International Journal of Enhanced Research in Management & Computer Applications,
IJERMCA, ISSN: 2319-7471, Volume 14 Issue 1, January 2025

Leveraging Artificial Intelligence to Classify DNS Tunneling Tools in Malicious DoH Traffic

Rafa Alaiat Alenezi[1], Marfua Alayyat Alanazi[2]

[1]Technical and Vocational Training Corporation, TVTC, Department of Computer and Information Technology
Hafer Albatin Technical College, TVTC, Hafer Albatin, KSA
[2]Change Management, Technical and Vocational Training Corporation, Riyadh, KSA

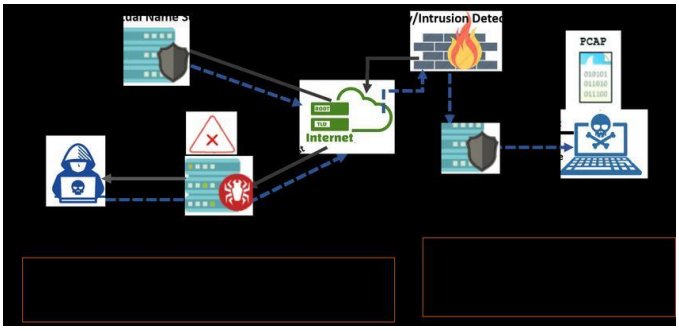
ABSTRACT

Cyber attackers are constantly devising new techniques to breach security systems and compromise computing infrastructures. Over the last ten years, there has been a significant rise in attacks targeting DNS systems, which manage information about domain names and their associated IP addresses (zone files). Combating these threats demands innovative approaches that incorporate Artificial Intelligence (AI) to analyze attack behaviors and strategies effectively. Detailed examination of the digital environment is critical for developing appropriate countermeasures against cyber threats. Although DNS over HTTPS (DoH) was originally introduced to defend against threats like DNS spoofing and tunneling, it has also created novel privacy and security challenges. This paper examines recent research leveraging AI and machine learning models to tackle DNS tunneling and DoH-related security issues. Specifically, it investigates the ability of AI classifiers to accurately distinguish between different types of DNS tunneling attacks using commonly adopted ML techniques.The study employs the CIRA-CIC-DoHBrw-2020 dataset for experimentation. Findings indicate that AI-based classifiers are highly effective in identifying DNS tunneling within DoH traffic. Model performance is evaluated using precision, recall, F1-score, accuracy, and confusion matrix metrics, confirming that AI-driven solutions can significantly enhance the detection and prevention of DNS-related cyber threats.

***Index Terms*—DNS tunneling, Malicious DoH traffic, Machine Learning**

INTRODUCTION

The rapid growth of internet usage, driven by technological advancements and the digital transformation of businesses and government entities, has provided cyber attackers with new opportunities to target ICT infrastructures. Over the past decade, websites have faced increasing threats, as adversaries intercept and maliciously redirect Domain Name System (DNS) traffic. Attackers exploit techniques such as DNS spoofing (also known as DNS cache poisoning) to manipulate DNS records and redirect users to fraudulent websites that closely resemble the intended destinations [1]. These deceptive sites often prompt users to input login credentials and sensitive information, which can then be leveraged for further cybercrimes. Additionally, such malicious sites frequently deploy worms and viruses, granting attackers prolonged access to users' devices and their stored information. While DNS tunneling can serve legitimate purposes, it is commonly misused by threat actors to establish covert communication channels between internal network devices and external malicious servers [2].



This enables attackers to bypass firewall protections, propagate malicious commands, and exfiltrate sensitive data [3].
To address these threats, cybersecurity specialists have introduced DNS over HTTPS (DoH), a protocol that encrypts DNS traffic by routing queries through a Hypertext Transfer Protocol Secure (HTTPS) session [4]. DoH enhances privacy and security by preventing eavesdropping, DNS manipulation, and Man-in-the-Middle (MitM) attacks [5]. Major web browsers, including Google Chrome, Mozilla Firefox, Apple Safari, and Microsoft Edge, have integrated DoH to bolster online data protection and privacy. By encrypting DNS queries, DoH increasingly replaces traditional DNS mechanisms for domain resolution [6]. Despite its advantages, critics warn that DoH encryption can introduce new vulnerabilities, potentially increasing the risk of spoofing, malware propagation, and unauthorized data exfiltration [7]. Attackers may exploit encrypted DoH traffic as a backdoor to steal sensitive information and coordinate malware through command-and-control (CC) communications [7]. Since DoH traffic is encrypted, traditional cybersecurity tools relying on passive DNS monitoring may fail to detect harmful activities [8].

To mitigate these challenges, security experts are leveraging Artificial Intelligence (AI) and Machine Learning (ML) techniques alongside threat intelligence analytics on DNS infrastructure. However, DoH traffic can bypass these defenses, exposing organizations to attacks that circumvent conventional DNS-based security filters [9]. In the DoH architecture, servers operate at the application layer, bypassing operating systemlevel settings [10]. As a result, many security tools, policies, and monitoring mechanisms deployed by system administrators and enterprise security teams become less effective [10].

Currently, a few DNS tunneling tools, including dns2tcp, DNSCat, and Iodine, are used to generate malicious DoH traffic. These tools encapsulate TCP traffic within DNS queries and transmit it via TLS-encrypted HTTPS requests to specialized DoH servers [11]. Consequently, a central challenge in DoH security is identifying the DNS tunneling tools employed in malicious traffic.

