Journal of Computer Sciences and Applications, 2020, Vol. 8, No. 2, 46-55
Available online at http://pubs.sciepub.com/jcsa/8/2/2
Published by Science and Education Publishing
DOI:10.12691/jcsa-8-2-2

Detecting Malicious DNS over HTTPS Traffic in Domain Name System using Machine Learning Classifiers

Yaser M. Banadaki*

Department of Computer Science, Southern University and A&M College, Baton Rouge, LA, 70813, USA
*Corresponding author: yaser_banadaki@subr.edu

Received July 09, 2020; Revised August 11, 2020; Accepted August 20, 2020

Abstract This paper presents a systematic two-layer approach for detecting DNS over HTTPS (DoH) traffic and distinguishing Benign-DoH traffic from Malicious-DoH traffic using six machine learning algorithms. The capability of machine learning classifiers is evaluated considering their accuracy, precision, recall, and F-score, confusion matrices, ROC curves, and feature importance. The results show that LGBM and XGBoost algorithms outperform the other algorithms in almost all the classification metrics reaching the maximum accuracy of 100% in the classification tasks of layers 1 and 2. LGBM algorithms only misclassified one DoH traffic test as non-DoH out of 4000 test datasets. It has also found that out of 34 features extracted from the CIRA-CIC-DoHBrw-2020 dataset, SourceIP is the critical feature for classifying DoH traffic from non-DoH traffic in layer one followed by DestinationIP feature. However, only DestinationIP is an important feature for LGBM and gradient boosting algorithms when classifying Benign-DoH from Malicious-DoH traffic in layer 2.

Keywords: *machine learning classifiers, DNS over HTTPs traffic, domain name system*

Cite This Article: Yaser M. Banadaki, “Detecting Malicious DNS over HTTPS Traffic in Domain Name System using Machine Learning Classifiers.” *Journal of Computer Sciences and Applications*, vol. 8, no. 2 (2020): 46-55. doi: 10.12691/jcsa-8-2-2.

1. Introduction

Domain Name System (DNS) was introduced based on the User Datagram Protocol (UDP), which is an unreliable delivery protocol. The security of DNS design was sufficient to satisfy the needs of the Internet at that point in time. However, providing a name to address mapping services for the chain of Internet connectivity makes the approach vulnerable network protocols for today’s internet traffic [1,2]. Cyber-attacks are considered as a new remote weapon [3,4] targeting critical infrastructures such as a presidential campaign [5], a nuclear program [6], government personnel data [7], and software providers [8]. It is vital to distinguish harmful and normal traffics while using the internet network efficiently. Securing the DNS system from any unauthorized access is critically important for the operation of private networks and the Internet. As hackers use sophisticated methodologies to attack the DNS requests and responses, DNS over HTTPS protocol is introduced by encrypting DNS queries and transmitting them in a covert channel. The approach enhances privacy and overcomes some of the DNS vulnerabilities, such as man-in-the-middle attacks.

An Intrusion Detection System (IDS) [9] plays an important role in monitoring the traffic of internet-connected devices and detect attacks for DoH traffic in a network topology. Intrusion detection was described as “the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions, defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or network” [10]. IDS is the most critical defense tool against the sophisticated and ever-growing network attacks. Different IDS systems have been developed to detect and distinguish malicious or normal traffics [9,11,12]. Machine learning algorithms [13] have been employed for attack detection such as naive Bayes [14], neural network regression [15], support vector machine [16], principal component analysis [17], and random forest [18].

In this paper, a systematic approach is proposed to evaluate the capability of six machine learning algorithms to be employed for analyzing, testing, and evaluating DoH traffic in covert channels and tunnels. This research focuses on time-series classifiers to detect and characterize DoH traffic in a two-layered machine learning approach that deploys DoH within an application and distinguish benign and malicious DoH traffic. Recently, Canadian Institute for Cybersecurity (CIC) has released a CIRA-CIC-DoHBrw-2020 dataset [19] that includes the implementation of DoH protocol within an application using five different browsers and tools and four servers to capture Benign-DoH, Malicious-DoH, and non-DoH traffic. In the two-layered approach used to capture benign and malicious DoH traffic along with non-DoH traffic, layer one is used to classify DoH traffic from non-DoH traffic, and layer 2 is used to characterize Benign-DoH from Malicious-DoH traffic. Different machine learning classifiers are evaluated for the task of distinguishing the benign and malicious DoH traffic along with non-DoH traffic.

The paper uses ML models in the IBM platform known as Auto AI [20] to identify the best type of model for the given data and efficiently compare the performance of ML models. Auto AI was described as "a suite of algorithms and feature transformations to automatically engineer new, high-value features for a given dataset" [21]. The performance of ML models for specific training datasets is subjected to the experience of the data scientists in tuning complex network parameters. The use of Auto AI ensures that the ML process generates the most accurate and optimal predictive results that effectively scales with time and resources. Several supervised classification algorithms such as Decision Tree Classifier [22], Extremely randomized Trees (Extra Trees) Classifier [23], Gradient Boosting Classifier [24], XGBoost (XGB) [25], Light Gradient Boosting Machine (LGBM) Classifier [26], and Random Forest Classifier [27]. Boosting makes a classifier strongly correlated with the true classification. The main objective of this research is to evaluate the classifiers in capturing benign and malicious DoH traffic as well as to detect and characterize DoH traffic in the two-layered ML approach. The models are evaluated using performance metrics such as detection accuracy, precision, recall, and F-score.

The rest of the paper is organized as follows. An overview of the CIRA-CIC-DoHBrw-2020 dataset, training procedure and machine learning models are presented in Section 2. Section 3 discusses the performance of machine learning-based attack detectors, including the relative importance of the features for each model and their performance considering precision, recall, F-score, sensitivity, and specificity. Section 4 discusses the comparison of the classifiers in classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2. The paper is summarized with some conclusions in section 5.

2. Network Data Collection and Training Process

Raw data for training ML models are adopted from the CIRA-CIC-DoHBrw-2020 dataset [19] that contains benign and malicious DoH traffic along with non-DoH traffic. Non-DoH traffics generated by accessing a website that uses HTTPS protocol and labels as non-DoH traffic and benign-DoH traffics using the same technique by browsing the web with Mozilla Firefox and Google Chrome. For generating malicious-DoH traffics, DNS tunneling tools such as dns2tcp, DNSCat2, and Iodine are used, which can create tunnels of encrypted data to send TCP traffic encapsulated in DNS queries using TLS-encrypted HTTPS requests to special DoH servers. To create a new representation of datasets, the dimensionality of data is reduced by a notion of packet clumps that defines as a sequence of one or more consecutive packets of a network flow. Figure 1(a) shows the network diagram used to capture the traffic for the CIRA-CIC-DoHBrw-2020 datasets [19]. For pre-processing and training the classifiers, the non-DoH HTTPS and benign DoH are captured using normal web browsing activities and malicious DoH using a combination of tools used to create DoH tunnels. A subset of CIRA-CIC-DoHBrw-2020 dataset containing 20,000 traffic from each class is taken to train and optimize the ML models, and finally, 15% of this data is used to evaluate the performance of classifiers in two-layer topology [19].

