# Exploring the feasibility of adapting an enterprise-level malicious DNS over HTTPS traffic detection framework to be applicable for consumer-level settings

# EXPLORING THE FEASIBILITY OF ADAPTING AN ENTERPRISE-LEVEL MALICIOUS DNS OVER HTTPS TRAFFIC DETECTION FRAMEWORK TO BE APPLICABLE FOR CONSUMER-LEVEL SETTINGS

A research Exploring the Feasibility of Adapting an Enterprise-Level Malicious DNS
Over HTTPS Traffic Detection Framework to be Applicable for Consumer-Level Settings
submitted in partial
fulfillment of the requirements for the degree of
Master of Science

By

Nabin Niroula
Metropolitan State University of Denver, 2019
Bachelor of Science in Mathematics and Computer Science

# ABSTRACT

Since DNS over Https (DoH) is a new technology in the ecology of computer networks, extensive research is still required to discover effective methods for preventing its exploitation. Existing research primarily focuses on identifying malicious DoH traffic within enterprise-level environments. Preliminary investigation, at the time of writing of this report, revealed a lack of studies addressing the testing of malicious DoH traffic in consumer-level settings, with limited if any available. To assess the viability of adapting an enterprise-level malicious DNS over HTTPS (DoH) traffic detection framework for consumer-level settings, a simulation was carried out by substantially reducing its features, utilizing resources from multiple studies. However, this approach resulted in a notable drop in accuracy by more than 20% compared to existing frameworks, particularly evident in a small dataset, and the impact is anticipated to worsen with larger datasets. Nevertheless, the potential for meaningful contributions to safeguarding DoH traffic at the consumer level exists if the research and the simulation incorporate advanced machine learning and deep learning algorithms that are capable of pattern and behavior matching.

This Project Report is approved for recommendation to the Graduate Committee.

Project Advisor:

________________________________________

Dr. Ilkyeun Ra

## TABLE OF CONTENTS

## LIST OF FIGURES

# 1. Introduction

## 1.1 Problem

As human-readable hostnames are not suitable for the digital activities or transactions of entities within computer networks and the Internet, there is a need for a Directory Service that facilitates hostname to an IP address translation. The Domain Name System (DNS) serves as the mechanism to fulfill this requirement. DNS operates as an application layer protocol, utilized by other application layer protocols such as HTTP, HTTPS, SMTP, and FTP. When DNS is integrated into HTTPS, it is referred to as the DNS Over HTTPS protocol [1]. Examining the functioning of HTTPS, it utilizes SSL/TLS technology with public key encryption to distribute a shared symmetric key for data encryption and authentication [2]. Hence, embedding unencrypted DNS queries in HTTPS should inherently secure them.

Since DNS over Https (DoH) is a new technology in computer networks, extensive research is still required to discover effective methods for preventing its exploitation [3]. Existing research primarily focuses on identifying malicious DoH traffic within enterprise-level environments. My preliminary investigation, at the time of writing of this report, revealed a lack of studies addressing the testing of malicious DoH traffic in consumer settings, with limited if any available. I wanted to explore the feasibility of adapting an enterprise-level malicious DoH traffic detection framework for consumer settings. Despite the growing adoption of DoH as a standard for DNS traffic, individual users at the consumer level exhibit hesitancy in enabling it within their web environment. This reluctance stems from the fact that the predominant focus of existing research has been on identifying malicious DoH traffic in enterprise-level contexts. There has been limited to no exploration into evaluating its effectiveness in consumer-level traffic, thereby leaving ordinary users uninformed about its potential shortcomings. Conducting thorough analytical research to distinguish between malicious and benign DoH traffic within consumer-level network settings becomes imperative. Such research would offer valuable insights into the security efficacy that the DoH infrastructure introduces to the network environment, serving as a deterrent against various cyber attacks.

## 1.2 Project Objective

Adapting a malicious DNS over HTTPS (DoH) traffic detection framework, originally designed for enterprise-level settings, to suit consumer-level environments by significantly reducing its features for smaller datasets is feasible. However, it necessitates the utilization of advanced machine learning and deep learning algorithms capable of accurately matching patterns and behaviors.

### 1.3 Approach

The initial phase of the project commences with an extensive review of peer-reviewed journals and articles to identify relevant content supporting the research topic. After scrutinizing over 5 dozen published articles, a substantial portion being peer-reviewed journals, the decision was made to opt for simulation. This choice involves amalgamating one or more pertinent works since establishing an implementation from the ground up proved to be impractical for me, given the high-level requirements for machine learning and deep learning concepts.