This study investigates a standard dataset of DoH-related cyber attacks. It focuses on enhancing AI and ML models to accurately predict and classify the DNS tunneling tools—such as dns2tcp, DNSCat2, and Iodine—used in malicious DoH traffic. In the dataset, the classes represent different DNS tunneling tools, while the features capture factors influencing the outcomes. Detailed information on the dataset is provided in the Data Description section.

The paper is organized into five sections following the introduction. Section II discusses related work. Section III details the methodology, including the DNS tunneling tools and algorithms applied for classification using ML models. Section IV presents the experiments and results obtained from various ML models. Finally, Section V summarizes the findings and conclusions of the study.

RELATED WORK

Various researchers and cybersecurity experts have highlighted Artificial Intelligence (AI), particularly machine learning (ML), as an effective approach for addressing DNS tunneling and DoH security challenges.

In [6], the study assessed five widely used ML techniques to identify the most accurate classifiers for DoH traffic, demonstrating that DoH detection accuracy exceeded 99.9

The work in [7] focused on predicting and classifying malicious versus benign DNS requests within DoH traffic using several ML classifiers. The study employed algorithms including (i) Naive Bayes (NB), (ii) Logistic Regression Classifier (LRC), (iii) Random Forest Classifier (RFC), (iv) K-Nearest Neighbor Classifier (KNC), and (v) Gradient Boosting Classifier (GBC) to detect malicious DNS activity. By leveraging a range of features, the study designed a robust model [7], revealing that Gradient Boosting Trees and Random Forest classifiers effectively detected most malicious activities, confirming that AI-driven methods are a strong solution for DNS attacks on DoH traffic.

According to [12], the diversity of ML techniques—including Support Vector Machine (SVM), Decision Tree (DT), Na¨ıve Bayes (NB), and K-Nearest Neighbor (KNN)—creates the challenge of selecting the most suitable classifier for detecting DNS tunneling. The study used a benchmark DNS tunneling dataset for comparison, and results showed that SVM achieved superior performance, reaching an f-measure of 83

In [13], the authors proposed a systematic two-layer AI approach to identify DoH traffic and distinguish legitimate from malicious DoH traffic using six ML algorithms. Experiments indicated that LightGBM and XGBoost outperformed other algorithms in both layers [13].

Shende and Thorat [14] demonstrated that deep learning techniques are well-suited for network intrusion detection, suggesting the use of LSTM to classify and detect attacks. Their models, trained on the NSL-KDD dataset, achieved accuracies of approximately 99.25

In [15], HaddadPajouh et al. explored using RNNs for IoT malware detection by analyzing ARM-based IoT operation codes (OpCodes). The system, trained on 281 malware samples and 270 benign samples, was evaluated using 100 new IoT viruses, achieving an accuracy of 98.18

Wang et al. [16] introduced a weighted recurrent neural network (W-RNN) to extract semantic features from text, representing vocabulary using word vectors, which are then processed by RNN to generate a text representation vector. This approach was applied for classifying news texts, outperforming conventional methods in Precision, Recall, F1, and loss metrics, highlighting RNN's suitability for darknet traffic classification.

Yang et al. [17] proposed a convolutional gated recurrent unit (GRU) neural network to classify malicious URLs using text features. The GRU effectively captured temporal features, producing a multi-category classification model with precision above 99.66

In [18], Random Forest (RF) was employed to analyze IP header information from darknet datasets. The RF model achieved high recall and precision, demonstrating its efficacy in identifying malicious IoT traffic.

K. Ramos et al. [19] applied the Decision Tree (DT) ML algorithm to classify heterogeneous botnet traffic, showing increased precision in classification. Similarly, S. Bagui et al. [20] used Gradient Boosting (GB) to classify VPN traffic, addressing security challenges in internet applications, with strong predictive performance across accuracy, precision, sensitivity, and specificity metrics.

In [21], the K-Neighbor ML algorithm was utilized to classify various network traffic types, including email, file transfer, VoIP, streaming, and instant messaging. The study reported a classification accuracy of 90.87

R. Kumar et al. [22] employed XGBoost to classify malware in Industry 4.0 and digital ecosystems using the Ember dataset, achieving a classification accuracy of 98.5

In the present paper, eight AI-driven ML methods were applied to classify and predict DNS tunneling used by malicious DoH traffic. The study further investigates the performance and accuracy of these models, reinforcing the value of AIbased approaches in enhancing DNS and DoH security.

METHODOLOGY

This section presents a comprehensive overview of DNS tunneling tools and explores the application of AI-driven machine learning models in addressing these security challenges.

A. DNS Tunneling Tools

Several AI-assisted DNS tunneling tools can be leveraged to detect malicious DoH traffic in the dataset:

1) Dns2tcp tool is an application that enables TCP traffic to be transmitted through DNS queries. It consists of two main components: the *Dns2tcpd*, typically deployed on a remote server, and the *Dns2tcpc*, which operates as a client [23]. The server maintains a configuration file listing available resources, each representing a local or remote service that listens for TCP connections [23], [24]. The client monitors predefined TCP connections and forwards incoming requests to the corresponding service endpoint. AI techniques can enhance monitoring and detection of abnormal traffic patterns generated through this tunneling mechanism.
2) Dnscat2 tool facilitates communication between two servers over the internet [24]. This Java-based utility channels all traffic via a local DNS server and is noted for its speed, efficiency, and high configurability across multiple platforms [25]. AI models can be applied to analyze the traffic generated by Dnscat2, distinguishing legitimate from potentially malicious behavior.
3) Iodine tool is a more modern solution that allows IPv4 data to be tunneled through DNS servers [25], [26]. The ongoing advancement of this tool, along with similar projects, underscores the potential for AI-driven monitoring systems to ensure that DoH achieves its goal of securing web infrastructure.