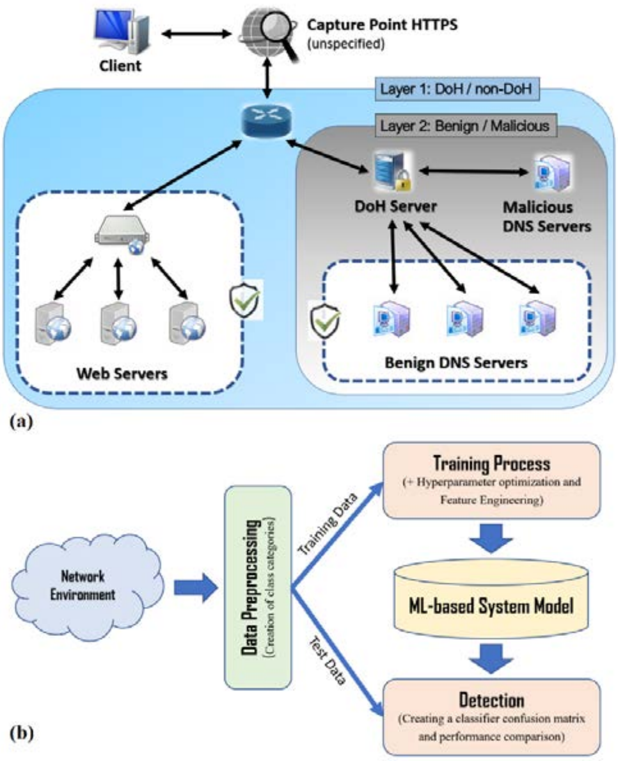


Figure 1. (a) Training procedure of intrusion detection including data preprocessing, training, and optimizing the training algorithms, deployment of ML-based classifiers, and testing of the model to extract the classification performance metrics. (b) The network topology used to capture the traffic datasets, including benign and malicious DoH traffic along with non-DoH traffic

TCPDUMP is used to capture the traffic between the DoH proxy and the DoH server. A DoH traffic flow is generated and analyzed for anomaly and attack detection and characterization. Different DoH tunneling scenarios are simulated, and the resulting HTTPS traffics is captured. In the simulation, the clients were run on ten servers simultaneously that are connected to a DNS nameserver. Adguard, Cloudflare, Google, and Quad9 are used as a DoH Server, and Iodine, DNS2TCP, and DNScat2 are used as a DNS tunneling tool. The delay between sending requests and DNS record types is created using tunneling client and server configurations. To diversify the dataset, the transmission rate is changed by a random value between 100 B/s to 1100 B/s. The statistical and time-series features of the captured PCAP files are extracted by the network traffic analyzer known as CICFlowMeter [28] in Python. The generated dataset is stored in a CSV file as output that labeled flow-wise based on the IP addresses of the servers in the network diagram. Table 1 lists the 34 captured statistical traffic features.

The procedure to train ML-based models to detect attacks from the network traffic is shown in Figure 1(b). The ML algorithms need to be kept generic so that a trained algorithm can predict an unseen instance correctly. As such, the available dataset is split into training and test dataset where the algorithm is trained using a training dataset with known attack labels, and a test dataset is used to evaluate the model performance in predicting the attack labels. A confusion matrix is generated using the number of correct predictions on the test dataset to find the actual class label against the predicted class label for each category and to extract the classification metrics.

Figure 2 shows the training progress pipelines based on six machine learning algorithms: decision tree classifier, extra trees classifier, gradient boosting classifier, LGBM classifier, XGB classifier, and random forest classifier. For each of these six ML algorithms, AutoAI generates the following pipelines: automated model selection (Pipeline 1), hyperparameter optimization (Pipeline 2), automated feature engineering (Pipeline 3), hyperparameter optimization (Pipeline 4). The Hyper-parameter optimization (HPO) process includes finding a set of optimal parameters for the learning procedure to enable fast convergence to a better performing solution. To achieve the most accurate detection of benign and malicious DoH traffic and the characterization of DoH traffic in the two-layered ML approach, the extracted features are equally scaled to reduce ML bias, and the raw data is transformed into the combination of features that best represents the intrusion detection problem. The models use transfer learning (TL), in which the knowledge gained while solving one problem is applied to a different but related problem. The approach extracts existing knowledge learned from one environment to solve new problems. The pre-trained models take advantage of training with a lower amount of data for the new problem and significantly shortens the training procedure.

Table 1. List of the 34 statistical features extracted from captured traffic

No.	Feature
1	SourceIP
2	DestinationIP
3	SourcePort
4	DestinationPort
5	TimeStamp
6	Duration
7	FlowBytesSent
8	FlowSentRate
9	FlowBytesReceived
10	FlowReceivedRate
11	PacketLengthVariance
12	PacketLengthStandardDeviation
13	PacketLengthMean
14	PacketLengthMedian
15	PacketLengthMode
16	PacketLengthSkewFromMedian
17	PacketLengthSkewFromMode
18	PacketLengthCoefficientofVariation
19	PacketTimeVariance
20	PacketTimeStandardDeviation
21	PacketTimeMean
22	PacketTimeMedian
23	PacketTimeMode
24	PacketTimeSkewFromMedian
25	PacketTimeSkewFromMode
26	PacketTimeCoefficientofVariation
27	ResponseTimeTimeVariance
28	ResponseTimeTimeStandardDeviation
29	ResponseTimeTimeMean
30	ResponseTimeTimeMedian
31	ResponseTimeTimeMode
32	ResponseTimeTimeSkewFromMedian
33	ResponseTimeTimeSkewFromMode
34	ResponseTimeTimeCoefficientofVariation

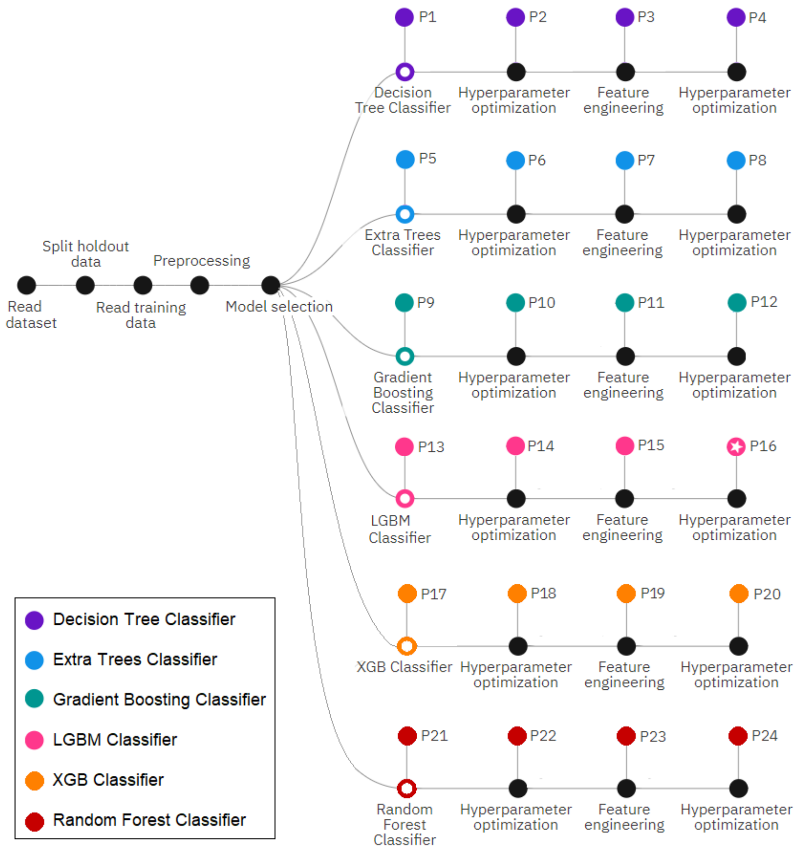


Figure 2. Training progress pipelines for four ML models: XGBoost classifier, random forest classifier, decision tree classifier, and gradient boosting classifier

