This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TIFS.2022.3183390,

An Explainable AI-based Intrusion Detection System for DNS over HTTPS (DoH) Attacks

Tahmina Zebin, Shahadate Rezvy, Yuan Luo

***Abstract*—Over the past few years, Domain Name Service (DNS) remained a prime target for hackers as it enables them to gain first entry into networks and gain access to data for exfiltration. Although the DNS over HTTPS (DoH) protocol has desirable properties for internet users such as privacy and security, it also causes a problem in that network administrators are prevented from detecting suspicious network traffic generated by malware and malicious tools. To support their efforts in maintaining a secure network, in this paper, we have implemented an explainable AI solution using a novel machine learning framework. We have used the publicly available CIRA-CIC-DoHBrw-2020 dataset for developing an accurate solution to detect and classify the DNS over HTTPS attacks. Our proposed balanced and stacked Random Forest achieved very high precision (99.91%), recall (99.92%) and F1 score (99.91%) for the classification task at hand. Using explainable AI methods, we have additionally highlighted the underlying feature contributions in an attempt to provide transparent and explainable results from the model.**

***Index Terms*—Secure Computing, Machine Learning, Intrusion Detection System, Explainable AI.**

I. Introduction

DOMAIN Name System (DNS) traffic is crucial for many existing security systems. Since an application must translate a domain name before a connection can be established, DNS traffic can identify many observable security threats in the network traffic. As per the EfficientIP and IDC 2021 Global DNS Threat Report, around 87% of the surveyed organizations have experienced DNS attacks in 2021 which is 8% more than the statistics in 2020 [1]. With the pandemic in recent times, a rapidly increasing number of people remotely working and using various cloud services on a daily basis, an increasing amount of cyber-attacks are disrupting the online services. The impact and cost of attacks remain extremely high and it affects company finances but also brand image and data confidentiality. Organizations have suffered more diverse types of attacks than ever before, showing that cyber-criminals are using all the tools at their disposal to exploit both the DNS protocol and misconfigurations.

Previously, DNS queries were made in plaintext, from an app to a DNS server, using the DNS settings of the local operating system received from its network provider, usually an Internet Service Provider (ISP). In recent times, a new protocol DNS over HTTPS (DoH) has been created to improve users' privacy on the internet. DoH changes this paradigm. DoH encrypts DNS queries, which are disguised as regular HTTPS traffic, hence the DNS-over-HTTPS name. These DoH queries are sent to special DoH-capable DNS servers (called DoH resolvers), which resolve the DNS query inside a DoH request and reply to the user, also in an encrypted manner. DoH can be used instead of traditional DNS for domain name translation with encryption as a benefit [2].

The companies and organizations that have DoH-capable products have been advertising DoH as a way to prevent ISPs from tracking users' web traffic and as a way to bypass censorship in oppressive countries. The readability of translated domain names in the traffic is exploited in application firewalls to check security policies, and intrusion detection systems to detect suspicious connections. Therefore, this paper focuses on the possibilities of encrypted traffic analysis, especially for the purpose of accurate detection of DoH attacks.

The contribution of this paper consists of the following:

- We have implemented one of the very first explainable AI solutions to provide accurate detection and classification of the DNS over HTTPS attacks. We have analyzed DoH communication traffic samples or captured packets. By analysis of traffic samples, we identified easy to interpret DoH traffic features from the CIRA-CIC-DoHBrw-2020 dataset that can provide insight for developing efficient methods of DoH traffic classification and Intrusion detection systems.
- We have proposed a Balanced Stacked Random forest classifier for this task. Our choice of methods and algorithms ensured high accuracy while maintaining transparency on how the model is governing the decision-making process. The improvements in performance using the proposed data split and sub-model development mainly was reflected by the three-fold reduction in training time because of the parallel nature of the sub-models. We have also obtained slightly improved precision and recall performance from the proposed implementation when compared to a generic random forest model trained without a balanced sub-division.
- We have also deployed our model and generated a tailored dashboard for visualization using state-of-the-art explainable AI methods to make the solution transparent to the human user of the system.

The remainder of this paper is organized as follows. Section II introduces the background literature in the recent development in DoH attack detection methods used in recent years, we introduced the concept of explainable AI in this section. We then introduced our dataset and the associated preprocessing stages in Section III. Section IV provides details on the workflow, model architecture and parameter settings for the implemented model. The performance of the developed model

T. Zebin is a lecturer at the School of Computing science, The University of East Anglia, Norwich, UK. Email: {t.zebin}@uea.ac.uk.

S. Rezvy is a lecturer at the Department of Computer Science, School of Science, Technology and Health, York St John University, UK. Email: {s.rezvy}@yorksj.ac.uk.

Y. Luo is a Senior Lecturer at the Faculty of Science and Technology, Middlesex University,London, UK. Email: {y.luo}@mdx.ac.uk.

for attack classification is evaluated in Section V. We presented the explanations from the model using a model dashboard we have deployed from the task in Section VI. Finally, the paper is concluded along with ideas for future work in the very last section.

II. Background Literature

This section discusses the literature relevant for DoH detection or the detection of malicious use of DoH. The related work is divided into multiple parts. It starts with the broad scope of detecting malicious DNS traffic in general including available datasets and the learning techniques used for the classification of encrypted data. The literature then gradually narrows down to the recent related work that has specifically used the dataset similar to this research for the detection of malicious DoH traffic. We then presented some background concepts necessary for the use of explainable AI in DoH intrusion detection systems.

A. DoH attack detection : Learning Techniques

Numerous organizations have less active monitoring plans in terms of security checking on DNS as it is not used for Data transfer, compared to other protocols like Web activity where attacks often take place. A malicious DNS attack can aim to exploit security vulnerabilities on the server that runs the DNS services and extract valuable data such as passwords, usernames, and other personal information. Since most of the Internet's traffic is encrypted and served by large content delivery networks, in many cases, domain name systems are the only clear text sign about the specific service being accessed. DNS Tunneling is a method of cyber-attack that encodes the data of other programs or protocols in DNS queries and responses. DNS tunnels, established between the controlled host and master server disguised as the authoritative domain name server, can be used as a secret data communication channel for malicious activities. Cybercriminals use multiple tunnelling techniques such as FTP-DNS tunnelling, HTTP-DNS tunnelling, HTTPS-DNS tunnelling, and POP3-DNS tunnelling to hide their identity[3]. Owing to the ready evasion of the DNS traffic to bypass the network security mechanism, DNS tunnelling can cause severe damage. DNS tunnelling often includes data payloads that can be added to an attacked DNS server and used to control a remote server and applications.

Several earlier methods proposed to detect malicious DNS traffic include Network and DNS Traffic Analysis [4], [5], Domain name blacklisting, and Detailing of Web Page Content by the visual platform to protect top-level domain name servers against DDoS attacks [6]. Aiello et al.[4] combined principal component analysis (PCA) and mutual information (MI) to calculate a novel metric as the identification index, based on several statistical features. However, they found that the different circumstances of DNS server size or the traffic encapsulated in DNS tunnelling would cause diverse manifestations of the value. Hence, the threshold could only be determined based on the condition of the non-overlapping of the MI value between legitimate and malicious traffic, which is affected by many environmental factors. In other words, it indicates the poor flexibility and generality of this method. Other works have focused on predicting the validity of information coming from the DNS and do not take into account that DNS data for the malicious activity have statistical, temporal and payload related differences, so the results obtained were less effective [6], [7].

