Title

Insights into DoH: Traffic Classification for DNS over HTTPS in an Encrypted Network

ISBN

9798400701658

Authors

Wala, Fatema Bannat
Campbell, Scott
Kiran, Mariam

Publication Date

2023-07-28

DOI

10.1145/3589012.3594895

Peer reviewed

Insights into DoH: Traffic Classification for DNS over HTTPS in an Encrypted Network

Fatema Bannat Wala
fatemabw@es.net
Lawrence Berkeley National Lab
Berkeley, California, USA

Scott Campbell
scottc@es.net
Lawrence Berkeley National Lab
Berkeley, California, USA

Mariam Kiran
mkiran@es.net
Lawrence Berkeley National Lab
Berkeley, California, USA

ABSTRACT

In the past few years there has been a growing desire to provide more built in functionality to protect user communications from eavesdropping. An example of this is DNS over HTTPS (DoH) which can be used to protect user privacy, confidentiality and against spoofing attacks. Since its first popularity in 2018 as used in browsers, there is much further study to test the effectiveness of DoH in protection schemes and whether it is possible to detect the protocol over the web. Detecting DoH traffic among normal web traffic is also a major challenge for network admins to allow filtering of malicious traffic flows. In this paper, we investigate machine learning classification to study the detection of DoH traffic and further analyze the key feature characteristics in the protocol behavior to help researchers build credibility in the DoH protocol detection. Our study reveals key features and statistical relationships among DoH test runs on the Alexa-recommended 100 most-used websites using three different DoH servers, showing up to 98% test accuracy in our built classifier.

CCS CONCEPTS

• **Security and privacy** → *Domain-specific security and privacy architectures.*

KEYWORDS

Privacy, Encrypted DNS, DNS over HTTPS, statistical analysis, network protocol

ACM Reference Format:
Fatema Bannat Wala, Scott Campbell, and Mariam Kiran. 2023. Insights into DoH: Traffic Classification for DNS over HTTPS in an Encrypted Network. In *Proceedings of the 2023 Systems and Network Telemetry and Analytics (SNTA '23), June 20, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3589012.3594895

1 INTRODUCTION

DNS has been a core component of the internet since the mid 1980's [24], making it one of the oldest protocols in use. DNS over HTTPS (DoH) protects user privacy by providing both confidentiality between the client and the first recursive resolver and integrity protection against query tampering via man-in-the-middle attacks (MITM). It does this by encrypting the DNS query from the client using the HTTPS protocol. DoH was first tested in a few popular browsers in March 2018, but since then research on the ability to detect the protocol within regular web traffic traces has been somewhat limited.

There have been several improvements around protecting the security and privacy of DNS users. DNSSEC is one example of the efforts towards preserving the DNS record integrity by using Zone signing and digital signatures so that lookups can be verify-able by the end user [6]. DNS over TLS (DoT), DNS over HTTPS (DoH), and, more recently, DNS over QUIC (DoQ) are newer efforts to tackle user privacy and confidentiality in DNS data in transit. DoT and DoH provide data encryption in transit using the TLS and HTTPS protocols, respectively. In spite of the outward similarities, there are quite a few fundamental differences between DoT and DoH. For example, DoT uses dedicated port 853 for application traffic [14], so can easily be detected based on that. On the other hand, DoH uses TCP port 443 (the default HTTPS application port) to send and receive encrypted DNS. For the well-known DoH resolvers, this traffic can still be identified by looking at the destination IPs of popular public DoH resolvers. The part where it gets tricky is when other DNS resolvers start providing DoH service with unique IPs not popularly known to the public. At this point, the problem of recognizing DoH data in normal web traffic gets difficult.

There are many reasons why it is important to distinguish DoH traffic from normal web traffic. Firstly, one characteristic of several DoH clients is the ability of an application to use its own DoH instance without alerting the end user of the application. As more and more application developers start using DoH, we could see more traffic to dedicated DoH servers rather than well-known public servers. Hence for an attacker, this is an easy way to hide an attack among normal web traffic while controlling destination activity of the application. This is important since most organizations allow outbound web traffic from their networks. Secondly, for a network where controlling DNS resolution is critical, firewalls and RPZs would not be able to monitor activity since DNS traffic would no longer be controlled using the infrastructure-provided DNS servers. Furthermore, as seen in DoH being maliciously used to conduct attack [2], here, the DoH bypasses the OS and is directly operated at the app layer. Every application can have a different DoH server, and a user running multiple DoH-enabled applications can have multiple connections to different DoH servers based on what DoH server each application is configured to use. Because of this, it is important to have a technique in place to determine if DoH traffic is in the network, to know when users start using DoH, and which servers are being used for name resolution. Ideally organizations provide their own DoH service from their DNS infrastructure and

SNTA '23, June 20, 2023, Orlando, FL, USA

ACM ISBN 979-8-4007-0165-8/23/06…$15.00
https://doi.org/10.1145/3589012.3594895

direct users to use their DoH service, blocking/monitoring any DoH traffic outside their network. Even for this to work, one needs to be able to determine DoH traffic in the normal web traffic to block or take any action.

The industry has recognized the importance of advanced intelligent decision-making or machine learning (ML) to help improve network application performance and management [19]. Machine learning algorithms can be used to predict network behavior such as 'which path selection, capacity or QoS change will cause what result or event X with what probability P'. Detecting anomalies can cut down costs and time spent finding impaired segments or misbehaving devices in network infrastructures. Classification techniques here, supervised and unsupervised classification, can encompass ML approaches that use labeled data to train algorithms that can classify data into specific classes. Unsupervised ML, on the other hand, works with unlabeled data sets to learn underlying patterns, perform dimensional reduction, or learn latent relationships among the data. For a specific example, unsupervised classification can recognize good flow performance or security anomalies [12]. In networking, it is often difficult to find labeled data sets, as performance logs are rarely labeled except in major event scenarios.

