2022 18th International Conference on Mobility, Sensing and Networking (MSN) | 978-1-6654-6457-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/MSN57253.2022.00073

Detection of DoH Tunnels with Dual-Tier Classifier

Yuqi Qiu*†‡, Baiyang Li*†‡, Liang Jiao*†‡§ Yujia Zhu*†‡✉, Qingyun Liu*†‡
*Institute of Information Engineering, Chinese Academy of Sciences
†School of Cyber Security, University of Chinese Academy of Sciences
‡National Engineering Laboratory for Information Security Technologies
§Shandong Branch of National Computer Network Emergency Response Technical Team/ Coordination Center of China
Email: zhuyujia@iie.ac.cn

***Abstract*—DNS over HTTPS (DoH) has been deployed to provide confidentiality in the DNS resolution process. However, encryption is a double-edged sword in providing security while increasing the risk of data tunneling attacks. Current approaches for plaintext DNS tunnel detection are disabled. Due to the diversity of tunneling tool variations and the low proportion of tunneled traffic in real situations, detecting malicious behaviors is becoming more and more challenging.**

In this paper, we propose a novel behavior-based model with Dual-Tier Tunnel Classifier (DTC) for tool-level DoH tunneling detection. The major advantage of DTC is that it can not only capture existing tunneling tools but also explore unknown ones in the wild. In particular, DTC considers data imbalance, which improves robustness of the model in the open environment. Our method has been proven successful in both closed and open scenarios, achieving 99.99% accuracy in detecting known malicious DoH traffic, 96.93% accuracy in unknown and 95.31% accuracy in identifying malicious DoH tunnel tools.

***Index Terms*—DoH Tunnels, Traffic Detection, Variational Auto-encoder, Machine Learning**

I. Introduction

The Domain Name System (DNS), a crucial component of the Internet's infrastructure, provides a service that maps human-readable domain names to computer-understandable IP addresses. However, the traditional DNS protocol (Do53), sent queries over UDP or TCP, is in plaintext, which makes Do53 open to attacks and other forms of exploitation.

As security issues of Do53 are gradually gaining attention, encrypted DNS protocols are introduced to secure user's privacy and ensure the trustworthiness and dependability of the domain name resolution process. Among these protocols, DNS-over-HTTPS (DoH) is the most popular, which uses HTTPS protocol to encapsulate DNS content, transmitting over TCP protocol [1]. Figure 1 shows the schematic diagram of DoH resolution.

Although DoH addresses some security issues in Do53, it is a double-edged sword that also makes attackers invisible. Malware can leverage DoH to bypass existing security monitoring tools for command and control (C&C) communication and data exfiltration, exhibiting more dynamic and stealthy behaviors [2]. The Spamhaus research group found that the number of botnet C&C activities increased by 23% in the fourth quarter of 2021. They could not track these activities because the communications were conducted through DoH [3]. DoH reduces the efficacy of security systems dependent on Do53 monitoring and filtering, making it more difficult to identify malicious behaviors. Major DoH providers do not filter malicious DNS resolutions for botnets, phishing, or malware domains which exacerbates the security issue.

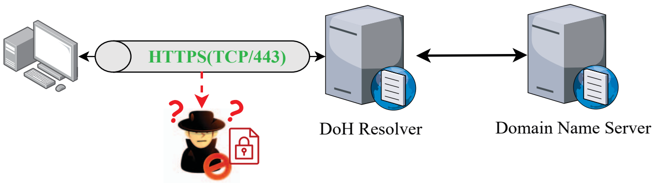


Fig. 1. DNS-over-HTTPS resolution.

It is essential to identify malicious traffic from DoH traffic. However, most current methods for detecting DNS tunnels based on domain characteristics would fail due to encryption. Our work aims to investigate DoH based data exfiltration and tunnel tools. We design an efficient algorithm for detecting malicious DoH traffic and identifying tunnel tools based on our investigation. The main contributions in this paper include:

- Considering the lack of DoH dataset and the fact that existing open source only use traditional DNS tunnel tools, different DoH capable tunnel tools are used in various settings to simulate real-world attack traffic to create a richer dataset.
- Dual-tier Tunnel Classifier (DTC) is proposed to firstly filter out suspicious DoH traffic by VAE, then detect tunneling DoH by multi-classifier and finally identify tunnel tools.
- Extensive experiments are conducted on both closed-world and open-world dataset. The results demonstrate the effectiveness of DTC in detecting unknown tunneling DoH traffic and identifying tunnel tools.