Some very recent research focused on analysing and detecting malicious and encrypted DNS traffic using various machine learning techniques. The research in [9] focused on the primary domain as a filter to classify the DNS traffic rather than the queries. The features have been extracted from sub-domains of multiple groups. The author used supervised machine learning for examining DNS traffic and filtering benign and malicious domains. However, this approach has a limitation of the inability to detect malicious queries in the main domain. In which the sub-domain is not enough for detecting the other types of attacks. Banadaki et al. [10] examined the dataset called CIRA-CIC-DoHBrw-2020[8] using several ML algorithms such as (Xgboost, Gradient Boosting, and Light Gradient Boosting algorithms). However, the preprocessing and optimization phase were unclear. Ramakrishnan et. al.[11] propose a NN (Neural Network) based IDS that can quickly respond to attacks by analyzing low-level network details. The proposed scheme is quite limited in terms of the number of features used and low accuracy where it averagely reports 90% accuracy. Jafar et al.[12] explored eight ML methods including Logistic Regression, Stochastic Gradient Descent, Decision tree and Random forest to name a few. The authors reported accuracy value only along with the computational time required to train the model, but there is again no detail on class-wise accuracy and other evaluation matrices. Keeping explainability in mind, we have not included a few deep learning model development available for this dataset. In this research, we are proposing one of the very first explainable AI solutions to provide an accurate solution to detect and classify the DNS over HTTPS attacks. In the next subsection, we will discuss the introductory explainable AI methods used in this research for DoH attack detection.

B. Explainable AI for DoH attack detection

Despite the growing popularity of machine learning models in cyber-security applications (e.g., an intrusion detection system (IDS)), most of these models are perceived as a black-box. The eXplainable Artificial Intelligence (XAI) has become increasingly important to interpret the machine learning models to enhance trust management by allowing human experts to understand the underlying data and to understand the impact of the malicious data to detect any intrusion in the system. The previous studies focused more on the accuracy of the various classification algorithms for trust in IDS. They do not often provide insights into their behaviour and reasoning provided by the sophisticated algorithm. Therefore, in this paper, we have addressed the XAI concept to enhance trust management by exploring the decision tree model in the area of IDS.

Over the last few years, there has been significant progress on Explainable AI. The pursuit of converting these black-box models into transparent and interpretable algorithms has

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TIFS.2022.3183390,

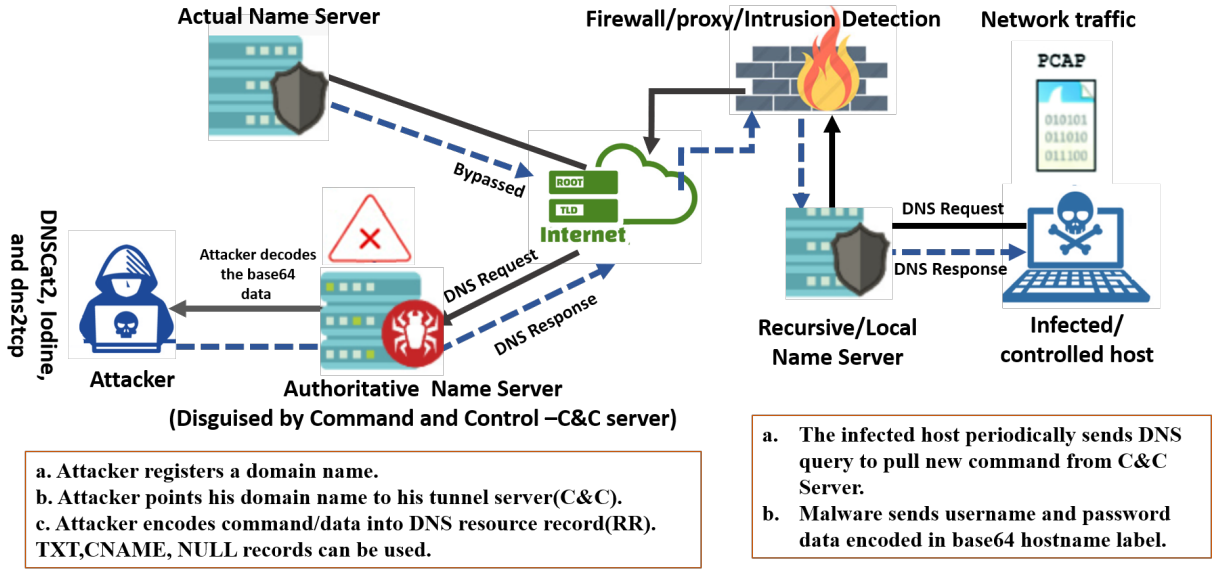


Fig. 1. The technical framework of DNS tunnels for generating Malicious DoH traffic.

TABLE I
TRAFFIC TYPES CAPTURED IN CIRA-CIC-DOHBRW-2020 DATASET [8]

Traffic Type	Instances captured	Description of the Attack
Non-DoH	889,809	Traffic generated by accessing a website that uses HTTPS protocol is captured. In order to capture ample traffic to balance the dataset, thousands of websites from Alexa domain are browsed.
Benign-DoH	19,746	Benign DoH is non-malicious DoH traffic generated using Mozilla Firefox and Google Chrome web browsers.
Malicious-DoH	249,553	DNS tunneling tools such as dns2tcp, DNSCat2, and Iodine are used to generate malicious DoH traffic. These tools create tunnels of encrypted data. Therefore, DNS queries are sent using TLS-encrypted HTTPS requests to special DoH servers.

gained traction in both academia, industry and other users of AI and machine learning models. While many packages and methodologies have developed in recent years, one of the most popular methods today, SHAP (SHapley Additive exPlanations) is a game theory-based approach to explain the output of any ML model [13]. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions. SHAP does a great job in decoding the strength of the influence of the input variables in the predictions with intuitive and engaging visualizations across various aspects of model explainability. SHAP values calculate the feature importance by comparing what a model predicts with and without the feature. However, since the order in which a model sees features can affect its predictions, this is done in every possible order, so that the features are fairly compared.

At the time of writing this paper, there was not much explainable AI literature available for DoH attack detection and classification. Hence we will present a thorough discussion and the explanations from our proposed model using a model dashboard in Section VI in this paper.

III. DATASET DESCRIPTION AND PRE-PROCESSING

For experimenting, we have used the publicly available CIRA-CIC-DoHBrw-2020 dataset[8] from the Canadian Institute for Cybersecurity (CIC). In this dataset, a two-layered approach is used to capture benign and malicious DoH traffic along with non-DoH traffic. In the first layer, the Non-DoH activity is generated by accessing different web servers. The DoH traffic has been collected using Several DNS tunnelling tools have been used such as DNSCat2, Iodine, and dns2tcp [12], [14]. In layer 2 data collection, Malicious-DoH traffic is generated using the above-mentioned tunnelling tools, where these tools sent TLS-encrypted HTTPS data in DNS queries to DoH servers (Adguard, Cloudflare, Google, Quad9). Fig. 1 shows the technical framework of DNS tunnels for generating Malicious DoH traffic. To capture Benign-DoH traffic, Several web browsers such as Chrome, Firefox, and safari have been used to generate Benign-DoH in the same mechanism as in scenario Non-DoH.

