UNIVERSITY OF OKLAHOMA
GRADUATE COLLEGE

EVALUATION OF MOVING TARGET DOH SERVER DETECTION USING MACHINE LEARNING MODELS

A THESIS
SUBMITTED TO THE GRADUATE FACULTY
in partial fulfillment of the requirements for the
Degree of
MASTER OF SCIENCE

By

MARC C. BERET
Norman, Oklahoma
2024

EVALUATION OF MOVING TARGET DOH SERVER DETECTION USING MACHINE LEARNING MODELS

A THESIS APPROVED FOR THE
SCHOOL OF COMPUTER SCIENCE

BY THE COMMITTEE CONSISTING OF

Dr. Anindya Maiti, (Chair)

Dr. Qi Cheng

Dr. Song Fang

© Copyright by MARC C. BERET 2024
All Rights Reserved.

Acknowledgements

I would like to thank all the people who have accompanied me, taught me, and helped me grow professionally during this thesis.

Firstly, my deepest gratitude goes to my advisor, Dr. Anindya Maiti, Assistant Professor at the University of Oklahoma, for his warm welcome, patience, and invaluable expertise, which have guided me throughout this journey.

I am also grateful to my committee members, Dr. Song Fang and Dr. Qi Cheng, for their valuable feedback, insightful suggestions, and support, which have greatly enhanced the quality of this work.

I would also like to thank Scott Seidenberger for his significant contributions to my research and his essential role in creating the *NinjaDoH* tool, which played a crucial part in this work.

A sincere thank you to all the members of the SECRET LAB for their generosity in sharing their experiences and insights with me, which greatly enriched my learning.

Lastly, thank you, the readers, for the time and attention you will devote to this thesis.

Table of Contents

List Of Tables

List Of Figures

Abstract

The rapid adoption of DNS over HTTPS (DoH) has introduced significant challenges in balancing privacy, security, and resistance to censorship. This thesis explores the feasibility of developing a censorship-resistant DoH service, named *NinjaDoH*, which leverages hyperscalers and the InterPlanetary File System (IPFS) to enhance accessibility and resilience against censorship efforts.

The study investigates two core research questions: first, the development and implementation of *NinjaDoH* , a dynamic, censorship-resistant DoH service utilizing hyperscalers and IPFS; and second, the efficiency of existing firewall solutions in targeting *NinjaDoH*, a moving DoH service, using various machine learning models.

By evaluating firewall responses and the success of advanced machine learning techniques in identifying *NinjaDoH* traffic, this research highlights the strengths and weaknesses of current detection methods. The findings demonstrate the potential of IPFS as a robust, censorship-resistant solution for secure DoH communication, offering a novel framework for safeguarding internet access against state-level and organizational censorship.

Chapter 1

Introduction

The adoption of DNS over HTTPS (DoH) is rapidly increasing as individuals and organizations prioritize privacy and security in their online activities. By encrypting DNS queries, DoH prevents third parties from monitoring, intercepting, or tampering with these queries. In many regions, DNS censorship remains a common tool for restricting access to information, often deployed to control or limit users' online experiences. Traditional DNS queries, sent over unencrypted channels, are easily monitored and intercepted, making DNS-based censorship straightforward to implement. DoH, in response, has emerged as a key technology to counter censorship efforts by encrypting DNS traffic, rendering it significantly more challenging to block or manipulate. DoH thus enhances privacy by shielding DNS queries from eavesdropping and interference, allowing users greater freedom and security in accessing information.

This thesis introduces *NinjaDoH*, a novel solution that employs a moving target defense strategy to further strengthen DoH's resistance against censorship. By dynamically altering IP addresses using hyperscalers and the InterPlanetary File System (IPFS) protocol, *NinjaDoH* provides a robust and adaptive approach to evade detection. This flexibility complicates the efforts of censorship tools to block traffic without risking interruptions to legitimate services. Central to this research is an examination of the effectiveness of AI/ML-based detection methods—sophisticated systems designed

to identify and block DoH traffic even within encrypted channels. *NinjaDoH* is specifically crafted to evade these AI-based detection mechanisms while maintaining high performance and usability.

This thesis fills a critical gap in evaluating *NinjaDoH* evasion techniques against advanced AI-based detection methods. We evaluate *NinjaDoH*'s ability to evade state-of-the-art machine learning models, including LSTM, CNN, and XGBoost, using comprehensive datasets encompassing both standard DoH and *NinjaDoH* traffic. By quantifying *NinjaDoH*'s evasion capabilities, this study sheds light on the effectiveness of moving target defenses in the context of encrypted DNS such as DoH. The findings have significant implications for the battle between censorship and internet freedom, it also offers insights for developing more resilient anti-censorship tools and exposing the limitations of current AI-based detection systems.

Thesis Statement: This thesis aims to evaluate the effectiveness of *NinjaDoH* in evading advanced AI-based detection systems designed to identify DoH traffic while maintaining high performance and usability.

