Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 10s (2025)

Investigate Effect of Various Feature Selection Methods for Attack Detection of DOH Traffic

Vaka Padmavathi[1], Dr. Bobba Basaveswararao[2], Simhadri Mallikarjuna Rao[3*], Dr.Guntupalli Neelima[4]

1,2,4 Department of Computer Science & Engineering, Acharya Nagarjuna University, Guntur, 522510, India.

3 Assistant Professor, Vasireddy Venkatadri International Technological University, Nambur,India

** Corresponding author's Email: mallikarjun1254@gmail.com*

Article History:

Received: 12-01-2025

Revised: 15-02-2025

Accepted: 01-03-2025

Abstract:

DNS over HTTPS (DoH) is advance version to the current DNS protocol. Which enhance the security and privacy to the internet browsing. In a existing DNS protocol, when address is typed in the browser, the device queries a DNS server to translate typed web address into IP address. This unscripted translate process means DNS queries can be read and monitored by the third party, leading to unsecure browsing. To prevent this unsecure browsing, DoH protocol is introduced. DoH is a modern protocol that performs Domain Name System (DNS) resolution over the HTTPS protocol; enhance the privacy and integrity of DNS queries through secure, encrypted channels. Past research is carried out on DOH using various classifiers and feature selection techniques, which are not given satisfactory results. This research uses machine learning (ML)-based algorithms to address the challenge of identifying malicious DoH connections. More specifically focused on efficacy of the k-NN classifier for detecting DoH tunnels by comparing its performance before and after applying feature selection techniques. Principal Component Analysis (PCA), Lasso Regression (LR), Logistic Regression (LoR), and Random Forest (RF) are the feature selection techniques which enhance the robustness and efficiency of the classifier. Two main approaches to feature selection are explored: selecting features before training the k-NN model to reduce dimensionality and eliminate potential noise, and selecting features after training to leverage insights from the k-NN model and prioritize important features. The k-NN algorithm's performance is evaluated before and after feature selection to ensure consistency and reliability in detecting malicious DoH connections. The results demonstrated promising outcomes, achieved tremendous accuracy in the classification task.

Keywords: DNS over HTTPS, Malicious DoH, Machine Learning, Classifier, K-NN Algorithm, Feature Selection Techniques.

1. Introduction

DNS plays a vital role in internet communication by translating domain names (e.g., www.example.com) into IP addresses (e.g., 192.0.2.1), enabling computers to locate and connect to resources seamlessly. Translation of human readable domain names into machine-usable IP addresses and vice versa is an essential feature that enables a user-friendly usage of the network services. Traditionally, this mechanism is performed by Domain Name System (DNS) [1,2] in the

internet environment. DNS traffic plays a vital role in various security systems. Before an application can establish a connection, it must translate a domain name, making DNS traffic a key indicator of potential security threats within network traffic [3]. The ability to read these translated domain names allows application firewalls to enforce security policies and enables intrusion detection systems to identify suspicious activities, such as botnet operations [4]. A surge in DNS queries can suggest the presence of communication tunnels over DNS, as examined in various studies [5]. This type of suspicious traffic may indicate data exfiltration, which is critical to detect promptly, particularly in commercial settings. Recently, visibility into DNS communication has been linked to potential eavesdropping and user activity profiling, often driven by commercial interests. This is because anyone along the path of the DNS query, especially Internet Service Providers (ISPs), can observe the contents of these queries, revealing users' online activities [6]. To enhance user privacy and reduce the profiling of DNS traffic by network operators, the Internet engineering community developed DNS over HTTPS (DoH) as a natural response.

The main focus of this paper is on DNS over HTTPS (DoH), was introduced in October 2018, as detailed in [7]. Communication through DoH is encrypted, making it visible only to the user and their DoH service provider, thereby rendering profiling nearly impossible. In addition to encrypting DNS data, DoH shifts the visibility of queried domain names from local DNS providers to more centralized DoH providers, which can be beneficial for users, as explored in [8]. Now a days, DNS not only addresses translation, also provide authentication and better security services to internet applications [9]. Recent advancements in cyber security, especially in addressing DNS over HTTPS (DoH) attacks, have led to the development of innovative techniques. A notable approach features a two-layered intrusion detection system that employs the Random Forest ensemble classifier [10]. In this approach, Random Forest generates multiple decision tree models during the training phase. These models are specifically crafted to distinguish between benign and malicious DoH traffic, while also effectively identifying non-DoH traffic. The contribution of this work can be summarized as follows:

1) The primary objective of any ML/DL algorithms using to detection of DoH traffic is evaluated with minimum computational complexity and without sacrificing the accuracy. In this study the k-NN Classifier is identified for evolutionary process of DoH traffic.

2) To achieve above objective, there is a need to investigate proper feature selection method among the several methods for reducing computational complexity. For this purpose, a comprehensive performance analysis of feature selection methods is carried out and discuss pros and cons for each method.

3) Principal Component Analysis (PCA), Lasso Regression, Logistic Regression, and Random Forest methods are chosen for comparative analysis and the experiments are conducted on adopted dataset CIRA-CIC-DoH-Brw-2020.

4) Finally decisive conclusions are drawn based on experimental results, which may be helpful to the security professionals for mitigating and prevention of malicious activities of DNS over HTTP systems.

The remaining paper is organized as follows: Section 2 discusses about the past research related to detection of malicious connections for DoH traffic using ML/DL algorithms. The description of the dataset, four feature selection methods and evolutionary process of feature selection methods are depicted in Section 3. In Section 4, obtained performance metrics values are presented and also discuss the merits and demerits of each feature selection method . Finally the conclusions and future scope of this work is given in section 5.

