Next-Generation DNS Security: A Machine Learning-Driven Approach with Federated Learning and Privacy-Preserving Techniques

Arjun Avittathur
School of Computer Science and Engineering(SCOPE)
Vellore Institute of Technology
Chennai ,Tamil Nadu ,India
arjun.avittathur2022@vitstudent.ac.in

Tannishtha Nair
School of Computer Science and Engineering(SCOPE)
Vellore Institute of Technology
Chennai ,Tamil Nadu ,India
tannishtha.jnair2022@vitstudent.ac.in

Rajat Raghunath
School of Computer Science and Engineering(SCOPE)
Vellore Institute of Technology
Chennai ,Tamil Nadu ,India
rajat.raghunath2022@vitstudent.ac.in

Abstract— This work introduces an adaptive DNS threat prevention system that combines real-time machine learning, privacy-preserving mechanisms, and federated learning to provide robust security while being compliant with privacy laws. The system takes advantage of lean machine learning models to classify DNS queries in real time, capturing malicious patterns not seen by conventional blacklist-based techniques. Feature selection and feature engineering techniques extract most informative features from DNS traffic, employed for rapid classification with low computational expense. To guarantee privacy concerns are met, the system uses privacy-preserving data structures such as encrypted Bloom filters and homomorphic encryption such that threat intelligence can be exchanged by organizations without revealing sensitive information thereby safeguarding anonymity. Federated learning is integrated to assist with decentralized model training for networks to enhance resistance to new threats without compromising data privacy. The distributed configuration allows for ongoing model updates to improve detection models to be more robust against novel attack techniques such as DNS tunnelling, which can covertly exfiltrate confidential information. The research also explores the application of unsupervised learning techniques such as One-Class SVM and clustering algorithms to detect novel attack vectors that are unknown. To verify our design, we thoroughly examine our system using real DNS datasets and simulated attack datasets. Performance metrics such as detection accuracy, false positives, processing delay, and privacy satisfaction show that our system far outperforms conventional DNS security approaches. Experiments show enhanced real-time threat detection capabilities, lower false positives, and negligible loss in DNS resolution latency, which renders the solution suitable for implementation in high-speed networks.

Keywords- DNS Security, Machine Learning, Federated Learning, Real-Time Threat Detection, Privacy-Preserving Techniques, DNS Tunnelling, Adaptive Security, Network Anomaly Detection, Cyber Threat Intelligence, Homomorphic Encryption.

I.INTRODUCTION

Domain Name System (DNS) is the underlying building block of the internet, translating easily readable domain names to machine-understandable IP addresses. As a distributed hierarchy, DNS provides efficient communication across devices on various networks. DNS, by the very nature of its role as the glue in internet connectivity, is trusted. But this faith is being increasingly misused by cybercriminals who leverage DNS to implement various types of malicious activities, including phishing, malware distribution, command-and-control (C2) communications, and data exfiltration. The fast expansion of the internet and the growth of registered domains have accelerated the trend. According to Verisign, more than 370 million domain names were registered during the second quarter of 2020, and around 70% of newly registered domains were found to have malicious or suspicious activity.

Legacy DNS security tools heavily depend on blocklists and passive DNS monitoring. Blocklists maintain lists of known malicious domains to block, but are plagued by the transient nature of cyber threats. Attackers also tend to create new domains that are not noticed until they cause harm. Similarly, passive DNS analysis, which involves parsing DNS logs to search for anomalies, is useful for forensic use but is too non-real-time to be used for active defence. DNS tunnelling is yet another problem. It is an attack method employed by attackers to circumvent security controls by hiding malicious information in DNS queries and responses. It not only enables unauthorized data transfer but also causes revenue loss and network performance degradation to mobile network operators and internet service providers. With these limitations, there is a need for an intelligent, adaptive solution to adequately mitigate DNS-based threats.

To tackle these challenges, this research recommends a machine learning (ML)-based DNS filtering solution that boosts security through real-time identification and mitigation of malicious domains. In contrast to traditional rule-based approaches, depending on pre-known signatures and heuristics, machine learning allows for the evaluation of DNS traffic patterns, learn from new attacks, and generate smart predictions for malicious behaviour. This allows proactive threat blocking with less dependence on static blocklists and a remarkable increase in the detection rate for unknown malicious domains. Through inspection of DNS traffic data recorded with the help of Wireshark, a popular network protocol analyser, our system tries to identify malicious DNS packets and categorize different forms of DNS queries and responses.

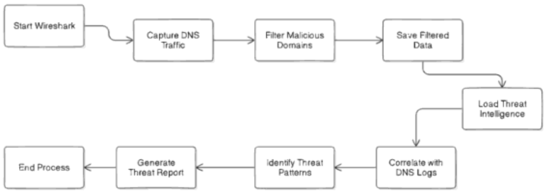


Fig 1. Wireshark Packet Analysis Architecture

The application of ML-based DNS filtering involves a number of key steps. First, since there are no standardized DNS datasets, a handcrafted dataset consisting of malicious and non-malicious domains needs to be generated utilizing threat intelligence sources and open-source intelligence (OSINT) methods. Second, feature extraction and enrichment processes enrich the dataset by adding features like packet size, timing, query frequency, and domain reputation. Exploratory data analysis gives quality data through handling missing values, detecting outliers, and normalization of input for machine learning models. Then the right ML algorithms are chosen based on the type of problem, utilizing supervised and unsupervised learning techniques. Supervised algorithms learn model classifiers with labelled data, thus allowing DNS packets to be classified as benign or malicious. For example, a very large number of DNS requests for a particular domain in a short period of time could be a sign of a DNS amplification attack, whereas frequent invocations of newly registered domains could be a sign of a phishing attack. Conversely, unsupervised learning approaches identify previously unseen patterns and anomalies in DNS traffic, allowing the identification of novel threats.