3. Results

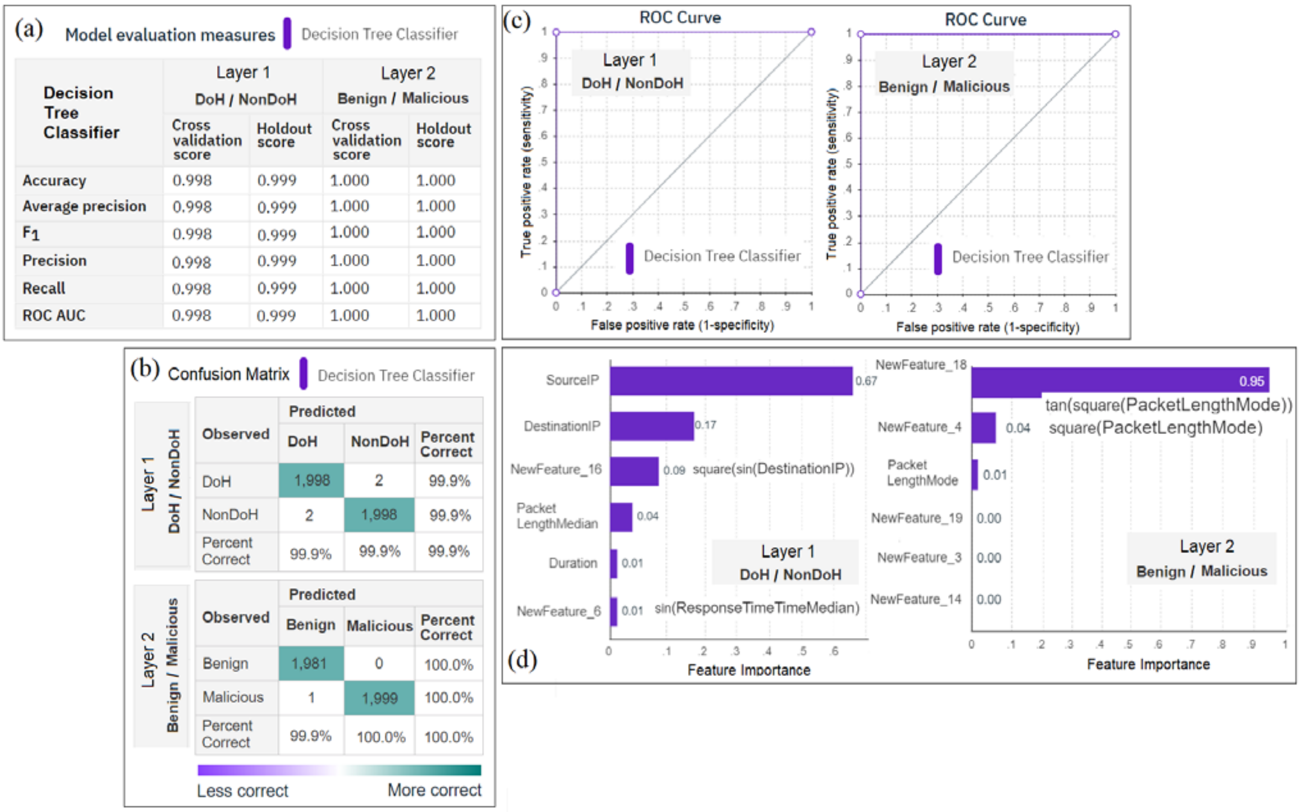
The classification metrics of six classification algorithms: Decision Tree, Extra Trees, Gradient Boosting, XGBoost, Light Gradient Boosting Machine, and Random Forest are presented for classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2. For each algorithm, the confusion matrix is generated using the number of correct predictions on the test dataset to find the actual class label against the predicted class label for each category and to extract the classification metrics. The classification metrics include accuracy, average precision, F1-score, precision, recall, and ROC AUC. The accuracy is calculated as a fraction of true positive among all the positive's recalled and can be viewed as a measure of a classifier's exactness. Precision is the fraction of relevant instances among the retrieved instances, while recall is the fraction of true positives among all the true events and can be viewed as a measure of a classifier's completeness. The F-score considers both precision and recall as the harmonic mean of the Precision and Recall indicating the worst accuracy when it becomes 0, while the best accuracy corresponds to 1. Cross-validation and holdout are reported for each evaluation measure in classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2. Cross-validation provides the model the opportunity to train on multiple train-test splits resulting in a better indication of how well your model will perform on unseen data while holdout is dependent on just one train-test split. The ROC curve is depicted for each algorithm as a fundamental tool used for diagnostic test evaluation providing measures of overall predictive accuracy of the models. The curve depicts the proportion of positive outcomes that are correctly predicted, also known as the true positive rate (i.e., sensitivity), against the proportion of negative outcomes that are falsely predicted to be positive, also known as the false positive rate (i.e., 1-Specificity). The sensitivity is a measure of a classifier's completeness, and the specificity measures the proportion of correctly identified negatives. The area under the ROC curve (ROC AUC) represents the measure of separability, demonstrating how much models are capable of distinguishing between classes. ROC curves can be used to evaluate the performance of the four classification models.

3.1. Decision Tree Algorithm

Decision trees use a series of sequential steps to decide to split a node into two or more sub-nodes based on probabilities, costs, or other consequences of making a particular decision. The algorithm splits the nodes on all available variables and then selects the split that results in most homogeneous sub-nodes. Figure 3(a) shows the accuracy, average precision, F1-score, precision, recall, and area under the receiver operating characteristic (ROC) curve (or ROC AUC) of the algorithms in classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2. For the decision tree algorithm, all of these measures for cross-validation and handout scores of layer one are calculated as 99.8% and 99.9%, respectively. All the measures of layer 2 for both cross-validation and handout scores are calculated as 100%. Figure 3(b) shows the confusion matrix of the algorithm. It can be observed that, out of almost 4000 test data, only 4 and 1 samples are misclassified in layers 1 and 2, respectively. Figure 3(c) shows the ROC curve of layers 1 and 2. The ROC curve of the algorithm passes through the upper left corner, indicating very high sensitivity and specificity with no overlap between the classes. Figure 3(d) shows the relative importance of the first six features of the algorithm in classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2. The importance of the features is extracted to ensure the highest accuracy of the algorithm. Correlation between a pair of features is analyzed to eliminate features that contribute the same information about the data. The *SourceIP* is the key feature in layer 1 with the feature importance of 0.67, while the most important feature for characterizing Benign-DoH from Malicious-DoH traffic in layer 2 is for a new feature engineered as *tan (square (Packet Length Mode)).*

3.2. Extra Tree Algorithm

Extra Trees Classifier (or Extremely Randomized Trees Classifier) is a type of ensemble learning technique that works by creating many unpruned decision trees from the training dataset. The algorithm aggregates the results of multiple de-correlated decision trees to generate the output of the classification algorithm. Predictions are made by using majority voting in the case of classification. Figure 4(a) shows the accuracy, average precision, F1-score, precision, recall, and ROC AUC of the extra tree algorithms in classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2. For this algorithm, all the measures for cross-validation and handout scores of layer one are calculated above 99.4% and 99.9%, respectively. All the measures of layer 2 for cross-validation and handout scores are calculated above 99.9%. Figure 4(b) shows the confusion matrix of the algorithm. It can be observed that, out of almost 4000 test data, only 3 and 2 samples are misclassified in layers 1 and 2, respectively. Figure 4(c) shows the ROC curve of layers 1 and 2. The ROC curve of the algorithm passes through the upper left corner, indicating very high sensitivity and specificity with no overlap between the classes. Figure 4(d) shows the relative importance of the first six features of the algorithm in classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2. The importance of the features is extracted, and the correlation between a pair of features is analyzed to eliminate features that contribute the same information about the data and maximize the accuracy of the algorithm. The most important features of the algorithm are calculated by producing the *DestinationIP* and *SourceIP* for layer one and *tan(square(PacketLengthMode))* for layer 2.



(a) Model evaluation measures | Decision Tree Classifier

Decision Tree Classifier	Layer 1 DoH / NonDoH Cross validation score	Layer 1 DoH / NonDoH Holdout score	Layer 2 Benign / Malicious Cross validation score	Layer 2 Benign / Malicious Holdout score
Accuracy	0.998	0.999	1.000	1.000
Average precision	0.998	0.999	1.000	1.000
F1	0.998	0.999	1.000	1.000
Precision	0.998	0.999	1.000	1.000
Recall	0.998	0.999	1.000	1.000
ROC AUC	0.998	0.999	1.000	1.000

