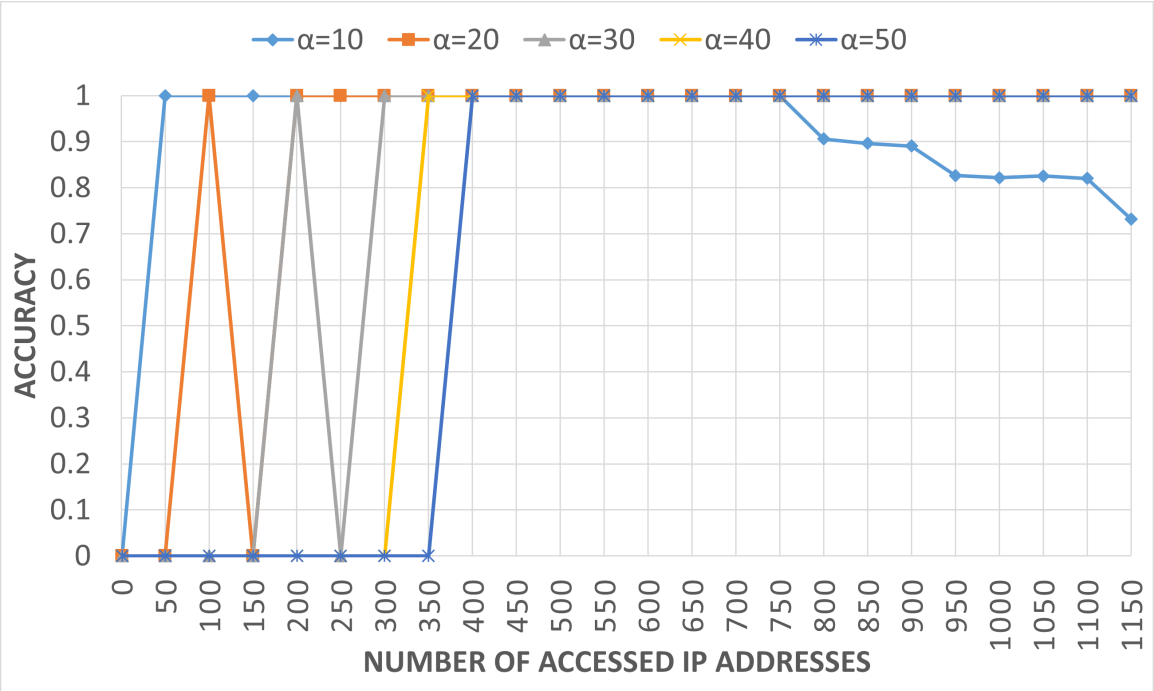

Figure 13: The average detection accuracy (Firefox-Cloudflare).

