Detecting Malicious DNS over HTTPS Traffic Using Machine Learning

Sunil Kumar Singh
School of Computer Science and Engineering
VIT-AP University, Near Vijayawada
Andhra Pradesh, India
sksingh.cse@gmail.com

Pradeep Kumar Roy
Department of Computer Science and Engineering
Indian Institute of Information Technology, Surat
Gujarat, India
pkroynitp@gmail.com

***Abstract*—Network with the internet has grown-up very faster compared with any other technology around the world. From the beginning of the Internet, the Domain name system (DNS) is an integral and important part of it. The primary task of DNS is to redirect the users at correct computers, applications, and files by mapping IP and domain name. Due to certain security flaws of DNS, it is always a major attack target for attackers like DNS-based malware, DNS-amplification, false-positive triggering, DNS tunneling, etc. DNS over TLS (DoT) and DNS over HTTPS (DoH) are recently developed and deployed by Google and Cloudflare to prevent these types of attacks. DoT and DoH are the standard protocols which mainly designed for privacy and security by encrypting the DNS traffic between users and DNS resolver servers. This paper uses various machine learning classifiers such as (i) Naive Bayes (NB), ii) Logistic Regression (LR), iii) Random Forest (RF), (iv) K-Nearest Neighbor (KNN), and (v) Gradient Boosting (GB) to detect the malicious activity at DNS level in the DoH environment. The experiments are conducted on a benchmark MoH dataset (CIRA-CIC-DoHBrw-2020). Several features are used to develop a robust model. The experimental outcome confirmed that the RF and GB classifiers are better choices for the said problem. Since, majority of the malicious activity detected by the developed model, it can be said that the ML-based algorithms are a better option for the prevention of DNS attacks on DoH traffic.**

***Index Terms*—DNS, DNS-over-HTTPS (DoH), Machine Learning, DNS encryption, DNS over TLS (DoT), DNS Security, ML classifiers**

I. Introduction

Domain Name System (DNS) supremacy on all network applications like email, web browsing, e-commerce, VoIP, etc. has made it vulnerable to cyber-attacks. The primary task of DNS is to change the domain name into the corresponding IP address with the help of a resolver [1]. The simple DNS query is generated at a system will go to a resolver. This resolver will inquire to different servers for the allocated IP address, and the response will be sent back to your computer. DNS not only works on a single server, but a large number of servers are also arranged hierarchically, and these servers are spread over the world. The root DNS server, TLD, and authoritative DNS are the three basic servers where TLD servers consist of generic domain names like gov, edu, com, org. TLD server gives the record to the corresponding authoritative DNS server, which maintains the domain. Finally, the authoritative DNS server returns the IP address of the website [2]. DNS is vulnerable to many attacks due to its important functions. There are different types of attacks we generally see on DNS majorly, DNS amplification, DNS cache poisoning, DNS hijacking, NXDomain attack, and DNS tunnelling [3] [4] [5] [6]. As per the EfficientIP and IDC DNS threat report 2020, nearly 79% of the surveyed organizations have experienced DNS attacks in 2020, with the average cost of each attack stood at around USD 924,000.

