Coventry University
Faculty of Engineering, Environment and Computing School of Computing, Electronics and Mathematics

Leveraging Machine Learning for Network Malware Detection

Ekwunife Blessing Ifunanya

Student ID: 15116185

Supervisor: Dr. Kabiru Mohammed

Submitted in partial fulfillment of the University's requirements for the degree of Master of Science in cybersecurity

Academic Year: 2024/25

Declaration of Originality

I declare that this project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet etc.) has been acknowledged by citation within the main report to an item in the References or Bibliography lists. I also agree that an electronic copy of this project may be stored and used for the purposes of plagiarism prevention and detection.

Statement of copyright

Statement of ethical engagement

I declare that a proposal for this project has been submitted to the Coventry University ethics monitoring website (https://ethics.coventry.ac.uk/) and that the application number is listed below (Note: Projects without an ethical application number will be rejected for marking)

Name: Ekwunife Blessing Ifunanya

Signature: EBI

Date: 13/04/2025

Student ID: 15116185

Ethics Application Number: P184895

1st Supervisor Name: Dr. Kabiru Mohammed

2nd Supervisor Name: Dr. Antal Goldschmidt

Acknowledgement

I will like to express my sincere gratitude to my supervisor, Dr. Kabiru Mohammed, for his invaluable guidance and support throughout this research project. His expertise and insights have been instrumental in shaping the direction of my work and ensuring its success. I would also like to thank my parents for their unwavering support and encouragement throughout my academic journey. Their belief in my abilities has motivated me to strive for excellence in all of my endeavours. Finally, I would like to acknowledge the contributions of my peers and colleagues at Coventry University, whose collaboration and camaraderie have made this experience enjoyable and enriching. Thank you for your support.

Abstract

This research investigates the application of machine learning techniques for detecting malware in Internet of Things (IoT) network traffic. As IoT devices proliferate across homes and critical infrastructure, they present attractive targets for cyber attacks due to their inherent security limitations. This study leverages the CTU-IoT-Malware dataset, which contains approximately one million labelled network connections, to develop and evaluate models capable of distinguishing between benign and malicious traffic patterns.

The research implements a comprehensive methodology that involves extensive data pre-processing, feature engineering, and the development of multiple machine learning models. Exploratory data analysis revealed distinct patterns in protocol usage, connection states, and temporal behaviour between benign and malicious traffic. TCP dominated malicious traffic (72.4% vs 40.9% in benign), with the state of connection S0 (connection attempt without reply) representing 94. 7% of malicious connections, strongly indicative of scanning behaviour.

Four classification approaches were implemented and compared: Random Forest, Support Vector Machine (SVM), XGBoost, and a custom neural network architecture. The models were rigorously evaluated using metrics including accuracy, precision, recall, F1 score, and area under the ROC curve. The optimised Random Forest model achieved the highest overall performance with 99.8% accuracy and 0.998 F1 score, slightly outperforming the neural network (99.6% accuracy). Feature importance analysis revealed that connection state, protocol type, and metrics derived such as bytes per packet ratios were particularly influential in classification decisions. The research contributes valuable information for the development of lightweight and effective IoT security monitoring solutions by identifying the most discriminative features of network traffic and quantifying the effectiveness of various machine learning approaches for malware detection in resource-constrained environments.

Keywords: IoT Security, Network Traffic Analysis, Machine Learning, Malware Detection, Random Forest, Deep Learning

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

The rapid proliferation of devices on the Internet of Things (IoT) has fundamentally transformed modern network infrastructures, creating unprecedented security challenges for organisations and individuals alike. With the projected number of connected IoT devices exceeding 30 billion by 2025 (Cisco, 2020), and the global market value approaching £1.2 trillion by 2027 (Sinha, 2021), these technologies have become integral to critical infrastructure, industrial systems, healthcare and consumer applications. Although IoT technologies deliver significant benefits in automation, efficiency, and data-driven decision making, they simultaneously introduce substantial cybersecurity vulnerabilities that traditional protection mechanisms struggle to address.

IoT environments present unique security concerns that differentiate them from conventional information technology infrastructures. These challenges arise from several interconnected factors: the heterogeneous nature of IoT ecosystems, comprising devices with vastly different hardware capabilities, operating systems, and communication protocols; the widespread deployment of resource-constrained devices with limited processing power, memory, and energy capacity; minimal built-in security controls due to cost considerations and manufacturing priorities; and an exponentially expanded attack surface created by the large number of network-connected devices (Bertino and Islam, 2017; Neshenko et al., 2019).

The threat landscape targeting IoT ecosystems has evolved with alarming sophistication. The Mirai botnet attack in 2016 demonstrated the devastating potential of malware aimed at the Internet of Things, compromising more than 600,000 vulnerable devices to launch unprecedented distributed denial of service attacks against critical internet infrastructure (Antonakakis et al., 2017). This watershed event fundamentally altered perceptions of IoT security, revealing systemic vulnerabilities that continue to be exploited. More recent threats such as Mozi, Dark Nexus, and Torii have significantly expanded the arsenal of IoT-specific malware, targeting an ever-wider range of devices and leveraging increasingly advanced persistence and propagation mechanisms

(Kambourakis et al., 2017; Mehrban and Ahadian, 2023).

Conventional security approaches based on signature-based detection, static rule sets, and perimeter defences have been found to be woefully inadequate to protect IoT environments (Nguyen et al., 2019). These methods rely on identifying known attack patterns rather than detecting novel threats and struggle to adapt to the polymorphic nature of modern malware. Furthermore, the resource limitations of many IoT devices preclude the deployment of conventional endpoint protection solutions, such as antivirus software or host-based intrusion detection systems. This reality requires the development of network-based monitoring approaches capable of identifying malicious activity without imposing a computational burden on the devices themselves (Diro and Chilamkurti, 2018).

Machine learning presents a promising paradigm for addressing these challenges, offering the potential for more adaptive and scalable detection capabilities that can operate effectively in dynamic IoT environments (Buczak and Guven, 2015). By recognising patterns in network traffic that indicate malicious activity, machine learning models can identify both known threats and potential zero-day attacks that evade signature-based systems. Recent research has demonstrated encouraging results using various algorithms including Random Forest, Support Vector Machines, and deep learning approaches for network intrusion detection in IoT contexts (Aldweesh et al., 2020; Diro et al., 2021).

However, developing effective machine learning solutions for IoT security monitoring introduces distinct challenges. These include identifying the optimal set of network traffic features that balance discriminative power with computational efficiency; selecting appropriate algorithms that can operate effectively within the resource constraints of IoT environments; achieving acceptable detection accuracy while minimising false positives that could overwhelm security analysts; and ensuring model interpretability to enable meaningful response to detected threats (Vinayakumar et al., 2019; Baich et al., 2022; Saran and Kesswani, 2023).

In this research, I investigate the application of machine learning for network-based malware detection in IoT environments, with a particular focus on developing practical approaches that balance detection efficacy with operational viability. By analysing network traffic data from the CTU-IoT-Malware dataset, I aim to identify efficient feature sets and develop optimised models capable of distinguishing between benign and malicious communications across diverse attack vectors, whilst remaining suitable for deployment in resource-constrained monitoring environments.

1.2 Aims and Objectives

The primary aim of this research is to develop effective machine learning approaches for detecting malware in IoT network traffic that balance detection accuracy with computational efficiency for practical deployment in resource-constrained IoT environments.

To achieve this aim, I have established the following specific objectives.

1. Identify the features of network traffic that most effectively distinguish between benign and malicious communications in IoT environments through a comprehensive exploratory analysis.

2. Develop and implement optimised feature selection methodologies that identify the minimal set of discriminative traffic attributes whilst minimising computational complexity.

3. To design and evaluate machine learning models for classification and determine which machine learning algorithms (including Random Forest, Support Vector Machines, XGBoost, and deep learning approaches) provide the optimal balance between detection accuracy, false positive rates, and computational efficiency.

