computers

MDPI

Article

Detection of DoH Traffic Tunnels Using Deep Learning for Encrypted Traffic Classification

Ahmad Reda Alzighaibi

College of Computer Science and Engineering, Taibah University, Yanbu 42353, Saudi Arabia; azighaibi@taibahu.edu.sa

Abstract: Currently, the primary concerns on the Internet are security and privacy, particularly in encrypted communications to prevent snooping and modification of Domain Name System (DNS) data by hackers who may attack using the HTTP protocol to gain illegal access to the information. DNS over HTTPS (DoH) is the new protocol that has made remarkable progress in encrypting Domain Name System traffic to prevent modifying DNS traffic and spying. To alleviate these challenges, this study explored the detection of DoH traffic tunnels of encrypted traffic, with the aim to determine the gained information through the use of HTTP. To implement the proposed work, state-of-the-art machine learning algorithms were used including Random Forest (RF), Gaussian Naive Bayes (GNB), Logistic Regression (LR), k-Nearest Neighbor (KNN), the Support Vector Classifier (SVC), Linear Discriminant Analysis (LDA), Decision Tree (DT), Adaboost, Gradient Boost (SGD), and LSTM neural networks. Moreover, ensemble models consisting of multiple base classifiers were utilized to carry out a series of experiments and conduct a comparative study. The CIRA-CIC-DoHBrw2020 dataset was used for experimentation. The experimental findings showed that the detection accuracy of the stacking model for binary classification was 99.99%. In the multiclass classification, the gradient boosting model scored maximum values of 90.71%, 90.71%, 90.87%, and 91.18% in Accuracy, Recall, Precision, and AUC. Moreover, the micro average ROC curve for the LSTM model scored 98%.

Keywords: DNS over HTTPS (DoH); CIRA-CIC-DoHBrw-2020; deep Learning; encrypted traffic classification

Citation: Alzighaibi, A.R. Detection of DoH Traffic Tunnels Using Deep Learning for Encrypted Traffic Classification. *Computers* **2023**, *12*, 47. https://doi.org/10.3390/computers12030047

Academic Editor: Paolo Bellavista

Received: 2 February 2023
Revised: 12 February 2023
Accepted: 13 February 2023
Published: 22 February 2023

1. Introduction

DNS over HTTPS is a new protocol that encrypts remote Domain Name System traffic using an encrypted HTTPS connection. DNS over HTTPS plays a vital role in DNS encryption by blocking DNS resolutions from active attackers to protect user privacy. According to [1], the DNS lacks in-built security mechanisms. Hence, DoH remains an ideal solution for data security and privacy. As result, the wider research community is engaging in the usage of encrypted communications in the area of the DNS protocol.

Most communication technologies depend on a Domain Name System that assigns the human reading destination of the Internet to an IP address until it communicates with two endpoints. In other words, the bulk of DNS queries and answers is transmitted and is open to traffic analyses and spies. Many research works are using standards, browser implementations, and DNS over HTTPS to send information between clients and third parties that can run the DoH resolutions, which help reduce privacy risks [2].

DoH traffic tunnel detection is vital to improve user privacy and security by avoiding spying and DNS data modification by encrypting the data between the DoH client and the DoH-based DNS resolution using the HTTPS protocol [3]. DoH empowers DNS clients to query hostnames with regular Transportation Layer Security (TLS) through HTTP exchange security. The DNS protection implementation offers confidentiality and integrity enhancements to end-users but compromises the network with additional challenges [4]. The DoH is essential to avoid information leakage through encrypting communications on the web using HTTPS connections for exchanging DNS traffic [5].

DoH operates in the same way as DNS, except that HTTPS sessions secure the requests and limit the amount of data transferred throughout inquiries. Internet browsers such as Google Chrome, Microsoft Edge, and Mozilla Firefox work by using encrypted DoH to provide users with increased data privacy and security. Instead of delivering the whole domain name that the user's browser attempts to resolve, DoH sends only the piece of the domain name required to complete the current step in the name resolution process [6].

Figure 1 presents how DNS over HTTPS (DoH) works.

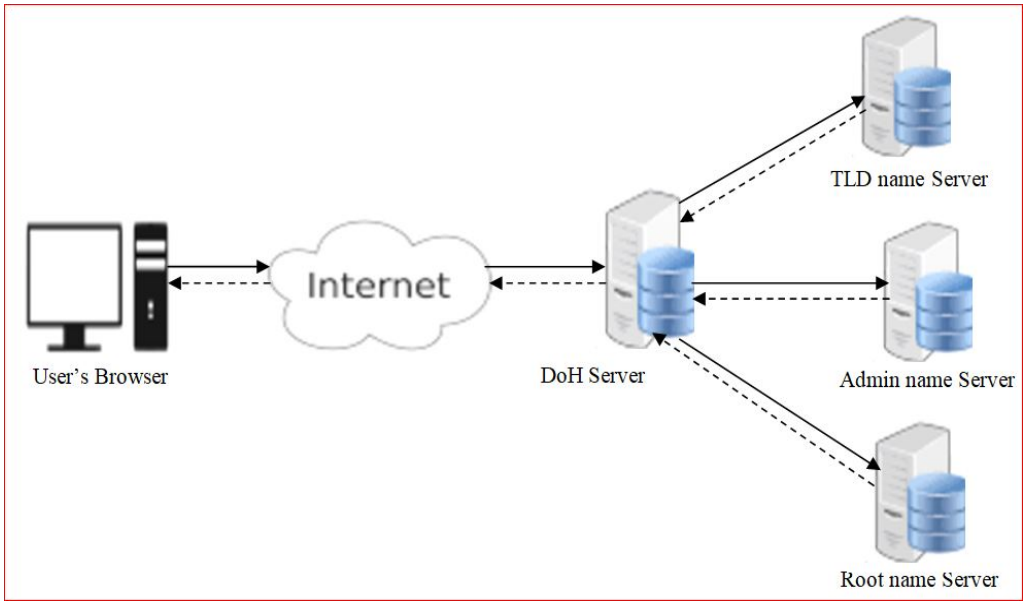

Figure 1. DNS over HTTPS (DoH).

Deep Learning (DL) algorithms, both supervised and unsupervised learning methods, learn the depths' hierarchy based on layers of Artificial Neural Networks (ANNs) [7–10]. Deep Learning advances in information technology of many layers of processing based on data from each layer's input level will generate non-linear responses. DL's functionality is imitated by the human brain and neuron systems for signal processing [11].

The LSTM model was used in this study due to its capability to give better results, which improve the model's prediction and robustness. The model overcomes the overfitting trade-off compared to other traditional machine learning methods. The CIRA-CIC-DoHBrw-2020 dataset, created by the Canadian Institute for Cybersecurity (CIC) project sponsored by the Canadian Internet Registration Authority (CIRA) in 2020, was used for model creation, validation, and deployment [6].

The contributions of this research work are outlined as follows:

- We introduce fewer features, which improves the training efficiency and classification performance.
- It employs time-series-based preprocessing, which ultimately minimizes the possibility of inconsistencies in the experimental results.
- It presents the leveraging of ML models to detect malicious activities designed to be deployed in the internal network of an enterprise.
- The proposed method combines the benefits of machine learning and deep neural networks in terms of learning the potential correlation between features.
- The proposed stacking model combines the SMOTE method and the stacking model to detect tunneling in DNS traffic in the imbalanced CIC-DoHBrw2020 dataset with high Accuracy.

The subsequent sections of the paper are organized as follows: Section 2 is about the relevant related works, and Section 3 presents the employed methodologies. Section 4

describes the dataset. Section 5 presents the experimental results and discussions. Finally, Section 6 comprises the conclusions and recommendations for future studies.