In this paper we look at two different ways of identifying DoH traffic - a light weight statistical method as well as a more full featured ML schema with the idea that they could be used together. We see a clear need to identify DoH traffic, given the future landscape of DoH use is going to be across a broad range of application providers, making the well-known DoH server lists non-exhaustive. Thus relying on the IP-based detection becomes challenging and will become obsolete. Here we address the challenge of DoH traffic detection using statistical analysis and machine learning based approaches to identify unseen patterns which help build better detectors. The primary goal is to detect DoH traffic from normal traffic and distinguish its behavioral features. The paper is divided as follows: Section 2 describes the background and related work. Section 3 describes our analysis techniques, data collection, and testing of various machine learning techniques to build classifiers. Sections 4 and 5 present an analysis and discussion of the approach, and Section 6 concludes the paper.

2 BACKGROUND

2.1 DNS Privacy Concepts

The DNS protocol was designed to carry out IP address resolution in a fast and reliable way. Performance was a highly motivating force in using UDP as the primary transport protocol. When more reliability in some of the DNS operations was needed, like DNS zone transfers, or when the size of the request/response was greater than a single UDP packet then TCP was used as transport protocol. This allowed some assurance that records shouldn't get lost during zone transfers between two DNS servers and that both can function reliably with the resource records in sync. Privacy and security were not given a high priority while designing the DNS protocol, and being one of the oldest protocol and being around for a while, it has attracted various adversaries to abuse DNS for different purposes [4] [9] [15] [29]. DNS messages and responses were not encrypted in transit, so anyone observing the wire could see these messages in clear text and infer a lot of useful information about the client and what they were doing even if the application protocol was itself encrypted (like HTTPS). Examples of this metadata include insights about what type of services, applications, and software the client is using. This information can then be used by the adversary to craft attacks on the clients.

Because of concerns associated with user privacy, and the need to protect against information leakage in form of clear text, several different DNS privacy and confidentiality standards have been proposed and developed. The three main standards that help protect DNS confidentiality and integrity are DNS Security Extensions (DNSSEC) protocol standard, DNS over TLS (DoT), and DNS over HTTPS (DoH).

2.2 DNS Applications and Standards

DNSSEC provides authentication using digital signatures for DNS records which can be verified by the client to prove the authenticity and integrity of a DNS resource record. This allows protection from the records being tampered with in transit. It uses public key cryptography to digitally sign the DNS resource records that can be verified by the client. This is grounded in the idea that only the real owner of the DNS record, i.e. the authoritative DNS server, can hold the private key used to sign the record that can be verified by the corresponding public key provided to the client. DNSSEC was first introduced in 2005, and the RFC 4033 [5] explains the definition and implementation requirements of the DNSSEC standard. It provides integrity in the DNS ecosystem.

DNS over TLS, DoT, was introduced as the standard in 2016 when IETF published the DoT RFC 7858 [16]. This RFC defines the protocol standards for using encrypted DNS queries and responses between the client and the server, hence protecting the first hop of the DNS messages, which is from the client to the first DNS recursive resolver the client connects to. This protects the privacy of clients and DNS records queried. DoT runs on TCP port 853 for its own server application port, which makes it uniquely identifiable over the network. [11].

DNS over HTTPS, DoH is another standard that was proposed to protect the privacy of the users by encrypting the DNS messages and responses in transit. Just like DoT, DoH would encrypt the communication between the DNS client and the first recursive resolver that it connects to. However, one major difference between DoT and DoH is, DoH uses HTTPS as the underlining application to encrypt communication in transit, and hence it runs on the web port 443. This introduces the challenge of mixed DNS queries with web traffic, although it provides security through obscurity, as many would argue. After its standardization by IETF in 2018 in RFC 8484 [13], many public recursive resolvers have already started experimenting with DoH. Mozilla, the creator of the Firefox browser, announced in early 2020 [3] that DoH would be enabled by default in the firefox browser for all it's US-based users. Google also has been experimenting with enabling DoH in its Chrome browser offering and making it available for all Chrome browser users since 2019 [1].

3 OUR METHODOLOGY

For our work on DoH recognition, we build a testbed to capture traffic with an enabled browser to several highly used websites. Our

intent is to model the users of an organization using DoH enabled web browsers to a number of common web sites. Details follow. This traffic appears to be normal web traffic, as DoH uses the same port as HTTPS (port 443). Here we also assume that analysts can monitor the organization's perimeter traffic. There is a wide range of security tools, open-source, free, and vendor offered, used as NSM tools to monitor their network.

Once the analysts have visibility into the network traffic, the proposed approaches can be applied in practice to deduce with high confidence whether a particular traffic stream could be DoH or not. Since our traffic is encrypted and we do not decrypt it and use the full pcap captures within the scope of preserving users' privacy. Feature selection is detailed in each of the individual analysis sections.

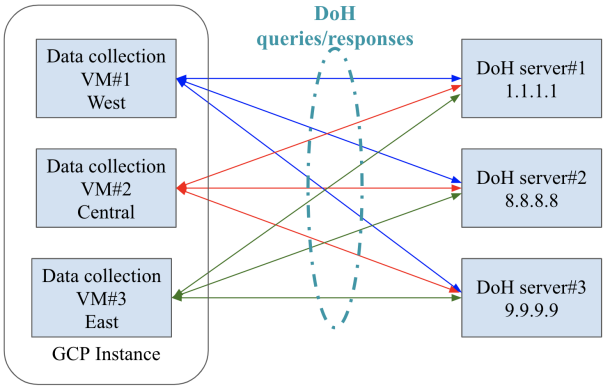


Figure 1: Data collection using GCP instances.