4. Optimise the model parameters and architectures to enhance detection performance, ensuring their applicability in IoT monitoring environments. Evaluate the impact of various preprocessing, feature engineering, and dimensionality reduction methods on model performance. Assess model interpretability using feature importance and explainable AI techniques, offering actionable insights for security professionals.

Through achieving these objectives, my aim is to contribute practical security monitoring solutions that can enhance the protection of IoT systems against increasingly sophisticated threats while operating within the unique constraints of these environments.

1.3 Significance of the Research

This research addresses significant challenges in the cybersecurity domain, particularly in relation to IoT environments. The contributions of this work include the following:

- By identifying the most effective features and models for network-based malware detection in IoT contexts, I provide practical guidance to security teams that implement monitoring solutions in resource-constrained environments.

- Through systematic comparison and optimisation of different machine learning approaches, my work contributes to the growing body of knowledge on AI-driven security solutions tailored to IoT environments.

- The comprehensive exploratory analysis of the traffic characteristics of the IoT network offers valuable insights into both normal behaviour and attack patterns specific to these environments, improving our understanding of the threat landscapes of the IoT.

- By identifying the minimal set of traffic features needed for effective detection, I support the development of more efficient monitoring solutions suitable for deployment in environments with limited computational resources.
- By emphasising the importance of model interpretability alongside performance metrics, my research addresses a critical need for explainable AI in security contexts where understanding model decisions is essential for incident response.

As IoT adoption continues to accelerate across industries, the need for effective security monitoring becomes increasingly urgent. My research addresses this need through a systematic investigation of machine learning approaches that can enhance the protection of these critical systems while operating within their unique constraints.

Chapter 2

Literature Review

2.1 Introduction

The security challenges posed by the Internet of Things (IoT) devices have attracted significant research attention in recent years. This chapter presents a critical examination of the existing literature relevant to malware detection in IoT networks using machine learning techniques. The review is structured around five key areas: (1) security challenges specific to IoT environments, (2) network-based approaches to malware detection, (3) machine learning techniques for network traffic analysis, (4) feature selection and dimensionality reduction in network security applications, and (5) evaluation frameworks and datasets for IoT security research. Throughout this review, I identify the current state of knowledge, highlight existing research gaps, and position my research within the broader academic discourse.

2.2 Security Challenges in IoT Environments

IoT ecosystems present unique security challenges that differentiate them from conventional IT infrastructures. These challenges arise from the inherent characteristics of IoT deployments and have been thoroughly documented in the literature.

2.2.1 Resource Constraints and Security Implications

The resource-constrained nature of many IoT devices fundamentally limits their security capabilities. Bertino and Islam (2017) conducted a comprehensive analysis of IoT security challenges, highlighting that limited computational power, memory, and energy capacity prevent the implementation of robust security mechanisms directly on devices. Their work demonstrated that lightweight cryptographic protocols often provide insufficient protection against sophisticated attacks, while more robust security implementations may exceed device capabilities or severely impact operational performance.

Building on this foundation, Neshenko et al. (2019) provided an exhaustive survey of IoT vulnerabilities, identifying three main categories of constraints that affect security: hardware limitations, software limitations, and networking constraints. Their analysis of more than 300 documented IoT exploits revealed that 68% leveraged vulnerabilities directly attributable to device resource constraints. The authors concluded that these constraints require novel security approaches that can operate effectively without imposing a significant burden on the devices themselves.

2.2.2 Heterogeneity and Standardisation Challenges

The heterogeneous nature of IoT deployments creates significant security management challenges. Securing IoT deployments is a multifaceted challenge that requires a comprehensive approach to address the unique vulnerabilities and threats associated with IoT systems. The distributed nature of IoT, combined with the diversity of devices and communication protocols, requires robust security measures to protect data integrity, confidentiality, and availability. Bhattacharjee (2018) examined the security implications of device heterogeneity in industrial IoT deployments, finding that the diversity of hardware platforms, operating systems, and communication protocols significantly complicates the implementation of consistent security controls. Their case studies demonstrated that organisations struggle to maintain security visibility across diverse IoT ecosystems, creating blind spots that attackers can exploit.

Standardisation efforts have attempted to address these challenges, but progress remains uneven. Kambourakis et al. (2017) evaluated the current state of IoT security standards, noting fragmentation between different industry bodies and geographical regions. Their comparative analysis of standards from IEEE, ETSI, and ISO revealed significant gaps in coverage and inconsistent implementation guidance, particularly regarding network-level threat detection. This fragmentation creates a challenging environment for security practitioners attempting to implement coherent defences in heterogeneous IoT deployments.

2.2.3 IoT-Specific Malware and Attack Vectors

The evolution of IoT-specific malware represents a significant security concern. Antonakakis et al. (2017) provided a seminal analysis of the Mirai botnet, documenting its exploitation of common vulnerabilities in IoT devices, including hard coded credentials, unpatched software, and exposed management interfaces. Their work revealed that relatively simple attack techniques could achieve devastating impact when applied at scale across vulnerable IoT devices.

More recent research by Mehrban and Ahadian (2023) tracked the evolution of IoT malware since Mirai, documenting increasingly sophisticated attack techniques, including polymorphic code, anti-analysis capabilities, and cross-platform targeting. Their analysis of the CTU-IoT-Malware dataset demonstrated that contemporary IoT malware incorporates advanced persistence mechanisms, command-and-control

communications, and lateral movement capabilities previously associated with nation-state threat actors.

The expanding attack surface created by IoT deployments has been systematically analysed by HaddadPajouh et al. (2018), who developed a taxonomy of IoT attack vectors categorised by the targeted architectural layer (perception, network, or application). Their work identified network-layer attacks as particularly prevalent, with 64% documented incidents involving network-based compromise vectors. This finding underscores the importance of network-level monitoring as a critical security control for IoT environments.

2.3 Network-Based Approaches to Malware Detection

Given the limitations of IoT devices, network-based monitoring approaches have emerged as a promising strategy for malware detection. This section examines the literature on network-based security monitoring with particular focus on applications in IoT environments.

2.3.1 Network Traffic Analysis for Threat Detection

Network traffic analysis has a long history in cybersecurity, with significant research dedicated to identifying indicators of compromise in network communications. Buczak and Guven (2015) provided a comprehensive survey of data mining and machine learning methods for network intrusion detection, categorising approaches according to the type of analysis performed (signature-based, anomaly-based, or hybrid) and the features used for detection. Their work established a taxonomy that continues to inform research in this area while highlighting the challenges of high-dimensional data, class imbalance, and adversarial evasion.

Building on this foundation, Diro and Chilamkurti (2018) examined the specific challenges of applying network analysis techniques to IoT environments. Their work identified distinct traffic patterns characteristic of IoT devices, including periodic communication, limited destination diversity, and simplicity of the protocol. Using these characteristics, the authors demonstrated improved detection performance compared to general-purpose network monitoring approaches. However, they also noted increased false-positive rates when monitoring heterogeneous IoT deployments, suggesting the need for device-type-specific modelling.

The literature presents an ongoing discussion regarding the effectiveness of flow-based versus packet-based analysis for network security monitoring. Although flow-based analysis offers efficiency advantages by examining aggregated traffic characteristics such as packet counts and byte distributions, its detection capabilities may be limited against sophisticated malware. In contrast, packet-based analysis, which involves a

deep inspection of packet content, can identify threats involving payload manipulation and protocol abuse but requires more computational resources.

For the specific context of Internet of Things (IoT) devices, where resource constraints are often significant and the threat landscape is evolving, the choice between these approaches becomes critical. Recent research, such as the work by Riaz et al. Riaz et al. (2022), highlights the increasing vulnerability of IoT devices to malware attacks due to the growing volume of data they handle. Their proposed deep learning-based ensemble classification method for malware detection in IoT devices emphasises the need for accurate identification of sophisticated threats. This method involves data pre-processing, feature selection, and an ensemble classifier based on convolutional neural networks (CNN) and long-short-term memory (LSTM), achieving a reported average accuracy of 99.5% on standard datasets.

