sustainability

MDPI

Article

Interpretable Machine Learning Models for Malicious Domains Detection Using Explainable Artificial Intelligence (XAI)

Nida Aslam [1,*], Irfan Ullah Khan [2], Samiha Mirza [2], Alanoud AlOwayed [2], Fatima M. Anis [2], Reef M. Aljuaid [2] and Reham Baageel [2]

[1] SAUDI ARAMCO Cybersecurity Chair, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia
[2] Department of Computer Science, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam 31441, Saudi Arabia; iurab@iau.edu.sa (I.U.K.); 2180007084@iau.edu.sa (S.M.); 2180004886@iau.edu.sa (A.A.); 2180007105@iau.edu.sa (F.M.A.); 2180004320@iau.edu.sa (R.M.A.); 2180003525@iau.edu.sa (R.B.)
* Correspondence: naslam@iau.edu.sa

Abstract: With the expansion of the internet, a major threat has emerged involving the spread of malicious domains intended by attackers to perform illegal activities aiming to target governments, violating privacy of organizations, and even manipulating everyday users. Therefore, detecting these harmful domains is necessary to combat the growing network attacks. Machine Learning (ML) models have shown significant outcomes towards the detection of malicious domains. However, the "black box" nature of the complex ML models obstructs their wide-ranging acceptance in some of the fields. The emergence of Explainable Artificial Intelligence (XAI) has successfully incorporated the interpretability and explicability in the complex models. Furthermore, the post hoc XAI model has enabled the interpretability without affecting the performance of the models. This study aimed to propose an Explainable Artificial Intelligence (XAI) model to detect malicious domains on a recent dataset containing 45,000 samples of malicious and non-malicious domains. In the current study, initially several interpretable ML models, such as Decision Tree (DT) and Naïve Bayes (NB), and black box ensemble models, such as Random Forest (RF), Extreme Gradient Boosting (XGB), AdaBoost (AB), and Cat Boost (CB) algorithms, were implemented and found that XGB outperformed the other classifiers. Furthermore, the post hoc XAI global surrogate model (Shapley additive explanations) and local surrogate LIME were used to generate the explanation of the XGB prediction. Two sets of experiments were performed; initially the model was executed using a preprocessed dataset and later with selected features using the Sequential Forward Feature selection algorithm. The results demonstrate that ML algorithms were able to distinguish benign and malicious domains with overall accuracy ranging from 0.8479 to 0.9856. The ensemble classifier XGB achieved the highest result, with an AUC and accuracy of 0.9991 and 0.9856, respectively, before the feature selection algorithm, while there was an AUC of 0.999 and accuracy of 0.9818 after the feature selection algorithm. The proposed model outperformed the benchmark study.

Keywords: network security; malicious domains; machine learning; ensemble models; explainable artificial intelligence

Citation: Aslam, N.; Khan, I.U.; Mirza, S.; AlOwayed, A.; Anis, F.M.; Aljuaid, R.M.; Baageel, R. Interpretable Machine Learning Models for Malicious Domains Detection Using Explainable Artificial Intelligence (XAI). *Sustainability* **2022**, *14*, 7375. https://doi.org/10.3390/su14127375

Academic Editors: Amjad Ali, Farman Ali, Jin-Ghoo Choi and Muhammad Shafiq

Received: 23 April 2022
Accepted: 8 June 2022
Published: 16 June 2022

1. Introduction

Nowadays, the expansion of the internet has raised the complexity of the network globally, creating more gaps that attract hackers to perform illegal activities aiming to target governments, violate companies' privacy, or even manipulate individuals using phishing websites [1]. One of the major threats that has grabbed researchers' attention is the detection of malicious domains. Moreover, it is no longer possible to ignore the relationship between cyberattacks, information security, and their economic damage to organizations.

In a recent report 2021, the International Data Corporation (IDC) showed that 87% of the organizations suffered from DNS attacks [2].

The Domain Name System (DNS) has played a key role in the internet revolution. The primary goal was to ease the users' experience by translating the IP address of a website into a memorable name and vice versa. When a user searches on a particular website, the DNS will reach the nearest root name server seeking to answer the requested query. Then, the root server will reach other top-level domain (TLD) servers—such as .com, .org, and .edu, collecting parts of the domine name until it finally finds the IP address of the website from the authoritative DNS server. Although it may seem that there is no control over the DNS, local DNS servers enable the internet service provider (ISP) to monitor the network traffic. Furthermore, DNS raises two main security concerns: it does not provide any authentication mechanism between the nodes nor encryption mechanism over the packet [3]. The vulnerability from the structural perspective, DNS traffic tends to move towards a centric manner despite that the DNS is a decentralized system, leading the attackers to attack multiple entities at once easily. Additionally, the availability of DNS server information can threaten multiple companies, especially those who have poor DNS configuration. Such exposure could affect the DNS server operations and jeopardize the companies' internal systems and data [4]. Initially, methods such as capturing network traffic, web content, and URL inspection were used to detect malicious domains; however, the demand for Artificial Intelligence (AI) solutions is needed to automate the process of the detection. Initially, for the automated reasoning and decision making, either the rule-based or case-based approach was used. Later, the Machine Learning (ML) approach was introduced, and in the supervised ML, the main focus has been on the input and output data rather than how the output was made using the input data.

Firstly, basic ML models were introduced, such as Decision Tree (DT), Naïve Bayes (NB), linear regression, etc. These models contain good interpretability; however, they could not perform well for the highly complex problems. The emergence of the advance ML models that contain the complex structure and advance statistical methods such as ANN and SVM, as well as ensemble methods like GB, RF, etc., has dramatically improved the prediction and decision making. However, these models have introduced opaqueness in the reasoning. This opaqueness sometimes can restrict the use of these models in real-life applications. ML and DL models have contributed substantially for network and cyber security problems, such as the intrusion detection system (IDS), DNS classification, etc. [1]. Despite the successful classification using ML and DL, most of the models failed to respond to the subject, which is deducing why a certain DNS has been classified as malicious. Considering this, these systems are less reliable and trustworthy.

Therefore, the current research trend has moved towards the development of the models that, along with the good prediction performance, also demonstrate interpretability and comprehensibility. The emergence of Explainable Artificial Intelligence (XAI) allows for the conversion of the black-box models into glass-box models by generating explanations [5]. XAI-based models perform similar to the experts, and the models are more reliable and trustworthy [6,7]. XAI systems' objectives are not merely to enhance a task with respect to the competence and accuracy, but also to offer explanations about how the particular decision was made. XAI provides an insight into the complex logic innate in the model [8].

XAI has been applied in the cybersecurity system specifically in the detection of IDS. Recently, Mahbooba et al. [9] performed a study to propose an explainable IDS model to enhance trust in the model. The aim of the study was to develop a model similar to the human way of reasoning to identify the effect of the malicious data for IDS. The stakeholders have the right to know why the particular activity has been identified as malicious. Correspondingly, Le et al. [10] used the XAI for the IDS not only to generate an explanation, but also to help the cyber security professionals explore the validity and acceptability of the decisions. These explanations will help them to evaluate their judgement and decision. The study used the ensemble tree and shapley for the detection of IDS. Furthermore, Guo [11] discussed that the emergence of 6G has greatly enhanced the use of Internet of Things

(IOT), but, on the other hand, has posed a challenge for the security system. Various applications like the clinical decision system, remote surgery, and autonomous vehicle driving system require a transparent and trustworthy system [12,13]. Guo found that XAI enhanced the trust between AI and humans by integrating interpretability, transparency, and explicability.

Thus, it is essential to build interpretable ML models that will stimulate confidence in the knowledge they engender and provide to decision-makers. Therefore, in the current research, a post hoc XAI approach is used for the DNS classification.

In this paper, we propose a comprehensive comparison among Machine Learning (ML) models, including Decision Tree (DT), Naïve Bayes (NB), Random Forest (RF), Extreme Gradient Boosting (XGB), AdaBoost (AB), and Cat Boost (CB) algorithms, in order to classify the malicious and non-malicious domains using publicly available DNS datasets. Furthermore, XAI was used to generate the explanation at global and local level.

Our paper is structured as follows: Section 2 covers the previous related work, Section 3 covers the material and methods, Section 4 discusses the experiments and results. Furthermore, Section 5 contains the XAI, followed by further discussion in Section 6, and finally, Section 7 concludes the paper.

Contribution

The contribution of the proposed study is multi-fold. We aimed to develop a model with the enhanced performance in terms of accuracy, precision, recall, etc. without compromising the interpretability of the model. The main contributions of the proposed study are as follows:

- Comparative analysis of interpretable and non-interpretable ML models for the classification of DNS.
- Develop an interpretable model by using the post hoc XAI techniques to enhance the trust.
- The current study achieved better results with reduced number of features when compared with the benchmark in terms of all evaluation measures. The model can be used as an effective tool for the identification of the malicious DNS.

2. Related Studies

ML and DL have been investigated for detecting the malicious domains for establishing the protected cyber environment. Some of the recent studies are discussed below. Initially, the studies that used the ML are discussed, followed by the DL models and the studies using both ML and DL models.

Kidmose et al. [14] performed a study for classifying the malicious domain. They classified the domain features into three groups, one being generic and the remaining two being lexical features, specifically simple lexical features and advanced lexical features. Across many instances, the study found that utilizing lexical features with the other features can increase the detection efficiency of fraudulent websites using Random Forest (RF) with the precision of 0.98. Furthermore, Zhu and Zou et al. [15] discovered that, as the detection procedure progresses, the detection performance of a standard Support Vector Machine (SVM) model decreases; however, a revised SVM technique (F-SVM) proposed by the study is effective in maintaining a significant precision rate throughout the detection phase. Due to its high performance, it is well-suited for online detection. F-SVM uses the concept of reinforcement learning to tackle the issues of expensive model upgrading costs. They achieved 0.983 precision rate in the real-time detection of malicious domains. Besides, Almashhadani et al. [16] built a system called MaldomDetector that utilized easy to compute features in order to classify malicious domain names. The study combined several datasets from previous studies and used a randomness measuring algorithm to build the MaldomDetector. The results showed that the built model obtained high detection accuracy when compared with simple ML classifiers.