2. Literature Survey

This section briefly discusses the research carried out by different researchers in the field of detecting DNS over HTTPS (DoH) using various approaches, focusing particularly on classification techniques. “A Survey of Feature Selection Techniques in Machine Learning with Applications” by L. A. Smith and J. T. Wang (2018): This survey covers various feature selection techniques and their applications, "providing a foundational understanding of ensemble methods in feature selection “Detecting DNS over HTTPS Traffic Using Ensemble Feature-based Machine Learning “ by Sajal Saha [2023]:proposes an advanced approach for classifying DNS over HTTPS (DoH) traffic using machine learning. This work introduces an ensemble feature selection technique that aims to enhance the accuracy and robustness of DoH traffic classification. “Detecting Malicious DNS over HTTPS (DoH) Connections via Machine Learning Techniques” by MHD Raja Abou Harb (2021) introduces a novel approach to classifying malicious DNS over HTTPS (DoH) connections through the use of advanced machine learning techniques. This work focuses on improving the detection of malicious activities in DoH traffic by integrating a feature selection process with multiple supervised learning algorithms. “Feature Engineering and Machine Learning Model Comparison for Malicious Activity Detection in DNS-Over-HTTPS Protocol” by Matthew Behnke (2021) investigates the effectiveness of various machine learning classifiers in detecting malicious activity within the DNS-over-HTTPS (DoH) protocol. By leveraging a publicly available dataset, Behnke conducts a comprehensive comparison of ten different machine learning models, utilizing stratified 10-fold crossvalidation to ensure robust evaluation and comparison. “An Ensemble Framework for Detection of DNS-Over-HTTPS (DoH) Traffic” by Akarsh Aggarwal (2021) proposes an innovative ensemble framework for detecting DNS-over-HTTPS (DoH) traffic. This approach leverages advanced ensembling techniques to enhance the accuracy and reliability of DoH traffic detection, addressing the challenges associated with identifying anomalies in network traffic. “An Explainable AI-Based Intrusion Detection System for DNS over HTTPS (DoH) Attacks” by Tahmina Zebin (2021) presents a robust approach for detecting and classifying DNS over HTTPS (DoH) attacks using an explainable AI-based intrusion detection system. Leveraging the publicly available CIRA-CIC-DoHBrw-2020 dataset, Zebin's work focuses on achieving high accuracy in identifying DoH attacks while providing transparency and interpretability through explainable AI methods.

3. Proposed DoH Detection Model

The proposed DoH model enhances security in DNS over HTTPS (DoH) connections using a k-Nearest Neighbours (kNN) approach integrated with PCA, Gini Index, and Entropy for feature selection. It starts with raw data input, which undergoes cleaning and preparation in the Data Preprocessing Module. The processed data is then stored in the Database Module. The Feature Selection Module applies PCA to identify the most significant features, storing them in the Feature

Library. These features are used by the Classification Module to train a k-NN classifier that predicts outcomes based on new data. The classifier's predictions are provided as results. The Evaluation Module assesses the classifier's performance, and the resulting feedback is used to retrain the classifier, creating a feedback loop that continuously enhances the model's accuracy and effectiveness. Finally, patterns for malicious and benign, as well as DoH and non-DoH traffic, are generated. These features are typically collected and analyzed in network traffic, performance monitoring, and cybersecurity contexts to understand communication patterns, detect anomalies, and assess overall network health and security. Each feature provides specific metrics or characteristics about the network traffic or communication behavior between machines. The list of all 34 features and their descriptions and examples of the adopted dataset are listed in Table 1.

Table.1 List of 34 statistical features and their description of the dataset

Feature #	**Feature Name**	**Description**
1	SourceIP	IP address of the source machine
2	DestinationIP	IP address of the destination machine
3	SourcePort	Port number of source machine
4	DestinationPort	Port number of destination machine
5	TimeStamp	Time stamp of the traffic packet
6	Duration	Duration of the communication
7	FlowBytesSent	Total number of bytes transmitted from source to destination
8	FlowSentRate	% of bytes transmitted from source to destination
9	FlowBytesReceived	Total number of bytes transmitted from destination to the source
10	FlowReceivedRate	% of bytes transmitted from destination to source
11	PacketLengthVariance	Variance value for the length of the packet
12	PacketLengthStandardDeviation	Standard Deviation for the length of the packet
13	PacketLengthMean	Mean of packet length
14	PacketLengthMedian	Median of packet length
15	PacketLengthMode	Mode value of packet Length
16	PacketLengthSkewFromMedian	Skewness from median for the packet

		length
17	PacketLengthSkewFromMode	Skewness from mode for the packet length
18	PacketLengthCoefficientofVariation	Coefficient of the variation value for the packet length
19	PacketTimeVariance	Variance of the time of packet received
20	PacketTimeStandardDeviation	Standard deviation value for the packet time
21	PacketTimeMean	Mean value for the packet time
22	PacketTimeMedian	Median value for the Packet time
23	PacketTimeMode	Mode value for the packet time
24	PacketTimeSkewFromMedian	Skewness of median for the packet time
25	PacketTimeSkewFromMode	Skewness of mode for the packet time
26	PacketTimeCoefficientofVariation	Coefficient of variation values for packet time of the server
27	ResponseTimeTimeVariance	Time variance of response time
28	Response TimeTimeStandardDeviation	Standard deviation values for response time of the server
29	ResponseTimeTimeMean	Mean value of response time of the server
30	ResponseTimeTimeMedian	Median time of response time of the server
31	ResponseTimeTimeMode	Mode of the response time of the server
32	ResponseTimeTimeSkewFromMedian	Skewness from median for response time of the server
33	ResponseTimeTimeSkewFromMode	Skewness from mode of the response time
34	ResponseTimeTimeCoefficientofVariation	Coefficient of variation for response time

Proposed DoH detection model is shown Figure.2. This model describes that the raw data is pre-processed. Identifying and selecting relevant features (attributes) from the dataset are most informative for detecting malicious DoH traffic. Split the dataset into training set and testing set. Train the k-NN model using the selected features from the feature selection module. Assess the performance of the k-NN model using metrics such as accuracy, precision, recall, and F1-score. Re-assess feature importance using the trained k-NN model. Refine feature selection based on insights gained and re-train the k-NN model if necessary. Evaluate the final model using the testing set.

