DoH Insight: Detecting DNS over HTTPS by Machine Learning

Dmitrii Vekshin
vekshdmi@fit.cvut.cz
FIT CTU in Prague
Prague, Czech Republic

Karel Hynek
hynekkar@fit.cvut.cz
FIT CTU in Prague & CESNET z.s.p.o.
Prague, Czech Republic

Tomas Cejka
cejkat@cesnet.cz
CESNET, z.s.p.o.
Prague, Czech Republic

ABSTRACT

Over the past few years, a new protocol DNS over HTTPS (DoH) has been created to improve users' privacy on the internet. DoH can be used instead of traditional DNS for domain name translation with encryption as a benefit. This new feature also brings some threats because various security tools depend on readable information from DNS to identify, e.g., malware, botnet communication, and data exfiltration. Therefore, this paper focuses on the possibilities of encrypted traffic analysis, especially on the accurate recognition of DoH. The aim is to evaluate what information (if any) can be gained from HTTPS extended IP flow data using machine learning. We evaluated five popular ML methods to find the best DoH classifiers. The experiments show that the accuracy of DoH recognition is over 99.9 %. Additionally, it is also possible to identify the application that was used for DoH communication, since we have discovered (using created datasets) significant differences in the behavior of Firefox, Chrome, and cloudflared. Our trained classifier can distinguish between DoH clients with the 99.9 % accuracy.

CCS CONCEPTS

• **Computing methodologies** → **Classification and regression trees**; • **Networks** → **Web protocol security**; *Network privacy and anonymity*; • **Security and privacy** → Browser security.

KEYWORDS

DNS over HTTPS, DoH, Detection, Classification, Machine Learning, Datasets

ACM Reference Format:
Dmitrii Vekshin, Karel Hynek, and Tomas Cejka. 2020. DoH Insight: Detecting DNS over HTTPS by Machine Learning. In *The 15th International Conference on Availability, Reliability and Security (ARES 2020), August 25–28, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3407023.3409192

1 INTRODUCTION

Translation of human-readable domain names into machine-usable IP addresses and vice versa is an essential feature that enables a user-friendly usage of the network services. Traditionally, this mechanism is performed by Domain Name System (DNS) [22, 23] in the Internet environment. DNS is one of the oldest network protocols and, therefore, it is based on transferring unencrypted queries and answers through the network links.

DNS traffic is crucial for many existing security systems. Since an application must translate a domain name before a connection can be established, DNS traffic can identify many security threats that are observable in the network traffic. Readability of translated domain names in the traffic is exploited in application firewalls to check security policies, and intrusion detection systems to detect suspicious connections, such as botnet activity.

Increased number of DNS queries can indicate communication tunnels over DNS, as it is analyzed, e.g., in [6]. Such suspicious traffic can be an indicator of data exfiltration, which must be detected as soon as possible, especially in commercial environments. There are many papers about the detection of Domain Generation Algorithms (DGA) based on observation of DNS queries either on DNS resolvers or at monitoring probes of the monitoring systems. Naturally, without visibility into DNS traffic, revealing infected machines and botnets becomes harder. Also, there are various tools (e.g., Next-Generation Firewall by Fortinet[1]) to check and enforce security policies, such as permitting access to a limited number of services in corporate networks, and parental control. This feature is also usually based on the analysis of DNS traffic. Finally, there are plenty publicly available or commercial blacklists (containing known Command and Control servers, phishing servers, malware hosting, infected devices, etc.) that are used for traffic filtering. Efficient filtering (used, e.g., in [32]) requires knowledge of looked up domain names either in TLS extension called Server Name Indication (SNI), or more easily in DNS traffic. To sum up, a long history of unencrypted DNS traffic caused that many tools for network security and forensic analysis depend on the information about looked up domain names.

On the other hand, visibility into DNS communication is recently related to the possible eavesdropping and profiling of user's activities, e.g., for commercial profit motivation. The reason is that practically everyone "in-path" (especially Internet Service Provider) is able to see the content of DNS queries, i.e., activities of users on the Internet. Therefore, DNS over HTTPS (DoH) and DNS over TLS (DoT) were a natural reaction of the Internet engineering community to improve user privacy and minimize the profiling of DNS traffic by network operators.

DoH, which is the main focus of this paper, was published in [18] in October 2018. Since the communication via DoH (DoT as well) is encrypted and visible only to the user and his DoH service provider, profiling is practically impossible. Besides the encryption of DNS data, DoH moves the visibility into looked up domain names from local DNS providers to more centralized DoH providers, which

ARES 2020, August 25–28, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-8833-7/20/08…$15.00
https://doi.org/10.1145/3407023.3409192

[1] https://www.fortinet.com/products/next-generation-firewall.html

may be useful for users as it is discussed in [13]. The first web browser, which tested DoH support, was Mozilla Firefox since 2017. It was followed by Chromium, an open-source version of Google Chrome. Nowadays, DoH is already supported natively in all most popular web browsers, such as Firefox, Chrome, and Edge (the list of web browsers that currently support DoH can be found in ZDNET article [10]). The support is still mainly in an opt-in mode, so users have to turn it on explicitly. However, Mozilla enabled DoH for US users by default on Feb 25, 2020. Hence, we expect a massive percentage increase of DoH traffic, even though only dozens of DNS providers support it [11, 24]. Cloudflare plays an essential role as a default DoH provider for some web browsers (Firefox, Opera). It provides an open-source DoH client that can be easily used as a DNS-to-DoH proxy for a local network. Additionally, in the time of writing of this paper, Microsoft published an announcement [19] about DoH support in the Windows operating system for testing. Therefore, it is highly expected that the use of DoH will increase rapidly in the near future.