3.1 Testbed Setup and Data Collection

For the data collection setup, we chose Google Cloud Platform (GCP) to run our virtual machine instances in three different geographic locations - west, central, and east zones. This was done so that we can cover a broader range of traffic and see if the DNS queries and responses are affected by the geographic locations of the clients. All three instances ran Ubuntu 20.04 LTS as the operating system. The website visit was executed by the Firefox browser, and since we had a list of websites to be visited and packets to be collected from each visit, we automated the task by using Python scripting language. Using selenium, the Firefox browser was invoked in the headless mode with DoH settings pointing to the DoH server of interest. Regarding the tested DoH resolvers, we chose three well-known DoH resolvers - Quad9 (9.9.9.9), Google (8.8.8.8), and Cloudflare (1.1.1.1).

For building a list of websites to test the DNS resolution, we picked the top 100 websites from the popular Alexa Top Sites list (https://alexa.amazon.com/). At the time of this research, Alexa Top Sites service is available online (the website list is available). The top 1m site list is updated daily. The initial set of websites was the top 100 sites, but a couple of websites were either moved permanently or broken. We excluded those from our list, which explains why the resulting list contains 98 instead of 100 websites.

Our DNS query iterates through the list of websites, invokes the Firebox browser, and captures the traffic using tcpdump. The tcpdump process was started before each website was fetched and stopped after 15 seconds of the website getting request, assuming this is enough time for a website to load. We used two tcpdump processes in parallel- the first one using the tcpdump filter for the DoH server's IP address to capture DoH-only traffic, and the second one using tcpdump filter excluding any traffic that has the DoH server's IP address in it, we called that pcap set as "other" traffic, which most likely would be web traffic. So for every website visit, we ended up with two pcaps - DoH-only (DoH) and non-DoH (other) pcaps. We used this to label the dataset as DoH and non-DoH for building the classifiers. The process was repeated for each DoH server from each VM instance. Hence, for each VM, we had 196 pcaps (98 DoH and 98 non-DoH/Other) from each DoH server, i.e total of 589 (3x196) pcaps from a run across three DoH servers. The diagram of the data collection is shown in Figure 1.

The pcap collection for our first run was on 01/27/2022, in which we only collected pcaps from the Cloudflare server (1.1.1.1). We ran the second pcap collection on 06/02/2022, and this time, we collected pcap across all three DoH servers. Similarly, our third and last pcap collection happened on 07/08/2022, and again we collected pcaps across all three DoH servers from each VM. The reason for collecting the pcaps on the spread timeline was to see if the changes in either the website hosting or the server's software upgrade impact the DNS queries and responses. Also, the time period between the runs was randomly selecting, waiting for few weeks before we run next round of pcap collection. The details on the pcap files are shown in Table 1.

3.2 Feature Selection and Padding

We selected features from the pcap file, removing ip addresses and features that were specific to the traffic. We ended up with 17 features shown in Table 2. Requirements for feature selection are driven to a large degree by the tool(s) used to do the analysis. We focus on a small number of variables to create an easy to understand mental model based on patterns that we see. For the ML analysis we have the ability to digest a richer variety of inputs without the same degree of difficulty.

For the statistical analysis we see a stream of packets with features described by Figure 4. Here the features are immutable characteristics of the traffic stream. s_i represents the packet size i, and t_{ij} represents the time delay between packets i and j. Since the inter-packet delay can be effected by a number of measurement, environmental, and analysis factors we chose to focus on *packet size* in the detection framework.

3.3 Two-way Traffic and Padding

As part of the original design of DoH, there is a discussion about the use of padding to provide some protections against the passive monitoring of traffic [13]. There are some suggestions about padding lengths in [17] as well as suggestions as to what it should be set in [18].

With this in mind, we can compare the distribution of packet sizes for DoH and non-DoH traffic for both the client and server sides of the conversation. Figures 2 and 3 show the distribution of

	Cloudflare (1.1.1.1)	Google (8.8.8.8)	Quad9 (9.9.9.9)	Total
01-27-2022	DoH-98, other-98	-	-	196
06-02-2022	DoH-98, other-98	DoH-98, other-98	DoH-98, other-98	589
07-18-2022	DoH-98, other-98	DoH-98, other-98	DoH-98, other-98	589

Table 1: Total datasets from each VM.

Feature	PC-1	PC-2
ip.len	0.000047	-0.000342
ip.flags.df	0.000000	0.000000
ip.flags.mf	0.000000	0.000000
ip.ttl	0.000000	0.000000
ip.proto	0.000000	0.000000
tcp.window size	0.557778	0.821999
tcp.ack	0.826302	-0.536917
tcp.seq	0.078161	-0.189836
tcp.len	0.000057	-0.000346
tcp.stream	0.000000	0.000000
tcp.urgent pointer	0.000000	0.000000
tcp.analysis.ack rtt	0.000000	0.000000
frame.time relative	0.000000	0.000000
frame.time delta	0.000000	0.000000
tcp.time relative	0.000000	0.000000
tcp.time delta	0.000000	0.000000

Table 2: Features and their PCA Importance. Values less than 1e-06 are rounded to zero.

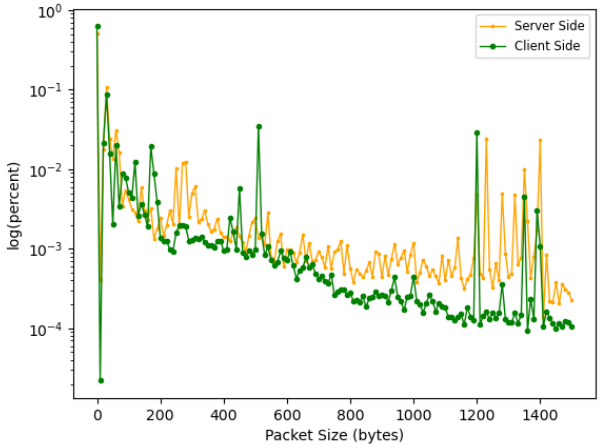


Figure 2: Packet size distribution for non-DoH traffic. Y-axis is log_{10}, and all values are normalized percents and rounded to the nearest 10-byte value.

packet sizes. The most critical observation about non-DoH traffic is that the size distribution is fairly well spread across the span of values from 0-1500 bytes. Server values are systematically larger than clients, but this is expected from the asymmetric characteristics of typical web browsing. Byte values between 0 and 1500 are well represented in the histogram.