The second phase of the approach involves installing the Firefox web browser to facilitate DNS Over HTTPS (DoH) traffic, followed by the installation of Wireshark for data capturing. Default settings in Firefox were adjusted to enable DoH traffic. To validate the browser's support for DoH traffic, a sample of website browsing was captured in Wireshark. Subsequently, the captured payload entries indicating HTTPS traffic were analyzed further to confirm their potential as DoH traffic. Once it has been confirmed that the setup functions as intended to allow DoH traffic, the research approach advances to the next stage.

The third phase of the project involves simulation, drawing upon resources from multiple articles to fulfill the requirements. These articles typically generate extensive results, comprising around ten thousand rows and over thirty columns of data for statistical analysis, as their programs are tailored for enterprise-level DoH traffic. However, my objective was to utilize 3 - 5 sample datasets instead of the extensive ten thousand entries. Consequently, I had to significantly reduce their features while striving to maintain essential functionalities at a bare minimum.

The fourth phase of the approach involves determining input parameters, understanding expected outputs, conducting experiments, and ultimately producing sample statistical results. Three iterations of outputs were obtained, and the average accuracy in identifying malicious DoH traffic was calculated.

### 1.4 Organization of this Project Report

The remaining structure of the paper encompasses sections 2, 3, 4, 5, and the bibliography. Section 2 delves into the background, with subsection 2.1 defining new terminologies essential for understanding the content in this paper, and 2.2 focusing on literature reviews. Section 3 underscores the approaches taken to accomplish this research, with subsections 3.1 providing a high-level overview of the project, 3.2 centering on the title of the paper, and 3.3 delving into the details of implementation.

Section 4 of the paper is dedicated to testing the project to assess if it meets expectations. Subsection 4.1 concentrates on methodologies, 4.2 presents the results, and 4.3 provides a statistical analysis of the outcome. Section 5 is dedicated to conclusions, where subsection 5.1 summarizes the paper briefly, subsection 5.2 explores the potential impact of the project, and finally, subsection 5.3 shifts the discussion to potential future work. The paper concludes with bibliographic references at the end.

# 2. Background

## 2.1 Key Concepts

Acquiring fundamental knowledge about the following terminologies would enhance the understanding of the content presented in this paper.

### 2.1.1 IP address and host name

Would it be surprising to learn that computers operate based on numbers rather than names? The concept of "going by numbers” might be initially perplexing, but it will become clearer shortly. Consider if it would be surprising to say that human beings are identified by names in public and by specific identification numbers, such as Social Security Numbers or State IDs in computer systems? Of course not, right? We are accustomed to them. When you enter a human-friendly mnemonic name like www.ucdenver.edu into your favorite browser, you are retrieving information about the University of Colorado Denver. The seamless cyber transaction that provides you with this information involves underlying technology. While the mnemonic name is user-friendly, routers in computer network systems prefer that this human friendly mnemonic name be represented with a number. This number that represents the human friendly name is an Ip address. There are two common formats for IP addresses: IPV4 and IPV6. An example IPV4 address for the University of Colorado Denver could be 140.226.9.168. IPV6 is the newer version of IP address representation. The human-friendly term for a website, such as www.ucdenver.edu, is referred to as a hostname.

### 2.1.2 DNS

It is comprehendible that human-readable hostnames are not suitable for the digital activities or transactions of entities in computer networks and the Internet. To address this, a hostname to IP address translation service is essential. The Domain Name System (DNS) serves this purpose by translating user-friendly hostnames into router-friendly IP addresses. Since its official establishment by the Internet Corporation for Assigned Names and Numbers (ICANN) in 1983 [4], DNS has been the de facto standard for interpreting between hostnames and IP addresses. The process involves a distributed database implemented in a hierarchy of DNS servers, according to the book "Computer Networking: A Top-Down Approach" [5]. As a database server, the DNS server stores information based on domain names and responds to client queries by matching them in the database.

In the context of the five-layer model in computer network topology, server-level protocols are implemented at the application layer. Therefore, DNS operates as an application layer protocol. Additionally, the article "DNS for IoT: A survey" highlights the role of DNS in serving IoT environments and implicitly mentions its use by various application layer protocols such as HTTP, HTTPS, SMTP, and FTP [6].

During a DNS query, when a user enters the URL of the desired website in their browser, the browser sends an unencrypted plaintext connection request over the Internet to the DNS server, seeking the IP address of the website. The DNS server responds accordingly. Since the request is not encrypted, adversaries can easily eavesdrop and determine the website the user intends to access.

