B. Tech. CSE Sixth Semester CS350 Mini Project - II Report

On

DNS Query Anomaly Detection

M Shree Harsha Bhat

(211CS137)

Shyam Balaji

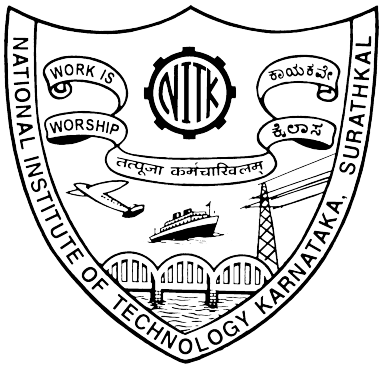
(211CS154)

Talware Abhishek Sandip

(211CS156)

Guide

Dr. Saumya Hegde



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA,

SURATHKAL, MANGALORE - 575025

April, 2024

DECLARATION

I hereby declare that the B. Tech . 6^{th} Semester **CS350 - Mini Project - II** report entitled **DNS Query Anomaly Detection** being submitted to the Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal, in fulfilment of the requirements of the CS350 course is a bona fide report of the work carried out by me. The material contained in this Report has not been submitted to any University or Institution for the award of any degree.

M Shree Harsha Bhat, Shyam Balaji, Talware Abhishek Sandip
211CS137, 211CS154, 211CS156
Department of Computer Science and Engineering
NITK, Surathkal

Place: NITK, Surathkal.
Date: 10/04/2024

CERTIFICATE

This is to certify that the B. Tech. 6th Semester **CS350 - Mini Project - II** report entitled **DNS Query Anomaly Detection** submitted by **M Shree Harsha Bhat**, (Roll Number: 211CS137),**Shyam Balaji**, (Roll Number: 211CS154),**Talware Abhishek Sandip**, (Roll Number: 211CS156) as the record of the work carried out by him/her, is accepted in fulfillment of the requirements of the CS350 course.

Guide

Dr. Saumya Hegde,

Department of Computer Science and Engineering,

NITK, Surathkal

Abstract

Abstract: Domain Name System (DNS) assaults have become a major danger to network security in the modern world, making detection and mitigation difficult. Due to their stealthy nature and ability to exfiltrate data, DNS tunnelling and exfiltration have drawn the most attention among these threats. For security professionals, identifying these attacks is a difficult task since conventional detection techniques frequently fail to recognise malicious activity concealed in normal DNS data.

We have explored the topic of DNS attacks in this mini-project, paying particular attention to DNS tunnelling and exfiltration methods. These advanced attack techniques use DNS protocol to create hidden communication channels or steal confidential information, making them hard to find and stop with traditional security tools.

Recognizing the pressing need for robust detection mechanisms, our contribution lies in the design and implementation of two separate classification models tailored to address these challenges. Firstly, we propose a two-class classifier for DNS tunneling detection. Leveraging various features extracted from DNS query data, our model aims to differentiate between benign and malicious DNS traffic, thereby enhancing the ability to detect and mitigate tunneling activities.

Secondly, we introduce a two-class classifier for DNS exfiltration detection. By analyzing patterns and anomalies in DNS traffic associated with data exfiltration attempts, our model endeavors to identify and classify such activities accurately, providing an additional layer of defense against data breaches and unauthorized data transfers via DNS channels.

Through our research and experimentation, we demonstrate the efficacy of our classification models in detecting DNS tunneling and exfiltration attacks, thereby contributing to the advancement of DNS security and aiding network defenders in their efforts to safeguard against evolving cyber threats. Our model for DNS exfiltartion detection gave accuracy around 99.9% and model for DNS tunnelling gave accuracy very close to 100% due to biased dataset.

Keywords: DNS, DNS tunnelling, DNS exfiltration

Contents

List of Figures

1 Introduction

1.1 Domain Name System

The Domain Name System (DNS) is a distributed hierarchical system that acts as the "phonebook" of the internet, mapping domain names to their corresponding IP addresses. DNS operates across a vast network of servers, encompassing authoritative name servers, recursive resolvers, and caching servers, to ensure efficient and reliable resolution of domain names.

Despite its indispensable role in facilitating internet communication, DNS is not immune to security threats. Various types of attacks exploit vulnerabilities within the DNS infrastructure to disrupt services, compromise data integrity, and facilitate unauthorized access. Common DNS attacks include DNS spoofing, cache poisoning, distributed denial-of-service (DDoS) attacks, and DNS hijacking, among others. These attacks pose significant risks to network security and necessitate robust defense mechanisms to mitigate their impact.