B. Machine Learning models

The study presents eight machine learning (ML) models, exploring their architectures, underlying equations, and practical applications. These models are widely adopted by researchers for various traffic classification tasks.

1)*Recurrent Neural Network (RNN) :* A recurrent neural network (RNN) is a type of neural network specifically designed to handle time-series and sequential data problems. Unlike standard neural networks, where training is performed independently for each input, RNNs require specialized architectures to retain memory over time [27]. This capability makes RNNs suitable for applications involving sequences, such as text, audio, and video processing [28]. The model achieves this by sharing weights across time steps, allowing it to capture temporal dependencies in the data [29].

The mathematical representation governing the RNN model is expressed in Equation (1) [30].

$$h_t = g(Wx_t + U_f h_{t-1} + b) \quad (1)$$

where, g () = activation function, U and W = flexible weight matrices of the h layer, b = bias, X = input vector.

2)*Long Short Term Memory(LSTM):* LSTM represents an advancement over the traditional RNN model, designed to handle sequences of varying lengths while overcoming the vanishing and exploding gradient problems commonly encountered in RNNs [31]. It finds applications in AI-driven tasks such as sequence forecasting, pattern recognition, identification, and sequence generation [31], [32]. Based on the specific AI problem being addressed, LSTM architectures can be categorized into four types: one-to-one, many-to-one, oneto-many, and many-to-many [31].

$$i_t = \sigma(w_i[h_{t-}1,x_t] + b_i)$$

$$f_t = \sigma(w_f[h_{t-}1,x_t] + b_f) \quad (2)\ o_t = \sigma(w_o[h_{t-}1,x_t] + b_o)$$

$$\tilde{C}_t = tanh(w_c[h_{t-1},x_t] + b_c)$$

$$Ct = ft * Ct-1 + it * \tilde{C}t \quad (3)\ h_t = o_t * tanh(C^t)$$

Table I: Meanings Of The Symbols Used

Symbol	Meaning
i*t*	input gate
f*t*	forget gate
f*t*	output gate
σ	sigmoid function
w*x*	weight for the respective gate(x)neurons
h*t*−1	output of the previous LSTM block at timestamp t-1
x*t*	input at current timestamp
b*x*	biases for the respective gates(x)
$\tilde{C}t$	represents candidate for cell state at timestamp(t)
C*t*	cell state (memory) at timestamp(t)

Governing equations of LSTM are represented by Equation (2,3). Table I summarizes different symbols utilized.

3) *Gated Recurrent Unit (GRU):* The memory and learning capabilities of RNNs are enhanced in LSTM models; however, this enhancement comes at the cost of additional parameters, which increase computational complexity. To overcome this challenge, Artificial Intelligence researchers proposed the gated recurrent unit (GRU) [28]. The operation of the GRU model is described by Equation (4).

$$rt = \sigma(Wrhht-1 + wrxxt + br)$$

$$zt = \sigma(Wzhht-1 + wzxxt + bz) \quad (4)$$

$$\tilde{h}t = tanh(Whh\tilde{} (rt * ht-1) + Whx\tilde{} xt + bz)\ ht = (1 - zt) * ht-1 + zt * \tilde{h}t$$

where,

xt=input at time(t). h_t= output of the cell at time(t),

$\tilde{h}_t$= candidate activation,

W h and W _x= weights,

b= bias, r _t= update gate, z t= reset gate.

4) *Random Forest Classifier (RFC):* The Random Forest Classifier (RFC), an ensemble learning technique, is employed for classification tasks and plays a key role in AI-based predictions [33]. In RFC, several critical parameters guide the model's behavior: "criterion=gini" evaluates the quality of a split, "criterion=entropy" determines how nodes branch within a decision tree, "max depth" sets the maximum depth of the tree, and "n estimators" specifies the total number of trees in the forest (as an integer). The Gini criterion is selected because its values range from 0 to 0.5, whereas Entropy ranges from 0 to 1 [34].

In this framework, the Gini index utilizes Equation (5) to guide the division of nodes in the decision tree, supporting the AI model in making accurate classifications.

$$Gini = \mathbf{1} - \sum_{i=1}^{C} (P_i)^{\mathbf{2}} \quad (5)$$

5) *Decision Tree Classifier (DTC):* Decision Tree is a widely used technique in Artificial Intelligence (AI) for classification and prediction tasks within supervised machine learning [35]. It classifies data by leveraging prior knowledge gained from training datasets. Classification represents a core task in ML and AI, where labeled data is employed to train models and enable accurate decision-making. The method utilizes entropy to assess the homogeneity of a given sample, as described in Equation (6) [34].

$$Entropy = \sum_{i=1}^{C} -P_i * \mathbf{log_2}(P_i) \quad (6)$$

6) *Gradient Boosting Classifier (GBC):* Gradient Boosting is an AI-driven machine learning technique applied to both regression and classification tasks. It generates a predictive model by combining multiple weak learners, usually decision trees, into an ensemble that improves overall accuracy [36]. As an additive AI model, Gradient Boosting can be represented mathematically as shown in Equation (7) [36].

$$F_m(X) = F_{m-1}(X) + \beta_m f_m(X) \quad (7)$$

where F is the ensemble model, f is the weak learner, β is the learning rate and X is the input vector.