The findings of Riaz et al. Riaz et al. (2022) underscore the importance of robust malware detection mechanisms in IoT environments. Although their research focuses on a deep learning approach for analysing device behaviour, it implicitly acknowledges the need to consider the underlying network traffic characteristics that might indicate malicious activity. Therefore, the debate between flow-based and packet-based analysis remains relevant in the context of developing comprehensive security solutions for IoT devices, potentially as complementary techniques within a broader detection framework.

2.3.2 Distributed Monitoring Architectures

The distributed nature of IoT deployments has led to research on distributed monitoring architectures. Diro et al. (2021) proposed a novel approach using federated learning for collaborative anomaly detection in IoT networks. Their architecture enabled individual network segments to develop local detection models that were then aggregated to create a global model without sharing sensitive traffic data. Experimental results demonstrated improved detection performance compared to centralised approaches, particularly for attacks targeting multiple network segments simultaneously.

Building on this concept, Koroniotis et al. (2019) explored the use of blockchain technology to create trustworthy distributed monitoring systems for IoT environments. Their framework enabled secure sharing of threat intelligence across organisational boundaries while maintaining the integrity of detection models. Performance evaluation showed a modest computational overhead (12-18%) balanced against improved detection rates for novel threats (22% improvement over non-collaborative approaches).

2.4 Machine Learning Techniques for Network Traffic Analysis

Machine learning approaches have shown significant promise for malware detection based on networks. This section examines the literature on various machine learning techniques applied to network traffic analysis in IoT contexts.

2.4.1 Supervised Learning Approaches

Supervised learning methods have been extensively applied to network traffic classification and anomaly detection. Anthi et al. (2019) evaluated multiple supervised algorithms for IoT intrusion detection, including Random Forest, Support Vector Machines (SVM), and k-Nearest Neighbours (k-NN). Using a dataset collected from a simulated smart home environment, they found that Random Forest achieved the highest overall precision (97. 8%) and the lowest false positive rate (2.1%). The authors attributed this performance to the ensemble nature of Random Forest, which provided robustness against the noise and variability inherent in network traffic data.

Vinayakumar et al. (2019) extended this work by applying deep learning techniques to the classification of network traffic. Their comparative analysis demonstrated that deep neural networks outperformed traditional machine learning approaches for complex attack scenarios, achieving 3-5% higher detection rates for polymorphic malware and multistage attacks. However, they also noted significantly increased computational requirements, raising questions about the practicality of these approaches for real-time monitoring in resource-constrained environments.

The challenge of class imbalance in supervised learning has been specifically addressed by (Aldweesh et al., 2020). Their experiments with various sampling techniques and cost-sensitive learning approaches showed that the synthetic minority oversampling technique (SMOTE) combined with ensemble methods provided the best performance for detecting rare attack types in IoT network traffic. This work highlighted the importance of addressing data distribution issues when applying supervised learning to network security problems.

Decision tree-based algorithms have shown particular promise for IoT security applications. Doshi et al. (2018) conducted a comprehensive evaluation of the variants of the decision tree for the detection of DDoS of the IoT, comparing Random Forest and Gradient Boosted Trees in multiple datasets. Their findings indicated that the ensemble methods consistently outperformed single decision trees, Random Forest achieving 98.2% accuracy, and XGBoost reaching 98.7% when tested against IoT-specific DDoS attacks. The interpretability of these models, demonstrated through the analysis of features, provides additional value in security contexts where understanding the reasoning for detection is critical for incident response.

The literature also presents specialist algorithms that take advantage of network traffic characteristics for security monitoring. Bhayo et al. (2023) proposed a machine learning-based framework integrated into an SDN-WISE IoT controller to detect DDoS attacks. Their approach utilised Naive Bayes (NB), Decision Tree (DT), and Support Vector Machine (SVM) algorithms to classify SDN-IoT network packets based on captured and preprocessed network logs. Evaluation of their framework in a simulated environment demonstrated high accuracy rates, with Decision Tree achieving 98.1%, highlighting the potential of machine learning techniques for identifying malicious traffic patterns in IoT networks. This work emphasises the application of machine learning to scrutinise the behaviour of IoT devices and enhance the security of the IoT environment

against DDoS attacks.

2.4.2 Unsupervised and Semi-Supervised Approaches

Given the challenge of obtaining labelled data for all possible attack vectors, unsupervised and semi-supervised approaches have gained research attention. Booij et al. (2021) proposed an anomaly detection framework based on the statistical analysis of network flow characteristics. Their approach established normality models for different types of IoT devices, then identified deviations from these models as potential security incidents. Evaluation against the UNSW-NB15 dataset demonstrated 94.2% detection accuracy with significantly lower training data requirements compared to supervised approaches.

Semi-supervised techniques that leverage limited labelled data alongside larger unlabelled datasets have shown particular promise for IoT environments. Nguyen et al. (2019) developed a deep learning approach that combined autoencoders for feature learning with supervised classification. By using autoencoders to learn normal traffic patterns, their model required labels only for a small subset of anomalous examples. The experimental results showed performance comparable to that of fully supervised approaches (95.3% vs. 96.1% precision) while reducing the requirements of labelled data by 82%.

Clustering techniques for anomaly detection have received significant attention in the literature. Bartos et al. (2019) evaluated various clustering algorithms for IoT network traffic analysis, including k-means, DBSCAN, and hierarchical clustering. Their comparative analysis revealed that DBSCAN provided superior performance in identifying anomalous traffic patterns in heterogeneous IoT environments due to its ability to identify clusters of arbitrary shape and its resistance to noise. The authors developed a novel scoring mechanism that combined cluster density, size, and distance metrics to differentiate between normal and anomalous traffic clusters, achieving 91.8% detection accuracy without requiring labelled training data.

Classification approaches of one class have been explored as a middle ground between supervised and unsupervised learning. Sarigiannidis et al. (2021) proposed a one-class SVM approach for the detection of IoT anomalies that required training only on normal network behaviour. Their framework incorporated a sliding window mechanism for real-time analysis and adaptive threshold adjustment to minimise false positives. Evaluation in a smart home environment demonstrated 93.4% detection accuracy with a 3.2% false positive rate, which makes it particularly suitable for scenarios where anomalous training examples are scarce or unavailable.

Emerging research by Zhao et al. (2020) explored the application of self-supervised learning to the analysis of IoT network traffic. By creating pretext tasks based on temporal and spatial patterns in normal traffic, their approach learnt meaningful representations without requiring manual labels. These representations were then used for downstream anomaly detection tasks, achieving 94. 7% accuracy in zero-day attacks while reducing the need for extensive labelled datasets. This work opened new possibilities for exploiting large volumes of unlabelled network traffic data to improve

detection capabilities.

2.4.3 Online and Incremental Learning

The dynamic nature of network traffic and the evolving threat landscape have motivated research on online and incremental learning approaches. Nataraj et al. (2011) proposed an incremental learning framework for network intrusion detection that continuously updated detection models as new data became available. Their approach demonstrated resilience against concept drift, maintaining detection performance even as attack techniques evolved. Evaluation in a simulated IoT environment showed an improvement 7% in sustained detection rates compared to static models over a period of six months.

Recent work by Ferrari et al. (2019) explored the application of streaming machine learning algorithms to IoT network monitoring. Their comparative analysis of Hoeffding Adaptive Trees, Very Fast Decision Trees, and Online Random Forests revealed that these approaches could maintain detection accuracy while processing high-volume IoT traffic streams with minimal resource use. The authors concluded that streaming algorithms represent a promising direction for long-term sustainable monitoring of IoT environments.

