Malicious DNS Tunnel Tool Recognition using Persistent DoH Traffic Analysis

Rikima Mitsuhashi, *Member, IEEE,* Yong Jin, *Member, IEEE,* Katsuyoshi Iida, *Member, IEEE,* Takahiro Shinagawa, *Member, IEEE,* and Yoshiaki Takai, *Member, IEEE*

***Abstract*—DNS over HTTPS (DoH) protocol can mitigate the risk of privacy breaches but makes it difficult to control network security services due to the DNS traffic encryption. However, since malicious DNS tunnel tools for the DoH protocol pose network security threats, network administrators need to recognize malicious communications even after the DNS traffic encryption has become widespread. In this paper, we propose a malicious DNS tunnel tool recognition system using persistent DoH traffic analysis based on machine learning. The proposed system can accomplish continuous knowledge updates for emerging malicious DNS tunnel tools on the machine learning model. The system is based on hierarchical machine learning classification and focuses on DoH traffic analysis. The evaluation results confirm that the proposed system is able to recognize the six malicious DNS tunnel tools in total, not only well-known ones, including dns2tcp, dnscat2, and iodine, but also the emerging ones such as dnstt, tcp-over-dns, and tuns with 98.02% classification accuracy.**

***Index Terms*—DNS over HTTPS (DoH), Network traffic classification, Machine learning methods, Gradient boosting decision tree algorithm, GBDT algorithm, Suspicious DoH traffic, Emerging malicious DNS tunnel tool recognition, CIRA-CICDoHBrw-2020, DoH-Tunnel-Traffic-HKD.**

I. Introduction

FOR privacy preservation and network security, there is growing momentum to encrypt DNS traffic on the Internet. A prospective approach for encrypting DNS traffic is DNS over HTTPS (DoH) using the SSL/TLS protocol, which is standardized in RFC8484 [1]. Recently, the number of software supporting DoH has been rapidly increasing. The latest versions of notable web browsers, such as Mozilla Firefox, Google Chrome, and Microsoft Edge, support DoH. As for operating systems, Windows 11, released in October 2021, has started to support encrypted DNS connections of the DoH protocol [2]. MacOS 11 and iOS 14, launched in September 2020, provide a DoH network configuration called NEDNSsettingsManager [3]. On a Linux system, DoH proxy software can convert the domain name resolution protocol between the conventional DNS and DoH, such as Cloudflare Tunnel (cloudflared) [4], DNS-over-HTTPS [5], DNS Over HTTPS Proxy [6], DNSCrypt [7], and doh-client [8]. Fig. 1 illustrates a structure of DoH-based domain name resolution. DoH carries DNS messages over HTTPS traffic between a client PC and a DoH server that works as a DNS full-service resolver. Then, the DoH server connects to authoritative DNS servers on the Internet and performs the domain name resolutions using conventional DNS protocols.

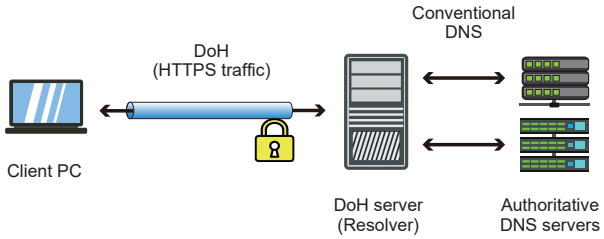

Fig. 1. Structure of DoH-based domain name resolution.

DoH protocol can mitigate the risk of privacy breaches when Internet users visit websites [9] but also makes it difficult for network administrators to control network security services due to the DNS traffic encryption. Namely, malicious DNS tunnel tools for the DoH protocol pose network security threats because network administrators cannot recognize suspicious communications by monitoring the network traffic of domain name resolutions. Once a malicious DNS tunnel connection is established between a client PC and an external DNS tunnel server, improper client users could pass through the firewall and access the Internet freely. If attackers exploit the DNS tunnel, a client PC would be under their control. To solve this problem, network administrators need to recognize malicious communications generated by the malicious DNS tunnel tools to maintain network security even after the DNS traffic encryption has become widespread. The recognition of malicious DNS tunnel tools enables network administrators to 1) block malicious communications, 2) alert the clients about the risky situation with warning messages, and 3) encourage the organizations to improve their security policies.

In this paper, we propose a malicious DNS tunnel tool recognition system using persistent DoH traffic analysis based on machine learning. Fig. 2 shows the concept of the proposed system using persistent DoH traffic analysis. The proposed system can accomplish continuous knowledge updates for emerging malicious DNS tunnel tools on the machine learning

R. Mitsuhashi and T. Shinagawa are with the University of Tokyo, Japan. e-mail: mitsuhashi@os.ecc.u-tokyo.ac.jp, shina@ecc.u-tokyo.ac.jp.

R. Mitsuhashi, K. Iida and Y. Takai are with Hokkaido University, Japan. e-mail: iida@iic.hokudai.ac.jp, ytakai@iic.hokudai.ac.jp.

Y. Jin is with Tokyo Institute of Technology, Japan. e-mail: yongj@gsic.titech.ac.jp.

Manuscript received March 31, 2022; revised June 14, 2022; accepted October 4, 2022. *(Corresponding author: Rikima Mitsuhashi.)*

Digital Object Identifier xxxx

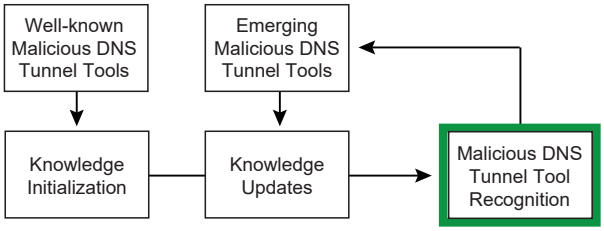

Fig. 2. Concept of proposed system using persistent DoH traffic analysis.

model. Furthermore, supervised training in machine learning provides stable operation with high classification accuracy for malicious DNS tunnel tool recognition. The evaluation results confirm that the proposed system is able to recognize the six malicious DNS tunnel tools in total, not only well-known ones, including dns2tcp, dnscat2, and iodine, but also emerging ones such as dnstt, tcp-over-dns, and tuns with 98.02% classification accuracy. The results indicate that the proposed system is practical and beneficial for network security management.

Our contributions are as follows.

- We propose a malicious DNS tunnel tool recognition system using persistent DoH traffic analysis based on machine learning.
- The proposed system mitigates the threats of emerging malicious DNS tunnel tools on the DoH traffic.
- The evaluation results confirm that the proposed system is able to recognize the six malicious DNS tunnel tools with high classification accuracy.