Moreover, Marques et al. [17] suggested a DNS firewall based on ML to enhance spontaneous detection of fraudulent domain requests, using a dataset containing 34 at-

tributes and 90,000 records derived from actual DNS traffic. The results demonstrate that ML techniques successfully categorized malicious and benign domains, with the CART algorithm having the best accuracy of 0.96 while using the Recursive Feature Elimination method.

Similarly, Palaniappan et al. [18] investigated an ongoing DNS approach by identifying it as safe or dangerous via the extraction of attributes as DNS, web, blacklisting, and lexical sources. A compact dataset of roughly 10,000 domain names was employed to train and test a Logistic Regression (LR) algorithm, which attained an accuracy of almost 0.60. Identically, Magalhaes et al. [19] provide a study on the effectiveness of using machine learning to detect fraudulent domain names taken from Alexa's top 1 million sites. The Decision Tree (DT) has the highest classifier performance with 0.92 accuracy in 11 s, while the Naive Bayes (NB) achieves the shortest period with 2.77 s, but the accuracy drops to 0.76.

Some studies used the CIRA-CIC-DoHBrw-2020 dataset [20,21] in order to identify malicious DNS over HTTP. These studies focused on implementing and comparing different ML and DL models to assess the dataset. For instance, Singh and Roy [22] used the DoH dataset to build and compare the performance of different Machine Learning (ML) classifiers, namely K-Nearest Neighbor (KNN), Naive Bayes (NB), Random Forest (RF), Logistic Regression (LR), and Gradient Boosting (GB). In the preprocessing phase, the authors removed the null attributes, selected the most efficient features, and performed a train-test split of 75–25 on the dataset. The results showed that GB and RF gave the best performance with precision, recall, and F-score values of 1.0 each. Additionally, Singh and Roy [23] applied further analysis on their previous work [22] by comparing RF with an ensemble learning framework of three classifiers: DT, LR, and KNN. Additionally, the DoHMeter was used as a tool for feature selection. The results showed that RF achieved a 1.0 accuracy on detecting both benign and malicious classes in contrast with the ensemble framework. In another study, Behnke et al. [24] used two feature selection methods, Chi-squared and Pearson's correlation, in order to choose the top 20 features from the dataset. After feature selection, 10-fold cross-validation was applied, and 10 ML classifiers were employed: RF, Decision Tree (DT), LightGBM, XGB, Extra Trees (ET), GB, Linear Discriminant Analysis (LDA), KNN, and AdaBoost (AB). The highest accuracy of 0.9978 was achieved by RF.

On the other hand, some studies used a DL approach in order to detect malicious domains. Hence, Akarsh et al. [25] focused on real-time malicious domains obtained by (DGA) using the LSTM architecture. The study performed both binary and multi-class classification using two locally generated datasets. The binary classification gave the best accuracy of 0.9871 and multi-class gave an accuracy of 0.683. In another study, Chen et al. [26] proposed an LSTM model that incorporates the attention mechanism, where the focus is more on important substrings in domains in order to improve the expression of domains. Using a real-life dataset, the study successfully achieved a reduced false alarm rate of 1.29% and false negative rate of 0.76%.

Likewise, Bharathi and Bhuvana [27] applied DL methods to recognize fraudulent domain names by extracting textual characteristics from domain names and passing them to LSTM and bidirectional LSTM. The results obtained performed best for binary classification than multiclass classification when tested on a publicly shared dataset. The recall for LSTM and bidirectional LSTM is 0.956 and 0.97, respectively. In addition, Ma et al. [28] presented a hybrid malicious domain detecting model based on Doc2vec, LSTM, and RNN for intense feature extraction and enhanced resilience. The composite result is linearly analyzed, and the probability distribution is determined by applying the Softmax classifier. When the hybrid model was compared to traditional domain identification algorithms, it came out on top with a 0.9781 accuracy. Amaizu et al. [24] aimed at a hybrid network model that uses a CNN and an LSTM to classify net traffic as benign, malicious, or non-DoH. The model was trained for a total of 28 epochs and observed the outcomes. According to simulation findings, the proposed technique has a 0.99 accuracy in distinguishing between benign, malicious, and non-DoH classes.

Additionally, Vinayakumar et al. [29] obtained the data for the study from local DNS records, and investigated the performance of several DL algorithms such Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and other classic ML models. Among DL approaches, LSTM has performed the best in identifying malicious DNS requests. However, Xu et al. [30] merged n-gram with CNN, and they suggested a new n-gram integrated character-based domain classification (n-CBDC) model. It simply needs the domain name alone and can autonomously evaluate the chance that the domain name was produced by domain generation algorithms (DGAs) without the assistance of manually obtained features or DNS context data. This model worked well in finding domain generation algorithm (DGA) domain addresses that were both comprehensible and centered on a wordlist. Conversely, Shi et al. [31] used Extreme Learning Machine (ELM) to detect malicious domain names using their own generated dataset containing around 50,000 samples of malware and benign domains combined. The ELM model obtained a high detection accuracy of 0.96.

Despite the fact that there have been various research studies on detecting malicious domains, evaluating and identifying emerging fraudulent domains in a reasonable timeframe is an important security contingency plan. However, there is a need for further improvement and investigation. Most of the previous studies have proposed models that have produced significant outcomes but suffer from the black-box problem. Those models do not allow researchers to find the reason why the prediction was made. For critical problems where the misclassification cost is high, there is a need for finding why the particular prediction was made. Therefore, there is a need to investigate the techniques that achieve significant outcomes in terms of accuracy, precision, recall, etc., as well as to have better interpretability. Post hoc XAI has integrated the interpretability in the complex models without compromising the performance of the models [32]. The post hoc explainable AI approach is applied after training the model; therefore, it will not affect the performance of the model. The XAI enables to incorporate the interpretability in the opaque models like SVM, ANN, ensemble-based models, and DL [33]. Therefore, in the current study we used post hoc XAI along with the ML models to produce models with a significant outcome and better interpretability.

3. Materials and Methods

This research aims to examine and classify domains as malicious or non-malicious using a recent DNS dataset. Figure 1 contains the conceptual framework of the proposed interpretable ML models for malicious domain detection. Initially ML models are implemented, then XAI techniques are employed to develop the interpretable models. Individually all the models are trained and tested using preprocessed features and selected features using sequential forward feature selector. The best performing model will be used to generate the explanations using XAI.

3.1. Dataset Description

The dataset used in this study was released by Marques et al. [34] in 2021 to classify the data sample as malicious or non-malicious. It was created from DNS logs, where the non-malicious domain were acquired from Rapid7 Labs [35] and the malicious domains from SANS Internet Storm Center (SANS) public list [36]. The dataset is available on Mendeley data repository [37]. Since it contains equal number of malicious and non-malicious domains, 45,000 samples for each, it does not suffer from any class imbalance. Further, it consists of 32 features of different datatype, i.e., text, Boolean, multi-class categorical, and numeric. Table 1 contains the description of the dataset. The exploratory analysis of the dataset attributes is represented in terms of mode percentage for the binary and three class attributes, and for the numeric attribute, the mean (μ) and standard deviation (σ) are used. However, for the multi-class attributes, only the number of categories is mentioned.

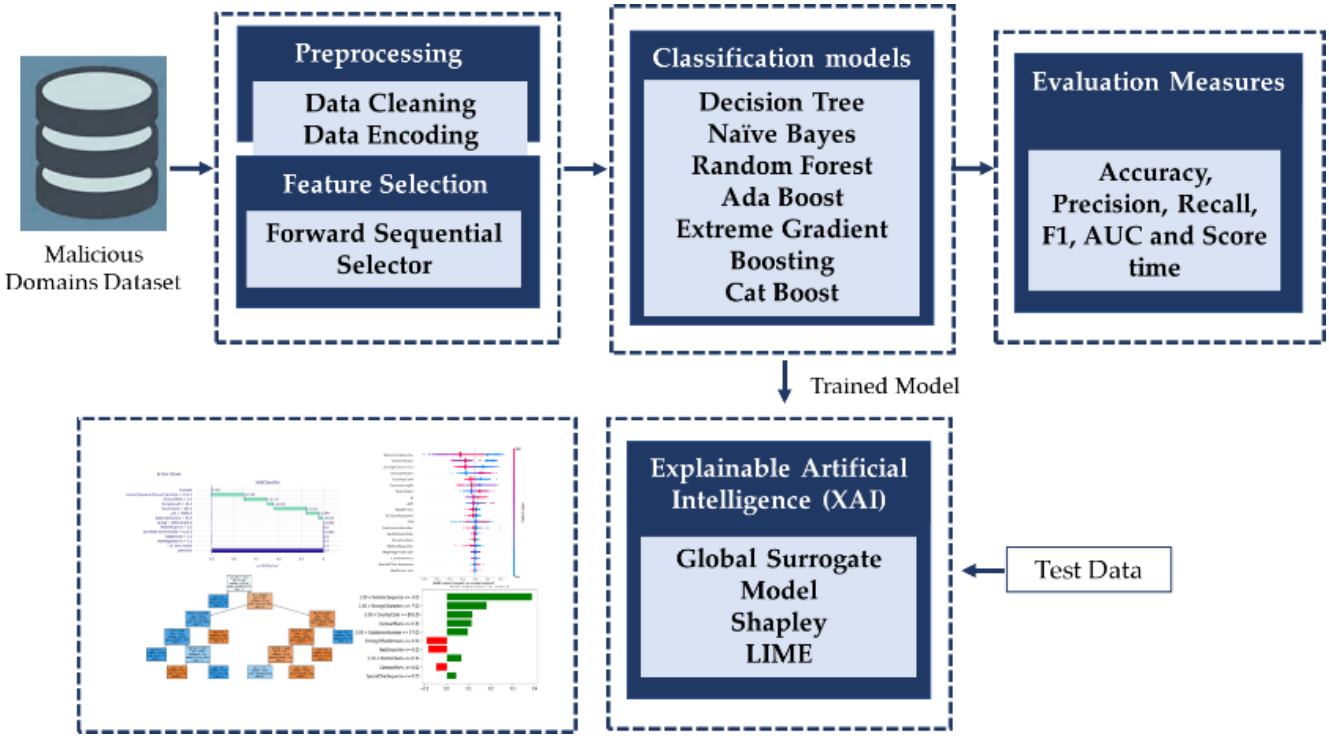

Figure 1. Conceptual framework of the proposed interpretable model for malicious domains detection.