1.2 DNS Tunnelling

One of such attacks on DNS is DNS Tunnelling. DNS tunneling is a sophisticated technique used by attackers to bypass network security measures by encapsulating unauthorized communication within DNS queries and responses. In DNS tunneling, malicious actors leverage the DNS protocol to create covert communication channels, enabling the transmission of unauthorized data payloads disguised as legitimate DNS traffic. By encoding data within DNS query strings or resource records, attackers can establish clandestine channels for command and control, data exfiltration, or malware propagation, effectively evading traditional security controls.

1.3 DNS Exfiltration

DNS exfiltration is a stealthy data exfiltration technique that exploits DNS infrastructure to leak sensitive information from a compromised network to an external attacker-controlled server. In DNS exfiltration attacks, malicious software or threat actors manipulate DNS queries or responses to encode and transmit sensitive data, such as intellectual property, personally identifiable information (PII), or credentials,

to external servers under their control. By leveraging DNS as a covert data transfer mechanism, attackers can evade detection and exfiltrate data without triggering alarms or raising suspicion.

1.4 Team Work

In response to the growing threat posed by DNS-based attacks, we propose the development of a two-class classifier for DNS traffic analysis to distinguish between benign and malicious traffic for DNS Tunnelling attack. Key features include entropy of DNS queries, number of characters, number of uppercase characters, number of numeric characters, and number of labels. Additionally, we propose a separate two-class classifier for DNS exfiltration detection, utilizing a comprehensive set of 17 features derived from DNS traffic patterns and anomalies. By leveraging machine learning techniques, our classifiers aim to enhance the accuracy and efficiency of DNS threat detection, thereby bolstering network security and resilience against evolving cyber threats.

2 Literature Survey

2.1 DNS-over protocols

2.1.1 DNS over HTTPS

DNS over HTTPS (DoH) is a protocol that encrypts DNS traffic by passing DNS queries through an HTTPS encrypted session. HTTP/2 is the minimum recommended version for DoH. A DoH client(a client who supports DoH protocol) encodes a single DNS query into an HTTP request using either the HTTP GET or POST method. The media type used for sending dns queries is "application/dns-message". When the GET method is used, the data payload is encoded with base64url and provided as a variable named “dns” to the URI.

DNS-over-HTTPS operates over TCP, which can retransmit data very quickly in the case of packet losses, whereas traditional DNS clients use UDP and wait for a fixed time before retrying.

2.1.2 DNS over QUIC

DNS-over-QUIC (DoQ) is a protocol that aims to secure and optimize DNS communications by leveraging the QUIC (Quick UDP Internet Connections) protocol. The design goal is to provide DNS privacy with minimum latency, for which DoQ uses QUIC as the underlying transport protocol. QUIC features mandatory encryption, provides multiplexing, and improves connection establishment time by combining the transport and encryption handshakes into a single round trip. QUIC allows you to establish a network connection much faster. As with the "Connection Migration", it’s especially useful when being on mobile. With DNS-over-QUIC implemented, the connection is established twice as fast as with DNS-over-TLS. DoQ already outperforms DoT as well as DoH, making it the best choice for encrypted DNS to date.

2.1.3 DNS over TLS

DNS-over-TLS (DoT) is a protocol that enhances the security and privacy of DNS communications by encrypting DNS queries and responses using the Transport Layer Security (TLS) protocol. It uses TLS to encrypt DNS traffic directly. It operates on a separate port, typically port 853, which is specific to DNS over TLS. DNS clients

initiate a TLS handshake with DNS servers to establish a secure communication channel.By encrypting DNS traffic and authenticating communication channels, DoT helps protect against various security threats, including DNS spoofing, eavesdropping, and tampering.

2.2 Monitoring Enterprise DNS Queries for Detecting Data Exfiltration From Internal Hosts

The paper et al Jawad Ahmed [2019] proposes a real-time detection mechanism for DNS exfiltration and tunneling. By analyzing attributes of DNS traffic and employing machine learning algorithms, the proposed scheme achieves high accuracy in identifying malicious DNS activity.

The DNS traffic, particularly query names, were collected from two enterprise networks were analyzed. The data was obtained by mirroring the Internet traffic from the border routers of the networks to a data collection system. Attributes included count of characters, entropy, and label patterns within fully qualified domain names (FQDNs). For anomaly detection, machine learning models were trained using benign data from the dataset, with ground truth drawn from reputable domain lists like the Majestic Million.