This article is an extended version of our previous publication [10]. In the previous paper, the proposed system was able to identify the well-known malicious DNS tunnel tools but could not recognize the newly emerged ones. Differing from the previous version, the implementation discussed in this paper provides persistent DoH traffic analysis, which allows for knowledge updates for emerging malicious DNS tunnel tools and stable operation with high classification accuracy. The knowledge updates are strongly required because there are many malicious DNS tunnel tools, and new malicious DNS tunnel tools may appear in the future.

The rest of this paper is organized as follows. Section II presents the related work on malicious DNS tunnel detection and DoH traffic classification. Section III describes the design of the proposed system for persistent DoH traffic analysis. Section IV shows the evaluations of the proposed system implementation. Finally, Section V concludes the paper.

II. Related Work

A. Malicious DNS Tunnel Detection

DNS name resolution is one of the most common network services, and many studies have reported attack countermeasures [11], [12], [13], [14]. Since the DNS protocol is less likely to be blocked by firewalls on the Internet, it is an attractive channel for attackers who want to communicate covertly across network boundaries. As a result, various DNS tunnel tools exist [15].

Detecting DNS tunnel traffic hidden behind name resolution traffic has been widely studied to eliminate malicious communications. P. Yang et al. [16] attempted to detect covert channels in the DNS protocol. Their detection method used a stacking model, which is an ensemble of three different machine learning algorithms: K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest. The DNS tunnel traffic used in their experiments was generated by a set of tunnel tools, including dns2tcp, dnscat2, DeNiSe, and Heyoka. A. L. Buczak et al. [17] used a random forest classifier to distinguish between normal DNS activity and DNS tunneling activity. The traffic data used in their experiment were the results of a penetration test using iodine, dnscat2, Cobalt Strike, and Pick Pocket. D. Lambion et al. [18] evaluated the performance of Convolutional Neural Network (CNN), Random Forest, and ensemble classifiers against one-day real-traffic data for detecting malicious DNS tunnel activities of iodine. Y. Chen et al. [19] suggested a framework for DNS tunnel detection using Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and CNNs. The experiments used eight DNS tunnel tools, including iodine, dnscat2, dns2tcp, DNShell, OzymanDNS, Cobalt Strike, DNSExfiltrator, and DET. A. Chowdhary et al. [20] combined two methods for detecting DNS tunneling queries generated by tunnel tools, such as dns2tcp, iodine, tuns, and DNScapy. One method utilizes cache misses in DNS full-service resolvers, and the other uses machine learning technology to classify DNS queries. C. Kwan et al. [21] selected dnstt [22] and investigated the impact of threshold-based attacks on tunnel tools. They implemented a defense mechanism for dnstt against threshold attacks and evaluated its performance. K. Wu et al. [23] introduced a three-stage DNS tunnel detection method called FTPB. The FTPB used character feature extraction to filter out the domain names resolved by DNS tunnels such as dns2tcp, dnscat2, iodine, DNScapy, OzymanDNS, Weasel, and PacketWhisper.

B. DoH Traffic Classification

The DoH protocol has been supported on various software and services recently, as described in Section I, and a lot of studies regarding DoH network classification have been reported. D. Vekshin et al. [24] extracted DoH traffic from HTTPS traffic by using machine learning. They also identified the application programs that generate the DoH traffic, such as Chrome, Firefox, and Cloudflare Tunnel. The DoH extraction and application identification resulted in a classification accuracy of 99.9% using the Adaboost model. The dataset used for their experiments consisted of the domain names of the top 1 million sites served by Alexa [25]. L. Csikor et al. [26] used machine learning to classify web traffic and DoH, achieving a classification accuracy of 97.4% in a closed environment and 90.0% in an open environment. They used six types of machine learning models for classification: SVM, Decision Tree, Random Forest, Adaboost, Naive Bayes, and Logistic Regression. The dataset used in the experiment included thousands of domains in Alexa's list of top-ranked websites.

Recently, there has been a sharp increase in reports of DoH tunnel detection. The following reports use the CIRA-CIC-

DoHBrw-2020 dataset [27] for their experimental evaluation. M. MontazeriShatoori et al. [28] used Random Forest to filter DoH traffic from HTTPS traffic and to detect malicious DoH traffic from DoH traffic. In their experiments, the former filtering obtained an F-score of 99.3%, and the latter detection resulted in an F-score of 99.9%. Y. Khodjaeva et al. [29] used statistical features extracted from traffic flows by several tools, such as Argus, DoHlyzer, and Tranalyzer2. They obtained an F-score of 93.5% by Random Forest in classifying DoH traffic as normal and malicious.

S. K. Singh et al. [30] improved the accuracy of classifying benign and malicious DoH traffic. Using the hold-out, they achieved 100% classification accuracy by Gradient Boost. R. Alenezi and S. A. Ludwig [31] identified the name of malicious DNS tunnel tools from DoH traffic. The classification accuracy was 99.2% achieved by XGBoost using the hold-out.

In a report using cross-validation, M. Behnke et al. [32] filtered DoH traffic from HTTPS traffic and then classified that DoH traffic into benign and malicious DoH traffic. They obtained an accuracy of 99.7% by Random Forest in the first filtering and an accuracy of 99.9% by LightGBM in the second classification. L. F. Gonzalez Casanova and P.-C. Lin [33] classified DoH traffic with deep learning models. They used the BiLSTM model to filter DoH traffic from HTTPS traffic with 99.0% accuracy and detect malicious DoH traffic from DoH traffic with 99.4% accuracy.

In summary, no studies have addressed the recognition of emerging malicious DNS tunnel tools using DoH traffic analysis between malicious DNS tunnel detection and DoH traffic classification. The proposed system can accomplish continuous knowledge updates for emerging malicious DNS tunnel tools to support network administrators.

III. Design

In Section II, we explained the related research on malicious DNS tunnel detection and investigated some of the methods of DoH traffic classification and unknown network traffic recognition. To keep network security, network administrators need to recognize any malicious DNS tunnel tools on the DoH traffic. In this section, we describe the design of the proposed system.

A. System Overview

Recognizing malicious DNS tunnel tools on the DoH traffic that exploit DoH-based domain name resolution requires a detailed network traffic analysis. The proposed system is based on hierarchical machine learning classification and focuses on DoH traffic analysis. For persistent DoH traffic analysis, the system is equipped with a knowledge updating mechanism to recognize emerging malicious DNS tunnel tools. As shown in Fig. 3, the proposed system consists of five blocks: 1) traffic capturing, 2) feature extraction, 3) model decision and knowledge initialization, 4) knowledge update, and 5) DoH traffic classification. In the following subsections, we describe each block.