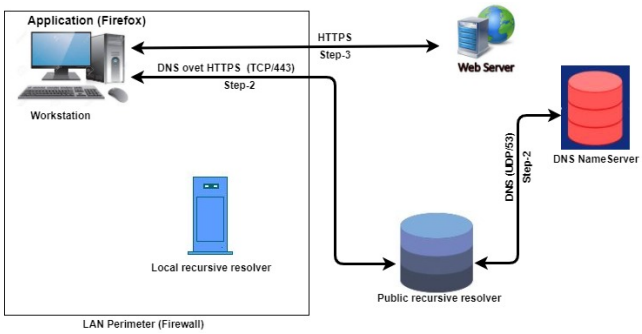


Figure 1: DNS over HTTPS (DoH) Process

Recently, two new protocols: DNS-over-TLS (DoT) and DNS-over-HTTPS (DoH) are designed using encryption techniques in 2016 and 2018 respectively. The DNS data encrypted with TLS and HTTPS between the computer and resolver. During processing any application, this DoH bypasses the traditional DNS. Figure 1 demonstrates the complete DoH process for a DNS request [7]. An application can directly give a DNS request to a public recursive resolver (step-1), and this public DNS resolver makes a query to the name server (step-2). Finally, the Name server returns the required information to the application. Now with this received DNS record, that application can access the web server by IP address. Google and Cloudflare have implemented DoH using public DNS resolver, and Mozilla Firefox is the first browser to implement it. In the USA, Mozilla has enabled DoH and makes default for the user in February 2020. Presently, most popular browsers support this DoH like Google Chrome, Edge, Firefox, etc. [8]. As per the latest news, Microsoft has also tested it at Windows 10 operating system platform;

978-1-7281-9673-2/20/$31.00 ©2020 IEEE

Insiders Fast Ring builds supports only three DoH resolvers at the moment (Cloudflare, Google, Quad9). The release of stable DoH support can be expected in the near future from Microsoft. Therefore, rapid growth can be expected in the use of DoH very soon.

But, as we know that every technology has two sides, positive and negative. Hence, apart from high privacy protection, DoH has plenty of security risks due to less visibility of local security tools and applications. This is the main motivation for this work to analyze the traffic of DoH. A Godlua malware was recognized using the DoH encryption method for freely establish communication with the command and control (C&C) server in July of 2019 [9]. Haddon et al. [3] discussed the major possible ways of data ex-filtration using DoH. Although the DoH is an encryption-based service, it has several security issues and privacy issues. The major issues about the DoH are as follows:

- It bypasses the local security measures like Firewalls, IDS et.
- DNS traffic's audit is not possible.
- DNS name may not stay confidential.
- Detecting threats are more complex.
- Technical support and troubleshooting change significantly due to new applications and a different DNS resolver.
- DNS blocking can not be performed

In this paper, we tried to detect the DoH threats using DoH malicious and benign traffic datasets to fulfill our motivation. The dataset was generated with the help of DoH traffic by accessing the top 10k Alexa websites. Mozilla Firefox and Google Chrome web browsers and DNS tunneling tools that support DoH protocol is also used for benign-DoH and malicious-DoH traffic. Figure 2 shows how datasets (benign-DoH and malicious-DoH) was captured [1]. This research aims to predict the malicious and benign DNS requests in DoH with different ML classifiers after the training on the mentioned datasets.

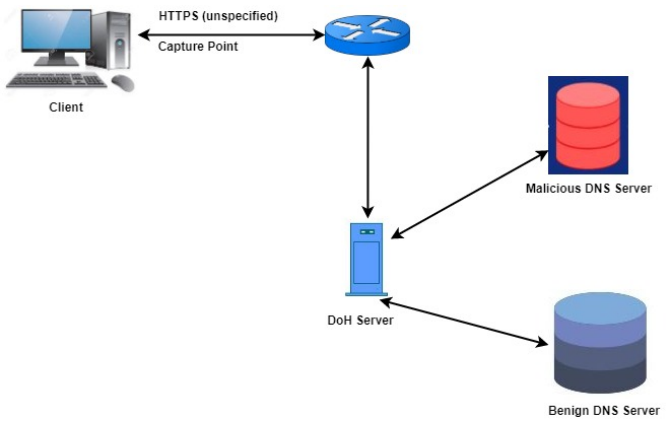


Figure 2: Dataset capturing process

[1]https://www.unb.ca/cic/datasets/dohbrw-2020.html

The rest of the article organised as follows: Section II discussed the existing work. In section III, methodology is discussed. Section IV discusses the results and finally section V concludes the paper with limitation and future scope.

II. Related Works

Since DoH is a very new technique in the DNS field, therefore only a few papers have been published till now. Houser et al. have developed a DoT fingerprint method to analyze DoT traffic [10]. This method determines the genuine request of a DNS by differentiating between a real user and an adversary. They conclude in their findings that even in encrypted DoT also information leakage possible. A performance analysis based on the security and privacy of DoH was discussed by Borgolte et al. [11] in their research. This paper mainly deals with security issues and privacy policies of DoH and it is not present any analysis of encryption at the network level. Siby et al. [12] presented a new feature set to perform the attacks related to DoH in their research. Their analysis claim that padding methods are efficient but it is not enough when attackers are more resourceful.