A. Exploratory Feature Analysis

The features of this dataset can be divided into multiple broad categories. Flow Statistics is one of the categories containing features such as the duration of the flow and the number of packets sent or received in that flow. The category Flow Bytes contains features describing the number of total bytes sent and/or received. Furthermore, there is a Packet Length category containing statistical features about the packet

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TIFS.2022.3183390,

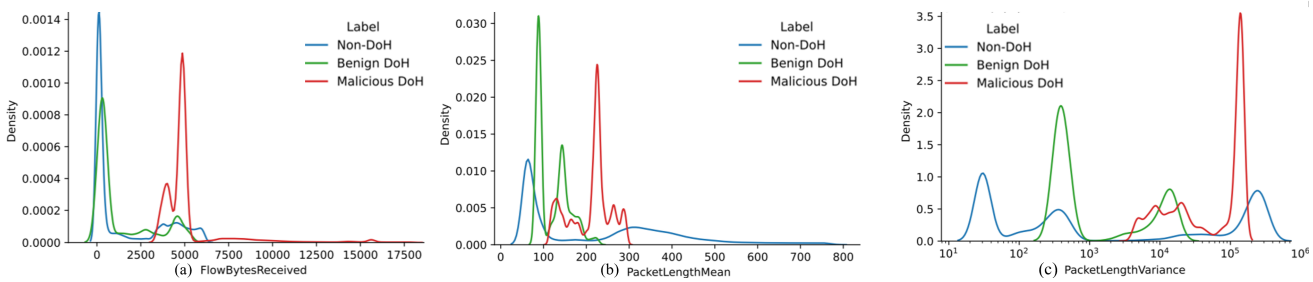


Fig. 2. Density plots displaying the difference in a number of features for the three types of traffic measurements available in the dataset. (a) the number of bytes sent or received for malicious DoH is clearly higher compared to non-DoH and benign-DoH as visible from the FlowBytesReceived feature distribution (b) The mean and (c) variance of incoming packets. DoH flows have more consistent packet lengths, resulting in a smaller variance compared to non-DoH shown by the narrow distribution plot in red.

lengths such as mean value or standard deviation. There are similar statistical features calculated for the Packet Time and Inter-Packet Delay categories.

To learn the differences in characteristics of DoH and non-DoH, thorough feature analysis is performed in this section. The value distribution for the feature is plotted using a Kernel Density Estimation (KDE) plot. Fig. 2 shows the class-wise density distribution for Flow Bytes Received, packet length mean and variance features. A KDE plot is similar to a histogram, however, the KDE plot shows the estimation of the probability density function of a variable instead of discrete bins. The duration of a network flow is a feature clearly distinguishing malicious DoH from non-DoH, with DoH flows having comparatively longer duration. The non-DoH web traffic network flows have a short duration since the whole web page is fetched in only a few seconds. We noticed some differences in the Flow byte measurements as well, the number of bytes sent or received for malicious DoH is clearly higher compared to non-DoH and benign-DoH (shown in the KDE plot in Fig. 2(a) and (b). Additionally, we have looked into the mean and variance of incoming packets (shown in the KDE plot in Fig. 2(b) and (c)). In general, DoH flows have more consistent packet lengths, resulting in a smaller variance compared to non-DoH shown by the narrow distribution plot in red in Fig. 2(c) . The outgoing packets showed similar properties. An interesting difference between benign and malicious DoH is that the variance for malicious DoH is always relatively high due to alternating small and larger packets. To be noted, we have done a thorough analysis of the 29 features available in CIRA-CIC-DoHBrw-2020 dataset [14], but we only presented some relatable insights from some features in this paper.

B. Dataset resampling and Train-test Split

To deal with the class imbalance in the training data, we have used a one-sided selection with the synthetic Minority Oversampling Technique (SMOTE) technique [15] while preparing our training data. The dataset has been split into training and testing sets of 90%, and 10% respectively. As can be seen from the system overview diagram in Fig. 3, we have created three balanced splits from the 90% training data to feed three independent sub-models. We have used three different splits of Non-DoH data while sharing the same malicious samples over various divisions. We applied SMOTE up-sampling of the benign group to avoid any evident bias from the majority groups available in the dataset. From the numbers available in Table I, the initial ratio of Non-DOH, Benign-DOH, and Malicious-DoH are 45:1:12 in the dataset, After balanced splitting and up-sampling of the benign group we had three splits of the training set with the ratio being 15:12:12 in each subset. With this sub-division, we had to use less amount of synthetic data from the minority group per sub-model. The improvements in performance using the proposed data split and sub-model development mainly were reflected by the three-fold reduction in training time because of the parallel nature of the sub-models. The training sets were further split into 10 folds to allow 10 fold cross-validation. To be noted, all of our experiments were performed on a Linux machine with an Intel Core i9 processor, 64 GB RAM and an NVIDIA RTX GPU.

C. Pre-processing: Scaling and Normalization

We performed a min-max normalization on the numerical feature vectors using equation (1).

$$x^{norm} = \frac{(x - min)}{(max - min)}. \tag{1}$$

We have utilized the MinMaxScaler implementation from the sklearn preprocessing library. The fit and transform method is used on the training set. Once column specific minimum and maximum values are measured by the method, the test data is transformed using the existing measurement from the training data. After this operation, all numeric feature values are ranged between 0 and 1.

D. Hyperparameter Tuning and Cross-Validation

For finding the optimal hyperparameters, resulting in the best classification, we used the GridsearchCV function that is used for the exhaustive search. In the search, different models are trained covering all (manually) pre-configured parameter values. Each model is tested after training and the search was done with 10- fold cross-validation so that the selected parameters are less susceptible to outliers.

IV. Model Implementation

In this study, we attempted to identify DoH traffic generated by various malicious DNS tunnel tools. The payload of DoH