2. Related Work

This section discusses related works analyzed concerning the detection of DoH traffic tunnels for encrypted traffic classification. To address the challenge, many researchers have looked into different Machine Learning (ML) techniques. However, still, there are challenges that make it hard to use ML. This section is divided into three distinct parts. In Part 1 gives a comprehensive analysis of a number of previous studies that are relevant to the dataset and findings on the same dataset. Part 2 includes a summary of the relevant research and a description of the gaps in the form of an extended problem statement. Finally, Part 3 explains the motivation for the current study and gives an explanation of why this study was performed. The recent related works are reviewed and presented as follows:

Vekshin et al. [12] suggested that DoH can be used instead of the traditional DNS for encrypted traffic analysis. The evaluation was performed to assess what information is gained from HTTPS extended IP flow data using five popular classifiers to find the best DoH methods. The experimental results showed that DoH distinguishing between DoH clients was 99.9% Accuracy.

Hounsel et al. [13] showed the impact of DoH and DoT on the output of name resolution and the delivery of information. The experimental results showed that the response times for DoH and DoT can be maximized in terms of page loading times compared to the traditional DNS (Do53), and DoT is better compared to the DoH and Do53 approaches.

Bushart et al. [14] proposed a traffic analysis approach that incorporates details about the size and time to reduce user visits to websites based solely on encrypted and padded DNS traces. The study focused on DNS encryption by DoT and DoH to preserve user privacy, hiding DNS resolutions from passive opponents. The results showed that the privacy targets of state-of-the-art message padding in DoT/DoH strategies and the attacks must eliminate the entropy between request responses.

Lu et al. [15] carried out an end-to-end and large-scale analysis on DNS-over-encryption. They found that 25% of DNS-over-TLS service providers use invalid SSL certificates compared to the traditional DNS. So far, fewer users use DNS-over-encryption, but it has witnessed a growing trend.

Singanamalla et al. [16] presented the DoH protocol for performance comparison with the DoH and DoT protocols in resolving protecting the client's information and identity to improve client privacy. Moreover, Deccio et al. [17] described the analysis of DoH and DoT accessibility on transparent host names and authoritative DNS servers. The result showed that DoH and DoT services operate on a fraction of available solutions, and among these, there are important providers of public DNS services.

Singh et al. [18] conducted a study to detect the DNS level's malicious activity in the DoH environment. To do this, the CIRA-CIC-DoHBrw-2020 dataset and different machine learning classifiers such as LR, GB, RF, NB, and KNN were deployed. The experimental results showed that the GB and RF outperformed the other classifiers.

Austin et al. [19] carried out a comparative study on the effects of DoH, DoT, and DNS by measuring Do53, DoT, and DoH on the query response and page load times from five global vantage points. The experimental results showed that the DoH and DoT response times were higher compared to Do53. Both protocols can perform better than Do53 on the page load times. However, web pages loaded successfully more often with Do53 and DoT than with DoH. Similarly, López et al. [20] presented a passive analysis of DoH traffic to map daily DNS requests and responses over TLS-encapsulated HTTP packets, and different techniques were employed to analyze the content of DoH communications for their detection. Palau et al. [21] implemented a CNN to detect the Accuracy and threats in the DNS, but lacked quality datasets to test DNS tunneling connections. The experimental

results showed an Accuracy of 92% for the correctly identified total model tunneling domains by the CNN with a false positive rate close to 0.8%.

Houser et al. [22] developed a DoT fingerprinting system to evaluate DoT traffic and decide whether adversaries have visited websites of interest to a person. The experimental results showed that, when DNS messages were not loaded, the DoT traffic for websites was detected with a false positive rate of less than 0.5 percent and a false negative rate of less than 17 percent.

Huang et al. [23] tested six browsers using four network attacks specific to preserving DNS privacy and dignity, and the experimental results from the proposed method showed that all combinations led to successful attacks. Moreover, Montazeri Shatoori et al. [6] identified tunneling activities that used DNS communications over HTTPS by providing a two-layered approach using time-series classifiers to detect and classify DoH traffic.

Banadaki et al. [24] proposed a two-layer approach for detecting DoH traffic from non-DoH traffic in Layer 1 and characterizing benign-DoH from malicious-DoH traffic in Layer 2 using the CIRA-CIC-DoHBrw-2020 dataset and six classifiers. The results showed that the XGBoost and LGBM classifiers outperformed the other classifiers, achieving 100% Accuracy in both Layers 1 and 2.

Meanwhile, the authors presented machine learning classifiers that can achieve faster classification by taking advantage of imbalanced classes in [6,25]. The study demonstrated that there was a high level of Accuracy and a low level of latency. In contrast to the previous works, our proposed work implements the imbalance processing technique (SMOTE) as a core component of the model. Additionally, our work uses a random search cross-validation method to perform pre-classification hyper-parameter tuning on both the base classifiers and the Deep Learning model. This helps to fine-tune the parameters of both of these models, and this technique motivated us to carry out this research work.

3. The Proposed Method

This section presents the methodology used in this study, including the Deep Learning model and the proposed research approach used in this study to detect DoH traffic tunnels in the collected dataset. Figure 2 presents the general steps of the proposed method.

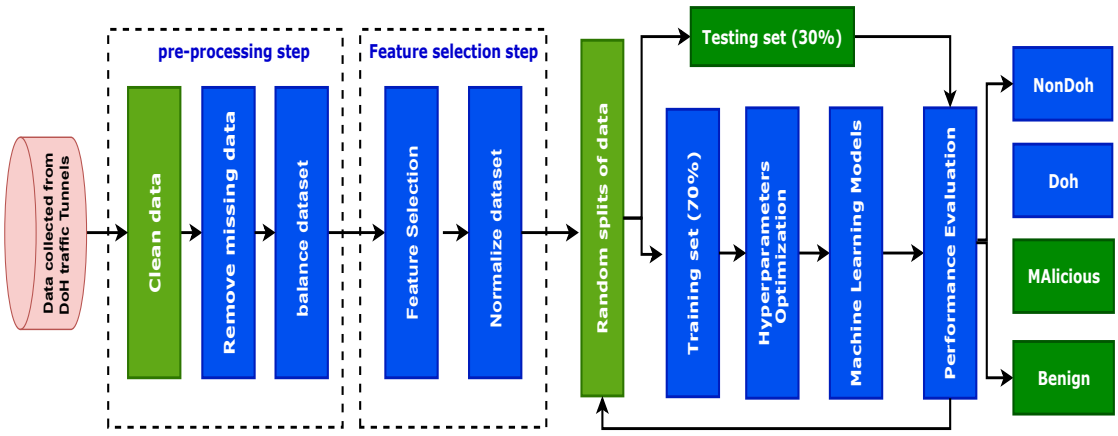

Figure 2. The steps of the proposed method.

3.1. Deep Learning

Deep Learning (DL) algorithms comprise supervised and unsupervised learning methods based on several layers of Artificial Neural Networks (ANNs), which help to learn various features across hidden layers in the depths' hierarchy [8,26,27]. The LSTM model is a typical one that utilizes hierarchical feature learning [28]. It is a form of Recurrent Neural Network (RNN), which may learn long-term dependencies, particularly in sequence prediction tasks. LSTM processes the full data sequence because of its feedback links. The fundamental role of an LSTM model is held by a memory cell, known as the "cell state",

which maintains its state over time. The cell state is represented by the horizontal line that runs through the top of Figure 3, where σ and tanh are the Sigmoid and Tanh layers. h_{t-1} is the output from the last LSTM unit, and h_t is the current output. C_{t-1} is the memory from the last cell unit, and C_t is the new updated memory.