Hjelm a graduate research student of the SANS institute provided complete details about the DoH detection by applying a real intelligence threat analysis framework [13]. During his study in labs, he has tested several times to bypass basic security controls by DoH. A popular and DoH enable browser Mozilla Firefox is used for browsing various news and entertainment websites. He has used Zeek IDS tools to create logs file based on the network traffic. Finally, checks in real intelligence threat analysis framework by passing DoH request that it is from a usual or unusual client. The study and identification of encrypted DoH traffic have done by Bushart et al. [14]. They have used some important feature vector for finding the performance of the classification. It also tries to identify the DoH content but it is possible only through known IP addresses of popular services. Vekshin et al. provide ML classification models to classify the DoH request and non-DoH request [15]. They have used five ML classifiers in their experiment and achieved 99.9% accuracy to detect DoH traffic. This paper only classifies the DoH traffic and traditional DNS traffic in all five ML classifier. One more paper has been presented in a conference in Aug 2020 related to DoH and ML but till now this paper is not published [16].

To the best of our knowledge, no research work has been done for detecting malicious activity for DoH traffic to date. However, it is very much needed in the current scenario to provide privacy and security to Internet users. This paper addresses this issue using machine learning. The methodology to detect malicious activities with ML algorithms and their outcomes are discussed in consecutive sections.

III. Methodology

This research aims to detect malicious activity over the DoH traffic. Complete steps followed to develop the model for the said problem is shown in Figure 3. To do so, we have used