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TIFS.2022.3183390,

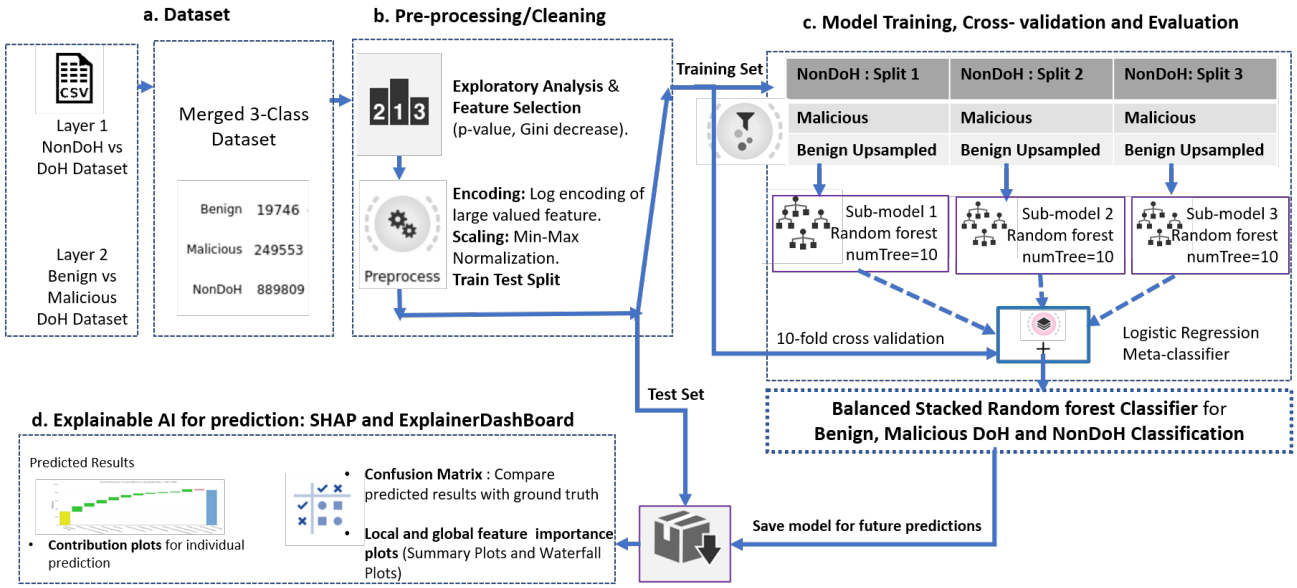


Fig. 3. Functional stages of the Balanced Stacked Random Forest Classifier for Benign, Malicious DoH and NonDoH Classification

Algorithm 1: Balanced Stacked Random Forest training algorithm

Input: Training dataset $X = \{x_1, x_2, ..., x_m\}$, Number of features 29, & Training Labels

Pre-processing layer;

Scale and normalize using equation(1); $X^{norm} = (x - min)/(max - min)$;

Balanced data sub-set creation from training data and application of SMOTE;

Classifier;

Initialize parameters at the supervised layer: Random forest, variable tree depth, 10 tree estimators;

Sub-model training for speeding up the calculation;

Calculate the labels for each sample x_n of the training dataset X;

Perform ten-fold cross-validation supervised manner to for model optimization and parameter tuning;

Stacked Ensemble layer;

Stacking sub-classification model to provide ensemble output for X;

XAI Output generation;

Application of TreeExplainer from SHAP library to generate contribution plot and contribution table; end;

Output: Class labels, Probability, Contribution plot, Contribution table

traffic is encrypted; thus, its content cannot be accessed. Therefore, we have used the statistical features of the packets to analyze the traffic in detail. For the detection and classification task, we employed two main functional stages in our proposed model. A balanced-training layer with multiple sub-models and a stacked classifier for classification based on DNS over HTTPS intrusion features. We describe our intuition for using these components in the system development in the coming subsections. To be noted, we have also trained Decision Tree, Random Forest, and Xgboost Classifiers for the purpose of mode performance comparison before settling to our final Balanced and Stacked Random Forest Classifiers.

A. Base-Classifiers: Random Forests

For our model implementation, we are using a popular ensemble classifier called the Random Forest [16] which operates by constructing multiple decision tree models at the training time. It is one of the most accurate supervised learning methods in recent times. Each decision tree in a Random Forest represents one class of observations that are being considered. Decision trees are constructed during the learning process with the training data. Random Forests mainly rely upon two parameters to control their growth: *numTrees*, the number of decision trees to be built and *numFeatures*, the number of random subset of features to assess at each tree node[17]. In our design, numTrees = 10 and numFeatures = 28. Each of the 10 decision trees is constructed in a top-down manner starting with a root node by selecting a set of N observations of size n at random with replacement from the training dataset and selecting the most significant features of these samples as the tree nodes. At each node a, the m number of features is selected at random from 28 features to grow the tree and the most significant feature that provides the best binary split on that node is selected among all according to an objective function. Feature significance is generally estimated using the Gini index[18]. To classify a new sample, the features values of the samples are tested with each of the decision trees present in the random forest. Each tree gives a classification score or "vote" and the class with the most votes is selected as the class to which the sample belongs. We have used the RandomForestClassifier from the sklearn.ensemble module in python for training the models [19].

B. Balanced and Stacked Classifier for Higher Predictive Performance

The simplest form of stacking can be described as an ensemble learning technique where the predictions of multiple classifiers are used as new features to train a meta-classifier [20]. The functional stages of the proposed algorithm is

outlined in Fig. 3. The workflow is demonstrating the stacking scheme we used to train and implement our multi-class traffic detection model. The meta-classifier of our choice is a logistic regression model. All the sub-models in this diagram are Random forest models with numTrees=10, and has a maximum branch depth of 5 in the individual decision trees to keep the computation fast enough during the prediction stage. For the implementation, we have used the StackingClassifier from the mlxtend.classifier module [21]. A pseudo-code of the algorithm development process is summarized in algorithm I.

V. Model Evaluation

Once the model development was done, we evaluated how well the model is performing on test data from various classes. For that, we have reported the scores such as accuracy, precision, recall, Area Under the ROC Curve (AUC), and F1-scores since these are directly comparable with other studies. Additionally, confusion matrices are reported to give insights into the strong and weak points of the classifiers, it shows which classes are often misclassified. A definition of the evaluation matrices is provided in the next sub-section.

A. Model evaluation matrices

If True Positive (T_P) is the number of attacks classified rightly as attack; True Negative (T_N) is the number of normal events rightly classified normal; False Positive (F_P) is the number of normal events misclassified as attacks and False Negative (F_N) is the number of attacks misclassified as normal, we can define accuracy, recall, precision and F1 values of a model using the following equations.

- Accuracy: It is an indicator of the total number of correct predictions provided by the model and defined as follows:

$$\text{Accuracy} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N}. \tag{2}$$

- Recall, precision and F1 Score: Three of the most commonly used performance measures with F1 score being the harmonic mean of recall and precision measures are defined as follows:

$$\text{Recall or True positive rate} = \frac{T_P}{T_P + F_N}. \tag{3}$$

$$\text{Precision} = \frac{T_P}{T_P + F_P}. \tag{4}$$

$$\text{F1 Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision+Recall}} \tag{5}$$

B. Confusion Matrix

We presented the confusion matrix plot in Fig. 4, for our model when evaluated with the test data set. The columns correspond to the predicted class and the rows correspond to the true class (Actual Class). The diagonal cells in the confusion matrix correspond to observations that are correctly classified (T_P and T_N's). The off-diagonal cells correspond to incorrectly classified observations (F_P and F_N's). Both the number of observations and the percentage of the total number of observations are shown side by side. For the proposed balanced stacked random forest classifier, class-wise model performance for train and test set is shown in confusion matrices (a) and (b) respectively. As can be seen in Fig. 4 (b), the proposed stacked random forest was able to detect 24949 malicious out of 24955, with only six misclassified instances where the model predicted those as NonDoH. The major source of misclassification was observed in the model for benign instances classified as NonDoH, these errors are caused due to the test instance benign similar in nature to NonDoH in terms of models top predictive features such as duration, packet length etc, However, these errors are less damaging to the system because of their benign nature.

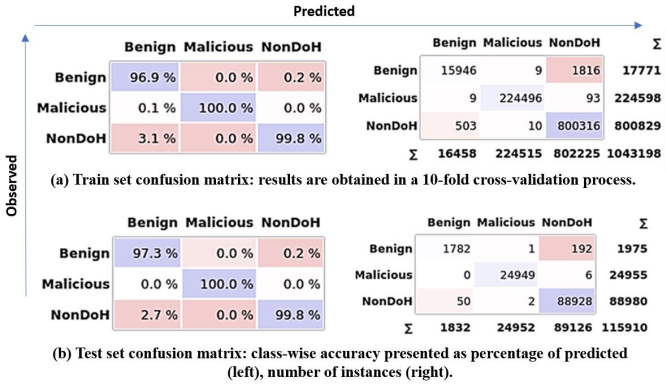


Fig. 4. Class-wise model performance for train and test set is shown in confusion matrices (a) and (b) respectively. For each case, class-wise accuracy is shown as a percentage of predicted on the left and a count of instances was shown on the right.