7) *XGBoost Classifier (XGBC):* The XGBoost classifier (XGBC) is employed for classification tasks. XGBC is an AIdriven machine learning technique that leverages the gradient boosting framework for predictive modeling. It is widely recognized for its high-speed performance and scalability [22]. XGBC originates from extreme gradient boosting, combining the principles of gradient boosting with enhancements specific to XGBC. As a boosting method, it iteratively calculates the initial prediction values along with their associated errors. The model operates using three key parameters:

1) learning rate = 0.1 is eta weights to make the boosting process more conservative
2) output probability is max depth =8 is the maximum depth of a tree
3) n estimators=100 is the number of boosting rounds

8) K-Neighbors Classifier (KNC): The K-Neighbors Classifier remains one of the most widely applied AI and machine learning techniques. Selecting the optimal value of K depends heavily on the dataset: generally, a larger K reduces the impact of noise but can also make the classification boundaries less precise [37]. In this method, the classifier determines the similarity between a given input and the points in the training dataset by computing the distance, as described in Equation (8). This AI-driven approach enables effective classification by leveraging the structure and patterns within the data.

$$d(x, y) = \sqrt{\sum_{i=1}^{P} (x_i - y_i)^2} \quad (8)$$

EXPERIMENTS AND RESULTS

In this section, the dataset is presented and discussed, followed by a detailed explanation of how the AI and machine learning (ML) models operate. Subsequently, the results produced by the various algorithms implemented within these AI models are examined. Moreover, the performance and accuracy of the resulting AI-driven models are thoroughly analyzed.

A. Data Set Description

The dataset used in this study was obtained from the Canadian Institute for Cybersecurity, known as UNB [38]. Specifically, the "CIRA-CIC-DoHBrw-2020" dataset was employed, and the MaliciousDoH-CSVs portion was selected for analysis. This dataset includes files corresponding to Dns2tcp, Dnscat2, and Iodine, which were merged into a single file for experimental purposes. The combined dataset consists of 249,969 samples with 31 features, which play a critical role in influencing Malicious DoH traffic through DNS tunneling tools. Detailed descriptions of these features are available on the

UNB website [38]. The combined dataset uses Dns2tcp, Dnscat2, Iodine, and Non as target classes for classification. By applying AI-driven models to this dataset, it becomes possible to effectively detect and differentiate between these types of DNS tunneling attacks in DoH traffic, demonstrating the capability of AI to enhance cybersecurity analysis and threat detection.

B. Work Of Classification ML Models

A total of eight experiments were conducted to perform classification tasks. Three of these experiments employed neural network models—specifically RNN, LSTM, and GRU—while the remaining five experiments utilized classical machine learning models, including RFC, DTC, GBC, KNC, and XGBC. The primary objective of these experiments was to classify and identify DNS tunneling within malicious DoH traffic in the applied dataset. To further enhance the performance and reliability of the models, a 10-fold cross-validation approach was implemented to partition the dataset effectively.

The structure and parameters were united for the NN models as below:

- Input layer = (units = 35,input shape = (31,1)).
- The numbers of hidden layers = 2 (20,10).
- Output layer= 4 .
- Activation=”relu” for hidden layers.
- Activation=’softmax’ for output layer.
- Loss = ’categorical crossentropy’.
- Optimization algorithm = ’Adam’.
- Metrics = accuracy, mean squared error, and mean absolute error.

RFC and DTC used parameter (criterion=’gini’) while Parameter(n estimators=100) was used by RFC, GBC and XGBC. Parameter (n _neighbors=3) was used by KNC as well as parametrs (learning rate=0.10, max depth=8) used for GBC and XGBC.

C. Results he outcomes achieved through various AI-driven machine learning models were analyzed. The classifiers evaluated included RFC, DTC, GBC, KNC, XGBC, RNN, LSTM, and GRU. The performance of these AI models was assessed using metrics such as Precision, Recall, F1-Score, Accuracy, as well as Mean Absolute Error (MAE), Mean Squared Error (MSE), and confusion matrices for each model. These evaluations demonstrate the effectiveness of AI approaches in accurately detecting and classifying DNS tunneling attacks.

1) Precision Results: In our experiments, precision is defined as the proportion of threat instances correctly identified among all retrieved instances that utilized DNS Tunneling Tools. The results of the AI and machine learning model experiments are summarized in Table II. As illustrated in Table II, the overall performance of the classifiers is satisfactory. Among the evaluated models, XGBC and RFC demonstrated the highest effectiveness, followed closely by the GBC model. Additionally, the macro and weighted averages of RFC, GBC, and XGBC outperformed those of the other machine learning models, indicating their superior capability in accurately detecting DNS tunneling threats.

2)

Table II: Precision Results

	RFC	DTC	GBC	KNC	XGBC	RNN	LSTM	GRU
Dns2tcp	1	0.99	1.00	0.99	1.00	0.99	0.99	0.99
Dnscat2	0	0.97	0.98	0.95	0.98	0.87	0.87	0.86
Iodine	0	0.98	0.98	0.95	0.98	0.93	0.92	0.93
Non	1	1.00	1.00	0.80	1.00	1.00	1.00	1.00
AVERAGE	0	0.98	0.99	0.92	0.99	0.95	0.94	0.95
WEIGHTED AVG	0	0.98	0.99	0.98	0.99	0.96	0.96	0.96

3)Recall Results: Recall measures the proportion of threat instances correctly identified out of the total number of threat instances associated with DNS Tunneling Tools. The recall outcomes from the AI/ML model experiments are summarized in Table III. As shown in Table III, the XGBC model achieved the highest recall, followed by the RFC model. Additionally, the macro-average recall values indicate that RFC and XGBC outperformed the other ML models, while the weighted average results highlight RFC, GBC, and XGBC as the top performing models among all evaluated approaches.