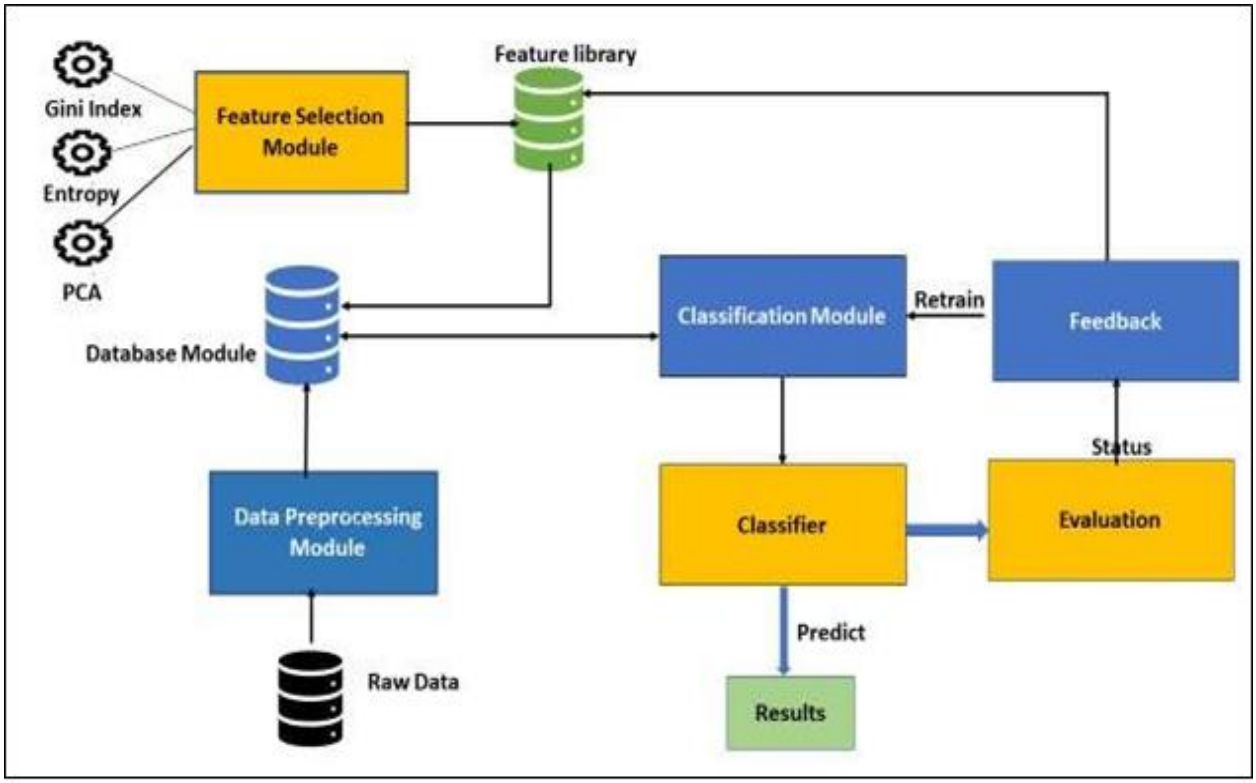


Figure 1: Evolutionary process of Feature Selection Methods

The proposed Detection of DNS over HTTPS (DoH) model described in Figure 2 follows a structured approach to identify and classify malicious DoH traffic.

3.1 Data Pre-processing: This step involves handling missing or erroneous data points in the dataset. It may include techniques such as imputation (replacing missing values) or removal of incomplete records. This step ensures that the data is in a suitable format for feature selection and model training.

Feature Selection: Feature selection aims to choose attributes that are most informative for detecting malicious DoH traffic. This process helps in reducing dimensionality and focusing on the most predictive features.

i. Principal Component Analysis (PCA):

PCA reduces the dimensionality of data while preserving as much variance as possible. This is achieved by transforming the original features into a new set of orthogonal features called principal components.

Covariance Matrix Calculation

Given the data matrix X, where rows represent samples and columns represent features, the covariance matrix C is calculated as:

$$C = \frac{1}{n-1} X^T X \qquad (1)$$

Where, n is the number of samples, X^T is the transpose of X.

ii. Lasso Regression (LR)

KNN is a popular supervised learning algorithm primarily used for classification tasks. KNN works by identifying the closest training examples in the feature space, making it effective for detecting patterns associated with different types of attacks. However, KNN alone can struggle when irrelevant or redundant features are present, potentially leading to overfitting or reduced accuracy. LR, is a linear regression that applies L1 regularization, is useful for feature selection. By penalizing less informative features and assigning them a zero weight, Lasso helps reduce dimensionality and improve model interpretability, retaining only the most significant features. It is an ideal companion for KNN, as it can preprocess data by reducing feature noise.

Lasso (Least Absolute Shrinkage and Selection Operator) regression minimizes the sum of squared errors with a penalty proportional to the absolute value of the coefficients, effectively setting some coefficients to zero and performing feature selection. This can help to reduce dimensionality before using KNN. The LR objective function is:

$$\min_{\beta}\left(\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j-1}^{p} |\beta_j)\right| \qquad (5)$$

y_i: Target variable for observation i.

x_{ij}: Feature *j* for observation i.

β_j: Coefficient for feature i.

λ Regularization parameter controlling the penalty for larger coefficients.

iii. Logistic Regression (LoR)

In recent years, with the exponential growth of digital traffic and increasing dependency on internet-based applications, the need for robust cybersecurity measures has become more pressing. Cyber-attacks continue to evolve, exploiting vulnerabilities across various layers of network and application protocols. Among these attacks, Domain Name System (DNS) abuse through encrypted DNS protocols, such as DNS over HTTPS (DoH), has become particularly challenging for detection mechanisms, as the encryption obscures traditional security inspections. The CIRA-CIC-DoHBrw-2020 dataset provides a valuable resource to analyze and classify such attacks, enabling researchers and security professionals to develop models that can effectively detect and prevent these threats. This project explores the use of logistic regression for classifying types of cyber attacks in the CIRA-CIC-DoHBrw-2020 dataset. LoR, a statistical method commonly used for binary and multiclass classification problems, is particularly suitable for understanding relationships between various network features and the likelihood of attack. By leveraging LoR interpretability and efficiency, this study aims to achieve accurate classification results while providing insights into feature importance and attack behavior patterns in encrypted DNS traffic. The scope of this project involves data preprocessing, feature selection, and model training and evaluation. Emphasis will be placed on assessing the performance of logistic regression in comparison to other algorithms for cyber attack detection and evaluating the model's efficacy in real-time applications. Through this approach, this project contributes to the development of lightweight and interpretable solutions for detecting

encrypted DNS-based attacks, which can be integrated into real-time security systems. The following steps used in this context:

Step 1: Sigmoid Function (Logistic Function): The logistic regression model uses the sigmoid function to transform the linear combination of input features into a probability output between 0 and 1. The sigmoid function is defined as:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \qquad (6)$$

Where $z = w^T x + b$ is the linear combination of the weights w and features x, with b as the bias term.