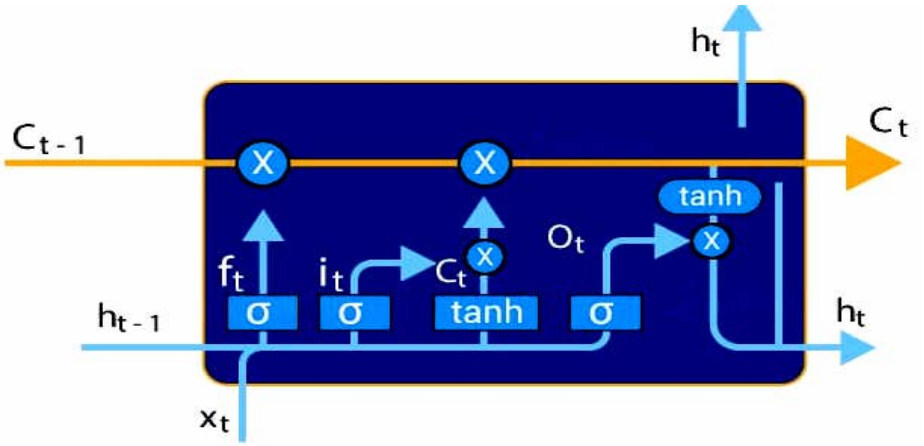

Figure 3. The cell state of the LSTM model.

To compute the output of the proposed method, h_t and C_t are used as shown in the subsequent equations.

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{1}$$

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{2}$$

$$C_t = \tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{3}$$

$$O_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{4}$$

$$C_t = f_t * C_{t-1} + i_t * C_t \tag{5}$$

$$h_t = O_t * \tanh(C_t) \tag{6}$$

where x_t is the input vector to the LSTM unit, f_t is the forget gate, W and b are the weight matrices and bias vector parameters, i_t is the input/update gate's activation vector, O_t is the output gate's activation vector, C_t is the cell input activation vector, and h_t is the hidden state vector, also known as the output vector of the LSTM unit.

In the LSTM-based approach, information is added to or withdrawn from the cell state, which is controlled by the gates. The gates allow information to flow into and out of the cells. Moreover, it contains a pointwise multiplication operation and a Sigmoid neural network layer, which assist the mechanism. Through simple addition or multiplication, LSTM performs minor adjustments to the input that flows through the cell states. This is how LSTM arbitrarily forgets and remembers information, which makes it an improvement over the RNN approach. This decision is made by a Sigmoid layer called the "forget gate layer". It looks at h_{t-1} and x_t and generates an output number between 0 and 1 for each number in the cell state C_{t-1}. A 1 represents "completely keep this", while a 0 represents "completely get rid of this".

The LSTM model is applicable in a wide range of applications, including machine translation, handwriting recognition, image captioning, protein secondary structure prediction, and speech recognition. Table 1 describes the LSTM model utilized in this study to detect DoH traffic tunnels on the CIRA-CIC-DoHBrw-2020 dataset.

Table 1. The description of the LSTM's layers.

Layer (Type)	Output Shape	Param #
Lstm (LSTM)	(None, 1, 40)	11,200
Dropout_3 (Dropout)	(None, 1, 40)	0
Lstm_1 (LSTM)	(None, 1, 40)	12,960
Dropout_4 (Dropout)	(None, 1, 40)	0
Lstm_2 (LSTM)	(None, 40)	12,960
Dropout_5 (Dropout)	(None, 40)	0
Dense_1 (Dense)	(None, 4)	164
Total params:		37,284
Trainable params:		37,284
Non-trainable params:		0

3.2. Machine Learning Approach

The machine learning methods are used to process and extract hidden information from vast data collected in the CIRA-CIC-DoHBrw-2020 dataset. In this work, the baseline machine learning methods were employed to see which algorithm works best with the CIRA-CIC-DoHBrw-2020 dataset after performing feature selection. Specifically, to validate the proposed method, eight machine learning algorithms were used in this study, namely, random forest [29], decision tree [30], K neighbors [31], Linear Discriminant Analysis (LDA) [32], Gaussian naive Bayes [33], Adaboost [34], gradient boosting [35], and logistic regression [36].

4. Dataset Description

The CIRA-CIC-DoHBrw-2020 dataset was created at the University of New Brunswick in Fredericton. DoH is a two-layered method that uses a time-series classification model within an application to detect and identify DoH traffic while also recording legitimate and malignant DoH data alongside non-DoH traffic. To build representative datasets, HTTPS (benign-DoH and non-DoH) and DoH data are created by inspecting the top 10,000 Alexa websites, as well as using browsers and DNS tunneling methods that adopt the DoH standard, respectively [6].

At Layer 1, the statistical characteristics classifier divides the captured data into DoH and non-DoH categories. In Layer 2, time-series classifiers are used to differentiate between benevolent and malicious-DoH traffic. The DoH protocol was used to receive traffic from benign-DoH, malicious-DoH, and non-DoH on four different servers using five other methods and browsers. The four servers that give responses to DoH requests are Cloudflare, Quad9, Google DNS, and AdGuard, while Iodine, DNSCat2, Mozilla Firefox, dns2tcp, and Google Chrome are the five methods and browsers that capture traffic [37].

Specifically, Layer 1 is used to distinguish DoH traffic from non-DoH traffic, and Layer 2 is used to distinguish benign-DoH traffic from malicious-DoH traffic in this situation. The non-DoH traffic generated by the website's HTTPS protocol is collected and labeled. Benign-DoH is benign-DoH traffic produced by the Mozilla Firefox and Google Chrome web browsers using the same technique as non-DoH. The DNS tunneling applications such as dns2tcp, DNSCat2, or Iodine are used to generate malicious-DoH traffic. These methods can submit DNS queries and create encrypted data tunnels for TCP traffic. The DNS queries are then sent to special DoH servers via HTTPS requests, which are encrypted with TLS [3].

Table 2 shows a few samples of the collected data, while Table 3 shows the label count for the Layer 1 vs. Layer 2 classifications.

Table 2. A sample of the CIRA-CIC-DoHBrw-2020 dataset.

	Source IP	Destination IP	Source Port	Destination Port	...	Response Time Skew from Mode	Response Time Coefficient of Variation	Label
0	192.168.20.191	176.103.130.131	50,749	443	...	0.024715	1.174948	DoH
1	192.168.20.191	176.103.130.131	50,749	443	...	−0.075845	1.402382	DoH

Table 3. The label count for Layer 1 vs. Layer 2.

Layer 1	Count	Layer 2	Count
Non-DoH	897,493	Malicious	249,836
DoH	269,643	Benign	19,807

5. Experimental Results and Discussions

This section presents the experiment results and discussion of the proposed method for the detection of DoH traffic tunnels using Deep Learning and eight base learner classifiers for encrypted traffic classification using the CIRA-CIC-DoHBrw-2020 dataset. In addition to binary classification, we also performed multiclass classification. This section presents the experiment results involving binary classification and multiclass classification for detecting DoH traffic tunnels. Section 5.1 briefly introduces the metrics used to evaluate the performance of the proposed method. Section 5.3 shows the results of the binary classification of the stacking classifier on the dataset; in Section 5.4, we present the multiclass classification results of the Deep Learning model, as well as the results of the other ensemble models that we evaluated for comparative comparison.

5.1. Performance Evaluations

In this study, four performance evaluation methods were employed to validate the performance of the proposed method. These are the Accuracy [38], Recall, Precision, and F1-Measure [39]. Accuracy represents the number of accurate predictions divided by the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

Precision is computed by dividing the true positives by the number of total positive predictions as shown in the following equation.

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

Moreover, Recall is the true positives divided by the true positives and false negatives. Recall is defined as follows:

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

The F1-Measure is the harmonic mean of the Precision and Recall and is defined as follows:

$$F1 - Measure = 2\frac{Precision * Recall}{Precision + Recall} \tag{10}$$

where TP, TN, FP, and FN are the True Positive, True Negative, False Positive, and False Negative numbers. These metrics were utilized to demonstrate the accuracy of DoH using the CIRA-CICDoHBrw-2020 dataset, together with time-series machine learning classifiers.