Table III: Recall Results

	RFC	DTC	GBC	KNC	XGBC	RNN	LSTM	GRU
Dns2tcp	0.99	0.99	0.99	0.98	0.99	0.99	0.98	0.98
Dnscat2	0.99	0.97	0.99	0.97	0.99	0.92	0.91	0.93
Iodine	0.99	0.97	0.99	0.97	1.00	0.90	0.91	0.91
Non	1.00	1.00	0.92	0.92	1.00	0.85	0.77	1.00
MACRO AVG	0.99	0.98	0.97	0.96	0.99	0.91	0.89	0.95
WEIGHTED AVG	0.99	0.98	0.99	0.98	0.99	0.96	0.96	0.96

4)F1-Score Results: The F1-Score, which represents the harmonic mean of Precision and Recall, is a key metric for evaluating AI model performance. Table IV summarizes the F1-Score results obtained from the experiments on various ML models. As shown in Table IV, the XGBC model achieved the highest F1-Score, followed closely by the RFC model. The F1Scores of other AI models, including GBC, DTC, KNC, RNN, GRU, and LSTM, were also satisfactory and demonstrated reasonable performance in descending order. Furthermore, the macro average F1-Scores of RFC and XGBC outperformed the other ML models, while the weighted average F1-Scores of RFC, GBC, and XGBC were superior compared to the remaining models.

Table IV: F1-Score Results

	RFC	DTC	GBC	KNC	XGBC	RNN	LSTM	GRU
Dns2tcp	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Dnscat2	0.98	0.97	0.98	0.96	0.99	0.90	0.89	0.89
Iodine	0.99	0.98	0.98	0.96	0.99	0.92	0.92	0.92
Non	1.00	1.00	0.96	0.86	1.00	0.92	0.87	1.00
MACRO AVG	0.99	0.98	0.98	0.94	0.99	0.93	0.92	0.95
WEIGHTED AVG	0.99	0.98	0.99	0.98	0.99	0.96	0.96	0.96

5) Accuracy,MAE and MSE Results: The accuracy, MAE, and MSE results are summarized in Table V. Among the AI models, the highest accuracies were achieved by XGBC (99.22%) and RFC (99.19%), respectively. Following these, GBC reached an accuracy of 99.00%. Regarding neural network models, the GRU model attained the highest accuracy at 99.12%, while RNN and LSTM achieved 96% and 95.99% accuracy, respectively. These results highlight the effectiveness of AI-based classifiers in achieving precise predictions.

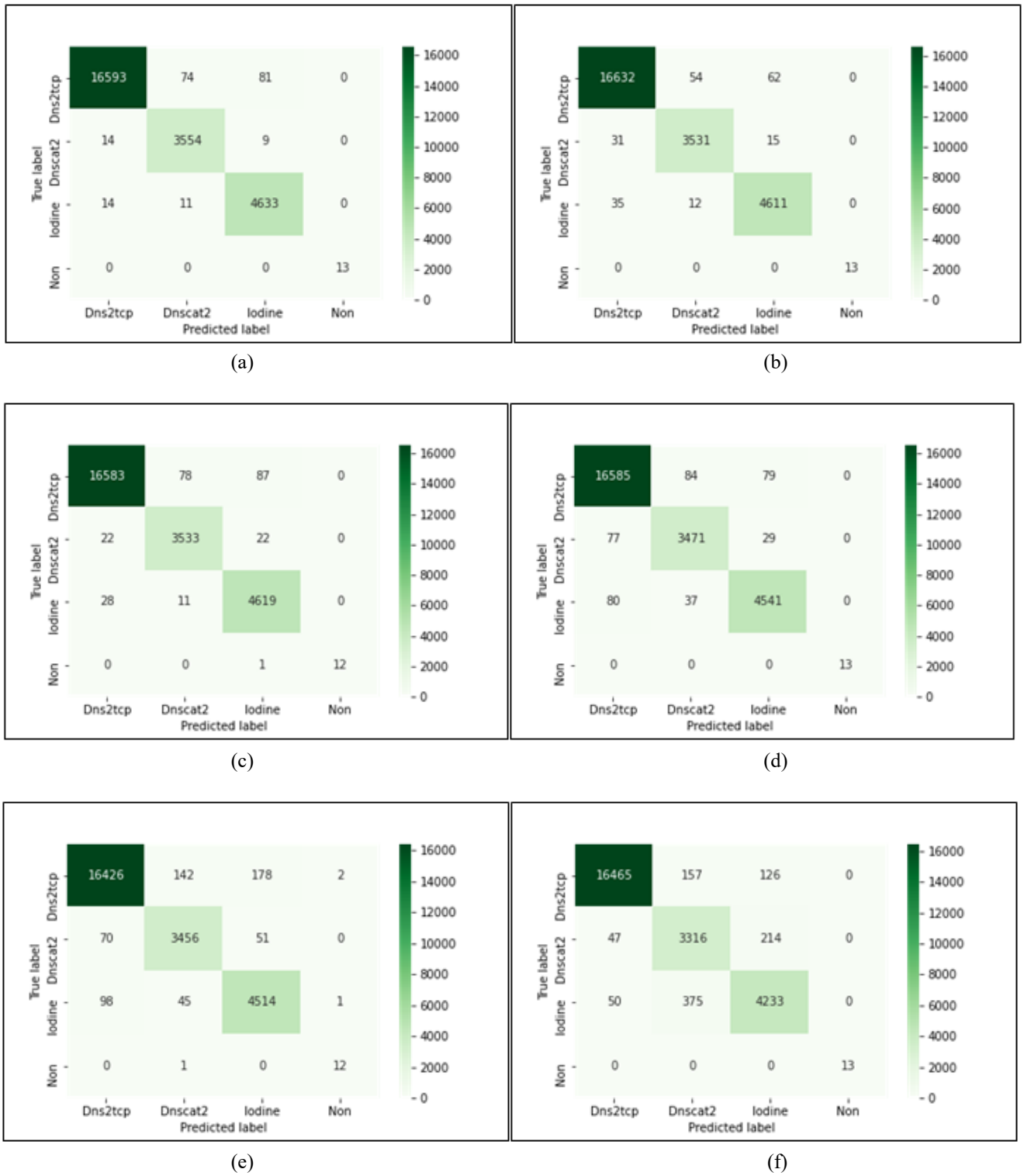
6)