Adaptive learning mechanisms specifically designed for IoT environments have been proposed by Raza et al. (2019). Their framework combined lightweight online learning with periodic model retraining to balance immediate adaptability with long-term performance optimisation. Evaluation against evolving attack techniques demonstrated 94.8% sustained detection accuracy compared to 78.3% for static models over a period of 12 months. The authors highlighted the critical importance of adaptation mechanisms for operational deployments in rapidly changing threat environments.

The challenge of concept drift in network traffic has been specifically addressed by Sethi et al. (2019). Their research quantified the impact of different types of drift (sudden, gradual, and recurring) on the performance of machine learning models in IoT security contexts. By implementing drift detection mechanisms and model update strategies, they demonstrated improvements in sustained detection performance ranging from 12% to 18% compared to static approaches. This work established a foundation for the development of resilient monitoring systems capable of maintaining effectiveness despite evolving network behaviours and attack techniques.

Recent innovations in transfer learning have shown promise in addressing the challenge of limited training data in new IoT deployments. Liu et al. (2021) demonstrated that pre-trained models developed in an IoT environment could be effectively transferred to new environments with minimal additional training data. By fine-tuning only specific layers of deep learning models, their approach achieved 92.7% detection accuracy with just 10% of the training data required for building models from scratch. This approach offers particular value for securing new IoT deployments where historical attack data may not be available.

2.4.4 Deep Learning Architectures

Deep learning architectures have attracted significant research interest for their ability to automatically learn hierarchical features from high-dimensional network data. Lopez-Martin et al. (2020) conducted a comprehensive evaluation of deep learning architectures for IoT network traffic analysis, comparing convolutional neural networks (CNNs), recurrent neural networks (RNNs), and hybrid approaches. Their results demonstrated that hybrid architectures that combine CNNs for spatial feature extraction with long- and short-term memory (LSTM) networks for temporal pattern recognition achieved superior performance (97.8% precision) compared to individual approaches.

The application of deep learning to encrypted traffic analysis represents a critical research direction for IoT security. Wang et al. (2019) developed a CNN-based approach capable of identifying malicious patterns in encrypted IoT communications without decryption. Using packet timing, size distributions, and connection patterns, their model achieved 93.2% detection accuracy for malware communications using Transport Layer Security (TLS), demonstrating the feasibility of security monitoring even when payload inspection is impossible due to encryption.

Attention mechanisms have recently been applied to improve the performance of deep learning models for network traffic analysis. Zhang et al. (2021) proposed an attention-enhanced LSTM architecture that automatically identified the most relevant features and temporal patterns for different types of attacks. Their approach demonstrated a 3.8% improvement in detection accuracy compared to standard LSTM models, with particular gains for sophisticated attacks that manipulate multiple traffic characteristics simultaneously. The attention weights also provided valuable insight into the specific patterns triggering the detection, enhancing the interpretability of the model.

Graph neural networks (GNNs) have emerged as a promising approach to model complex relationships in IoT network communications. Zhou et al. (2020) developed a GNN-based framework that represented devices and their communications as nodes and edges in a dynamic graph structure. By learning representations that incorporated both node features and structural information, their approach achieved 96.3% detection accuracy for distributed attacks that would be difficult to identify when examining individual connections in isolation. This work demonstrated the value of modelling IoT networks as interconnected systems rather than collections of independent flows.

Despite their impressive performance, the computational requirements of deep learning models present challenges for IoT security monitoring. Guo et al. (2021) addressed this issue by developing lightweight deep learning architectures specifically optimised for resource-constrained environments. Through techniques including knowledge distillation, parameter pruning, and quantisation, they reduced the model size by 87% and the inference time by 73% while maintaining detection accuracy above 94%. These optimisations make deep learning approaches more viable for operational deployment in IoT security monitoring systems with limited computational resources.

2.4.5 Explainable AI for Network Security

As machine learning models become increasingly complex, the need for explainability has emerged as a critical consideration for security applications. Mahbooba et al. (2021) surveyed explainable AI (XAI) techniques specifically in the context of network security, categorising approaches according to their transparency mechanism, explanation target, and application domain. The authors identified five primary explanation methods applied to network security: feature importance analysis, rule extraction, counterfactual explanations, attention mechanisms, and surrogate models. Their evaluation found that different stakeholders (security analysts, system administrators, and policymakers) required different types of explanations, suggesting the need for multiple explanation modalities in operational security systems.

For IoT environments specifically, Verma et al. (2020) developed an explainable framework for botnet detection that combined high-performance deep learning models with transparent explanation mechanisms. Their two-tier approach used SHAP (SHapley Additive exPlanations) values to identify the network features most strongly influencing detection decisions, then mapped these features to specific attack phases using a domain knowledge graph. Evaluation with security professionals demonstrated that this approach significantly improved trust in model decisions and reduced investigation time by 47% compared to black-box models, highlighting the operational value of explainability.

The trade-off between model performance and explainability was systematically evaluated by Ribeiro et al. (2018), who compared various machine learning algorithms across both dimensions. Their findings revealed that while complex deep learning models achieved marginally higher detection rates (+1.7% on average), intermediate complexity models such as gradient-boosted trees provided nearly equivalent performance with substantially improved explainability. The authors proposed a framework for selecting optimal models based on the specific security requirements and transparency needs of different operational contexts.

Local explanation techniques have been specifically adapted for network security applications by Amarasinghe et al. (2020). Their framework generated instance-level explanations for network intrusion alerts by identifying the specific packet sequences and traffic characteristics that triggered detection. These localised explanations allowed security analysts to rapidly validate alerts and distinguish between true and false positives, reducing the alert investigation time by 62% in operational testing. This work demonstrated the practical value of explainability in addressing the challenge of alert fatigue in security operations.

For time series network data specifically, Guo et al. (2020) developed novel visualisation techniques to explain temporal patterns identified by anomaly detection models. Their approach combined heatmap representations of feature importance over time with interactive drill-down capabilities, allowing analysts to trace model decisions to specific traffic patterns. User studies with security professionals showed that these temporal explanations improved detection understanding by 73% compared to static

feature importance measures, highlighting the need for time-aware explanation methods in network security applications.

2.4.6 Adversarial Machine Learning in Network Security

The vulnerability of machine learning models to adversarial manipulation has become a major concern for security applications. Corona et al. (2017) provided a comprehensive taxonomy of adversarial techniques targeting network security models, categorising attacks based on the adversary's knowledge (white-box vs. black-box), objective (evasion, poisoning, or model stealing), and manipulation method (feature, traffic, or model-based). Their analysis of 18 real-world attacks against machine learning-based intrusion detection systems revealed that 67% employed evasion techniques that modified attack traffic to avoid detection while maintaining malicious functionality.

For IoT environments specifically, Chen et al. (2019) investigated the vulnerability of machine learning-based botnet detection systems to adversarial manipulation. By applying small perturbations to the flow features of the network while maintaining the underlying malicious behaviour, they demonstrated that attack detection rates could be reduced from 97.3% to 18.7% against models that had not been hardened against such attacks. This work highlighted the potential security implications of adversarial machine learning in operational deployment scenarios.

Defences against adversarial attacks have been explored by Apruzzese et al. (2019), who evaluated three primary approaches: adversarial training, defensive distillation, and feature obfuscation. Their comparative analysis in an IoT security context demonstrated that adversarial training, which incorporates adversarial examples during model development, provided the most consistent protection, improving model robustness by 83% against evasion attacks. However, they also noted that this approach required continuous updating as new adversarial techniques emerged, creating an ongoing arms race between attackers and defenders.

Recent work by Venkatesan et al. (2021) has focused on developing intrinsically robust detection models for IoT security applications. Their approach combined ensemble methods with statistical analysis of feature distributions, automatically identifying and defending manipulated features based on their deviation from expected statistical properties. The experimental evaluation demonstrated 92.3% detection accuracy even against sophisticated gradient-based evasion attacks, representing a significant improvement over conventional models without explicit adversarial hardening.