(b) Confusion Matrix | Decision Tree Classifier

Layer 1 DoH / NonDoH

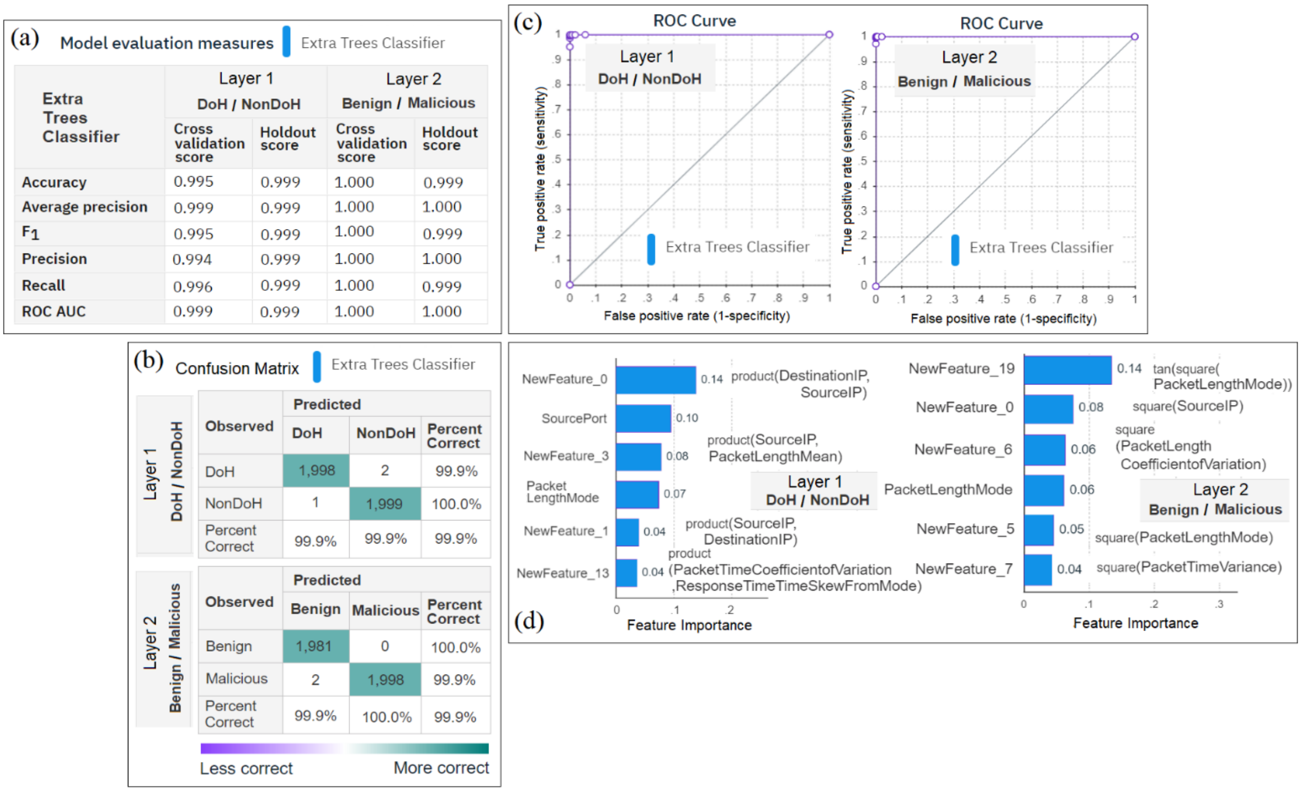
Observed	Predicted DoH	Predicted NonDoH	Percent Correct
DoH	1,998	2	99.9%
NonDoH	2	1,998	99.9%
Percent Correct	99.9%	99.9%	99.9%

Layer 2 Benign / Malicious

Observed	Predicted Benign	Predicted Malicious	Percent Correct
Benign	1,981	0	100.0%
Malicious	1	1,999	100.0%
Percent Correct	99.9%	100.0%	100.0%

Less correct — More correct

Figure 3. (a) Evaluation measures, (b) confusion matrices, (c) ROC curve, and (d) feature importance of Decision Tree algorithm in classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2



(a) Model evaluation measures | Extra Trees Classifier

Extra Trees Classifier	Layer 1 DoH / NonDoH Cross validation score	Layer 1 DoH / NonDoH Holdout score	Layer 2 Benign / Malicious Cross validation score	Layer 2 Benign / Malicious Holdout score
Accuracy	0.995	0.999	1.000	0.999
Average precision	0.999	0.999	1.000	1.000
F1	0.995	0.999	1.000	0.999
Precision	0.994	0.999	1.000	1.000
Recall	0.996	0.999	1.000	0.999
ROC AUC	0.999	0.999	1.000	1.000

(b) Confusion Matrix | Extra Trees Classifier

Layer 1 DoH / NonDoH

Observed	Predicted DoH	Predicted NonDoH	Percent Correct
DoH	1,998	2	99.9%
NonDoH	1	1,999	100.0%
Percent Correct	99.9%	99.9%	99.9%

Layer 2 Benign / Malicious

Observed	Predicted Benign	Predicted Malicious	Percent Correct
Benign	1,981	0	100.0%
Malicious	2	1,998	99.9%
Percent Correct	99.9%	100.0%	99.9%

Less correct — More correct

(c)
ROC Curve
Layer 1
DoH / NonDoH
True positive rate (sensitivity)
False positive rate (1-specificity)
0 .1 .2 .3 .4 .5 .6 .7 .8 .9 1
Extra Trees Classifier
ROC Curve
Layer 2
Benign / Malicious
True positive rate (sensitivity)
False positive rate (1-specificity)
Extra Trees Classifier
(d)
NewFeature_0 0.14 product(DestinationIP, SourceIP)
SourcePort 0.10
NewFeature_3 0.08 product(SourceIP, PacketLengthMean)
Packet LengthMode 0.07
NewFeature_1 0.04 product(SourceIP, DestinationIP)
NewFeature_13 0.04 product (PacketTimeCoefficientofVariation ,ResponseTimeTimeSkewFromMode)
Layer 1
DoH / NonDoH
0 .1 .2
Feature Importance
NewFeature_19 0.14 tan(square(PacketLengthMode))
NewFeature_0 0.08 square(SourceIP)
NewFeature_6 0.06 square (PacketLength CoefficientofVariation)
PacketLengthMode 0.06
NewFeature_5 0.05 square(PacketLengthMode)
NewFeature_7 0.04 square(PacketTimeVariance)
Layer 2
Benign / Malicious
0 .1 .2 .3
Feature Importance

Figure 4. (a) Evaluation measures, (b) confusion matrices, (c) ROC curve, and (d) feature importance of Decision Tree algorithm in classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2

3.3. Gradient Boosting algorithm

Gradient boosting corrects the shortcomings of existing weak learners in random forests by building one tree at a time. The algorithm also combines results along the way, improving the random forests algorithm that combines the results at the end of the process. Boosting makes a strong learner by optimizing step for every new tree, allowing the classification model to generate less False Alarms and higher accuracy of classification. Figure 5(a) shows the accuracy, average precision, F1-score, precision, recall, and ROC AUC of the gradient boosting algorithms in classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2. For this algorithm, all the measures for cross-validation and handout scores of layer one are calculated above 99.8% and 99.9%, respectively. All the measures of layer 2 for cross-validation and handout scores are calculated as 100%, except the precision of the cross-validation score, which is 99.9%. Figure 5(b) shows the confusion matrix of the algorithm. It can be observed that, out of almost 4000 test data, only 2 and 1 samples are misclassified in layers 1 and 2, respectively. Figure 5(c) shows the ROC curve of layers 1 and 2. The ROC curve of the algorithm passes through the upper left corner, indicating very high sensitivity and specificity with no overlap between the classes. Figure 5(d) shows the relative importance of the first six features of the algorithm in classifying DoH traffic from non-DoH traffic in layer 1 and characterizing Benign-DoH from Malicious-DoH traffic in layer 2. The most important features of the algorithm are calculated 0.68 for *SourceIP* in layer 1 and new feature is generated in layer 2 as *tan(DestinationIP)* in layer 2.