Table V: Accuracy, Mae And Mse Result
s

ModelAccuracy	MAE	MSE
RFC0.9911185789726357	0.0122419587	0.0200032005
DTC0.9845575292046728	0.0218034886	0.0345255241
GBC0.9900384061449832	0.01456233	0.0237638022
KNC0.9764762361977917	0.0347655625	0.0574091855
XGBC 0.9921987518002880	.0113218115	0.0183629381
RNN0.9606336951255798	0.02530716173350811	0.0127535136416554
LSTM 0.9599136114120483	0.02469869330525398	0.0127977291122078
GRU0.9612337946891785	0.02455694414675235	0.012592909857 6307

7) Confusion Matrices Results: Figures 1 and 2 illustrate the confusion matrices for the four categories of DNS Tunneling Tools across all AI-based and machine learning models for both datasets. The confusion matrix in Figure 1(a) illustrates that the XGBC model, when applied to the data set, accurately classified 24,793 samples across all four classes. Among these, the Dns2tcp class achieved the highest number of correct classifications with 16,593 samples. The Iodine class correctly classified 4,633 samples, Dnscat2 accounted for 3,554 correct samples, and the Non class had 13 accurately classified

samples. Minor misclassifications are also evident for each class within the matrix. Figure 1(b) presents the confusion matrix for the RFC model applied to the data set, showing 24,787 samples correctly classified across all four classes. The Dns2tcp class achieved 16,632 correct classifications, while

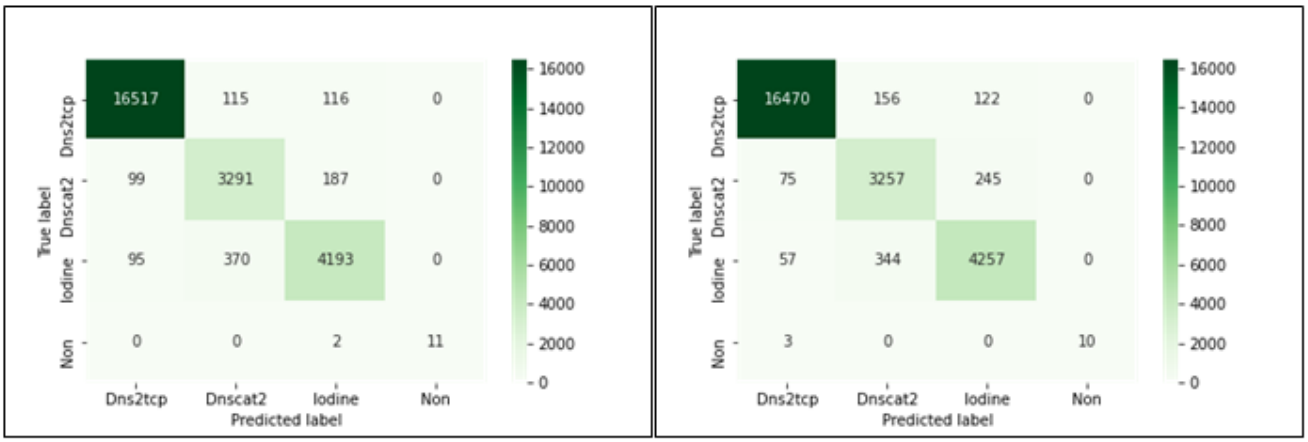


(a) (b)

(c) (d)

(e) (f)

Figure 1. (a) XGBC, (b) RFC, (c) GBC, (d) DTC, (e) KNC, (f) GRU



(a) (b)

Figure 2. (a) RNN, (b) LSTM

the Non class had the fewest with 13 samples. The Iodine class had 4,611 correctly classified samples, and Dnscat2 had 3,531. The GBC model, illustrated in Figure 1(c), achieved correct classification of 24,787 samples for all classes. Specifically, Dns2tcp accounted for 16,583, Iodine for 4,611, Dnscat2 for 3,531, and the Non class for 12 samples. Figure ??(d) shows the confusion matrix of the DTC model, where 24,710 samples were accurately classified across all four classes. The Dns2tcp class classified 16,585 samples correctly, Iodine 4,641, Dnscat2 3,471, and Non 13 samples. The confusion matrix in Figure 1(e) for the KNC model applied to the data set indicates 24,408 correctly classified samples. Dns2tcp had the highest number with 16,426, followed by Iodine at 4,514 and Dnscat2 at 3,456. The Non class had the lowest number of correctly classified samples with 12. In Figure 1(f), the GRU model classified 24,027 samples correctly across all four classes, with Dns2tcp at 16,465, Iodine at 4,233, Dnscat2 at 3,316, and Non at 13 samples.

The RNN model, shown in Figure 2(a), achieved 24,012 correct classifications. The Dns2tcp class accounted for 16,517, Iodine 4,193, Dnscat2 3,291, and Non 11 samples.

Figure 2(b) displays the confusion matrix for the LSTM model, which correctly classified 23,994 samples. The Dns2tcp class had 16,470 correct classifications, Iodine 4,257, Dnscat2 3,257, and Non 10 samples.