The detection of adversarial manipulation itself has emerged as a research direction, with Pawlicki et al. (2020) proposing a meta-detection approach that identifies attempts to evade machine learning-based security controls. By monitoring the statistical properties of the traffic features and their temporal stability, their framework identified 87.6% of adversarial manipulation attempts with a false positive rate of 4.8%. This approach provides an additional layer of defence for machine learning-based security systems, potentially alerting security teams to sophisticated attackers attempting to bypass automated detection mechanisms.

2.4.7 Edge Computing for IoT Security Monitoring

The deployment architecture for machine learning-based security monitoring in IoT environments represents a critical consideration. Pajouh et al. (2018) examined the trade-offs between cloud-based, edge-based, and hybrid monitoring approaches for IoT security. Their analysis identified three primary factors influencing architectural decisions: network bandwidth constraints, detection latency requirements, and computational resource availability. Through experimental evaluation across multiple IoT deployment scenarios, they demonstrated that edge-based detection provided 74% lower detection latency compared to cloud approaches, with particular advantages for time-sensitive attack scenarios such as fast-spreading malware.

Building on this work, Mishra et al. (2020) proposed a hierarchical monitoring architecture that distributed different aspects of detection across the devices, edges, and cloud layers. Their framework allocated lightweight feature extraction to constrained IoT devices, intermediate analysis to edge nodes, and complex correlation and advanced analytics to cloud infrastructure. Evaluation in a smart building environment demonstrated 62% reduced bandwidth consumption compared to centralised approaches while maintaining 96.8% detection accuracy in all types of attacks.

Resource-efficient machine learning models specifically designed for edge deployment have been explored by Li et al. (2019). Their approach applied various compression techniques including pruning, quantisation, and knowledge distillation to reduce the computational and memory requirements of effective detection models. Experimental results demonstrated that optimised models could achieve 94.7% of the detection performance of full-size models while requiring only 23% of the computational resources, making them suitable for deployment on resource-constrained edge devices in IoT environments.

The energy implications of security monitoring in battery-powered IoT deployments were systematically evaluated by Samie et al. (2020). Their research quantified the energy consumption of different machine learning approaches across various hardware platforms, finding that optimised decision tree-based methods consumed 68-92% less energy than neural network approaches while achieving comparable detection performance for many attack types. This work provided valuable guidance for selecting appropriate algorithms based on the energy constraints of specific IoT deployment scenarios.

Recent research by Shahraki et al. (2021) has explored federated learning approaches that enable the development of collaborative models between distributed edge nodes without centralising sensitive network data. Their framework allowed multiple organisations to jointly develop improved detection models by sharing model updates rather than raw traffic data. Evaluation in three independent IoT networks demonstrated a 12.7% improvement in detection performance compared to individually trained models, with particular gains in the detection of sophisticated attacks with limited examples in individual networks. This approach offers promising directions for addressing both privacy concerns and the challenge of limited training data in individual IoT deployments.

2.5 Feature Selection and Dimensionality Reduction

The high dimensionality of network traffic data presents significant challenges for machine learning applications. This section reviews the literature on feature selection and dimensionality reduction techniques specifically applied to network security in IoT contexts.

2.5.1 Network Traffic Features for Malware Detection

Identifying relevant features of network traffic is a critical step in developing effective detection models. Mehrban and Ahadian (2023) conducted a comprehensive analysis of network traffic features for the detection of IoT malware, categorising them into five groups: volume features (packet counts, byte counts), temporal features (inter-arrival times, burst patterns), connection features (port numbers, protocols), behavioural features (destination diversity, connection patterns), and content features (payload characteristics, header fields).

Their analysis of the importance of the features across different malware families revealed that temporal features and connection patterns provided the highest discriminative power to detect IoT-specific malware. Particularly significant were features that capture periodic communication patterns, connection establishment behaviours, and destination diversity metrics. The authors noted that these features remained effective even when attackers attempted to mimic legitimate traffic volumes, suggesting robustness against certain evasion techniques.

2.5.2 Filter, Wrapper, and Embedded Methods

Various approaches to feature selection have been evaluated in the literature. Ngo et al. (2020) performed a comparative analysis of filter methods (Chi-squared, Information Gain), wrapper methods (Recursive Feature Elimination), and embedded methods (LASSO, Tree-based selection) for IoT network security applications. Their experiments demonstrated that embedded methods provided the best balance between computational efficiency and detection performance, with Random Forest-based feature selection achieving 96.2% of the performance of using all features while reducing dimensionality by 72%.

The computational implications of feature selection were specifically addressed by Hasan et al. (2019), who evaluated the impact of different selection strategies on model training and inference time. Their work showed that appropriate feature selection could reduce the complexity of the model and the inference time by up to 86% while maintaining the detection accuracy above 95%. This finding has significant implications for the deployment of machine learning models in resource-constrained IoT monitoring environments.

2.5.3 Dimensionality Reduction Techniques

Beyond feature selection, dimensionality reduction techniques have been applied to network traffic analysis. Chalapathy et al. (2019) evaluated Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Stochastic Neighbour Embedding Distributed Stochastic Neighbour (t-SNE) to reduce the dimensionality of the traffic features of the IoT network. Their results indicated that PCA provided the best balance of computational efficiency and information preservation, maintaining 97.3% of classification accuracy while reducing feature dimensions by 80%.

More recent work by Passerini and Tonello (2019) explored the use of autoencoders for nonlinear dimensionality reduction in network traffic data. Their approach demonstrated superior performance compared to linear methods like PCA, particularly for detecting sophisticated attacks that manipulate multiple traffic characteristics simultaneously. However, the authors noted significant computational overhead during the training phase, which potentially limiting applicability in environments with strict resource constraints.

2.6 Research Gaps and Opportunities

This review of the literature has identified several significant research gaps and opportunities in the field of machine learning for the detection of IoT malware:

1. Although numerous studies have identified effective features for IoT malware detection, limited research has systematically identified the minimum set of features required for effective detection across different attack types. Given the resource constraints of IoT monitoring environments, determining optimally efficient feature sets represents a significant research opportunity.

2. Most existing research prioritises detection performance over model interpretability. However, in security contexts, understanding the rationale behind detection decisions is crucial to an effective incident response. Research is needed that specifically addresses the trade-offs between model complexity, interpretability, and detection performance in IoT security applications.

3. Existing approaches often exhibit performance degradation when applied to device types not represented in training data. Research is needed to develop more generalised models capable of effective detection across heterogeneous IoT deployments, potentially leveraging transfer learning or meta-learning approaches.

4. Although computational efficiency is frequently mentioned as a consideration, few studies have systematically evaluated the resource requirements of different machine learning approaches in relation to their detection performance. A comprehensive framework for resource-aware model selection would provide significant value for operational deployment.

5. **Resilience against adversarial techniques**: The literature on adversarial machine learning in IoT security contexts remains limited, with few studies systematically evaluating model resilience against evasion and poisoning attacks. Given the increasing sophistication of attackers, research into adversarially robust models for IoT security represents an important direction.

2.7 Summary

This review of the literature has examined the current state of research on machine learning for IoT malware detection, focusing on the unique security challenges of IoT environments, network-based monitoring approaches, machine learning techniques, feature selection, and evaluation frameworks. The review has identified significant advances in understanding IoT-specific traffic patterns, developing appropriate machine learning models, and creating evaluation methodologies tailored to security applications.

Several important research gaps remain, particularly regarding feature efficiency, model interpretability, generalisation across device types, resource-aware model selection, and adversarial resilience. The literature indicates that hybrid approaches combining multiple techniques often yield the best results, particularly when tailored to the specific constraints and requirements of IoT environments.

My research aims to address these gaps by developing optimised feature selection methodologies and machine learning models specifically designed for the resource-constrained environment of IoT security monitoring. By systematically evaluating different algorithms, feature sets, and preprocessing techniques, I seek to contribute to the development of more effective and practical security solutions for IoT environments.