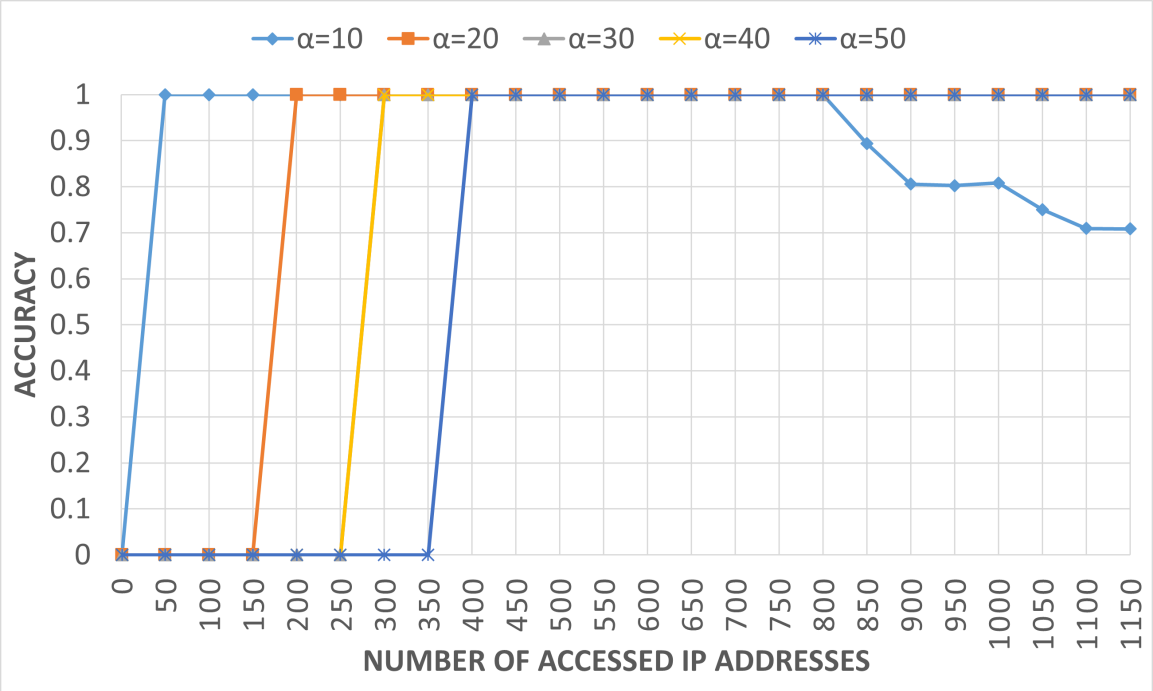

Figure 14: The average detection accuracy (Firefox-Google).

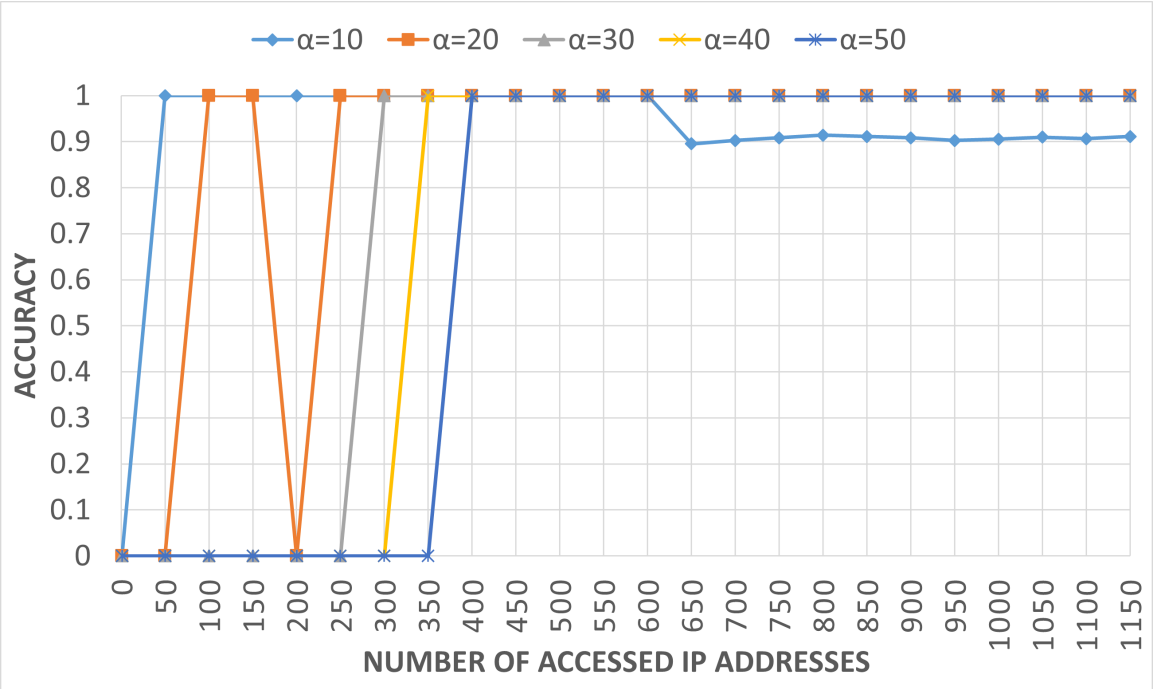

Figure 15: The average detection accuracy (Firefox-Quad9).

Table 16: Analysis results in parameter $\alpha = 1.5$ for applying the proposed method to datasets (Firefox-Cloudflare)