Chapter 2

Literature Review

2.1 Domain Name System

The **Domain Name System (DNS)**, often compared to the Internet's "phonebook," is a critical component of the Internet. It translates human-readable domain names into machine-understandable IP addresses, enabling browsers to access websites. Typically operating over port 53/UDP, traditional DNS queries are transmitted in plaintext, making them vulnerable to attacks such as spoofing, hijacking, or surveillance by malicious actors [9]. As a result, DNS is particularly susceptible to censorship by governments and organizations, which often deploy DNS-based firewalls to restrict access to specific domains [10], [9], [3].

Nowadays, **DNS blocking** is employed to censor access to external DNS servers on large-scale networks, and also to provide manipulated or filtered responses [9]. This technique involves either intercepting and altering DNS responses or completely blocking access to unfiltered, external DNS servers, forcing users to rely on censored DNS resolvers. DNS blocking can be implemented at various scales, from organizational networks to national-level internet censorship [8].

2.2 DNS over HTTPS

To address these vulnerabilities, **DNS over HTTPS (DoH)** has been introduced. The purpose of DoH is to encrypt DNS traffic within HTTPS requests, which are typically transmitted over port 443/TCP. By encapsulating DNS queries within HTTPS (c.f. Figure 2.1), DoH allows these requests to be blended with regular HTTPS traffic [4]. This encryption not only enhances user privacy but also makes it significantly more difficult for censors to block DNS queries without disrupting other HTTPS traffic, allowing minimizing interference with the user's browsing experience [7].

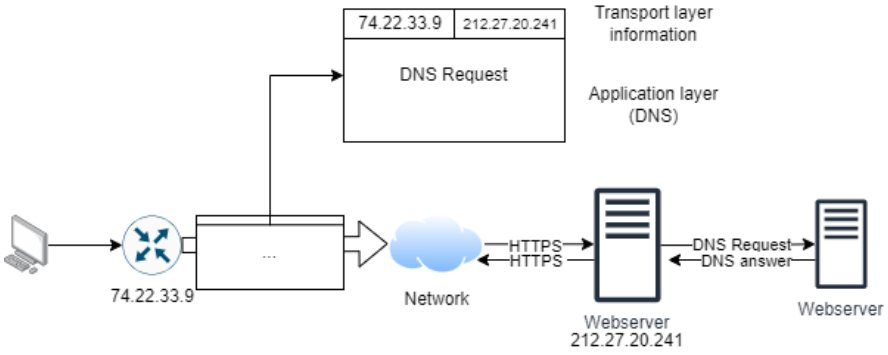

Figure 2.1: Explanation of the DoH Protocol

The **DoH blocking** can be setup by using traditional methods like IP and domain-based blocking. Those methods have proven inadequate due to the increasing deployment of DoH service on unknown server and domains, making it challenging to maintain up-to-date blocklists without disrupting legitimate services.

2.3 Machine Learning Detection

As a result, machine learning (ML) methods have emerged for networking [2], using data-driven models to identify patterns in network traffic such as DoH activity. This

section reviews the key ML models employed in DoH detection, highlighting their capabilities.

A notable advancement in **ML-based DoH detection** is presented in *DNS Over HTTPS Detection Using Standard Flow Telemetry* by Jerabek et al [6]. This study introduces an innovative approach that combines IP-based techniques, machine learning, and active probing to detect DoH traffic using standard flow monitoring features. The authors achieved impressive results, with a classification accuracy of 0.999 and an F1 score of 0.998, notably producing no false positives. Their approach is particularly significant as it relies on standard flow features that can be computed on running sequences, making it deployable in real-world network infrastructures such as intelligent switches, firewalls, or routers. The compatibility with existing network monitoring tools enhances its practical applicability. By demonstrating high accuracy without the need for deep packet inspection, this research addresses both the performance and privacy concerns often associated with DoH detection. The study sets a new benchmark in the field, showcasing the potential of ML techniques to effectively identify DoH traffic while maintaining compatibility with standard network monitoring practices.

Moreover models such as Deep Neural Network (DNN) architectures, including Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), and simple Dense layers, are employed to detect DNS over HTTPS (DoH) traffic. These models have shown strong performance in sequence-based tasks, with LSTMs capturing long-term dependencies in traffic patterns, CNNs detecting local features, and Dense layers learning complex patterns in the data. The flexibility of these architectures enables their application in a variety of domain detection scenarios, from real-time analysis to improving accuracy and minimizing false positives. By leveraging these models, this

research aims to evaluate moving target DoH detection, using models present in the DoHLyzer [1] repository.

2.4 Literature Gap

While machine learning (ML) methods have proven effective in detecting DNS over HTTPS (DoH) traffic, there is a significant gap in the literature concerning moving target DoH systems—such as *NinjaDoH* and their detection. *NinjaDoH* introduces a dynamic approach to DNS over HTTPS by frequently changing server identities and utilizing decentralized technologies like IPFS and IPNS, making detection more challenging. Despite the growing use of DoH to bypass censorship and enhance privacy, existing research predominantly focuses on static DoH server, leaving the adaptability of ML detection models to more advanced and evasive DoH systems largely unexplored.