For the server-side DoH traffic, we see two different things. First, the distribution of packet sizes sits on a small number of points. *How* these points are distributed is broken out into two very different shapes. Cloudflare (1.1.1.1) and Google (8.8.8.8) have strong bimodal distributions, with peaks around 0-30 bytes and 490-530 bytes, suggesting some sort of padding. For Quad9 (9.9.9.9), we see a peak at zero as well as a (fairly) normal distribution with a mean of around 170 bytes. In both cases, the distribution of size values is sufficiently different from 'typical' non-DoH traffic that further examination is warranted. We use the distinction between these distributions as well as the absence of large packets in the DoH flows, as a core justification in building the analysis.

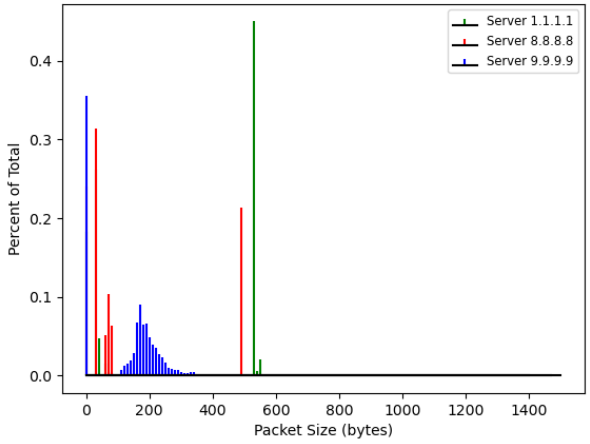


Figure 3: Packet size distribution for DoH server-side traffic. All values are normalized percent and rounded to the nearest 10-byte value. Note Quad9 lack of padding.

4 CLASSIFICATION ALGORITHM

Supervised techniques use some knowledge about data sets (such as labeled data) to group data into clusters. Unsupervised techniques start without any knowledge of data sets, identify features, and cluster similar records into unique sets. Algorithms that use a mix of techniques fall under semi-supervised area.

4.1 Training and Test Datasets

From the flow data, we filter the data in the direction of source-destination and acks from destination-source. We use the data from

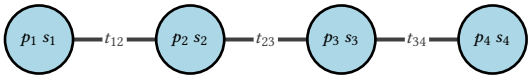


Figure 4: Packets and their size and inter-arrival times.

06-02-2022 and 07-18-2022 as training data and test it with 01-27-2022 data set.

4.2 Dimension Reduction for Classification

From this data we need to identify flows into two clusters - DoH and Other (non-DoH). The aim is to produce two distinct clusters and to find patterns, relationships, and similarities across them. Many ML techniques involve digesting large data sets with many features to produce meaningful results. However, where these may give good results in some cases, this makes the training extremely slow and difficult to find a good solution. Reducing the dimension can help select certain important features compressing the data but also losing some information along the way. There are many ways to reduce dimension, we discuss a number of them here:

4.2.1 Principal Component Analysis (PCA). This works by identifying the hyperplanes which lie close to the data sets, explaining the maximum amount of variations called the principal components. The axis orthogonal to the first plane becomes the second principal component, and so on.

4.2.2 T-distributed stochastic neighbor embedding (tSNE). takes high dimensional data and reduces it to low dimension by retaining as much information as possible. It is often used as a quick visualization technique.

4.2.3 UMAP (Uniform Manifold Approximation and Projection). This is a nonlinear dimension reduction method that uses manifold learning and dimension reduction to find relative proximity in clusters. Out of all of these, tSNE has the most processing time.

5 RESULTS AND VALIDATION

We conducted all classification techniques PCA, tSNE, and UMAP. We found that PCA was able to give us optimum classification results with the explained variation of 73% and 26% in just the first two principal components accounting for 99% of the data analysis. Looking into the important features listed in Table 2, we see that ip.len or duration of the connection, TCP window size, ACK, and duration are the most important features which is able to classify the data. This is an important result because we do not depend on the IP addresses and destinations to find the DNS traffic. Results are shown in Figure 5. Figure 6 shows the exact tSNE visualization in 3D. Figure 7 shows the UMAP distributions. The graphs show that PCA is able to distinguish most of the DoH traffic in one class rather than TSNE and UMAP. This is because even with simple dimension reduction of certain features, we see distinct DoH behavior. Nearly identical results were found for the East and West server data so for reasons of space conservation we will not display these.

5.1 Classification Accuracy

We used PCA to build a classifier for the DoH data and, using the training and test data tested the accuracy. We calculate classification accuracy as the number of correct predictions/number of total predictions. Our results show the PCA was able to achieve 99% accuracy for correctly predicting whether traffic is DoH or not.

Table 3: Consistency Between East/Cent/West Data Centers Per Provider: Pre Overfit Correction

Location	Provider	Mean	StdDeviation
EAST	Cloudflare	0.9168	0.0313
EAST	Google	0.9497	0.0193
EAST	Quad9	0.8727	0.0492
CENT	Cloudflare	0.9328	0.0225
CENT	Google	0.9641	0.0142
CENT	Quad9	0.8583	0.0539
WEST	Cloudflare	0.9141	0.0377
WEST	Google	0.9567	0.0190
WEST	Quad9	0.8758	0.0531

Table 4: Consistency Between Providers

	Cloudflare	Google	Quad9
Cloudflare	1.0000	0.8711	0.6290
Google	0.8711	1.0000	0.5430
Quad9	0.6290	0.5430	1.0000

6 DISCUSSION

There are a number of tests we run on the data set to examine their use in developing useful signatures. Table 3 looks at internal consistency - location and single providers data, to see if things are internally consistent.