IP	TP	TN	FP	FN	Recall	Precision	F-score	ML_accuracy	Accuracy	CNNP	IQR_c	$Q3_c$
10	1	9	0	0	1	1	1	1	1	10	1	2
20	1	19	0	0	1	1	1	1	1	20	3	5
30	1	27	2	0	1	0.3333	0.5	0.9333	0.6444	29	2	4
40	1	38	1	0	1	0.5	0.6667	0.975	0.7826	36	2.5	4.5
50	1	48	1	0	1	0.5	0.6667	0.98	0.8113	43	3	5
60	1	58	1	0	1	0.5	0.6667	0.9833	0.8276	48	3	5
70	1	66	3	0	1	0.25	0.4	0.9571	0.6875	55	2	4
80	1	77	2	0	1	0.3333	0.5	0.975	0.75	60	3	5
90	1	86	3	0	1	0.25	0.4	0.9667	0.7113	69	2	5
100	1	97	2	0	1	0.3333	0.5	0.98	0.7653	75	3	6
150	1	148	1	0	1	0.5	0.6667	0.9933	0.8	116	6	8
200	1	187	12	0	1	0.0769	0.1429	0.94	0.4068	144	4	7
250	1	243	6	0	1	0.1429	0.25	0.976	0.5396	184	6	9
300	1	292	7	0	1	0.125	0.2222	0.9767	0.549	224	6	9
350	1	338	11	0	1	0.0833	0.1538	0.9686	0.4954	267	6	9
400	1	380	19	0	1	0.05	0.0952	0.9525	0.427	307	5	8
450	1	428	21	0	1	0.0455	0.087	0.9533	0.4261	346	5	8
500	1	476	23	0	1	0.0417	0.08	0.954	0.4204	383	5	8
550	1	526	23	0	1	0.0417	0.08	0.9582	0.4314	421	5	8
600	1	576	23	0	1	0.0417	0.08	0.9617	0.4453	456	5	8
650	1	622	27	0	1	0.0357	0.069	0.9585	0.4283	496	5	8
700	1	670	29	0	1	0.0333	0.0645	0.9586	0.4183	535	5	8
750	1	713	36	0	1	0.027	0.0526	0.952	0.3894	572	5	8
800	1	758	41	0	1	0.0238	0.0465	0.9488	0.3742	607	5	8
850	1	805	44	0	1	0.0222	0.0435	0.9482	0.3604	648	5	8
900	1	849	50	0	1	0.0196	0.0385	0.9444	0.3446	684	5	8
950	1	894	55	0	1	0.0179	0.0351	0.9421	0.3279	723	5	8
1000	1	941	58	0	1	0.0169	0.0333	0.942	0.3237	765	5	8
1050	1	981	68	0	1	0.0145	0.0286	0.9352	0.309	808	5	8
1100	1	1030	69	0	1	0.0143	0.0282	0.9373	0.3163	852	5	8
1150	1	1073	76	0	1	0.013	0.0256	0.9339	0.3041	887	5	8

Table 17: Analysis results in parameter $\alpha = 10$ for applying the proposed method to datasets (Firefox-Cloudflare)

IP	TP	TN	FP	FN	Recall	Precision	F-score	ML_accuracy	Accuracy	CNNP	IQR_c	$Q3_c$
50	1	49	0	0	1	1	1	1	1	43	3	5
100	1	99	0	0	1	1	1	1	1	75	3	6
150	1	149	0	0	1	1	1	1	1	116	6	8
200	1	199	0	0	1	1	1	1	1	144	4	7
250	1	249	0	0	1	1	1	1	1	184	6	9
300	1	299	0	0	1	1	1	1	1	224	6	9
350	1	349	0	0	1	1	1	1	1	267	6	9
400	1	399	0	0	1	1	1	1	1	307	5	8
450	1	449	0	0	1	1	1	1	1	346	5	8
500	1	499	0	0	1	1	1	1	1	383	5	8
550	1	549	0	0	1	1	1	1	1	421	5	8
600	1	599	0	0	1	1	1	1	1	456	5	8
650	1	649	0	0	1	1	1	1	1	496	5	8
700	1	699	0	0	1	1	1	1	1	535	5	8
750	1	749	0	0	1	1	1	1	1	572	5	8
800	1	798	1	0	1	0.5	0.6667	0.9988	0.906	607	5	8
850	1	848	1	0	1	0.5	0.6667	0.9988	0.8963	648	5	8
900	1	898	1	0	1	0.5	0.6667	0.9989	0.8906	684	5	8
950	1	947	2	0	1	0.3333	0.5	0.9979	0.8263	723	5	8
1000	1	997	2	0	1	0.3333	0.5	0.998	0.8217	765	5	8
1050	1	1047	2	0	1	0.3333	0.5	0.9981	0.8253	808	5	8
1100	1	1097	2	0	1	0.3333	0.5	0.9982	0.82	852	5	8
1150	1	1145	4	0	1	0.2	0.3333	0.9965	0.7318	887	5	8

Table 18: Analysis results in parameter $\alpha = 20$ for applying the proposed method to datasets (Firefox-Cloudflare)