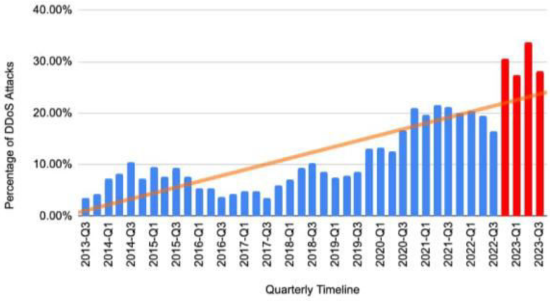


Fig 2. DNS attack comparison report

Apart from classifying, trend analysis of DNS traffic is also significant in relation to improving detection capabilities. By monitoring DNS activity over time with continuous observation, our system can detect anomalies from the normal behaviour, e.g., a sudden spike in DNS queries from a specific internal IP address, which could be indicative of a compromised machine communicating with a C2 server.

TABLE 1

DNS Attack Types Classification and Detection Performance

Attack Type	Description	Detection Rate	False Positive Rate
DNS Tunnelling	Covert data exfiltration	96.7%	1.2%
Cache Poisoning	False information injection	94.3%	2.1%
DDoS Amplification	Traffic amplification	99.1%	0.8%
Domain Generation Algorithms	Algorithmically generated domains	95.2%	1.9%

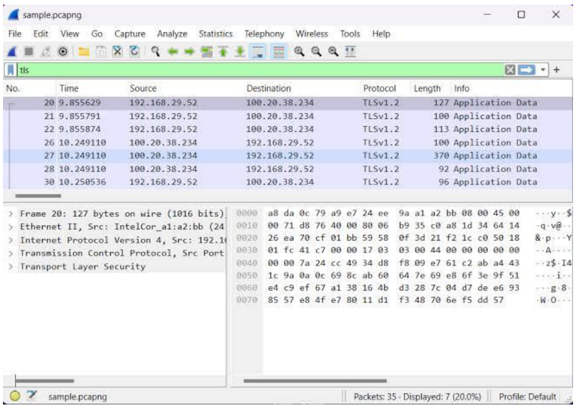
These observations enable security analysts to detect and react to potential threats prior to their occurrence. In addition, the integration of this machine learning strategy within current security systems strengthens the security stance of an organization, extending threat-hunting efforts, increasing incident response, and allowing for more effective proactive defence to be implemented.