6.1 Overfitting

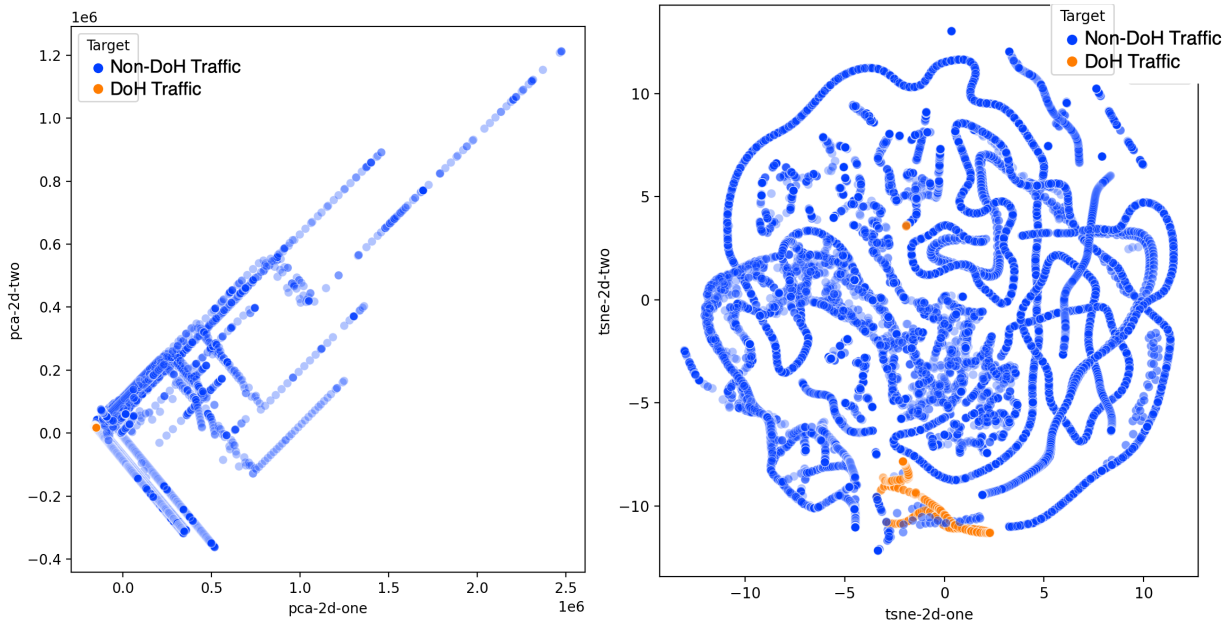
Results from Table 3 suggest a reasonably close match for measured characteristics. This means that individual data sets are sufficiently consistent that they can be used for signature generation. Note these numbers were generated before possible overfitting was addressed.

The next thing to examine is how well a model generated for one provider works against data from the other providers. Results from this are seen in Table 4. Note that Quad9 is not consistent with the other two providers and has a much larger Jensen–Shannon distance.

Assuming that aggregate of each provider is sufficiently similar to one another, we aggregate the data and compare the providers against one another in terms of similarity of distribution.

One thing to keep in mind while creating statistical models of phenomena is that after a critical point it is possible that additional data begins to stressing small differences more than the main structure resulting in overfitting. Figure 8 indicates an ideal value around 14. The values shown are for a test network being compared against itself.

After taking overfitting into consideration, the final results can be seen in Table 5. From this we see that the two padded service providers (Cloudflare and Google) have nearly perfect detection for themselves and one another. Given that Quad9 does not pad their



(a) PCA on Central VM data.

(b) tSNE on Central VM data.

Figure 5: Central VM data.

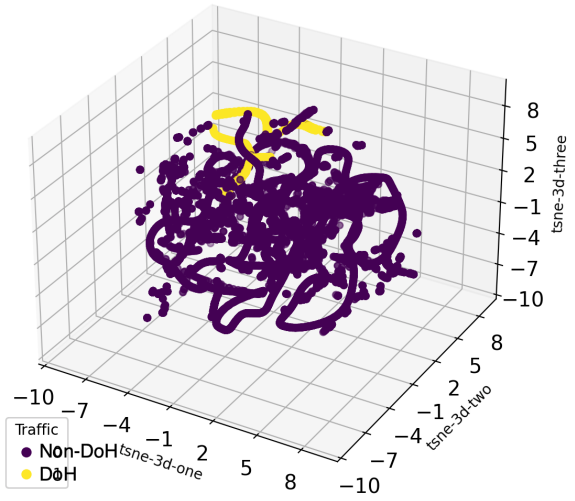


Figure 6: tSNE in 3d on Central VM data.

server side communications, the poor performance seen by that model is not surprising.

6.2 Sensitivity Analysis

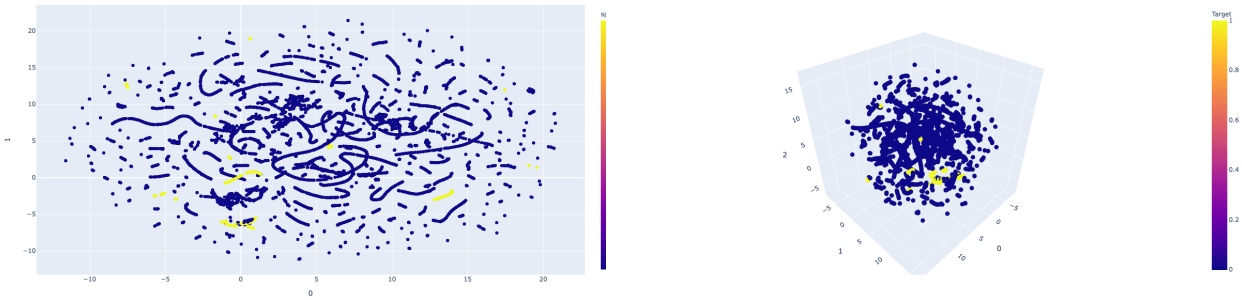
Figure 9 shows the histogram comparison for different services, data centers, and sample times. Converting from a 2D x_n,y_m coordinate system of size nxm to a simple histogram index requires a simple translation: $h_i = m(x-1) + (y-1)$. The downward diagonal of low values is a value being compared to itself, which is expected. The log graph here is extremely sensitive to differences, so Cloudflare and Google are not showing as similar compared to a non-logarithmic scale but are closer than Quad9, which is almost unrelated to the other two service providers.