2.3 EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis

This paper et al Leyla Bilge [2014] introduced a system that employs large-scale, passive DNS analysis techniques to detect domains that are involved in malicious activity.

The Data Collector module captured DNS traffic occurring within the network under observation. This entailed recording various facets of DNS transactions, including the domain names being queried, the timing of these queries, response times, and additional metadata. Operating in a passive manner, the Data Collector observed DNS traffic without introducing disruptions to the network's regular operations. It employed techniques such as packet sniffing or network taps to intercept DNS packets for subsequent analysis.

The collected features encompasses patterns in query volume, frequencies of queries

for particular domains, shifts in query behaviors over time, among others. The Feature Attribution module then attributes these extracted features to the respective domains within the dataset. The supervised learning were employed. The classification algorithms such as **Decision Trees, Support Vector Machines, or Neural Networks** were used. The output of the Classifier dictates the classification of each domain, facilitating the identification and mitigation of potential threats within the network.

2.4 Mapping an Enterprise Network by Analyzing DNS Traffic

The paper et al Minzhao Lyu [2019] developed an automated method to map internal hosts of an enterprise network by focusing only on DNS traffic. By capturing and analyzing DNS traffic in/out of the organization, they dynamically and continually identified the DNS resolvers, DNS name-servers, (non-DNS) public-facing servers, and regular client hosts behind or not behind the NAT in the enterprise.

They developed K-means unsupervised algorithm to determine if an enterprise host with a given DNS activity is a "name server", "recursive resolver", "mixed DNS server", or a "regular end-host".

Parameters used were: query fraction of all outgoing DNS packets (QryFraqOut), fraction of total number of external servers queried (numExtSrv) per individual enterprise host, fraction of total number of external hosts that initiate query in (numExtClient) per individual enterprise host etc.

3 Experimental Setup

This section will brief about the experimental setup for the machine learning algorithm implementation for DNS Exfiltration detection. There are two major sections - two implementations for different DNS attacks.

3.1 DNS tunnelling detection

Here we will talk about the machine experimental setup for the machine learning algorithm on this dataset.

3.1.1 Dataset Description

The features included in data set are:

- hostname
- label(benign or malicious)

3.1.2 Data Preprocessing

3.1.2.1 Test Data The test dataset has a total of 14999 records and 3000 of them were benign while 11999 were malicious. To ensure that there are no biases while training our model we randomly select 3000 entries that are malicious so that the number of benign and malicious entries are same.

3.1.2.2 Train Data The train dataset has a total of 4999 records and 1000 of them were benign while 3999 were malicious. To ensure that there are no biases while training our model we randomly select 1000 entries that are malicious so that the number of benign and malicious entries are same.

3.1.2.3 Making the Dataframe : Parameters chosen for classification Since only the hostname of the DNS query is given we need to extract some features from it manually. The following is the list of features from the hostname which will be added to the dataframe:

1. **Entropy**: Entropy measures the randomness or uncertainty in the domain name. Higher entropy values may indicate irregular or suspicious patterns, which could be associated with tunneling or other malicious activities.

2. **Number of characters in query**: The length of the domain name can be indicative of certain types of malicious behavior. Malicious domain names may be longer or shorter than regular domain names, depending on the specific attack technique employed.

3. **Number of uppercase characters**: The presence of uppercase characters in domain names may be indicative of obfuscation or encoding attempts by attackers.

4. **Number of numeric characters**: Similar to uppercase characters, the presence of numeric characters in domain names can be associated with certain types of malicious behavior.

5. **Number of labels**: The number of labels refers to the number of domain name labels separated by periods (e.g., subdomains). This attribute provides insights into the structure and complexity of the domain name.

3.1.3 Methodology

The pre-processed data is used for training of classification models. We have used **Decision Tree Classifier**, **Naive Bayes** and **Support Vector Machines** for classification of the DNS quesries.

A Decision Tree Classifier is a natural choice due to its versatility and interpretability. It can handle both numerical and categorical data effortlessly while automatically performing feature selection. This makes it particularly useful in scenarios where there may be irrelevant or redundant features.

Naive Bayes offers computational efficiency, requiring only a small amount of training data to estimate its parameters. Naive Bayes can effectively handle both numerical and categorical features present in the dataset. Its probabilistic nature makes it adept at estimating class probabilities, making it a valuable tool for binary classification tasks like the one at hand.