### 2.1.3 HTTPS

Even though HTTPS had its inception in the mid-1990s, it was officially specified in May 2000 through RFC 2818 documentation [7]. Serving as a secure version of HTTP (Hypertext Transfer Protocol), HTTPS ensures the encryption of all communications and data exchange transactions between the client and the server. In the event of an adversary intercepting HTTPS traffic with the intent to extract data, the obtained information appears garbled, rendering it impossible to convert into a human-readable format without the appropriate encryption device, typically the user's private key. The underlying process is straightforward: "HTTPS employs SSL/TLS technology with public key encryption to distribute a shared symmetric key for data encryption and authentication" [8]. Prior to transmitting data via HTTPS, an SSL/TLS handshake occurs, initiating the exchange of data.

### 2.1.4 DNS Over HTTPS (DoH)

Referencing section 2.1.2, the embedding of DNS into HTTPS is the DNS Over HTTPS protocol. The Internet Engineering Task Force officially adopted the DNS over HTTPS protocol in 2018 through RFC 8484 documentation to address privacy concerns associated with the plain text transmission of the DNS protocol [9]. Unlike the traditional DNS protocol, the DNS Over HTTPS protocol brings about a shift in the request-response mechanism for DNS queries. In this protocol, the browser conveys the URL of the website to HTTPS via an encrypted connection. To be more specific, the DNS Over HTTPS protocol secures DNS queries by camouflaging them as regular

HTTPS traffic [10]. This complexity enhances the difficulty for adversaries attempting to exploit the traffic and discern the user's intended website access. Nevertheless, DoH has been susceptible to attacks. While concealing DNS queries in the HTTPS protocol can be advantageous for regular users, it presents challenges in enterprise-level work environments, where employees may struggle to identify the query's destination, potentially leading to cyber-attacks [11].

### 2.1.5 Encryption

Encryption, particularly network encryption, serves as a method for ensuring data privacy by safeguarding egress traffic against adversarial interception. This process involves encoding plaintext messages into ciphertext format using algorithmic cryptographic models. To revert the message to its original form, a decryption key is necessary, generated by the same algorithm. In the context of DoH, encryption occurs by concealing DNS queries within the HTTPS payload [12]. The DoH packet appears akin to an HTTPS packet [13].

### 2.1.6 DoH traffic

The DNS message traversing port 443, secured with TLS encryption, and keeping all transmitted content hidden is referred to as DoH traffic [14]. Specifically, it entails the secure transmission of DNS queries between the client and the server [15].

### 2.1.7 Handshake

Prior to transmitting data via HTTPS, the client initiates a request to the server, and the server acknowledges the client's request. This process is known as a handshake. Throughout the handshake, the client and the server mutually agree on various connection parameters, essential for ensuring secure data exchange[16].

## 2.2 Related Work or Literature Review

The literature review will encompass a range of topics crucial for the successful accomplishment of this research. These areas comprise DNS over HTTPS attacks, frameworks to differentiate DoH and non-DoH traffic, frameworks for identifying malicious DoH traffic, sources of datasets, various DoH traffic detection algorithms, simulation guidelines, and the performances and features of existing frameworks.

### 2.2.1 DNS Over HTTPS (DoH) Attacks

Although DNS Over HTTPS is a recent technology, several cyber attacks targeting it have already been documented. Various articles explore different types of DoH attacks, and understanding the traffic and behavioral patterns associated with these attacks is crucial for choosing the right algorithm. One of the known attacks on DoH is the Traffic Analysis attack, where attackers employ a machine learning-based approach to identify unencrypted features of DNS by considering request/response time and delay latency between two consecutive queries[17]. According to guidance in RFC 8467, this attack is addressed by having DoH clients and resolvers insert null bytes into DNS queries and responses, padding them to specific block sizes before encryption[18]. However, this padding methodology is ineffective in countering cyber attacks [19].

One other attack worth discussing here is the DoH tunneling attack. A detecting mechanism for this attack has been developed, which can be integrated with the security system of an enterprise network[20]. H. Jha et al. mentioned that "bad actors use DoH Tunneling to inject malware or stolen credentials into DNS queries, creating a covert communication channel that bypasses most firewalls and results in sensitive data leakage" [21].

The suggested solution for modifying frameworks intended for enterprise-level detection of malicious DoH traffic to be suitable for consumer-level detection considers these attacks and similar ones, aiding in the selection of suitable algorithms.