Table 1. Description of the attributes in the Domain Name Service dataset.

Feature	Datatype	Description	Mode	Mean (μ) ± Std (σ)
Domain	Text	Encoded undisclosed DNS identifier		
Ip		Internet Protocol Address		
RegisteredOrg		Name of the organization registered the domain in WHOIS		
TLD		Top-Level Domain		
MXDnsResponse	Boolean	Mail Exchange request response from DNS	True (25.05%), False (74.96%)	
TXTDnsResponse		Text File request response from DNS	True (50.49%), False (49.51%)	
HasSPFInfo		Sender policy framework	True (49.13%), False (50.87%)	
HasDkimInfo		Domain keys identified emails	True (0.024%), False (99.9%)	
HasDmarcInfo		Domain-based message authentication	True (1.99%), False (98.01%)	
IpReputation		IP is block listed or not	True (4.51%), False (95.49%)	
DomainReputation		Domain name blocklist or not	True (0.12%), False (99.88%)	
DomainInAlexaDB		Domain name registered in Alexa Database	True (97.63%), False (2.361%)	
CommonPorts		Domain name in common port	True (86.833%), False (13.17%)	
DNSRecordType	multi-class categorical	Information about domain and host names	A (550.3%), Cname (39.9%), MX (4.97%)	
CountryCode		IP address country code	20-Categories	
RegisteredCountry		Country code in WHOIS	27-Categories	
CreationDate		Domain name creation date in WHOIS	5 categories	
LastUpdateDate		Domain name last updated date in WHOIS	5 categories	
HttpResponseCode		Response code status	6 categories	

Table 1. *Cont.*

Feature	Datatype	Description	Mode	Mean (μ) ± Std (σ)
ASN	numeric	IP address associated with domain available	-	23,335.8 ± 37,004.9
SubdomainNumber		Number of sub-domains in this DNS		103.07 ± 4243.8
Entropy		Entropy of the DNS		2.87 ± 0.488
EntropyOfSubDomains		Mean entropy of the subdomains		0.0031 ± 0.081
StrangeCharacters		Non-English language characters		3.498 ± 4.4716
ConsoantRatio		Ratio of consonant characters		0.4596 ± 0.146
NumericRatio		Ratio of numeric characters		0.144 ± 0.147
SpecialCharRatio		Ratio of special characters		0.007 ± 0.026
VowelRatio		Ratio of vowel characters		0.262 ± 0.099
ConsonantSequence		Maximum number of consonants		2.719 ± 1.699
VowelSequence		Maximum number of vowels		1.3428 ± 0.555
NumericSequence		Maximum number of numerals		1.516 ± 1.539
SpecialCharSequence		Maximum number of characters		0.1124 ± 0.432
DomainLength		Length of the DNS		26.44 ± 22.341

It can be seen from the table that some of the boolean features like 'HasDkimInfo', 'HasDmarcInfo', 'IpReputation', 'DomainReputation', 'DomainInAlexaDB', and 'CommonPorts' have a less discriminative value. Most of the records have false values except for the 'DomainInAlexaDB' attribute. Therefore, these attributes were not selected during the feature selection mechanism. The entropy is calculated using Shannon entropy method. In the DNS record type MX indicate the mail exchange, CName indicate the registered name of the DNS, while A indicate the translator to convert DNS to IPv4 address. In the dataset the minimum DNS length is 4 and maximum is 153.

3.2. Preprocessing and Feature Selection

The dataset initially contains a total of 32 features describing relevant information about the malicious domains. 'DomainName', 'DNSRecordType', 'CountryCode', 'RegisteredCountry', and 'RegisteredOrg' attributes were removed. 'DomainName' and 'DNSRecordType' were removed from the dataset because they serve as identifiers in the dataset. Furthermore, 'CountryCode', 'RegisteredCountry', and 'RegisteredOrg' contain a huge number of null values; therefore, they were removed. However, binary encoding was performed on the Boolean features like ''MXDnsResponse', 'TXTDnsResponse', 'HasSPFInfo', 'HasDkimInfo', 'HasDmarcInfo', 'DomainInAlexaDB', 'CommonPorts', 'IpReputation', and 'DomainReputation'. Furthermore, the multinomial attribute like 'TLD' was converted into sequential numbers. After the initial preprocessing, 27 features were selected. Feature selection was applied using the Sequential Forward Feature Selector (SFFS). This technique sequentially adds features to an empty set of features and assesses on until the inclusion of more features does not diminish the criterion and the algorithm can achieve the best performance and outcome. The algorithm for the SFFS is represented below (Algorithm 1).

Algorithm 1: Sequential Forward Feature Selection Algorithm

1. Create null set: $X_n \rightarrow \{\varnothing\}$, $n \leftarrow 0$
2. Select optimum remaining features in a set:
   $x^+ = argmax_{x^+ \in X_n}\left[\left(X_n + x^+\right)\right]$
3. If $model_{performance(X_n + x^+)} > model_{performance(X_n)}$
   a. Update $X_{n+1} \leftarrow X_n + x^+$
   b. $n \rightarrow n + 1$
   c. Continue with Step-2

Figure 2 below shows the ranking of the selected features based on information gain. The number of features after the feature selection was 10. The significance of the selected features is shown in the Figure 2. Mostly, the feature selected in the current study was similar to the baseline study [17] except for the numeric ratio and vowel ratio. While the creation date was selected in the baseline, the forward sequential feature selector was not selected for this feature. They used AutoML to select the features and develop the models. In the current study, 10 features were selected, while in the baseline study, 9 features were used.

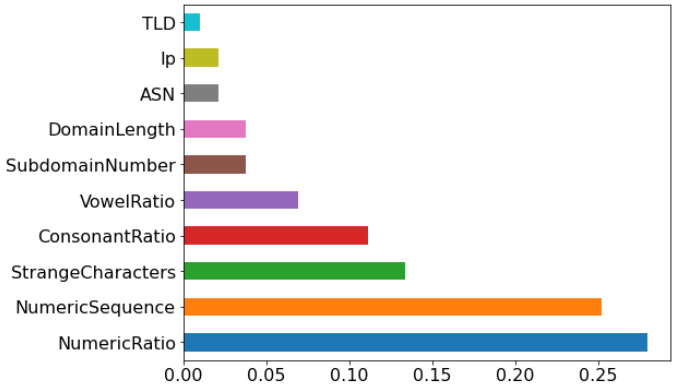

Figure 2. Ranking the features based using feature importance.

3.3. Machine Learning Models

The following models were used in the current study for the detection of malicious DNS. In ML, some of the classifiers contain huge number of parameters and the value of these parameters have a high impact on the learning performance of the classifier and are known as hyperparameters. The value of these hyperparameters needs to be tuned to find the most optimal value, with the aim to minimize the loss function. In the current study, grid search uses the exhaustive searching mechanism for the optimal parameter values. In each iteration, the algorithm finds the parameter values and then evaluates the performance of the algorithm, and the values with the best outcome will be the final values of the classifier's parameter [38]. Gird search cross-validation is applied in the current to find the optimal values for the set of classifiers using GridSearchCV method in the sk-learn python library.

Decision Tree: A decision tree (DT) is a structure that resembles a flowchart, with every inner node indicating an attribute test, each branch representing the test's result, and each leaf node (terminal node) storing a class label. By starting with the root node, the technique for establishing classes for a provided dataset in a DT begins. This method compares the root attribute values with the recorded (actual dataset) attribute values and thus pursues the branch and leaps to the following node depending on the difference.

Naïve Bayes: Naive Bayes (NB) is a statistical classification algorithm that follows the Bayes theorem, and each feature is unrelatable to the other features. Moreover, the Bayes theorem states that the probability of target class A given predictor B is equal to the

likelihood of the predictor B given target class A multiplied by the probability of target class A and divided by the probability of predictor B.

Random Forest: Random Forest (RF) is a classification algorithm that calculates the mean of outcomes of a collection of DTs with various subgroups of a dataset to enhance the dataset's projected accuracy. Instead of depending on an individual DT, the RF gathers predictions from all trees and expects the ultimate result depending on the majority of votes. Performance increases with more trees in the forest and the risk of overfitting is reduced. Table 2 shows the parameters used for the RF classifier.

Table 2. Random Forest classifier parameters value.

Parameter	Value
random_state	1
max_depth	15
min_samples_split	5
min_samples_leaf	1

Extreme Gradient Boosting: Gradient boosting (GB) is an ensemble method for developing predictive models. Generally, the approach is employed in regression and classification operations. Gradient boosting, like other boosting approaches, allows for the generalization and optimization of differentiable loss functions. The Extreme Gradient Boosting (XGB) technique extends the extent of gradient boosting. It mainly aimed at increasing the performance and speed of a ML model. By parallelizing DT computations, it enhances the model's performance. Table 3 contains the parameter values of the XGB using in the study.

Table 3. Extreme Gradient Boosting classifier parameters value.