The remainder of this paper is organized as follows. Section II gives an overview of related work on DoH tunnel detection. We provide a detailed dataset in Section III, followed by a proposed methodology in Section IV. Section V presents experiment on DoH malicious traffic detection and analyses of the results. Finally, Section VI concludes the work.

II. Related Work

This section discusses literature related to DNS tunneling detection and detection of malicious tunnels of DoH.

978-1-6654-6457-4/22/$31.00 ©2022 IEEE
DOI 10.1109/MSN57253.2022.00073

TABLE I
OVERVIEW OF PREVIOUS DoH TRAFFIC DATASET

Author	Benign Browser	Malicious Tunnel Tool	Resolver	Year	Publicly Available
M. MontazeriS [4]	firefox, chrome	dns2tcp [5], dnscat2 [6], iodine [7]	Adguard, Cloudflare, Google, Quad9	2020	✓
T. A. Nguy [8]	firefox	dns2tcp, dnscat2, iodine	Adguard, Cloudflare, Google, Quad9	2021	✗
M. Zhan [9]	firefox	dnsexfiltrator [10]	Cloudflare, Google, Quad9	2022	✗

A. DNS Tunnel Detection

DNS tunnels have been widely used by malware to transfer data secretly [11]. Studies [12], [13] and [14] proposed DNS tunnel detection solutions based on deep packet inspection.Ellens et al. [15] and Liu et al. [16] extracted features from traffic and designed various detectors to detect DNS tunnels. Wu et al. [17] used an auto-encoder-based semi-supervised feature learning approach for DNS covert channel detection, avoiding manual feature extraction. All these methods are applicable to Do53, not encrypted ones.

B. DoH Tunnel Detection

While DoH enhances DNS privacy and security, it also provides opportunities for malicious activity. Lyu et al. [18] surveyed the current state of encrypted DNS, highlighting that DoH can be exploited for botnet and data exfiltration.

Recent studies have constructed DoH tunnel detection efforts through machine learning techniques. Montazeri et al. [4] extracted statistical features of DoH flows and used various classifiers to detect malicious DoH traffic with an accuracy of 99.3%. They generated a dataset called CIRA-CIC-DoHBrw-2020 [19]. Studies [20], [21], [22] use the dataset to detect malicious activities in DoH, achieving accuracy above 99%. Zhan et al. [9] performed DoH based exfiltration in different locations worldwide and tested multiple classifiers with detection accuracy above 99%.

With the development of the DoH, various DoH capable tunnel tools have arisen. Previous studies have not tested their model on new-emerging tunnel tools. As the ability to detect unknown samples is essential, there is room for optimization in existing work.

III. DoH Traffic Dataset

A. Open Datasets

Due to the novelty of the DoH protocol, it is difficult to obtain real DoH attack traffic. Available open datasets are scarce. Several research efforts have configured browsers to access websites using DoH resolution services to capture benign DoH traffic and replaced malware with open source DNS tunneling tools to generate DoH tunnel traffic in a self-built environment. The DoH datasets used in related work in recent years are shown in Table I.

The ”CIRA-CIC-DoHBrw-2020” dataset introduced in section II is the most used in malicious DoH traffic detection. We also use the datset in our experiment and call it as DoHBrw thereafter. The datasets shown in Table I only use four DoH service providers, Adguard, Cloudflare, Google and Quad9. In addition, aside from dnsexfiltration, the tunnel tools they used are traditional unencrypted DNS, translated into DoH using a proxy. To evaluate the effectiveness and robustness of our method, we deploy DoH capable tunnel tools in a preset environment to simulate malicious traffic.

B. Data Collection

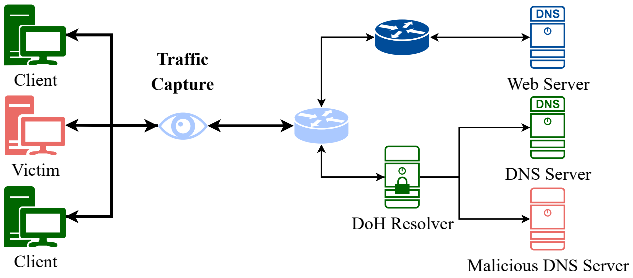


Fig. 2. Network topology diagram for capturing DoH tunnel traffic

Several settings are employed to simulate possible scenarios to gather benign and tunneling DoH traffic. The details of settings are described explicitly in Table II and the network topology is shown in Figure 2. Four hosts are located in Beijing and Hong Kong, with two of them in each city, serve as the victim clients. Another one host in Hong Kong simulates an attacker-controlled authoritative domain name server. We employ 6 different resolvers, four used in previous studies and two unused ones, Alidns and 360dns. To collect the final DoH traffic, we used tcpdump to capture the network traffic between the victim and DoH resolver.