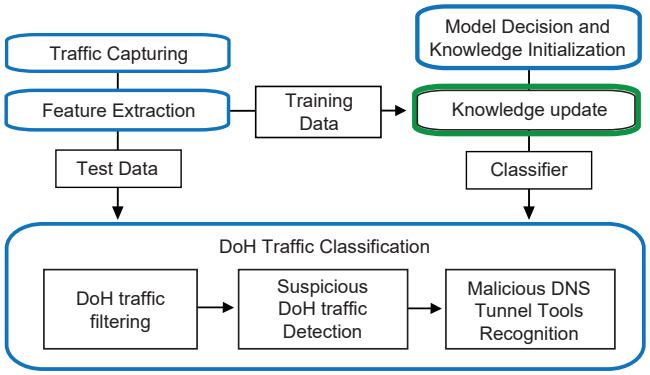

Fig. 3. Overview of malicious DNS tunnel tool recognition system.

B. Traffic Capturing

The proposed system captures the HTTPS traffic as input because the DoH protocol uses the SSL/TLS protocol for encrypting DNS traffic. Although there are likely to be various types of traffic on a network, it is possible to determine whether the packet is the HTTPS traffic from the source port number and destination port number. The port number is 443 and is included in the TCP header without encryption.

Fig. 4 shows the network connections where the CIRA-CIC-DoHBrw-2020 dataset was captured. The client PC connecting to the web servers is to retrieve web content on the Internet (Non-DoH). The client PC also connects to the normal DNS server through the DoH server for the domain name resolution (Normal DoH). Malicious DNS tunnel tools, including dns2tcp [34], dnscat2 [35], and iodine [36], are connected to the suspicious DNS server to receive further attack instructions or send out sensitive information (well-known malicious DNS tunnel tools).

Moreover, we assume a situation where other malicious DNS tunnel tools, including dnstt [22], tcp-over-dns [37], and tuns [38], [39], are newly emerging and connected to the suspicious DNS server (emerging malicious DNS tunnel tools). Fig. 5 shows details of network connections for emerging malicious DNS tunnel tools, which is a part of the connections of Fig. 4. The dnstt directly connects to the DoH server because it supports the DoH protocol. On the other hand, the tcp-over-dns and tuns use the DoH proxy to convert protocols from conventional DNS to DoH. We publish the traffic data of emerging malicious DNS tunnel tools as the DoH-Tunnel-Traffic-HKD dataset [40].

C. Feature Extraction

We extract traffic features from captured HTTPS traffic, such as packet number, packet length, packet direction (e.g., end client to DoH server and vice versa), interval time between packets, etc. However, the packet payload is excluded because it is encrypted. To automate the feature extraction, we can use DoHlyzer [41]. It extracts 34 traffic features from HTTPS traffic, as shown in Table I, and we selected 28 statistical features for evaluation. We use the DoHlyzer only for the feature extraction of emerging malicious DNS tunnel tools

TABLE I
LIST OF TRAFFIC FEATURES EXTRACTED BY DOHLYZER
(FEATURES SHADED IN GRAY ARE NOT STATISTICAL FEATURES).

1	Number of Flow Bytes Sent	18	Coefficient of Variation of Packet Time
2	Rate of Flow Bytes Sent	19	Skew from Median Packet Time
3	Number of Flow Bytes Received	20	Skew from Mode Packet Time
4	Rate of Flow Bytes Received	21	Mean Request/response Time Difference
5	Mean Packet Length	22	Median Request/response Time Difference
6	Median Packet Length	23	Mode Request/response Time Difference
7	Mode Packet Length	24	Variance of Request/response Time Difference
8	Variance of Packet Length	25	Standard Deviation of Request/response Time Difference
9	Standard Deviation of Packet Length	26	Coefficient of Variation of Request/response Time Difference
10	Coefficient of Variation of Packet Length	27	Skew from Median Request/response Time Difference
11	Skew from Median Packet Length	28	Skew from Mode Request/response Time Difference
12	Skew from Mode Packet Length	29	Duration
13	Mean Packet Time	30	Source IP address
14	Median Packet Time	31	Destination IP address
15	Mode Packet Time	32	Source Port number
16	Variance of Packet Time	33	Destination Port number
17	Standard Deviation of Packet Time	34	Time Stamp

since the CIRA-CIC-DoHBrw-2020 dataset already contains traffic features extracted by the DoHlyzer. The traffic features we extracted are published as the DoH-Tunnel-Traffic-HKD dataset [40].

D. Model decision and knowledge initialization

This subsection explains how to decide on and initialize machine learning models to be used as a classifier of DoH traffic classification. The proposed system uses the XGBoost [42], LightGBM [43], and CatBoost [44] libraries that include the gradient boosting decision tree (GBDT) algorithm. According to S. R et al. [45], these GBDT libraries are more flexible and require much shorter training times compared with other current machine learning algorithms. The above libraries are widely used in competitive machine learning contests, such as Kaggle [46], because of their high classification accuracy.

The GBDT uses boosting, which is an ensemble technique that produces a strong classifier from a large number of weak

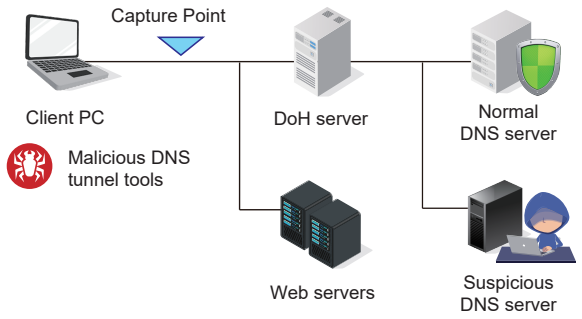

Fig. 4. Connections where CIRA-CIC-DoHBrw-2020 dataset was captured.

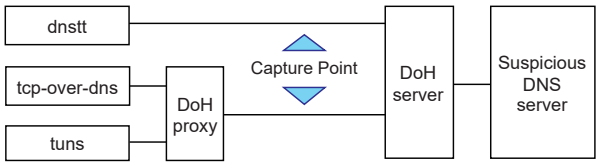

Fig. 5. Detailed connections for emerging malicious DNS tunnel tools.

TABLE II
PARAMETERS FOR GRID SEARCH
(UNDERLINE INDICATES DEFAULT PARAMETERS).

XGBoost	LightGBM	CatBoost
max_depth: 2, 4, 6, 8, 10, 12	num_leaves: 7 ,15, 31, 63, 127, 255, 511	max_depth: 2, 4, 6, 8, 10, 12 14, 16
max_bin: 128, 256, 512, 1024, 2048, 4096, 8192, 16384	max_bin: 127, 255, 511, 1023, 2047, 4095, 8191, 16383	l2_leaf_reg: 1, 3, 5, 7

classifiers. Briefly, GBDT works as follows. First, a simple tree is constructed to predict the average of the target labels. Next, the tree's residual error (observed - predicted) is calculated. Then, the next decision tree is added to reduce the residual error. If there is still a significant error remaining, another decision tree is added to decrease the error. By repeating this procedure, GBDT produces a strong classifier.