3.4. LGBM algorithm

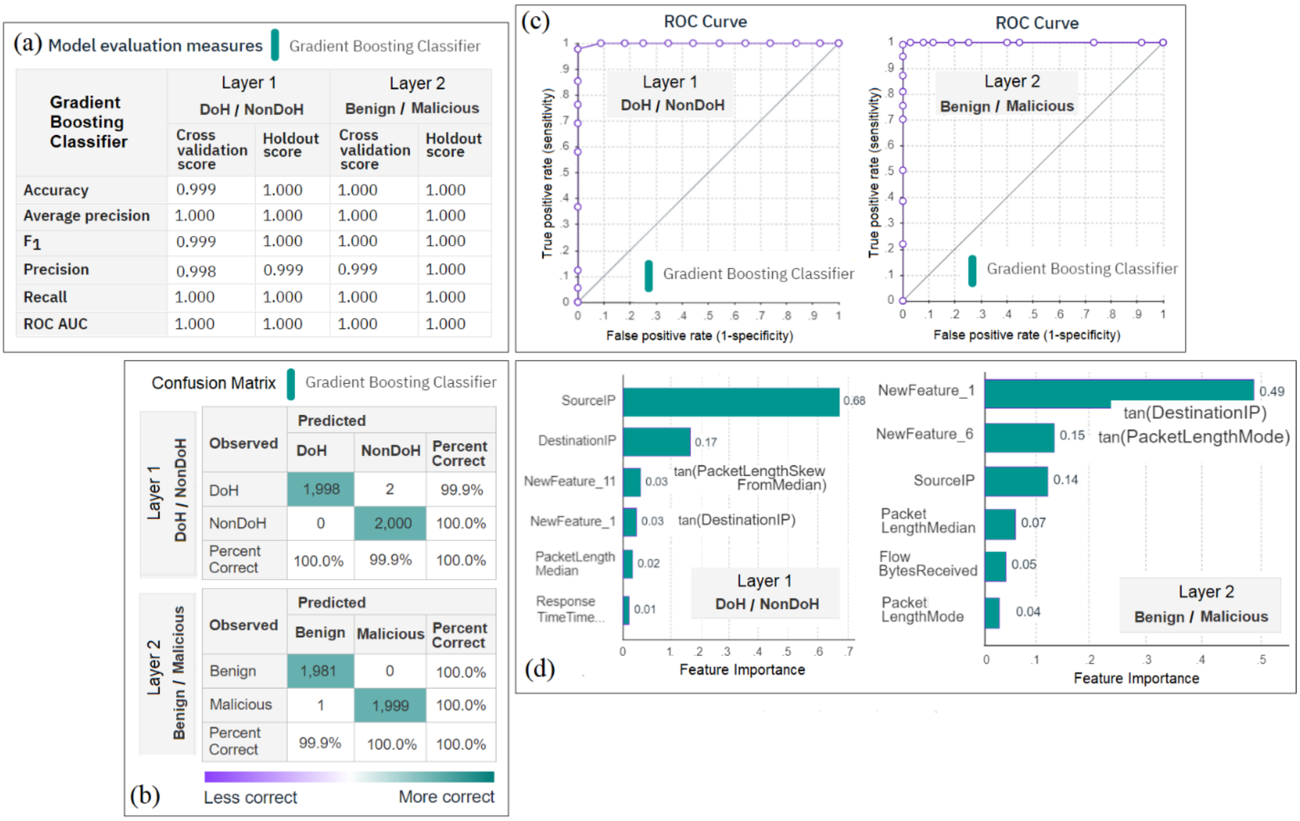
LGBM is a light gradient boosting framework that has faster execution time than the XGBoost algorithm and outperforms it in training speed and the dataset sizes it can handle. Similarly, boosting makes a strong learner by optimizing step for every new tree, allowing the classification model to generate less False Alarms and higher accuracy of classification. Figure 6(a) shows the accuracy, average precision, F1-score, precision, recall, and ROC AUC of the LGBM algorithms in classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2. For this algorithm, all the measures for both cross-validation and handout scores of layer one are calculated 100% except precision, which is 99.9%. Figure 6(b) shows the confusion matrix of the algorithm. It can be observed that, out of almost 4000 test data, only 1 sample is misclassified in layer one and all test dates are correctly classified in layer 2. Figure 6(c) shows the ROC curve of layers 1 and 2. The ROC curve of the algorithm passes through the upper left corner, indicating very high sensitivity and specificity without overlap between the classes. Figure 6(d) shows the relative importance of the first six features of the algorithm in classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2. The most important features of the algorithm are DestinationIP in layers 1 and 2 that have been calculated as 0.28 and 0.06, respectively.

3.5. XGBoost algorithm

XGBoost algorithm is a strong classifier because of its regularization aspect that avoids data overfitting problems. The optimizing step for every new tree that attaches reduces false alarms further and improve the classification accuracy. The approach makes the classifier fast to deal with the system overwhelmed by a float of attacks effectively. Thus, the XGBoost algorithm outperforms many existing models demonstrating robust IDS to deal with the majority of the attacks in a real-world network. Figure 7(a) shows the accuracy, average precision, F1-score, precision, recall, and ROC AUC of the XGBoost algorithms in classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2. For this algorithm, all the measures for cross-validation and handout scores of layer one are calculated 100% except precision, which is 99.9%. Figure 7(b) shows the confusion matrix of the XGBoost algorithm. It can be observed that, out of about 4000 test data, only two samples are misclassified in each layer. Figure 7(c) shows the ROC curve of layers 1 and 2. The ROC curve of the algorithm passes through the upper left corner, indicating very high sensitivity and specificity without overlap between the classes. Figure 7(d) shows the relative importance of the first six features of the XGBoost algorithm in classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2. The two most important features of the XGBoost algorithm in layer 1 are PacketLengthMedian and SourceIP that have been calculated as 0.47 and 0.34, respectively. However, the most important feature of the XGBoost algorithm in layer 2 is tan(PacketLengthSkewFromMode) that has been calculated as 0.51.

3.6. Random Forest Algorithm

Random forests are an ensemble algorithm that has many trees combined using averages or majority rule at the end of the process. From a randomly selected subset of the training dataset, a random forest classifier creates a set of decision trees and then aggregates the votes from different decision trees to make decisions about the final class of the test object. Figure 8(a) shows the accuracy, average precision, F1-score, precision, recall, and ROC AUC of the random forest algorithms in classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2. For this algorithm, all the measures for cross-validation and holdout scores of layer one are calculated above 99.7% and 99.9%, respectively. Figure 8(b) shows the confusion matrix of the random forests algorithm. It can be observed that, out of about 4000 test data, only four test samples are misclassified in layer one and two test data are misclassified in layer 2. Figure 8(c) shows the ROC curve of the layer 1 and 2 that passes through the upper left corner indicating the classes have no overlap. Figure 8(d) shows the relative importance of the first six features of the random forests algorithm in classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2. The most important feature of the random forests algorithm in layer 1 is the product of *DestinationIP* and *SourceIP* that has been calculated as 0.25. The most important feature of the algorithm in layer 2 is a new feature engineered as *tan(PacketLengthMode)* that has been calculated as 0.54.



(a) Model evaluation measures | Gradient Boosting Classifier

Gradient Boosting Classifier	Layer 1 DoH / NonDoH		Layer 2 Benign / Malicious	
	Cross validation score	Holdout score	Cross validation score	Holdout score
Accuracy	0.999	1.000	1.000	1.000
Average precision	1.000	1.000	1.000	1.000
F_1	0.999	1.000	1.000	1.000
Precision	0.998	0.999	0.999	1.000
Recall	1.000	1.000	1.000	1.000
ROC AUC	1.000	1.000	1.000	1.000