Parameter	Value
learning_rate	0.1
max_depth	5
min_child_weight	1
eval_metric	mlogloss

AdaBoost: Adaboost (AB) is also an ensemble-based model that utilizes predictions from tiny one-level DTs to create a single prediction. AdaBoost algorithms are used because they attempt to utilize many weak models and then adjust predictions by adding further weak models. The training approach starts with a single decision tree, which is used to detect and weight misclassified samples in the training dataset. Table 4 contains the parameter values for AB.

Table 4. AdaBoost classifier parameters value.

Parameter	Value
n_estimators	100
learning_rate	0.01

CatBoost: CatBoost stands for categorical boosting. It is also based on the DT and gradient boosting but with reduced parameters. Therefore, CatBoost takes less training and testing time as compared with the other ensemble models. This model usually gives good performance with the default parameters. CatBoost can be applied on the numeric data after performing the data encoding. In the proposed study, most of the features were categorical; furthermore, for the non-categorical attributes, data encoding was performed; therefore, Catboost was used. Table 5 contains the parameters values for the Catboost used in the current study.

Table 5. Catboost classifier parameters value.

Parameter	Value
iterations	5
learning_rate	0.8

3.4. Evaluation Metrics

The proposed model's performance was compared in terms of accuracy, precision, recall, F1, Area Under the Curve (AUC), and score time. Furthermore, the outcome of the classifiers was also represented using a confusion matrix. The matrix contains four variables, i.e., malicious DNS that is classified as malicious is represented as True Positive (TP), and malicious DNS that is classified as non-malicious is represented as False Positive (FP). Furthermore, non-malicious DNS that is classified as non-malicious is represented by True Negative (TN), and non-malicious DNS that is classified as malicious is represented by False Negative (FN).

Accuracy (Acc) represents the ratio of correctly classified DNS to the total number of DNS in the test set.

$$Acc = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{1}$$

Recall (R), also known as sensitivity, represents the ability of the model to correctly classify the malicious DNS.

$$R = \frac{TP}{(TP + FN)} \tag{2}$$

Precision (P) represents ratio of correctly predicted malicious DNS to the sum of correctly and incorrectly predicted malicious DNS.

$$P = \frac{TP}{(TP + FP)} \tag{3}$$

F1-score represents the harmonic mean of precision and recall.

$$F1 = \frac{(2 \times P \times R)}{(P + R)} \tag{4}$$

Area Under the Curve (AUC) is used to test and compare binary classification model performance. It quantifies your predictive classification model's discriminating ability.

$$AUC = \frac{(Percent\ Concordant + 0.5 \times Percent\ Tied)}{100} \tag{5}$$

Furthermore, for the detection of malicious DNS, time is one of the significant factors. Therefore, score time is also calculated, which corresponds to the time taken for the model to make the prediction. Score time is represented in seconds.

4. Experiments and Results

To perform the experiments, six ML models were used. The ML classifiers included both simple classifiers, such as DT and NB, and ensemble classifiers, such as RF, XGB, RF, AB, and Catboost. Python (ver. Python 3.10.1) was used to build the proposed models using google collab. The libraries used in the study were pandas, numpy, matplot, sklearn, and Dalex. The dataset used in the study initially contained 32 features and a target class that classifies the samples into malicious and non-malicious domains. Two sets of experiments were performed, i.e., the dataset before feature selection (number of features 27) and after the feature selection (number of features 10). Feature selection was performed using the forward sequential feature selector. The holdout method was used to divide the dataset into training and testing with the ratio of 75:25. Figure 3 contains the flow of the experimental

setup adopted in the current study. Table 6 contains the testing result of the classification models before feature selection. However, Table 6 represents the testing results of the classification model after the feature selection.

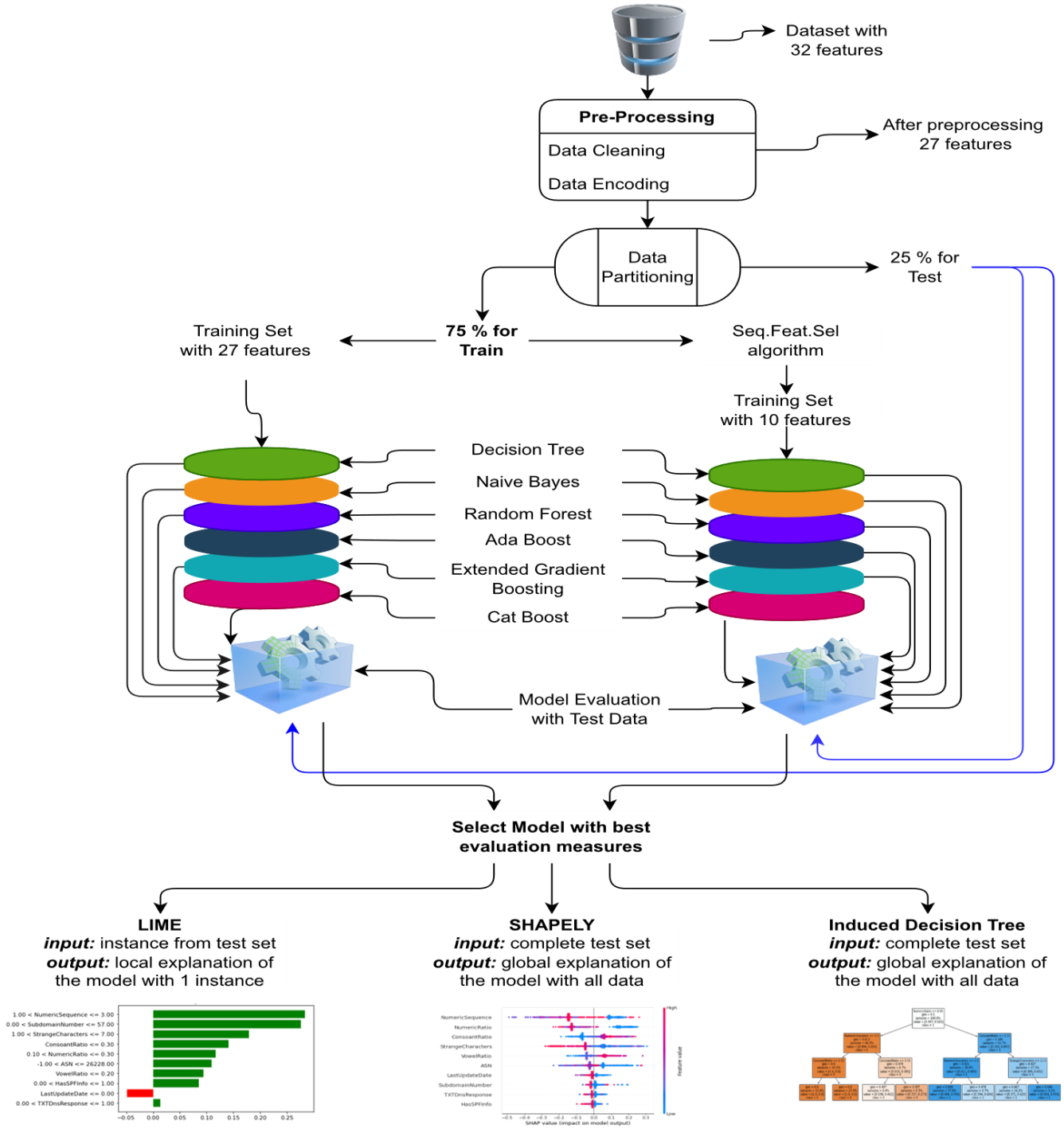

Figure 3. Flow of model development and experimental setup.

Table 6. Classification results of the models before feature selection.

Classifier	AUC	Accuracy	Recall	Prec	F1-Score	Score Time
Decision Tree	0.9582	0.9006	0.8561	0.9385	0.8954	**0.0077**
Naïve Bayes	0.9392	0.9024	0.8840	0.9168	0.9001	0.0138
Random Forest	0.9989	0.9838	**0.9886**	0.9790	0.9838	0.5121
Ada Boost	0.9971	0.9721	0.9743	0.9697	0.9720	0.3601
XGB	**0.9991**	**0.9856**	0.9876	**0.9836**	**0.9856**	0.0962
Cat Boost	0.9974	0.9753	0.9841	0.9668	0.9754	0.0147

Tables 6 and 7 list the performance measures of the applied classifiers before and after feature selection, respectively. XGB outperformed the other classifier with the full features in terms of accuracy, AUC, precision, and F1-score. However, RF achieved the highest recall, which is similar to the recall achieved by XGB. While the DT attained the least score time of 0.0077 s. Furthermore, the accuracy of RF and XGB is also related in both the experiments. Similarly, in the second experiment, with the selected features, XGB attained the best outcome as compared to the other classifier with all the measures except the recall and time elapsed. The performance of the XGB with selected features was reduced by 0.0038, which is a very small difference; however, the number of features was reduced from 27 to 10. However, the score time was reduced to more than half. Like the first experiment, in the second experiment, DT achieved the least score time of 0.0078 s; nevertheless, it achieved a lower outcome in other measures. However, NB achieved the lowest accuracy of 0.8479 after the feature selection. While RF took the huge score time of 0.4500 s. Among the ensemble models, CatBoost achieved the lowest score time, since one of the key benefits of Catboost is that it is faster compared to the other ensemble models.

Table 7. Classification results of the models after feature selection.