In addition to selecting high-performing models, parameter tuning is required to make the classifier suitable for network traffic classification. The parameters we use are shown in Table II. These parameters are explained in the documents of each model [47], [48], [49] to be effective in improving classification accuracy. To explore the optimal combination of parameters, we use a grid search. The number of models to be explored is 48 for XGBoost, 56 for LightGBM, and 32 for CatBoost, totaling 136.

Fig. 6 shows the model decision and knowledge initialization process. First, we prepare parameter-tuned models and train them on the sub-training data, which is 90% of the training data. Next, we validate the trained models on the validation data, which is 10% of the training data, i.e., the remainder of the sub-training data. Then, we decide on the best model among 136 parameter-tuned models by comparing the validation accuracy. Finally, we train the best model on the training data to use it as an initial classifier. The process is done for the three stages in DoH Traffic classification, resulting in the three classifiers.

As for the parameter tuning, we need to be careful not to overfit. Overfitting is an issue wherein a machine learning model closely related to a particular dataset cannot classify additional data accurately. If the model is overfitted to the validation data by parameter tuning, it cannot classify the test data with sufficient accuracy. Thus, the performance of parameter-tuned models should be evaluated using the classification accuracy of the validation data and the test data.

E. Knowledge update

In our previous work [10], the implementation only recognized well-known malicious DNS tunnel tools, including dns2tcp, dnscat2, and iodine. However, there are a number of malicious DNS tunnel tools, such as ozymanDNS, DeNiSe, Heyoka, DNScapy, and many more. Unfortunately, new malicious DNS tunnel tools may appear in the future. Therefore, the implementation of knowledge updates is essential for persistent DoH traffic analysis.

The knowledge update process is shown in Fig. 7. The knowledge update is accomplished by retraining the initial classifier using four types of HTTPS traffic as training data: non-DoH, normal-DoH, well-known malicious DNS tunnel tools, and emerging malicious DNS tunnel tools. This process is done for each stage to create updated three classifiers.

F. DoH Traffic Classification

The proposed system is based on hierarchical machine learning classification to recognize malicious DNS tunnel tools on DoH traffic. Fig. 8 illustrates the hierarchical machine learning classification process. When the test data is input, the 1st stage classifier filters the DoH traffic from the HTTPS traffic, and the 2nd stage classifier detects the suspicious DoH traffic from the entire DoH traffic. Then, the 3rd stage classifier recognizes the name of the malicious DNS tunnel tools from the suspicious DoH traffic. The knowledge update described in Section III-E enables a recognition of the emerging malicious DNS tunnel tools in the 3rd stage.

IV. Evaluation

In Section III, we illustrated the overall picture of the proposed system and described how to update the knowledge for emerging malicious DNS tunnel tools. In this section, we evaluate the performance of the proposed system implementation.

A. Implementation

In this evaluation, we used two machines for the hardware environment: one with an Intel Xeon Silver 4210R CPU, 96-GiB memory, and Nvidia GeForce RTX 3090 GPU, and the other with an Intel Xeon W-2223 CPU, 64-GiB memory, and Nvidia GeForce RTX 3090 GPU. The operating system was Ubuntu 20.04 LTS. The machine learning libraries were XGBoost 1.4.2, LightGBM 3.3.0, and CatBoost 1.0.0. XGBoost and CatBoost were run on the GPU, while LightGBM was run on the CPU. The malicious DNS tunnel tools that we ran were dnstt v1.20210812.0, tcp-over-dns 1.3, and tuns 0.9.3. To convert the conventional DNS into DoH for the tcp-over-dns and tuns, we used doh-client contained in DNS-over-HTTPS [5] as a DoH proxy. In addition, the DoH server to which malicious DNS tunnel tools connect consists of NGINX [50] and dohhttps-proxy in DNS Over HTTPS Proxy [6].

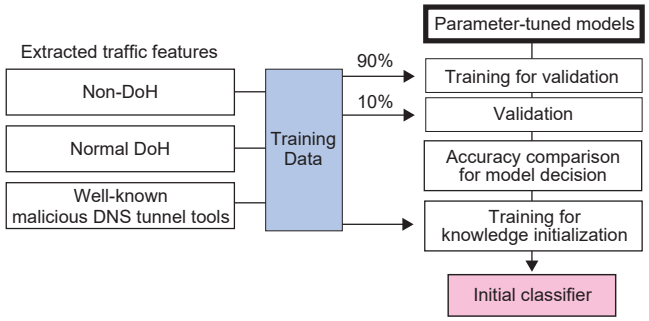

Fig. 6. Model decision and knowledge initialization process.

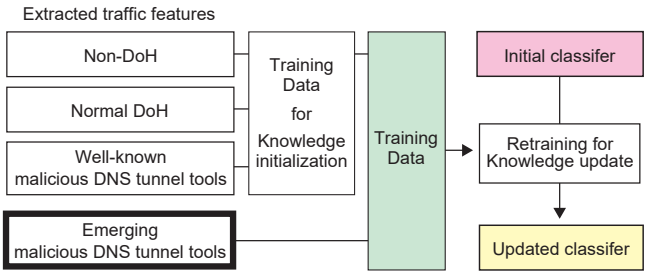

Fig. 7. Knowledge update process.

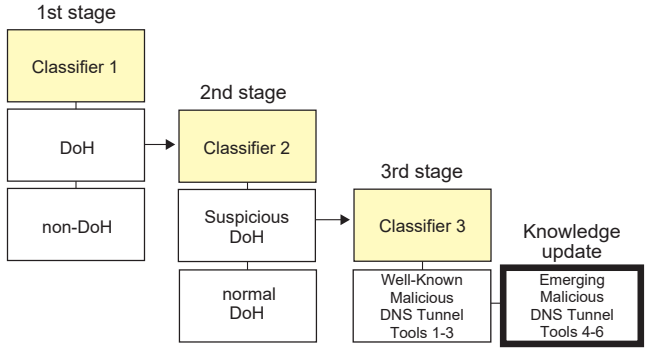

Fig. 8. Hierarchical machine learning classification process.

B. Dataset

We used two evaluation datasets: the CIRA-CIC-DoHBrw-2020 [27] and the DoH-Tunnel-Traffic-HKD [40]. Table III shows the labels and number of traffic flows obtained from the CIRA-CIC-DoHBrw-2020 dataset. Table IV describes the labels and number of traffic flows obtained from the CIRA-CIC-DoHBrw-2020 and the DoH-Tunnel-Traffic-HKD combined dataset. The purpose of selecting the CIRA-CIC-DoHBrw-2020 dataset for the evaluation is to compare the classification accuracy with previous studies. The traffic flows of emerging malicious DNS tunnel tools in Table IV, detailed in Section III-C, were captured by running each tunnel tool for 48 hours and then augmented by assuming that 20 client PCs would be using the tunnel tools.

