Name: Jayakrishnan Lakshmanan

Student ID: 24244457

Date of Submission: 19/May/2025

Module: CS6132

Assignment: 2

Leveraging AI/ML for Robust Detection of DNS Tunnelling Attacks

Jayakrishnan Lakshmanan – Student ID: 24244457

Dissertation in partial fulfilment of the requirements of the degree of

MSc in Artificial Intelligence

submitted to the

[department name]

Supervisor: Dr. Lubna Luxmi Dhirani

[Month, Year]

1. Introduction

Domain Name System (DNS) is one of the fundamental protocols of the Internet. It is a hierarchical, distributed database that translates human-friendly domain names (e.g., example.com) to machine-friendly IP addresses (e.g., 192.254.192.254) (Mockapetris, 1987). DNS is often called the "phonebook of the Internet," enabling users to access websites without memorising the IP addresses (Palo Alto Networks, no date).

One of the first steps in a network communication is a DNS query. It is common to overlook DNS as a potential security risk and allow the DNS traffic without proper security checks. DNS is not intended for data exchange; it works over UDP port 53. Attackers exploit this trust and misuse DNS for data exfiltration, infiltration, and establishing command and control channels (C2).

This dissertation aims to explore the use of AI/ML technologies to detect DNS tunnelling attacks. DNS tunnelling is a cyberattack technique that exploits the DNS protocol to bypass traditional network security measures. Attackers use DNS tunnelling to encapsulate non-DNS traffic within DNS queries and responses (Wang et al. 2021). It is tough to detect these attacks due to their covert and stealthy nature. This chapter outlined the motivation behind the research, the goals, and an overview of the structure of the dissertation.

1.1 Motivation

As we discussed earlier, DNS is one of the building blocks of the Internet. DNS is often trusted and rarely scrutinised. DNS Tunnelling is a covert channel used by threat actors to bypass traditional security measures and exfiltrate data using DNS queries and responses. Traditional security systems often fail to recognise these attacks due to the benign nature of DNS traffic.

The Global DNS Threat Report issued by IDC in 2022, after surveying a number of companies, has revealed that 88% of the companies surveyed have been impacted by DNS attacks, and 28% of these attacks were DNS tunnelling-based ones (IDC, 2023). Kumar & Sood (2020) explained the need for advanced detection mechanisms leveraging AI/ML for detecting these sophisticated cyber threats.

1.2 Research Problems and Goals

This research aims to investigate how machine learning algorithms can be effectively utilised to detect DNS tunnelling attacks. The goal is to design and evaluate AL/ML-based models to classify tunnelling and legitimate DNS traffic with high accuracy and low false positive rate.

1.3 Research Questions

- What DNS traffic features are most indicative of DNS tunnelling behaviour?
- Can ML models properly classify DNS tunnelling traffic and legitimate DNS requests?
- How do different ML algorithms perform on this task?

1.4 Dissertation Structure

The structure of the dissertation is as follows:
Chapter 2, Background, introduces key concepts such as the DNS protocol, common data types, DNS hierarchy, DNS tunnelling, and machine learning fundamentals.
Chapter 3 Related Works, reviews existing papers and literature on DNS tunnelling detection and AL/ML-based solutions for detection.
Chapter 4 Methodology explains the simulation setup, data collection, feature extraction, model design, and evaluation approach.
Chapter 5 Results and Discussion presents the performance analysis and interpretation of findings.
Chapter 6: Conclusion and Future Work summarizes the research, discusses the limitations, and discusses the scope for future research

2. Background

2.1 Domain Name System (DNS)

DNS is one of the fundamental protocols of the Internet. It is a hierarchical, distributed database that translates human-friendly domain names (e.g., example.com) to machine-friendly IP addresses (e.g., 192.254.192.254) (Mockapetris, 1987). DNS is ubiquitous and stateless, which makes it convenient for covert communication.