SVM can handle nonlinear relationships between features and the target variable. This adaptability, combined with its capacity to handle high-dimensional data effectively, makes SVM a strong contender for the binary classification task.

3.2 DNS exfiltration detection

3.2.1 Dataset Description

The dataset utilized for this project was sourced from the internet. The decision to use an internet-sourced dataset was made so that we are ready with the machine learning model to test on real data when it is made available. The features included in data set are:

- user_ip (IP Address): This feature likely captures the source IP address of the DNS request. It can be useful in identifying patterns of malicious behavior associated with certain IP addresses.
- domain (top level domain): The domain name itself can provide valuable information about the nature of the request. Malicious domains often exhibit patterns that differ from those of benign domains.
- timestamp: The timestamp indicates when the DNS request occurred. Analyzing the timing of requests can reveal patterns of malicious activity, such as sudden spikes in activity or regular intervals characteristic of botnet communication.
- attack (True for malicious, false otherwise): This is the target variable indicating whether the DNS request is malicious or benign. It is used for supervised learning to train the classifier.
- request (original dns query): The original DNS query contains the actual query made by the user. Analyzing the content of the query can reveal suspicious patterns or keywords associated with malicious activity.
- len (length of the request - without TLD): The length of the DNS request without the top-level domain (TLD) may be indicative of certain types of malicious behavior, such as domain flux or DGA-generated domains, which often have longer or more complex names.

- subdomains_count (number of subdomains without TLD): Similar to the length of the request, the number of subdomains can provide insights into the complexity and structure of the domain name, which may be relevant for identifying malicious domains.

- w_count (number of English words in the request): This feature counts the number of English words in the DNS request. Malicious requests may exhibit different linguistic characteristics compared to benign ones.

- w_max (length of the longest English word in the request): The length of the longest English word in the request may capture the complexity or sophistication of the query, which could be relevant for distinguishing between malicious and benign requests.

- entropy (entropy of DNS request): Entropy measures the randomness or uncertainty in the DNS request. Higher entropy values may indicate more irregular or suspicious patterns in the request.

- w_max_ratio (longest English word length to request length ratio): This ratio normalizes the length of the longest English word by the total length of the request. It may help identify anomalous linguistic patterns associated with malicious activity.

- w_count_ratio (number of English words to request length ratio): Similar to the previous feature, this ratio normalizes the number of English words by the total length of the request, providing insights into linguistic patterns.

- digits_ratio (percentage of digits in the request): The presence of digits in the DNS request may be indicative of certain types of malicious behavior or encoding schemes used by attackers.

- uppercase_ratio (percentage of capital letter in the request): Similarly, the presence of uppercase letters may be indicative of certain patterns associated with malicious activity, such as obfuscation or encoding.

The following features are calculated using the current and the previous 9 requests (window size = 10). Requests are grouped by (user_ip, domain) key.

	user_ip	domain	timestamp	attack	request	len	subdomains_count	w_count
3942	186.169.146.147	e5.sk	1624438294225	1	sebubx76xk4erpp3rwehoo3ubmbqeaqbaeaq.a.e.e5.sk	40	3	3
4297	186.169.146.147	e5.sk	1624438295586	1	4az3kiecotwu3okbtvfm7pdpcabqeaqbaeaq.a.e.e5.sk	40	3	5
4590	186.169.146.147	e5.sk	1624438296656	1	x3i2wbqsiucuviqyfaaoxz3lzybqeaqbaeaq.a.e.e5.sk	40	3	1
6096	186.169.127.58	e5.sk	1624438302237	1	ez2vzwchw3ce5m6wz6cw3nnc2ibqeaqbaeaq.a.e.e5.sk	40	3	1
6187	186.169.146.147	e5.sk	1624438302672	1	htm7xrligq2enc4lsjhkzdnd6mbqeaqbaeaq.a.e.e5.sk	40	3	3
6495	186.169.127.58	e5.sk	1624438303710	1	f4clwtzqaonejfevfnc3vnm334bqeaqbaeaq.a.e.e5.sk	40	3	2
6724	186.169.127.58	e5.sk	1624438304691	1	hshm7dgsfuvungjbsgjocfazoibqeaqbaeaq.a.e.e5.sk	40	3	3
6968	186.169.127.58	e5.sk	1624438305372	1	uk7xg4v2usyupazkwfjietmf3ybqeaqbaeaq.a.e.e5.sk	40	3	2
7721	186.169.127.58	e5.sk	1624438308174	1	ijjuunvalweehk2jgbquu2atwabqeaqbaeaq.a.e.e5.sk	40	3	6
7722	186.169.127.58	e5.sk	1624438308181	1	mnwmw2m3timeblpdxzjqmnvf3ibqeaqbaeaq.a.e.e5.sk	40	3	3