Step 2: Model Output (Probability of Class 1): The output of the logistic regression model represents the probability that a given instance belongs to class 1 (e.g., indicating a cyber attack):

$$\hat{y} = P(y = 1/x = \frac{1}{1 + e^{-(w^T x + b)}} \qquad (7)$$

Here, $w^T x + b$ represents the linear decision boundary.

Step 3: Decision Boundary: For binary classification (cyber attack vs. normal), a threshold τ is set, typically at 0.5. If $\hat{y} \geq \tau$, the instance is classified as class 1 (cyber attack), and if $\hat{y} < \tau$, it is classified as class 0 (normal traffic).

$$Predicted\ Class = \begin{cases} 1 & if\ \hat{y} \geq \tau \\ 0 & if\ \hat{y} < \tau \end{cases} \qquad (8)$$

Step 4: Loss Function (Log-Loss or Binary Cross-Entropy): To optimize the logistic regression model, the log-loss (or binary cross-entropy) is used as the cost function:

$$J(w, b) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})] \qquad (9)$$

Here, m is the number of training examples, $y^{(i)}$ is the actual label, and $\hat{y}^{(i)}$ is the predicted probability for the i-th instance.

Step 5: Parameter Update (Gradient Descent): The model parameters *w* and *b* are updated using gradient descent to minimize the loss function:

$$w := w - \propto \frac{\partial J}{\partial w} \qquad (10)$$

$$b := b - \propto \frac{\partial J}{\partial b} \qquad (11)$$

Where α is the learning rate, and $\frac{\partial J}{\partial w}$ and $\frac{\partial J}{\partial b}$ are the gradients of the loss function with respect to w and b, respectively.

By applying these equations to features derived from the CIRA-CIC-DoHBrw-2020 dataset, logistic regression can be used to distinguish between normal and malicious network behaviors.

iv. Random forest (RF)

In this context, another RF algorithm is model that leverages the strengths of both algorithms to improve detection accuracy. Random Forest, a robust ensemble learning method, complements KNN by aggregating the decisions from multiple decision trees, thereby reducing overfitting and enhancing generalizability. The RF is used as feature extraction technique that obtaines the significant attack patterns which is also called features shows the both accurate and resilient to various attack patterns. This approach offers a promising solution for real-time, accurate classification of cyber attacks in encrypted traffic, thereby enhancing the security of networks against sophisticated threats in the DoH environment. In the following sections, we discuss the methodology, implementation, and results of applying this combined approach to the CIRA-CIC-DoHBrw-2020 dataset. RF uses an ensemble of decision trees for classification, where each tree contributes a "vote" to the final prediction.

1. ***Prediction from Each Tree:*** Given M trees, each tree T_m (where m=1, 2, …., M) predicts a class label $\hat{y}_{m.}$ Each T_m is trained on a bootstrapped subset of the data.

2. ***Majority Voting:*** The final class prediction $\hat{y}$ for a given sample x is determined by majority voting across all M trees.

$$\hat{y} = mode\{\hat{y}_1, \hat{y}_2, \ldots..,, \hat{y}\mathrm{M}\} \qquad (12)$$

4. Results and Discussions

In this section the feature selection methods Principle Component Analysis(PCA), Linear Regression(LR), Lasso Regression(LoR), Random Forest(RF) are evaluated through K-NN Classification with adoption of CIRA-CIC-DoHBrw-2020 dataset.

The dataset contains two types of DNS over HTTPS traffic, they are DoH, and Non DoH traffic. The DoH traffic includes Malicious and Benigin traffics. The K-NN classification algorithm is adopted for classification evolutionary process. The classification process divided into two models i.e. DoH and Non-DoH (Model 1), and the DoH traffic classified into Malicious and Benigin (Model 2). The performance metrics are presented below for four feature selection methods of two models.

I. PCA Index based feature selection

The feature subsets are obtained based on eigen values and eigenvectors of the covariance matrix. The eigenvectors are directions (axes) in which data varies under PCA, and the eigen values tell how much variation is there along them. Eigen values are arranged in descending order. The first principal component (PC1), which explains most variance in the data, is the eigenvector corresponding to the largest eigen value. A subset k of principal components is selected based on the explained variance ratio (sum of eigen values of selected components / total eigen values). It establishes the dimensionality of the condensed feature space. The other features with numbers 10,15,20,25,30, 16,18,21,22,24 and 13,15,16,17,18 are coefficients (from eigenvectors) that define the contribution of each original feature to the new feature. These three feature subsets are evaluate for model 1 and model 2 through K-NN classification and the results are presented in Table 2 as well as graphs.

Table 2: Performance Measures for Model 1

Features #	Accuracy	Precision	Recall	F1	**ROC AUC**
10,15, 20,25,30	0.996 958	0.99 7543	0.999 179	0.998 360	0.984 0
16,18, 21,22,24	0.997 607	0.99 7981	0.999 439	0.998 709	0.986 986
13,15, 16,17,18	0.9 97441	0.99 7986	0.999 261	0.99 862	0.986 459