For benign traffic, three kinds of browsers are configured to use DoH only in client hosts to visit the top 2000 websites in Chinaz Top, collecting DoH traffic for each visit. The automation tool, Selenium, is used to drive the browsers to visit the websites automatically in headless mode.

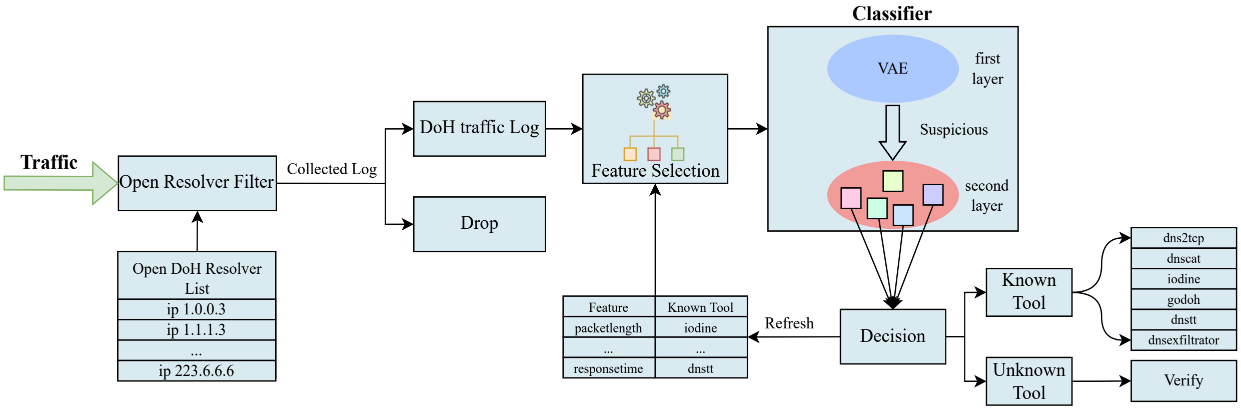
To configure the DoH tunnel, we registered two domains and set our authoritative name server. Dnstt, godoh and dnsexfiltrator are the three tools used to emulate the DoH tunnel. Different DoH proxies are used to send local files (we use text files) to the name server through the DoH tunnel. We set different query intervals and different data lengths as shown in Table II. We name the collected dataset DoHDTC.

IV. Framework

In this section, we propose a tunneling detection model called DTC. The overall framework diagram is shown in Figure 3, which consists of five parts.

TABLE II
OVERVIEW OF THE SETTINGS OF OUR DATA CAPTURING PROCESS.

Settings	**Tunneling**	**Benign**
Location	2	2
OS	Centos, Windows	Centos, Windows
Domain Name	noraup.xyz, norawork.xyz	-
DoH Resolver	Google, Cloudflare, Adguard, Quad9, Alidns, 360dns	Google, Cloudflare, Adguard, Quad9, Alidns, 360dns
Tool/Browser	dnstt [23], godoh [24], dnsexfiltrator	chrome, firefox, edge
Time Interval	1, 5, 25, 125, 625, 1000, 2000, 5000ms	-
Length	20 to 200	-
Websites	-	2000



Feature	Known Tool
packetlength	iodine
...	...
responsetime	dnstt

Fig. 3. Framework of proposed method to identify tunneling DoH traffic.

A. DoH Traffic Filtering

Using the method of automatically discovering DoH resolvers proposed by Wu et al. [25] and the list of public DoH resolver ip (publicly available at the server[1]), we filter the captured traffic to obtain the DoH traffic.

TABLE III
SAMPLE OF FEATURES

Description of features	
F1	duration
F2-F5	rate/number of flow bytes sent/receive
F6-F13	packet length(mean,median, mode, variance, standard deviation, coefficient of variation, skew from median, skew from mode)
F14-F21	packet time(mean,median, mode, variance, standard deviation, coefficient of variation, skew from median, skew from mode)
F22-F29	req/res time(mean,median, mode, variance, standard deviation, coefficient of variation, skew from median, skew from mode)

B. Traffic Log Collecting

We use the DoHlyzer tool[2] to extract time series and statistical features for model training. The specific feature information is shown in Table III. We process the data as follows afterward.