Figure 1: Initial few columns of dataset

digits_ratio	uppercase_ratio	time_avg	time_stdev	size_avg	size_stdev	throughput	unique	entropy_avg	entropy_stdev
0.125	0.0	2197.222222	2875.261022	48.2	53.370404	24.372977	0.0	3.691242	0.910175
0.100	0.0	2348.444444	2779.448601	48.2	53.370404	22.803615	0.0	3.685581	0.906808
0.075	0.0	2460.111111	2695.151964	51.8	51.228898	23.394454	0.0	3.884313	0.687639
0.175	0.0	1799.222222	1935.781934	44.0	27.712813	27.170557	0.0	3.835620	0.663023
0.100	0.0	3105.444444	2782.422466	51.8	51.228898	18.533095	0.0	3.905225	0.700116
0.125	0.0	1382.111111	1447.797417	44.0	27.712813	35.369775	0.0	3.824709	0.660105
0.025	0.0	1327.555556	1453.227538	44.0	27.712813	36.823165	0.0	3.813797	0.653225
0.100	0.0	852.555556	514.388256	44.0	27.712813	57.336461	0.0	3.861596	0.674604
0.050	0.0	1163.888889	734.667347	47.6	24.033310	45.437190	0.0	4.036861	0.192920
0.075	0.0	1006.777778	819.147389	40.0	0.000000	44.140366	0.0	4.010757	0.165480

Figure 2: Last few columns of dataset

- time_avg (average time between requests)
- time_stdev (standard deviation of times between requests)
- size_avg (average size (length) of the requests)
- size stdev (standard deviation of sizes of requests)
- throughput (number of characters in requests transmitted per second)
- unique (uniqueness indicator with values in range [0-1] (0 - all requests are equal, 1 - all requests are different))
- entropy_avg (average value of entropy)
- entropy_stdev (standard deviation of entropy)

3.2.2 Data Preprocessing

Figure 1 and Figure 2 describes how the dataset looks like. The dataset has **349558** entries and **22** features. The dataset has equal no of entries for benign and malicious

DNS queries that is **174779** in number.

For feeding the data into model for training we have to eliminate the text based features or convert them to numerical format. The features 'request' and 'domain' are already converted into numerical format through features like 'entropy' , 'w_count' etc. So 'request' and 'domain' can be eliminated. the feature 'user_ip' and 'timestamp' have been used for statistical features. So even they can be eliminated. Thus out of 22 features, we have eliminated 4 features and one of the feature ('attack') acts as label for supervised learning.

The processed dataset has been split into train data and test data using standard libraries in the ratio 4:1 respectively.

3.2.3 Methodology

The pre-processed data is used for training of classification models. We have used **Decision Tree Classifier** and N̂aive Bayes for classification of the DNS quesries.

Decision trees can effectively handle nonlinear decision boundaries in high-dimensional feature spaces. They recursively partition the feature space into regions based on feature values, allowing for the creation of complex decision boundaries that can adapt to the data distribution.Decision trees are robust to irrelevant features and can automatically select the most informative features for classification.

Similarly, Naive Bayes classifiers are computationally efficient and well-suited for handling large datasets with high-dimensional feature spaces. So these classification models were used for the purpose. Standard python libraries and framework are used for training and prediction.

Thus this is the experimental setup for testing on two different datasets for DNS tunnelling and DNS exfiltration detection.