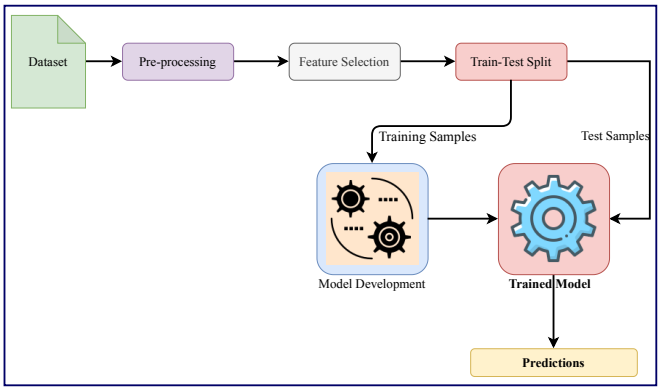


Figure 3: Framework for the malicious event detection

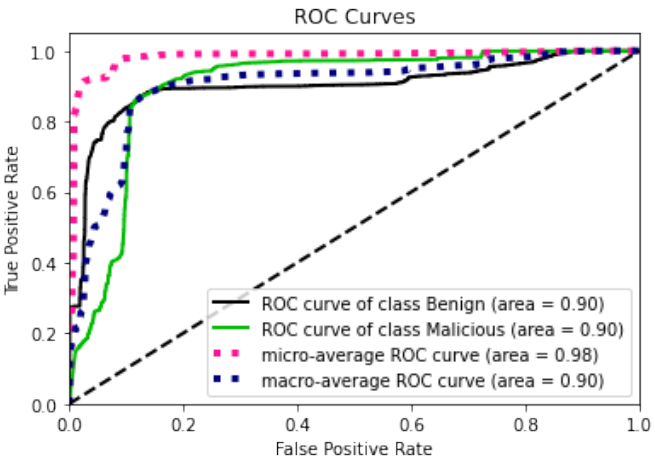


Figure 4: AUC curve using Naive Bayes Classifier

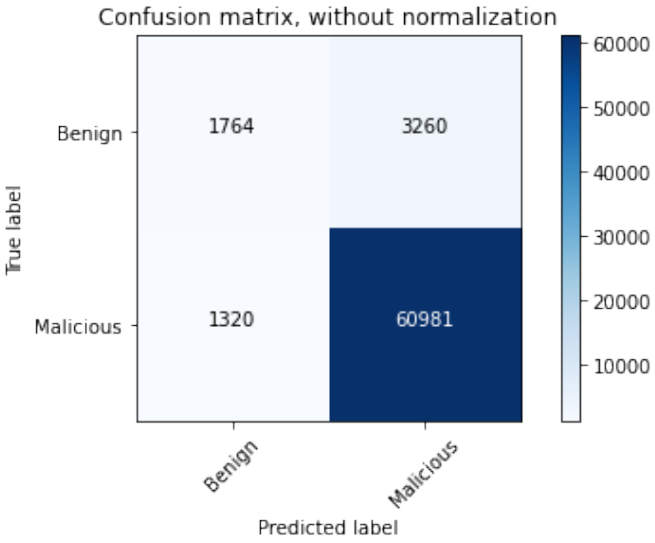


Figure 5: Confusion Matrix using Naive Bayes Classifier

a recently developed dataset that was shared publicly [2]. The dataset contained two separated CSV file namely: *Benign.csv* and *Malicious.csv*. In *Benign.csv*, a total of 19,807 number of instances are present whereas *Malicious.csv* contained 249,836 number of instances. We have merged it and make a combined file. The combined data is preprocessed and removed the rows having null attributes. After the removal of null attributes, a total of 269,299 samples are present in the combined dataset. To extract the important and efficient feature from the datasets, we have used DoHMeter tools [3]. This tool is developed in Python and freely available for extracting the feature from the captured PCAP files. It gives CSV files as an output.

Several features selected from the combined dataset for further processing. The selected features are listed below, due to page limitation the feature description is not added here however, it can be access from source of dataset[4]: *'SourcePort', 'DestinationPort', 'Duration', 'FlowBytesSent', 'FlowSentRate', 'FlowBytesReceived', 'FlowReceivedRate', 'PacketLengthVariance', 'PacketLengthStandardDeviation', 'PacketLengthMean', 'PacketLengthMedian', 'PacketLengthMode', 'PacketLengthSkewFromMedian', 'PacketLengthSkewFromMode', 'PacketLengthCoefficientofVariation', 'PacketTimeVariance', 'PacketTimeStandardDeviation', 'PacketTimeMean', 'PacketTimeMedian', 'PacketTimeMode', 'PacketTimeSkewFromMedian', 'PacketTimeSkewFromMode', 'PacketTimeCoefficientofVariation', 'ResponseTimeTimeVariance', 'ResponseTimeTimeStandardDeviation', 'ResponseTimeTimeMean', 'ResponseTimeTimeMedian', 'ResponseTimeTimeMode', 'ResponseTimeTimeSkewFromMedian', 'ResponseTimeTimeSkewFromMode', 'ResponseTimeTimeCoefficientofVariation'*.

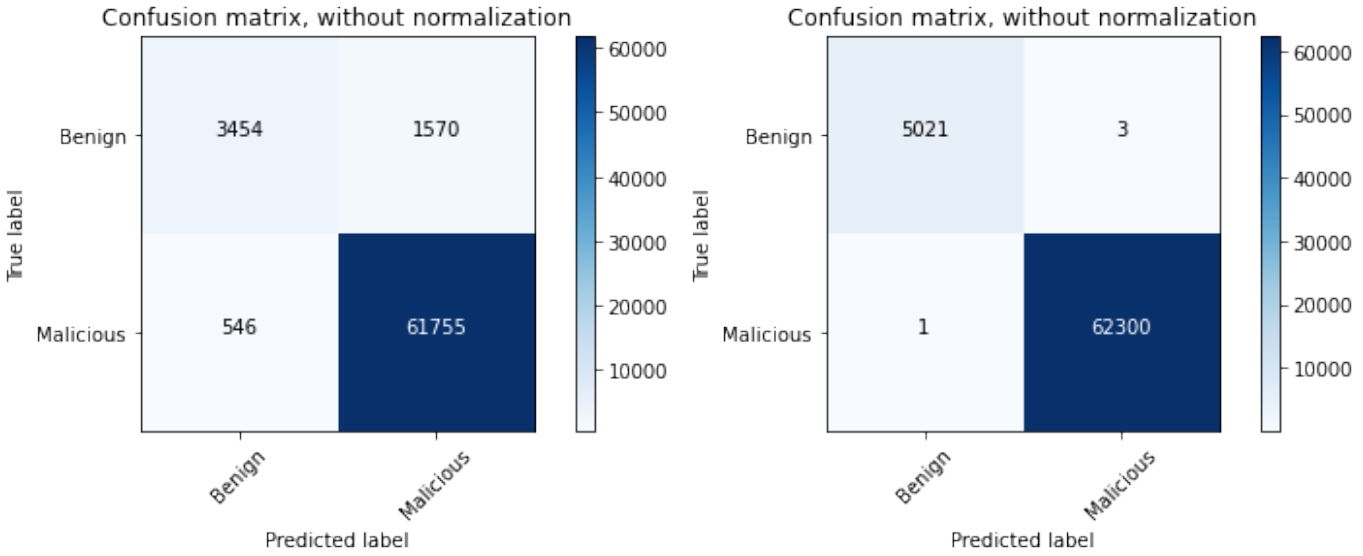
Further, to develop the model and check their performance, we have split the dataset into train and test in 3:1 ratio, i.e., for training, the model 201,974 samples whereas for testing 67,325 samples are used to test the performance. There were many combinations of train-test split ratios used by researchers such as 2:1, 3:1, and others. However, our experimental results confirmed that the 3:1 train-test split ratio is the best choice for the said problem. For model development, we have mainly used five classifiers namely, i) Naive Bayes (NB) [17], ii) Logistic regression (LR) [18], iii) Random Forest (RF) [19], (iv) K Nearest Neighbour (KNN) [20], and v) Gradient Boosting (GB) [21]. The extensive results obtained using these classifiers are discussed in section IV.