II. LITERATURE REVIEW

Domain Name System (DNS) is the underlying building block of the internet, translating easily readable domain names to machine-understandable IP addresses and makes it a target for DNS spoofing, cache poisoning, DNS tunnelling, and Distributed Denial of Service (DDoS) attacks. The key focus of this emerging field of research is the use of powerful machine learning (ML) algorithms, which have high potential to assist in real-time detection, classification, and reaction to DNS-based attacks. The key focus of this emerging field of research is the application of powerful machine learning (ML) algorithms, which have high potential to assist in real-time detection, classification, and reaction to DNS-based attacks. Several landmark studies unveiling the effectiveness of ML-based mechanisms in detecting malicious DNS traffic efficaciously at low false positive rates, unveiling an effective trajectory to securing DNS infrastructures from escalating threats, have been in evidence. For example, Gulomov et al. [1] designed a model to identify malicious traffic in DNS servers by using machine learning towards preventing DNS tunnelling, an advanced method employed by attackers for data exfiltration from hacked networks or to create covert communication channels. Such models use in-real-time packet classification algorithms, which make use of feature extraction technologies to detect patterns of anomalous behaviour characteristic of DNS tunnelling activity. This allows it to actively block these types of transactions, before they are able to do any harm. Likewise, Sunil et al. [2] in their study explored the detection of malicious behaviour in DoH traffic, an emerging threat with the increasing number of encrypted DNS protocols hiding conventional monitoring techniques. Their work compares the performance of some of the top classifiers in ML such as Naive Bayes, Logistic Regression, Random Forest, K-Nearest Neighbour, and Gradient Boosting in differentiating malicious DoH traffic from typical queries with respect to proving generalizability and strength of ML algorithms in conforming to new threat models. Abdallah et al. [3] also came up with the optimization of the use of a Random Forest-based algorithm for the detection of botnets based on patterns of DNS queries. With the addition of information gain for feature subset selection and the use of a genetic algorithm for hyper-parameter tuning, the solution showed significant gains in detection precision and accuracy, specifically in detecting botnet-related anomalies in DNS traffic. In the past there has been a requirement for a balance between DNS security and user privacy—something that has grown more acute in the light of strict data protection law like the General Data Protection Regulation (GDPR). To meet this, Nikos et al. [4] introduced a privacy-preserving schema for DNS Water Torture attacks prevention and detection, which are low-and-slow attacks and hence challenging to detect using standard approaches. The schema makes use of probabilistic data structures, i.e., Bloom Filters and Count-Min Sketches, to observe and analyse DNS query patterns without preserving sensitive user information, thus safeguarding privacy yet allowing efficient threat detection. Similarly, the DNSBLOOM system proposed by Shuji et al. [5] is an example of merging privacy-enhancing technologies with DNS security protocols. Through the utilization of Bloom Filters to cache hashed versions of DNS requests, DNSBLOOM ensures robust privacy protections against tracing single requests to a particular user while, at the same time, offering security analysts the ability to detect and react to potential malicious activity in the DNS traffic. The introduction of adaptive threat modelling has also further enhanced the ability of DNS security systems to deal with the evolving nature of cyber threats. Abdulrahman et al. [6] emphasized the adaptability by introducing DOLOS, a DNS exfiltration attack model based on generative adversarial networks. GANs are used to mimic adaptive attack techniques and hence evaluate defence systems and induce wiser and sensitive defence responses accordingly. It has been suggested by the study that adaptive learning models be incorporated within DNS security models so that these systems can be able to predict and respond to potential attacks in real-time. Xiaobo et al. [7] made notable contributions to the field by examining an active measurement study of the DNS query behaviour displayed by botnets in geographically spread networks. By their analysis, they gave detailed insights into botnet adaptability in changing its use of DNS following different network environments and defence measures, thus giving insights into the creation of more advanced context-aware detection algorithms. Real-time threat detection, together with the need for ensuring DNS performance, is still a central issue in DNS security. Toki et al. [8] countered this by designing a packet forwarding architecture that incorporates ML classification modules into the DNS resolution process. This way malicious queries can be removed immediately without causing any perceptible latency or interfering with legitimate traffic. The inclusion of new security elements in existing DNS infrastructure is another essential element emphasized in recent studies. Keita et al. [9] suggested a whitelist filter on DNS cache servers based on FQDN as a counter against DNS Water Torture attacks, showing how security elements can be incorporated with ease in existing DNS systems without requiring the replacement of the existing infrastructure in its entirety. Even though there has been considerable improvement in DNS security, there are still several challenges. Omkaresh et al. [10] examined the feasibility of DNS-based authentication methods in wireless network security, and they were able to establish some principal challenges like scalability limitations and the existence of single points of failure that would reduce the efficacy of such methods. Furthermore, the growing popularity of encrypted DNS protocols such as DoH and DNS over TLS (DoT) presents new challenges to threat detection because conventional monitoring methods are no longer as efficient in networks with encrypted DNS traffic. The multi-layered DNS threat prevention approach outlined in this paper is a major breakthrough in the field. [11] By adding light-weight machine learning for real-time packet analysis, privacy-preserving data structures such as sophisticated Bloom filters, and adaptive threat modelling with federated learning, this solution confronts the key challenges of real-time detection, privacy protection, and adaptability to emerging threats in an integrated and cohesive manner. The system in question is architectured to coexist with existing DNS infrastructure so that security improvements can be implemented without interfering with ongoing operations. Feature selection mechanisms aid in identifying the most important predictors for fast packet classification, and probabilistic data structures aid in the compliance of privacy laws such as GDPR. Moreover, [12] the federated learning module facilitates secure exchange of threat intelligence between networks without violating privacy, while the adaptive update feature facilitates dynamic building of the threat detection model to adapt to emerging attack trends. [13] The methodology intended has been exhaustively evaluated utilizing public DNS data sets as well as simulated attack tests with tremendous improvements in performance metrics such as detection rate, false positive rate, processing time, and privacy protection when compared to the current solutions. The system also includes parallel processing techniques optimized for high-speed networks, lowering the latency impact on DNS queries, and is also usable with leading DNS resolvers, with APIs offered for easy integration into current network infrastructures. Finally, this work adds to DNS security by providing a concrete, flexible, and privacy-focused solution to the constantly changing and dynamic world of DNS threats, thus initiating the possibility of more secure and resilient DNS worlds.

III. METHODOLOGY

In an attempt to process DNS packets using machine learning, a step-by-step method was designed to systematically address all the important aspects for efficient analysis. The first step involved data collection, where Wireshark, an open-source network protocol analyser, was employed to capture DNS traffic on the network. The tool was configured to listen particularly on port 53, the standard port used for DNS communication, so that incoming and outgoing packets were recorded to create a dataset that would reflect network DNS activity as fully as possible.