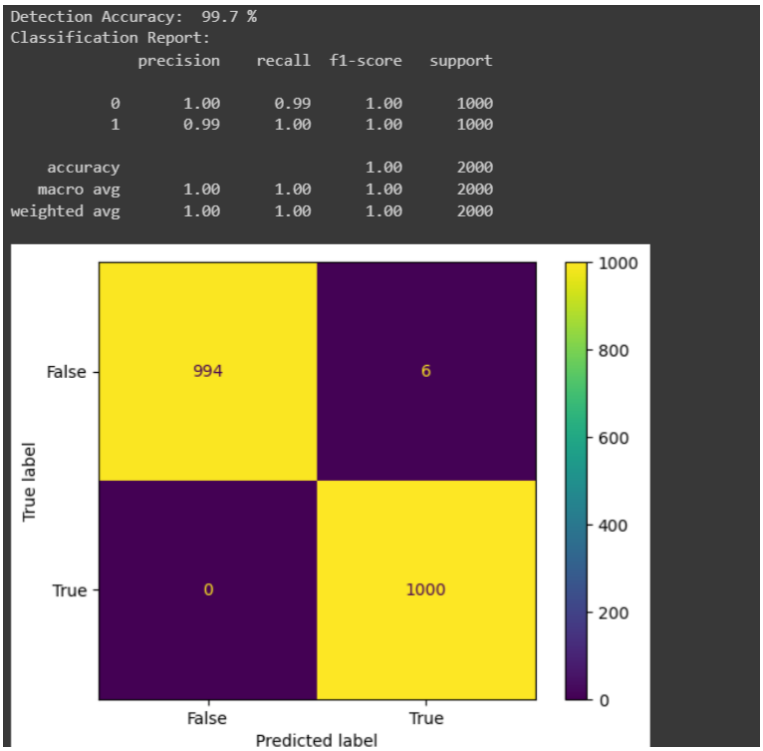


Figure 3: Results of Decision Tree classifier - tunnelling

4 Results

This section will discuss the results obtained on the test datasets. The predictions made by the models post training are compared with existing label of test data and correspondingly accuracy, classification report and heat maps are generated. This has two sections discussing about the results of the DNS tunnelling and DNS exfiltration detection models.

4.1 DNS Tunnelling

4.1.1 Decision Tree Classifier

The result after testing the Decision Tree Classifier model is shown in Figure 3. The accuracy for Decision Tree Classifier model on an average is around 99.7%

4.1.2 Naive Bayes

The result after testing the Naive bayes model is shown in Figure 4. The accuracy for Naive bayes model on an average is around 99.85%.

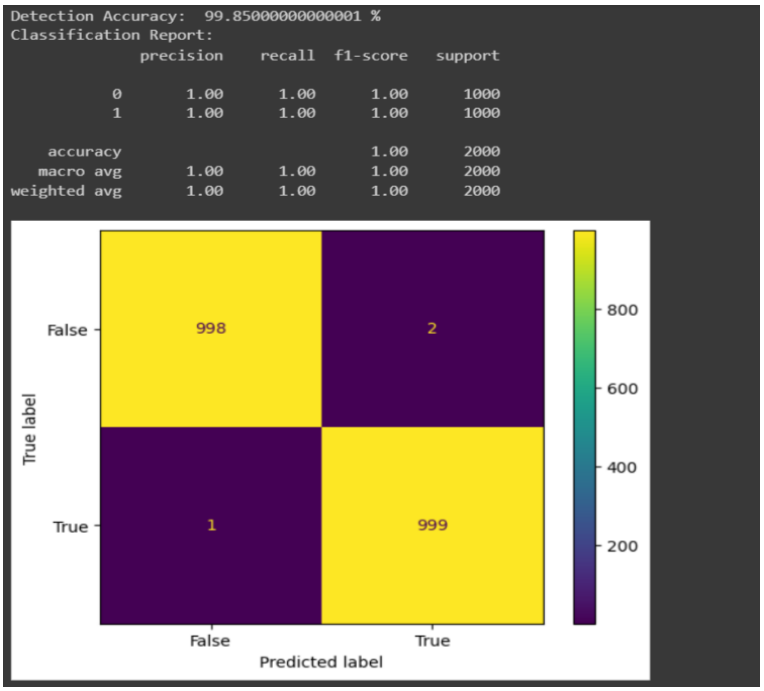


Figure 4: Results of Naive Bayes - tunnelling

4.1.3 Support Vector Machines

The result after testing the Support Vector Machines model is shown in Figure 5. The accuracy for Support Vector Machines model on an average is around 100%

4.2 DNS Exfiltration

4.2.1 Naive Bayes

The result after testing the Naive bayes model is shown in Figure 6. The accuracy for Naive bayes model on an average is around 90%.

4.2.2 Decision Tree Classifier

The result after testing the Decision Tree Classifier model is shown in Figure 7. The accuracy for Decision Tree Classifier model on an average is around 99.9%. Decision tree performed much better than Naive bayes on this data set.

The Figure 8 shows the trade-off between the true positive rate (sensitivity) and the false positive rate. The Area Under the ROC Curve (AUC) quantifies the overall performance of the classification model across all possible threshold settings. A higher AUC value (closer to 1) indicates better discrimination ability, as the model can distinguish between positive and negative instances effectively.