[2]https://www.unb.ca/cic/datasets/dohbrw-2020.html
[3]https://github.com/ahlashkari/DoHlyzer
[4]https://www.unb.ca/cic/datasets/dohbrw-2020.html

IV. Results

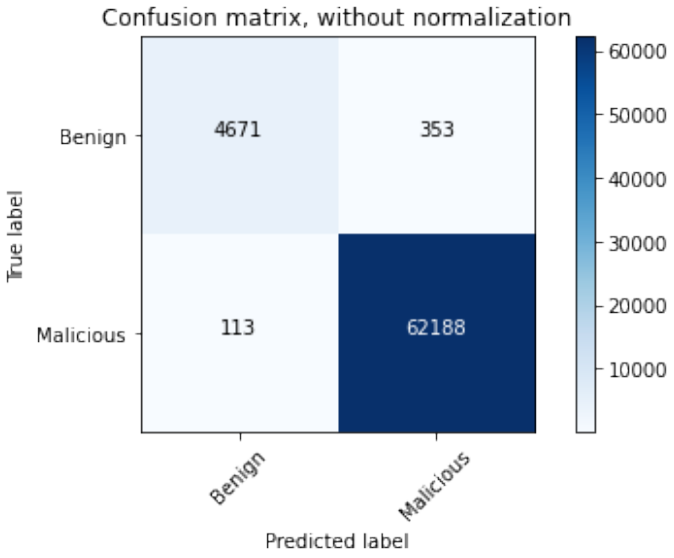
In this section, we presented the results obtained using different machine learning-based classifier models such as based classifiers such as NB, LR, RF, KNN, and GB. To evaluate the models performances, we used metrics such as Precision (Eq. 1), Recall (Eq. 2), and *F1*-Score (Eq. 3).