2.1.1 Common DNS Record Types

The DNS uses various record types to store various resource information associated with the domain. It is important to understand these record types to analyse the DNS traffic in the context of detecting anomalies such as DNS tunnelling.
Attackers often abuse specific record types such as TXT, NULL, and CNAME to encode payloads due to their flexible and less restrictive data fields (Cheng et al., 2020). AL/ML-based detection models can leverage the frequency and structure of these records as input features to distinguish between benign and malicious DNS traffic

A (Address Record)
AAAA (IPv6 Address Record)
CNAME (Canonical Name Record)
MX (Mail Exchange Record)
NS (Name Server Record)

PTR (Pointer Record)
SOA (Start of Authority Record)
TXT (Text Record)
NULL (Null Record)

<**TO DO:** Short description of the Records, considering a table>

2.1.2 DNS Hierarchy

DNS is a distributed, hierarchical system that maps human-readable domain names to machine-readable IP addresses. The DNS namespace is structured in a tree-like hierarchy, consisting of several key components:

- **Root Zone**: At the top of the hierarchy is the root zone, managed by the Internet Assigned Numbers Authority (IANA). The root contains pointers to all Top-Level Domains (TLDs), such as .com, .org, and country code TLDs like .ie, .co, .uk, etc.
- **Top-Level Domains (TLDs):** These are the next level in the hierarchy and include generic TLDs (gTLDs) like .com, .net, and .org, as well as country-code TLDs (ccTLDs). Each TLD is managed by a registry that is responsible for delegating domains beneath it.
- **Second-Level Domains**: These are domains registered by individuals or organizations beneath a TLD, such as example.com.
- **Subdomains**: These are optional divisions of a domain, created by the domain owner to organize or manage services (e.g., mail.example.com, vpn.example.com).
- **Authoritative Name Servers**: These servers store DNS records for a domain and provide final answers to DNS queries.

2.1.3 DNS Resolution Process

The DNS resolution process starts when a user enters a human-readable domain name, such as www.google.com, in their browser. The name resolution process, domain name to IP Address translation, typically involves the following steps:

1. **Query Initiation:** The DNS resolution starts at the user's device, where the DNS resolver checks the local DNS cache for a record. If found, it returns the IP address immediately.
2. **Recursive Resolver**: If the local cache doesn't contain the record, the resolver sends a query to a recursive DNS server, often operated by an ISP or third-party providers like Google (8.8.8.8) or Cloudflare (1.1.1.1).

3. **Root Server Lookup**: If necessary, the recursive resolver queries one of the root DNS servers to find the appropriate TLD server.
4. **TLD Server Query**: The root server responds with the address of the appropriate TLD server (e.g., for .com).
5. **Authoritative Name Server Lookup**: The resolver queries the TLD server, which returns the authoritative name server for the requested domain.
6. **Record Retrieval**: Finally, the recursive resolver queries the authoritative name server, which returns the requested DNS record (e.g., an A or AAAA record containing the IP address).
7. **Response to Client**: The resolver caches the result and sends the IP address back to the user's device.

The DNS resolution process is very fast and typically takes only milliseconds to complete.

2.2 DNS Tunnelling

2.2.1 What is DNS Tunnelling?

DNS tunnelling is a **covert communication technique** that exploits the Domain Name System (DNS) protocol to bypass network security controls. Since DNS is a trusted protocol used for domain-to-IP resolution, it is rarely inspected for malicious activity, making it an attractive vector for attackers (Wang et al., 2021).

DNS tunnelling involves encoding data within DNS queries and responses, effectively creating a bidirectional communication channel. This method is commonly used for:

- Data Exfiltration
- Data Infiltration
- Command and Control (C2) communication

2.2.2 How does DNS Tunnelling Works

DNS Tunnelling requires:

1. **A controlled domain** (e.g., evilsite.example.com).
2. **A malicious DNS server** (acting as an authoritative server for the domain).
3. **Tunnelling tools** (e.g., iodine, dnscat2, dns2tcp).

Example Attack Flow:

1. **Data Encoding:**
   - The attacker encodes stolen data into subdomains (e.g., <encoded_data>.evilsite.example.com).
   - Common encoding methods include **Base32, Base64, or hexadecimal** (Palau et al., 2019).
2. **DNS Query Transmission:**
   - The infected host sends DNS queries to the attacker's server.
   - Since DNS uses **UDP port 53**, these requests often bypass security checks.
3. **Response and Data Extraction:**
   - The attacker's server decodes the data and may send commands via DNS responses (e.g., using **TXT or NULL records**).

2.2.3 DNS Tunnelling Detection Challenges

DNS tunnelling is difficult to detect due to various factors, including legitimate-looking traffic, low throughput, and encryption. Attackers encode the data mimicking normal DNS queries using tools like 'DNSExfilterator', which can send minimal traffic to evade detection. There are even tools available to encrypt the traffic, making the payload inspection ineffective.