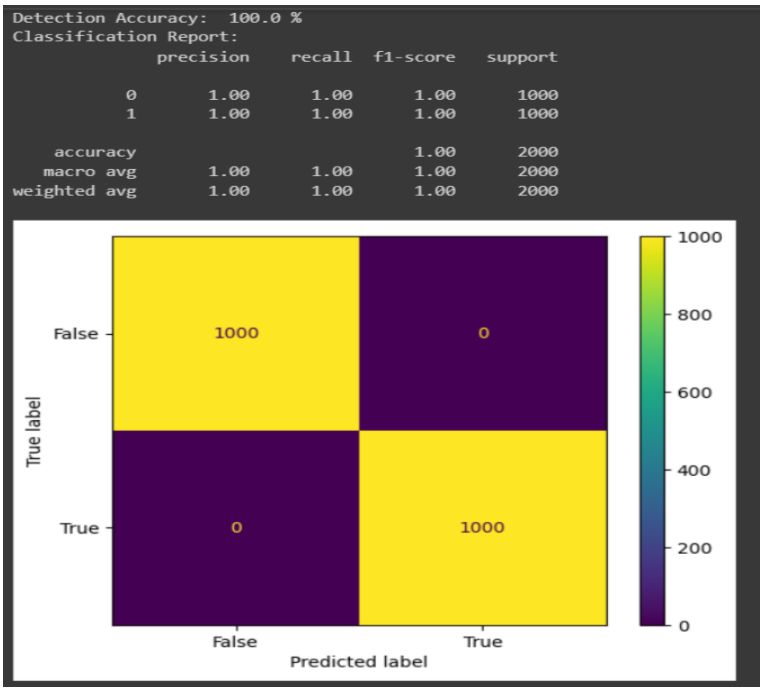


Figure 5: Results of Support Vector Machines - tunnelling

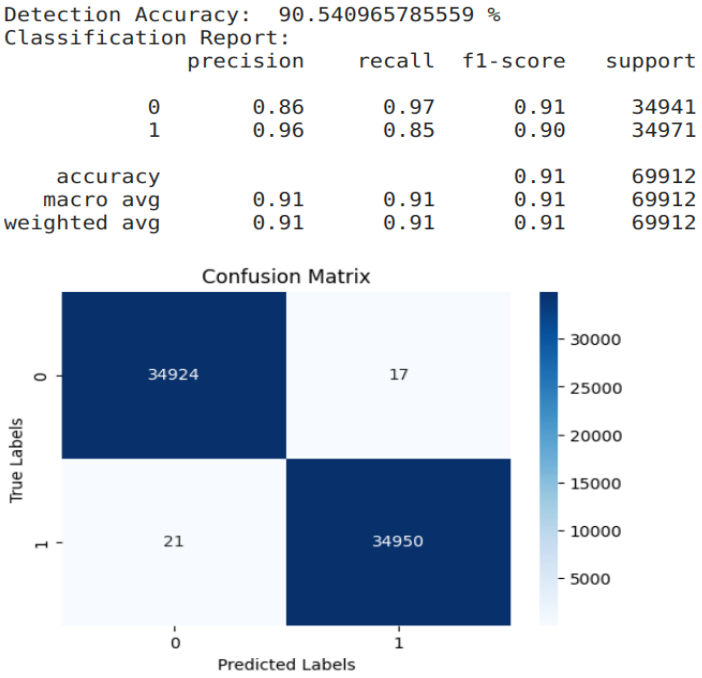


Figure 6: Results of Naive Bayes - exfiltration

```
Detection Accuracy:  99.94564595491475 %
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     34941
           1       1.00      1.00      1.00     34971

    accuracy                           1.00     69912
   macro avg       1.00      1.00      1.00     69912
weighted avg       1.00      1.00      1.00     69912
```