- Redundant values and missing values will be removed in whole rows.
- All values are normalized within a specific range to improve the computation of the classification algorithm.
- Benign and malicious data are represented by 0 and 1, respectively.
- All the data is stored in log files.

C. Feature Selection

In this phase, we calculate the information value of each feature using mutual information, then adjust the features and their weights according to the results. The combination with the highest accuracy is selected as the final set of features.

D. Malicious DoH Detection Module

The main goal of this module is to detect DoH tunneling from DoH flows. First, we use the variational auto-encoder as the one-class classifier to filter tunneling DoH. As shown in Figure 4, VAE is bidirectional, i.e. a model not only learns a probabilistic model $p_\theta(x|z)$ of observed variable x conditioned on a latent variable z, but also a model $q_\phi(z|x)$ for the latent variable conditioned on the observed variable. The approximation function $q_\phi(z|x)$ is the probabilistic encoder and the conditional probability $p_\theta(x|z)$ is the probabilistic decoder. In the training step, the loss function is defined as:

$$Loss = Reconstruction\ Loss + KL\ Loss \quad (1)$$

KL loss is the KL-Divergence between the distribution modeled by the encoder and the prior distribution over the latent

[1] https://github.com/curl/curl/wiki/DNS-over-HTTPS

[2] https://github.com/ahlashkari/DOHlyzer/tree/master/

space. We take the upper limit of the 95% confidence interval as the classification threshold t. In the testing phase, samples with Mean squared error less than t are classified as suspicious flows. The network structure we use is feedforward neural network, with 2 hidden layers, and the dimensionality is 50. Experiments use Adam optimizer for parameter optimization, the model learning rate is 0.001.

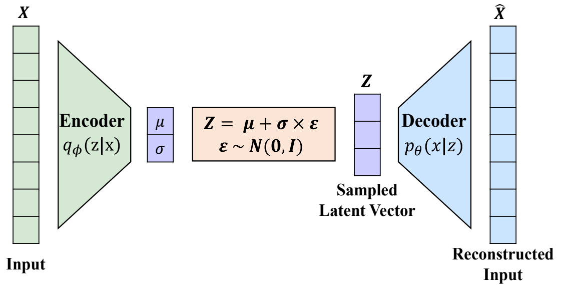


Fig. 4. Illustration of Variational Auto-Encoder Model Architecture.

Then these suspicious flows are fed into a multi-classifier to detect tunneling DoH in the second layer. We use decision trees, random forests and LGBM as multi-classifiers in all experiments, implemented with the scikit-learn with no additional tuning of the model parameters.

E. DoH Tunnel Tools Detection Module

In the tunnel tool detection phase, we employ the above-mentioned multi-classifier for identification. In the next section, we focus on the identification of six tools, along with the well-known and frequently used DNS tunnel tools, DoH capable tunnel tools like the three we utilized in Section III. If the identification results of the same flow are particularly scattered, we consider them as traffic generated by an unknown tool and add them to the list of those requiring further validation.

V. Evaluation

A. Evaluation Setup

Table IV shows the details of the number of flows used in the experiment. A closed-world dataset and an open-world dataset are used in the evaluation experiments. From each flow in the dataset, we extracted 29 statistical traffic features and preprocessed them.

The closed-world dataset consists of the data we collected in section IV and the publicly available dataset DoHBrw mentioned before. We collect traffic at the gateway of our partners to create our open-world dataset. All flows are obtained by bypass and normal communication is not interfered with. In addition, dnstt is used to construct DoH tunnel in this network, so the dataset contains tunneling traffic. Finally, we obtain 8695 flows. In terms of ethics, all source IP addresses are hidden, so privacy risks should be minimized.

The dataset is randomly partitioned 4:1 into a training set and test set. We use 10-fold cross-validation to classify the traffic, and the model achieved the highest F-score on the validation set is used for testing. Accuracy, Precision, Recall, and F1 Score are used as evaluation metrics for the model.