Unfortunately, besides the benefits of user privacy, there is a significant security risk of DoH that is related to the decreased visibility for the security tools and applications that were mentioned above. Therefore, the motivation to analyze DoH traffic rises to find a feasible way to provide useful information about devices that start communicate via encrypted DoH. Additionally, there are already observations about DoH misused for malicious activities, e.g., [9] announces the first occurrence of malware that intentionally uses DoH to hide its communication with Command and Control servers. Haddon et al. in the paper [15] describe possible ways of data exfiltration using DoH, which is much more difficult to detect using current tools.

Based on the described motivation, we have decided to analyze the encrypted traffic of DoH to evaluate what information (if any) is possible to reveal for network security analysis. The aim is to check the possibilities of analysis using machine learning (ML) algorithms with newly prepared training and validation datasets. The primary goal is to detect DoH communication, i.e., distinguish DoH from ordinary HTTPS traffic. Furthermore, we have focused on the DoH traffic from several DoH clients and elaborated a way to recognize specific clients just by behavioral features of the network traffic. Our discussion in Sec. 6 summarizes our lessons learnt and observations about evaluated DoH clients.

The paper is divided as follows: Sec. 2 lists existing related works. Sec. 3 describes our analysis of DoH. Sec. 4 describes datasets that we created to allow the analysis. Sec. 5 shows the results of the analysis and evaluates our approach. Sec. 6 discusses some interesting observations from our experiments. Finally, Sec. 7 concludes the paper.

2 RELATED WORK

Even though DoH is a very novel technology, there are already some published papers that target various aspects of it. Borgolte et al. [2] provides general discussion about DoH and several areas such as performance, security, and privacy. However, the paper does not analyze the encrypted communication of DoH at the network level.

Ph.D. thesis [26] is focused on DNS and covert channels using DNS. The thesis discusses various characteristics of DNS and possible threats. More specifically, the author shows the feasibility of deep neural networks to detect covert channels. As a special case, DoH is mentioned to be a possible enhancement of known DNS tunnels, and it is listed as a possible future work by the author.

Hjelm et al. [16] by SANS Institute provides a detailed description of DoH service, and by using the Real Intelligence Threat Analytics (RITA) framework, they identify behavioral patterns of DoH. RITA does not analyze the network traffic itself but uses logs provided by the Zeek IDS. It is worth noting that the authors assume that DoH performs a regular behavior pattern represented by autocorrelation. We have tried to use RITA in the way the authors described in their paper with our created datasets, and the results were, unfortunately, abysmal. RITA identified some suspicious HTTPS connections (about 44); however, none of them was DoH, and no real DoH connection was actually detected.

Patsakis et al. [25] focused on DGA and botnets that use DNS as a communication channel with Command and Control servers. The paper also mentions DoH and DoT in the context of existing botnets that use such mechanisms based on encrypted communication. The authors evaluated the use of the Hodrick-Prescott filter with autocorrelation and autoregressive moving average. The paper aimed to analyze several existing datasets of botnet communication that used DoH and DoT. It showed some regular patterns and autocorrelation observable in botnet communication. Compared to this paper, we have analyzed more features, and our experiments were not limited to botnet communication only. Contrary, we focused on recognition of DoH in general, whereas the reason is quite clear — botnets usually have some regularity in their behavior, which is the reason such patterns can be discovered more easily, and DoH can contain lots of other security threats that must be addressed.

Bushart et al. [4] and Siby et al. [28] study identification of encrypted traffic on the Alexa's top websites list[2]. As a component of their feature vector, the authors use sequences of message bursts and gaps, which is a similar principle we used as a part of our tested feature vector (however, we have improved this metric to enhance the performance of the classification). The papers are focused rather on fingerprinting and recognition of websites.

Bushart et al. [4] also try to recognize DoH content. However, the DoH traffic is identified only by known IP addresses of the popular services. Additionally, the paper proposes to mitigate the visibility into the content (i.e., fingerprinting) based on the use of non-standard port and different providers. This is not our case since we target detection of DoH communication regardless of (known) IP addresses and ports.

To our best knowledge, we are not aware of any published paper, which evaluates multiple packet-level information of DoH and HTTPS traffic with the goal to recognize DoH from classic HTTPS with high accuracy. Our experiments evaluate several ML models and possible feature vectors. Additionally, our paper focuses also on distinguishing particular DoH clients (applications) based on the specific behavioral patterns represented by our evaluated feature vector. It is worth noting that Siby et al. [28] claim that they were able to train the DoH client classifier based on TCP packets

[2] https://s3.amazonaws.com/alexa-static/top-1m.csv.zip

length with 100 % accuracy. However, they did not provide any other information and we were unable to reproduce their approach.