2.2.4 DNS Tunnelling Detection Approaches

Rule Based Detection (Traditional methods).

1. **Signature-based**: Scan for known tunnelling tool patterns, for example, Iodine's unique packet headers.
2. **Threshold-based**: Scan for higher rate of anomalies like long domain names, high entropy (random subdomains), and excessive uncommon record types.

Machine Learning based detection.

Various AI/ML-based detection methods are suggested in various studies. These methods excel in detecting newer attack patterns that are not detected by the traditional methods. A number of such studies are discussed in the next section.

3. Related Works

Similar research has been done in detecting DNS tunnelling using traditional detection techniques, statistics, heuristic methods, and AI/ML-based detection approaches.

3.1 Traditional Detection Techniques

Early detection systems relied on rule-based and signature-based methods, which fail to generalize to unknown attack variants. These systems struggle with encrypted payloads and high volumes of DNS traffic (Liu et al., 2018).

3.2 Statistical and Heuristic Methods

Several works used statistical analysis of query lengths, entropy, and query frequency to distinguish abnormal DNS behaviour (Wang et al., 2021). While effective to an extent, they often lack adaptability to evolving attack patterns.

3.3 ML-Based Detection Approaches

Gao et al. (2023) presented an efficient approach to detect DNS Tunnelling in GraphTunnel: Robust DNS Tunnel Detection Based on DNS Recursive Resolution Graph. The author proposes a Graph Neural Networks (GNN) based method which effectively captures spatio-temporal features. They generate a graph structure using DNS query paths, where each node represents a unique domain and contains the extracted features like entropy, TTL, etc. These graphs are analysed using GraphSAGE, a GNN algorithm for classification. While this model demonstrates high accuracy, the performance may depend on the empirically set graph size (K=20) and may not generalize well on all environments.

Wang et al. (2021) presented a thorough overview of various DNS Tunnelling detection techniques in A Comprehensive Survey on DNS Tunnel Detection. The authors classified detection methods into rule-based (signature and threshold) and model-based (deep learning and machine learning) approaches. They identified that the rule-based methods are relatively easy to deploy, but the signatures must be updated to keep up with the new tunnelling techniques. Meanwhile, model-based methods offer superior performance but require large datasets and computational resources to train. This survey shows the need for hybrid approaches combining the strength of both methods to balance performance and practicality.

Machmeier et al. (2024) presented a novel approach considering DNS Tunnelling attacks as time series anomalies. They used k-nearest-neighbour (kNN) detection capabilities with Dynamic Time Wrapping (DTW) as the distance metric. The proposed

model efficiently classified benign and malicious DNS Traffic by analysing features such as name entropy, packet size, and so on over a specific period. The model uncovers the effectiveness of time series analysis in detecting low throughput attacks like DNS Tunnelling.

Palau et al. (2019) proposed a 1D Convolutional Neural Network (CNN) to detect DNS Tunnelling by analysing lexicographical features of domain names such as character patterns and entropy. The authors proposed an embedding layer for character-level feature extraction, a convolution layer to detect n-gram patters and a dense layer for classification. The model achieves 92% detection accuracy with a .8% false positive rate. The study also mentions about collecting the data on a VM based lab environment which his easy to replicate.

3.4 Challenges in the Existing Approaches

Gao et al. (2023) introduced an efficient GNN-based method, but dependency on recursive resolution data and computational requirements could be a drawback in specific networks.

Wang et al. (2021) provided a comprehensive survey of detection methods but overlooked the real-world deployment challenges.

Machmeier et al. (2024) presented a novel kNN model with DTW. However, the computational demands may hinder real-time applications in high-traffic networks.

Palau et al. (2019) proposed a 1D Convolutional Neural Network (CNN). The model achieved decent accuracy but struggles for low throughout attack scenarios.

References

Cheng, J., Cao, Y., Li, H., and Shen, C. (2020) 'A Survey of DNS Tunnelling: Attacks, Detection, and Counter measures', *IEEE Access*, 8, pp. 177415-177431.

Gao, G., Niu, W., Gong, J., Gu, D., Li, S. and Zhang, M. (2023) 'GraphTunnel: Robust DNS tunnel detection based on DNS recursive resolution graph', *IEEE Transactions on Information Forensics and Security*, 19, pp.7705-7719. doi: 10.1109/TIFS.2024.3443596.