TABLE III
LABELS AND TRAFFIC FLOWS FROM CIRA-CIC-DOHBRW-2020 DATASET.

	Labels	Traffic flows
1st stage	Non-DoH	897493
	DoH	269643
2nd stage	Normal DoH	19807
	Suspicious DoH	249836
3rd stage (Well-Known malicious DNS tunnel tools)	dns2tcp	167486
	dnscat2	35770
	iodine	46580

TABLE IV
LABELS AND TRAFFIC FLOWS FROM CIRA-CIC-DOHBRW-2020 AND DOH-TUNNEL-TRAFFIC-HKD COMBINED DATASET.

	Labels	Traffic flows
1st stage	Non-DoH	897493
	DoH	374803
2nd stage	Normal DoH	19807
	Suspicious DoH	354996
3rd stage (Well-Known malicious DNS tunnel tools)	dns2tcp	167486
	dnscat2	35770
	iodine	46580
(Emerging malicious DNS tunnel tools)	dnstt	46080
	tcp-over-dns	30040
	tuns	29040

C. Classification Metrics

To evaluate the classification results, we used stratified 10-fold cross-validation, splitting the training and test data into a 9:1 ratio. We also used accuracy, recall, precision, and F-score metrics to measure classification accuracy. The formulas for each of these metrics are as follows. Note that, for multi-class classification in the 3rd stage, we used the macro-average of each metric.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$$

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

$$F\text{-}score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

where TP means true positive, FP means false positive, FN means false negative, and TN means true negative.

D. Model decision

We decided on machine learning models through the process in Section III-D. As a result of the grid search on the training data of the CIRA-CIC-DoHBrw-2020 dataset, the parameters with the highest classification accuracy are shown in Table V. By comparing the accuracy at each stage, XGBoost is selected for DoH traffic filtering in the 1st stage, CatBoost is selected for suspicious DoH traffic detection in the 2nd stage, and LightGBM is selected for malicious DNS tunnel tool recognition in the 3rd stage. Since all the parameters were larger than the minimum value and smaller than the maximum value in the search range shown in Table II, we concluded that there was no need to extend the grid search range any further. As for combinations of max_depth: 16 and L2_leaf_reg: [1,3,5,7] in the 3rd stage, CatBoost used the CPU because of a shortage of GPU memory.

TABLE V
GRID SEARCH RESULTS.

		1st stage	2nd stage	3rd stage
XGBoost	accuracy	0.99816	0.99998	0.97191
	max_depth	10	4	6
	max_bin	1024	128	8192
LightGBM	accuracy	0.99815	0.99997	0.97213
	num_leaves	255	15	63
	max_bin	511	127	8191
CatBoost	accuracy	0.99795	0.99999	0.96885
	max_depth	14	4	10
	L2_leaf_reg	5	1	3

TABLE VI
WELL-KNOWN MALICIOUS DNS TUNNEL TOOL RECOGNITION (DNS2TCP, DNSCAT2, AND IODINE).

	Classifier	Accuracy	Precision	Recall	F-score
1st stage	XGBoost	0.998172	0.998112	0.999513	0.998812
	LightGBM	0.998154	0.998088	0.999514	0.998800
	CatBoost	0.997939	0.997834	0.999489	0.998661
2nd stage	CatBoost	0.999988	1.0	0.999987	0.999993
	XGBoost	0.999974	0.999987	0.999983	0.999985
	LightGBM	0.999974	0.999987	0.999983	0.999985
3rd stage	LightGBM	0.972069	0.949394	0.954026	0.951668
	XGBoost	0.971701	0.949106	0.953340	0.951188
	CatBoost	0.968871	0.944116	0.948957	0.946503

E. Well-known malicious DNS tunnel tool recognition

We trained models decided in Section IV-D on the training data of the CIRA-CIC-DoHBrw-2020 dataset to create initial classifiers. The result of well-known malicious DNS tunnel tool recognition by the initial classifiers on the test data of the CIRA-CIC-DoHBrw-2020 dataset is shown in Table VI. The malicious DNS tunnel tool recognition in the 3rd stage had an accuracy of 97.20% and an F-score of 95.16%. Then, the filtering of DoH traffic in the 1st stage had an accuracy of 99.81% and an F-score of 99.88%. The detection of DoH traffic in the 2nd stage had an accuracy of 99.99% and an F-score of 99.99%. The accuracy results with initial classifiers were better than that with the other final candidate models, suggesting no overfitting problem due to parameter tuning.

Table VII and Table VIII compare the performances of the proposed system and those of the previous studies [28],

TABLE VII
COMPARISON WITH PREVIOUS STUDIES ON CROSS-VALIDATION.

	Classifiers	Accuracy	Precision	Recall	F-score
1st stage	BiLSTM [33]	0.990	N/A	N/A	N/A
	RF [32]	0.997	N/A	N/A	N/A
	XGBoost (**ours**)	**0.998**	0.998	0.999	0.998
2nd stage	BiLSTM [33]	0.994	N/A	N/A	N/A
	LightGBM [32]	0.99996	N/A	N/A	N/A
	CatBoost (**ours**)	**0.99998**	1.0	0.9999	0.9999
3rd stage	LightGBM (**ours**)	**0.972**	0.949	0.954	0.951

TABLE VIII
COMPARISON WITH PREVIOUS STUDIES ON HOLD-OUT.

	Classifiers	Accuracy	Precision	Recall	F-score
1st stage	RF [28]	N/A	0.993	0.993	0.993
	XGBoost (**ours**)	0.998	**0.998**	**0.999**	**0.998**
2nd stage	RF [29]	N/A	0.941	0.935	0.935
	RF [28]	N/A	0.999	0.999	0.999
	GB [30]	N/A	1.0	1.0	1.0
	CatBoost (**ours**)	1.0	**1.0**	**1.0**	**1.0**
3rd stage	LightGBM (**ours**)	0.973	**0.951**	**0.955**	**0.953**
	XGBoost [31]	0.992	0.99	1.0	0.99

[29], [30], [31], [32], [33]. These tables show the results of cross-validation and hold-out, respectively. For comparison on hold-out, we chose the best results in the ten tests of cross-validation.

Our proposed system obtained the highest accuracy on the cross-validation for all stages in Table VII. As far as we know, no study has evaluated the 3rd stage in cross-validation. In Table VIII, our proposed system obtained the highest precision, recall, and F-score in the 1st and 2nd, while those in the 3rd stage were the second best. However, the cross-validation results are more important than the hold-out because the accuracy, precision, recall, and F-score obtained on hold-out may become much lower depending on how the training and test data are split.