Naturally, our results can be used as an improvement for the listed related works, since they usually assume DoH is directly identified only by IP addresses, which is not our case (neither IP addresses nor ports are in use for our classifier).

3 ANALYSIS AND FEATURE SELECTION

Since DoH is quite a new protocol, there are still two significantly different implementations. The RFC [18] compliant definition uses classic DNS Wireformat [23] encapsulated in the HTTPS protocol using the GET, or POST methods. The other approach introduced by Google uses JSON based messages transferred via HTTPS GET.

The majority of DNS providers support both implementations. However, all web browsers, including chrome based ones, and most of other DoH clients are currently using RFC compliant Wireformat messages with HTTPS POST method.

To evaluate the possibility of DoH recognition, we captured traffic produced by several browsers with enabled DoH protocol (see Sec. 4). We consequently filtered the DoH packets by the IP address of the DoH resolver and analyzed them for the protocol implementations. Some of the DoH connections were also decrypted (using exported cryptographic keys) to understand the contents of each packet. We identified patterns and their differences between DoH traffic and regular HTTPS and also between implementations.

Finally, we studied whether ML algorithms can detect the identified DoH communications patterns. Our analysis is based on bidirectional IP Flows extended with per-packet information (PPI). Besides the traditional IP Flow information (such as IP addresses, ports, and the amount of transferred data), we also have lengths of individual packets and their timestamps. The PPI is general enough for creating discriminative features to classify DoH from regular traffic.

The feature selection is one of the most important parts because it affects the accuracy of any ML classifier. By looking into the raw packet data, we noticed several differences from classic HTTPS traffic.

Currently recommended protocol for DoH is HTTPS2. Therefore, regular DoH connections start with a TLS handshake followed by an HTTP2 connection preface. The rest of the communication looks like a classical request-response scheme. However, there are several differences between classical web-browsing. The typical DoH connection parameters compared to other types of HTTP connections are presented in Tab. 1.

According to our observations, a single DNS request and response has at least five packets in DoH. Therefore, we can directly mark a shorter connection as a classical HTTPS. The most significant difference between DoH and classic HTTPS is the duration of the flow. According to our experiments, browsers create a single connection to the DoH server, which is then used for a longer time. During the operation, there might occur some reconnections or a completely new connection to different DoH servers; however, it does not happen very often. The longer connections can also be created by different communication, like file downloading, video streaming, and so on. However, these types of connections tend to transmit much more data in a shorter time than the DoH, ideally in

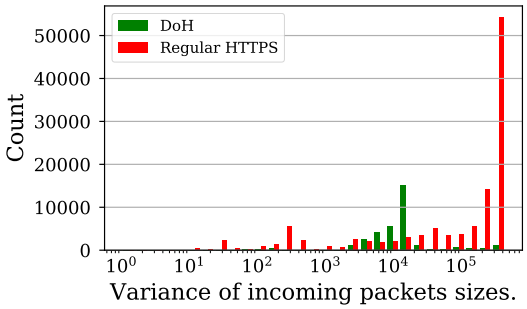

Figure 1: Histogram of variance of incoming packet sizes.

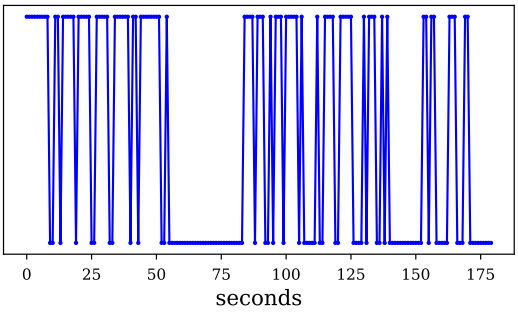

Figure 2: Activity in selected DoH flow record created by the Firefox web browser.

form of a continuous burst of data. This can be clearly seen in Tab. 1, where the connections to Facebook CDN are much shorter, with almost the same amount of transferred data. The row "Web Page" represents an average of typical pages from Alexa's top websites list, i.e., from our captured dataset.

The DoH communication can also be distinguished from regular HTTP by the size of transmitted packets. In Fig. 1, we can see that the DoH variance of response packets sizes is much lower. We can observe the same trend with the sizes of outgoing packets; however, it is less significant, because HTTP requests also tend to have similar sizes.

The specific activity pattern can also reveal the DoH directly implemented in browsers. Fig. 2 shows the example of an activity of one DoH connection, where we can see packet bursts and pauses depending on the user interaction. The number of packets inside bursts, pauses, and their ratio is included in our feature vector.

We detect a packet burst when the interpacket time is shorter than a predefined *burst threshold*. Therefore, we can count the number of packets "within a burst," which all have short interpacket times. Similarly, we detect a pause when the interpacket time is longer than a predefined *pause threshold*. We understand that the packet delays depend on a web server, user connection quality, and many other factors. Therefore, the thresholds must be considered relatively for each connection. We evaluated several HTTPS connections, and we set the *burst threshold* value as the 33.3 % percentile from the inter-packet times of each connection (i.e., a biflow record). The *pause threshold* is set similarly to the 66.6 % percentile.

Table 1: The typical connection parameters of DoH compared to other types of HTTPS communications.