The **goal** of this thesis is to evaluate the effectiveness of *NinjaDoH*, a dynamic and censorship-resistant DNS over HTTPS (DoH) solution, in evading machine learning-based detection systems. By exploring the limitations of existing DoH detection methods, this work aims to provide insights into how *NinjaDoH* can be a censorship resistant tool.

Chapter 3

Methods

3.1 Adversary Model

The adversary model for *NinjaDoH* assumes that censors implement DNS-based restrictions by blocking access to external DNS resolvers and enforcing the use of controlled DNS servers with filtering rules. Here the censor could be ISPs, an enterprise administrators, or a government authorities and maintains **blocklists of IP addresses or domains** associated with known DoH servers. Furthermore, adversaries may use **machine learning (ML)** techniques to detect DoH traffic based on flow patterns. *NinjaDoH* mitigates this threat by reducing flow durations through frequent IP address changes and obfuscating detection with randomized query paths, complicating efforts to identify its traffic via active probing.

NinjaDoH's efficiency is based on several key assumptions. First, it assumes that **DNS-based censorship** is the primary restriction method, without extensive blocking of website IP ranges or endpoint control such as SSL/TLS interception. **Trusted hyperscalers** are also required to ensure that hosting providers do not collaborate with adversaries. While *NinjaDoH* can be used alongside tools like VPNs or Tor to provide low-latency DNS resolution, it is particularly valuable in environments where these tools are unavailable. Additionally, *NinjaDoH* relies on **private client-server communication**, with shared secrets preventing adversaries from accessing updated

server IPs. Together, these measures allow *NinjaDoH* to bypass censorship and provide resilient access to uncensored DNS services.

3.2 *NinjaDoH*

NinjaDoH is a censorship-resistant DNS over HTTPS (DoH) service using a moving target defense strategy. The main idea behind *NinjaDoH* is to frequently hop from different IP addresses, making it difficult for network-based censorship mechanisms to detect and block it. This dynamic approach is possible thanks to the hyperscaler cloud infrastructure, in our case Amazon Web Services (AWS), and the decentralized capabilities of the InterPlanetary File System (IPFS).

3.2.1 Moving Target Defense

The main principle of *NinjaDoH* is to rotate its IP address periodically, which makes the service a moving target. Here we are using AWS's Elastic Network Interfaces (ENIs) to dynamically swap IP addresses. For AWS, the IPv4 public address pool is estimated to be over 100 million addresses. In our case each EC2 instance has 3 different ENIs. One of the ENI is only dedicated for the management of the DoH server whereas the two others, the primary and alternate, are deployed and deleted, but not simultaneously, allowing continuous IP changes without disrupting ongoing client connections. In our case, the IP rotation occurs every minute; however, additional ENIs can be assigned to the EC2 instance if a shorter rotation interval is required.

3.2.2 Decentralized Updates via IPFS

To distribute the IP addresses updated by *NinjaDoH*, we use the InterPlanetary File System (IPFS) and the InterPlanetary Name System (IPNS). Here, the server publishes

IP address (primary or alternate) as a JSON object on IPFS, and clients resolve the latest IP address using a stable IPNS key. The purpose of this decentralization is to avoid reliance on traditional DNS infrastructure and provide a better resistance to censorship.

3.2.3 Automated Certificate Management

Because of the frequent change of IP, the server requires dynamic SSL/TLS certificate handling. *NinjaDoH* uses a private Certificate Authority (CA) hosted on the server to automatically generate self-signed certificates whenever the IP address changes. After each modification, the reverse proxy (in our case nginx) is reloaded with the new certificate, ensuring a secure connection for the client without relying on external CAs.

3.2.4 Obfuscating Query Paths

To prevent detection through active probing, *NinjaDoH* uses a custom query path instead of the standard */dns-query* path. The query path is based on the IPNS hash and can be more randomized if needed, making it harder for censorship tools to identify the service by its traffic patterns. For this thesis, the randomization is not activated.

3.2.5 Adaptive Client Design

On the client side, *NinjaDoH* uses a Python-implemented program and dnscrypt-proxy for encrypted DNS resolution. This program connects to a local IPFS node to resolve the server's IP address using an IPNS hash. Then, the client updates its configuration and reloads his proxy, ensuring continuous connectivity. The client checks for any server's IP update via IPNS PubSub or at regular intervals.

3.2.6 Censorship Resistance and Availability

By rotating the IP addresses, by using a decentralized IP distribution process, and by handling certificates dynamically, *NinjaDoH* offers a strong resistance against list blocking attempts. Compared to traditional load balancers, it operates on a single server with dynamic IP changes.

3.2.7 System Overview

The figure 3.1 explains how *NinjaDoH* works. Using the IPNS record, the client can retrieve the latest IP server information through IPFS.Each time a DNS request is initiated, the request is sent to the client's localhost DNS proxy, which routes the request to *NinjaDoH*'s current IP. In parallel, each T seconds, the server allocates release the old IP address into the IP pool and reallocate a new one. Then the server publishes this update to IPNS. On his side the client will check if a new IP address is available by retrieving the updates via IPNS. This new IP is used for future DNS queries (until the next IP update).