(b) Confusion Matrix | Gradient Boosting Classifier

Layer 1 DoH / NonDoH

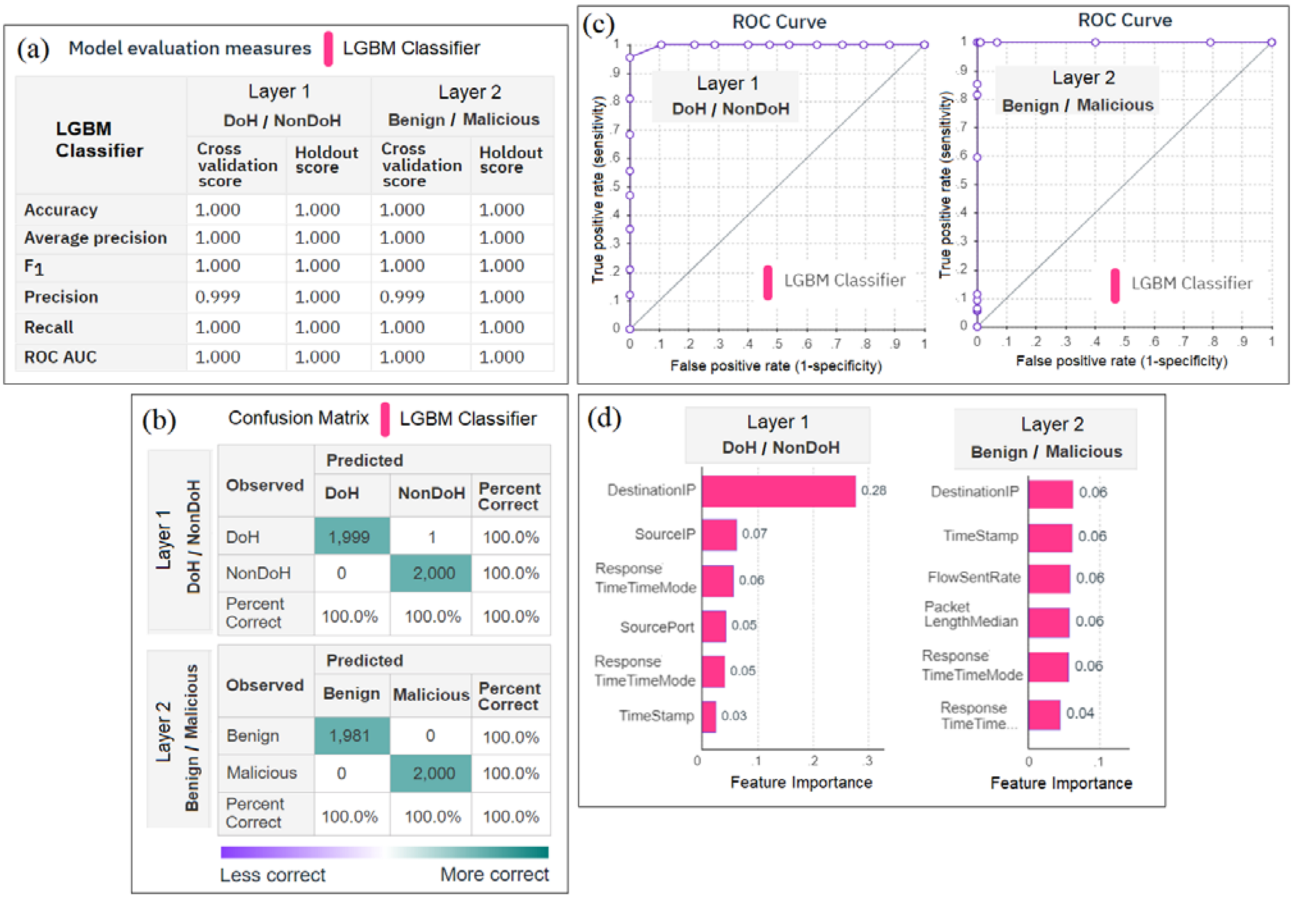
Observed	Predicted DoH	NonDoH	Percent Correct
DoH	1,998	2	99.9%
NonDoH	0	2,000	100.0%
Percent Correct	100.0%	99.9%	100.0%

Layer 2 Benign / Malicious

Observed	Predicted Benign	Malicious	Percent Correct
Benign	1,981	0	100.0%
Malicious	1	1,999	100.0%
Percent Correct	99.9%	100.0%	100.0%

Less correct — More correct

Figure 5. (a) Evaluation measures, (b) confusion matrices, (c) ROC curve, and (d) feature importance of Gradient Boosting algorithm in classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2



(a) Model evaluation measures | LGBM Classifier

LGBM Classifier	Layer 1 DoH / NonDoH		Layer 2 Benign / Malicious	
	Cross validation score	Holdout score	Cross validation score	Holdout score
Accuracy	1.000	1.000	1.000	1.000
Average precision	1.000	1.000	1.000	1.000
F_1	1.000	1.000	1.000	1.000
Precision	0.999	1.000	0.999	1.000
Recall	1.000	1.000	1.000	1.000
ROC AUC	1.000	1.000	1.000	1.000

(b) Confusion Matrix | LGBM Classifier

Layer 1 DoH / NonDoH

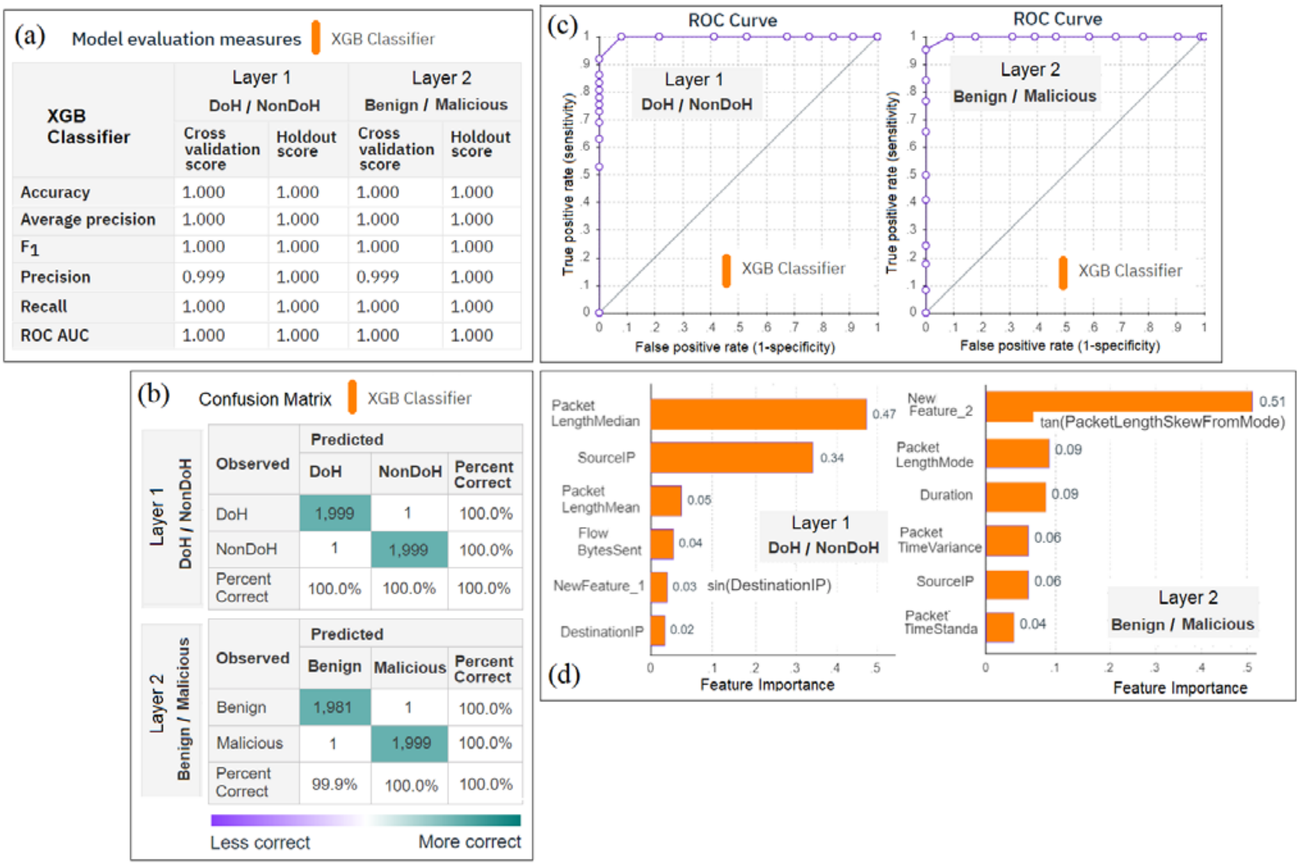
Observed	Predicted DoH	NonDoH	Percent Correct
DoH	1,999	1	100.0%
NonDoH	0	2,000	100.0%
Percent Correct	100.0%	100.0%	100.0%

Layer 2 Benign / Malicious

Observed	Predicted Benign	Malicious	Percent Correct
Benign	1,981	0	100.0%
Malicious	0	2,000	100.0%
Percent Correct	100.0%	100.0%	100.0%

Less correct — More correct

Figure 6. (a) Evaluation measures, (b) confusion matrices, (c) ROC curve, and (d) feature importance of LGBM algorithm in classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2



(a) Model evaluation measures | XGB Classifier

XGB Classifier	Layer 1		Layer 2	
	DoH / NonDoH		Benign / Malicious	
	Cross validation score	Holdout score	Cross validation score	Holdout score
Accuracy	1.000	1.000	1.000	1.000
Average precision	1.000	1.000	1.000	1.000
F1	1.000	1.000	1.000	1.000
Precision	0.999	1.000	0.999	1.000
Recall	1.000	1.000	1.000	1.000
ROC AUC	1.000	1.000	1.000	1.000