Table 5: True Positive Detection Rates for DoH Traffic.

Provider1	Provider2	True Positive Percent	Count
Cloudflare	Cloudflare	100.00	686
Cloudflare	Google	99.66	589
Cloudflare	Quad9	97.45	589
Google	Cloudflare	100.00	294
Google	Google	100.00	589
Google	Quad9	97.62	589
Quad9	Cloudflare	00.00	294
Quad9	Google	00.00	294
Quad9	Quad9	47.11	589

6.3 Limitations and Future Work

The ability to identify DoH traffic based on network traffic characteristics can be seen as a fundamental technique to classify the traffic efficiently without too much effort and fine-tuning. This work is a first pass at analyzing encrypted DoH traffic to mimic the resources available to network analysts/engineers at a given point and create a near-realistic user environment of work. However, there are several limitations to this work: 1) Intentional padding of



(a) Umap 2d.

(b) Umap 3d.

Figure 7: Umap on Central VM Data (0-Other, 1-DoH).

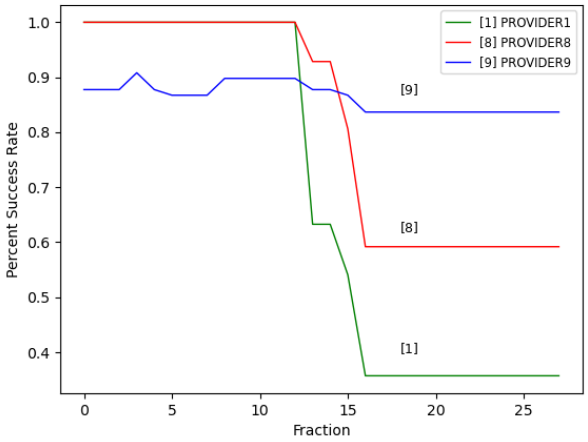


Figure 8: Absolute error count for various sample sizes used in training data.

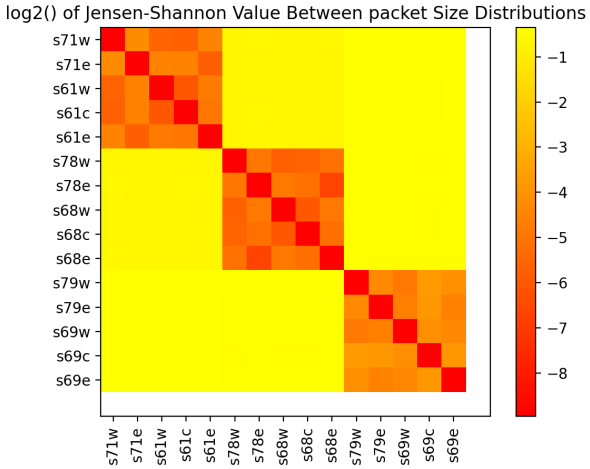


Figure 9: $log_2()$ results of comparing differing services, data centers, and times. Key for axis, aNMb: a: server-side, N:month, M:service provider, b: west/east/central

the queries and responses was not considered. 2) Our experiments are not conducted in a true open-world scenario. 3) Our experiments were based on the assumption that the network traffic analysts will have visibility into their network traffic going in and out of the network. We discuss these limitations and potential future work in detail below.

Several studies [8], [25] have explored the effectiveness of DNS message padding to obfuscate from the encrypted traffic-based analysis. By itself, padding is not a guaranteed way to protect the privacy of DNS. To explore a generalized scenario of users accessing DoH servers without prior knowledge of configured client-side padding, we used very basic headless browser invocations with default DoH settings for the browser. This helps us to test the most common scenario of client queries. However, it seems that by default, the queries were also padded by the browser. As Firefox does not provide settings to configure padding, we did not perform different tests for different padding configurations. We still believe that even with different padding configurations, it wouldn't be hard to apply statistical analysis techniques to detect patterns of DoH traffic.

We conducted our experiments in a test setup by using a cloud-based virtual machine running a basic headless browser with DoH enabled, configured to use major DoH servers. Since in a real world scenario a lot of other internet activity would be happening, we should try to mimic this more complex system and see how it effects detection efficiency. Different web clients might also be using different DoH servers with their own padding configurations, which would result in mixed DoH traffic and complicated detection.

For this study, we used headless browser invocations, which might not be the way clients would be normally using browsers to surf the web, and we expect the results to be similar to using a normal browser with a GUI. Furthermore, another advanced technique to consider would be building the ML classifiers. We may further improve the results by using other machine learning classifiers and

using tuning hyper-parameters as an extension of this study and future work.

7 RELATED WORK

Even though applying machine learning to the DNS protocol is still in its infancy, there has been some work on using machine learning to find DNS traffic characteristics. However, several of these papers do not analyze encrypted data coming from the network level. Nguyen et al [22] developed a DoH detection system using deep learning methods such as attention methods and transformers. However, the results shown did not discuss which features were important behavior characteristics that could help understand the behavior of DoH. Similarly, Vekshin et al. [28] compared different classifiers together showing upto a 99% prediction accuracy, but similarly do not show the important features in their data sets.