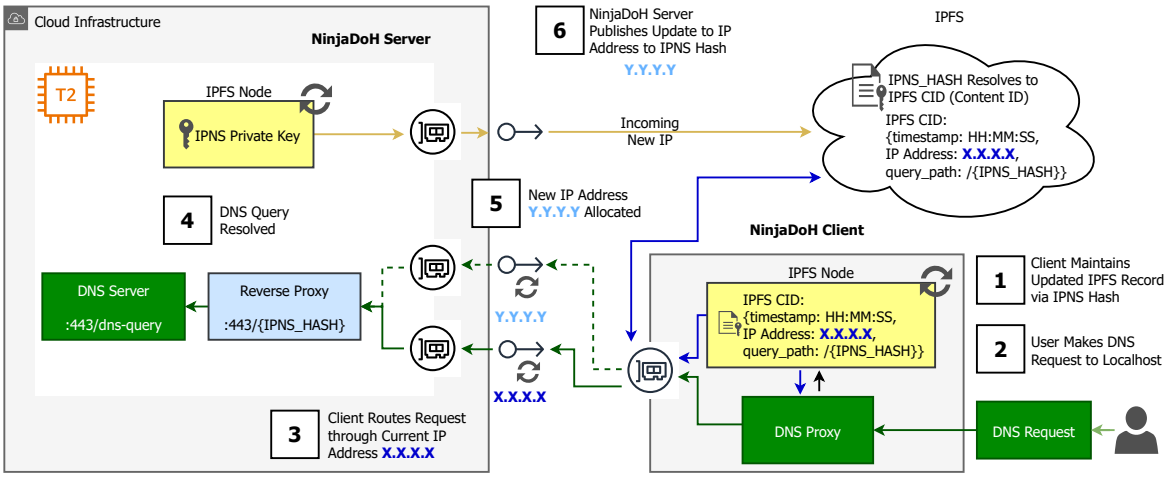

Figure 3.1: Overview of the *NinjaDoH* protocol and architecture.

3.3 Datasets and Flow Stitching

3.3.1 Training and Testing Dataset

The training dataset for the deep neural network (DNN) models was sourced from the *DoHLyzer* repository [1], featuring both DoH and non-DoH traffic samples. For the XGBoost model, training data was derived from the DoH-Gen-F-AABBC database [5], consisting of PCAP files enriched with key TLS attributes such as `TLS_ALPN`, `TLS_JA3`, and `TLS_SNI`. These features provide valuable insights into the encrypted traffic, enabling the model to effectively differentiate between DoH and standard HTTPS flows. Additionally, the dataset incorporates a comprehensive list of known public DoH resolver IP addresses, further aiding in accurate traffic classification and model evaluation.

3.3.2 Flow Stitching

To prepare the dataset for training, individual network packets are transformed into bidirectional flows using **flow stitching**. This process effectively reconstructs the entire communication session by grouping packets that share common characteristics (i.e. source and destination IP addresses, ports, transport protocols, and sequential timestamps). This overview of client-server interactions allows for a more general analysis of the traffic, capturing flow-level behavior and not focus on isolated packets.

For the *DoHLyzer* models [1], the flow stitching and feature extraction were performed using the *meter* tool included in the repository. However, the XGBoost model required the implementation of custom tool designed to reproduce specific features needed for training. Those features are Mean Payload Size (the average size of the

payloads in the flow), Number of Packets transmitted in each direction, Client-to-Server Packet Ratio (measuring the relative activity between client and server), and Mean Time Between Packets (indicating the timing characteristics of the traffic). This ensures that each model receives the relevant input features, increasing their ability to accurately identify DNS over HTTPS (DoH) traffic and non-DoH traffic.

The classification of each flow either DoH or non-DoH is determined using a list of known DoH resolver IP addresses. If the client or server IP address matched with an entry in this list, the corresponding flow is labeled as DoH. This method ensures the labeling of DoH service providers and other known server, providing an effective training and evaluation of the machine learning models.

3.4 Models Selected

The task of identifying DNS over HTTPS (DoH) traffic amidst general HTTPS traffic presents unique challenges, especially due to the encrypted nature of the data. Machine learning (ML) models have become the state-of-the-art solution for distinguishing DoH traffic, using features extracted from network flows. This section provides a detailed overview of the five ML models employed: LSTM-Based Model, Fully Dense Model, CNN-Based Model, Hybrid LSTM-Dense Model from *DoHLyzer*[1] repository and the XGBoost model from *Dns over https detection using standard flow telemetry*[5] paper.

3.4.1 LSTM-Based Model

The LSTM (Long Short-Term Memory) is a model designed for processing sequential data. LSTM networks is efficient in capturing temporal dependencies due to their memory cell structure. This allows them to remember information over long sequences.

In the context of DoH detection, the LSTM model is used to analyze sequences of network flow.

3.4.2 Fully Dense Model

The Dense Model, is a simpler but robust approach for structured data analysis. This model is built with several layers of densely connected neurons and each layer receiving input from all neurons of the previous layer. In the context of DoH detection, the Fully Dense Model utilizes features cited before (cf. part 3.3.2) and are fed into the network, allowing it to learn complex feature interactions.