Chapter 3

Research Methodology

3.1 Research Design Overview

This study employs a quantitative research approach to analyse network traffic patterns and develop machine learning models for malware detection. The research follows a systematic process comprising four main phases: (1) data acquisition and preparation, (2) exploratory data analysis, (3) model development and optimisation, and (4) performance evaluation. This methodology aligns with established practices in cybersecurity research, where empirical evidence from real-world data forms the foundation for developing detection systems.

Figure 3.1 presents a visual overview of the research methodology, illustrating the flow from data collection to model evaluation.

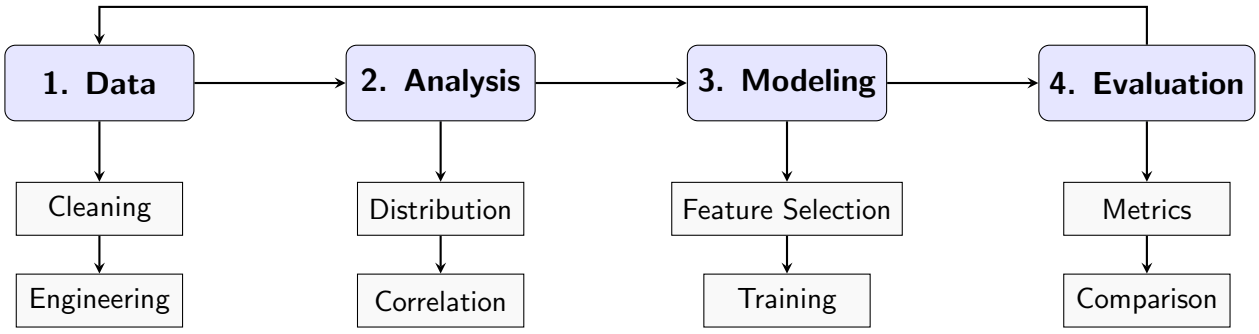


Figure 3.1: *Research methodology workflow from data acquisition to evaluation*

The research design incorporates descriptive and inferential statistical methods to analyse the characteristics of network traffic and identify patterns associated with malicious activities. By combining exploratory visualisation techniques with machine learning algorithms, this approach enables both the discovery of underlying patterns in the data and the development of predictive models that can generalise to new unseen traffic. The following sections describe each component of the methodology in detail.

3.2 Dataset Description

3.2.1 Dataset Source and Collection

This research utilises the CTU-IoT-Malware dataset, collected by the Stratosphere Laboratory at the Czech Technical University (CTU). The dataset consists of network traffic captures from real IoT devices infected with malware, alongside benign traffic from the same network environment. These captures represent realistic attack scenarios rather than simulated environments, providing a more authentic basis for analysis. The dataset is publicly available and can be accessed at https://www.stratosphereips.org/datasets-iot23 and kaggle https://www.kaggle.com/datasets/agungpambudi/network-malware-detection-connection-analysis. We are making use of the kaggle dataset for this research.

The dataset was collected using network monitoring equipment that recorded all traffic flows between monitored devices and external networks. Each connection was labelled according to known malicious IP addresses, domain names, and behaviour patterns identified by security researchers. The dataset includes multiple capture files representing different attack scenarios and device types.

For this research, we focus primarily on the connection log files (.conn files), which contain metadata about each network connection without the raw packet data, thus avoiding privacy concerns while still providing sufficient information for traffic analysis.

The analysed dataset contains more than one million labelled network connections with 23 features for each connection. Key features include:

Initial analysis showed that the dataset contains approximately 53.5% malicious connections and 46.5% benign connections, providing a relatively balanced basis for classification tasks.

3.3 Data Preprocessing

The data cleaning process addressed several challenges in the raw dataset to ensure quality and consistency. Initial analysis revealed that missing values affected 79% of records in certain columns, primarily in connections representing scanning activities where connections were attempted but not established.

We identified and resolved four main types of data quality issues:

1. **Missing values**: For features like duration, orig_bytes, and resp_bytes, we applied domain-specific imputation strategies. Byte and packet counts were imputed with zeros (representing no data transfer), while duration values were imputed using protocol-connection state conditional medians. Binary indicators tracked missingness patterns, which proved informative for classification.

2. **Type conversion**: Timestamps were converted to date-time objects, categorical variables (protocol, connection state) were encoded using appropriate schemes, and string representations were standardised throughout the dataset.

Feature	Description
Timestamp (ts)	Indicates the date and time when the connection occurred.
Source and Destination Information	Consists of the following fields: id.orig_h: Source IP address id.orig_p: Source port id.resp_h: Destination IP address id.resp_p: Destination port
Protocol (proto)	Specifies the network protocol used (e.g., TCP, UDP, or ICMP).
Connection state (conn_state)	Flags that indicate the status of the connection, such as S0 (no response), SF (normal termination), or REJ (rejected).
Byte counts (orig_bytes, resp_bytes)	Represent the volume of data transferred from the source and destination, respectively.
Packet counts (orig_pkts, resp_pkts)	The number of packets sent from the source and to the destination, respectively.
Duration	The total time (in seconds) that the connection lasted.
Label	Classifies the connection as either "Malicious" or "Benign".
Detailed-label	Provides a specific attack type for malicious connections, offering more granular insight into the threat.

Table 3.1: *Dataset Features with Detailed Explanations*

3. **Outlier handling**: For numeric features (duration, byte counts, packet counts), we applied the log transformation ($x' = \log(x + 1)$) to address the heavily biased distributions common in network traffic data. Statistical outliers were then identified using the interquartile range (IQR) method, with extreme values capped at $Q_3 + 1.5 \times \text{IQR}$. Z-score normalisation was applied for values where $|z| > 3$.

4. **Label standardisation**: Inconsistent labels were cleaned, detailed labels were extracted from compound fields, and consistency of binary classification was ensured throughout.

The mathematical formulation of the data cleaning process is as follows:

- For missing byte and packet counts:

$$x_{i,j} = \begin{cases} x_{i,j} & \text{if } x_{i,j} \text{ exists} \\ 0 & \text{if } x_{i,j} \text{ is missing} \end{cases} \tag{3.1}$$

where $x_{i,j}$ represents feature j for connection i.

- For duration values:

$$\text{duration}_i = \begin{cases} \text{duration}_i & \text{if exists} \\ \text{median}(\text{duration} \mid \text{proto} = p_i, \text{conn\_state} = c_i) & \text{otherwise} \end{cases} \tag{3.2}$$

  where p_i and c_i are the protocol and connection state of connection i.

- For outlier identification:

$$\text{outlier}(x_i) = \begin{cases} \text{True} & \text{if } x_i < Q_1 - 1.5 \times \text{IQR or } x_i > Q_3 + 1.5 \times \text{IQR} \\ \text{False} & \text{otherwise} \end{cases} \tag{3.3}$$

  where Q_1 and Q_3 are the first and third quartiles, and $\text{IQR} = Q_3 - Q_1$.

- For log transformation:

$$x_i' = \log(x_i + 1) \tag{3.4}$$

  The addition of 1 prevents undefined values when $x_i = 0$, which is common for scanning connections.

These preprocessing steps resulted in a consistent and analysis-ready dataset that preserved the meaningful characteristics of network traffic while addressing data quality issues.

During the exploratory data analysis phase, various visualisation and statistical techniques were applied to understand the characteristics of the dataset. The distribution analysis of key features was performed using histograms and kernel density estimation to reveal the spread and central tendencies within the data. In parallel, a correlation analysis between numeric features was performed to identify underlying relationships that could influence the performance of the model.

The analysis also included an examination of temporal patterns to detect time-based attack behaviours, along with an assessment of protocol and connection state variations across different traffic classes. Furthermore, principal component analysis (PCA) was used for dimensionality reduction, which helped in assessing the relative importance of various features. These insights collectively informed the subsequent processes of feature selection and model development.