F. Suspicious DoH traffic detection before knowledge update

Even before knowledge updates, We hypothesized that the proposed system could detect the traffic flows generated by emerging malicious DNS tunnel tools in the 2nd stage as suspicious DoH. To test the hypothesis, we attempted to detect the traffic flows generated by emerging malicious DNS tunnel tools by the initial classifiers created in Section IV-E. For the evaluation, we used the test data of the CIRA-CIC-DoHBrw-2020 and the DoH-Tunnel combined dataset. Table IX shows the classification results in the 1st and 2nd stages. While the 1st stage achieved a high classification accuracy of over 97%, the accuracy in the 2nd dropped to about 71%. To understand the causes of deterioration, we examined a confusion matrix in the ten tests of cross-validation. Table X showed that normal DoH traffic flows were accurately classified, but many suspicious DoH traffic flows were misclassified as normal DoH traffic flows. The breakdown of misclassification was

TABLE IX
SUSPICIOUS DOH TRAFIC DETECTION BEFORE KNOWLEDGE UPDATE.

	Classifier	Accuracy	Precision	Recall	F-score
1st stage	XGBoost	0.978200	0.970244	0.999758	0.984780
2nd stage	CatBoost	0.719692	1.0	0.704053	0.826326

TABLE X
CONFUSION MATRIX IN 2ND STAGE AND SUSPICIOUS DOH BREAKDOWN.

		Normal DoH	Suspicious DoH
Normal DoH		1980	0
Suspicious DoH		10502	24998
Suspicious DoH Breakdown	dns2tcp	0	16749
	dnscat2	0	3577
	iodine	0	4658
	dnstt	4608	0
	tcp-over-dns	2990	14
	tuns	2904	0

all emerging malicious DNS tunnel tools, including dntt, tcp-over-dns, and tuns. The result indicates that the hypothesis is incorrect, and knowledge update is essential for persistent DoH traffic analysis.

G. Knowledge update and malicious DoH traffic recognition

Through the process in Section III-E, we updated the initial classifiers on the training data of the CIRA-CIC-DoHBrw-2020 and DoH-Tunnel combined dataset to make updated classifiers. Table XI shows the result of malicious DNS tunnel tool recognition by the updated classifiers on the test data of the CIRA-CIC-DoHBrw-2020 and DoH-Tunnel combined dataset. The 3rd stage recognized the six malicious DNS tunnels, including dns2tcp, dnscat2, iodine, dnstt, tcp-over-dns, and tuns, with an accuracy of 98.02% and an F-score of 97.57%. Then, the 1st stage filtered the DoH traffic with an accuracy of 99.83% and an F-score of 99.87%. The 2nd stage detected the DoH traffic with an accuracy of 99.99% and an F-score of 99.99%.

Table XII compares the performance before and after knowledge updates on the cross-validation. Before knowledge updates, as described in Section IV-E, our proposed system obtained the highest accuracy in all stages compared to previous studies. After knowledge updates, our proposed system maintained high classification accuracy in all stages.

H. Consideration of important features

To analyze the background of the classification accuracy, we show important features used by the three-stage classifiers in Table XIII. For filtering DoH traffic in the 1st stage, XGBoost considered the "Variance of Packet Time" as the most important feature. It means a spread of the packet transmission time in a traffic flow. The average "Variance of Packet Time" in the 1st stage is 76.10 for the non-DoH traffic and 670.58 for the DoH traffic. The difference indicates that the "Variance of Packet Time" of the DoH server is more widespread than that of the web server due to the time of the DoH server to query for unknown domains.

TABLE XI
RESULT OF MALICIOUS DNS TUNNEL TOOL RECOGNITION (DNS2TCP, DNSCAT2, IODINE, DNSTT, TCP-OVER-DNS, AND TUNS).

	Classifier	Accuracy	Precision	Recall	F-score
1st stage	XGBoost	0.998301	0.998123	0.999470	0.998796
2nd stage	Catboost	0.999959	0.999977	0.999980	0.999978
3rd stage	LightGBM	0.980225	0.974732	0.976765	0.975732

TABLE XII
COMPARISON OF BEFORE AND AFTER UPDATES ON CROSS-VALIDATION.

	Update	Accuracy	Precision	Recall	F-score
1st stage	before	0.998	0.998	0.999	0.998
	after	**0.998**	0.998	0.999	0.998
2nd stage	before	0.999	1.0	0.999	0.999
	after	**0.999**	0.999	0.999	0.999
3rd stage	before	0.972	0.949	0.954	0.951
	after	**0.980**	0.974	0.976	0.975

Regarding detecting suspicious DoH traffic in the 2nd stage, CatBoost considered the "Mode Packet Length" as the most important feature. It is the value that appears most often in a traffic flow. The average "Mode Packet Length" in the 2nd stage is 74.05 for the normal DoH traffic and 67.13 for the suspicious DoH traffic. The difference is due to the fact that normal DoH traffic contains a lot of SSL/TLS key exchange data between the client and the DoH server. In contrast, suspicious DoH traffic has less of that data because most malicious DNS tunnel tools keep the connection with the DoH server for a long time.

With respect to recognizing DoH tunnel tools in the 3rd stage, LightGBM considered the "Median Request/response time difference" as the most important feature. It means a middle point of the inter-arrival time of the received packets in a traffic flow. The average "Median Request/response time difference" in the 3rd stage is 0.17s for dns2tcp, 2.74s for dnscat2, 1.97s for iodine, 1.26s for dnstt, 1.02s for tcp-over-dns, and 0.86s for tuns. The difference is highly dependent on the time interval of sending queries from tunnel tools to DoH servers so that the fluctuation in the server performance or network latency up to several tens of milliseconds has little effect on it.

I. Discussion

The evaluation demonstrated that the proposed system was able to recognize malicious DNS tunnel tools using persistent DoH traffic analysis. By knowledge updates, the proposed system recognized the well-known and emerging malicious DNS tunnel tools with 98.02% classification accuracy. The results indicate that the proposed system is practical and beneficial for network security management.

As noted in Section III-E, there are a number of malicious DNS tunnel tools, and new malicious DNS tunnel tools may appear in the future. Due to knowledge updates, we believe the proposed system is effective for the latest varieties. We have a plan to conduct specific experiments in future work.

TABLE XIII
MOST IMPORTANT FEATURES IN THE PROPOSED SYSTEM.

	Important features	Value
1st stage	Variance of Packet Time	0.4898
	Mode Packet Length	0.2022
2nd stage	Mode Packet Length	34.97
	Median Packet Length	12.19
3rd stage	Median Request/response Time Difference	3081
	Skew from Median Request/response Time Difference	2004

As shown in Table III and Table IV, the datasets have more suspicious DoH traffic flows than normal DoH. We believe the situation could occur in a real network based on our experience because DNS tunnel tools automatically and continuously send DoH traffic flows to keep the connection with a malicious DNS server. On the other hand, how the ratio between suspicious and normal traffic flows affects classification accuracy is an important issue that should be examined in the future.