Name	Packets	Bytes	Packets A→B	Packets B→A	Bytes A→B	Bytes B→A	Duration
DoH Firefox	55,312	7,293 kB	27,822	27,490	3,021 B	4,271 B	2088,2 s
Facebook CDN	5,893	7,474 kB	996	4,898	84 kB	7,390 kB	164.95 s
Web Page	233	275 kB	48	185	4,690 B	271 kB	5.75 s

Another identified feature represents the symmetry of the amount of incoming and outgoing data. The DNS responses, especially in DNS wireformat, have almost the same sizes as requests, and communication tends to be balanced (compared to HTTPS). We also split the sequence of packets to thirds and calculate three symmetry metrics separately. HTTPS traffic might by similar at the beginning of the connection, but later on, it becomes strongly asymmetric.

To measure the periodicity of the traffic, we used the autocorrelation metric as another feature. In several previous works, autocorrelation is claimed as crucial for identifying DNS traffic inside covert channels.

In total, we have identified and tested 19 traffic features. After calculating the feature importance with the Gini index, we reduced the feature list to the final 18 features for DoH recognition, and 9 features for DoH client classification. All identified features are clearly outlined in Tab. 2. As it is seen in the table, the most important feature for DoH recognition is the duration of an IP flow. The average inter-packet delay is also essential. Surprisingly, the autocorrelation, regularly used in related work, is quite insignificant.

In the case of DoH client classification, the significantly important feature is the variance of incoming packet sizes. We analyzed, why this feature is separating the clients so well. We found out that Chrome is using EDNS padding feature [21], so wast majority of incoming DoH packets have the same size.

During the time of writing this paper, there is a one-year-old request[3] in the Mozilla bug report platform for implementing EDNS padding; however, the support is still not confirmed.

4 DATASETS

A proper dataset is an essential prerequisite for an excellent ML model. The quality of the model is directly linked with the heterogeneity of information contained in the dataset. However, to the best of our knowledge, there are no publicly available annotated datasets targeted for DoH recognition and DoH client classification. Therefore, we created our own and made it publicly available on the Zenodo platform [30].

Currently, there are only two options for using DoH on an everyday basis. The first option is to enable DoH in the web browser. The second way is to redirect all traditional DNS queries via a central DoH proxy, which translates DNS queries to DoH. We set up both options to produce DoH traffic, and the described scenarios are also shown in the simplified scheme for dataset creation depicted in Fig. 3.

The left side of Fig. 3 presents a capturing of the traffic from DoH enabled web browsers. We installed Google Chrome and Mozilla Firefox into separate virtual machines and controlled them with the Selenium framework, which simulates the user browsing according

[3]https://bugzilla.mozilla.org/show_bug.cgi?id=1543811

Table 2: Importance of the evaluated features. The features with importance typed in bold font were included in a feature vector of the corresponding usage.

Feature name	DoH Importance	Client Importance
duration	**0.239**	**0.169**
minIntrPckDelay	**0.040**	0.001
maxIntrPckDelay	**0.089**	**0.158**
avgIntrPckDelay	**0.221**	0.001
varPktSizeIn	**0.015**	**0.225**
varPktSizeOut	**0.023**	**0.111**
bytesInoutRatio	**0.034**	**0.012**
pktsInoutRatio	**0.011**	**0.159**
avgPktSizeIn	**0.037**	**0.030**
avgPktSizeOut	**0.038**	**0.115**
medianPktSizeIn	**0.045**	0.003
medianPktSizeOut	**0.015**	**0.011**
burstPausesRatio	**0.049**	0.001
pktInBursts	**0.027**	0.001
pktInPauses	**0.063**	0.001
autocorrelation	**0.015**	0.003
symmetry-1thrd	**0.011**	0.001
symmetry-2thrd	0.001	0.001
symmetry-3thrd	**0.010**	0.001

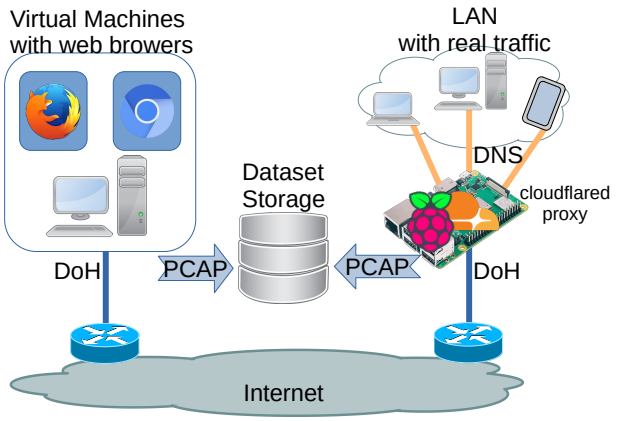

Figure 3: Simplified scheme of environments for datasets creation. On the left, there is a capture of DoH traffic generated directly by web browsers, captured on the machine. On the right, there is a whole LAN with several normally operating network devices and users that generate DNS; DNS is resent by the *cloudflared* proxy using DoH.

Table 3: Overall information about created dataset containing the number of DoH IP Flows and the total number of IP Flows.