IP	TP	TN	FP	FN	Recall	Precision	F-score	ML_accuracy	Accuracy	CNNP	IQR_c	$Q3_c$
50	0	49	0	1	0	NaN	NaN	0.98	0	43	3	5
100	1	99	0	0	1	1	1	1	1	75	3	6
150	0	149	0	1	0	NaN	NaN	0.9933	0	116	6	8
200	1	199	0	0	1	1	1	1	1	144	4	7
250	1	249	0	0	1	1	1	1	1	184	6	9
300	1	299	0	0	1	1	1	1	1	224	6	9
350	1	349	0	0	1	1	1	1	1	267	6	9
400	1	399	0	0	1	1	1	1	1	307	5	8
450	1	449	0	0	1	1	1	1	1	346	5	8
500	1	499	0	0	1	1	1	1	1	383	5	8
550	1	549	0	0	1	1	1	1	1	421	5	8
600	1	599	0	0	1	1	1	1	1	456	5	8
650	1	649	0	0	1	1	1	1	1	496	5	8
700	1	699	0	0	1	1	1	1	1	535	5	8
750	1	749	0	0	1	1	1	1	1	572	5	8
800	1	799	0	0	1	1	1	1	1	607	5	8
850	1	849	0	0	1	1	1	1	1	648	5	8
900	1	899	0	0	1	1	1	1	1	684	5	8
950	1	949	0	0	1	1	1	1	1	723	5	8
1000	1	999	0	0	1	1	1	1	1	765	5	8
1050	1	1049	0	0	1	1	1	1	1	808	5	8
1100	1	1099	0	0	1	1	1	1	1	852	5	8
1150	1	1149	0	0	1	1	1	1	1	887	5	8

Table 19: Analysis results in parameter $\alpha = 30$ for applying the proposed method to datasets (Firefox-Cloudflare)

IP	TP	TN	FP	FN	Recall	Precision	F-score	ML_accuracy	Accuracy	CNNP	IQR_c	$Q3_c$
50	0	49	0	1	0	NaN	NaN	0.98	0	43	3	5
100	0	99	0	1	0	NaN	NaN	0.99	0	75	3	6
150	0	149	0	1	0	NaN	NaN	0.9933	0	116	6	8
200	1	199	0	0	1	1	1	1	1	144	4	7
250	0	249	0	1	0	NaN	NaN	0.996	0	184	6	9
300	1	299	0	0	1	1	1	1	1	224	6	9
350	1	349	0	0	1	1	1	1	1	267	6	9
400	1	399	0	0	1	1	1	1	1	307	5	8
450	1	449	0	0	1	1	1	1	1	346	5	8
500	1	499	0	0	1	1	1	1	1	383	5	8
550	1	549	0	0	1	1	1	1	1	421	5	8
600	1	599	0	0	1	1	1	1	1	456	5	8
650	1	649	0	0	1	1	1	1	1	496	5	8
700	1	699	0	0	1	1	1	1	1	535	5	8
750	1	749	0	0	1	1	1	1	1	572	5	8
800	1	799	0	0	1	1	1	1	1	607	5	8
850	1	849	0	0	1	1	1	1	1	648	5	8
900	1	899	0	0	1	1	1	1	1	684	5	8
950	1	949	0	0	1	1	1	1	1	723	5	8
1000	1	999	0	0	1	1	1	1	1	765	5	8
1050	1	1049	0	0	1	1	1	1	1	808	5	8
1100	1	1099	0	0	1	1	1	1	1	852	5	8
1150	1	1149	0	0	1	1	1	1	1	887	5	8

Table 20: Analysis results in parameter $\alpha = 40$ for applying the proposed method to datasets (Firefox-Cloudflare)

IP	TP	TN	FP	FN	Recall	Precision	F-score	ML_accuracy	Accuracy	CNNP	IQR_c	$Q3_c$
50	0	49	0	1	0	NaN	NaN	0.98	0	43	3	5
100	0	99	0	1	0	NaN	NaN	0.99	0	75	3	6
150	0	149	0	1	0	NaN	NaN	0.9933	0	116	6	8
200	0	199	0	1	0	NaN	NaN	0.995	0	144	4	7
250	0	249	0	1	0	NaN	NaN	0.996	0	184	6	9
300	0	299	0	1	0	NaN	NaN	0.9967	0	224	6	9
350	1	349	0	0	1	1	1	1	1	267	6	9
400	1	399	0	0	1	1	1	1	1	307	5	8
450	1	449	0	0	1	1	1	1	1	346	5	8
500	1	499	0	0	1	1	1	1	1	383	5	8
550	1	549	0	0	1	1	1	1	1	421	5	8
600	1	599	0	0	1	1	1	1	1	456	5	8
650	1	649	0	0	1	1	1	1	1	496	5	8
700	1	699	0	0	1	1	1	1	1	535	5	8
750	1	749	0	0	1	1	1	1	1	572	5	8
800	1	799	0	0	1	1	1	1	1	607	5	8
850	1	849	0	0	1	1	1	1	1	648	5	8
900	1	899	0	0	1	1	1	1	1	684	5	8
950	1	949	0	0	1	1	1	1	1	723	5	8
1000	1	999	0	0	1	1	1	1	1	765	5	8
1050	1	1049	0	0	1	1	1	1	1	808	5	8
1100	1	1099	0	0	1	1	1	1	1	852	5	8
1150	1	1149	0	0	1	1	1	1	1	887	5	8