### 2.2.2 Frameworks to differentiate DoH versus non-DoH traffics

Various scholarly articles delve into different network frameworks designed to distinguish web traffic as either DoH or non-DoH. Examining some of these frameworks would certainly contribute to making informed decisions when it comes to simulating the proposed solution. Some of them are as below.

Montazerishatoori et al. conducted a study on DoH traffic detection, implementing a machine learning algorithm to distinguish DoH traffic from HTTPS traffic[22]. They achieved an F1 score of 0.993 using the CIRA-CIC-DoHBrW-2020 dataset, considering 28 out of 33 common DoH traffic features. The dataset being used is called the standard sources for DoH detection.

Similarly, In [15], the authors introduced a two-layer network architecture that assesses traffic at distinct phases: initially identifying whether traffic is DoH, followed by a secondary identification to confirm DoH traffic at level 2. While this approach may seem redundant, they assert that their overall prediction accuracy reached at least 99.5%. The tools and resources employed in their proposal offer substantial support for the simulation conducted in this research.

### 2.2.3 Frameworks to detect malicious DoH traffics

Numerous architectures for detecting malicious DoH traffic have been proposed in various studies, but a predominant focus lies in offering solutions for enterprise-level environments. This inclination could be attributed to the requirement for extensive datasets and resources, as elaborated in sections 2.2.4 and 2.2.5. This section opts to forego a detailed discussion of some of these frameworks.

First, as discussed in section 2.2.2, Banadaki used the standard dataset for his purpose, and claims that his study achieved a malicious DoH traffic prediction accuracy of 100% [23]. His research mostly revolved around implementing time-related features along with IP addresses and port numbers by employing machine learning algorithms.

Second, Behnke et al. tried improving the model proposed by Banadaki by removing IP addresses, port numbers, and other features, which they referred to as insignificant concerns and were not utilized in the original model [24]. Their modified architecture still provides an F1 score of 0.998, which correlates to the prediction accuracy of approximately 99.8%. Their method of altering existing technology by eliminating unused features motivated me to undertake the challenge of simulating an enterprise-level malicious DNS Over HTTPS traffic detection framework. The goal is to significantly reduce the existing features to adapt it to consumer-level settings, aiming to identify malicious DoH traffic in devices with limited resources.

Third, L. F. Gonzalez et al. also used the de facto datasets mentioned in section 2.2.2, and have applied a two fold process in predicting malicious DoH traffic. In their first phase, their proposed architecture distinguishes traffic into DoH or non-DoH ones [25]. In their second phase, they took DoH traffic obtained from phase I and further analyzed it to be either malicious or benign by implementing 26 out of 34 most common features that describe network traffic patterns. The authors claimed that this architecture is a good example of the malicious DoH traffic detection framework that could be used with computers having limited resources. This claim became further inspiration for me to try exploring the feasibility of adapting a framework that was designed to be incorporated in machines with ample resources to make it usable with computers having limited resources, if possible, by removing the features requiring high level resources.

### 2.2.4 Datasets and tools used by existing research

Most of the aforementioned research utilized the standard datasets outlined in section 2.2.2 [26]. They appear to have organized their data into four distinct CSV files, encompassing DoH, non-DoH, malicious, and benign DoH datasets. Common tools employed in their investigations include dns2tcp, DNSCat2, Iodine, DoHLyzer, DoHMeter tool, machine learning algorithms, deep learning algorithms, networking protocols, and more. It appears that nearly all of them focused on exploring the realm of the 34 most common network traffic patterns.

### 2.2.5 DoH Traffic Detection Algorithms and Models

[10] uses statistical variance and different machine learning techniques to identify malicious traffic. The authors highlighted that AdaBoost algorithm gave them the highest possible prediction accuracy. While [23] implemented LightGBM and XGBoost algorithms to obtain perfect accuracy, [27] employs Random Forest and Gradient Boosting classifying algorithms in their tests. [28] discusses different algorithms and models for detecting malicious DoH traffic.

### 2.2.6 Research direction and Simulation parameters

There are many articles that contributed towards giving an appropriate direction to my research either directly or indirectly. The materials in [29] adhered to my project by providing the light on the simulation parameters. The variables and functions in the simulation project follow the same convention defined in it, by utilizing various other resources to hit minimum expectation. The articles [30], [31], [32], [33], and other articles provided great ideas and introductions to different concepts that were of utmost importance for accomplishing this project.