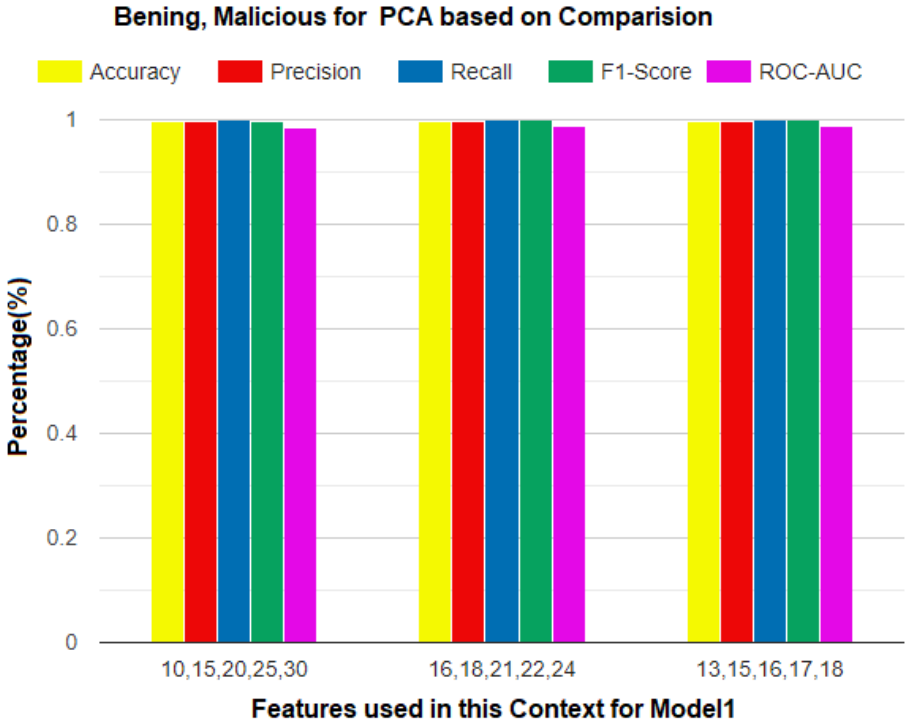


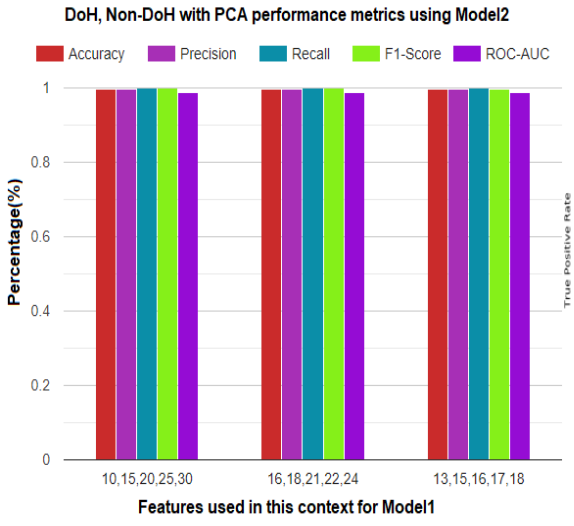
Figure 2: Bening-Malicious for model1

The K-NN Classification evaluating results for various metrics are presented below with discussions.

This figure shows a comparison of the performance metrics for Model-1 based on different sets of features used in the context of benign and malicious classification with PCA (Principal Component Analysis). From the above observations on Performance Metrics are Accuracy, Precision, Recall, F1-Score, and ROC-AUC reach nearly 100% for all three feature sets. This indicates that Model-1 performs exceptionally well in classifying benign and malicious data in every feature context. There is minimal to no variation and exhibits consistency in the metrics across the three feature sets. Set-1 are 10, 15, 20, 25, 30, set -2 are 16, 18, 21, 22, 24 set-3 are 13, 15, 16, 17, 18 This implies that the model's performance is stable and not significantly affected by changes in the selected feature sets.

Table 3: Performance Measures for Model-2(B-M)

Features #	Accuracy	Precision	Recall	F1	ROC AUC
10,15, 20,25,30	0.997 552	0.998 161	0.999 199	0.998 680	0.987 939
16,18, 21,22,24	0.99 775	0.99 814	0.999 439	0.998 789	0.987 995
13,15, 16,17,18	0.997 737	0.998 304	0.999 261	0.998 782	0.988 545



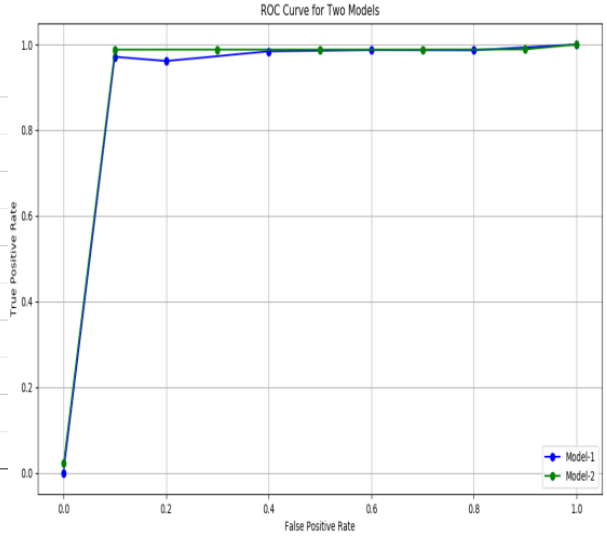


Figure 3: Feature based performance of Model2 Figure: 4 PCA based ROC Curve for two models as Shown in Table 5.

The curves for Model-1 (blue) and Model-2 (green) overlap significantly, indicating that the two models perform almost identically in terms of classification. Both curves flatten after the steep rise, maintaining a high TPR as FPR increases. This suggests that the models perform consistently well across different decision thresholds. Although the AUC is not explicitly mentioned, the close-to-optimal shape of the curves implies that the AUC for both models is likely near 1. This indicates excellent discrimination ability.

II. LR Index based feature selection

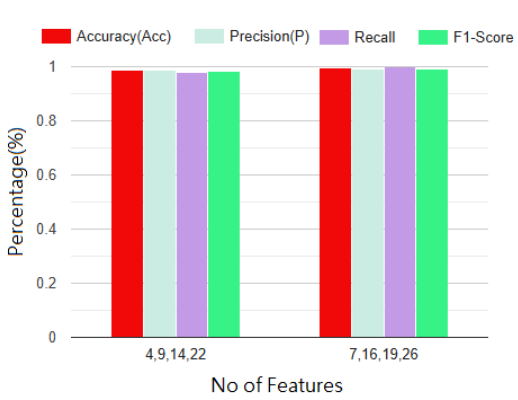
The feature subsets are obtained on magnitude of coefficients(weights) reflects the importance of feature selection. Here each feature coefficient is associated with p-value. Based on these values least important features are being removed and refit the features until the optimal subset of features are finalized. It shows the dimensionality of proposed feature space. The final feature subset groups 4,9,14,22, and 7,16,19,26 are define the contribution of each original feature to new feature. These two subsets are evaluate for model1 and model2 through K-NN classification using Table 4 and 5 respectively and final ROC curves are represented in graphs.