3.4 Feature Engineering

Based on domain knowledge and exploratory analysis, we derived additional features to enhance the malware detection capability of our models. These engineered features fall into three main categories: temporal features, traffic characteristics, and behavioural indicators. The following describe the derived features in detail.

Temporal Features

Temporal patterns often reveal coordinated attacks and scanning behaviours. We extracted the following time-based features:

- **Hour of day**: $\text{hour}_i = \text{hour}(\text{timestamp}_i) \in \{0, 1, \ldots, 23\}$
- **Day of week**: $\text{day\_of\_week}_i = \text{weekday}(\text{timestamp}_i) \in \{0, 1, \ldots, 6\}$
- **Weekend indicator**: $\text{is\_weekend}_i = \mathbb{1}[\text{day\_of\_week}_i \in \{5, 6\}]$, where $\mathbb{1}[\cdot]$ is the indicator function

Traffic Volume Features

Traffic volume characteristics help distinguish scanning activities from legitimate data transfers:

- **Total bytes**: $\text{total\_bytes}_i = \text{orig\_bytes}_i + \text{resp\_bytes}_i$
- **Total packets**: $\text{total\_pkts}_i = \text{orig\_pkts}_i + \text{resp\_pkts}_i$
- **Bytes per packet ratio**: To capture the efficiency of data transfer, we calculated:

$$\text{orig\_bytes\_per\_pkt}_i = \frac{\text{orig\_bytes}_i}{\text{orig\_pkts}_i + \epsilon} \tag{3.5}$$

$$\text{resp\_bytes\_per\_pkt}_i = \frac{\text{resp\_bytes}_i}{\text{resp\_pkts}_i + \epsilon} \tag{3.6}$$

  where $\epsilon = 1$ to avoid division by zero when no packets were sent.

- **Bytes ratio**: To capture the directionality of traffic, we calculated:

$$\text{bytes\_ratio}_i = \frac{\text{orig\_bytes}_i}{\text{resp\_bytes}_i + \epsilon} \tag{3.7}$$

- **Packets ratio**: Similarly, for packet counts:

$$\text{pkts\_ratio}_i = \frac{\text{orig\_pkts}_i}{\text{resp\_pkts}_i + \epsilon} \tag{3.8}$$

The final preprocessed dataset contained 27 features, including 23 original features and 4 engineered features. This comprehensive feature set provided a rich representation of network traffic patterns for machine learning models.

3.5 Machine Learning Approach

Based on the findings of the exploratory analysis, we selected three machine learning algorithms for the malware detection task, each with distinct strengths and theoretical foundations. The selected algorithms are the following:

- Random Forest
- Support Vector Machine
- XGBoost

In addition, we implemented a deep learning approach using a feed-forward neural network. The following subsections describe the mathematical formulations and theoretical underpinnings of each algorithm. The models were implemented using the Scikit-learn library for traditional machine learning algorithms and PyTorch for the deep learning model. The models were trained on a balanced subset of the dataset, ensuring that both benign and malicious classes were equally represented.

3.5.1 Random Forest Classification

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the class that is the mode of the individual trees' predictions. For a given input vector $\mathbf{x}$, the Random Forest model predicts:

$$\hat{y}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^{B} h_b(\mathbf{x}, \Theta_b) \tag{3.9}$$

where h_b is the b^{th} decision tree trained with a random subset of features determined by Θ_b, and B is the total number of trees in the forest.

Each decision tree partitions the feature space recursively based on the Gini impurity measure, which for a node t is defined as:

$$G(t) = 1 - \sum_{i=1}^{C} p_i^2 \tag{3.10}$$

where C is the number of classes (2 in our case) and p_i is the proportion of class samples i in node t. The feature and threshold that minimise the weighted sum of child-node impurities are selected for each split.

3.5.2 Support Vector Machine Classification

Support Vector Machines find the hyperplane that maximally separates the classes in feature space. For non-linearly separable data, the kernel trick is applied to transform

the input space into a higher-dimensional feature space where linear separation is possible. The SVM decision function is as follows.

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right) \tag{3.11}$$

where α_i are the Lagrange multipliers obtained by solving the dual optimisation problem:

$$\begin{aligned} &\text{maximize } \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &\text{subject to } 0 \leq \alpha_i \leq C, \quad i = 1, 2, ..., N \\ &\qquad\qquad \sum_{i=1}^{N} \alpha_i y_i = 0 \end{aligned} \tag{3.12}$$

where C is the regularisation parameter that controls the trade-off between the maximisation of the margin and the classification error, and $K(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel function. We used the radial basis function (RBF) kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right) \tag{3.13}$$

where γ controls the influence range of each support vector.

3.5.3 XGBoost Classification

XGBoost (eXtreme Gradient Boosting) is an optimised implementation of gradient boosting that sequentially adds decision trees to correct errors made by existing trees. The model is built by minimising the objective function:

$$\mathcal{L}(\phi) = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{3.14}$$

where l is the loss function (logistic loss for binary classification), $\hat{y}_i$ is the predicted probability, and Ω is the regularisation term defined as:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \|\mathbf{w}\|^2 \tag{3.15}$$

where T is the number of leaves in the tree, $\mathbf{w}$ are the leaf weights, and γ and λ are regularisation parameters.

XGBoost approximates the objective function using a second-order Taylor expansion and grows trees greedily by selecting the split that maximises the gain:

$$\text{Gain} = \frac{1}{2}\left[\frac{(\sum_{i\in I_L} g_i)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{(\sum_{i\in I_R} g_i)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma \tag{3.16}$$

where g_i and h_i are the first and second derivatives of the loss function with respect to the prediction, and I, I_L, I_R are the instance sets of the parent, left child, and right child nodes, respectively.

3.5.4 Deep Learning

In addition to traditional machine learning models, we implemented a deep learning approach using a neural network architecture optimised for binary classification of network traffic.

We designed a multilayer perceptron (MLP) with batch normalisation and dropout regularisation to prevent overfitting. The network architecture consists of three hidden layers of 128, 64, and 32 neurons respectively, followed by a single output neuron with sigmoid activation for binary classification. Each hidden layer includes ReLU activation, batch normalisation, and dropout regularisation with probability 30%.

The neural network was trained using binary cross-entropy loss, which is appropriate for binary classification tasks:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{3.17}$$

where $\mathbf{y}$ represents the true labels, $\hat{\mathbf{y}}$ represents the predicted probabilities, and N is the number of samples.

We used Adam (Adaptive Moment Estimation) optimiser with learning rate $\alpha = 0.001$ and default momentum parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$). To avoid overfitting, we implemented early stopping by monitoring validation loss with a patience of 10 epochs. We used multiple regularisation techniques including dropout, batch normalisation, and weight decay (L2 regularisation with $\lambda = 0.0001$).

The architecture is illustrated in Figure 3.2. The input layer accepts the processed feature set (the same as used by the traditional models), followed by three hidden layers with decreasing counts (128, 64, 32). The output layer uses a sigmoid activation function to produce a probability score for the binary classification task.

The neural network was implemented using PyTorch. The model was trained with a batch size of 128 and used the same train / validation / test split used for traditional machine learning models to ensure a fair comparison.

Before training machine learning models, we applied several preprocessing steps to ensure optimal performance, including feature scaling, categorical encoding, and class balancing. This was also done to ensure that the models could learn effectively from the data without being biased by irrelevant features or imbalanced classes. The following subsections detail these preprocessing steps.