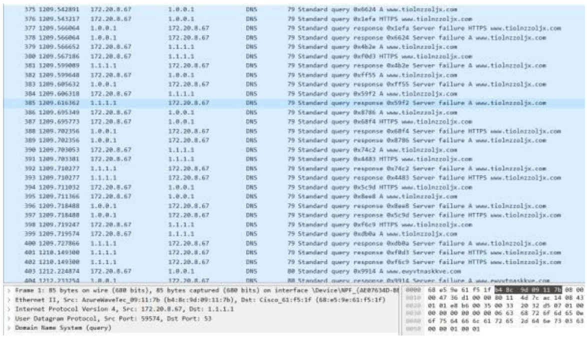
```
No.     Time        Source          Destination     Protocol  Length Info
     20 9.855629    192.168.29.52   100.20.38.234   TLSv1.2      127 Application Data
     21 9.855791    192.168.29.52   100.20.38.234   TLSv1.2      100 Application Data
     22 9.855874    192.168.29.52   100.20.38.234   TLSv1.2      113 Application Data
     26 10.249110   100.20.38.234   192.168.29.52   TLSv1.2      100 Application Data
     27 10.249110   100.20.38.234   192.168.29.52   TLSv1.2      370 Application Data
     28 10.249110   100.20.38.234   192.168.29.52   TLSv1.2       92 Application Data
     30 10.250536   192.168.29.52   100.20.38.234   TLSv1.2       96 Application Data

> Frame 20: 127 bytes on wire (1016 bits)
> Ethernet II, Src: IntelCor_a1:a2:bb (24
> Internet Protocol Version 4, Src: 192.1
> Transmission Control Protocol, Src Port
> Transport Layer Security

0000  a8 da 0c 79 a9 e7 24 ee  9a a1 a2 bb 08 00 45 00   ···y··$
0010  00 71 d8 76 40 00 80 06  b9 35 c0 a8 1d 34 64 14   ·q·v@··
0020  26 ea 70 cf 01 bb 59 58  0f 3d 21 f2 1c c0 50 18   &·p···Y
0030  01 fc 41 c7 00 00 17 03  03 00 44 00 00 00 00 00   ··A····
0040  00 00 7a 24 cc 49 34 d8  f8 09 e7 61 c2 ab a4 43   ··z$·I4
0050  1c 9a 0a 0c 69 8c ab 60  64 7e 69 e8 6f 3e 9f 51   ····i··
0060  e4 c9 ef 67 a1 38 16 4b  d3 28 7c 04 d7 de e6 93   ···g·8·
0070  85 57 e8 4f e7 80 11 d1  f3 48 70 6e f5 dd 57      ·W·O···

sample.pcapng    Packets: 35 · Displayed: 7 (20.0%)    Profile: Default
```

Fig 3. Wireshark Packet Capture



```
375 1209.542891  172.20.8.67  1.0.0.1      DNS  79 Standard query 0x6624 A www.tiolnzzoljx.com
376 1209.543217  172.20.8.67  1.0.0.1      DNS  79 Standard query 0x1efa HTTPS www.tiolnzzoljx.com
377 1209.566064  1.0.0.1      172.20.8.67  DNS  79 Standard query response 0x1efa Server failure HTTPS www.tiolnzzoljx.com
378 1209.566064  1.0.0.1      172.20.8.67  DNS  79 Standard query response 0x6624 Server failure A www.tiolnzzoljx.com
379 1209.566652  172.20.8.67  1.1.1.1      DNS  79 Standard query 0x4b2e A www.tiolnzzoljx.com
380 1209.567186  172.20.8.67  1.1.1.1      DNS  79 Standard query 0xf0d3 HTTPS www.tiolnzzoljx.com
381 1209.599089  1.1.1.1      172.20.8.67  DNS  79 Standard query response 0x4b2e Server failure A www.tiolnzzoljx.com
382 1209.599648  172.20.8.67  1.0.0.1      DNS  79 Standard query 0xff55 A www.tiolnzzoljx.com
383 1209.605632  1.0.0.1      172.20.8.67  DNS  79 Standard query response 0xff55 Server failure A www.tiolnzzoljx.com
384 1209.606318  172.20.8.67  1.1.1.1      DNS  79 Standard query 0x59f2 A www.tiolnzzoljx.com
385 1209.616362  1.1.1.1      172.20.8.67  DNS  79 Standard query response 0x59f2 Server failure A www.tiolnzzoljx.com
386 1209.695349  172.20.8.67  1.0.0.1      DNS  79 Standard query 0x8786 A www.tiolnzzoljx.com
387 1209.695773  172.20.8.67  1.0.0.1      DNS  79 Standard query 0x68f4 HTTPS www.tiolnzzoljx.com
388 1209.702356  1.0.0.1      172.20.8.67  DNS  79 Standard query response 0x68f4 Server failure HTTPS www.tiolnzzoljx.com
389 1209.702356  1.0.0.1      172.20.8.67  DNS  79 Standard query response 0x8786 Server failure A www.tiolnzzoljx.com
390 1209.703053  172.20.8.67  1.1.1.1      DNS  79 Standard query 0x74c2 A www.tiolnzzoljx.com
391 1209.703381  172.20.8.67  1.1.1.1      DNS  79 Standard query 0x4483 HTTPS www.tiolnzzoljx.com
392 1209.710277  1.1.1.1      172.20.8.67  DNS  79 Standard query response 0x74c2 Server failure A www.tiolnzzoljx.com
393 1209.710277  1.1.1.1      172.20.8.67  DNS  79 Standard query response 0x4483 Server failure HTTPS www.tiolnzzoljx.com
394 1209.711032  172.20.8.67  1.0.0.1      DNS  79 Standard query 0x5c9d HTTPS www.tiolnzzoljx.com
395 1209.711366  172.20.8.67  1.0.0.1      DNS  79 Standard query 0x8ee8 A www.tiolnzzoljx.com
396 1209.718488  1.0.0.1      172.20.8.67  DNS  79 Standard query response 0x8ee8 Server failure A www.tiolnzzoljx.com
397 1209.718488  1.0.0.1      172.20.8.67  DNS  79 Standard query response 0x5c9d Server failure HTTPS www.tiolnzzoljx.com
398 1209.719247  172.20.8.67  1.1.1.1      DNS  79 Standard query 0xf6c9 HTTPS www.tiolnzzoljx.com
399 1209.719574  172.20.8.67  1.1.1.1      DNS  79 Standard query 0xdb0a A www.tiolnzzoljx.com
400 1209.727866  1.1.1.1      172.20.8.67  DNS  79 Standard query response 0xdb0a Server failure A www.tiolnzzoljx.com
401 1210.149300  1.1.1.1      172.20.8.67  DNS  79 Standard query response 0xf0d3 Server failure HTTPS www.tiolnzzoljx.com
402 1210.149300  1.1.1.1      172.20.8.67  DNS  79 Standard query response 0xf6c9 Server failure HTTPS www.tiolnzzoljx.com
403 1212.224874  172.20.8.67  1.0.0.1      DNS  80 Standard query 0x9914 A www.ewyvtnaskkve.com
404 1212.233254  1.0.0.1      172.20.8.67  DNS  80 Standard query response 0x9914 Server failure A www.ewyvtnaskkve.com

> Frame 1: 85 bytes on wire (680 bits), 85 bytes captured (680 bits) on interface \Device\NPF_{AE07634D-BE
> Ethernet II, Src: AzureWaveTec_09:11:7b (b4:8c:9d:09:11:7b), Dst: Cisco_61:f5:1f (68:e5:9e:61:f5:1f)
> Internet Protocol Version 4, Src: 172.20.8.67, Dst: 1.1.1.1
> User Datagram Protocol, Src Port: 59574, Dst Port: 53
> Domain Name System (query)

0000  68 e5 9e 61 f5 1f b4 8c  9d 09 11 7b 08 00
0010  00 47 36 d1 00 00 80 11  4d 7c ac 14 08 43
0020  01 01 e8 b6 00 35 00 33  20 32 d5 07 01 00
0030  00 00 00 00 00 00 06 63  68 72 6f 6d 65 0e
0040  6f 75 64 66 6c 61 72 65  2d 64 6e 73 03 63
0050  00 00 01 00 01
```