TABLE IV
DATASET DETAILS

	Dataset	Type	Browser/Tool	Flows
Closed-World	DoHBrw	benign	firefox	16273
			chrome	3534
		malicious	dns2tcp	4531
			dnscat	4549
			iodine	5334
	DoHDTC	benign	firefox	20000
			chrome	20000
			edge	20000
		malicious	godoh	3579
			dnstt	3284
			dnsexfiltrator	2284
Open-World	collected from gateway			8695

B. Detection on Closed-World Dataset

TABLE V
Experimental results of the overall dataset

Classifier	Accuracy	Precision	Recall	F-Score
VAE+DT	0.99520	0.99519	0.99520	0.99518
VAE+RF	0.99971	0.99971	0.99971	0.99971
VAE+LGBM	0.99990	0.99990	0.99990	0.99990

Initially, we first combine all the closed-world data into one overall dataset to test the performance of the model. The experimental results are shown in Table V. It can be demonstrated that tunneling DoH traffic differs from normal DoH traffic and that the classifier can achieve good results.

TABLE VI
Performance of models in unknown sample detection

Research	Classification algorithms	Recall
M. MontazeriS [4]	DT, RF, SVM, NB, CNN	0.94647
S. K. Singh [20]	LR, RF, KNN, GB	0.89769
Banadaki [21]	LGBM, XGBoost	0.73471
M. Behnke [22]	DT, RF, GB, LGBM, XGBoost	0.84969
Our Model	**DTC**	**0.96935**

While the model is able to achieve high detection rates, real-world attack traffic is more diverse. To evaluate the generalization ability of the classifier, we conducted unknown sample detection experiments. The traffic in DoHDTC and DoHBrw dataset were generated in completely different environments, so we trained the model through DoHBrw and performed test experiments with data from DoHDTC to verify the feasibility of the model. In addition, we compare it with the model used in previous studies. The results are shown in Tabel VI.

Our model DTC achieved the highest recall of 96.935% compared to other existing methods. It can be seen that DTC can also show good detection ability for unknown samples. We analyse the results and find that traffic using Alidns and 360dns DoH service makes up the majority of the misclassified traffic. Resolvers use various implementation and deployment techniques, resulting in variations in response times while resolving domain names, which affects the classification. However, our approach first filters the tunneling traffic using VAE to avoid most normal traffic from being misclassified.

TABLE VII
PERFORMANCE OF MODEL IN OPEN-WORLD DATASET

Data	Malicious DoH	Benign DoH	Non-DoH	Accuracy
2-minute	2	14	1313	100%
9-minute	7	24	7335	100%

C. Detection on Open-World Dataset

We examined the gateway traffic in the open-world dataset to test whether there would be error detection in a more realistic environment. Closed-world dataset used as the training dataset.The open-world dataset contains some one-way traffic and a very low percentage of DoH traffic. We collected 2-minute and 9-minute TLS flows at any two times and fed them into the model for testing. The results are shown in Table VIII. Our model achieves 100% accuracy on both parts of the data, demonstrating the robustness of the model.

D. Identification of the DoH Tunnel Tools

TABLE VIII
DOH TUNNEL TOOLS DETECTI EXPERIMENTAL RESULTS

Classifier	Accuracy	Precision	Recall	F-Score
DT	0.93412	0.93501	0.93412	0.93449
RF	0.95317	0.95333	0.95317	0.95281
LGBM	0.95317	0.95280	0.95317	0.95286

Six DNS tunnel tools—dns2tcp, dnscat, iodine, godoh, dnstt, and dnsexfiltrator—were used in this experiment, and the results are presented in Table VII. The results indicate that the proposed detection method is effective in identifying different tunneling tools with an accuracy of 95.32%. To the best of our knowledge, this is the first attempt to identify the main tunneling tools supporting DoH.

VI. CONCLUSION

In this paper, we propose DTC, a Dual-tier Tunnel Classifier to detect DoH tunneling traffic. We first construct a scalable DoH traffic collection environment and build a rich traffic dataset. Various factors affecting DoH traffic are considered, including different DoH services, geographic locations, packet sizes, and data sending intervals. Then we perform closed-world and open-world experiments to detect DoH tunnels. The results show that DTC has a better generalization to detect unknown malicious traffic. The accuracy of DTC can reach 99.99% for malicious DoH traffic detection and 95.31% for tunneling tool identification. Our evaluation shows that DTC can achieve high accuracy in a large-scale traffic environment and detect unknown samples.

ACKNOWLEDGMENT

This work is supported by the Strategic Priority Research Program of the Chinese Academy of Sciences with No. XDC02030000 and the DongGuan Innovative Research Team Program with No. 201636000100038.