The primary advantage of this model relies in its simplicity and computational efficiency. Also it serves as a strong baseline for DoH detection, providing a reliable performance metric against which more complex models can be compared.

3.4.3 CNN-Based Model

Convolutional Neural Networks (CNNs) are traditionally used for image processing but in this case, what we are using its remarkable ability in identifying local patterns within structured data to detect DoH patterns in the flows. The CNN-based model applies convolutional filters to the input features, detecting localized patterns that may indicate the presence of DoH traffic or not. For instance, the variations in packet size distributions and the frequency of specific packet sequences can be seen as a pattern and can be effectively captured by convolutional layers. Even if the data is not visual, the concept of spatial relationships can be extended to structured network features.

3.4.4 Hybrid LSTM-Dense Model

The Hybrid LSTM-Dense Model combines the strengths of LSTM and dense neural networks to combine temporal and spatial information of the network flow data. The LSTM component processes sequential data and capture temporal dependencies, while the dense layers handle the feature interactions. This hybrid approach aims to enhance the model's ability to detect complex and encrypted traffic patterns that may not be fully captured by a single model type.

3.4.5 XGBoost (Decision Tree Classifier)

The XGBoost model is a decision tree-based ensemble model mainly known for its efficiency in structured data analysis. It builds multiple decision trees and it combines them to create a predictive model using gradient boosting. The choice of XGBoost is justified by its use in the paper *Dns over https detection using standard flow telemetry*[5]

3.5 Model Training

Two different trainings has been done to evaluate the efficiency of *NinjaDoH* to avoid detection. One is has been done with regular DoH and non-DoH dataset and one with dataset of the first dataset mixed with a *NinjaDoH*' traffic dataset.

3.5.1 Baseline Training

The datasets were split into a training set and a test set to facilitate model training and evaluation (Step 1 and Step 2 in 3.2). The models were first trained on the training set. For the deep neural network (DNN) models, training is conducted using varying flow sequence lengths, ranging from 4 to 10. The choice of sequence length defines

the length of the flow considered in each sequence and directly impact the model's ability to capture temporal dependencies. This parameter is important for models like LSTM, designed to learn from sequential information. By experimenting with different sequence lengths, we aimed to identify the optimal setup that maximized detection performance for each model. The best sequence length for each model is selected based on the highest F1 score achieved, ensuring a balanced trade-off between precision and recall.

3.5.2 Adaptive Adversary Training

The purpose of the adaptive adversary training is to train the models using a dataset where *NinjaDoH* is known. Here, the adversary has access to the equivalent of 15 minutes of browsing involving around 10 IP rotations. This sample of *NinjaDoH*'s traffic is mixed with the initial datasets as shown on the figure 3.2 with the dashed arrow. As the baseline training, the data are once again split into a training set and a test set and the training is perform on the training dataset.

3.6 *NinjaDoH* Evasion Evaluation Setup

3.6.1 Evaluation Dataset

To perform the evaluation of *NinjaDoH*, a 3 minutes browsing `PCAP` file has been captured. This sample represents a 3 IP rotations and mocks the behavior of a normal user using over internet. The evaluation dataset for both experience is the same. This dataset is stitched for each model using the same tools cited above 3.3.2.

3.6.2 Evaluation Metrics

The evaluation of the machine learning models was based on a comprehensive set of **metrics**, including precision, recall, and F1 score. The goal is to evaluate their effectiveness in distinguishing DNS over HTTPS (DoH) traffic from standard HTTPS and non-DoH traffic.

The *precision* defined as the ratio of true positives to the total positive predictions. This highlights the model's accuracy in identifying DoH traffic and emphasize the reliability of its positive detections.

$$Precision = \frac{TP}{TP + FP} \tag{3.1}$$

The *recall*, or sensitivity, measures the proportion of true positive predictions out of all actual DoH instances. It indicates the model's capability to detect relevant traffic and minimize missed detections.

$$Recall = \frac{TP}{TP + FN} \tag{3.2}$$

The *F1 score* is the harmonic mean of precision and recall. IT provides a balanced measure of performance and particularly, useful in cases with imbalanced datasets.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3.3}$$

These combined metrics offer a nuanced and robust evaluation of the models' capabilities and help to determine their effectiveness in detecting DoH traffic while balancing detection accuracy with the minimization of false positives.

3.6.3 Evaluation Process

The evaluation of the models compare the evaluation on the existing testing dataset and on the *NinjaDoH* evaluation dataset (Step 2 in 3.2). The purpose of those two evaluations is to compare the performance of the models on the two type of DoH traffic.