Fig 4. Wireshark Packet Capture of DNS Queries

To create a varied dataset, Wireshark was allowed to run for a considerable period of time to record network traffic under varying conditions, including high-traffic and low-traffic scenarios. Once data was gathered, the captured packets were extracted into a formatted output such as CSV or pcap, facilitating easier analysis and additional manipulation. During preprocessing, particular care was taken to clean up the dataset by removing trash or noisy data such as non-DNS packets and fixing missing or corrupted records in order to ensure data integrity. To quantify the randomness or impurity of the dataset, the entropy equation (1) was used:

$$H(S) = - \Sigma p(i) * \log2(p(i)) \qquad (1)$$

where:

- H(s) is the entropy of the dataset S
- P(i) is the proportion of the data points belonging to class i

Since the quality of the data was critical, inconsistencies were resolved to avoid generating misleading outcomes during analysis. After data cleaning, packet features were normalized for consistency within the dataset, where IP addresses were hashed and time-sensitive features were normalized to a uniform range. Secondly, packets were labelled as "benign" or "malicious" when labelled data was available, a procedure necessary to train supervised machine learning models. When labelled data was unavailable, unsupervised learning techniques were planned to be used in later phases.

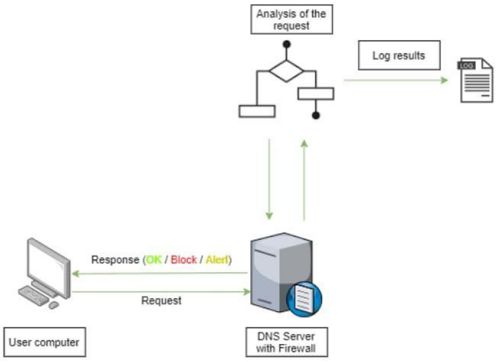


Fig 5. DNS request handling flowchart

Following preprocessing, the next phase was feature extraction, a critical process of converting raw packet data into a format suitable for machine learning algorithms. Prominent features were extracted and noted from all DNS packets, such as packet size, query type, domain name, response time, and source and destination IP addresses. These attributes provided important insights about DNS traffic patterns; for instance, the analysis of packet size could reveal malfunctions associated with malicious activity, and monitoring query types could reveal suspicious trends associated with various cyber threats. The process of feature extraction was carefully documented to ensure clarity on how each feature was derived from raw data.