Table 4 : Algorithms based LR for Performances with DoH Non-DoH

Features #	Accuracy	Precision	Recall	F1	ROC AUC
4, 9,14, 22	0.99876	0.99123	0.99912	0.999126	0.99765
7, 16, 19, 26	0.99231	0.98098	0.9812	0.99756	0.98023

Table 5 : Algorithms based LR for Performances B-M with Entropy

Features#	Accuracy	Precision	Re call	F1	**ROC AUC**
4, 9,14, 22	0.98784	0.98781	0.97813	0.98541	0.97681
7, 16, 19, 26	0.99456	0.99129	0.99801	0.98987	0.99876



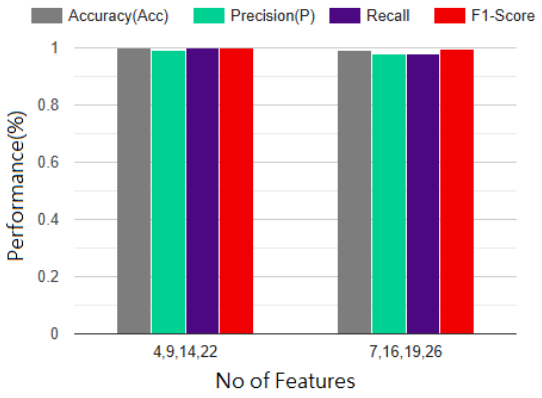


Figure 5: Performance of LR Algorithm for Benign and Malicious data with Entropy Figure 6: Performance of LR Algorithm for DoH, Non-DoH

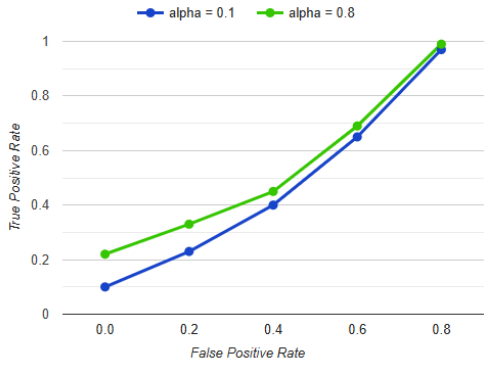


Figure 7: Performance in terms of AU-ROC for two types of Data.

From Table 11, the features 4,9,14 and 7,16,19,26 were extracted according to feature coefficients (β_i) set exactly to zero. The non-zero coefficients are interpreted as relevant feature selection. We also learned from these features that these are the most determining predictors.

From the above graph we can say that as Learning Rate increases TPR is increases it means more positive comes. But it should ideally be accompanied by a high precision to avoid unnecessary trade-offs. Balancing TPR and precision ensures better overall model performance.

III. LOR Index based feature selection

LoR regression model works over a range of λ values to determine the optimal subset of features. Here Co-efficients that are non-zero after training represents selected features. Where features with Co-efficients of zero are excluded. The features 11,16,21,28 and 6,12,18,26 based on magnitude of the weights, the sign of the weights indicate if it is positively correlated (+) or negatively correlated (-) with the output. These weights correspond to the coefficients of features in the prediction equation for the dependent variable. The higher the absolute values of weights, the more important the features. These are positive scalars that scale up the class label, and negative scalars that scale down it.

Table 6 : Algorithms based LOR for Performances DoH, Non-DoH

Features#	Accuracy	Precision	Recall	F1	ROC AUC
11,16,21,28	0.99123	0.99453	0.98971	0.98981	0.99121
6,12,18,26	0.99531	0.98123	0.99812	0.99987	0.99123

Table 7: Algorithms based LOR for Performances B-M with entropy

Features#	Accuracy	Precision	Re call	F1	**ROC AUC**
11,16,21,28	0.97898	0.99876	0.98431	0.99781	0.98781
6,12,18,26	0.98987	0.99321	0.99123	0.99786	0.98987

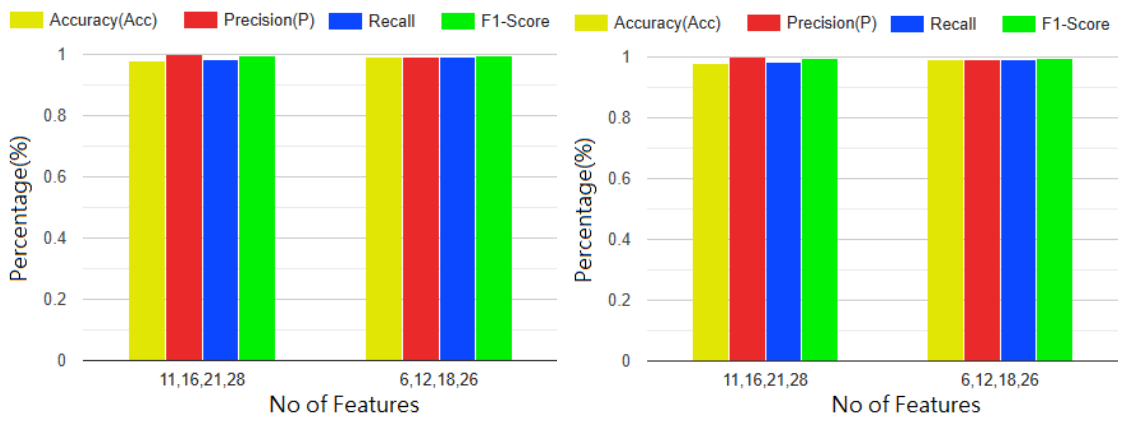


Figure 8: Performance of LOR Algorithm for Benign and Malicious data with Entropy.
Figure 9: Performance of LOR Algorithm for DoH, Non-DoH.