Encrypted DNS technologies are still considered a novice in the security industry, specifically, DoH, which peaked the attention of security practitioners when some of the major browser clients [3] [1] enabled DoH by default a couple of years ago, which raised the concerns of adversaries hiding malicious traffic and clients/users bypassing local security measures employed by the service providers. Many researchers came up with techniques to analyze and classify traffic associated with DoH by exploring the data available in encrypted communication.

For example, Banadaki [7] explored different techniques in machine learning to detect malicious DNS over HTTPS traffic versus benign DoH traffic by building classifiers based on six different ML algorithms. They used data from the CIRA-CIC-DoHBrw-2020 dataset and evaluated the accuracy of various algorithms, concluding that LGBM and XGBoost algorithms outperform other algorithms that were explored. However, they didn't mention how practical the solution would be to deploy in practice and how long it would take to create a baseline for the ML classifiers for production traffic. For most of the algorithms, they found source IP and Destination IP to be key features for classifying DoH traffic from non-DoH traffic, which we wonder is because the dataset might not have enough randomization of IPs, as one would wonder how effective those techniques would be when IPv6 space is explored or will it hold the same effectiveness in dynamic IP settings of the real world scenarios.

Another similar study by Singh et al. [26] also explored various ML-based algorithms to classify malicious DoH traffic versus benign and used the same CIRA-CIC-DoHBrw-2020 dataset to explore these techniques. The major difference from the previous piece of study is the use of different algorithms - i) Naive Bayes (NB), ii) Logistic Regression (LR), iii) Random Forest (RF), (iv) K-Nearest Neighbor (KNN), and (v) Gradient Boosting (GB), and their results showed RF and GB classifiers are better choices for the said problem, making an argument that ML-based techniques would be a way to go in detecting malicious DoH traffic. Again, the study has the same limitations as the previously discussed one.

All of these recent studies [23] [20] [21] [27] show how Machine learning or Deep learning models can be used to detect malicious DoH traffic using 2 or 3-layer approaches. [20] built a hierarchical machine learning classification system to identify malicious DNS tunnel tools used in practice, including dns2tcp, dnscat2, iodine, dnstt, tcp-over-dns, and tuns using Gradient boosting decision tree algorithm on CIRA-CIC-DoHBrw-2020 dataset. In [21], authors focus on the importance of statistical analysis of features that singularize malicious traffic from benign traffic and conclude that it is possible to differentiate traffic based on certain statistical parameters, which is close to what we are trying to achieve in our study to classify DoH in encrypted traffic.

Two studies that are closest to what we are trying to achieve in this piece of work are [28] [10]. Both of these explore different ways of identifying DoH traffic in encrypted communication by using various ML algorithms and selecting a limited set of features. However, they also have similar limitations as associated with previously discussed ML-based approaches for identifying DoH tunnels and malicious DoH traffic vs benign. Our approach to classifying DoH versus web/non-DoH traffic is based on a statistical analysis of the network traffic collected as part of passive network traffic monitoring, which mimics most real-world production networks. And the fact that we generated our own dataset and collected traces from different geographic regions over a period of time gives us an edge on the dataset that would be closer to traffic seen in practice. Furthermore, our study differs in another unique way as we look at the statistical pattern of packet data sequences that constitute a complete DoH transaction.

8 CONCLUSION

In this study, we conducted an investigation to get insights into DoH traffic by classifying the traffic in an encrypted network using novel machine learning and statistical analysis method. We were able to demonstrate that by using simple PCA classification and basic data visualization techniques, one can get a baseline classification of DoH traffic which can be further extended to build high-class classifiers that would be able to detect DoH traffic in real-world scenarios. Our results achieved up to 99% accuracy. These results are similar to the results achieved by Vekshin et al. [28] and can be analyzed further with DoH traffic from other setups as well to build a universal DoH classifier.

We also found that padding solely doesn't play a major role in protecting the privacy of DNS and doesn't hamper the classification. Both - DNS requests and responses have very distinct patterns in encrypted traffic that help them to stand out and are easy to classify. Based on visibility into the traffic, either request from the client or responses to the client would be enough to identify if DoH traffic is seen and whether a client is using an external DoH server. We expect that our research will represent a baseline for DoH classification and, more importantly, it will help in the future to develop lightweight and less intensive effective DoH classifiers that can be deployed in production networks and will give network defenders the ability to still get useful traffic insights even when the internet goes completely dark.

REFERENCES

[1] 2019. Google to run DoH experiment in Chrome. https://www.zdnet.com/article/google-to-run-dns-over-https-doh-experiment-in-chrome/.
[2] 2019. New Godlua Backdoor Found Abusing DNS Over HTTPS (DoH) Protocol. https://www.trendmicro.com/vinfo/es/security/news/cybercrime-and-digital-threats/new-godlua-backdoor-found-abusing-dns-over-https-doh-protocol.
[3] 2020. Mozilla enables DoH by Default. https://www.zdnet.com/article/mozilla-enables-doh-by-default-for-all-firefox-users-in-the-us/.

[4] Kamal Alieyan, Mohammed M Kadhum, Mohammed Anbar, Shafiq Ul Rehman, and Naser KA Alajmi. 2016. An overview of DDoS attacks based on DNS. In *2016 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 276–280.

[5] R Arends, R Austein, M Larson, Daniel Massey, Scott W Rose, et al. 2005. DNS Security Introduction and Requirements, RFC 4033. (2005).

[6] Giuseppe Ateniese and Stefan Mangard. 2001. A new approach to DNS security (DNSSEC). In *Proceedings of the 8th ACM conference on Computer and Communications Security*. 86–95.