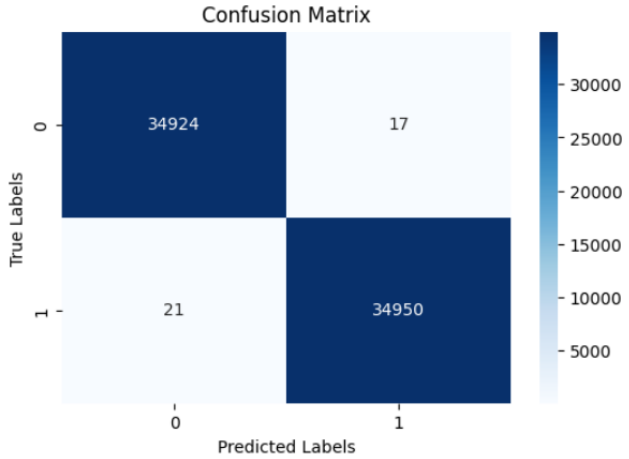


Figure 7: Results of Decision Tree classifier - exfiltration

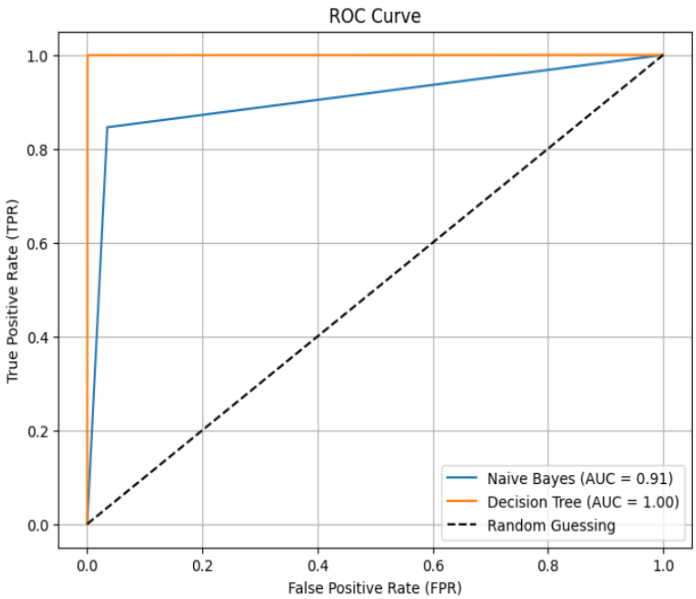


Figure 8: Receiver Operating Characteristic (ROC) curve - exfiltration

5 Conclusions and Future Work

In this project, our primary objective was to develop a pipeline or machine learning model capable of detecting DNS Tunnelling and DNS Exfiltration, while also classifying DNS queries as either benign or malicious. Leveraging machine learning techniques, we successfully designed models for DNS Exfiltration detection, utilizing Decision tree classifier and Naive Bayes classifiers. Notably, the decision tree classifier exhibited exceptional performance, achieving an impressive 99.9% accuracy rate. And the model for the DNS Tunnelling utilized Decision Tree Classifier, Naive Bayes Classifier and Support Vector Machines. **Support Vector Machine** had the best accuracy with **100%**. The reason for this high accuracy was that the dataset had few flaws which lead to biases such as long host names map to malicious.

Looking ahead, there are several avenues for future exploration and enhancement. One such direction involves the collection and in-depth analysis of real DNS traffic from our organization, NITK. By capturing and examining the attributes of DNS messages, we can further refine and validate our model for identifying DNS attacks.

Additionally, conducting simulations of DNS exfiltration attacks will allow us to demonstrate the efficacy of our model in detecting such malicious activities with high accuracy.

In conclusion, our project lays a solid foundation for the detection and classification of DNS-related threats, offering valuable insights and potential solutions for enhancing cybersecurity measures.

Bibliography

Qasim Raza Craig Russell Vijay Sivaraman Jawad Ahmed, Hassan Habibi Gharakheili. Monitoring enterprise dns queries for detecting data exfiltration from internal hosts. In *IEEE Transactions on Network and Service Management*, volume 17, pages 265–279, 2019. doi: 10.1109/TNSM.2019.2940735. URL https://ieeexplore.ieee.org/document/8832271. 2.2

Davide Balzarotti Engin Kirda Christopher Kruegel Leyla Bilge, Sevil Sen. Exposure: A passive dns analysis service to detect and report malicious domains. In *ACM Transactions on Information and System Security*, volume 16, page 1–28, 2014. URL https://dl.acm.org/doi/10.1145/2584679. 2.3

Craig Russell Vijay Sivaraman Minzhao Lyu, Hassan Habibi Gharakheili. Mapping an enterprise network by analyzing dns traffic. In *International Conference on Passive and Active Network Measurement*, 2019. doi: 10.1007/978-3-030-15986-3_9. URL https://www.researchgate.net/publication/331692097_Mapping_an_Enterprise_Network_by_Analyzing_DNS_Traffic. 2.4