Classifier	AUC	Accuracy	Recall	Prec	F1-Score	Score Time
Decision Tree	0.9700	0.9223	0.9066	0.9426	0.9243	**0.0078**
Naïve Bayes	0.9640	0.8479	0.9260	0.7580	0.8336	0.0137
Random Forest	0.9990	0.9812	0.9851	0.9773	0.9812	0.4500
Ada Boost	0.9963	0.9683	0.9718	0.9649	0.9683	0.3235
XGB	**0.9990**	**0.9818**	0.9857	**0.9779**	**0.9818**	0.0424
Cat Boost	0.9970	0.9721	0.9736	0.9709	0.9723	0.0088

Keeping in view the performance of the models and time elapsed, XGB with the reduced features are selected as a final model for the detection of the malicious DNS. XAI was applied on the XGB trained model with the reduced feature set. Figure 4 contains the confusion matrix of the models without the feature selection algorithm. Figure 5 contains the confusion matrix of the models with the selected features. Furthermore, Figure 6 contains the ROC curve of the models after the feature selection.

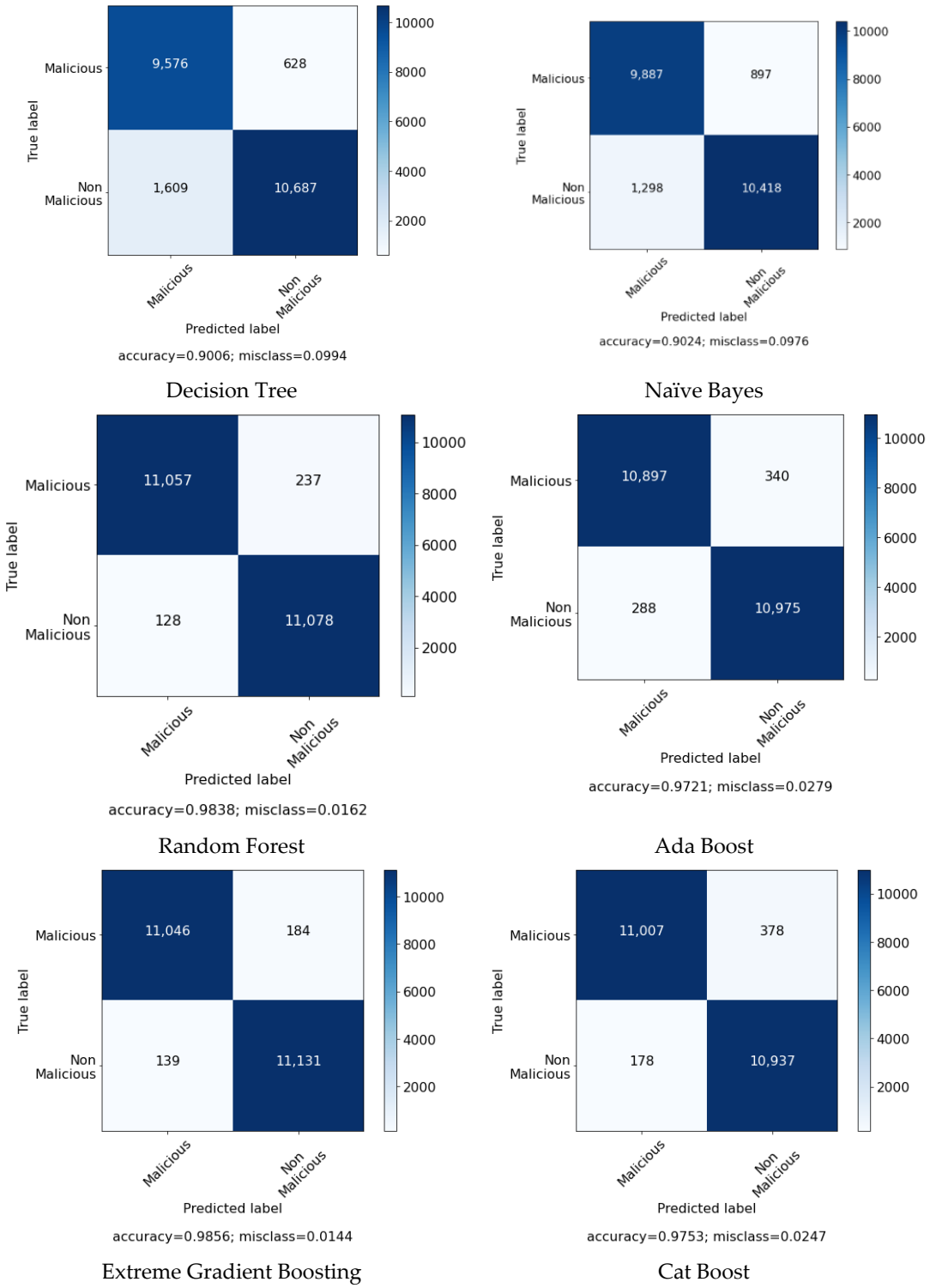

Figure 4. Confusion matrix of the models without using forward sequential feature selector.

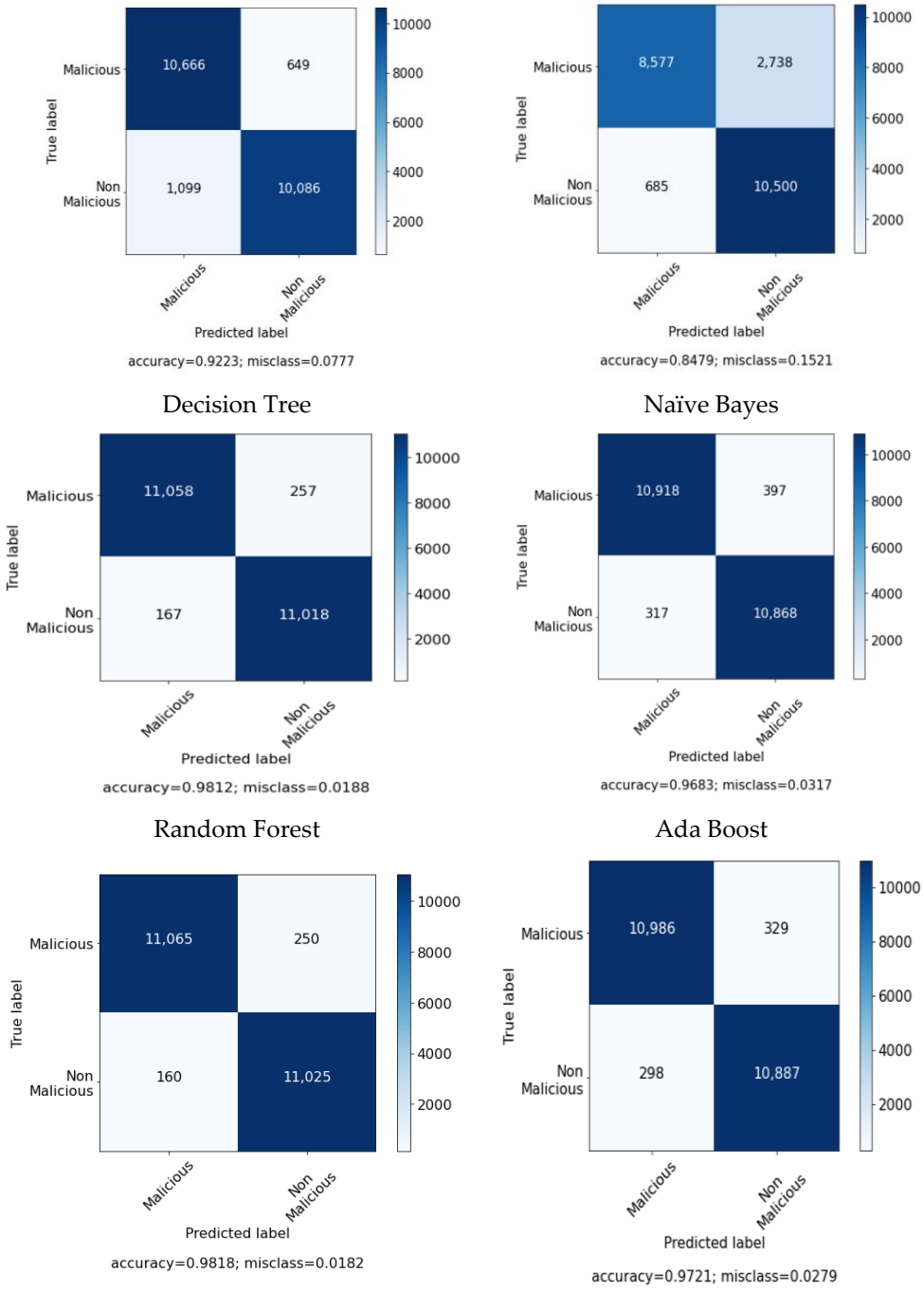

Figure 5. Confusion matrix of the models with the selected features using forward sequential feature selector.

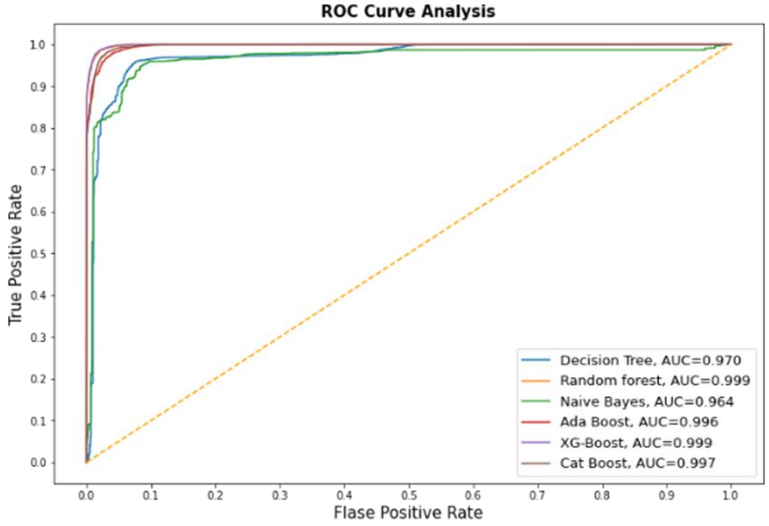

Figure 6. ROC curve of the models with the selected features using forward sequential feature selector.