[7] Yaser M. Banadaki. 2020. Detecting Malicious DNS over HTTPS Traffic in Domain Name System using Machine Learning Classifiers. *Journal of Computer Sciences and Applications, 2020, Vol. 8, No. 2, 46-55* (2020). https://doi.org/10.12691/jcsa-8-2-2

[8] Jonas Bushart and Christian Rossow. 2020. Padding Ain't Enough: Assessing the Privacy Guarantees of Encrypted DNS. In *10th USENIX Workshop on Free and Open Communications on the Internet (FOCI 20)*. USENIX Association. https://www.usenix.org/conference/foci20/presentation/bushart

[9] Nikolaos Chatzis. 2007. Motivation for behaviour-based DNS security: A taxonomy of DNS-related internet threats. In *The International Conference on Emerging Security Information, Systems, and Technologies (SECUREWARE 2007)*. IEEE, 36–41.

[10] Levente Csikor, Himanshu Singh, Min Suk Kang, and Dinil Mon Divakaran. 2021. Privacy of DNS-over-HTTPS: Requiem for a Dream?. In *2021 IEEE European Symposium on Security and Privacy (EuroSP)*. 252–271. https://doi.org/10.1109/EuroSP51992.2021.00026

[11] Trinh Viet Doan, Irina Tsareva, and Vaibhav Bajpai. 2021. Measuring DNS over TLS from the edge: adoption, reliability, and response times. In *International Conference on Passive and Active Network Measurement*. Springer, 192–209.

[12] Shuang Hao, Alex Kantchelian, Brad Miller, Vern Paxson, and Nick Feamster. 2016. PREDATOR: Proactive Recognition and Elimination of Domain Abuse at Time-Of-Registration. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria) *(CCS '16)*. Association for Computing Machinery, New York, NY, USA, 1568–1579. https://doi.org/10.1145/2976749.2978317

[13] Paul Hoffman and Patrick McManus. 2018. *DNS queries over HTTPS (DoH)*. Technical Report.

[14] Zi Hu, Liang Zhu, John Heidemann, Allison Mankin, Duane Wessels, and Paul Hoffman. 2016. *Specification for DNS over transport layer security (TLS)*. Technical Report.

[15] Georgios Kambourakis, Tassos Moschos, Dimitris Geneiatakis, and Stefanos Gritzalis. 2007. Detecting DNS amplification attacks. In *International workshop on critical information infrastructures security*. Springer, 185–196.

[16] A Mankin, D Wessels, and P Hoffman. 2016. Internet Engineering Task Force (IETF) Z. Hu Request for Comments: 7858 L. Zhu Category: Standards Track J. Heidemann. (2016).

[17] A. Mayrhofer. 2016. The EDNS(0) Padding Option. *Internet Engineering Task Force [IETF]* (2016).

[18] A. Mayrhofer. 2018. Padding Policies for Extension Mechanisms for DNS (EDNS(0)). *Internet Engineering Task Force [IETF]* (2018).

[19] D. Meyer. 2016. Networking Meets Artificial Intelligence: A Glimpse into the (Very) Near Future. CTO corner. Dated: 08- 19-2016.

[20] Rikima Mitsuhashi, Yong Jin, Katsuyoshi Iida, Takahiro Shinagawa, and Yoshiaki Takai. 2022. Malicious DNS Tunnel Tool Recognition using Persistent DoH Traffic Analysis. *IEEE Transactions on Network and Service Management* (2022), 1–1. https://doi.org/10.1109/TNSM.2022.3215681

[21] Marta Moure-Garrido, Celeste Campo, and Carlos Garcia-Rubio. 2022. Detecting Malicious Use of DoH Tunnels Using Statistical Traffic Analysis. In *Proceedings of the 19th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, Ubiquitous Networks* (Montreal, Quebec, Canada) *(PE-WASUN '22)*. Association for Computing Machinery, New York, NY, USA, 25–32. https://doi.org/10.1145/3551663.3558605

[22] Tuan Anh Nguyen and Minho Park. 2022. DoH Tunneling Detection System for Enterprise Network Using Deep Learning Technique. *Applied Sciences* 12, 5 (2022). https://doi.org/10.3390/app12052416

[23] Amirreza Niakanlahiji, Soeren Orlowski, Alireza Vahid, and J. Haadi Jafarian. 2023. Toward practical defense against traffic analysis attacks on encrypted DNS traffic. *Computers Security* 124 (2023), 103001. https://doi.org/10.1016/j.cose.2022.103001

[24] Jim Reid and Anton Holleman. 1998. Domain Name System: The Origin Solution. In *Proceedings of the Annual Conference on USENIX Annual Technical Conference* (New Orleans, Louisiana) *(ATEC '98)*. USENIX Association, USA, 28.

[25] Sandra Siby, Marc Juarez, Claudia Diaz, Narseo Vallina-Rodriguez, and Carmela Troncoso. 2020. Encrypted dns- privacy. *A Traffic Analysis Perspective (Proc. of the NDSS)* (2020).

[26] Sunil Kumar Singh and Pradeep Kumar Roy. 2020. Detecting Malicious DNS over HTTPS Traffic Using Machine Learning. In *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*. 1–6. https://doi.org/10.1109/3ICT51146.2020.9312004

[27] David Stalder. 2021. Machine-learning based Detection of Malicious DNS-over-HTTPS (DoH) Traffic Based on Packet Captures. https://files.ifi.uzh.ch/CSG/staff/vonderassen/extern/theses/ba-stalder.pdf.

[28] Dmitrii Vekshin, Karel Hynek, and Tomas Cejka. 2020. DoH Insight: Detecting DNS over HTTPS by Machine Learning. In *Proceedings of the 15th International Conference on Availability, Reliability and Security* (Virtual Event, Ireland) *(ARES '20)*. Association for Computing Machinery, New York, NY, USA, Article 87, 8 pages. https://doi.org/10.1145/3407023.3409192

[29] Fatema Bannat Wala and Chase Cotton. 2022. " Off-Label" use of DNS. *Digital Threats: Research and Practice* (2022).