DoH client + Version	Size	DoH	Total
Mozilla Firefox 73.01 Lin.	28 GB	698	523,824
Google Chrome 81.0.4044.129 Win.	8 GB	729	154,201
Cloudflared 2020.2.0 Lin.	1.5 GB	32,752	450,879

to the predefined script. The browsers received commands to visit domains taken from the Alexa's top websites list. The capturing was performed on the host by listening on the network interface of the virtual machine. This created dataset contains about 5,000 web-pages visited by Mozilla Firefox, and about 1,000 pages visited by Google Chrome.

The right side of Fig. 3 presents collection of DoH data using a DoH proxy. There are several DoH proxy implementations. We decided to use a DoH client developed by Cloudflare [12] — `cloudflared` — because we believe it is one of the mostly used solutions. We installed the `cloudflared` software into a Raspberry Pi computer. The IP address of the Raspberry was set as the default local DNS resolver for the two independent offices at our university. This DNS resolver was a provided option by local DHCP servers, so any auto-configured device connected to the office network used this resolver by default. The Raspberry was continuously capturing both DNS and DoH traffic created by about up to 20 devices consisting of operating computers, laptops, and smartphones for around three months.

Additionally, we run several scripts for a rapid generation of DNS requests to the Raspberry. These scripts simulated a busy middle-sized network that generates a significant amount of DNS queries. The higher number of DoH queries in a sequence results in entirely different behavioral patterns of the connection. The aim was to add a simulated DoH traffic with as many similar parameters to ordinary HTTPS data transmission as possible. This is crucial to include into the dataset to prevent potential misclassification. The scripts generated DNS queries for domain names taken from Alexa's top websites list. Overall, we captured more than 3,845,000 DoH packets at the proxy.

We also added traffic produced by instant messaging clients (IM) into our dataset. We believe, that IM traffic is the most similar to the DoH since it also follows the request-response scheme with a small amount of transmitted data.

The captured PCAP data were immediately converted into extended IP flows due to user privacy. To convert packets into extended IP Flows, we used a `flow_meter` flow exporter from the NEMEA system [5]. More specifically, we used a particular PStat plugin, which is capable of computing additional packet-level statistics, usually called as PPI feature. The resulting flows were consequently processed by python scripts to add more computed features and annotation (ground truth) labels. The DoH labels were reliably completed according to our knowledge of proxy setup and, for manually generated traffic, according to known IP addresses of the DoH services and target HTTPS servers.

Overall, the created dataset consists of 1,128,904 flows (aggregated into bidirectional records), with around 33,000 of them labeled as DoH. To deal with the resulting imbalance between DoH and regular HTTPS classes, we used the mechanism described in Sec. 5. The size statistics and information about used the software in the dataset are listed in in Tab. 3.

Table 4: Comparison of the overall precision of the evaluated ML algorithms for DoH identification in HTTPS traffic (Recognition column), and identification of particular DoH client (Classification column).

Algorithm Name	Recognition	Classification
5-NN	99.4 %	99.6 %
C4.5	99.4 %	99.9 %
Random Forest	99.5 %	99.9 %
Naive Bayes	96.8 %	95.5 %
Ada-boosted Dec. Tree	99.6 %	99.9 %

5 RESULTS

This section describes the results of our measurements of the ML-based DoH recognition and DoH client classification. For our evaluation, we used the dataset described in Sec. 4, and standard ML tools/libraries implemented in Python: Sci-kit learn library, some methods from NumPy, and *DataFrame* from Pandas.

We applied methods for imbalanced learning since we do not have equally distributed classes in the dataset. Currently, applying *oversampling* and *undersampling* methods is the most common approach for dealing with imbalanced classes (e.g., according to [20]). Specifically, we used SMOTE [7] for oversampling and NearMiss-3 [31] as an undersampling method. The ratio between DoH class and regular HTTPS in our dataset is around 1:13. The SMOTE increases the number of minority (DoH) class samples to ratio 1:5. The undersampling method then reduces the number of majority (regular HTTPS) classes to the final ratio of 1:2.

The dataset balancing methods are applied only on the data given to the training phase of the algorithms (selected using the standard n-Fold cross-validation, see later) since it is usually not recommended to apply it on the testing data.

5.1 Classification Algorithm

In order to classify and recognize DoH traffic, we experimented with five ML algorithms: K-Nearest Neighbours [29] (We use 5-NN in our study), C4.5 Decision Tree [27], Random Forest [17], Naive Bayes [8], and Ada-boosted Decision Tree [14]. These algorithms are commonly used in Networking applications [3].

We used 5-Fold cross-validation to evaluate the precision of each algorithm. The input parameters (also called hyperparameters) of each algorithm were set experimentally by evaluating each parameter separately and observing the precision of results.

The overall performance of the algorithms is very similar across all evaluated algorithms, which shows that our feature vector is very robust and discriminative enough for DoH recognition and classification. The detailed results are written in Tab. 4. The Naive Bayes performs the worst in both tasks; however, its precision is still very high. For the further evaluations, we selected the Ada-Boosted Decision tree, which has the best accuracy.

Table 5: Confusion matrix of DoH recognition from a regular HTTPS traffic. The table contains class accuracy and class recall for both classified classes: DoH and regular HTTPS.