# 3. Approach

## 3.1 High Level Design

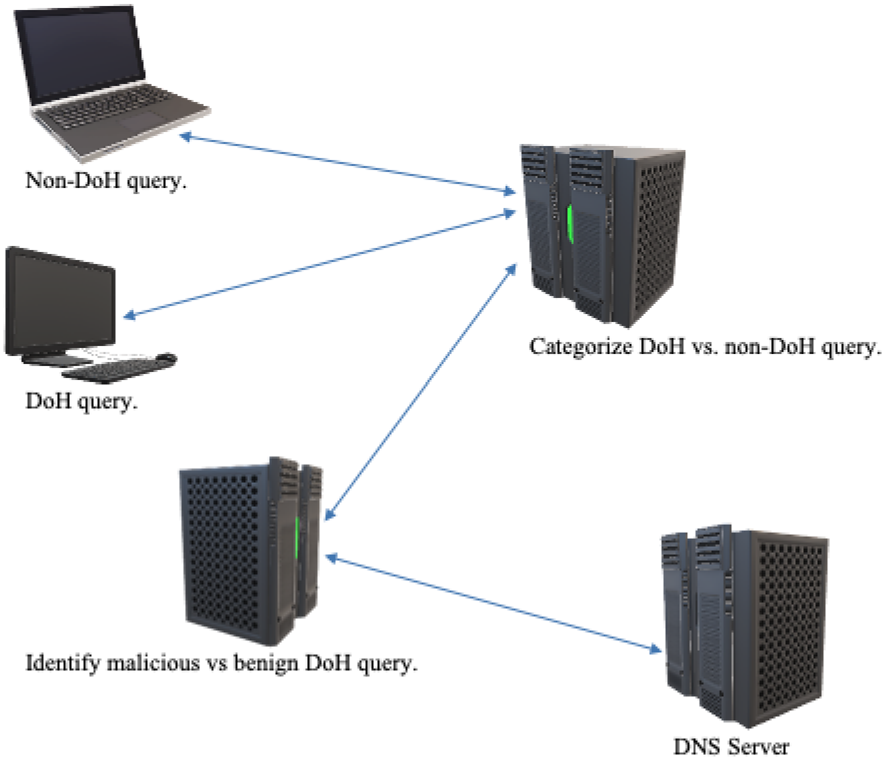

Non-DoH query.
DoH query.
Categorize DoH vs. non-DoH query.
Identify malicious vs benign DoH query.
DNS Server

**fig: high level view of the design**

## 3.2 Infrastructure Design

The client query may either be DoH-enabled or a traditional DNS query. Regardless of the type, the primary objective is to ascertain its validity as a DoH query. Once validated, the DoH query undergoes analysis to determine whether it is malicious or benign. Upon confirming its non-malicious nature, the query can then be forwarded to the DNS server. This process helps safeguard the server from initial contact attacks.

### 3.3 Implementation

The implementation phase of the project commences with an idea for simulation. This choice involves amalgamating one or more pertinent works since setting up an implementation from scratch proved to be impractical for me, given the high-level requirements for machine learning and deep learning concepts.

The second step of the approach includes setting up the Firefox web browser to enable DNS Over HTTPS (DoH) traffic, followed by the installation of Wireshark for data capture. The default settings in Firefox were modified to facilitate DoH traffic. To verify the browser's DoH traffic support, a sample of website browsing was captured in Wireshark. The captured payload entries, indicating HTTPS traffic, were then scrutinized to confirm their potential as DoH traffic. With confirmation that the setup effectively allows DoH traffic, the research approach proceeds to the next phase.

In the third stage of the project, simulation is undertaken, leveraging resources from various articles to meet the specified requirements. Typically, these articles produce extensive results, yielding approximately ten thousand rows and over thirty columns of data for statistical analysis, as their programs are designed for enterprise-level DoH traffic. However, my aim was to work with 3 - 5 sample datasets instead of the extensive ten thousand entries in a device with limited resources. Consequently, a substantial reduction in features was necessary while ensuring the preservation of essential functionalities at a minimal level.

The fourth stage of the implementation entails defining input parameters, comprehending anticipated outputs, executing experiments, and ultimately generating sample statistical results. Three samples of outputs were acquired, and the average accuracy in identifying malicious DoH traffic was computed.