REFERENCES

[1] P. M. P. Hoffman, "Dns queries over https (doh)," https://datatracker.ietf.org/doc/html/rfc8484, 2018.

[2] C. de Faria Cristas, "A closer look at flubot's doh tunneling," https://blog.f-secure.com/flubot_doh_tunneling/, 2022.

[3] S. M. Team, "Spamhaus botnet threat update: Q4-2021," https://www.spamhaus.org/news/article/817/spamhaus-botnet-threat-update-q4-2021, 2021.

[4] M. MontazeriShatoori, L. Davidson, G. Kaur, and A. H. Lashkari, "Detection of doh tunnels using time-series classification of encrypted traffic," in *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)*. IEEE, 2020, pp. 63–70.

[5] Dembour, "Dns2tcp," https://www.kali.org/tools/dns2tcp/.

[6] T. Pietraszek, "Dnscat," http://tadek.pietraszek.org/projects/DNScatl.

[7] Andersson, "Iodine," http://code.kryo.se/iodine/2011.

[8] T. A. Nguyen and M. Park, "Doh tunneling detection system for enterprise network using deep learning technique," *Applied Sciences*, vol. 12, no. 5, p. 2416, 2022.

[9] M. Zhan, Y. Li, G. Yu, B. Li, and W. Wang, "Detecting dns over https based data exfiltration," *Computer Networks*, vol. 209, p. 108919, 2022.

[10] L. Jacobs, "Data exfiltration over dns request covert channel," https://github.com/Arno0x/DNSExfiltrator.

[11] L. Nussbaum, P. Neyron, and O. Richard, "On robust covert channels inside dns," in *IFIP International Information Security Conference*. Springer, 2009, pp. 51–62.

[12] V. Paxson, M. Christodorescu, M. Javed, J. Rao, R. Sailer, D. L. Schales, M. Stoecklin, K. Thomas, W. Venema, and N. Weaver, "Practical comprehensive bounds on surreptitious communication over {DNS}," in *22nd USENIX Security Symposium (USENIX Security 13)*, 2013, pp. 17–32.

[13] A. Nadler, A. Aminov, and A. Shabtai, "Detection of malicious and low throughput data exfiltration over the dns protocol," *Computers & Security*, vol. 80, pp. 36–53, 2019.

[14] M. Luo, Q. Wang, Y. Yao, X. Wang, P. Yang, and Z. Jiang, "Towards comprehensive detection of dns tunnels," in *2020 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2020, pp. 1–7.

[15] W. Ellens, P. Żuraniewski, A. Sperotto, H. Schotanus, M. Mandjes, and E. Meeuwissen, "Flow-based detection of dns tunnels," in *IFIP International Conference on Autonomous Infrastructure, Management and Security*. Springer, 2013, pp. 124–135.

[16] J. Liu, S. Li, Y. Zhang, J. Xiao, P. Chang, and C. Peng, "Detecting dns tunnel through binary-classification based on behavior features," in *2017 IEEE Trustcom/BigDataSE/ICESS*. IEEE, 2017, pp. 339–346.

[17] K. Wu, Y. Zhang, and T. Yin, "Tdae: Autoencoder-based automatic feature learning method for the detection of dns tunnel," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–7.

[18] M. Lyu, H. H. Gharakheili, and V. Sivaraman, "A survey on dns encryption: Current development, malware misuse, and inference techniques," *arXiv preprint arXiv:2201.00900*, 2022.

[19] C. I. for Cybersecurity, "Cira-cic-dohbrw-2020," https://www.unb.ca/cic/datasets/dohbrw-2020, 2020.

[20] S. K. Singh and P. K. Roy, "Detecting malicious dns over https traffic using machine learning," in *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*. IEEE, 2020, pp. 1–6.

[21] Y. M. Banadaki, "Detecting malicious dns over https traffic in domain name system using machine learning classifiers," *Journal of Computer Sciences and Applications*, vol. 8, no. 2, pp. 46–55, 2020.

[22] 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.

[23] "dnstt," https://github.com/pjanisze/dnstt-uTLS/tree/dnstt-utls.

[24] Arno, "godoh," https://github.com/sensepost/godoh.

[25] J. Wu, Y. Zhu, B. Li, Q. Liu, and B. Fang, "Peek inside the encrypted world: Autoencoder-based detection of doh resolvers," in *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2021, pp. 783–790.