		Ground truth		Class Accuracy
		DoH	HTTPS	
Result	DoH	32,668	81	99.7 %
	HTTPS	1,511	411,791	99.6 %
Class Recall		95.5 %	99.9 %	

Table 6: Confusion matrix of DoH client classification. The column headers are as follows: Ch — Chrome, C — Cloudflared, F — Firefox

		Ground truth			Class Accuracy
		Ch	C	F	
Result	Ch	722	3	4	99.0 %
	C	5	32744	11	99.9 %
	F	2	5	683	98.9 %
Class Recall		99.0 %	99.9 %	98.8 %	

5.2 Detailed evaluation of DoH recognition

Based on the results in Sec. 5.1, we used Ada-boosted Decision Tree with the maximal depth set to 15, and the number of estimators set to 5. The evaluation was done using 5-Fold cross-validation again to obtain the results. The trained model achieved an excellent result of 99.6 % accuracy with an F1 score of 0.996. The detailed results are presented in the form of a confusion matrix shown in Tab. 5.

The Majority of DoH flows in the dataset is originating from Cloudflare's client `cloudflared`. Therefore, we also evaluated the precision of our trained classifier only on the web browser traffic. The classifier achieved the same accuracy and slightly higher F1 score value of 0.997 in this case. This experiment proves that our classifier can precisely identify even the minority classes. We also evaluated whether the accuracy of detection depends on the DoH client that generated the traffic. However, we did not observe any change in the accuracy of the classifier based on the traffic source.

5.3 Detailed evaluation of DoH client classification

The classifier of the particular DoH client was trained using only IP flows representing DoH communication. In practice, this kind of classification makes sense only on the confirmed DoH traffic from the previously described classifier, so the training on the DoH subset of the dataset is feasible.

In this case, the maximal depth of Ada-Boosted decision trees was set to 10 (based on experiments), with the number of estimators set to 5. We also used the 5-Fold cross-validation to obtain the results. The model achieved even higher accuracy — 99.9 %, and F1 score of 0.999. The confusion matrix is shown in Tab. 6.

5.4 Limitations

The proposed ML algorithms achieved excellent results on the created dataset. However, they also have some limitations. The DoH detection and client recognition are only possible on connection with multiple DNS queries. The proposed ML algorithm cannot recognize DoH connection with a single query, because of the similarity with another request/response API. The DoH implementation in browsers and the burst shape of packets are crucial for the correct operability of the algorithm. Therefore it is easy to mask DoH connection from our classifier by creating new connections for each query, which would also significantly increase latency due to TLS handshake.

The attackers can also hide the use of DoH by masking the traffic shape. For example, they might synthetically create more asymmetrical connections — adding padding into DoH query packets. This type of DoH connection would be misclassified due to its similarity to the multimedia stream.

6 DISCUSSION

This section contains several useful information/observations we gained thanks to our experiments and analysis. The following paragraphs conclude several remarks, our lessons learned, and network security and privacy reflections that are, from our perspective, helpful for other researchers interested in this DoH topic.

During the time of our study and experiments, we observed several issues with the existing tools that support DoH service, and with the analyzing tools. The first significant issue we observed was related to the PCAP conversion into the extended IP flows. At first, we tried to use Cisco Joy [1] flow exporter, but we discovered that the resulting IP flows had severe flaws in the number of counted packets. Especially for the long connections, the numbers of packets are not correct, and most of the packets were not included. We encountered this issue mainly during the creation of the Cloudflare part of the dataset. The solution was to find another flow exporter program with the PPI feature, and therefore we used `flow_meter`.

The second crucial observation was related to the Chrome dataset part. Primarily, the resolution of a domain name via DoH was unstable, i.e., sometimes, Chrome unexpectedly switched from DoH resolution to the standard DNS mechanism. This effect complicated our creation of the dataset, but more importantly, it means a privacy risk for users who intentionally enable DoH to hide their activities. Even though the DNS was used at the background, the DoH connection persisted. We assume it contained some keep-alive messages without real data. This creates an illusion of working DoH without any actual effect. This DoH "outage" remained until the restart of the browser, and it appeared in random time after the browser started. We consider this behavior as a bug in the tested browser version.

Contrary, we have to mention that DoH implementations by Cloudflare and Mozilla work perfectly without any outages or any other observable issues. A minor note about the operation of `cloudflared` is about ACK packets: in comparison with Mozilla, the Cloudflare client does not reply to the DoH server regularly to all the data with ACK packets as Mozilla does. This peculiarity slightly breaks the structural patterns of the Cloudflare DoH communication, and it creates a difference in behavior between these two clients. However, DoH communication by Chrome and Mozilla follows the structural pattern of a constant number of packets per one query that we observed before. This leads to our hypothesis

that we can estimate also a number of queries and responses inside a DoH connection, which we plan to study in the future.

Another note is about a difference between a default policy of using DoH applied by Google and Mozilla in their browsers. In the case of Mozilla, DoH is enabled by default for all users in the United States, and it is set to Cloudflare, even though the default DNS resolver in the operating system is different. Google has also enabled DoH by default, but only for users that already use the DNS provider who supports DoH. The resolver from system configuration has a higher priority, and if it does not support DoH, the standard DNS resolution is used.