- Precision: The fraction of *malicious* event among the

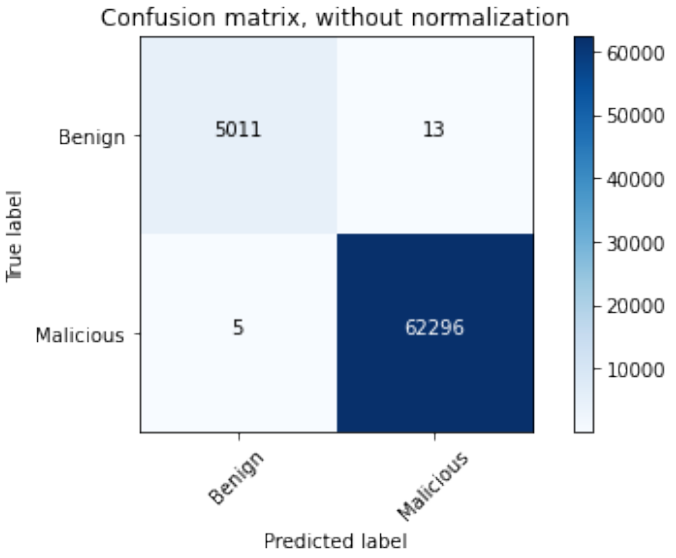


(a) Logistic Regression

(b) Random Forest



(c) KNN



(d) Gradient Bossting

Figure 6: Confusion Matrix using different classifiers

retrieved *malicious* event. It is computed as:

$$Precision\ (P) = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

- Recall: It is the fraction of *malicious* event that have been identified from the total number of *malicious* event present.

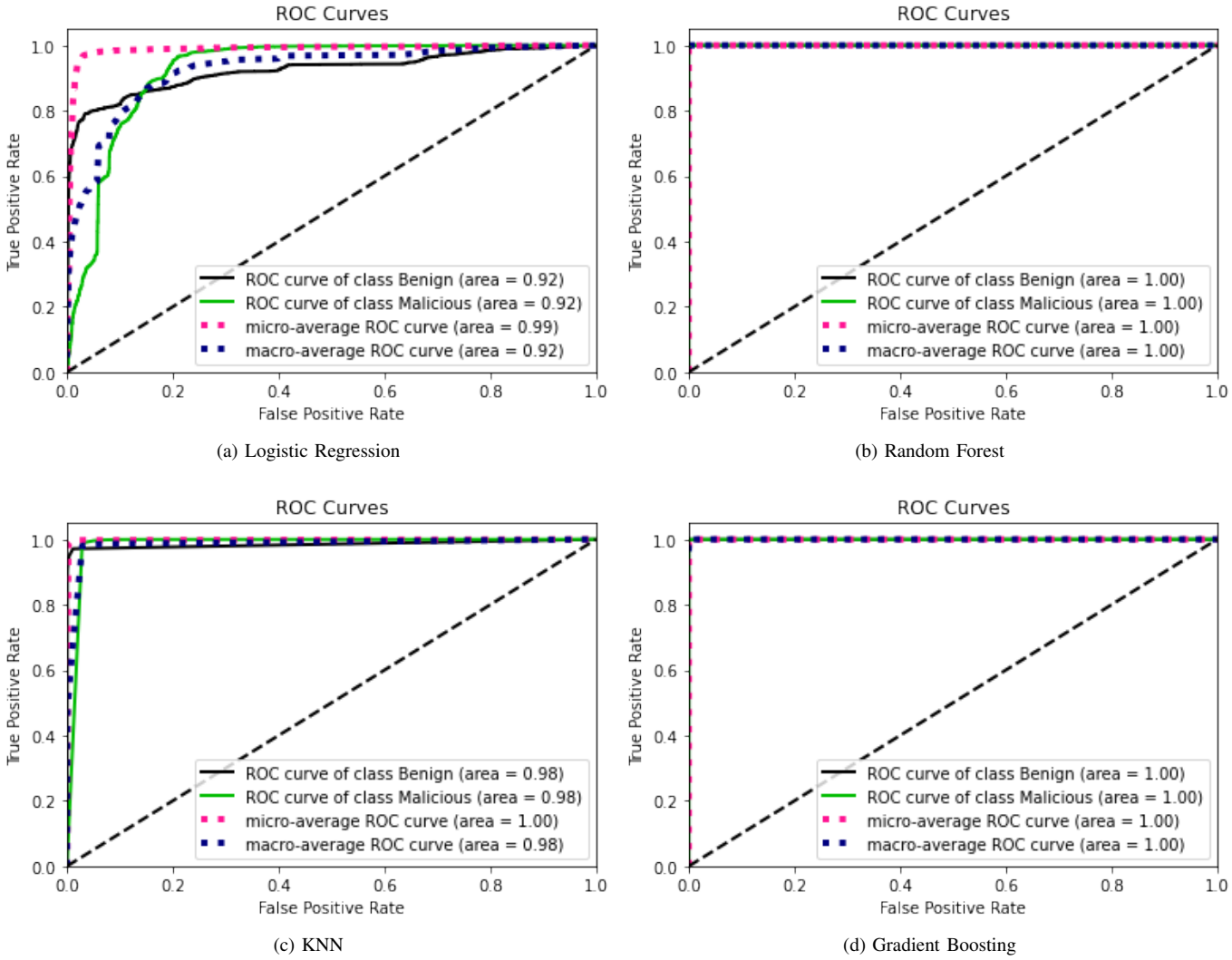
$$Recall\ (R) = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