As described in Section IV-B, the DoH traffic flows of emerging tunnel tools were captured for 48 hours to update the knowledge of the proposed models. The time is aligned as closely as possible with the conditions of the CIRA-CIC-DoHBrw-2020 dataset in which the well-known malicious DNS tunnel tools have been captured. However, based on our empirical estimates, we believe it is possible to reduce the time while keeping the classification accuracy obtained in this paper. To shorten the lead time for knowledge updates, we plan to investigate the optimal capturing time for the emerging tunnel tools.

Even if attackers modify parts of existing malicious DNS tunnel tools or add new functionality to them, the proposed system can update the knowledge for these tools. However, to respond flexibly to such attacks, we consider that expanding the kinds of statistical traffic features might be effective.

It should be noted that knowledge updates for the proposed system begin with running emerging malicious DNS tunnel tools to create training data of the DoH traffic. Therefore, it means that the system needs the initial assistance of security analysts to discover and obtain newly emerging malicious DNS tunnel tools.

V. CONCLUSION

The DoH protocol has been standardized to provide privacy for Internet users by encrypting DNS traffic. However, the DoH protocol also prevents network administrators from analyzing network traffic. However, since malicious DNS tunnel tools for the DoH protocol increase network security threats, network administrators need to recognize malicious communications in their networks.

We proposed a malicious DNS tunnel tool recognition system using persistent DoH traffic analysis. The proposed system can accomplish continuous knowledge updates for emerging malicious DNS tunnel tools on the machine learning model. We implemented the proposed system and evaluated its performance on the CIRA-CIC-DoHBrw-2020 dataset and the DoH-Tunnel-Traffic-HKD dataset created in this research.

The evaluation results confirm that the proposed system is able to recognize the six malicious DNS tunnel tools, not only well-known ones, including dns2tcp, dnscat2, and iodine, but also emerging ones, such as dnstt, tcp-over-dns, and tuns, with 98.02% classification accuracy. The results indicate that the proposed system is practical and beneficial for network security management.

Future work includes knowledge updates for the latest malicious DNS tunnel tools, examinations of the ratio between suspicious and normal traffic flows, investigations of the optimal capturing time, and expansions of the statistical traffic features.

REFERENCES

[1] P. Hoffman and P. McManus, "DNS Queries over HTTPS (DoH)," RFC 8484, Oct. 2018.

[2] "How to Enable DNS Over HTTPS on Windows 11," https://www.howtogeek.com/765940/how-to-enable-dns-over-https-on-windows-11/.

[3] "DNS on IOS v14 in Apple Developer Forums," https://developer.apple.com/forums/thread/663371.

[4] "Cloudflare Tunnel (Cloudflared)," https://developers.cloudflare.com/cloudflare-one/connections/connect-apps.

[5] "DNS-over-HTTPS," https://github.com/m13253/dns-over-https.

[6] "DNS Over HTTPS Proxy," https://github.com/facebookarchive/doh-proxy.

[7] "DNSCrypt," https://github.com/DNSCrypt.

[8] "doh-client," https://docs.rs/crate/doh-client/1.1.5.

[9] S. Siby, M. Juarez, C. Diaz, N. Vallina-Rodriguez, and C. Troncoso, "Encrypted DNS → Privacy?" in *Proceedings of Network and Distributed Systems Security (NDSS) Symposium 2020*, 2020.

[10] R. Mitsuhashi, A. Satoh, Y. Jin, K. Iida, T. Shinagawa, and Y. Takai, "Identifying Malicious DNS Tunnel Tools from DoH Traffic Using Hierarchical Machine Learning Classification," in *Information Security*, 2021, pp. 238–256.

[11] S. Ajmera and T. Pattanshetti, "A Survey Report on Identifying Different Machine Learning Algorithms in Detecting Domain Generation Algorithms within Enterprise Network," in *Proceedings of 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2020, pp. 1–5.

[12] H. Ichise, Y. Jin, and K. Iida, "Analysis of DNS TXT Record Usage and Consideration of Botnet Communication Detection," *IEICE Transactions on Communications*, vol. E101, no. 1, pp. 70–79, 2018.

[13] H. Ichise, Y. Jin, K. Iida, and Y. Takai, "NS record History Based Abnormal DNS traffic Detection Considering Adaptive Botnet Communication Blocking," *IPSJ Journal of Information Processing*, vol. 28, pp. 112–122, 2020.

[14] Y. Iuchi, Y. Jin, H. Ichise, K. Iida, and Y. Takai, "Detection and Blocking of DGA-based Bot Infected Computers by Monitoring NXDOMAIN Responses," in *Proceedings of 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, 2020, pp. 82–87.

[15] Y. Wang, A. Zhou, S. Liao, R. Zheng, R. Hu, and L. Zhang, "A Comprehensive Survey on DNS Tunnel Detection," *Comput. Netw.*, vol. 197, no. C, oct 2021.

[16] P. Yang, X. Wan, G. Shi, H. Qu, J. Li, and L. Yang, "Naruto: DNS Covert Channels Detection Based on Stacking Model," in *Proceedings of the 2020 The 2nd World Symposium on Software Engineering (WSSE)*, 2020, p. 109–115.

[17] A. L. Buczak, P. A. Hanke, G. J. Cancro, M. K. Toma, L. A. Watkins, and J. S. Chavis, "Detection of Tunnels in PCAP Data by Random Forests," in *Proceedings of the 11th Annual Cyber and Information Security Research Conference (CISRC)*, 2016.

[18] D. Lambion, M. Josten, F. Olumofin, and M. De Cock, "Malicious DNS Tunneling Detection in Real-Traffic DNS Data," in *Proceedings of 2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 5736–5738.

[19] Y. Chen and X. Li, "A High Accuracy DNS Tunnel Detection Method Without Feature Engineering," in *Proceedings of 2020 16th International Conference on Computational Intelligence and Security (CIS)*, 2020, pp. 374–377.

[20] A. Chowdhary, M. Bhowmik, and B. Rudra, "DNS Tunneling Detection using Machine Learning and Cache Miss Properties," in *Proceedings of 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2021, pp. 1225–1229.

[21] C. Kwan, P. Janiszewski, S. Qiu, C. Wang, and C. Bocovich, "Exploring Simple Detection Techniques for DNS-over-HTTPS Tunnels," in *Proceedings of the ACM SIGCOMM 2021 Workshop on Free and Open Communications on the Internet*, ser. FOCI '21, 2021, p. 37–42.

[22] "dnstt," https://www.bamsoftware.com/software/dnstt/.