5.2. Feature Selection

The preprocessing was carried out to clean the data instead of directly utilizing them for model training and validation. In the preprocessing step, the missing values were replaced with valid values after performing model-based observations for feature selection in the same attribute. To remove the irrelevant features and reduce the models' computing complexity for the CIRA-CIC-DoHBrw-2020 dataset, the Chi-squared filtering technique was used. Similarly, features with non-numerical values were replaced by a numerical value using the same Chi-squared filtering algorithm. As a result, the proposed method achieved a robust performance by providing the maximum Accuracy, a faster training time, and minimizing the model over-fitting [40].

Using the Chi2 function from sklearn, feature analysis was conducted on the CIRA-CIC-DoHBrw-2020 dataset. The *p*-values were sorted for both layers of the dataset and presented in Table 4. Table 4 shows the ordered list of features from the lowest *p*-value to the highest *p*-value. It is clear that, in Layer 1, there are many zeros, which means that there are many features that directly correlate with the target classification. It has to be noted that the *p*-values with the lowest value imply the most-significant and highly correlated features.

Figure 4 shows the correlation matrix for the information. From this figure, it is clear that some matrix values are −1 and more than 1; hence, some information is correlated with other information. for example, DestinationPort is related to "PacketTimeVariance", "PacketTimeStandardDeviation", "PacketTimeMean", and "PacketTimeMedian". "PacketTimeVariance", "PacketTimeStandardDeviation", "PacketTimeMean", and "PacketTimeMedian" are correlated with each other.

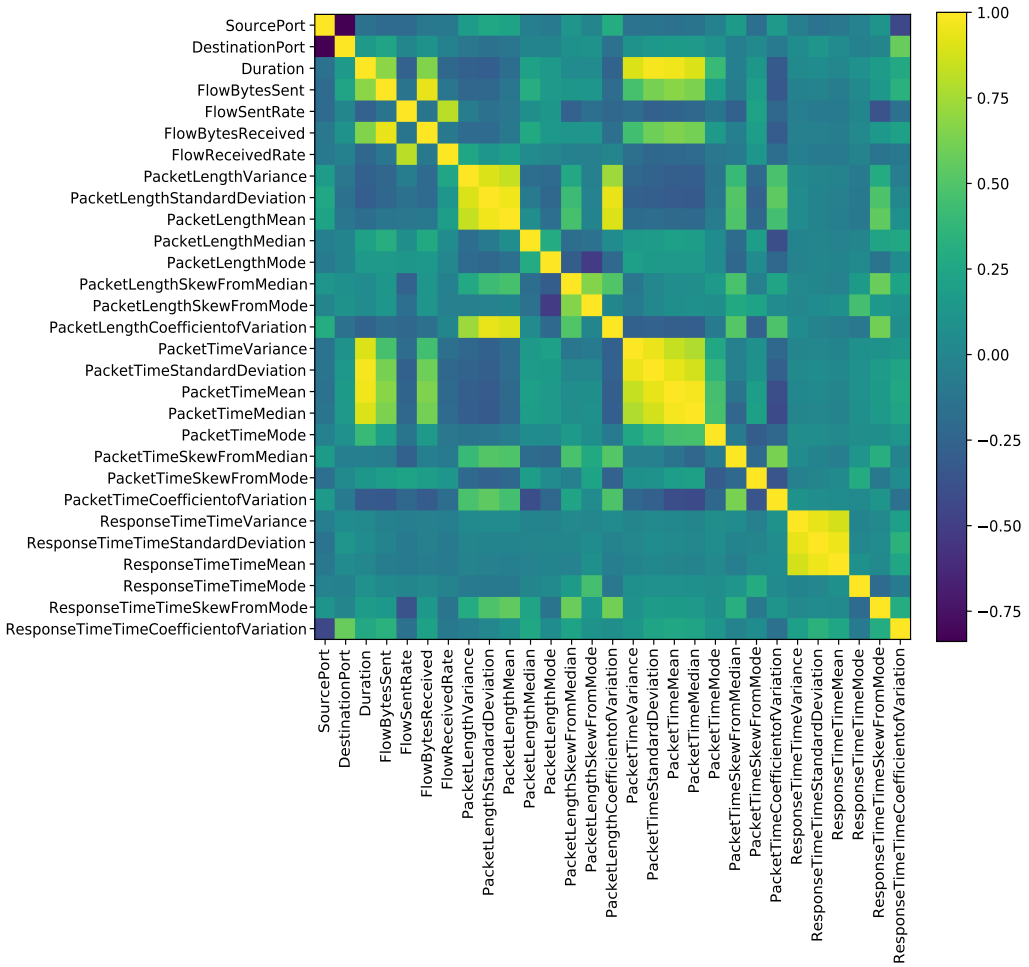

Figure 4. The correlation matrix for the features of the dataset.

Table 4. *p*-values in ascending order (the lowest value implies the most-significant and highly correlated features).

Layer 2	***p*-Value**	**Layer 1**	***p*-Value**
PacketLengthStandardDeviation	0.0	Duration	0.0
PacketLengthCoefficientofVariation	0.0	ResponseTimeTimeSkewFromMedian	0.0
FlowReceivedRate	0.0	ResponseTimeTimeMode	0.0
PacketLengthMean	0.0	ResponseTimeTimeMedian	0.0
Duration	0.0	ResponseTimeTimeMean	0.0
PacketTimeSkewFromMedian	0.0	PacketTimeSkewFromMedian	0.0
FlowSentRate	0.0	PacketTimeMode	0.0
PacketLengthVariance	0.0	PacketTimeMedian	0.0
PacketTimeMean	0.0	PacketTimeMean	0.0
PacketTimeStandardDeviation	0.0	ResponseTimeTimeSkewFromMode	0.0
ResponseTimeTimeMedian	0.0	PacketTimeVariance	0.0
PacketTimeMedian	0.0	PacketLengthCoefficientofVariation	0.0
ResponseTimeTimeSkewFromMode	0.0	PacketTimeStandardDeviation	0.0
ResponseTimeTimeMean	0.0	PacketLengthMode	0.0
ResponseTimeTimeMode	0.0	PacketLengthMedian	0.0
PacketTimeCoefficientofVariation	0.0	PacketLengthMean	0.0
ResponseTimeTimeSkewFromMedian	0.0	FlowBytesSent	0.0
PacketTimeMode	0.0	ResponseTimeTimeCoefficientofVariation	0.0
FlowBytesSent	0.0	PacketLengthStandardDeviation	0.0
FlowBytesReceived	0.0	PacketLengthVariance	0.0
PacketLengthMode	0.0	PacketTimeCoefficientofVariation	0.0
ResponseTimeTimeCoefficientofVariation	0.0	FlowReceivedRate	0.0
PacketLengthSkewFromMedian	0.0	ResponseTimeTimeStandardDeviation	0.0
PacketTimeVariance	0.000008364485	PacketLengthSkewFromMode	0.0
PacketLengthMedian	0.00005997378	FlowBytesReceived	0.0
PacketTimeSkewFromMode	0.00006506026	PacketLengthSkewFromMedian	0.001868
ResponseTimeTimeStandardDeviation	0.01694301	FlowSentRate	0.505078
ResponseTimeTimeVariance	0.03453484	ResponseTimeTimeVariance	0.552312
PacketLengthSkewFromMode	0.9945070	PacketTimeSkewFromMode	0.642348

5.3. Binary Class Classification

The stacking classifier (comprising random forest and decision tree as the base classifier) was used to determine the classes in either of the classes to which they belong in a binary class classification problem. Table 5 depicts the number of instances of each class within the dataset. In the binary classification of the "DoH" and "Non-DoH" classes, the stacking classifier achieved a classification Accuracy of 99.76 percent, a 99.76 percent Precision, and a 99.76 percent Recall, respectively. Moreover, the confusion matrix depicts correctly and wrongly classified samples in a given classification problem. In this study, a confusion matrix was applied to visualize the performance of the classifiers. Figure 5 depicts the confusion matrix for the binary classification of the "DoH" and "Non-DoH" classes using the stacking classifier. For instance, from the experimental results shown in

Figure 5, it is evident that the False Negatives (FNs) were 611, while the False Positives (FPs) were 195.