C. Performance comparison

Along with our final Balanced Stacked Random Forest Classifier, we have additionally trained a Decision Tree, a Random Forest and an Xgboost (Gradient Boosting) Classifier for comparison purposes. As shown in Table II, the proposed classifier results are better than the generic ensemble learning framework such as the Gradient Boosting and The RF classifier with SMOTE balancing. Compared to the other classifiers, the created ensemble framework misclassified a few samples from NonDoH and benign class but, there is only six wrong classifications in malicious class. And a very low misclassification for the malicious class would be desirable in this scenario. In the table, we have also compared the AUC score for the various classifiers we have developed for the task.

From the results shown in Fig. 4 and Table II, we noticed our system can identify malicious DNS traffic with more than 99% accuracy. The model can distinguish DoH traffic from normal HTTPS network traffic 99.9% of the time and the class-wise accuracy of Benign, Malicious and Non-DoH traffic on the test set was found to be 97.3%, 99.99% and 99.8% respectively. In Table II, we have reported the AUC, F1-score, precision and recall value from the models we have developed along with some comparable results from the literature on the same dataset. There were several machine learning

TABLE II
MODEL ACCURACY COMPARISON IN TERMS OF AREA UNDER THE CURVE (AUC), ACCURACY, PRECISION, RECALL AND F1-SCORE FOR NONDOH, BENIGN AND MALICIOUS DOH TRAFFIC CLASSIFICATION

Model	AUC	Accuracy	F1	Precision	Recall
Models we developed:					
Decision Tree (Tree Depth=10, SMOTE balanced)	0.8617	0.9770	0.8197	0.9658	0.7120
Gradient Boosting(XGB, SMOTE balanced)	0.9986	0.9927	0.9843	0.9956	0.9732
Random Forest (number of Trees=10, SMOTE balanced)	0.9999	0.9998	0.9987	0.9989	0.9985
Proposed model (Balanced Stacked random forest)	0.9999	0.9998	0.9991	0.9991	0.9992
Comparison with literature:					
Decision Tree[10]	0.998	0.998	0.998	0.998	0.999
Gradient Boosting(XGB)[10]	1	0.999	1	1	1
Random Forest [10]	1	0.998	0.997	0.999	0.998
Decision Tree[12]	–	0.999715	–	–	–
Random Forest [12]	–	0.999802	–	–	–
Decision Tree[22]	–	0.993	–	0.992	0.993
Gradient Boosting(XGB) [22]	–	0.951	–	0.957	0.951
Random Forest [22]	–	99.5	–	99.4	99.6

methods presented in [10]–[12], [22] using various tree-based algorithms, ensemble classifiers, other neural networks and deep learning algorithms on the CIRA-CIC-DoHBrw-2020 dataset. Though the experimental method in this literature is not directly comparable, we have listed the results from similar methods reported in these for comparison purposes in the lower half of Table II. Banadaki et al. [10] reported several ML algorithms with very high accuracy, precision and recall scores. However, this research did not include any detail on the preprocessing stages, or model parameters to repeat the experiments. They also used 4000 observations in the test set, which is very small compared to the dataset itself and the set may not have enough variation and looks overfitted. Ahakonye et al.[22] proposed a time-efficient Ensemble Learning (EL) model reporting an accuracy of 99.5% with reduced processing time. Ramakrishnan et. al.[11] proposed a NN (Neural Network) based IDS that can quickly respond to attacks by analyzing low-level network details. The proposed scheme is quite limited in terms of the number of features used and low accuracy where it averagely reports 90% accuracy. Jafar et al.[12] explored eight ML methods including Logistic Regression, Stochastic Gradient Descent, Decision tree and Random forest to name a few. This research reported accuracy value only along with the computational time required to train the model. We have included the results from the tree-based methods in our comparison table which reported a 99.9% accuracy, but there is again no detail on class-wise accuracy and other evaluation matrices. Compared to this result, our proposed Balanced and stacked random forest model was able to distinguish DoH traffic from normal HTTPS network traffic 99.9% of the time and the class-wise accuracy of Benign, Malicious and Non-DoH traffic on the test set was found to be 97.3%, 99.9% and 99.8% respectively. The improvements in performance using the proposed data split and sub-model development mainly was reflected by the three-fold reduction in training time because of the parallel nature of the sub-models. We have also obtained slightly improved precision and recall performance from the proposed implementation when compared to a generic random forest model trained without a balanced sub-division. Our proposed balanced and stacked random forest achieved slightly higher precision (99.91%), recall(99.92%) and F1 score (99.91%) than the other candidate models we developed, which is desirable for the task at hand. For the purpose of comparison, the Xgboost model had a precision value of 99.56%), a recall value of 97.322% and an F1-score of 98.43% in this scenario.

VI. EXPLAINING THE DECISIONS USING XAI

In this section, we highlighted our use of XAI methods to visualize the decision-making process of our proposed model. We used the methods available from the SHAP (SHapley Additive exPlanations) library to look into the model's decision-making process, expected impact from various features and potential biases. It helped us characterize model accuracy, transparency and outcomes to be validated by a human user.

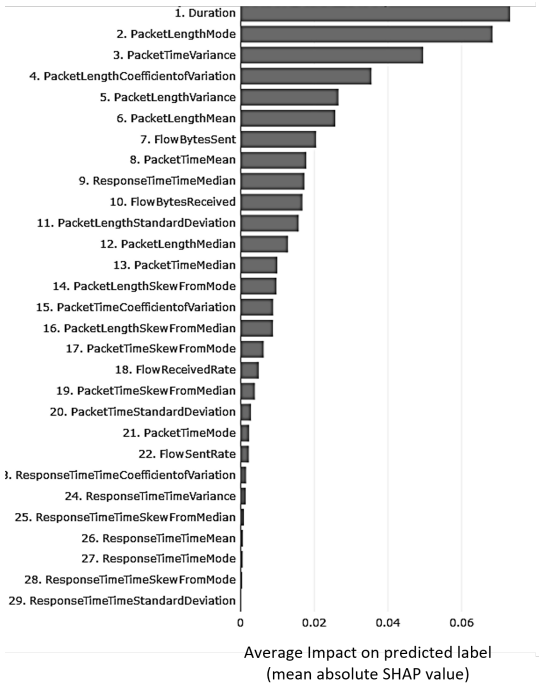
A. Feature importance plots

As DoH can be used for benign and malicious purposes, so if DoH is detected, the analysis of the features that are helping to detect DoH traffic would be highly beneficial. For our use case, Fig. 5 is highlighting a SHAP summary plot from the proposed model that is giving us the global feature importance values obtained from the training data. On the X-axis of the summary plot, we have the average impact (mean absolute SHAP values) of a particular feature on the decision making of a particular sample. SHAP values show how much a given feature changed our prediction (compared to if we made that prediction at some baseline value of that feature). On Y-axis the features are presented according to their importance globally from the entire training set. From our visualization, we found out that the duration of a network flow and the packet length related features were the features that helped the model heavily to distinguish the malicious DoH from Non-DoH. The packet length related features were found to be most powerful in separating the benign DoH traffic from the dataset.