5. Explainable Artificial Intelligence (XAI)

In the past years, ML- and DL-based models have revolutionized automated prediction and decision-making systems. Nevertheless, the non-linear models like SVM, ANN, ensemble-based models, and DL have achieved empirical success in solving highly complex problems by introducing and increasing several parameters and layers. This increase has no doubt substantially enhanced the performance of the models, but also compromised the interpretability of the models. These models are considered as the black-box models [5]. The opaqueness of these complex models sometimes leads to the resistance because the predictions or decision made are not justifiable. Similarly, in the cyber security, interpretability of the prediction models is a significant factor as it enhances trust. Otherwise, they may endanger critical information, making it vulnerable to threats.

Therefore, the focus of the proposed study is to develop the models for the efficient and interpretable malicious domain detection using ML and XAI. Predictions with the explanations empower the AI-based system with trust and reliability. Furthermore, it allows us to address the question, "why has a particular prediction been made?" [39]. Some of the ML models are inherently interpretable, while others are non-interpretable models. There are two types of interpretation: model specific and model agnostic. Model agnostics have a post hoc interpretation that has been generated using XAI. These post hoc interpretations introduce the interpretability without compromising the performance of the model. Furthermore, the scope of the interpretation can be either global or local. In the current study, the global interpretation is implemented using global surrogate and Shapley. However, for the local interpretation, LIME is used.

5.1. Global Surrogate Model

The Global Surrogate Model (GSM) is a post hoc interpretation technique that is learned to estimate the behavior of the prediction of the opaque models. The aim is to approximate the predictions of a complex model (XGB) using a simple interpretable model (e.g., a DT) that is intuitive, easy to validate, and explainable to a layman. These models are applied on the already trained black-box ML or DL model rather than the training dataset. These models provide the general description of how prediction has been made by the model. In the proposed study, we extract the rules using the induced Decision Tree. Figure 7 contains the induced Decision Tree for the proposed XGB model.

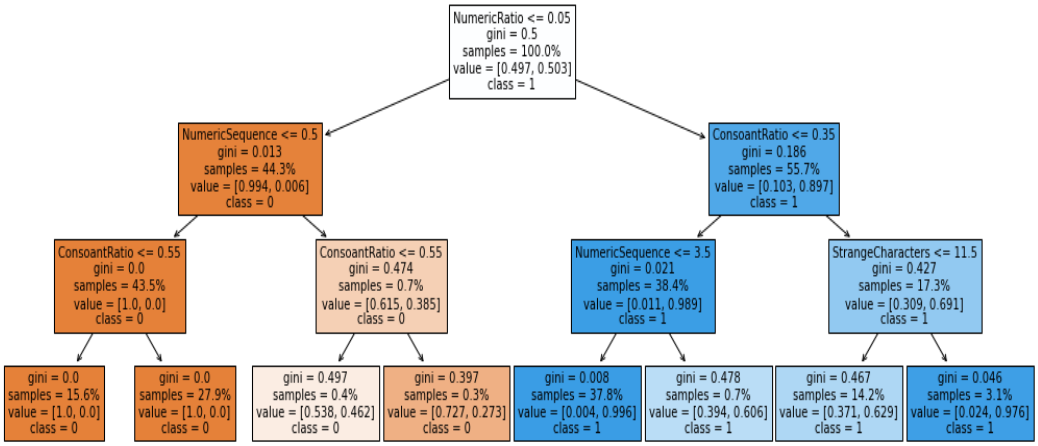

Figure 7. Global Surrogate Model using Decision Tree for XGB.

The Algorithm 2 below is used to create an interpretable global surrogate Decision Tree (DT) model for XGB.

Algorithm 2: Global Model Explanation with Surrogate Model Decision Tree

Input:

a. Train classifier ($fModel$)
b. Pretrain Surrogate Decision Tree model ($sgModel$)
c. Dataset ($Dset$)

Output:

Global Explanation with Global Surrogate Model

Method:

//Get instance from Dataset and collect prediction from fModel

1. LOOP record IN $Dset$ DO

//collect model prediction

a. $Pred \leftarrow fModel(record)$

//train the surrogate Decision Tree model

b. $sgModel.fit(record, Pred)$

2. return $sgModel$

The following rules are extracted from the global surrogate using DT for XGB.

→ if (NumericRatio > 0.05) & (ASN > 375.5) & (NumericSequence $\leq$ 3.5) → class: 1 (probability: 98.12%) | based on 39,205 samples

→ if (NumericRatio $\leq$ 0.05) & (NumericSequence $\leq$ 0.5) & (SubdomainNumber $\leq$ 25.5) → class: 1 (probability: 100.0%) | based on 36,965 samples

→ if (NumericRatio > 0.05) & (ASN $\leq$ 375.5) & (StrangeCharacters $\leq$ 6.5) → class: 1 (probability: 83.55%) | based on 4316 samples

→ if (NumericRatio > 0.05) & (ASN $\leq$ 375.5) & (StrangeCharacters > 6.5) → class: 1 (probability: 89.67%) | based on 3408 samples

→ if (NumericRatio > 0.05) & (ASN > 375.5) & (NumericSequence > 3.5) → class: 1 (probability: 57.84%) | based on 733 samples

→ if (NumericRatio $\leq$ 0.05) & (NumericSequence > 0.5) & (ASN $\leq$ 7618.5) → class: 1 (probability: 98.15%) | based on 324 samples

→ if (NumericRatio $\leq$ 0.05) & (NumericSequence > 0.5) & (ASN > 7618.5) → class: 1 (probability: 79.14%) | based on 302 samples

→ if (NumericRatio $\leq$ 0.05) & (NumericSequence $\leq$ 0.5) & (SubdomainNumber > 25.5) → class: 1 (probability: 99.19%) | based on 247 samples

5.2. Shapley Values

Shapley values allow us to generate the global and local explanation of the ML models. Shapley uses the concept of game theory to identify the contribution of each feature in the prediction. In the current study, Shapley values were computed for the testing data to check the global interpretation. Figure 8 contains the Shapley values for the malicious DNS detection. The figure contains two colors for representing the higher and lower value attributes. Red represents the high, while blue represents the low.

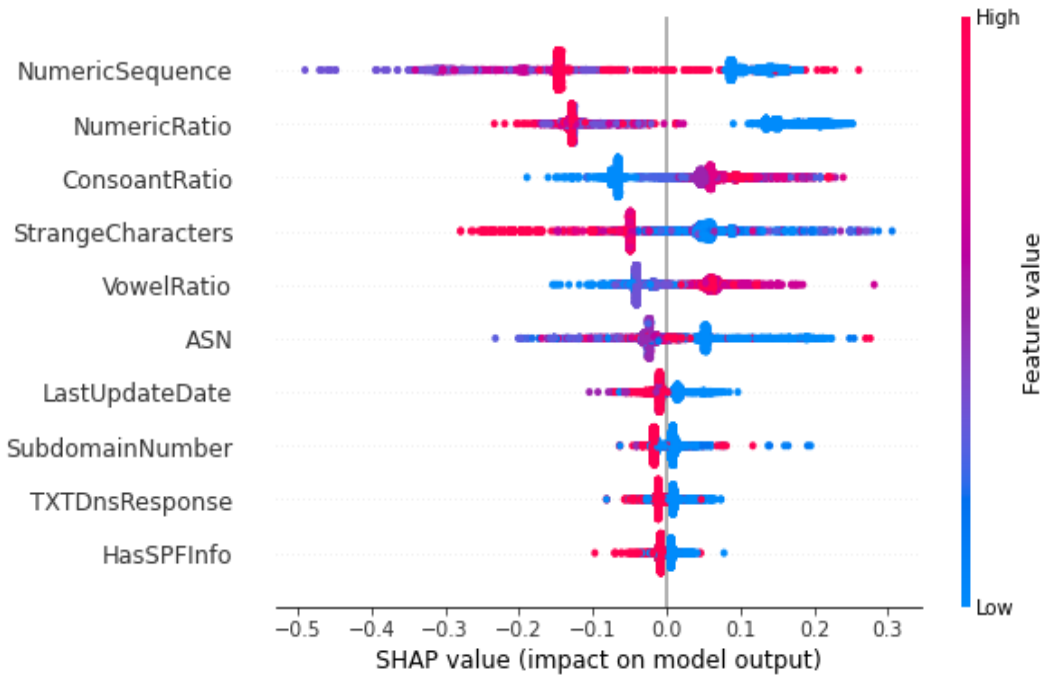

Figure 8. Global explanation using Shapley values.

The Algorithm 3 given below is used for calculating the Shapley value for a single attribute is.

5.3. Local Interpretable Model-Agnostic Explanations (LIME)

Conversely, Local Interpretable Model-Agnostic Explanations (LIME) generate the local interpretation of the model and is used to interpret the behavior of a model with a single instance of dataset, i.e., it is used to explain the individual predictions or to find out which input features are important for the particular prediction. For LIME, from the test data, samples are randomly selected for generating an explanation. In the current study, we randomly selected two samples, one sample for each category, i.e., malicious and non-malicious. Figure 9 represents the LIME for the malicious instance and Figure 10 represents the LIME for the non-malicious instance. In the figures, the green color represents the features that are contributing to the malicious instance and the red color indicates the features that are contributing the non-malicious class. In Figure 9, among the selected features, the last update date contributes to the non-malicious domain class. However, the remaining nine attributes contribute towards the malicious class.