Table 5. The number of instances of each class label within the dataset.

	DoH	Non-DoH	Benign	Malicious
Before	269,299	889,809	19,746	249,553

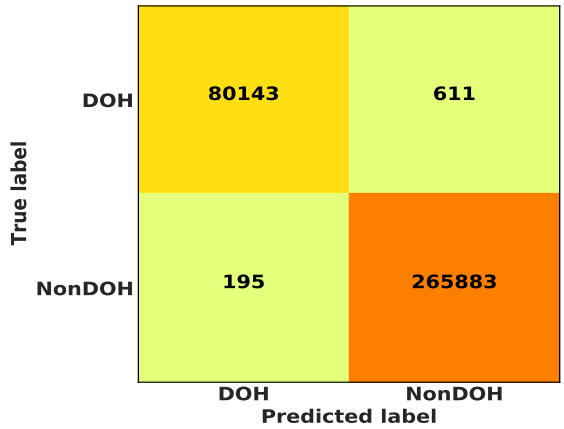

Figure 5. The confusion matrix for binary classification of "DoH" and "Non-DoH" using the stacking classifier.

For the binary classification of the "Benign" and "Malicious" classes, the stacking classifier achieved an Accuracy of 99.99 percent, 99.99 percent Precision, and 99.99 percent Recall. Figure 6 shows the confusion matrix for the binary classification of the "Benign" and "Malicious" classes using the stacking classifier, from which it is clear that the False Negatives (FN-) were 7, while the False Positives (FP-) were 3 samples.

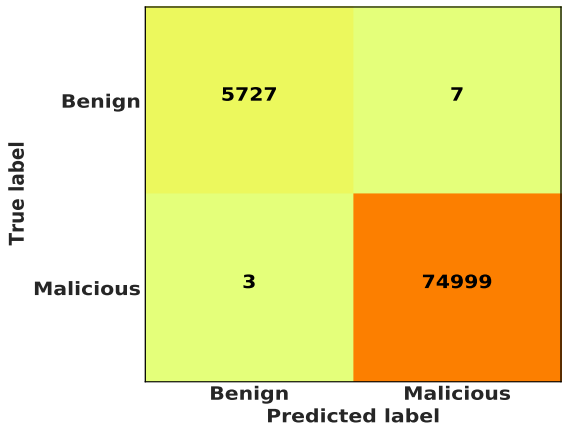

Figure 6. The confusion matrix for binary classification of "Benign" and "Malicious" using the stacking classifier.

5.4. Multi-Class Classification

This section briefly explains the experimental results of the multiclass classification using the LSTM and ensemble learning algorithms. The experimentation was carried out using one (1) feature, and the results are presented in Table 6. Accordingly, the gradient boosting model achieved maximum experimental values of 0.9071, 0.9071, 0.9087, and 0.9118 for the Accuracy, Recall, Precision, and AUC, respectively. It was observed that the Precision from the gradient boosting model achieved the best experimental results compared to the decision tree and linear discriminant analysis. Hence, the gradient boosting model was able to find a good fit for a small portion of the anomalous patterns in the data.

Table 6. Experimental result using the top feature.

Classifier	Accuracy	Recall	Precision	AUC
Random Forest	0.8665 ± 0.0269	0.8665 ± 0.0269	0.8647 ± 0.0344	0.8637 ± 0.0683
Decision Tree	0.8598 ± 0.0264	0.8598 ± 0.0264	0.8596 ± 0.0344	0.7993 ± 0.0636
K-Nearest Neighbors	0.9021 ± 0.0309	0.9021 ± 0.0309	0.9001 ± 0.0313	0.8711 ± 0.0686
Linear Discriminant Analysis	0.8069 ± 0.0234	0.8069 ± 0.0234	0.7952 ± 0.0301	0.8463 ± 0.0473
Gaussian Naive Bayes	0.8059 ± 0.0240	0.8059 ± 0.0240	0.7940 ± 0.0304	0.8463 ± 0.0473
Adaboost	0.9004 ± 0.0307	0.9004 ± 0.0307	0.9035 ± 0.0250	0.9006 ± 0.0655
Gradient Boost	0.9071 ± 0.0318	0.9071 ± 0.0318	0.9087 ± 0.0277	0.9118 ± 0.0592
Logistic Regression	0.8074 ± 0.0227	0.8074 ± 0.0227	0.7969 ± 0.0303	0.8463 ± 0.0473
Stacking Model	0.8485 ± 0.0207	0.8485 ± 0.0207	0.8485 ± 0.0207	-

5.4.1. The LSTM Model

The experiment was conducted with the LSTM model, and the features used for the experimentation are presented in Table 4. From the experimental results shown in Table 7, the LSTM model achieved 0.2797, 0.8082, 0.2783, and 0.8092 for the loss, Accuracy, validation loss, and validation Accuracy. Hence, it was observed that the LSTM model performed less accurately compared with the gradient boosting model, in the sense that the LSTM model did not fit well with a small portion of the anomalous patterns in the data.

Table 7. Experimental result using the LSTM model.

Loss	Accuracy	Val Loss	Val Accuracy
0.2797	0.8082	0.2783	0.8092

Figure 7 depicts the confusion matrix of the results from running the LSTM model on the testing set of the SMOTE data. Figure 8 shows the Accuracy of the results of the LSTM model on the testing set of SMOTE data. Figure 9 shows the loss of the results from the LSTM model on the testing set of the SMOTE data, and Figure 10 shows the ROC curve of the classes for the LSTM model on the same dataset.

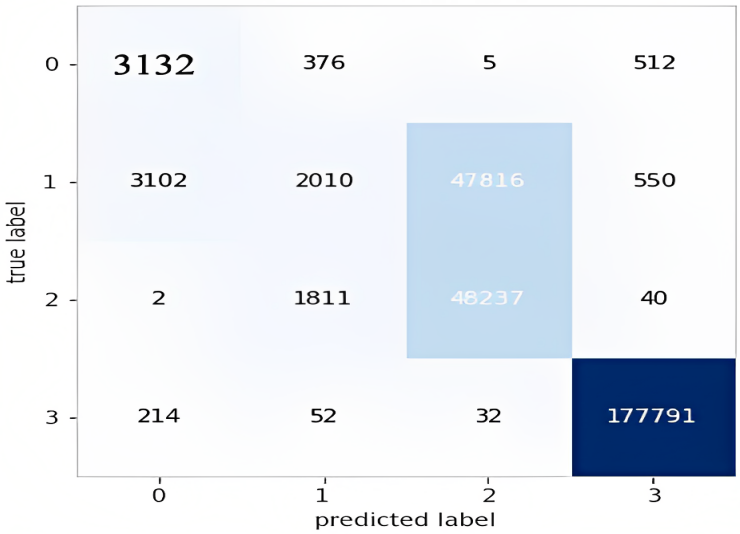

Figure 7. The confusion matrix for the LSTM model.