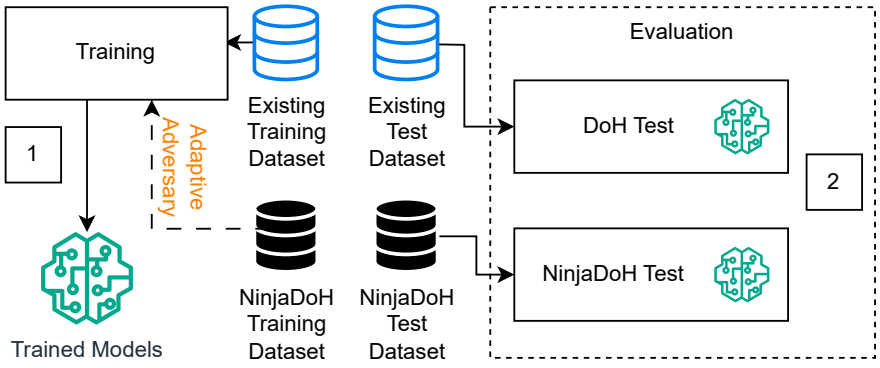

Figure 3.2: Training and Evaluation Process for the Baseline and the Adversary models.

Chapter 4

Results

4.1 Baseline Model Performance

The results of the evaluation on the baseline training data and adversary data showed high consistency with benchmarks from prior work, achieving F1 scores, precision, and recall metrics in the range of 0.98 to 0.99 (cf. solid bars in the Figures 4.1 and 4.2). These results demonstrate the strong effectiveness of the models used in accurately detecting DNS over HTTPS (DoH) traffic.

4.2 *NinjaDoH* Detection Results

4.2.1 Baseline Model Detection

The figure 4.1 presents the evaluation of the *DoLyzer* and the XGBoost models and provides a balanced summary that accounts for class imbalances.

The LSTM model achieved a weighted average precision of 0.227, recall of 0.472, and F1 score of 0.306.

The Dense model showed modest improvement, with weighted averages of 0.263 for precision, 0.510 for recall, and 0.347 for F1 score.

The CNN and hybrid models exhibited similar performance, with a weighted precision of 0.284, recall of 0.529, and F1 score of 0.369.

Finally the XGBoost model performed a weighted precision of 0.305, recall of 0.536, and F1 score of 0.410.

The results indicate difficulties in accurately identifying DoH and non-DoH traffic, with a **high rate of false negatives** impacting overall user performance. The low precision across all models points the challenge of reducing false positives.

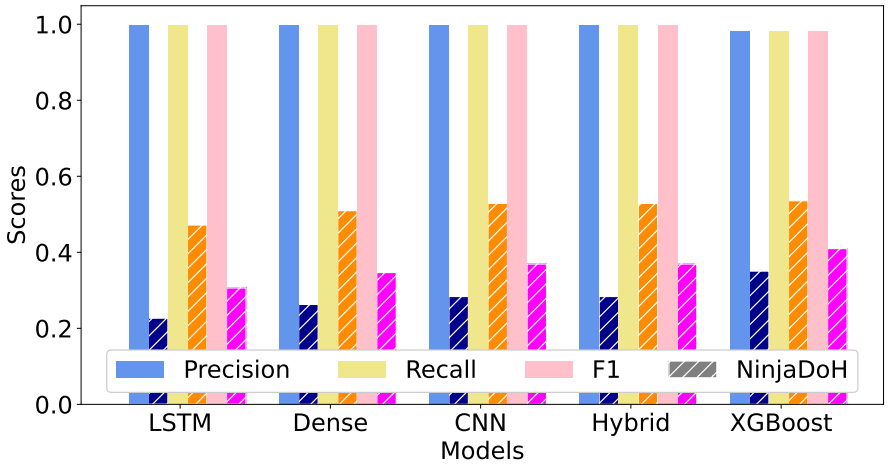

Figure 4.1: Performance of the adversary models trained with the 'benign' DoH traffic

4.2.2 Adversarial Model Detection

To compare, the figure 4.2 presents the evaluation of the *DoLyzer* and the XGBoost models trained using the adversarial dataset.

Here LSTM model achieved with a weighted precision of 0.764, recall of 0.635, and F1 score of 0.578, demonstrating robust detection capabilities and a relatively balanced trade-off between precision and recall.

The Dense model showed strong precision (0.713) but low recall (0.504), leading to an F1 score of 0.368, highlighting issues in consistently detecting evasive traffic patterns.

The CNN model's performance was mixed, with a precision of 0.471, recall of 0.532, and an F1 score of 0.377, reflecting its struggle to balance detection accuracy, particularly against adversarial traffic.

The hybrid model achieved strong scores (precision: 0.745, recall: 0.530, F1 score: 0.418) suggesting difficulties in recall and a high rate of false negatives.

Finally the XGBoost model performed a weighted precision of 0.429, recall of 0.559, and F1 score of 0.431.

Here we can see that the performances improved for all the model types that we evaluated with the *NinjaDoH* traffic present in the training set. The results indicate that while precision was generally strong across the models, the low recall rates suggest challenges in handling diverse and *NinjaDoH* traffic.