(b) Confusion Matrix | XGB Classifier

Layer 1 DoH / NonDoH

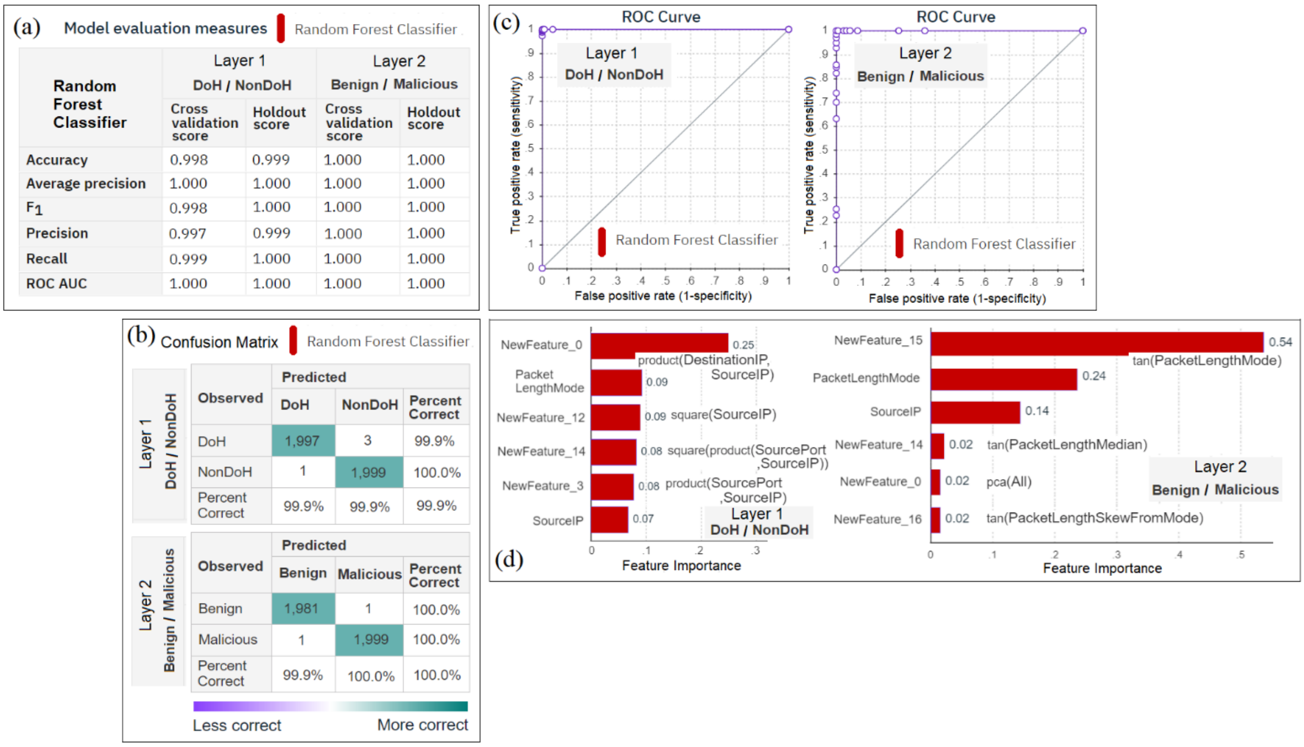
Observed	Predicted		
	DoH	NonDoH	Percent Correct
DoH	1,999	1	100.0%
NonDoH	1	1,999	100.0%
Percent Correct	100.0%	100.0%	100.0%

Layer 2 Benign / Malicious

Observed	Predicted		
	Benign	Malicious	Percent Correct
Benign	1,981	1	100.0%
Malicious	1	1,999	100.0%
Percent Correct	99.9%	100.0%	100.0%

Less correct More correct

Figure 7. (a) Evaluation measures, (b) confusion matrices, (c) ROC curve, and (d) feature importance of XGBoost algorithm in classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2



(a) Model evaluation measures | Random Forest Classifier

Random Forest Classifier	Layer 1		Layer 2	
	DoH / NonDoH		Benign / Malicious	
	Cross validation score	Holdout score	Cross validation score	Holdout score
Accuracy	0.998	0.999	1.000	1.000
Average precision	1.000	1.000	1.000	1.000
F1	0.998	1.000	1.000	1.000
Precision	0.997	0.999	1.000	1.000
Recall	0.999	1.000	1.000	1.000
ROC AUC	1.000	1.000	1.000	1.000

(b) Confusion Matrix | Random Forest Classifier

Layer 1 DoH / NonDoH

Observed	Predicted		
	DoH	NonDoH	Percent Correct
DoH	1,997	3	99.9%
NonDoH	1	1,999	100.0%
Percent Correct	99.9%	99.9%	99.9%

Layer 2 Benign / Malicious

Observed	Predicted		
	Benign	Malicious	Percent Correct
Benign	1,981	1	100.0%
Malicious	1	1,999	100.0%
Percent Correct	99.9%	100.0%	100.0%

Less correct More correct

(c)
ROC Curve
Layer 1
DoH / NonDoH
Random Forest Classifier
True positive rate (sensitivity)
False positive rate (1-specificity)
0 .1 .2 .3 .4 .5 .6 .7 .8 .9 1
ROC Curve
Layer 2
Benign / Malicious
Random Forest Classifier
True positive rate (sensitivity)
False positive rate (1-specificity)
(d)
NewFeature_0 0.25 product(DestinationIP, SourceIP)
Packet LengthMode 0.09
NewFeature_12 0.09 square(SourceIP)
NewFeature_14 0.08 square(product(SourcePort ,SourceIP))
NewFeature_3 0.08 product(SourcePort ,SourceIP)
SourceIP 0.07
Layer 1
DoH / NonDoH
Feature Importance
0 .1 .2 .3
NewFeature_15 0.54 tan(PacketLengthMode)
PacketLengthMode 0.24
SourceIP 0.14
NewFeature_14 0.02 tan(PacketLengthMedian)
NewFeature_0 0.02 pca(All)
NewFeature_16 0.02 tan(PacketLengthSkewFromMode)
Layer 2
Benign / Malicious
Feature Importance
0 .1 .2 .3 .4 .5

Figure 8. (a) Evaluation measures, (b) confusion matrices, (c) ROC curve, and (d) feature importance of Random Forest algorithm in classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2

4. Discussion

Six classification algorithms: Decision Tree, Extra Trees, Gradient Boosting, XGBoost, Light Gradient Boosting Machine, and Random Forest are evaluated for classifying DoH traffic from non-DoH traffic in layer one and characterizing Benign-DoH from Malicious-DoH traffic in layer 2. LGBM and XGBoost algorithms outperform the other algorithms in almost all the classification metrics reaching the maximum accuracy of 100% in classifying DoH traffic from non-DoH traffic in layer 1. The cross-validation precision of these algorithms is calculated as 99.9%, which is larger than the other four algorithms. However, the handout precisions of LGBM and XGBoost algorithms are calculated as 99.9%, which are the same as the other four algorithms. The weakest performance results in 99.4% in calculating the cross-validation classification metrics of extra tree algorithms in classifying DoH traffic from non-DoH traffic in layer 1. A slightly lower precision and recall of extra tree algorithms indicate a few false positives and a few false negatives.

For the six classification algorithms, confusion matrices are generated that indicate the number of correct predictions on the test dataset to find the actual class label against the predicted class label for each category. For classifying DoH traffic from non-DoH traffic in layer 1, 4000 test datasets are used to evaluate each algorithm. LGBM algorithms outperform the other algorithms as only 1 DoH test data is predicted as non-DoH. The lowest performance belongs to the decision tree algorithm as two DoH test data is predicted as non-DoH, and non-DoH test data is predicted as DoH. For classifying Benign-DoH from Malicious-DoH traffic in layer 2, LGBM algorithms outperform the other algorithms as all test dates are classified correctly. Extra tree algorithm has the weakest performance in layer two by misclassifying two malicious-DoH traffic as benign-DoH. Both XGBoost and random forest algorithms perform the same by misclassifying one malicious-DoH traffic as benign-DoH, and one benign-DoH traffic as malicious-DoH.