B. Dependence plots and Interaction plots

It is also possible to create local summary plots displaying positive SHAP-values indicative of a feature supporting the decision confidence. Negative SHAP-values are indicative of the feature negatively impacting the decision confidence.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TIFS.2022.3183390,



Category	Feature
Flow Statistics	Duration
Flow Bytes	Number of flow bytes sent
	Number of flow bytes received
	Rate of flow bytes sent
	Rate of flow bytes received
Packet Length	Mean Packet Length
	Median Packet Length
	Mode Packet Length
	Variance of Packet Length
	Standard Deviation of Packet Length
	Coefficient of Variation of Packet Length
	Skew from median Packet Length
	Skew from mode Packet Length
Packet Time	Mean Packet Time
	Median Packet Time
	Mode Packet Time
	Variance of Packet Time
	Standard Deviation of Packet Time
	Coefficient of Variation of Packet Time
	Skew from median Packet Time
	Skew from mode Packet Time
Inter-Packet delay	Mean Request/response time difference
	Median Request/response time difference
	Mode Request/response time difference
	Variance of Request/response time difference
	Standard Deviation of Request/response time difference
	Coefficient of Variation of Request/response time difference
	Skew from median Request/response time difference
	Skew from mode Request/response time difference

Fig. 5. SHAP summary plot from the model that is giving us a birds-eye view of feature importance. From our analysis, we found the duration of a DoH traffic is the most important predictor of whether the traffic is malicious or not, followed by some features related to packet length and variance in packet time. Our list of features included computed flow statistics, flow bytes, packet length, packet time and Inter-packet delay features as shown on right.

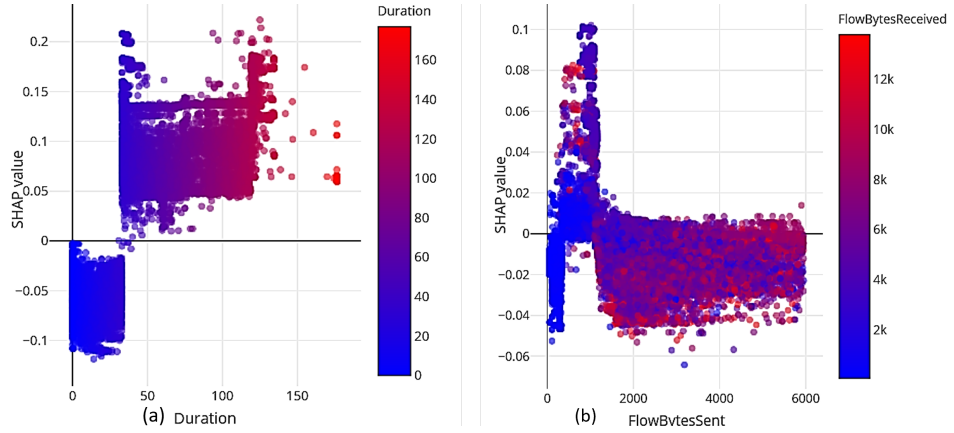


Fig. 6. (a) SHAP dependence plot for Duration as a feature. On y-axis we have the SHAP Values for each observation. This plot holds the duration of all the observations of the test set to monitor the impact of duration in the model's classification. Each dot represents a row of the data. We can infer from this diagram that the model is using a duration threshold above 40 seconds to determine malicious DoH traffic reflected by the positive SHAP Values above this range. (b) Interaction plot of FlowBytesSent and FlowBytesReceived. An interesting grouping is revealed on the upper left cluster in this plot when bytes received sent in some instances is bigger than bytes originally sent indicating the suspicious or probable malicious nature of these grouped instances.

SHAP dependence plots provide useful insight if we want to delve into the impact of a single feature in terms of the samples the model has processed.

In Fig. 6, we have plotted the Duration and FlowByteSent feature for all the samples in the test set, where each dot represents a row of the data. The horizontal location is the actual value from the dataset, and the vertical location shows the SHAP impact value for that prediction. Higher the SHAP value, the bigger the impact of the feature for one observation in its decision making. In these diagrams, the malicious traffic is indicated with positive SHAP values, benign and nonDoH traffic represented using negative shap values for classification. In Fig. 6(a) we plotted the duration of all the observations of the test set to monitor the impact of duration in the model's classification. Most of the instances, that were classified as malicious DoH traffic by the model is having a duration above 40 seconds. In Fig. 6(b) We are showing an interaction plot of FlowBytesSent with FlowBytesReceived. The interesting grouping is revealed on the upper left cluster in this plot when bytes received sent in some instances is bigger than bytes originally sent indicating the suspicious or probable malicious nature of the grouped instance.

C. Explaining a malicious and a non-DoH test packet

We have deployed the model on an interactive Explainer Dashboard to test the model functionality in a transparent manner. Fig. 7 and Fig. 8 included a visualization of a contribution table and a contribution plot. These help to assess

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TIFS.2022.3183390,

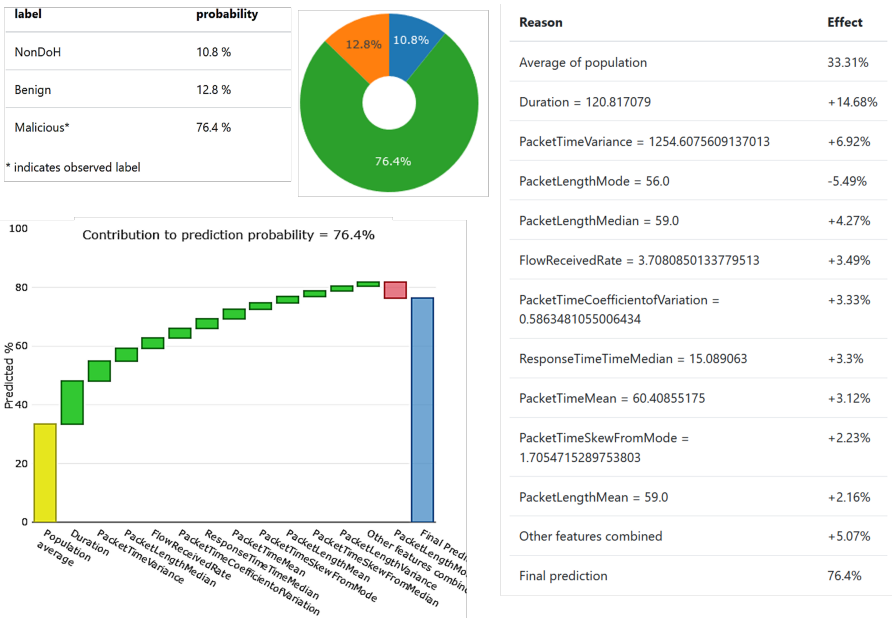


Fig. 7. Explaining a malicious test packet. On the top left, the dashboard provides a prediction probability and a pie chart shows the percentages. The label with the asterisk sign is the label the model outputs as the decision. On the right, we have a contribution table with the value of each feature of the data sample being processed by the model and their effect value or positive-negative contribution in models decision making. On the bottom left, we have the feature contribution plot as a waterfall plot, green bars displaying positive contributors and the red bars displaying negative contributors in the decision making process.