where $True\ Positive$ means *malicious* event predicted as *malicious*, $False\ Negative$ means *malicious* event predicted as *non-malicious* questions, and $False\ Positive$ means *non-malicious* event predicted as *malicious* event.

- *F1*-Score: It is the harmonic mean of *P* (Eq. 1) and *R* (Eq. 2).

$$F1 - Score\ (F1) = 2 * \frac{P * R}{P + R} \quad (3)$$

The experiment started with an NB classifier. The training samples used to train the classifier. Then on test data, the performance of the trained model evaluated in terms of precision (P), recall (R), F1-measure (F1), and the AUC value. The P, R, and F1 for malicious activity detection are 0.95, 0.98, 0.96 and for non-malicious activity are 0.57, 0.35,0.44, (TableI), the AUC value for the same is 0.90 as shown in Figure 4.



(a) Logistic Regression

(b) Random Forest

(c) KNN

(d) Gradient Boosting

Figure 7: AUC Curve using different classifiers

Table I: Result using Naive Bayes Classifier

Class	*P*	*R*	*F1*
Benign	0.57	0.35	0.44
Malicious	0.95	0.98	0.96

The confusion matrix obtained using the NB classifier is shown in Figure 5. As can be seen from Figure 5, the majority of malicious activity is successfully detected by the model. Out of 62,301 test samples, 60,961 samples (97.84%) are successfully detected; however, for the Benign event, out of 5024 test samples, only 1764 (35.11%) are truly classified, remaining 64.88% test samples are miss-classified and predicted as the malicious event.

To develop a more accurate model and reduce the miss-classification rate for all classes, we used another set of classifiers, namely: Logistic Regression (LR), Random Forest (RF), K-Nearest Neighbors (KNN), and Gradient Boosting (GB). The performance of these classifiers is measured with the same metrics which were used for the NB classifier. The complete observations are presented in Table II.

Table II: Results using different classifiers

Classifier	Class	*P*	*R*	*F1*
LR	*Benign*	0.86	0.69	0.77
	Malicious	0.98	0.99	0.98
RF	*Benign*	1.00	1.00	1.00
	Malicious	1.00	1.00	1.00
KNN	*Benign*	0.98	0.93	0.95
	Malicious	0.99	0.99	0.99
GB	*Benign*	1.00	1.00	1.00
	Malicious	1.00	1.00	1.00

As can be seen from Table II, the performance of the selected classifiers are boosted as compared to the NB classifier. Among all, the best performance in terms of P, R,

and F1-measure is yielded by RF with 99.99% accuracy. The confusion metrics using these classifiers are shown in Figures 6a, 6b, 6c, 6d.

From the above observations, we can say the RF classifier performs best for the selected problem where P, R, and F1-measure for both the classes, i.e., Benign and Malicious, is 1.00. The same can be seen from the confusion matrix also 6b, where only 3 samples of Benign class and 1 sample of Malicious class are miss-classified. The AUC value obtained using the RF classifier is also 1 (Figure 7b). A closer result also produced by the GB classifier, where the miss-classification rate is near to 0 (Figure 6d), also the AUC value are same as RF classifier (Figure 7d). The other two classifiers, such as LR and KNN, not yielded good accuracy. As shown in Figure 6a, 1570 and 546 test samples of Benign and Malicious are miss-classified using the LR classifier. Similarly, with the KNN classifier, 353 and 113 samples of Benign and Malicious are miss-classified (Figure 6c). The AUC value of LR and KNN classifier are shown in Figure 7a, and Figure 7c respectively. From the above observations, it can be said that the ensemble learning-based (Bagging or Boosting) classifiers such as RF and GB are the best choice for the said problem.