[23] K. Wu, Y. Zhang, and T. Yin, "FTPB: A Three-Stage DNS Tunnel Detection Method Based on Character Feature Extraction," in *Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2020, pp. 250–258.

[24] D. Vekshin, K. Hynek, and T. Cejka, "DoH Insight: Detecting DNS over HTTPS by Machine Learning," in *Proceedings of the 15th International Conference on Availability, Reliability and Security (ARES)*, 2020.

[25] "Amazon Alexa Voice AI," https://developer.amazon.com/en-US/alexa/.

[26] L. Csikor, H. Singh, M. S. Kang, and D. M. Divakaran, "Privacy of DNS-over-HTTPS: Requiem for a Dream?" in *Proceedings of the 2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2021, pp. 252–271.

[27] "CIRA-CIC-DoHBrw-2020," https://www.unb.ca/cic/datasets/dohbrw-2020.html.

[28] M. MontazeriShatoori, L. Davidson, G. Kaur, and A. Habibi Lashkari, "Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic," in *Proceedings of the 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2020, pp. 63–70.

[29] Y. Khodjaeva and N. Zincir-Heywood, "Network Flow Entropy for Identifying Malicious Behaviours in DNS Tunnels," in *Proceedings of the 16th International Conference on Availability, Reliability and Security (ARES)*, 2021.

[30] S. K. Singh and P. K. Roy, "Detecting Malicious DNS over HTTPS Traffic Using Machine Learning," in *Proceedings of the 2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies*, 2020, pp. 1–6.

[31] R. Alenezi and S. A. Ludwig, "Classifying DNS Tunneling Tools For Malicious DoH Traffic," in *Proceedings of the 2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2021, pp. 1–9.

[32] M. Behnke, N. Briner, D. Cullen, K. Schwerdtfeger, J. Warren, R. Basnet, and T. Doleck, "Feature Engineering and Machine Learning Model Comparison for Malicious Activity Detection in the DNS-Over-HTTPS Protocol," *IEEE Access*, vol. 9, pp. 129 902–129 916, 2021.

[33] L. F. Gonzalez Casanova and P.-C. Lin, "Generalized Classification of DNS over HTTPS Traffic with Deep Learning," in *Proceedings of the 2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2021, pp. 1903–1907.

[34] "dns2tcp," https://github.com/alex-sector/dns2tcp.

[35] "dnscat2," https://github.com/iagox86/dnscat2.

[36] "iodine," https://code.kryo.se/iodine/.

[37] "tcp-over-dns," https://analogbit.com/software/tcp-over-dns/.

[38] "tuns," https://github.com/lnussbaum/tuns.

[39] L. Nussbaum, P. Neyron, and O. Richard, "On robust covert channels inside DNS," in *Proceedings of the IFIP International Information Security Conference*, 2009, pp. 51–62.

[40] "The DoH-Tunnel-Traffic-HKD dataset," https://github.com/doh-traffic-dataset/DoH-Tunnel-Traffic-HKD.

[41] "DoHlyzer," https://github.com/ahlashkari/DoHlyzer.

[42] C. Tianqi and G. Carlos, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, p. 785–794.

[43] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 30, 2017.

[44] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 31, 2018.

[45] S. R, S. S. Ayachit, V. Patil, and A. Singh, "Competitive Analysis of the Top Gradient Boosting Machine Learning Algorithms," in *Proceedings of the 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, 2020, pp. 191–196.

[46] "Kaggle," https://www.kaggle.com/.
[47] "XGBoost Documentation - Xgboost Parameters," https://xgboost.readthedocs.io/en/latest/parameter.html.
[48] "LightGBM Documentation - Parameters," https://lightgbm.readthedocs.io/en/latest/Parameters-Tuning.html.
[49] "CatBoost Documentation - Parameters," https://catboost.ai/en/docs/concepts/parameter-tuning.
[50] "NGINX," https://www.nginx.com/.

Rikima Mitsuhashi received Bachelor of Informatics in Computer Science from Shizuoka University in 2000, M.E. in Computational Intelligence and Systems Science from Tokyo Institute of Technology in 2002, and Ph.D. in Information Science and Technology from the University of Tokyo in 2022. He is currently a Visiting Researcher for the Information Initiative Center, Hokkaido University, and the Information Technology Center, the University of Tokyo. His research interests include secure computing, network security, and artificial intelligence. He is a member of IEEE, IEICE, and IPSJ.

Yong Jin received M.E. and Ph.D. in Communication Network Engineering from Okayama University, Okayama, Japan in 2009 and 2012, respectively. Currently, he is an Associate Professor for Institute Management in the Global Scientific Information and Computing Center of Tokyo Institute of Technology, Tokyo, Japan. His research interests include Internet technology, network security, traffic engineering, and network architecture. He is a member of ACM, IEEE, IEICE, and IPSJ.

Katsuyoshi Iida received B.E. in Computer Science and Systems Engineering from Kyushu Institute of Technology (KIT), Iizuka, Japan in 1996, M.E. in Information Science from Nara Institute of Science and Technology, Ikoma, Japan in 1998, and Ph.D. in Computer Science and Systems Engineering from KIT in 2001. Currently, he is an Associate Professor in the Information Initiative Center, Hokkaido University, Sapporo, Japan. His research interests include network systems engineering such as network architecture, performance evaluation, QoS, and mobile networks. He is a member of IEEE and a senior member of IEICE. He received the 18th TELECOM System Technology Award, and Tokyo Tech young researcher's award in 2003, and 2010, respectively.

Takahiro Shinagawa received B.E., M.S., and Ph.D. in science from the University of Tokyo in 1998, 2000, and 2003, respectively. After working as an Assistant Professor at Tokyo University of Agriculture and Technology and a Lecturer at University of Tsukuba, he has been an Associate Professor at the University of Tokyo since 2011. He has also been a Visiting Researcher at Imperial College London from 2021 to 2022. His research interests include operating systems, system software, virtualization, and secure computing. He is a member of ACM, IEEE, and USENIX.

Yoshiaki Takai received B.E. in electronic engineering, and M.E. and Ph.D. in information engineering from Tohoku University in 1983, 1985 and 1988, respectively. From 1988 to 1989, he was a Research Associate of the Faculty of Science, the University of Tokyo. In 1989, he joined the Faculty of Engineering, Hokkaido University. From 2011 to 2019, he was a Director of Information Initiative Center, Hokkaido University. He is currently a Professor of Information Initiative Center, Hokkaido University. He is also an Executive Adviser for CIO of Hokkaido University. His research interests include parallel computer architecture, parallel processing, distributed processing, computer networks, augmented reality, virtual reality, physically-based modeling, and computer graphics applications. He is a member of IEEE, IEICE, and IPSJ.