## 4. Methodology, Results and Analysis (or similar title)

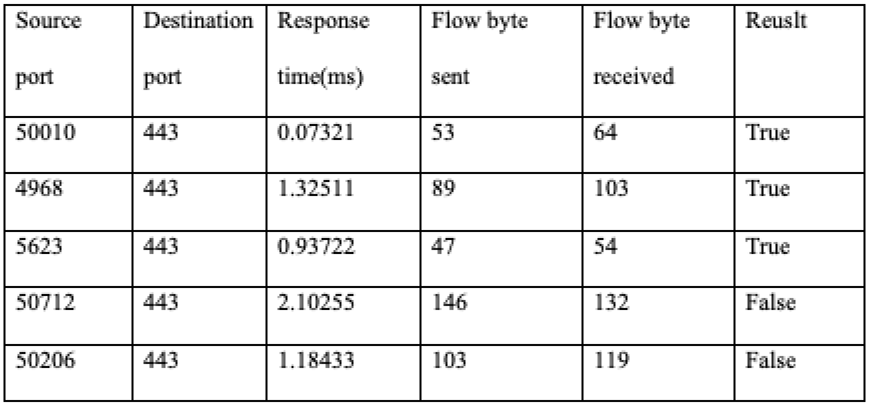
### 4.1 Methodology

Describe test configuration and test methodology (set of steps used for testing) Following steps were run after the preliminary setup was completed for the simulation of the project.

1. Install firefox and enable DNS over HTTPS.
2. Install Wireshark and capture sample website surfacing data to ensure it supports DoH traffic.
3. Download the existing source code and remove any thing that requires significant resources utilization.
4. Input and output parameters are significantly removed, leaving only needed ones. Some of the removed parameters include source Ip address, destination Ip address, deviation, and many more.
5. Removed majority of the Machine learning algorithm and models so that code stays bare minimum. Later on, they have to be incorporated based on the need.
6. Run the code to make sure that it is still functional.
7. Imported the dataset and tried printing out 5 sample outputs using malicious DNS Over HTTPS dataset. This step involves identifying malicious DoH traffic. The output parameters are set to true or false, indicating that false would reflect the traffic being non-malicious and true signals that the query was malicious.
8. At this time it generates some sample output, but there is no guarantee that the output is valid.
9. The simulation halts at this moment as incorporating back some machine learning algorithm for data prediction crashes the program. A lot of work is needed to understand what went wrong by deeply learning what the machine learning algorithms in action do and what their requirements are.

### 4.2 Results

The matplotlib library is yet to be implemented in the above steps, the consequence of which is that the implementation does not programmatically generate a visualization graph for the output. However, the console output can be tabulated as follows.



| Source port | Destination port | Response time(ms) | Flow byte sent | Flow byte received | Reuslt |
|---|---|---|---|---|---|
| 50010 | 443 | 0.07321 | 53 | 64 | True |
| 4968 | 443 | 1.32511 | 89 | 103 | True |
| 5623 | 443 | 0.93722 | 47 | 54 | True |
| 50712 | 443 | 2.10255 | 146 | 132 | False |
| 50206 | 443 | 1.18433 | 103 | 119 | False |

## 4.3 Analysis

There are a number of things to account for in this project. Three data samples were taken as shown above in the table, and their average accuracy was calculated as follows.
In the first iteration, 1 malicious traffic was sent, it identified 2 as malicious ones. This yields 20% inaccuracy.
In the second iteration, 2 malicious queries were sent, it predicted correctly. This has 0% inaccuracy.
The third iteration had 3 malicious queries and it predicted 1 correctly. So, it has an inaccuracy of 40%.
Hence, average inaccuracy = (20 + 0 + 40)/3 = 20%.

With proper setup of machine learning algorithms and the use of an appropriate model, the prediction accuracy could be enhanced. The findings suggest that significant future work is necessary to establish the practical viability of the proposed framework.

# 5. Conclusions

## 5.1 Summary

DNS transforms user-friendly hostnames into router-friendly IP addresses. HTTPS, the secure iteration of Hypertext Transfer Protocol (HTTP), integrates with DNS to form the DNS over HTTPS (DoH) protocol, primarily designed to secure DNS queries. During the handshake, the client and server collaboratively establish different connection parameters.

Some types of DoH attacks include Traffic Analysis attacks and DoH Tunneling attacks. Several research articles propose different frameworks for detecting DoH traffic. While many malicious DoH traffic detection frameworks have been suggested for enterprise-level settings, only a few address the challenge of designing such frameworks for devices with limited resources. Tools and algorithms commonly employed in DoH traffic detection frameworks include LightGBM, XGBoost, Random Forest, and others.

Adapting a malicious DoH traffic detection framework, designed for enterprise-level settings, to suit consumer-level environments by significantly reducing its features for smaller datasets is feasible. However, advanced machine learning and deep learning algorithms as well as models capable of accurately matching patterns and behaviors must be employed.