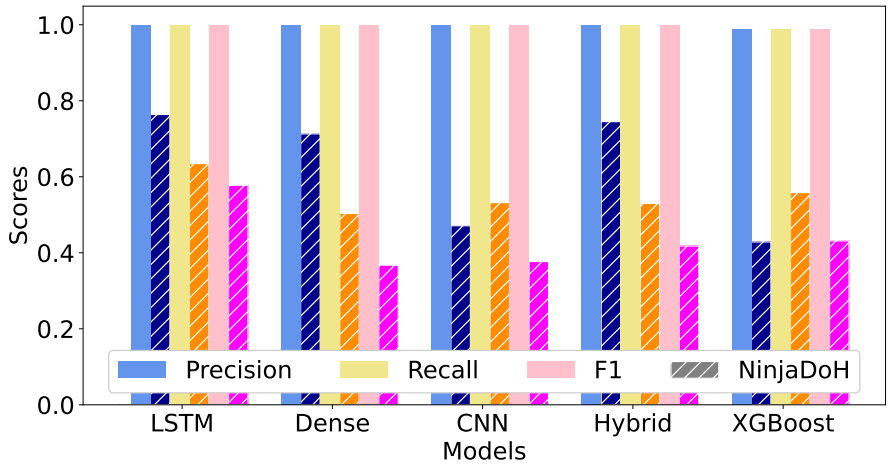

Figure 4.2: Performance of the adaptive adversary models trained with 'benign' DoH traffic and *NinjaDoH* 's traffic

4.2.3 Comparison

The table 4.2.3 presents a comparison of the detection model performance when trained with and without *NinjaDoH* traffic in the training dataset. Incorporating *NinjaDoH* traffic has led to an improvement across all evaluation metrics. The precision increased

from 0.227 to 0.764, indicating a significant **reduction in false positives** and a more accurate identification of DoH traffic. The recall also improved, rising from 0.472 to 0.635, demonstrating that the model enhance his ability to detect a higher proportion of true *NinjaDoH* DoH instances. Consequently, the F1 score, which balances precision and recall, rose from 0.369 to 0.578.

These results highlight the importance of including adversarial *NinjaDoH* traffic during training, as it enables the models to better learn and adapt to dynamic, evasive traffic patterns, improving overall detection effectiveness.

Metric	**Without *NinjaDoH* Traffic**	**With *NinjaDoH* Traffic**
Precision	0.305	0.764
Recall	0.536	0.635
F1-Score	0.410	0.578

Table 4.1: Comparison of Detection Models Trained With and Without *NinjaDoH* Traffic in Training Data

Chapter 5

Discussion

5.1 Analysis of *NinjaDoH*'s Effectiveness in Evading ML-Based Detection

The **evaluation of the models** with and without *NinjaDoH* traffic in the training dataset provides valuable insights of how the system resists machine learning-based detection and its impact on user experience.

First, the results indicate that adding *NinjaDoH* traffic during training significantly improves the detection capabilities of all models, as seen in Figure 4.2 and Table 4.2.3. Especially, the weighted precision increased from 0.305 to 0.764, showing a reduction in false positives and a more accurate identification of *NinjaDoH* DoH traffic. This improvement is crucial, as high precision minimizes the rate of incorrect classifications, reducing the likelihood of disrupting non-DoH traffic.

However, the increase in recall is less pronounced, rising only from 0.536 to 0.635. This proves that even if the models became better when adversarial *NinjaDoH* traffic is included, they still struggled with identifying all instances of evasive DoH traffic. The relatively low recall also suggests that some *NinjaDoH* traffic patterns could still evade detection.

Then, the overall F1 score improvements— from 0.410 to 0.578 —highlight a better balance between precision and recall after including *NinjaDoH* traffic in training. This

indicates that the models can adapt to the dynamic and evasive patterns of *NinjaDoH*, increasing their ability to detect DoH flows. However, the need for this increased adaptability may also point to the potential challenges posed by *NinjaDoH* in generating traffic patterns that differ significantly from standard DoH, making them harder for the models to learn without specific training.

Moreover, for both evaluations, the randomization of the query path is not activated. This is why the adversary model performs better. By enabling the randomization path, the entropy of the queries will increase, thereby improving the evasion of *NinjaDoH*.

Regarding **user experience**, the significant increase in precision suggests a reduced rate of false positives, which is critical for minimizing disruptions to legitimate traffic. However, the modest recall improvement raises concerns about the models' effectiveness in consistently detecting all *NinjaDoH* traffic. This partial evasion capability could potentially allow *NinjaDoH* to operate without being fully detected, achieving its goal of resisting censorship.

5.2 Scalability of ML-Based DoH Detection

Whereas that the deployment of *NinjaDoH* seems achievable and largely scalable (everybody can own his own *NinjaDoH* server on AWS), the scalability of the detection remains a significant concern.

Indeed, detecting DoH traffic in large-scale networks using ML models faces significant scalability challenges, particularly against *NinjaDoH*. As network scale increases, the feasibility of this approach diminishes due to the growing data volume and computational demands. To be effective, an adversary must capture enough packets to reconstruct flows, extract features, and perform blocking before *NinjaDoH* rotates to

a new IP address—every 60 seconds in our case, though this interval can vary depending on the number of ENIs. Flow arrivals are modeled as a Poisson process, and the inter-arrival times are exponentially distributed at a rate λ (flows per second), while flow durations follow a log-normal distribution, reflecting the positive skew seen in real data. For example, *NinjaDoH* flows have a mean duration of 742 ms, with a median of 246 ms, compared to non-DoH flows with a mean of 7,430 ms and a median of 313 ms. These statistics highlight the challenge of processing short-lived *NinjaDoH* flows in real-time. The adversary must process each flow within $T_{rotation}$, considering processing times per flow (T_p) of 0.1 ms, 1 ms, or 10 ms.