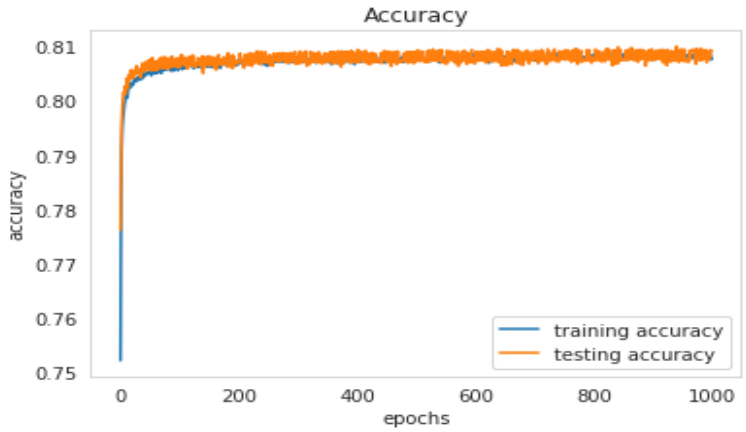

Figure 8. The Accuracy using the LSTM model.

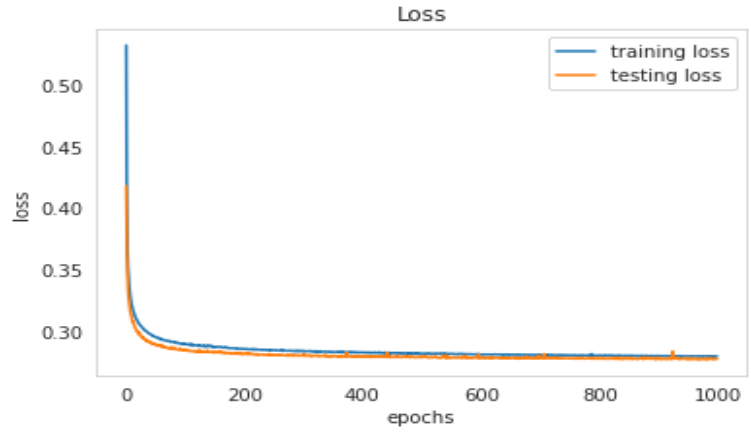

Figure 9. The loss using the LSTM model.

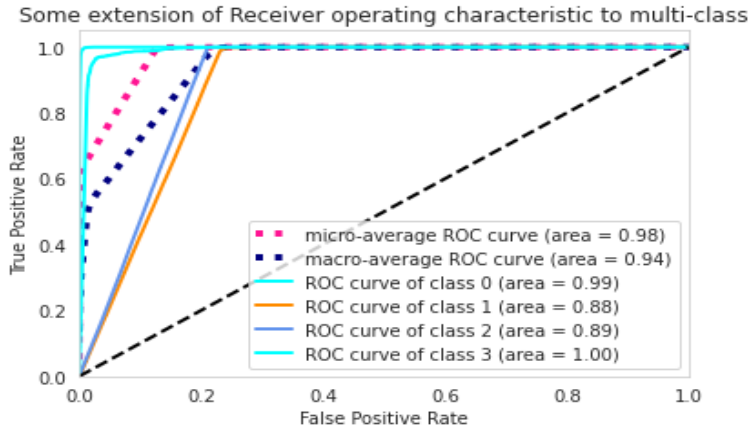

Figure 10. The ROC curve for the LSTM model.

5.4.2. Data Balancing and SMOTE

Due to the nature of the dataset, there is a small amount of anomalous time periods generated. Therefore, the dataset needs to be balanced and the data integrity maintained for the training and testing tasks. The famous Synthetic Minority Oversampling Technique (SMOTE) is a random oversampling technique with replacement [41]. SMOTE is used to obtain a balanced dataset by oversampling the minority classes. Next, both base classifiers were trained on the SMOTE dataset and validated on the hold-out dataset. Accordingly, Table 8 shows the numbers of each class label before and after applying the SMOTE method to generate a synthetic dataset.

Table 8. Experimental results before and after applying SMOTE to generate a synthetic dataset.

	DoH	Non-DoH	Benign	Malicious
Before	269,299	889,809	19,746	249,553
After	889,809	889,809	889,809	889,809

The combination of the SMOTE oversampling method and the LSTM classifier was applied to the imbalanced dataset to compare and measure the performances of the proposed method. To compare the performance of the combination of the SMOTE method and LSTM classifier, traditional classification metrics such as the Accuracy, Precision, and Recall, were used. Moreover, there are other widely used metrics used to measure the classifier's performance, such as the F-Measure, the Area Under the Curve (AUC), and the Receiver Operating Characteristic (ROC) curve. The ROC curve is a standard technique for summarizing the performance of the classifier over a range of trade-offs between true positive and false positive error rates.

The experiment was conducted with the SMOTE oversampling method and the LSTM classifier with all metrics as presented in Table 9. Accordingly, the LSTM model achieved values of 0.71612, 0.699716, 0.71612, 0.681352, and 0.714445 for the Accuracy, Precision, Recall, F1, and F_b, respectively. Figure 11 shows the confusion matrix of the experimental results from running the LSTM model on the testing set of the SMOTE dataset.

Table 9. The results of the SMOTE oversampling method and the LSTM classifier.

Accuracy	Precision	Recall	F1	F_b
0.71612	0.699716	0.71612	0.681352	0.714445

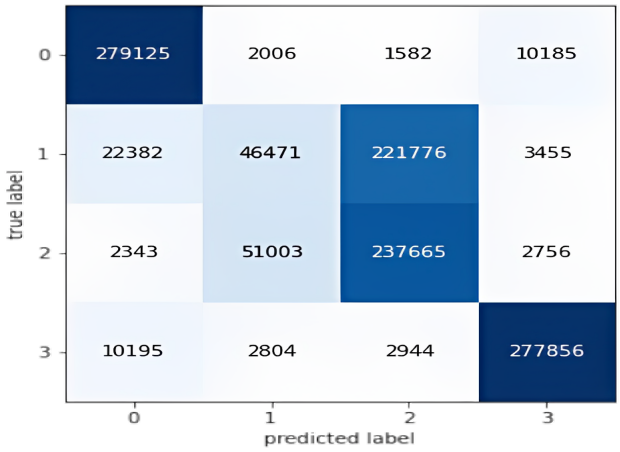

Figure 11. The confusion matrix for the SMOTE oversampling method and the LSTM classifier.

Figure 12 shows the Accuracy of the results from the LSTM model on the testing set of the SMOTE data.

Figure 13 shows the loss of the results from running the LSTM model on the testing set of the SMOTE data.

Figure 14 shows the ROC curve of the classes for the LSTM model on the testing set of the SMOTE data. The Area Under the Curve (AUC) ranges in value from 0 to 1. A model that has predictions that are 100% wrong has an AUC of 0.0, and a model that has predictions that are 100% correct has an AUC of 1.0. The majority of Classes 0 and 3 had an ROC curve close to 100%.

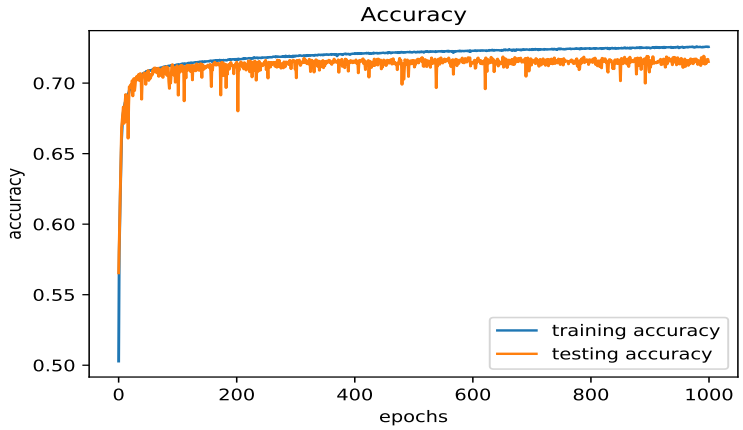

Figure 12. The Accuracy for the SMOTE oversampling method and the LSTM classifier.