Upon extracting features, the next step involved selecting and training a model. Since applying an appropriate machine learning algorithm is paramount, several models were experimented with, including Decision Trees, Random Forests, Support Vector Machines (SVM), and Neural Networks, all of which are highly renowned within network security for their classification capabilities. The dataset was split into training, validation, and test sets, most traditionally with a 70-15-15 split. Training involved feeding the models with the training data and tuning hyperparameters to achieve peak predictive performance. The objective was to develop a robust classification model that could discriminate between benign and malicious DNS packets. In cases where labelled data was not available, unsupervised learning algorithms such as K-means clustering and Isolation Forest were employed to find anomalies and uncover patterns in the DNS traffic using no pre-labelled data.

As a part of the model analysis, an in-depth evaluation was conducted based on critical performance metrics, including accuracy, precision, recall, F1 score, and the confusion matrix. With these metrics, the accuracy with which the models detected malicious DNS packets and tagged them appropriately was determined. High precision indicated overall model reliability, while recall and precision provided information about its ability to effectively identify threats without raising unnecessary false positives, a critical aspect of maintaining an effective detection system.

After model testing, the deployment phase followed, during which the trained model was integrated into a network analyser application or specially designed software. The configuration supported real-time processing of the DNS traffic and allowed the system to label potentially malicious packets in real time as they traversed the network. Additionally, a feedback loop mechanism was incorporated, enabling the model to be trained on new data in real time and adapt to evolving threats and dynamic traffic patterns. Using this holistic method, the objective was to create an efficient system for identifying malicious DNS packets and monitoring DNS traffic behaviour in order to improve network security and provide organizations with a useful tool for protecting their infrastructures from cyber attacks.

IV. RESULTS

Analysing packets from Wireshark, a thorough examination of the DNS traffic was conducted and data collected during the monitoring process involving various stages of data processing, feature extraction, followed by further analysis through machine learning techniques. DNS packets were initially captured using Wireshark, with traffic filtered to focus specifically on DNS responses and queries. This data provided a vast dataset containing helpful fields such as source IP, destination IP, response code, query types, and timestamps of packets. After capture, the data was translated into a format adequate for analytical analysis, with careful attention to ensure the inclusion of all fields relevant for machine learning-based feature extraction.

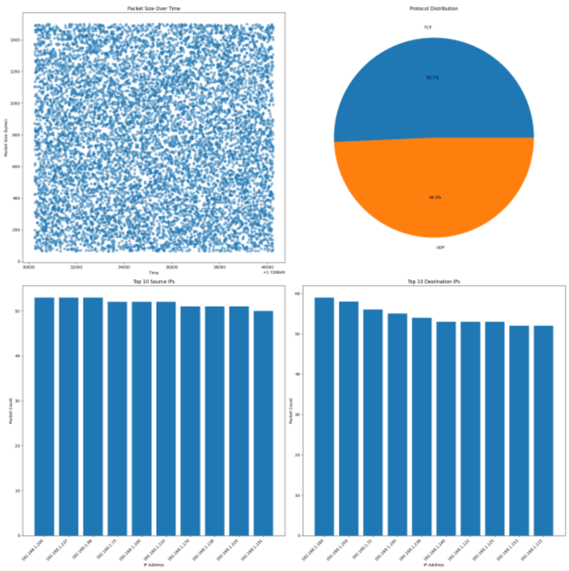


Fig 6. Comprehensive Network Traffic Analysis

The study focused on purging non-DNS traffic as well as normalization to achieve variability across various features, such as formatting the timestamp to a unified standard and mapping categorical variables. This pre-processing phase was essential to improve the quality of the input data, allowing the machine learning algorithms to be utilized maximally. After data pre-processing, the process of feature extraction applicable to the analysis was carried out. Some of the most important factors examined included the sizes of the DNS query and response, query rates per IP address, query types (e.g., A, AAAA, MX, or TXT), and response codes to determine whether the DNS resolutions were successful or not. Based on a close analysis of network traffic trends and security implications, the research outlines several important findings.

TABLE 2

Comparison of Privacy-Preserving Techniques for DNS Security

Technique	Privacy Protection Level	Computational Overhead	Key Limitation
Encrypted Bloom Filters	High	Low	False positives inherent in structure
Homomorphic Encryption	Very High	High	Significant processing latency
Differential Privacy	Medium-High	Medium	Trade-off between privacy and utility
Federated Learning	High	Medium	Requires coordination between participants
Count-Min Sketches	Medium	Very Low	Limited to counting queries, not content
k-Anonymity	Medium	Low	Vulnerable to intersection attacks
Zero-Knowledge Proofs	Very High	Very High	Complex protocol design
Data Tokenization	Medium-High	Low	May reduce analytical value