Table 21: Analysis results in parameter $\alpha = 50$ for applying the proposed method to datasets (Firefox-Cloudflare)

IP	TP	TN	FP	FN	Recall	Precision	F-score	ML_accuracy	Accuracy	CNNP	IQR_c	$Q3_c$
50	0	49	0	1	0	NaN	NaN	0.98	0	43	3	5
100	0	99	0	1	0	NaN	NaN	0.99	0	75	3	6
150	0	149	0	1	0	NaN	NaN	0.9933	0	116	6	8
200	0	199	0	1	0	NaN	NaN	0.995	0	144	4	7
250	0	249	0	1	0	NaN	NaN	0.996	0	184	6	9
300	0	299	0	1	0	NaN	NaN	0.9967	0	224	6	9
350	0	349	0	1	0	NaN	NaN	0.9971	0	267	6	9
400	1	399	0	0	1	1	1	1	1	307	5	8
450	1	449	0	0	1	1	1	1	1	346	5	8
500	1	499	0	0	1	1	1	1	1	383	5	8
550	1	549	0	0	1	1	1	1	1	421	5	8
600	1	599	0	0	1	1	1	1	1	456	5	8
650	1	649	0	0	1	1	1	1	1	496	5	8
700	1	699	0	0	1	1	1	1	1	535	5	8
750	1	749	0	0	1	1	1	1	1	572	5	8
800	1	799	0	0	1	1	1	1	1	607	5	8
850	1	849	0	0	1	1	1	1	1	648	5	8
900	1	899	0	0	1	1	1	1	1	684	5	8
950	1	949	0	0	1	1	1	1	1	723	5	8
1000	1	999	0	0	1	1	1	1	1	765	5	8
1050	1	1049	0	0	1	1	1	1	1	808	5	8
1100	1	1099	0	0	1	1	1	1	1	852	5	8
1150	1	1149	0	0	1	1	1	1	1	887	5	8

Table 16 to Table 21 show the results of the dataset analysed, which is the scenario of using Firefox as web browser and Cloudflare as DoH resolver in detail with parameter $\alpha = 1.5, 10, 20, 30, 40, 50$, the CNNP of 1.1.1.1, and the DoH destination of Cloudflare, as *CNNP* in the table items.

Overall, the experiments with the dataset show a similar trend to the experiments with actual communications, with the smaller parameter α being able to detect DoH destinations earlier, but Precision and F-score decreasing as the number of accessed IP addresses increases. On the other hand, the larger the parameter α, the earlier the DoH destination cannot be detected, but as the number of accessed IP addresses increases, the DoH destination can be detected without false positives. Based on Table 16, as in the actual communication experiment above, it is inappropriate to use parameter α as a fixed value of 1.5, which is a value commonly used in interquartile outlier detection methods.

From the above, it can be concluded that the real communication experiments and the data set experiments show similar trends, and that the environment of the real communication experiments is fair.

6 Conclusion

This paper pointed out the problems of DoH for network administrators and proposed a method for detecting the destination of DoH communication using only non-encrypted information. Actual communication experiments showed a detection accuracy of 100% when the number of accessed IP addresses exceeded 350 and the parameter α was set to 30. The experiments with previously captured datasets also showed that the proposed method is applicable regardless of browsers and DoH recursive resolvers.

For network administrators, if they detect DoH communications using this method, they may be able to restrict the use of DoH by blocking the destination. We have already executed another experiments that a DoH server are blocked at the middlebox during Firefox communicates with web servers, although we cannot write about the experiments by the upper page limit. "network.trr.mode" in Table 3 is set to 3, i.e., only DoH is used, Firefox displays error messages to stop communicating. When "network.trr.mode" is set to a default value 2, Firefox switches from DoH to normal DNS for about 3 seconds. Therefore, users will not recognize the delay.