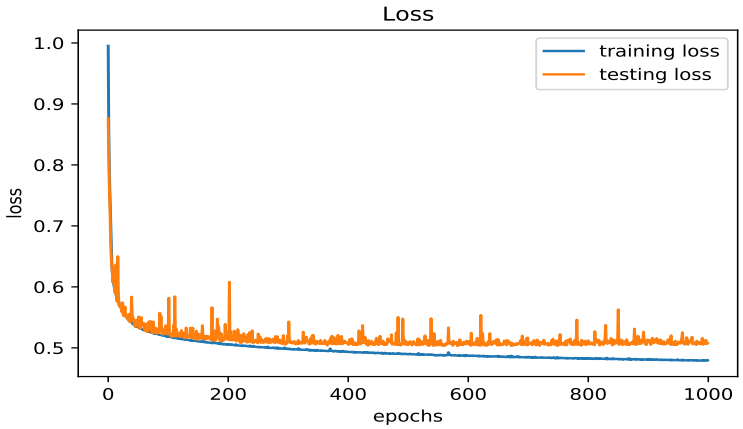

Figure 13. The loss for the SMOTE oversampling method and the LSTM classifier.

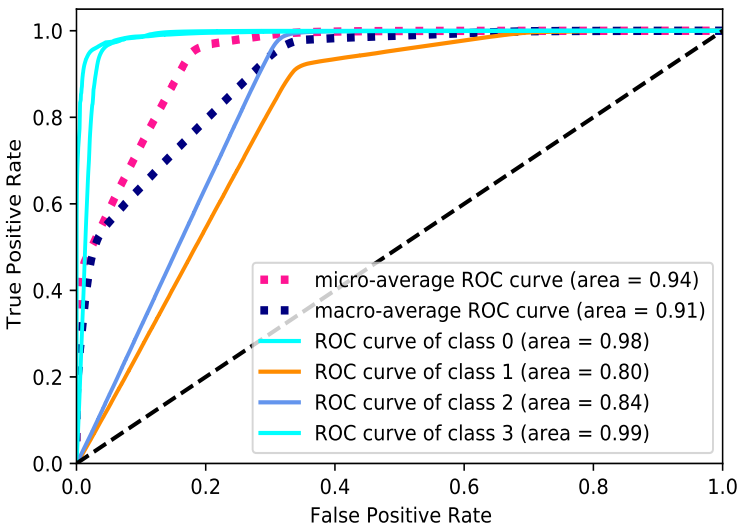

Figure 14. The ROC curve for the SMOTE oversampling method and the LSTM classifier.

The experimental results showed that the stacking method significantly improved the detection rate by reducing the false positive rate. To show the efficacy of the stacking model, a comparison of it with the state-of-art models was performed. Table 10 shows the comparative analysis of the proposed method and previous works. When compared to

the other models, it is clearly obvious that the stacking model achieved a higher level of Accuracy.

Table 10. The results of the comparison of the proposed model and previous works.

Model	Accuracy	Precision	Recall	F1
Stacking	99.99	99.99	99.99	99.99
XGBoost [42]	97.6	97.6	97.6	97.6
XGBoost [24]	99.8	99.7	99.99	99.8

6. Conclusions and Future Work

DNS over HTTPS (DoH) plays a vital role in DNS encryption by blocking DNS resolutions from active attackers to protect user privacy. The use of encrypted communications is growing in the DNS protocol. The DNS lacks in-built security mechanisms; thus, DoH becomes a solution for data security and privacy. Studies have introduced various forms of implementations of DoH and have shown how to use them to protect user privacy. In this study, a machine-learning-based predictive model was introduced for anomaly detection in DoH by data encryption. The proposed predictive method was created to catch trend changes in encrypted traffic data at an early stage. The experimental results revealed that a single ML algorithm is unable to efficiently control the time required to detect malicious DNS requests while maintaining a high level of Accuracy. Rather, the experimental findings demonstrated that the detection accuracy of the stacking algorithms achieved a maximum performance of 99.99% in binary class datasets. In the case of multiclass classification, the gradient boosting model scored the maximum performance of 90.71%, 90.71%, 90.87%, and 91.18% for the Accuracy, Recall, Precision, and AUC, respectively. The micro average ROC curve for the LSTM model scored 98% on the CIC-DoHBrw2020 dataset in detecting tunneling in DNS traffic. Moreover, the LSTM and machine learning models predicted the future DoH labels using different features and time information. Still, there are many research gaps worthy of further consideration. For instance, during the training time, the machine learning methods' performance depends on the dataset size, the quality of the data, and the sampling rate. Therefore, improving the performance of machine learning models on small-sized datasets shall be examined. Future researchers can investigate interesting topics, including malicious-DoH, the use of autoencoders for anomaly detection, and the detection of malicious-DoH using unsupervised learning techniques.

Funding: This research received no external funding.

Data Availability Statement: The datasets can be downloaded from the following link: http://205.174.165.80/CICDataset/DoHBrw-2020/Dataset/Total-CSVs.zip

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

References

1. Böttger, T.; Cuadrado, F.; Antichi, G.; Fernandes, E.L.; Tyson, G.; Castro, I.; Uhlig, S. An Empirical Study of the Cost of DNS-over-HTTPS. In Proceedings of the Internet Measurement Conference, Amsterdam, The Netherlands, 21–23 October 2019; pp. 15–21.
2. Borgolte, K.; Chattopadhyay, T.; Feamster, N.; Kshirsagar, M.; Holland, J.; Hounsel, A.; Schmitt, P. How DNS over HTTPS is reshaping privacy, performance, and policy in the internet ecosystem. In Proceedings of the TPRC47: The 47th Research Conference on Communication, Information and Internet Policy, Washington, DC, USA, 20–21 September 2019.
3. Jafar, M.T.; Al-Fawa'reh, M.; Al-Hrahsheh, Z.; Jafar, S.T. Analysis and Investigation of Malicious DNS Queries Using CIRA-CIC-DoHBrw-2020 Dataset. *Manch. J. Artif. Intell. Appl. Sci. (MJAIAS)* **2021**, *2*, 65–70.
4. Bumanglag, K.; Kettani, H. On the Impact of DNS Over HTTPS Paradigm on Cyber Systems. In Proceedings of the 2020 3rd International Conference on Information and Computer Technologies (ICICT), San Jose, CA, USA, 9–12 March 2020; pp. 494–499.
5. Siby, S.; Juarez, M.; Vallina-Rodriguez, N.; Troncoso, C. DNS Privacy not so private: The traffic analysis perspective. In Proceedings of the 11th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2018), Barcelona, Spain, 27 July 2018.