Observation of the Mozilla and Chrome DoH policies leads us to a question: "How good is the DoH protocol usage in global scope?" This protocol increases users' data privacy, helps to fight against attacks related to a domain name abuse during the resolution process (e.g., DNS hijacking), gives users an ability to bypass oppressive restrictions (e.g., right to free speech violation in some countries) on the content, and mitigates eavesdropping and Man-in-the-Middle attacks.

On the other hand, the use of DoH propagates centralization into the domain name resolution mechanisms, which is the enforcement of trust only to several leading DoH providers, replacing the decentralized DNS principle, where users are able to choose a service provider. Also, DoH brings several security issues, such as bypassing of enterprise policies and complicating a network monitoring process for security tools by hiding DNS data. Furthermore, DoH enhances some types of Command and Control communication by malware that uses domain queries as a way to transmit commands and receive responses. Last but not least, the DoH protocol leads to bypassing local filters based on blacklists containing malicious domain names, and rely on the blocking mechanism of DoH resolvers, which is a potential risk for users when they access unknown destinations.

7 CONCLUSION

DNS over HTTPS is a natural reaction of the engineering community related to IETF to deal with privacy issues of the currently used DNS protocol. The main principle of DNS is the use of encrypted communication channels based on the popular HTTPS ecosystem, in the case of DoH. It is clear, that the encryption hides the content of the users' queries. From the network security perspective, DoH brings several security threats due to limited visibility by existing tools that depend on readable data. Therefore, we have focused on the analysis of encrypted DoH traffic. Our aim was to evaluate what information is potentially available using ML algorithms trained on the prepared datasets.

For this purpose, we captured and published a large dataset consisting of DoH traffic from several most popular tools, i.e., web browsers and a DoH resolver used as a proxy. Our infrastructure helped to create a unique annotated dataset partially using Alexa's top websites list, whereas the dataset was further used to evaluate a feasible feature set.

The main contributions of this paper is the dataset and the method for creation of ML models, that achieved excellent results above 99 % accuracy. Specifically, the first ML classifier is able to recognize DoH communication precisely, and the second classifier provides even more detailed information about the DoH client.

As a result, our classifier can identify a DoH client in the network traffic regardless of IP addresses and ports (which is one of the main differences from the related works). Information about DoH clients on the network is essential for network operators and security analysts, since it may indicate breaking/bypassing the security policies. It is worth noting that DoH has already become a protocol exploited by attackers and malware to hide their activities. Besides, the earlier papers recommend running DoH on non-standard ports with uncommon resolver to avoid DNS fingerprinting. However, such DoH connections "hidden" in this manner still cannot hide from our trained classifier, because it does not rely on IP addresses and ports; therefore, these recommendations are not valid anymore.

During our experiments, preparation of the environment for dataset creation, and testing the tools, we gained some experiences about behavioral patterns, benefits and weaknesses of the existing DoH tools. Therefore, we felt it is highly useful to sum up our "lessons learned" in the Discussion section (Sec. 6).

Possible Future Work

As our future work, we will focus on detection even more details about the contents of DoH. We believe our classifiers can be enhanced to recognize low-level information about an HTTP method that can be used according to DoH standards.

Also, as it was briefly discussed in Sec. 6, according to our observations, we have a hypothesis that it is also possible to estimate the number of queries inside a DoH connection. However, this must be evaluated more thoroughly. In case we will be able to split an already identified DoH connection into particular parts representing each query and response, it also makes sense to experiment with fingerprinting the queries as it is suggested in some related works.

Finally, this paper was focused on DoH only, however, DoT is also a potential candidate to replace DNS. Therefore, it would be interesting to extend our research to a more general TLS connection, which is also a possible transport layer for private DNS resolution.

ACKNOWLEDGMENTS

This work was supported by the European Union's Horizon 2020 research and innovation program under grant agreement No. 833418 and also by the Grant Agency of the CTU in Prague, grant No. SGS20/210/OHK3/3T/20 funded by the MEYS of the Czech Republic.

REFERENCES

[1] Blake Anderson, David McGrew, Philip Perricone, and Bill Hudson. 2019. Joy - A package for capturing and analyzing network flow data and intraflow data. [online] Available: https://github.com/cisco/joy.

[2] Kevin Borgolte, Tithi Chattopadhyay, Nick Feamster, Mihir Kshirsagar, Jordan Holland, Austin Hounsel, and Paul Schmitt. 2019. How DNS over HTTPS is Reshaping Privacy, Performance, and Policy in the Internet Ecosystem. *Performance, and Policy in the Internet Ecosystem (July 27, 2019)* (2019).

[3] Raouf Boutaba, Mohammad A. Salahuddin, Noura Limam, Sara Ayoubi, Nashid Shahriar, Felipe Estrada-Solano, and Oscar M. Caicedo. 2018. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *Journal of Internet Services and Applications* 9, 1 (Jun 2018). https://doi.org/10.1186/s13174-018-0087-2

[4] Jonas Bushart and Christian Rossow. 2019. Padding Ain't Enough: Assessing the Privacy Guarantees of Encrypted DNS. *arXiv preprint arXiv:1907.01317* (2019).