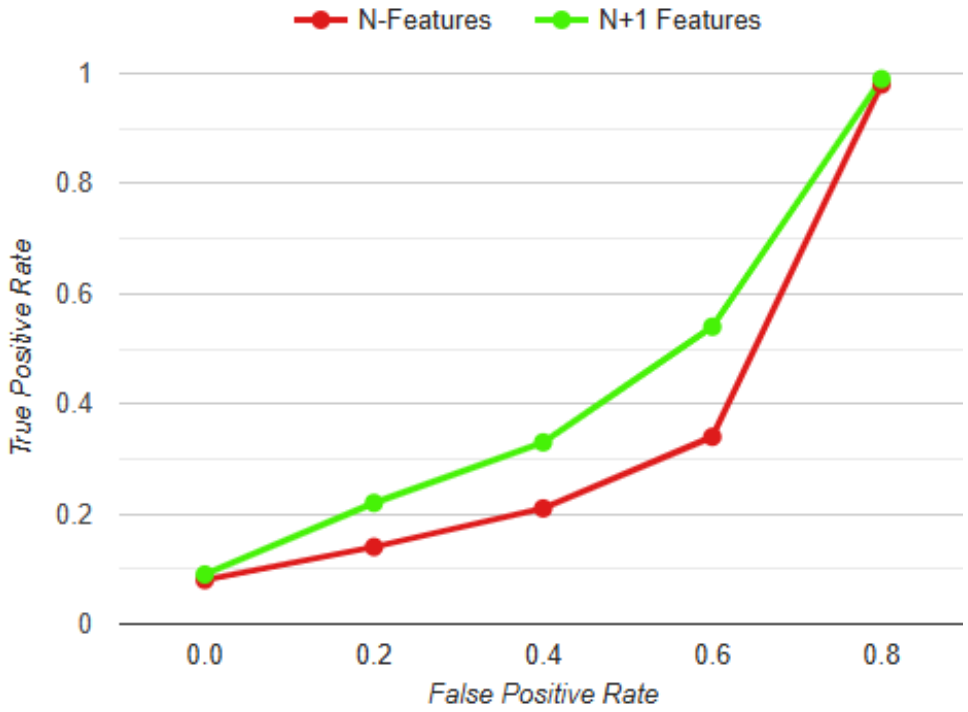


Figure 10: Performance in terms of AU-ROC for two types of Data.

From the above figure as features increases TPR increases. An increase in TPR (recall) can improve the F1-Score, which is the harmonic mean of precision and recall. However, this depends on whether precision remains steady or improves. If TPR is increased by lowering the decision threshold, it could lead to a higher number of false positives (FP), which might reduce precision.

IV. RF Index based feature selection

In Random Forest features with higher importance scores contribute more to reducing the variance of models. Features are ranked based on the above scores are set with threshold value to select the features. Table 8: obtain features 8, 10, 15 and 9, 17, 22 by iteratively extracting the features that best split the data and improve performance during training. Features are sorted by their importance scores. Keep the features that have the highest importance scores and drop the remaining ones. After that the top features are extracted as features for the next tasks which could be training another model. These are the most significant features to lower computational complexity.

Table 8: Algorithms based RF for Performances for DoH, Non-DoH

Features Used	Accuracy	Precision	Recall	F1	ROC AUC
8, 10, 15	0.98971	0.98971	0.97601	0.99878	0.99981
9, 17, 22	0.99234	0.99123	0.99341	0.98987	0.99982

Table 9 : Algorithms based RF for Performances for B-M with Entropy

Features Used	Accuracy	Precision	Re call	F1	**ROC AUC**
8, 10, 15	0.98761	0.99876	0.99871	0.99098	0.99823
9, 17, 22	0.99876	0.99321	0.99541	0.99887	0.97891

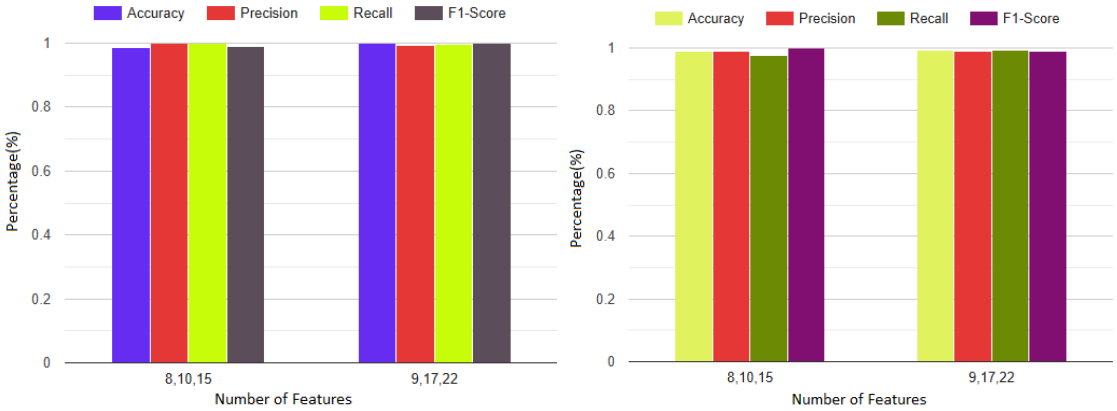


Figure 11: Performance of RF Algorithm for Benign and Malicious data with Entropy. **Figure 12: Performance of LOR Algorithm for DoH, Non-DoH.**