## 5.2 Potential Impact

If this project report were correct, it offers valuable insights into the security effectiveness of the DoH infrastructure in the network environment. This may potentially inspire the network community to develop a DoH traffic detection framework tailored for consumer settings. Despite the widespread adoption of DoH, individual users at the consumer level still show reluctance in enabling it within their web environment, often due to a lack of awareness about its significance or potential drawbacks. This research increases awareness among users, contributing to the mission of emphasizing the importance of DoH for ordinary web users.

## 5.3 Future Work

While this Project Report provides the basic foundation towards adapting an enterprise level malicious DoH detection framework at consumer level settings, there are many things not accomplished in this report. Having those things done in the future would add meaning to this project. Some of the future works are listed below.

1. Simulation needs to be completed in the right way, where it implements various machine learning algorithms and models to improve detection capacity.
2. Based on various network traffic patterns, appropriate feature selection is of paramount importance.
3. For classifying malicious versus non-malicious traffic, classifiers such as decision tree, Random Forest Classifiers, KNeighbors classifiers etc should be incorporated in the project.
4. An intensive research in the area of machine learning is equally important.

## References

[1] Q. Huang, "A Comprehensive Study of DNS-Over-HTTPS Downgrade Attack.", eScholarship, University of California, 2020.

[2] X. Liu *et al*, "Attention-based bidirectional GRU networks for efficient HTTPS traffic classification," *Information Sciences,* vol. 541, pp. 297-315, 2020.

[3] A. T. Nguyen and M. Park, "Detection of DoH tunneling using semi-supervised learning method," in 2022, DOI: 10.1109/ICOIN53446.2022.9687157.

[4] O. M. Bonastre, A. Vea and D. Walden, "Origins of the Domain Name System," *IEEE Annals of the History of Computing,* vol. 41, *(2),* pp. 48-60, 2019.

[5] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet.* (6th ed.) Boston: Pearson/Addison Wesley, 2012.

[6] I. Ayoub *et al*, "DNS for IoT: A Survey," *Sensors (Basel, Switzerland),* vol. 23, *(9),* pp. 4473, 2023.

[7] D. Glăvan *et al*, "Man in the middle attack on HTTPS protocol," *Scientific Bulletin ("Mircea Cel Bătrân" Naval Academy),* vol. XXIII, *(1),* pp. 199-201, 2020.

[8] K. Jerabek *et al*, "DNS Over HTTPS Detection Using Standard Flow Telemetry," *IEEE Access,* vol. 11, pp. 50000-50012, 2023.

[9] K. Hynek, D. Vekshin, J. Luxemburk, T. Cejka and A. Wasicek, "Summary of DNS Over HTTPS Abuse," in *IEEE Access*, vol. 10, pp. 54668-54680, 2022, doi: 10.1109/ACCESS.2022.3175497.

[10] M. Moure-Garrido, C. Campo and C. Garcia-Rubio, "Real time detection of malicious DoH traffic using statistical analysis," *Computer Networks (Amsterdam, Netherlands: 1999),* vol. 234, pp. 109910, 2023.

[11] M. Zhan *et al*, "Detecting DNS over HTTPS based data exfiltration," *Computer Networks (Amsterdam, Netherlands: 1999),* vol. 209, pp. 108919, 2022.

[12] H. Jha *et al*, "Detection of tunneling in DNS over HTTPS," in 2021, DOI: 10.1109/ICSC53193.2021.9673380.
[13] L. Csikor *et al*, "Privacy of DNS-over-HTTPS: Requiem for a dream?" in 2021, . DOI: 10.1109/EuroSP51992.2021.00026.

[14] M. Moure-Garrido, C. Campo and C. Garcia-Rubio, "Real time detection of malicious DoH traffic using statistical analysis," *Computer Networks (Amsterdam, Netherlands: 1999),* vol. 234, pp. 109910, 2023.

[15] A. Aggarwal and M. Kumar, "An ensemble framework for detection of DNS-Over-HTTPS (DOH) traffic," *Multimedia Tools and Applications,* 2023.

[16] N. A. Khan, A. S. Khan, H. A. Kar, Z. Ahmad, S. Tarmizi and A. A. Julaihi, "Employing Public Key Infrastructure to Encapsulate Messages During Transport Layer Security Handshake Procedure," *2022 Applied Informatics International Conference (AiIC)*, Serdang, Malaysia, 2022, pp. 126-130, doi: 10.1109/AiIC54368.2022.9914605.