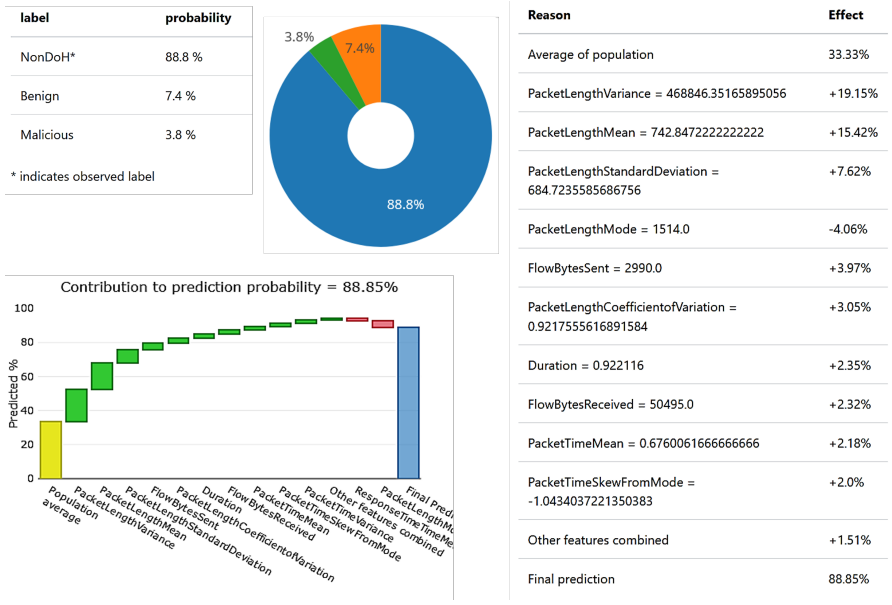


Fig. 8. Explaining a Non-DoH test packet. For this decision, we can see the model had an 86.1% confidence score for Non-DOH based on the feature averages and the impact the model has seen for various samples during the training phase.

the contribution of various feature values for an exemplar malicious DOH traffic and a Non-DOH traffic sample from the dataset. We have used 29 features to classify DoH traffic in our model and it is possible to calculate how much each feature contributed to generating the confidence value of a certain decision with the help of the dashboard functionalities we have put together for this. We have added a video demonstration of the dashboard as supplementary material with this submission.

In Fig. 7, we are demonstrating a detailed explanation of a malicious test packet by the model. The deployed model dashboard provides a prediction probability (table and pie chart on the top left), which is 76.4% for this Malicious instance. The contribution plot below provides a further breakdown of which feature contributed positively or negatively in models decision making for this particular instance. For example, this test flow has a duration of 120.81 seconds, which is above the threshold for NonDoH and benign traffic contributed positively (+14.68%) to models decision for classifying this instance as malicious. The next highest contribution was coming from a high packet time variance measurement for this case. However, the packet length mode from the mode for this transaction was low compared to usual malicious traffic, which affected the model confidence value negatively by 5.49%.

In Fig. 8, we are showing a similar analysis of a Non-DOH

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TIFS.2022.3183390,

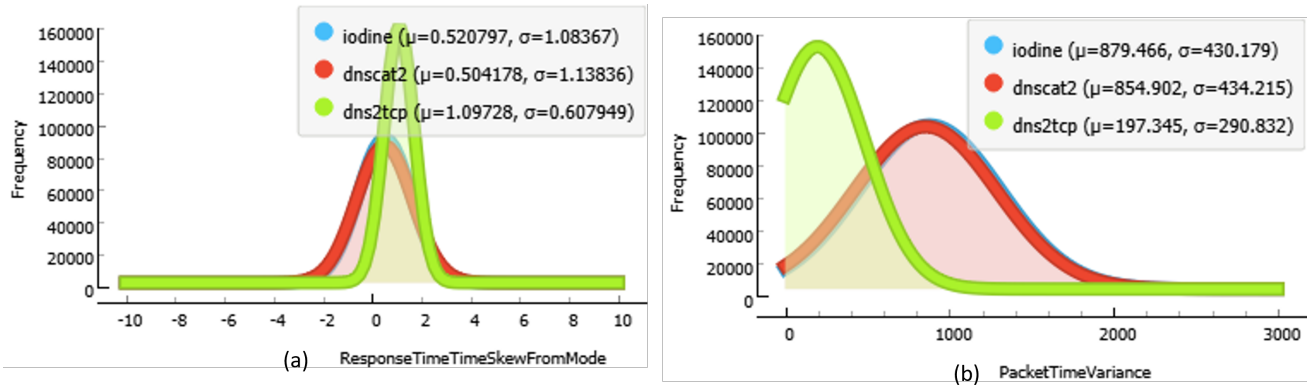


Fig. 9. Identifying the distinction between from various malicious DNS tunnel tools, such as dns2tcp, DNSCat2, and Iodine for (a) ResponseTime-TimeSkewFromMode and (b) PcketTimeVariance feature.

test packet. Notice the value difference in Duration in both the decisions and their contribution differences. Additionally, for this decision, we can see the model had a 3.8% similarity of malicious traffic in some of the features. It is also possible to identify the features that impacted the model's decision that way. In this instance, the PacketLengthMode value of this traffic was leaning towards a higher value of the malicious group than to the Non-DOH group of training samples the model experienced during training.

D. Identifying the source of attack

The malicious-DoH traffic generated in this research dataset is coming from various malicious DNS tunnel tools, such as dns2tcp, dnscat2, and iodine, which can create tunnels of encrypted data. Therefore, DNS queries are sent via these tools using TLS-encrypted HTTPS requests to special DoH servers. In our quest of whether it is possible to sub-classify various sources of attack, we have included the malicious source tools with our malicious instances and presented some comparisons based on the features we are using in our modelling. As can be seen from the distribution plots in Fig. 9, we noticed various feature distributions for iodine and dnscat2 are similar in nature. In Fig. 9 (a) and (b), we are reporting ResponseTimeSkewFromMode and PcketTimeVariance feature along with mean and standard deviation for each source of malicious DNS traffic. The dns2tcp tools have a seemingly lower standard deviation for both these parameters. Because of this, our system identified the source of malicious traffic with 99.2% accuracy for dns2tcp, the accuracy for iodine and dnscat2 being 92.9% and 91.3 % respectively.

VII. Conclusions

DoH technology has been developed to provide security and privacy for Internet users by encrypting the DNS traffic. However, over the past few years, DNS remained a prime target for hackers as it enables them to gain first entry into networks and gain access to data for exfiltration due to network traffic generated by malware and malicious tools. Although many studies on encrypted network traffic classification and DNS tunnel detection have been reported before, as DoH is a new protocol we need new set of intrusion detection tools. In this paper, we reported an explainable AI model dashboard that can detect malicious DoH traffic accurately. To prove that our system can identify malicious DNS tunnel tools and evaluate the performance, we have used the publicly available CIRA-CIC-DoHBrw-2020 dataset. Our proposed model can distinguish DoH traffic from normal HTTPS network traffic 99.9% of the time and the class-wise accuracy of Benign, Malicious and Non-DoH traffic on the test set was found to be 97.3%, 99.9% and 99.8% respectively. We have also reported the AUC, F1-score, precision and recall value from the model. In comparison to state-of-the-art ensemble models such as gradient boosting and generic random forest, our proposed balanced and stacked random forest achieved slightly higher precision (99.91%), recall(99.92%) and F1 score (99.91%) which is desirable for the task at hand. Additionally, With the help of the SHAP values, we have also highlighted the feature contributions for the underlying classification decision by the model. We have also discussed the conditions under which high classification accuracy can be achieved by using these features. In summary, the proposed method provides an accurate solution to detect and classify the DNS over HTTPS attacks.