In summary, the AI-driven XGBC model demonstrated the best performance, achieving the highest total of 24,793 correctly classified samples among all evaluated models. Additionally, across all AI models, the Dns2tcp class consistently achieved the largest number of correct classifications, highlighting its detectability within the data set.

CONCLUSION

The study analyzed a large cybersecurity dataset, CIRACIC-DoHBrw-2020, obtained from the UNB repository. The dataset consisted of four distinct classes. Various AI and machine learning models, including LSTM, GRU, RNN, RFC, DTC, GBC, KNC, and XGBoost, were applied to classify the data. Model performance was evaluated using multiple metrics such as accuracy, MAE, MSE, classification tables, and confusion matrices.

As presented in the results, the XGBC and RFC models achieved the highest accuracy and F1-score for detecting malicious DoH traffic. Specifically, the XGBC model reached an accuracy of 99.219%, while the RFC model achieved 99.11%, and the GBC model obtained 99% accuracy. Regarding MAE, XGBC recorded the lowest value at 1.13%, followed by RFC at 1.22% and GBC at 1.45

Analysis of the confusion matrices further confirmed that the XGBC model correctly classified 24,793 samples across all four dataset classes, outperforming all other models. The RFC model closely followed, correctly classifying 24,787 samples.

These experiments indicate that AI-driven classifiers, particularly XGBC and RFC, are the most effective choices for addressing classification problems in malicious DoH traffic. Given that this work focused on classifying the types of DNS tunneling tools used in malicious DoH traffic, future research should explore AI-based mechanisms to enhance the security of these tunneling methods.

REFERENCES

[1] N. P. Hoang, A. Akhavan Niaki, N. Borisov, P. Gill, and M. Polychronakis, Assessing the Privacy Benefits of Domain Name Encryption. Proceedings of the 15th ACM Asia Conference on Computer and Communications Security, (Oct. 2020), doi: 10.1145/3320269.3384728.

[2] Gudekli, Ugur Tanik, and Ciylan, Bunyamin. DNS tunneling effect on DNS packet sizes. International Journal of Computer Science and Mobile Computing, (2019), 8(1), 154–162.

[3] Almusawi, Ahmed,and Amintoosi,Haleh, DNS Tunneling Detection Method Based on Multilabel Support Vector Machine Security and Communication Networks,(2018).

[4] Kevin Borgolte, Tithi Chattopadhyay, Nick Feamster, Mihir Kshirsagar,Jordan Holland, Austin Hounsel,and Paul Schmit. How DNS over https is reshaping privacy, performance, and policy in the internet ecosystem. Performance, and Policy in the Internet Ecosystem (July 27, 2019).

[5] M. MontazeriShatoori, L. Davidson, G. Kaur and A. Habibi Lashkari, Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic, 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), 2020, pp. 63-70, doi: 10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00026.

[6] Vekshin, Dmitrii, Hynek, Karel, Cejka, Tomas. DoH insight: Detecting

[7] DNS over https by machine learning. In Proceedings of the 15th International Conference on Availability, Reliability, and Security,(2020, August),(pp. 1-8).

[8] S. K. Singh and P. K. Roy, Detecting Malicious DNS over HTTPS Traffic Using Machine Learning, 2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT), 2020, pp. 1-6, doi: 10.1109/3ICT51146.2020.9312004.

[9] S. Siby, M. Juarez, C. Diaz, N. Vallina-Rodriguez, and C. Troncoso, “Encrypted DNS –¿ Privacy? A Traffic Analysis Perspective.” Proceedings 2020 Network and Distributed System Security Symposium, 2020, doi: 10.14722/ndss.2020.24301.

[10] Yan, Zhiwei, Lee, Jong-Hyouk., The road to DNS privacy. Future Generation Computer Systems,(2020), 112, 604-611.

[11] A. Hounsel, K. Borgolte, P. Schmitt, J. Holland, and N. Feamster, “Comparing the Effects of DNS, DoT, and DoH on Web Performance.” Proceedings of The Web Conference 2020, 2020, doi: 10.1145/3366423.3380139.

[12] F. Palau, C. Catania, J. Guerra, S. Garcia, and M. Rigaki. “DNS Tunneling: A Deep Learning based Lexicographical Detection ..” Jan. 14, 2020. https://arxiv.org/abs/2006.06122 (accessed: Apr. 28, 2024).

[13] M. Sammour, B. Hussin, and M. F. I. Othman. Comparative Analysis for Detecting DNS Tunneling Using Machine Learning Techniques. Nov. 12, 2017. https://www.ripublication.com/ijaer17/ijaerv12n22 _137. pdf (accessed: May 01, 2023).

[14] Y. M. Banadaki, Detecting Malicious DNS over HTTPS Traffic in Domain Name System using Machine Learning Classifiers. Journal of Computer Sciences and Applications, vol. 8, no. 2, pp. 46-55, 2020, doi: 10.12691/jcsa-8-2-2.

[15] S. Shende and S. Thorat, “Long Short-Term Memory (LSTM) Deep Learning Method for Intrusion Detection in Network Security.” International Journal of Engineering Research and, no. 6, Jun. 2020, doi: 10.17577/ijertv9is061016.

[16] .H. HaddadPajouh, A. Dehghantanha, R. Khayami, and K.-K. R. Choo, “A deep Recurrent Neural Network based approach for Internet of Things malware threat hunting.” Future Generation Computer Systems, vol. 85, pp. 88-96, 2018, doi: 10.1016/j.future.2018.03.007.