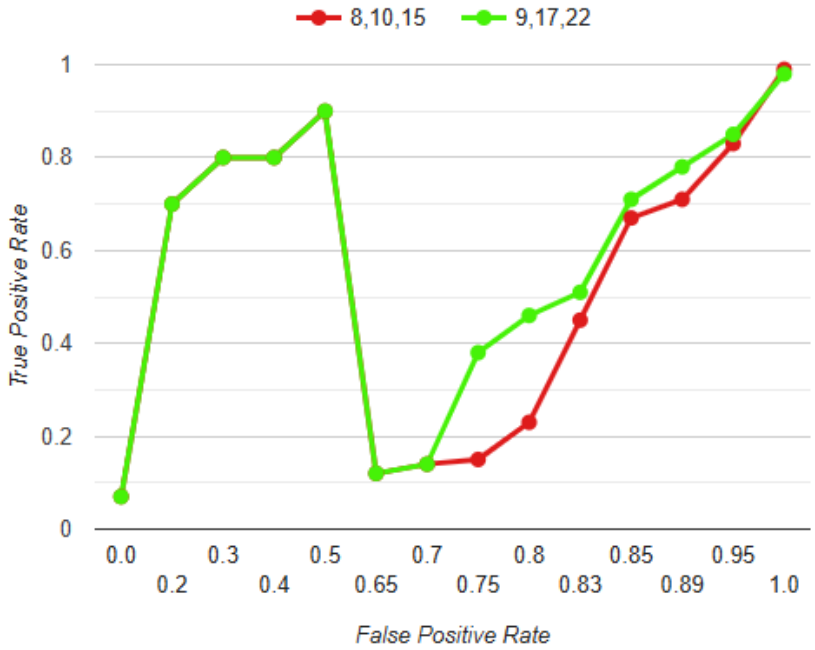


Figure 13: Performance in terms of AU-ROC for two types of Data using RF

The green curve (label: 9,17,22) generally outperforms the red curve (label: 8,10,15), as it has a higher TPR for a given FPR in most segments of the curve. The red curve has a notable flat section with very low TPR at an FPR range around 0.65–0.75. This indicates poor performance in this range, where the model struggles to identify positives correctly. The green curve has a more consistent rise in TPR across all FPR ranges, showing robustness. The green curve (9,17,22) may be better suited for applications prioritizing higher sensitivity (more true positives) at lower false positive rates. The red curve (8,10,15) might perform better in situations where a higher FPR is acceptable after 0.85, though its overall performance is less robust.

5. Conclusion

In this work, the feature extraction techniques PCA (Principal Component Analysis), LR, LoR, and RF feature selection methods represents a robust strategy for enhancing security in DNS over HTTPS (DoH) connections. The model effectively reduces dimensionality by preprocessing data with PCA while retaining significant variance, optimizing computational efficiency without sacrificing accuracy. Using the LR, LoR, and RF for feature selection ensures that the most informative features is prioritized, thereby improving the model's ability to detect malicious DoH traffic accurately using KNN algorithm. Through rigorous evaluation using metrics such as accuracy, precision, recall, and F1-score, the approach's effectiveness in distinguishing between legitimate and malicious DoH traffic has been demonstrated. This systematic approach enhances detection capabilities and provides insights into the underlying characteristics of DoH traffic patterns, which are crucial for developing proactive cyber security measures.

DNS traffic provides critical insights into potential security threats. Its analysis is vital for intrusion detection systems and application firewalls to identify suspicious activities, such as botnet communications or data exfiltration via DNS tunneling. Traditional DNS is susceptible to eavesdropping and user profiling because DNS queries are visible to entities along the communication path, such as ISPs. This visibility can reveal users' online activities, raising privacy concerns. DoH changes the dynamics of DNS traffic visibility, moving it from local DNS providers to centralized DoH providers. While this improves user privacy, it also raises concerns about the concentration of data with fewer providers.

Advanced intrusion detection systems, such as the two-layered approach using the Random Forest ensemble classifier, are effective in distinguishing between benign and malicious DoH traffic. These systems also excel in identifying non-DoH traffic, improving network security. The development of DoH exemplifies the balance between enhancing user privacy and addressing security concerns. However, it also underscores the need for vigilance to ensure that centralization and encryption do not become tools for malicious actors.

References

[1] P. Mockapetris, “Domain names - concepts and facilities,” RFC 1034 (Standard), Internet Engineering Task Force, Nov. 1987. [Online]. http://www.ietf.org/rfc/rfc1034.txt

[2] Paul Mockapetris. 1987. Domain names - implementation and specification," RFC 1035 (Standard), Internet Engineering Task Force, Nov. 1987. [Online]. http://www.ietf.org/rfc/rfc1035.txt.

[3] Cejka, Z. Rosa and H. Kubatova, "Stream-wise detection of surreptitious traffic over DNS," 2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Athens, Greece, 2014, pp. 300-304, doi: 10.1109/CAMAD.2014.7033254.

[4] Yury Zhauniarovich, Issa Khalil, Ting Yu, and Marc Dacier "Survey on Malicious Domains Detection through DNS Data Analysis" ACM Comput. Surv. July 2018, Article 67 - 36 pages. DOI:10.1145/3191329

[5] Mahmoud Sammour 1 , Burairah Hussin 2 , Mohd Fairuz Iskandar Othman "DNS Tunneling: a Review on Features" International Journal of Engineering & Technology, July,2018. Page 1-5. DOI:10.14419/ijet.v7i3.20.17266.

[6] Olivier van der Toorn, Moritz Müller, Sara Dickinson, Cristian Hesselman, Anna Sperotto, Roland van Rijswijk-Deij, Addressing the challenges of modern DNS a comprehensive tutorial, Computer Science Review, Volume 45, 2022.

[7] Paul E. Hoffman and Patrick McManus. 2018. DNS Queries over HTTPS (DoH). Technical Report 8484. https://doi.org/10.17487/RFC8484.

[8] Ben Dickson. 2019. Does Google Chrome's DNS-over-HTTPS (DoH) feature enhance your privacy. https://bdtechtalks.com/2019/12/11/ google-chrome-dns-over-https-privacy/.

[9] Jalalzai MH, Shahid WB, Iqbal MMW (2015) DNS security challenges and best practices to deploy secure DNS with digital signatures, in 2015 12th International Bhurban Conference on Applied Sciences and Technology (IBCAST), pp. 280–285. doi: 10.1109/IBCAST.2015.7058517.

[10] Z. Azam, M. M. Islam and M. N. Huda, "Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree," in IEEE Access, vol. 11, pp. 80348-80391, 2023, doi: 10.1109/ACCESS.2023.3296444.

[11] Caiwu Lu Yunxiang Cao and Zebin Wang " Research on Intrusion Detection Based on an Enhanced Random Forest Algorithm", Appl. Sci. 2024, 14(2), 714. doi.org/10.3390/app14020714.