V. Conclusion

DoH is mainly designed for privacy and security enhancement using encryption methods over traditional DNS. It is true that encryption hides the user information related to the DNS query. Due to the limited knowledge about DoH by the existing security measures, it nourishes many security threats. Hence, we interested to do the analysis of encrypted DoH traffic. To achieve this, we have used a new freely available benchmark dataset and apply popular ML classifiers to it. The performance evaluation is showing a clear classification of two different traffic benign and malicious DoH requests. As we can see in the result section, the RF and GB recorded a maximum of 100% accuracy and F1-measure in both the traffic. KNN and LR accuracy rates are 99% and 98% respectively for malicious DoH, which also better performance. NB classifier shows lower performance as compared to the other four classifiers. Based on the performance analysis, it can be suggested that ensemble learning-based classifiers like RF and GB are the best alternatives for this type of problem. As we have taken a classification problem with the available datasets for the benign DoH and malicious DoH traffic. In the future, we will try to capture some new data using DoH and non-DoH and try to predict the malicious and normal DNS request or DoH DNS request.

References

[1] A. Ramdas and R. Muthukrishnan, "A survey on dns security issues and mitigation techniques," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, 2019, pp. 781–784.

[2] Q. Li, X. Qi, J. Liu, and H. Han, "Design and implementation of traditional dns protocol," in *2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC)*. IEEE, 2017, pp. 1384–1390.

[3] D. A. Haddon and H. Alkhateeb, "Investigating data exfiltration in dns over https queries," in *2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3)*. IEEE, 2019, pp. 212–212.

[4] A. A. Maksutov, I. A. Cherepanov, and M. S. Alekseev, "Detection and prevention of dns spoofing attacks," in *2017 Siberian Symposium on Data Science and Engineering (SSDSE)*. IEEE, 2017, pp. 84–87.

[5] S. Bortzmeyer, "Dns privacy considerations," *Work in Progress, draft-ietf-dprive-problem-statement-06*, vol. 1, 2015.

[6] G. Hogben and M. Dekker, "European union agency for network and information security (enisa)," *Procure Secure–A guide to monitoring of security service levels in cloud contracts, Brüssel*, 2012.

[7] D. Detecting, "Sans institute," 2019.

[8] C. Cimpanu. (2020) Here's how to enable doh in each browser, isps be damned. [Online]. Available: https://www.zdnet.com/article/dns-over-https-will-eventually-roll-outin-all-major-browsers-despite-isp-opposition/.

[9] A. Turing *et al.*, "An analysis of godlua backdor," *360 Netlab Blog*, 2019.

[10] R. Houser, Z. Li, C. Cotton, and H. Wang, "An investigation on information leakage of dns over tls," in *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, 2019, pp. 123–137.

[11] K. Borgolte, T. Chattopadhyay, N. Feamster, M. Kshirsagar, J. Holland, A. Hounsel, and P. Schmitt, "How dns over https is reshaping privacy, performance, and policy in the internet ecosystem," *Performance, and Policy in the Internet Ecosystem (July 27, 2019)*, 2019.

[12] S. Siby, M. Juarez, C. Diaz, N. Vallina-Rodriguez, and C. Troncoso, "Encrypted dns–¿ privacy? a traffic analysis perspective," *arXiv preprint arXiv:1906.09682*, 2019.

[13] F. Nijeboer, "Detection of https encrypted dns traffic," B.S. thesis, University of Twente, 2020.

[14] J. Bushart and C. Rossow, "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)*, 2020.

[15] D. Vekshin, K. Hynek, and T. Cejka, "Doh insight: Detecting dns over https by machine learning," in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, ser. ARES '20. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3407023.3409192

[16] G. K. Mohammadreza MontazeriShatoori, Logan Davidson and A. H. Lashkari, "Detection of doh tunnels using time-series classification of encrypted traffic," 2020, pp. 1–6.

[17] I. Rish, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*. IBM, 2001, pp. 41–46.

[18] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[19] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[20] Y. Liao and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," *Computers & security*, vol. 21, no. 5, pp. 439–448, 2002.

[21] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.