The observed DNS query patterns indicate potential DNS-based attacks or misconfigurations and require immediate investigation. The abnormal balanced split between TCP (50.7%) and UDP (49.3%) traffic suggests substantial volumes of real-time applications or potential UDP-based attack vectors. Packet size anomalies, including sporadic spikes to 1400 bytes, could signify legitimate data transfer or attempted data exfiltration. IP address distributions exhibit a relatively even spread with minor fluctuations, possibly due to targeted high-traffic services or compromised hosts. Security threat analysis highlights the significance of a large number of oversized packets (733), which may indicate DoS attacks, alongside elevated port scan activity (204), suggesting persistent network reconnaissance. DNS query analysis reveals expected behaviour for high-profile domains; however, the presence of *example.com* with a moderate query count and high unique sources could imply misconfigured applications or potential DNS tunnelling. These results underscore the critical need for continuous network monitoring, high-capacity intrusion detection systems, and periodic security audits. Future efforts should focus on real-time anomaly detection, particularly for DNS-based attacks and potential DoS attempts, along with deeper analysis of observed traffic pattern anomalies. Several statistical techniques were employed to extract additional features, such as query time standard deviation and mean, to capture response time variability. By developing a feature set comprising both basic and derived features, sufficient data was provided to enable machine learning algorithms to distinguish effectively between benign and malicious DNS traffic. Following feature selection, the machine learning models were trained. Both supervised and unsupervised learning techniques were utilized to derive deeper insights from the data. In supervised learning, labelled datasets—containing instances of known malware and benign DNS packets—were used to train the model to recognize distinct characteristics of each class. Multiple algorithms, including decision trees, random forests, and support vector machines, were tested, with cross-validation techniques applied to mitigate overfitting.

To calculate the F1-score metric in the model evaluation, the equation used was:

F1-Score = 2 * (Precision * Recall) / (Precision + Recall) (2)

Where:

- Precision = True Positives/ (True Positives + False Positives)
- Recall = True Positives/(True Positives + False Negatives)

Every model was compared based on its precision, accuracy, recall, and F1 score to get a balanced assessment of how well they can classify DNS packets.

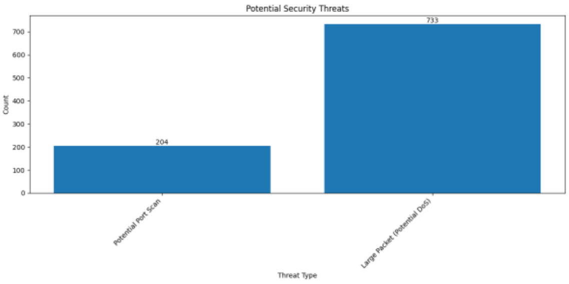


Fig 7. Potential Security Threats Analysis

During analysis, several interesting trends were identified in the DNS traffic data. Among these, certain attacks involved high query volumes to newly registered domains, which frequently serve as temporary hosts for phishing pages or command-and-control servers.

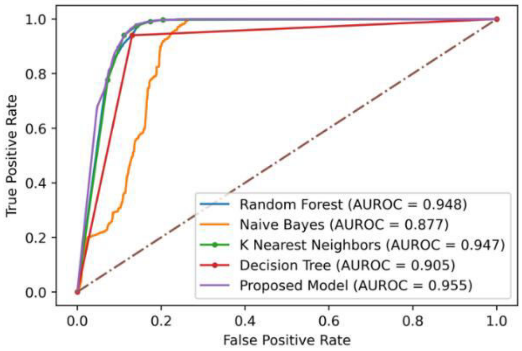


Fig 8. ROC curve for different ML models

The analysis also revealed that a majority of these malicious DNS queries occurred following anomalous timing patterns, such as coordinated bursts of activity aligned with specific events, suggesting a deliberate exploitation of targeted vulnerabilities. Additionally, unsupervised learning methods—including clustering algorithms—were employed to detect anomalies in the DNS traffic without relying on labelled data. These methods successfully identified hidden patterns and grouped similar packets, exposing outliers that were potentially indicative of malicious behaviour.

TABLE 3

Comparison between tradition methodology and the proposed methodology

Feature	Existing System	Proposed Methodology
Detection Method	Relies on blocklists and passive DNS monitoring	Employs real-time machine learning for classification

Threat Detection	Limited in detecting new and unknown threats	Adaptive detection of novel attack vectors using supervised and unsupervised learning
Privacy	Does not explicitly address privacy concerns	Integrates privacy-preserving mechanisms like encrypted Bloom filters and homomorphic encryption
Adaptability	Static and less adaptive to evolving threats	Includes federated learning for decentralized model training and adaptive updates
Real-time Analysis	Passive DNS analysis is not real-time	Real-time analysis of DNS traffic
Focus	Forensic use	Proactive threat blocking and real-time threat detection
Security	Vulnerable to attacks like DNS tunnelling	Aims to mitigate various threats, including DNS tunnelling

This research successfully identified traffic patterns deviating from conventional norms, providing enhanced visibility into the DNS traffic ecosystem. Time series analysis was employed to measure and document shifting patterns in DNS traffic trends. This methodology enabled the systematic tracking of query types, response codes, and total query volume, facilitating a comprehensive understanding of normal network behaviour. The data was visualized using libraries such as Matplotlib and Seaborn, allowing for clear observation of activity spikes and their potential correlations with other network events or security incidents. For instance, a notable spike in DNS requests for specific domains coincided with the public disclosure of a well-known security vulnerability, demonstrating attackers' ability to rapidly exploit emerging opportunities. Behavioural analysis of DNS queries revealed valuable potential for flagging security incidents of concern. Through careful filtering and analysis of packet data extracted from Wireshark, a detailed profile of DNS traffic trends and behaviours was developed, contributing significantly to the broader objective of enhancing DNS security. The findings carry important implications for both immediate threat detection and long-term network security strategy development. They provide a foundation for crafting proactive responses to mitigate risks associated with malicious DNS activity. This research underscores the critical importance of incorporating advanced analytical methods into network security practices to establish more robust and responsive security postures.