For the six classification algorithms, the ROC curves are generated that indicate the overlap between the classes. The ROC curves of classifying DoH traffic from non-DoH traffic in layer 1 passes through the upper left corner has 100% sensitivity and 100% specificity for the decision tree, extra tree, and random forest classifiers, indicating a negligible overlap between the classes. On the other hand, XGBoost classifier has a minor overlap between classes of DoH and non-DoH traffics. For classifying Benign-DoH from Malicious-DoH traffic in layer 2, the ROC curves of all algorithms are close to the upper left corner indicating high sensitivity and specificity, as well as the accuracy of the classifiers. The perfect area under the ROC curve is built on the truth value of ‘1’ and ‘0’, resulting in the angle-shaped elbow seen in the ROC curve.

For the six classification algorithms, the six most important features are generated to understand the significance of the signatures in data collected for classifications in layers 1 and 2. DestinationIP and SourceIP are the key features for classifying DoH traffic from non-DoH traffic in layer 1. SourceIP is the most important feature in the decision tree algorithm and gradient boosting algorithm, the second most important feature in the XGBoost algorithm, and the engineered SourceIP is the most important feature in the extra trees algorithm and random forests algorithm. For layer one classification, DestinationIP is also an important feature as this feature is the most important feature in LGBM algorithms, the second most important feature in the decision tree algorithm, and its engineered version are the most important feature in extra trees algorithm and random forests algorithms. From the weight of the features, it can be understood that the average importance of SourceIP and DestinationIP for layer one models are roughly 0.42 and 0.21, respectively. Packet Length Median is the only important feature of the XGBoost algorithm. For classifying Benign-DoH from Malicious-DoH traffic in layer 2, SourceIP is not an important feature, while DestinationIP is still a key feature for LGBM and gradient boosting algorithms. For layer two classification, tan (Packet Length Mode) is the most important feature for three algorithms of the decision tree, extra trees, and random forests, with the average feature importance of 0.54. Other than Packet Length Mode, tan (Packet Length Skew From Mode) is also important for layer 2 classification using the XGBoost algorithm.

5. Conclusion

This paper has introduced a systematic approach to evaluating the capability of six machine learning algorithms to be employed for analyzing, testing, and evaluating DoH traffic in two-layered machine learning models. DoH traffic is distinguished from non-DoH traffic in layer 1, and Benign-DoH traffic is separated from Malicious-DoH traffic in layer 2. Six different machine learning models are used in two layers for distinguishing the benign and malicious DoH traffic along with non-DoH traffic, and the performance of ML models are compared considering their classification performance such as accuracy, precision, recall, and F-score, confusion matrices, ROC curves, and feature importance. The results show that LGBM and XGBoost algorithms outperform the other algorithms in almost all the classification metrics reaching the maximum accuracy of 100% in the classification tasks of layers 1 and 2. The confusion matrix of LGBM algorithms shows only 1 DoH test data is predicted as non-DoH out of 4000 test datasets. For most of the algorithms, SourceIP is found as the key feature for classifying DoH traffic from non-DoH traffic in layer one, followed by DestinationIP feature. However, SourceIP is not an important feature for classifying Benign-DoH from Malicious-DoH traffic in layer two and DestinationIP is an important feature only for two algorithms of LGBM and gradient boosting. Overall, besides *SourceIP* and DestinationIP, Packet Length Median, Packet Length Mode, and Packet Length Skew From Mode are essential features for the proposed two-layer approach out of 34 features extracted from the CIRA-CIC-DoHBrw-2020 dataset.

References

[1] Davidowicz, D., "Domain name system (DNS) security," Yahoo Geocities, 1999.

[2] Chatzis, N., "Motivation for behaviour-based DNS security: A taxonomy of DNS-related internet threats," *International Conference on Emerging Security Information, Systems, and Technologies (SECUREWARE 2007)*, 2007, 36-41.

[3] Ahmim, A., Ghoualmi–Zine, N., "A new adaptive intrusion detection system based on the intersection of two different classifiers," *International Journal of Security and Networks,* 9(3), 125-132, Jan.2014.

[4] Ahmim, A., Zine, N. G., "A new hierarchical intrusion detection system based on a binary tree of classifiers," *Information & Computer Security,* Mar. 2015.

[5] Detrow, S. "Obama on Russian Hacking:‘We Need to Take Action. And We Will’," *NPR News,* 2016.

[6] Langner, R., "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security & Privacy,* 9(3), 49-51, 2011.

[7] Bouteraa, I., Derdour, M., Ahmim, A., "Intrusion Detection using Data Mining: A contemporary comparative study," *3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, *IEEE,* 2018, 1-8.

[8] Kshetri, N., "Kaspersky Lab: from Russia with anti-virus," *Emerald Emerging Markets Case Studies,* 2011.

[9] Liao, H.-J., Lin, C.-H. R., Lin, Y.-C., Tung, K.-Y. "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, 36(1), 16-24, 2013.

[10] Bace, R., Mell, P., "NIST special publication on intrusion detection systems," *BOOZ-ALLEN AND HAMILTON INC MCLEAN VA*, 2001.

[11] Ertam, F., Kilincer, L. F., Yaman, O., "Intrusion detection in computer networks via machine learning algorithms," *International Artificial Intelligence and Data Processing Symposium (IDAP), IEEE*, 2017, 1-4.

[12] Lazarevic, A., Kumar, V., Srivastava, J. "Intrusion detection: A survey," *Managing Cyber Threats*: Springer, 2005, 19-78.

[13] Tsai, C.-F., Hsu, Y.-F., Lin, C.-Y., Lin, W.-Y. "Intrusion detection by machine learning: A review," *expert systems with applications,* 36(10), 11994-12000, 2009.

[14] Li, W., Li, Q., "Using naive Bayes with AdaBoost to enhance network anomaly intrusion detection," *Third International Conference on Intelligent Networks and Intelligent Systems*, *IEEE,* 2010, 486-489.

[15] Gautam, S. K., Om, H., "Computational neural network regression model for Host based Intrusion Detection System," *Perspectives in Science,* 8, 93-95, 2016.

[16] Jha, J. Ragha, L., "Intrusion detection system using support vector machine," *International Journal of Applied Information Systems (IJAIS),* 3, 25-30, 2013.

[17] Liu, G., Yi, Z., Yang, S., "A hierarchical intrusion detection model based on the PCA neural networks," *Neurocomputing,* 70 (7-9), 1561-1568, 2007.

[18] Zhang, J., Zulkernine, M., Haque, A., "Random-forests-based network intrusion detection systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* 38(5), 649-659, 2008.

[19] Montazeri Shatoori, M., Davidson, L., Kaur, G., Habibi Lashkari, A., "Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic," in *The 5th IEEE Cyber Science and Technology Congress*, Calgary, Canada, 2020.

[20] Hoyt, R. E., Snider, D. H., Thompson, C. J., Mantravadi, S., "IBM Watson analytics: automating visualization, descriptive, and predictive statistics," *JMIR public health and surveillance*, 2(2), 157, 2016.

[21] Regkas, G., "Empowering Citizen Data Scientists with IBM Watson AutoAI," in https://towardsdatascience.com/empowering-citizen-data-scientists-with-watson-autoai-49a087df99e5, 2020.

[22] Li, X., Ye, N., "Decision tree classifiers for computer intrusion detection," *Journal of Parallel and Distributed Computing Practices,* 4(2), 179-190, 2001.

[23] Geurts, P., Ernst, D., Wehenkel, L., "Extremely randomized trees," *Machine learning,* 63(1), 3-42, 2006.

[24] Verma, P., Anwar, S., Khan, S., Mane, S. B., "Network intrusion detection using clustering and gradient boosting," *9th International Conference on Computing, Communication and Networking Technologies (ICCCNT) IEEE*, 2018, 1-7.

[25] Dhaliwal, S. S., Nahid, A.-A., Abbas, R., "Effective intrusion detection system using XGBoost," *Information,* 9(7), 149, 2018.

[26] Alzamzami, F., Hoda, M., Saddik, A. El, "Light Gradient Boosting Machine for General Sentiment Classification on Short Texts: A Comparative Evaluation," *IEEE Access,* 2020.

[27] Farnaaz, N., Jabbar, M., "Random forest modeling for network intrusion detection system," *Procedia Computer Science,* 89(1), 213-217, 2016.

[28] Lashkari, A. H., Seo, A., Gil, G. D., Ghorbani, A., "CIC-AB: Online ad blocker for browsers," *International Carnahan Conference on Security Technology (ICCST) IEEE*, 2017, 1-7.