Algorithm 3: Global Feature Explanation with SHAP

Input:

a. Pretrain classifier ($fModel$)
b. Example/instance (x)
c. Dataset ($Dset$)
d. Number of iteration N

Output:

Global Explanation with average SHAP values for each feature (S)

Method:

//Get instance from Dataset and collect prediction from fModel

1. LOOP n IN N DO
   //select random instance from Dset
   a. $rec \leftarrow Dset(n)$
   //choose random premutation p of the feature values
   b. *Order instance x :* $x_0 = \{x_1, \ldots x_n, \ldots x_p\}$
   c. *Order instance rec :* $rec_0 = \{rec_1, \ldots, rec_n, \ldots rec_p\}$
   //construct two new instances
   //with features
   d. $n:\quad x_{+n} = \{x_1, \ldots x_{n-1}, x_n, rec_{n+1}, \ldots rec_p\}$
   //without features
   e. $n:\quad x_{-n} = \{x_1, \ldots x_{n-1}, rec_{n+1}, \ldots rec_p\}$
   //compute marginal contribution
   f. $C_i^n = f(x_{+n}) - f(x_{-n})$
   //compute SHAP value as the average
   g. $C_i(x) = \frac{1}{N} \sum_{i=1}^{N} C_i^n$
   h. $S \leftarrow C_i(x)$
2. return S

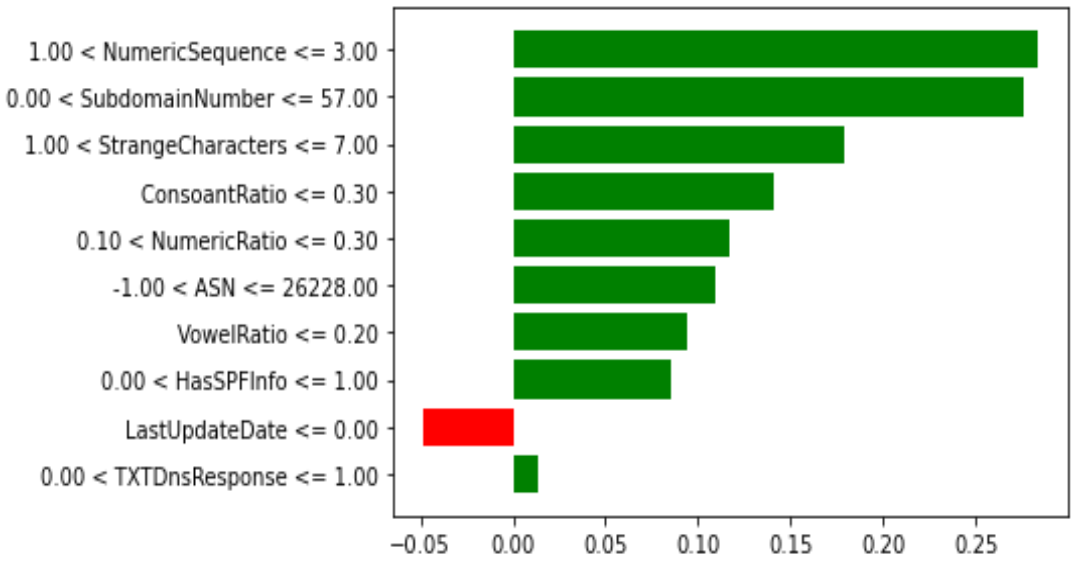

Figure 9. Local explanation for the malicious domain sample using LIME.

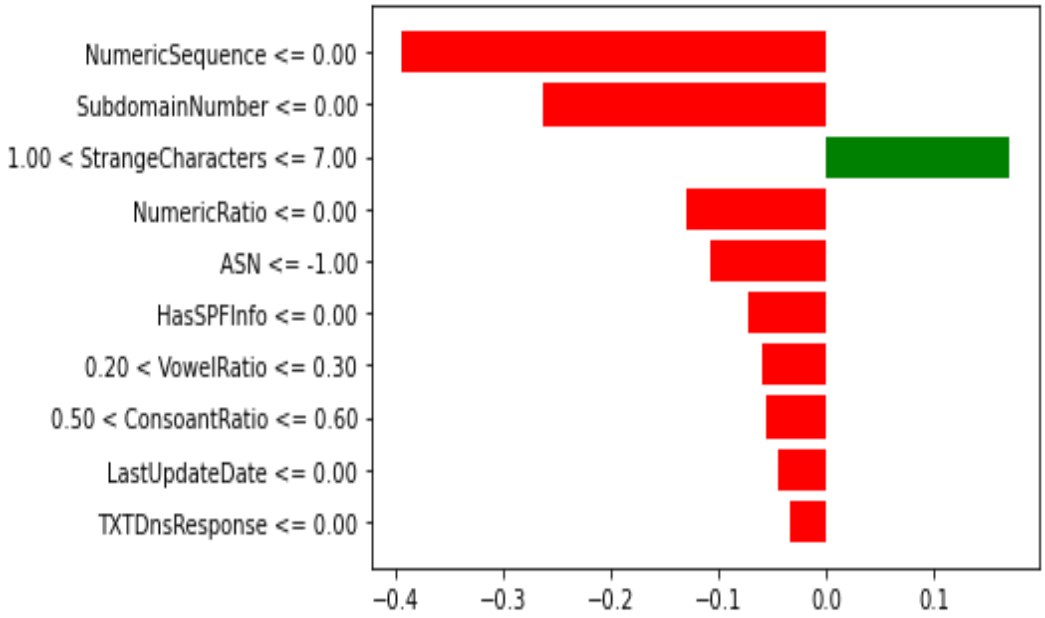

Figure 10. Local explanation for the non-malicious domain sample using LIME.

The Algorithm 4 given below is used to create the LIME model.

Algorithm 4: Local Explanation with LIME

Input:

a. Train classifier ($fModel$)
b. Instances (x) and it's Perturb datapoints (x')
c. Dataset ($Dset$) with N samples
d. Similarity kernel SK_x

Output:

Local Explanation with LIME ($\mathbb{L}$)

Method:

//create empty cluster
1. $\mathcal{C} \leftarrow \{\}$
2. LOOP *ind* IN $\{1, 2, 3, \ldots, \mathbb{N}\}$ DO
   //collect Perturb data around (x')
   a. $c'_{ind} \leftarrow sample\_around(x')$
   //collect the data in cluster
   b. $\mathcal{C} \leftarrow \mathcal{C} \cup \left(c'_{ind}, fModel(c_{ind}), SK_x(c_{ind})\right)$
   //Calculate LR with c'_{ind} *as features,* $fModel(c_{ind})$ *as target*
3. $\mathbb{L} \leftarrow LinearRegression(\mathcal{C}, \mathbb{N})$,
4. return $\mathbb{L}$

6. Further Discussion

The proposed study aims to develop a model for the detection of malicious DNS with the better performance, speed, and interpretability, which could protect the system and data from the malicious attack. The models used were classical ML models like DT and NB; furthermore, ensemble models like RF, AB, XGB, and CB were used with the balanced malicious DNS dataset published by Magalhaes et al. [19]. Feature selection was performed using the sequential feature selector. Furthermore, XAI was used to incorporate the interpretability in the ensemble-based black-box ML models. The current dataset was also used in the study by Marques et al. [17]. The authors initially published the dataset and then used the ML models for the malicious DNS. They used several classical ML models like SVM, LR, LDA, KNN, DT (CART), and NB. Furthermore, three feature selection algorithms were compared like univariate filter, Recursive Feature Elimination (RFE), and feature

importance. They found that CART achieved the highest results compared with the other models. Table 8 contains the comparison of the proposed model with the benchmark. As seen from the table, the proposed study achieved better results compared with the benchmark. However, the score time of benchmark was 0.025 s, which is lower compared with the current study, i.e., 0.0424 s. Nevertheless, score time is one of the significant measures, but considering the outcome achieved in terms of recall, accuracy, precision, and F1-score, the difference of the score time is acceptable.

Table 8. Performance comparison of the proposed study and the benchmark.

Ref	Accuracy	Recall	Prec	F1-Score	Score Time
[17]	0.962	0.952	0.973	0.9590	0.025 s
Proposed Study	**0.9818**	0.9857	**0.9779**	**0.9818**	0.0424 s

To summarize, the proposed study achieved significant outcomes. Furthermore, the XAI considered in the study enhanced the interpretability along with the performance of the model. XAI has recently gained a lot of attention in several domains like healthcare, education, business, engineering, cyber security, and network monitoring. Indeed, the model achieved promising results but also suffers from the limitation that the study only investigate a single dataset. Furthermore, the time to detect the malicious DNS needs to be further reduced by investigating some light models like LGBM.

7. Conclusions

The significant effects of malicious domains motivated many researchers to find more effective methods to classify malicious and non-malicious domains. Additionally, the traditional approaches would not be beneficial in the near future due to the dependency on human expertise that may not cooperate with the rapid development of the internet. In this paper, we examined multiple ML models. such as DT, NB, RF, AB, XGB, and CB, to detect malicious domains using the datasets containing 45,000 samples of each of the malicious and non-malicious domains. Feature selection was performed using the forward feature selector. Two experiments were conducted, i.e., using the preprocessed data (27 features) and the reduced features data (10 features) after the feature selection. XGB outperformed all the classifiers using 27 features with an AUC of 0.9991, accuracy of 0.9856, precision of 0.9836, F1 score of 0.9856, while the RF achieved the highest recall of 0.9886 without feature selection. The score time is one of the most important factors, and the results achieved by XGB with and without feature selection were similar, while the score time was much less; therefore, we considered XGB with the feature selection as the final model for the XAI. The XGB algorithm using feature selection techniques produced values of 0.999, 0.9818, 0.9779, and 0.9818 in AUC, accuracy, precision, and F1-score, respectively, with the score time of 0.0424. XAI was used for the global and local interpretation using the Global Surrogate Model, Shapley, and LIME. The study outperformed the benchmark study except for the score time and has thus shown promising results. However, there is still room for improvement by investigating the performance of the proposed model with larger and more datasets.