In future, it is necessary to confirm the degree of false positives by parameter when DoH is not used. Moreover, we should discuss how to effectively change α. For example, a network manager may want to dynamically change α to suit the number of IP addresses (IP) such that $\alpha = 10$ while IP < 100, $\alpha = 20$ while IP < 200, $\alpha = 30$ while IP < 300, and so on. In addition, although the IQR method was used as the outlier detection method in this study, other outlier detection methods may be available.

References

[1] Zi Hu, Liang Zhu, John Heidemann, Allison Mankin, Duane Wessels, and Paul E. Hoffman Specification for DNS over Transport Layer Security (TLS), RFC7858, 2016.

[2] Paul E. Hoffman and Patrick McManus, DNS Queries over HTTPS (DoH), RFC8484, 2018.

[3] Yuya Takanashi and Shigetomo Kimura, Detection Method of DoH Communications from Non-Encrypted Information at a Middlebox, IEICE Technical Report, Vol. 122, No. 407, pp. 199–204, 2023. (in Japanese)

[4] Yuya Takanashi and Shigetomo Kimura, Method for Detecting DoH Communications from Non-Encrypted Information at a Middlebox, 2023 Eleventh International Symposium on Computing and Networking Workshops (CANDARW), 7 pages, 2023.

[5] Mozilla, Firefox DNS-over-HTTPS, https://support.mozilla. org/ja/kb/firefox-dns-over-https, Viewed February 2023.

[6] Kenji Baheux, A Safer and More Private Browsing Experience with Secure DNS, https://blog.chromium.org/2020/05/a-safer-and-more-private-browsing-DoH.html, Viewed December 2023.

[7] Cloudflare, Connect to 1.1.1.1 Using DoH Clients, https://developers.cloudflare.com/1.1.1.1/encryption/dns-over-https/dns-over-https-client, Viewed December 2023.

[8] Google, DNS-over-HTTPS (DoH), https://developers.google.com/speed/public-dns/docs/doh, Viewed December 2023.

[9] Quad9, DoH with Quad9 DNS Servers, https://www.quad9.net/news/blog/doh-with-quad9-dns-servers/, Viewed December 2023.

[10] Timm Böttger, Felix Cuadrado, Gianni Antichi, Eder Leão Fernandes, Gareth Tyson, Ignacio Castro, and Steve Uhlig An Empirical Study of the Cost of DNS-over-HTTPS, IMC '19: Proceedings of the Internet Measurement Conference, pp. 15–21, 2019.

[11] Jiating Wu, Yujia Zhu, Baiyang Li, Qingyun Liu, and Binxing Fang, Peek Inside the Encrypted World: Autoencoder-Based Detection of DoH Resolvers, Proceedings of 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 783–790, 2021.

[12] Carmen Kwan, Paul Janiszewski, Shela Qiu, Cathy Wang, and Cecylia Bocovich, Exploring Simple Detection Techniques for DNS-over-HTTPS Tunnels, Proceedings of the ACM SIGCOMM 2021 Workshop on Free and Open Communications on the Internet (FOCI '21), pp. 37–42, 2021.

[13] Mengqi Zhan, Yang Li, Guangxi Yu, Bo Li, and Weiping Wang, Detecting DNS over HTTPS based data exfiltration, Computer Networks, Vol. 209, No. 108919, 2022.

[14] Mohammadreza MontazeriShatoori, Logan Davidson, Gurdip Kaur, and Arash Habibi Lashkari, Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic, The 5th IEEE Cyber Science and Technology Congress, pp. 63–70, 2020.

[15] Canadian Institute for Cybersecurity, CIRA-CIC-DoHBrw-2020, https://www.unb.ca/cic/datasets/dohbrw-2020.html, Viewed August 2023.

[16] DoHlyzer, https://github.com/ahlashkari/DoHlyzer, Viewed August 2023.

[17] Mitsuhashi Rikima, Jin Yong, Iida Katsuyoshi, Shinagawa Takahiro, Takai Yoshiaki, Malicious DNS Tunnel Tool Recognition using Persistent DoH Traffic Analysis, IEEE Transactions on Network and Service Management, Vol. 20, No. 2, pp. 2086–2095, 2023.

[18] David Diez, Mine Çetinkaya-Rundel, and Christopher D Barr, OpenIntro Statistics Fourth Edition, p. 50, 2024.