6. Montazeri Shatoori, M.; Davidson, L.; Kaur, G.; Lashkari, A.H. Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic. In Proceedings of the 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Calgary, AB, Canada, 17–22 August 2020; pp. 63–70.
7. Munteanu, D.; Bejan, C.; Munteanu, N.; Zamfir, C.; Vasić, M.; Petrea, S.M.; Cristea, D. Deep-Learning-Based System for Assisting People with Alzheimer's Disease. *Electronics* **2022**, *11*, 3229. [CrossRef]
8. Amaratunga, T. Building Your First Deep Learning Model. In *Deep Learning on Windows*; Apress: Berkeley, CA, USA, 2020; pp. 67–100.
9. Gad, I.; Hosahalli, D.; Manjunatha, B.R.; Ghoneim, O.A. A robust Deep Learning model for missing value imputation in big NCDC dataset. *Iran J. Comput. Sci.* **2020**, *4*, 67–84. [CrossRef]
10. Hosahalli, D.; Gad, I. A Generic Approach of Filling Missing Values in NCDC Weather Stations Data. In Proceedings of the 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 19–22 September 2018.
11. Mohammadi, M.; Al-Fuqaha, A.; Sorour, S.; Guizani, M. Deep learning for IoT big data and streaming analytics: A survey. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2923–2960. [CrossRef]
12. Vekshin, D.; Hynek, K.; Cejka, T. DoH insight: Detecting dns over https by machine learning. In Proceedings of the 15th International Conference on Availability, Reliability, and Security, Virtual, 25–28 August 2020; pp. 1–8.
13. Hounsel, A.; Borgolte, K.; Schmitt, P.; Holland, J.; Feamster, N. Analyzing the costs (and benefits) of DNS, DoT, and DoH for the modern web. In Proceedings of the Applied Networking Research Workshop, Montreal, QC, Canada, 22 July 2019; pp. 20–22.
14. Bushart, J.; Rossow, C. Padding Ain't Enough: Assessing the Privacy Guarantees of Encrypted {DNS}. In Proceedings of the 10th USENIX Workshop on Free and Open Communications on the Internet (FOCI 20), Online, 11 August 2020.
15. Lu, C.; Liu, B.; Li, Z.; Hao, S.; Duan, H.; Zhang, M.; Leng, C.; Liu, Y.; Zhang, Z.; Wu, J. An end-to-end, large-scale measurement of dns-over-encryption: How far have we come? In Proceedings of the Internet Measurement Conference, Amsterdam, The Netherlands, 21–23 October 2019; pp. 22–35.
16. Singanamalla, S.; Chunhapanya, S.; Vavruša, M.; Verma, T.; Wu, P.; Fayed, M.; Heimerl, K.; Sullivan, N.; Wood, C. Oblivious DNS over HTTPS (ODoH): A Practical Privacy Enhancement to DNS. *arXiv* **2020**, arXiv:2011.10121.
17. Deccio, C.; Davis, J. DNS privacy in practice and preparation. In Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies, Orlando, FL, USA, 9–12 December 2019; pp. 138–143.
18. Singh, S.K.; Roy, P.K. Detecting Malicious DNS over HTTPS Traffic Using Machine Learning. In Proceedings of the 2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT), Virtual, 20–21 December 2020; pp. 1–6.
19. Hounsel, A.; Borgolte, K.; Schmitt, P.; Holland, J.; Feamster, N. Comparing the effects of dns, dot, and doh on web performance. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 562–572.
20. López Romera, C. DNS Over HTTPS Traffic Analysis and Detection. Master's Thesis, Universitat Oberta de Catalunya, Barcelona, Spain, 2020.
21. Palau, F.; Catania, C.; Guerra, J.; Garcia, S.; Rigaki, M. DNS tunneling: A Deep Learning based lexicographical detection approach. *arXiv* **2020**, arXiv:2006.06122.
22. Houser, R.; Li, Z.; Cotton, C.; Wang, H. An investigation on information leakage of DNS over TLS. In Proceedings of the 15th International Conference on Emerging Networking Experiments and Technologies, Orlando, FL, USA, 9–12 December 2019; pp. 123–137.
23. Huang, Q.; Chang, D.; Li, Z. A Comprehensive Study of DNS-over-HTTPS Downgrade Attack. In Proceedings of the 10th USENIX Workshop on Free and Open Communications on the Internet (FOCI 20), Online, 11 August 2020.
24. Banadaki, Y.M. Detecting Malicious DNS over HTTPS Traffic in Domain Name System using Machine Learning Classifiers. *J. Comput. Sci. Appl.* **2020**, *8*, 46–55. [CrossRef]
25. de Vries, L. Detection of DoH Tunnelling: Comparing Supervised with Unsupervised Learning. Master Thesis, University of Twente, Enschede, The Netherlands, 2021.
26. Hayashi, Y. Emerging Trends in Deep Learning for Credit Scoring: A Review. *Electronics* **2022**, *11*, 3181. [CrossRef]
27. Doreswamy.; Gad, I.; Manjunatha, B.R. Multi-label Classification of Big NCDC Weather Data Using Deep Learning Model. In *Soft Computing Systems*; Springer: Singapore, 2018; pp. 232–241.
28. Tang, J.; Yang, R.; Yuan, G.; Mao, Y. Time-Series Deep Learning Models for Reservoir Scheduling Problems Based on LSTM and Wavelet Transformation. *Electronics* **2022**, *11*, 3222. [CrossRef]
29. Mantas, C.J.; Castellano, J.G.; Moral-García, S.; Abellán, J. A comparison of random forest based algorithms: Random credal random forest versus oblique random forest. *Soft Comput.* **2018**, *23*, 10739–10754. [CrossRef]
30. Bhukya, D.P.; Ramachandram, S. Decision Tree Induction: An Approach for Data Classification Using AVL-Tree. *Int. J. Comput. Electr. Eng.* **2010**, *2*, 660–665. [CrossRef]
31. Doreswamy.; Hooshmand, M.K.; Gad, I. Feature selection approach using ensemble learning for network anomaly detection. *CAAI Trans. Intell. Technol.* **2020**, *5*, 283–293. [CrossRef]
32. Gładyszewska-Fiedoruk, K.; Sulewska, M.J. Thermal Comfort Evaluation Using Linear Discriminant Analysis (LDA) and Artificial Neural Networks (ANNs). *Energies* **2020**, *13*, 538. [CrossRef]

33. Sa, S. Comparative Study of Naive Bayes, Gaussian Naive Bayes Classifier and Decision Tree Algorithms for Prediction of Heart Diseases. *Int. J. Res. Appl. Sci. Eng. Technol.* **2021**, *9*, 475–486. [CrossRef]
34. Shahraki, A.; Abbasi, M.; Haugen, Ø. Boosting algorithms for network intrusion detection: A comparative evaluation of Real AdaBoost, Gentle AdaBoost and Modest AdaBoost. *Eng. Appl. Artif. Intell.* **2020**, *94*, 103770. [CrossRef]
35. Pingalkar, A.S. Prediction of Solar Eclipses using Extreme Gradient Boost Algorithm. *Int. J. Res. Appl. Sci. Eng. Technol.* **2020**, *8*, 1353–1357. [CrossRef]
36. Nayebi, H. Logistic Regression Analysis. In *Advanced Statistics for Testing Assumed Casual Relationships*; Springer International Publishing: Zurich, Switwerland, 2020; pp. 79–109.
37. Schulze, J.P.; Sperl, P.; Böttinger, K. Double-Adversarial Activation Anomaly Detection: Adversarial Autoencoders are Anomaly Generators. *arXiv* **2021**, arXiv:2101.04645.
38. Bramer, M. Estimating the Predictive Accuracy of a Classifier. In *Principles of Data Mining*; Springer: London, UK, 2020; pp. 79–92.
39. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
40. Wei, G.; Zhao, J.; Feng, Y.; He, A.; Yu, J. A novel hybrid feature selection method based on dynamic feature importance. *Appl. Soft Comput.* **2020**, *93*, 106337. [CrossRef]
41. Majzoub, H.A.; Elgedawy, I. AB-SMOTE: An Affinitive Borderline SMOTE Approach for Imbalanced Data Binary Classification. *Int. J. Mach. Learn. Comput.* **2020**, *10*, 31–37. [CrossRef]
42. Silveira, M.R.; Cansian, A.M.; Kobayashi, H.K. Detection of Malicious Domains Using Passive DNS with XGBoost. In Proceedings of the 2020 IEEE International Conference on Intelligence and Security Informatics (ISI), Arlington, VA, USA, 9–11 November 2020. [CrossRef]