Author Contributions: Conceptualization, N.A., I.U.K., S.M., A.A., F.M.A., R.M.A. and R.B.; methodology, N.A., I.U.K., S.M., A.A., F.M.A., R.M.A. and R.B.; software, N.A. and I.U.K.; validation, N.A. and I.U.K.; formal analysis, N.A., I.U.K., S.M., A.A., F.M.A., R.M.A. and R.B.; investigation, N.A., I.U.K., S.M., A.A., F.M.A., R.M.A. and R.B.; resources, N.A. and I.U.K.; data curation, N.A. and I.U.K.; writing—original draft preparation, N.A., S.M., A.A., F.M.A., R.M.A. and R.B.; writing—review and editing, N.A. and I.U.K.; visualization, N.A. and I.U.K.; supervision, N.A. and I.U.K.; project administration, N.A. and I.U.K.; funding acquisition, N.A., I.U.K., S.M., A.A., F.M.A., R.M.A. and R.B. All authors have read and agreed to the published version of the manuscript.

Funding: We would like to thank SAUDI ARAMCO Cybersecurity Chair for funding this project.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, K.; Yu, X.; Wang, J. A Review: How to Detect Malicious Domains. In *International Conference on Artificial Intelligence and Security*; Springer: Cham, Switzerland, 2021; pp. 152–162.
2. DNS Was Not Designed for Security. Available online: https://www.cloudflare.com/learning/insights-dns-landscape/ (accessed on 5 April 2022).
3. Ramdas, A.; Muthukrishnan, R. A Survey on DNS Security Issues and Mitigation Techniques. In Proceedings of the International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, 15–17 May 2019; pp. 781–784.
4. Kim, T.H.; Reeves, D. A survey of domain name system vulnerabilities and attacks. *J. Surveill. Secur. Saf.* **2020**, *1*, 34–60. [CrossRef]
5. Arrieta, A.B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; Garcia, S.; Gil-Lopez, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2019**, *58*, 82–115. [CrossRef]
6. Linardatos, P.; Papastefanopoulos, V.; Kotsiantis, S. Explainable AI: A Review of Machine Learning Interpretability Methods. *Entropy* **2020**, *23*, 18. [CrossRef] [PubMed]
7. IBM-Watson-Explainable AI. Available online: https://www.ibm.com/watson/explainable-ai (accessed on 24 May 2022).
8. Holzinger, A.; Saranti, A.; Molnar, C.; Biecek, P.; Samek, W. Explainable AI methods—A brief overview. In *International Workshop on Extending Explainable AI beyond Deep Models and Classifiers*; Springer: Cham, Switzerland, 2022; pp. 13–38.
9. Mahbooba, B.; Timilsina, M.; Sahal, R.; Serrano, M. Explainable Artificial Intelligence (XAI) to Enhance Trust Management in Intrusion Detection Systems Using Decision Tree Model. *Complexity* **2021**, *2021*, 6634811. [CrossRef]
10. Le, T.T.H.; Kim, H.; Kang, H.; Kim, H. Classification and Explanation for Intrusion Detection System Based on Ensemble Trees and SHAP Method. *Sensors* **2022**, *22*, 1154. [CrossRef] [PubMed]
11. Guo, W. Explainable Artificial Intelligence for 6G: Improving Trust between Human and Machine. *IEEE Commun. Mag.* **2020**, *58*, 39–45. [CrossRef]
12. Antoniadi, A.; Du, Y.; Guendouz, Y.; Wei, L.; Mazo, C.; Becker, B.; Mooney, C. Current Challenges and Future Opportunities for XAI in Machine Learning-Based Clinical Decision Support Systems: A Systematic Review. *Appl. Sci.* **2021**, *11*, 5088. [CrossRef]
13. Mankodiya, H.; Obaidat, M.S.; Gupta, R.; Tanwar, S. XAI-AV: Explainable Artificial Intelligence for Trust Management in Autonomous Vehicles. In Proceedings of the International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI), Beijing, China, 15–17 October 2021; pp. 1–5. [CrossRef]
14. Kidmose, E.; Stevanovic, M.; Pedersen, J.M. Detection of Malicious domains through lexical analysis. In Proceedings of the International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Glasgow, UK, 11–12 June 2018; pp. 1–5.
15. Zhu, J.; Zou, F. Detecting Malicious Domains Using Modified SVM Model. In Proceedings of the IEEE 21st International Conference on High Performance Computing and Communications, Zhangjiajie, China, 10–12 August 2019; pp. 492–499.
16. Almashhadani, A.O.; Kaiiali, M.; Carlin, D.; Sezer, S. MaldomDetector: A system for detecting algorithmically generated domain names with machine learning. *Comput. Secur.* **2020**, *93*, 101787. [CrossRef]
17. Marques, C.; Malta, S.; Magalhães, J. DNS Firewall Based on Machine Learning. *Future Internet* **2021**, *13*, 309. [CrossRef]
18. Palaniappan, G.; Sangeetha, S.; Rajendran, B.; Goyal, S.; Bindhumadhava, B.S. Malicious Domain Detection Using Machine Learning On Domain Name Features, Host-Based Features and Web-Based Features. *Procedia Comput. Sci.* **2020**, *171*, 654–661. [CrossRef]
19. Magalhaes, F.; Magalhaes, J.P. Adopting Machine Learning to Support the Detection of Malicious Domain Names. In Proceedings of the 7th International Conference on Internet of Things: Systems, Management and Security (IOTSMS), Paris, France, 14–16 December 2020; pp. 1–6. [CrossRef]
20. MontazeriShatoori, M.; Davidson, L.; Kaur, G.; Lashkari, A.H. Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic. In Proceedings of the IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Calgary, AB, Canada, 17–22 August 2020; pp. 63–70. [CrossRef]
21. DoHBrw 2020 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. Available online: https://www.unb.ca/cic/datasets/dohbrw-2020.html (accessed on 1 March 2022).
22. Singh, S.K.; Roy, P.K. Detecting Malicious DNS over HTTPS Traffic Using Machine Learning. In Proceedings of the International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT), Sakheer, Bahrain, 20–21 December 2020; pp. 1–6.
23. Singh, S.S.K.; Roy, P.K. Vulnerability Detection of DNS over HTTPS Traffic using Ensemble Machine Learning. *Int. J. Comput. Digit. Syst.* **2021**. Available online: https://journal.uob.edu.bh/handle/123456789/4472 (accessed on 7 June 2022).

24. Behnke, M.; Briner, N.; Cullen, D.; Schwerdtfeger, K.; Warren, J.; Basnet, R.; Doleck, T. Feature Engineering and Machine Learning Model Comparison for Malicious Activity Detection in the DNS-Over-HTTPS Protocol. *IEEE Access* **2021**, *9*, 129902–129916. [CrossRef]
25. Akarsh, S.; Sriram, S.; Poornachandran, P.; Menon, V.K.; Soman, K.P. Deep Learning Framework for Domain Generation Algorithms Prediction Using Long Short-term Memory. In Proceedings of the 5th International Conference on Advanced Computing Communication Systems (ICACCS), Coimbatore, India, 15–16 March 2019; pp. 666–671. [CrossRef]
26. Chen, Y.; Zhang, S.; Liu, J.; Li, B. Towards a Deep Learning Approach for Detecting Malicious Domains. In Proceedings of the IEEE International Conference on Smart Cloud (SmartCloud), New York, NY, USA, 21–23 September 2018; pp. 190–195. [CrossRef]
27. Bharathi, B.; Bhuvana, J. Domain name detection and classification using deep neural networks. In *International Symposium on Security in Computing and Communication*; Springer: Singapore, 2019. [CrossRef]
28. Ma, D.; Zhang, S.; Kong, F.; Fu, Z. Malicious Domain Name Detection Based on Doc2vec and Hybrid Network. In *IOP Conference Series: Earth and Environmental Science*; IOP Publishing: Bristol, UK, 2021; Volume 693. [CrossRef]
29. Vinayakumar, R.; Soman, K.P.; Poornachandran, P. Detecting malicious domain names using deep learning approaches at scale. *J. Intell. Fuzzy Syst.* **2018**, *34*, 1355–1367. [CrossRef]
30. Xu, C.; Shen, J.; Du, X. Detection method of domain names generated by DGAs based on semantic representation and deep neural network. *Comput. Secur.* **2019**, *85*, 77–88. [CrossRef]
31. Shi, Y.; Chen, G.; Li, J. Malicious Domain Name Detection Based on Extreme Machine Learning. *Neural Process. Lett.* **2017**, *48*, 1347–1357. [CrossRef]
32. Gunning, D.; Aha, D.W. DARPA's explainable artificial intelligence program. *AI Mag.* **2019**, *40*, 44–58.
33. Molnar, C. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*, 2nd ed.; Munich, Germany, 2022. Available online: https://christophm.github.io/interpretable-ml-book/ (accessed on 7 June 2022).
34. Marques, C.; Malta, S.; Magalhães, J.P. DNS dataset for malicious domains detection. *Data Brief* **2021**, *38*, 107342. [CrossRef] [PubMed]
35. Forward DNS (FDNS) | Rapid7 Open Data. Available online: https://opendata.rapid7.com/sonar.fdns_v2/ (accessed on 20 April 2022).
36. SANS Internet Storm Center. 2020. Available online: https://www.dshield.org/feeds/suspiciousdomains_Low.txt (accessed on 20 April 2022).
37. Benign and Malicious Domains Based on DNS Logs. Available online: https://data.mendeley.com/datasets/623sshkdrz/5 (accessed on 24 May 2022).
38. Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for hyper-parameter optimization. *Adv. Neural Inf. Process. Syst. NIPS* **2011**, *24*, 1–9. Available online: https://proceedings.neurips.cc/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf (accessed on 8 June 2022).
39. Diepgrond, D. Can Prediction Explanations Be Trusted? On the Evaluation of Interpretable Machine Learning Methods. Ph.D. Thesis, University of Groningen, Groningen, The Netherlands, 2020.