[17] S. Xiong, A. D. Sarwate and N. B. Mandayam, "Network Traffic Shaping for Enhancing Privacy in IoT Systems," *IEEE/ACM Transactions on Networking,* vol. 30, *(3),* pp. 1-16, 2022.

[18] A. Niakanlahiji et al, "Toward practical defense against traffic analysis attacks on encrypted DNS traffic," *Computers & Security,* vol. 124, pp. 103001, 2023.

[19] S. Siby *et al*, "Encrypted DNS --> privacy? A traffic analysis perspective," Cornell University Library, arXiv.org, Ithaca, 2019. DOI: 10.48550/arxiv.1906.09682.

[20] T. A. Nguyen and M. Park, "DoH Tunneling Detection System for Enterprise Network Using Deep Learning Technique," *Applied Sciences,* vol. 12, *(5),* pp. 2416, 2022.

[21] H. Jha, I. Patel, G. Li, A. K. Cherukuri and S. Thaseen, "Detection of Tunneling in DNS over HTTPS," *2021 7th International Conference on Signal Processing and Communication (ICSC)*, Noida, India, 2021, pp. 42-47, doi: 10.1109/ICSC53193.2021.9673380.

[22] M. MontazeriShatoori, L. Davidson, G. Kaur and A. H. Lashkari, "Detection of DoH tunnels using time-series classification of encrypted traffic", *Proc. IEEE Int. Conf.*

*Dependable Autonomic Secure Comput. Int. Conf. Pervasive Intell. Comput. Int. Conf. Cloud Big Data Comput. Int. Conf. Cyber Sci. Technol. Congr. (DASC/PiCom/CBDCom/CyberSciTech)*, pp. 63-70, Aug. 2020

[23] Y. M. Banadaki, "Detecting malicious DNS over HTTPS traffic in domain name systems using machine learning classifiers", *J. Comput. Sci. Appl.*, vol. 8, no. 2, pp. 46-55, Aug. 2020.

[24] M. Behnke, N. Briner, D. Cullen, K. Schwerdtfeger, J. Warren, R. Basnet, et al., "Feature engineering and machine learning model comparison for malicious activity detection in the DNS-over-HTTPS protocol", *IEEE Access*, vol. 9, pp. 129902-129916, 2021.

[25] L. F. Gonzalez Casanova and P. Lin, "Malicious Network Traffic Detection for DNS over HTTPS using Machine Learning Algorithms," *APSIPA Transactions on Signal and Information Processing,* vol. 12, *(2),* 2023.

[26] D. Vekshin, K. Hynek and T. Cejka, "Dataset used for detecting DNS over HTTPS by Machine Learning," 2020. . DOI: 10.5281/zenodo.3818004.

[27] S. K. Singh and P. K. Roy, "Detecting malicious DNS over HTTPS traffic using machine learning", *Proc. Int. Conf. Innov. Intell. Inform. Comput. Technol.*, pp. 1-6, 2020.

[28] M. Behnke *et al*, "Feature Engineering and Machine Learning Model Comparison for Malicious Activity Detection in the DNS-Over-HTTPS Protocol," *IEEE Access,* vol. 9, pp. 129902-129916, 2021.

[29] R. Rawat *et al*, "Analysis and detection of malicious activity on DoH traffic," in 2021, . DOI: 10.1109/GCAT52182.2021.9587555.

[30] T. Zebin, S. Rezvy and Y. Luo, "An Explainable AI-Based Intrusion Detection System for DNS Over HTTPS (DoH) Attacks," *IEEE Transactions on Information Forensics and Security,* vol. 17, pp. 2339-2349, 2022.

[31] A. Almusawi and H. Amintoosi, "DNS Tunneling Detection Method Based on Multilabel Support Vector Machine," *Security and Communication Networks,* vol. 2018, pp. 1-9, 2018.

[32] Z. Sui *et al*, "A Comprehensive Review of Tunnel Detection on Multilayer Protocols: From Traditional to Machine Learning Approaches," *Applied Sciences,* vol. 13, *(3),* pp. 1974, 2023.

[33] Q. Abu Al-Haija, M. Alohaly and A. Odeh, "A Lightweight Double-Stage Scheme to Identify Malicious DNS over HTTPS Traffic Using a Hybrid Learning Approach," *Sensors (Basel, Switzerland),* vol. 23, *(7),* pp. 3489, 2023.

This is the final page of a Project Report and should be a blank page