The simulation of the results reveals that the probability of detecting DoH traffic $P_{DetectDoH}$, given by formula 5.1, is inversely related to network scale and processing constraints. For a system with $\lambda = 10,000$ flows per minute and $T_p = 1ms$, achieving a detection probability near the true positive rate ($TPR = 0.52$) requires at least 24 processor cores. However, as λ increases to 1,000,000 flows per minute, even with $N_{proc} = 29$, detection probability declines drastically due to processing delays. Moreover, the short live IP rotation counter the adversary if he manages to detect and block an IP. The imbalance between the adversary's high costs to maintain a and *NinjaDoH*'s minimal disruption of the user experience shows the infeasibility of ML-based detection in large-scale networks.

$$P_{DetectDoH} = min(\frac{N_{proc}}{\lambda T_p}) * TPR \tag{5.1}$$

5.3 Limitations and Future Work

This thesis demonstrates the potential of *NinjaDoH* in evading ML-based detection but there is several limitations. The use of a *IP whitelisting blocking* will this prevent

every use of DoH server with a non allowed IP address. A future work could be to work on the obfuscation of *NinjaDoH* inside necessary servers and services. Also here the evaluation has been done in a restricted environment. A real-world deployment with diverse traffic sources would provide a better comprehensive of *NinjaDoH* efficiency.

Chapter 6

Conclusion

This work presents an evaluation of *NinjaDoH* , a censorship-resistant DNS over HTTPS (DoH) service designed to evade detection and maintain accessibility in environments where DNS-based restrictions are applied. By using dynamic IP rotation via hyperscalers, decentralized IP distribution via IPFS, and techniques like randomized query paths, *NinjaDoH* effectively counters adversarial tactics such as IP/domain blocking and machine learning-based detection. The experimental results demonstrate that state-of-the-art ML models face significant challenges in detecting *NinjaDoH* traffic, especially in large-scale networks. The short flow durations, frequent IP rotations, and computational overhead of real-time detection make widespread ML-based censorship efforts impractical.

The findings highlight the considerable investment of resources the adversary needs to provide to maintain a high detection accuracy and to minimize the disruption of the user experience. This research underscores the potential of moving target defense strategies in the domain of encrypted DNS such as DoH. *NinjaDoH* represents a step forward in the protection of internet freedom and provides a robust framework for resisting censorship while ensuring a secure and private access to information.

Reference List

[1] Ahlashkari, A., 2023: Dohlyzer: Deep learning models for detecting dns over https (doh) traffic. Accessed: 2024-11-16, https://github.com/ahlashkari/DoHLyzer.

[2] Boutaba, R., M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, 2018: A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *Journal of Internet Services and Applications*, **9 (1)**, 16, https://doi.org/10.1186/s13174-018-0087-2, URL https://doi.org/10.1186/s13174-018-0087-2.

[3] Hoang, N. P., and Coauthors, 2021: How great is the great firewall? measuring china's {DNS} censorship. *30th USENIX Security Symposium (USENIX Security 21)*, 3381–3398.

[4] Hoffman, P. E., and P. McManus, 2018: No. 8484, Request for Comments. DNS Queries over HTTPS (DoH). RFC Editor, URL https://www.rfc-editor.org/info/rfc8484, RFC 8484, https://doi.org/10.17487/RFC8484.

[5] Jeřábek, K., K. Hynek, T. Čejka, and O. Ryšavỳ, 2022: Collection of datasets with DNS over HTTPS traffic. *Data in Brief*, **42**, 108 310.

[6] Jerabek, K., K. Hynek, O. Rysavy, and I. Burgetova, 2023: Dns over https detection using standard flow telemetry. *IEEE Access*, **11**, 50 000–50 012, https://doi.org/10.1109/ACCESS.2023.3275744.

[7] Kosek, M., L. Schumann, R. Marx, T. V. Doan, and V. Bajpai, 2022: Dns privacy with speed? evaluating dns over quic and its impact on web performance. *Proceedings of the 22nd ACM Internet Measurement Conference*, 44–50.

[8] Lvovich, Y. E., A. P. Preobrazhenskiy, Y. P. Preobrazhenskiy, and Y. Klimenko, 2022: The analysis of software and hardware solutions for blocking prohibited information in information and telecommunication networks. *Informacionnye Tehnologii*, URL https://api.semanticscholar.org/CorpusID:251601588.

[9] Master, A., 2023: Modeling and characterization of internet censorship technologies. Ph.D. thesis, Purdue University.

[10] Wikimedia Foundation, 2023: Wikimedia Foundation urges Pakistan Telecommunications Authority to restore access to Wikipedia in Pakistan. https://wikimediafoundation.org/news/2023/02/03/wikimedia-foundation-urges-pakistan-telecommunications-authority-to-restore-acc\\ess-to-wikipedia-in-pakistan/.