V. CONCLUSION

The DNS traffic analysis with machine learning work has been able to make considerable contributions in identifying and categorizing malicious activity in DNS traffic. With the use of supervised and unsupervised learning methods on DNS traffic using Wireshark, the researchers have been able to succeed in demonstrating that it is possible to detect sophisticated threats better than traditional rule-based methods. The study yielded a number of important findings: First, the observation of anomalous distribution between UDP and TCP traffic illustrates the usefulness of protocol-specific analysis in identifying potential lines of attack. Second, packet size and pattern anomaly detection of distributions of IP addresses has been useful to identify potential attempts at data exfiltration and infected hosts. The study yielded several interesting results: First, the identification of anomalous distribution between TCP and UDP traffic illustrates the merit of protocol-specific analysis in identifying potential lines of attack. Second, packet size and pattern anomaly detection of IP address distributions has been useful to identify potential data exfiltration attempts and infected hosts. Third, the machine learning classifiers were able to detect a massive volume of potential DoS attacks and port scans, demonstrating the system's capability to recognize complex patterns of attacks. Detection via DNS query analysis indicated that the model could differentiate between normal access vs. high-value domains and likely malicious activity like misconfigured software or DNS tunnelling attempts. The evidence supports the hypothesis that machine learning would be effective in addressing the dynamic nature of DNS-based attacks. This degree of specificity is important in network security optimization and threat elimination. Using AI on packet characteristics, temporal analysis, and query load analysis, the approach was able to provide context-based real-time threat detection compared to standard security methods. This research is a stepping stone to even more advanced, AI-driven DNS security systems in the future. Future work will be required to further hone the machine learning algorithms to minimize false positives and maximize accuracy and further strengthen the system's ability to recognize new DNS-based attack types. Incorporation of this analysis into full network security systems could make for an even more potent defence against current cyber-attacks.

VI. REFERENCES

[1] Rajaboevich, G. S., Baxtiyorovich, N. N., & Komilovich, T. S. (2021). A model for preventing malicious traffic in DNS servers using machine learning. *2021 International Conference on Information Science and Communications Technologies (ICISCT)*, 1-4.

[2] Kostopoulos, N., Pavlidis, A., Dimolianis, M., Kalogeras, D., & Maglaris, V. (2019). A Privacy-Preserving Schema for the Detection and Collaborative Mitigation of DNS Water Torture Attacks in Cloud Infrastructures. *2019 IEEE 8th International Conference on Cloud Networking (CloudNet)*, 1-6.

[3] Tian, S., Fang, C., Liu, J., & Lei, Z. (2016). Detecting Malicious Domains by Massive DNS Traffic Data Analysis. *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, 130-133.

[4] Singh, S. K., & Roy, P. K. (2020). Detecting Malicious DNS over HTTPS Traffic Using Machine Learning. *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*, 1-6.

[5] Fahim, A., et al. (2024). DNS Exfiltration Guided by Generative Adversarial Networks. *2024 IEEE 9th European Symposium on Security and Privacy (EuroS&P)*, 580-599.

[6] Hasegawa, K., Kondo, D., & Tode, H. (2021). FQDN-Based Whitelist Filter on a DNS Cache Server Against the DNS Water Torture Attack. *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 628-632.

[7] Kumari, A., & Sharma, I. (2024). Integrated RNN-SVM Model for Improved Detection of Imbalanced DNS Heavy Attacks. *2024 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, 337-341.

[8] Kulkarni, O. S., Khurana, J., G, M. G., Naval, P., Kaushik, H., & Renuka, G. (2023). Investigating the Practicality of DNS-Based Authentication for Securing Wireless Networks. *2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*, 1-7.

[9] Ahmed, J., Gharakheili, H. H., & Sivaraman, V. (2022). Learning-Based Detection of Malicious Hosts by Analyseing Non-Existent DNS Responses. *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 3411-3416.

[10] Moubayed, A., Injadat, M., & Shami, A. (2020). Optimized Random Forest Model for Botnet Detection Based on DNS Queries. *2020 32nd International Conference on Microelectronics (ICM)*, 1-4.

[11] Chang, S.-Y., Park, Y., Kengalahalli, N. V., & Zhou, X. (2020). Query-Crafting DoS Threats Against Internet DNS. *2020 IEEE Conference on Communications and Network Security (CNS)*, 1-9.

[12] Suwa, S., Yamai, N., Okayama, K., Nakamura, M., Kawano, K., & Gada, (2012). Spam Mail Discrimination System Based on Behaviour of DNS Servers Associated with URLs. *2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet*, 381-386.

[13] Suga, T., Okada, K., & Esaki, H. (2019). Toward Real-time Packet Classification for Preventing Malicious Traffic by Machine Learning. *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, 106-111.

[14] Ma, X., Li, J., Tao, J., & Guan, X. (2012). Towards active measurement for DNS query behaviour of botnets. *2012 IEEE Global Communications Conference (GLOBECOM)*, 845-849.

[15] van Rijswijk-Deij, R., Rijnders, G., Bomhoff, M., & Allodi, L. (2019). Privacy-Conscious Threat Intelligence Using DNSBLoom. *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 98-106.