[5] Tomas Cejka and et al. 2016. NEMEA: A framework for network traffic analysis. In *12th International Conference on Network and Service Management (CNSM)*.

[6] Tomas Cejka, Zdenek Rosa, and Hana Kubatova. 2014. Stream-wise detection of surreptitious traffic over DNS. In *2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. IEEE, Athens, Greece, 300–304. https://doi.org/10.1109/CAMAD.2014.7033254

[7] Nitesh Chawla, Kevin Bowyer, Lawrence Hall, and W. Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res. (JAIR)* 16 (01 2002), 321–357. https://doi.org/10.1613/jair.953

[8] C. K. Chow. 1957. An optimum character recognition system using decision functions. *IRE Transactions on Electronic Computers* EC-6, 4 (1957), 247–254.

[9] C Cimpanu. 2019. First-ever malware strain spotted abusing new DoH (DNS over HTTPS) protocol.

[10] C. Cimpanu. 2020. Here's how to enable DoH in each browser, ISPs be damned. https://www.zdnet.com/article/dns-over-https-will-eventually-roll-out-in-all-major-browsers-despite-isp-opposition/.

[11] C. Cimpanu. 2020. Mozilla enables DOH by default for all Firefox users in the US. https://www.zdnet.com/article/mozilla-enables-doh-by-default-for-all-firefox-users-in-the-us/

[12] Cloudflare. 2020. cloudflare/cloudflared. https://github.com/cloudflare/cloudflared

[13] Ben Dickson. 2019. Does Google Chrome's DNS-over-HTTPS (DoH) feature enhance your privacy? https://bdtechtalks.com/2019/12/11/google-chrome-dns-over-https-privacy/

[14] Yoav Freund and Robert E. Schapire. 1996. Experiments with a New Boosting Algorithm. In *ICML*.

[15] D. A. E. Haddon and H. Alkhateeb. 2019. Investigating Data Exfiltration in DNS Over HTTPS Queries. In *2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3)*.

[16] Drew Hjelm. 2019. A New Needle and Haystack: Detecting DNS over HTTPS Usage. (2019).

[17] Tin Kam Ho. 1995. Random Decision Forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1 (ICDAR '95)*. IEEE Computer Society, USA, 278.

[18] Paul E. Hoffman and Patrick McManus. 2018. *DNS Queries over HTTPS (DoH)*. Technical Report 8484. https://doi.org/10.17487/RFC8484

[19] Tommy Jensen. 2020. Windows Insiders can now test DNS over HTTPS. https://techcommunity.microsoft.com/t5/networking-blog/windows-insiders-can-now-test-dns-over-https/ba-p/1381282

[20] Octavio Loyola-González, Milton García-Borroto, Miguel Medina-Pérez, José Francisco Martínez-Trinidad, Jesús Carrasco-Ochoa, and Guillermo De Ita. 2013. An Empirical Study of Oversampling and Undersampling Methods for LCMine an Emerging Pattern Based Classifier. *Lecture Notes in Computer Science* 7914, 264–273. https://doi.org/10.1007/978-3-642-38989-4_27

[21] Alexander Mayrhofer. 2016. The EDNS(0) Padding Option. RFC 7830. https://doi.org/10.17487/RFC7830

[22] P.V. Mockapetris. 1987. Domain names - concepts and facilities. RFC 1034 (Internet Standard). , 55 pages. https://doi.org/10.17487/RFC1034

[23] Paul Mockapetris. 1987. *Domain names - implementation and specification*. Technical Report 1035. https://doi.org/10.17487/RFC1035

[24] Mozilla Foundation. 2020. Firefox DNS-over-HTTPS. https://support.mozilla.org/en-US/kb/firefox-dns-over-https

[25] Constantinos Patsakis, Fran Casino, and Vasilios Katos. 2020. Encrypted and covert DNS queries for botnets: Challenges and countermeasures. *Computers & Security* 88 (2020), 101614.

[26] Tomás Antonio Peña. 2020. *A Deep Learning Approach to Detecting Covert Channels in the Domain Name System*. Ph.D. Dissertation. Capitol Technology University.

[27] J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[28] Sandra Siby, Marc Juarez, Claudia Diaz, Narseo Vallina-Rodriguez, and Carmela Troncoso. 2019. Encrypted DNS –> Privacy? A Traffic Analysis Perspective. arXiv:cs.CR/1906.09682

[29] Craig Stanfill and David Waltz. 1986. Toward Memory-Based Reasoning. *Commun. ACM* 29, 12 (Dec. 1986), 1213–1228. https://doi.org/10.1145/7902.7906

[30] Dmitrii Vekshin, Karel Hynek, and Tomas Cejka. 2020. Dataset used for detecting DNS over HTTPS by Machine Learning. https://doi.org/10.5281/zenodo.3818004

[31] J. Zhang and I. Mani. 2003. KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction. In *Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets*.

[32] Tomáš Čejka, Radoslav Bodó, and Hana Kubátová. 2015. Nemea: Searching for Botnet Footprints. In *The 3th Prague Embedded Systems Workshop*. Roztoky u Prahy, Czech Republic. https://www.liberouter.org/wp-content/uploads/2015/07/pesw2015-nemea-botnet.pdf