[17] D. Wang, J. Gong, and Y. Song. “W-RNN: News text classification based on a Weighted RNN.” Sep. 28, 2019. https://arxiv.org/abs/1909.13077 (accessed: May 01, 2024).

[18] W. Yang, W. Zuo, and B. Cui, “Detecting Malicious URLs via a Keyword-Based Convolutional Gated-Recurrent-Unit Neural Network.” IEEE Access, vol. 7, pp. 29891-29900, 2019, doi: 10.1109/access.2019.2895751.

[19] F. Shaikh, E. Bou-Harb, J. Crichigno, N. Ghani, A machine learning model for classifying unsolicited IoT devices by observing network telescopes,” 2018 14th International Wireless Communications Mobile Computing Conference (IWCMC), pp. 938-943, 2018. doi:10.1109/iwcmc.2018.8450404.

[20] K. S. H. Ramos, M. A. S. Monge, J. M. Vidal, “Benchmark-based reference model for evaluating botnet detection tools driven by traffic-flow analytics,” Sensors, vol. 20, no. 16, 1-31, 2020. doi:10.3390/s20164501.

[21] S. Bagui, X. Fang, E. Kalaimannan, S. C. Bagui, J. Sheehan, “Comparison of machine-learning algorithms for classification of VPN network traffic flow using time-related features,” Journal

[22] of Cyber Security Technology, vol. 1, no. 2, pp. 108-126, 2017. doi:10.1080/23742917.2017.1321891.

[23] B. Yamansavascilar, M. A. Guvensan, A. G. Yavuz, M. E. Karsligil, Application identification via network traffic classification. 2017 International Conference on Computing, Networking and Communications (ICNC), 2017. doi:10.1109/iccnc.2017.7876241.

[24] R. Kumar, S. Geetha, Malware classification using XGboost – Gradient boosted decision tree, Advances in Science Technology and Engineering, vol. 5, no. 5, pp. 536-549, 2020. doi:10.25046/aj050566.
[25] W. Jammal. Multi-Stage Detection Technique for DNS-Based Botnets. 2017. https://repository.stcloudstate.edu/cgi/viewcontent.cgi?article= 1058&context=msia etds&httpsredir=1&referer= (accessed: May 01, 2023).
[26] M. Al-kasassbeh and T. Khairallah, "Winning tactics with DNS tunnelling." Network Security, vol. 2019, no. 12, pp. 12-19, 2019, doi: 10.1016/s1353-4858(19)30144-8.
[27] H. Bai, Refined identification of hybrid traffic in DNS tunnels based on regression analysis. ETRI Journal, vol. 43, no. 1, pp. 40-52, 2020, doi: 10.4218/etrij.2019-0299.
[28] S. Shafieian, D. Smith, and M. Zulkernine, Detecting DNS Tunneling Using Ensemble Learning. Network and System Security, pp. 112-127, 2017, doi: 10.1007/978-3-319-64701-2 9.
[29] D. Dang, F. Di Troia, and M. Stamp, Malware Classification using Long Short term Memory Models. Proceedings of the 7th International Conference on Information Systems Security and Privacy, 2021, doi: 10.5220/0010378007430752.
[30] Y. Yu, X. Si, C. Hu, and J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures." Neural Computation, vol. 31, no. 7, pp. 1235-1270, 2019, doi: 10.1162/neco a 01199.
[31] Y. Fu, S. Saab Jr, A. Ray, and M. Hauser,A Dynamically Controlled Recurrent Neural Network for Modeling Dynamical Systems. 2019. (accessed: May 03, 2023). https://arxiv.org/abs/1911.00089
[32] H. Apaydin, H. Feizi, M. T. Sattari, M. S. Colak, S. Shamshirband, and K.-W. Chau, "Comparative Analysis of Recurrent Neural Network Architectures for Reservoir Inflow Forecasting." Water, vol. 12, no. 5, p. 1500, 2020, doi: 10.3390/w12051500.
[33] K. Smagulova and A. P. James, Overview of Long Short-Term Memory Neural Networks. Modeling and Optimization in Science and Technologies, pp. 139-153, 2019, doi: 10.1007/978-3-030-14524-8 11.
[34] A. Sherstinsky, Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. Physica D: Nonlinear Phenomena, vol. 404, p. 132306, 2020, doi: 10.1016/j.physd.2019.132306.
[35] M. Thenuwara and H. R. K. Nagahamulla, "Offline handwritten signature verification system using random forest classifier," 2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer), 2017, pp. 1-6, doi: 10.1109/ICTER.2017.8257828.
[36] L. Khaidem, S. Saha, and S. Roy Dey. Predicting the direction of stock market prices using random forest. arXiv.org. Apr. 29, 2016. https:// arxiv.org/abs/1605.00003 .
[37] S. Patil and U. Kulkarni, "Accuracy Prediction for Distributed Decision Tree using Machine Learning approach," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 1365-1371, doi: 10.1109/ICOEI.2019.8862580.
[38] C. Ben Issaid, C. Anton-Haro, X. Mestre and M. -S. Alouini, "User´ Clustering for MIMO NOMA via Classifier Chains and GradientBoosting Decision Trees," in IEEE Access, vol. 8, pp. 211411-211421, 2020, doi: 10.1109/ACCESS.2020.3038490.
[39] M. T. Hoang et al., "A Soft Range Limited K-Nearest Neighbors Algorithm for Indoor Localization Enhancement," in IEEE Sensors Journal, vol. 18, no. 24, pp. 10208-10216, 15 Dec.15, 2018, doi: 10.1109/JSEN.2018.2874453.
[40] Canadian Institute for Cybersecurity — UNB. https://www.unb.ca/cic/ (accessed: April. 24, 2024).