Our future research will apply the explainable DoH detection methods for deep neural network-based solutions [23]. Currently, DoH traffic is only distinguished from browser traffic. However, there might be HTTPS traffic created by more applications than browsers only, with characteristics more similar to DoH traffic. Other types of malicious use of DoH can also be an interesting topic for exploration. Botnets often use fast domain fluxing or Domain Generating Algorithms(DGA). DGAs in botnets might abuse DoH [7], [24]. Future extensions of the current work can aim to distinguish DGA related DoH traffic from other HTTPS traffic.

Supplementary Material

This paper has supplementary material available at https://github.com/TZebin/DoH-Attack-explainer. The material includes a video recording of the explainable AI dashboard functionalities and code implementation for the model. Contact t.zebin@uea.ac.uk for further questions about this work.

REFERENCES

[1] R. Fouchereau, *2021 Global DNS Threat Report*, 2021. [Online]. Available: https://www.efficientip.com/wp-content/uploads/2021/06/2021-IDC-DNS-Threat-Report-Infobrief-final_compressed.pdf.

[2] 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*, 2020, pp. 1–8.

[3] Y. Wang, A. Zhou, S. Liao, *et al.*, "A comprehensive survey on dns tunnel detection," *Computer Networks*, p. 108 322, 2021.

[4] M. Aiello, M. Mongelli, and G. Papaleo, "Basic classifiers for dns tunneling detection," in *2013 IEEE Symposium on Computers and Communications (ISCC)*, IEEE, 2013, pp. 000 880–000 885.

[5] H. Zhao, Z. Chang, G. Bao, *et al.*, "Malicious domain names detection algorithm based on n-gram," *Journal of Computer Networks and Communications*, vol. 2019, 2019.

[6] L. A. Trejo, V. Ferman, M. A. Medina-Pérez, *et al.*, "Dns-advp: A machine learning anomaly detection and visual platform to protect top-level domain name servers against ddos attacks," *IEEE Access*, vol. 7, pp. 116 358–116 369, 2019.

[7] L. Vries, "Detection of DoH tunnelling: Comparing supervised with unsupervised learning," M.S. thesis, University of Twente, 2021.

[8] *Cira-cic-dohbrw-2020 dataset*, 2020. [Online]. Available: https://www.unb.ca/cic/datasets/dohbrw-2020.html.

[9] R. Preston, "Dns tunneling detection with supervised learning," in *2019 IEEE International Symposium on Technologies for Homeland Security (HST)*, IEEE, 2019, pp. 1–6.

[10] Y. 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, pp. 46–55, 2020.

[11] S. Ramakrishnan and A. Senthil Rajan, "Network attack detection with qnnbadt in minimal response times using minimized features," in *Computer Networks and Inventive Communication Technologies*, Springer, 2022, pp. 563–579.

[12] M. T. Jafar, M. Al-Fawa'reh, Z. Al-Hrahsheh, *et al.*, "Analysis and investigation of malicious dns queries using CIRA-CIC-DoHBrw-2020 dataset," *Manchester Journal of Artificial Intelligence and Applied Sciences*, vol. 2, pp. 65–70, 2021.

[13] S. M. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st international conference on neural information processing systems*, 2017, pp. 4768–4777.

[14] M. Montazeri Shatoori, L. Davidson, G. Kaur, *et al.*, "Detection of doh tunnels using time-series classification of encrypted traffic," in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing*, 2020, pp. 63–70.

[15] I. H. Witten, E. Frank, M. A. Hall, *et al.*, *Data Mining: Practical machine learning tools and techniques*. Burlington: Morgan Kaufmann, 2016.

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

[17] D. Devetyarov and I. Nouretdinov, "Prediction with confidence based on a random forest classifier," in *IFIP International Conference on Artificial Intelligence Applications and Innovations*, Springer, 2010, pp. 37–44.

[18] T. Ogwang, "A convenient method of decomposing the gini index by population subgroups," *Journal of Official Statistics*, vol. 30, no. 1, p. 91, 2014.

[19] M. Feurer, A. Klein, K. Eggensperger, *et al.*, "Auto-sklearn: Efficient and robust automated machine learning," in *Automated Machine Learning*, Springer, Cham, 2019, pp. 113–134.

[20] N. Hatami and R. Ebrahimpour, "Combining multiple classifiers: Diversify with boosting and combining by stacking," *International Journal of Computer Science and Network Security*, vol. 7, no. 1, pp. 127–131, 2007.

[21] S. Raschka, "Mlxtend: Providing machine learning and data science utilities and extensions to python's scientific computing stack," *Journal of open source software*, vol. 3, no. 24, p. 638, 2018.

[22] L. A. C. Ahakonye, C. I. Nwakanma, S. O. Ajakwe, *et al.*, "Countering DNS vulnerability to attacks using ensemble learning," in *2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, IEEE, 2022, pp. 007–010.

[23] S. Rezvy, Y. Luo, M. Petridis, *et al.*, "An efficient deep learning model for intrusion classification and prediction in 5G and IoT networks," in *2019 53rd Annual Conference on information sciences and systems (CISS)*, IEEE, 2019, pp. 1–6.

[24] M. Behnke, N. Briner, D. Cullen, *et al.*, "Feature engineering and machine learning model comparison for malicious activity detection in the dns-over-https protocol," *IEEE Access*, vol. 9, pp. 129 902–129 916, 2021.

Dr. Tahmina Zebin is a Lecturer in the School of Computing Sciences at the University of East Anglia and is one of the academic leads at the On-device and Explainable AI Laboratory. Tahmina completed her PhD studies at the University of Manchester and has been the recipient of the Presidents Doctoral Scholarship (2013-2016) in Electrical and Electronic Engineering. Her research expertise includes Advanced Video and Signal Processing, Explainable and Inclusive AI, Human Activity Recognition, Risk Prediction modelling using various statistical machine learning and deep learning techniques.

Dr. Shahadate Rezvy is a Lecturer in Computer Science at York St. John University, UK. He completed his PhD in Wireless Communications from Middlesex University London, UK in 2016. He also holds an M.Sc in Telecommunications. Shahadate received his first degree in Computer Science and Engineering. His current research interests include Deep Learning, AI, Telecommunications and Network Security, Wireless Communication, IoT based Vehicular Communication Systems.

Dr. Yuan Luo is a Senior Lecturer in the Faculty of Science and Technology at Middlesex University, UK. He completed his PhD in School of Computing, Staffordshire University, UK, in 2002. He is instructor of Cisco CCNA, CCNP, and DevNet. His current research interests include network protocols and operations, network programmability and automation, network management, network security, cloud computing and visualization, machine learning and deep learning, AI, telecommunications, wireless communications.