IDC. (2023) *2023 Global DNS Threat Report* Available at: https://efficientip.com/resources/cyber-threat-intelligence-idc-2023-global-dns-threat-report/ (Accessed: 17 May 2025).

Kumar, A. and Sood, S. K. (2020) 'Advanced Machine Learning based Detection and Classification of DNS Tunnel', *Journal of Network and Computer Applications*, 157, 102564.

Liu, X., Chen, X., Wang, Y., Wang, H. and Zhang, Y. (2018) 'Detecting DNS Tunnels Using Statistical and Machine Learning Methods', *Security and Communication Networks*, 2018, pp. 1–9.

Machmeier, S. and Heuveline, V. (2024) 'Detecting DNS Tunnelling and Data Exfiltration Using Dynamic Time Warping', 2024 *8th Cyber Security in Networking Conference (CSNet)*, Paris, France, 04-06 December 2024. Paris: IEEE, pp. 83-91.

Mockapetris, P. (1987) *Domain Names – Implementation And Specification*. RFC 1035. Available at: https://datatracker.ietf.org/doc/html/rfc1035 (Accessed: 03 May 2025).

Palau, F., Catania, C., Guerra, J., Garcia, S. and Rigaki, M. (2019) 'DNS tunneling: A deep learning based lexicographical detection approach', *arXiv:2006.06122*, v2. Available at: https://10.48550/arXiv.2006.06122 (Accessed: 5 may 2025).

Palo Alto Networks. (no date) *What is DNS Tunnelling?* Available at: https://www.paloaltonetworks.com/cyberpedia/what-is-dns-tunneling (Accessed: 02 May 2025).

Wang, Y., Zhou, A., Liao, S., Zheng, R., Hu, R. and Zhang, L. (2021) 'A comprehensive survey on DNS tunnel detection', *Computer Networks*, 197(C), pp. 19. doi: 10.1016/j.comnet.2021.108322.

Bibliography

Cheng, J., Cao, Y., Li, H., and Shen, C. (2020) 'A Survey of DNS Tunnelling: Attacks, Detection, and Counter measures', *IEEE Access*, 8, pp. 177415-177431.

Gao, G., Niu, W., Gong, J., Gu, D., Li, S. and Zhang, M. (2023) 'GraphTunnel: Robust DNS tunnel detection based on DNS recursive resolution graph', *IEEE Transactions on Information Forensics and Security*, 19, pp.7705-7719. doi: 10.1109/TIFS.2024.3443596.

IDC. (2023) *2023 Global DNS Threat Report* Available at: https://efficientip.com/resources/cyber-threat-intelligence-idc-2023-global-dns-threat-report/ (Accessed: 17 May 2025).

Kumar, A. and Sood, S. K. (2020) 'Advanced Machine Learning based Detection and Classification of DNS Tunnel', *Journal of Network and Computer Applications*, 157, 102564.

Liu, X., Chen, X., Wang, Y., Wang, H. and Zhang, Y. (2018) 'Detecting DNS Tunnels Using Statistical and Machine Learning Methods', *Security and Communication Networks*, 2018, pp. 1–9.

Machmeier, S. and Heuveline, V. (2024) 'Detecting DNS Tunnelling and Data Exfiltration Using Dynamic Time Warping', 2024 *8th Cyber Security in Networking Conference (CSNet)*, Paris, France, 04-06 December 2024. Paris: IEEE, pp. 83-91.

Mockapetris, P. (1987) *Domain Names – Implementation And Specification*. RFC 1035. Available at: https://datatracker.ietf.org/doc/html/rfc1035 (Accessed: 03 May 2025).

Palau, F., Catania, C., Guerra, J., Garcia, S. and Rigaki, M. (2019) 'DNS tunneling: A deep learning based lexicographical detection approach', *arXiv:2006.06122*, v2. Available at: https://10.48550/arXiv.2006.06122 (Accessed: 5 may 2025).

Palo Alto Networks. (no date) *What is DNS Tunnelling?* Available at: https://www.paloaltonetworks.com/cyberpedia/what-is-dns-tunneling (Accessed: 02 May 2025).

Wang, Y., Zhou, A., Liao, S., Zheng, R., Hu, R. and Zhang, L. (2021) 'A comprehensive survey on DNS tunnel detection', *Computer Networks*, 197(C), pp. 19. doi: 10.1016/j.comnet.2021.108322.