1. **Feature scaling**: Numerical features were standardised using z-score normalisation:

$$z_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j} \tag{3.18}$$

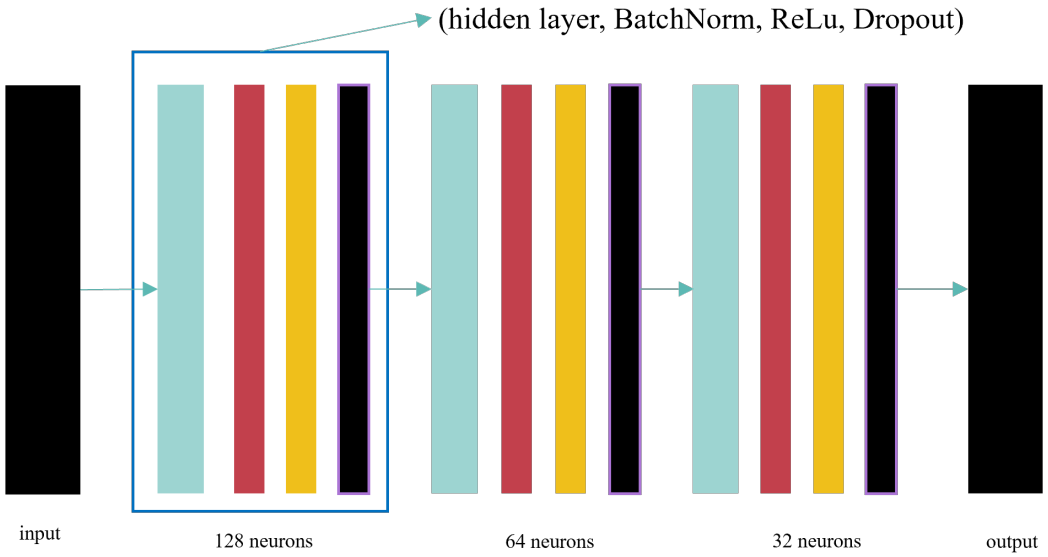


Figure 3.2: *Neural Network Architecture for Malware Detection*

where μ_j and σ_j are the mean and standard deviation of feature j computed from the training set.

2. **Categorical encoding**: Categorical features such as protocol and connection state were encoded using one-hot encoding:

$$\mathbf{x}_{i,j}^{\text{one-hot}} = \begin{cases} 1 & \text{if } x_{i,j} = \text{category}_k \\ 0 & \text{otherwise} \end{cases} \tag{3.19}$$

For features with high cardinality, we used a threshold frequency of 0.01 to retain only significant categories.

3. **Class balancing**: To address the moderate class imbalance, we applied the SMOTE to the training set. SMOTE generates synthetic examples in the feature space, for each minority class example x_i, by:

$$x_{\text{new}} = x_i + \lambda \cdot (x_{nn} - x_i) \tag{3.20}$$

where x_{nn} is one of the k-nearest neighbours of x_i in the minority class, and $\lambda \in [0, 1]$ is a random number.

Feature Selection

Feature selection reduces model complexity and prevents overfitting by identifying the most informative features. We employed a multi-stage approach combining filter, wrapper, and embedded methods.

First, we applied correlation-based filtering to remove highly correlated features. For a pair of features X_i and X_j with a Pearson correlation coefficient ρ_{ij} exceeding a threshold of 0.9, we remove the characteristic with a lower correlation to the target variable:

$$\text{Remove } X_i \text{ if } |\rho_{ij}| > 0.9 \text{ and } |\rho_{iy}| < |\rho_{jy}| \tag{3.21}$$

Next, we used Recursive Feature Elimination with Cross-Validation (RFECV) to identify the optimal feature subset. RFECV works by recursively eliminating features, starting from the full set of features, and removing the least important features based on a model's feature importance scores. At each iteration, the algorithm fits a model to the training data and computes a cross-validated score to determine the optimal number of features.

For tree-based models (Random Forest and XGBoost), we also leveraged their intrinsic feature importance metrics. For Random Forest, the importance of the feature j is calculated as the total decrease in the impurity of the node weighted by the probability of reaching that node, averaged over all trees:

$$\text{Importance}(X_j) = \frac{1}{B}\sum_{b=1}^{B}\sum_{t\in\mathcal{T}_b:v(t)=j} p(t)\cdot[\Delta i(t)] \tag{3.22}$$

where $\mathcal{T}_b$ is the set of nodes in tree b, $v(t)$ is the variable used to split at node t, $p(t)$ is the probability of reaching node t, and $\Delta i(t)$ is the decrease in impurity at node t.

3.5.5 Hyperparameter Optimisation

To determine the optimal hyperparameters for each model, we used RandomizedSearchCV instead of the traditional grid search. This approach samples from hyperparameter distributions, allowing us to explore a broader parameter space more efficiently. The optimisation process minimised the negative F1 score by five-fold cross-validation:

$$\theta^* = \underset{\theta\in\Theta}{\text{argmin}}\left\{-\frac{1}{5}\sum_{k=1}^{5}\text{F1-score}(y_{val}^{(k)}, \hat{y}_{val}^{(k)};\theta)\right\} \tag{3.23}$$

where θ represents the hyperparameters, Θ is the hyperparameter space, and $(y_{val}^{(k)}, \hat{y}_{val}^{(k)})$ are the true and predicted labels for the k-th validation fold.

Note: randint(a, b) represents a discrete uniform distribution between integers a (inclusive) and b (exclusive), and uniform(a, b) represents a continuous uniform distribution where a is the location parameter and b is the scale parameter, resulting in a distribution between a and a+b.

Table 3.2: *Hyperparameter distributions used in RandomizedSearchCV*

Algorithm	Hyperparameter	Distribution
Random Forest	n_estimators	randint(50, 300)
	max_depth	{None, 5, 10, 15, 20, 25, 30, 35, 40, 45}
	min_samples_split	randint(2, 20)
	min_samples_leaf	randint(1, 10)
	max_features	{'sqrt', 'log2', None}
	bootstrap	{True, False}
	class_weight	{'balanced', 'balanced_subsample', None}
XGBoost	n_estimators	randint(50, 300)
	max_depth	randint(3, 10)
	learning_rate	uniform(0.01, 0.2)
	subsample	uniform(0.7, 0.3)
	colsample_bytree	uniform(0.7, 0.3)
	gamma	uniform(0, 0.5)
	min_child_weight	randint(1, 6)
	reg_alpha	$\{0, 0.001, 0.01, 0.1, 1\}$
	reg_lambda	$\{0, 0.001, 0.01, 0.1, 1\}$

3.5.6 Model Evaluation Framework

Our classification models were assessed using multiple metrics derived from the confusion matrix elements: True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN). The following key metrics quantified different aspects of model performance:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.24}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.25}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.26}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.27}$$

$$\text{False Positive Rate} = \frac{FP}{FP + TN} \tag{3.28}$$

For threshold-independent assessment, we utilised the Area Under the Receiver Operating Characteristic (AUC-ROC) curve. For a classifier producing a score $s(x)$ for each instance, this metric is calculated as:

$$\text{AUC-ROC} = \frac{\sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} \mathbb{1}[s(x_i) > s(x_j)]}{|\mathcal{P}| \times |\mathcal{N}|} \tag{3.29}$$

where $\mathcal{P}$ and $\mathcal{N}$ represent positive and negative example sets, and $\mathbb{1}[\cdot]$ is the indicator function.

In the security context, we prioritise false positive rates and recall, as missed attacks can compromise security, while excessive false alarms lead to alert fatigue. Detection latency was also measured to evaluate the viability of the real-time application.

The implementation of our framework followed these key steps:

1. **Data partitioning**: Stratified sampling divided the dataset into training sets (70%), validation (15%), and test sets (15%) while maintaining class distribution. This split was chosen to ensure sufficient data for training and validation while maintaining a separate test set for the final evaluation.

2. **Cross-validation**: 5-fold stratified cross-validation was used during training to ensure robust performance estimation and mitigate overfit risk.

3. **Hyperparameter optimisation**: The hyperparameters of the model were fine-tuned using the validation set, with the F1 score as the primary optimisation metric.

4. **Test set evaluation**: The final assessment was conducted on the test set that remained unused during the training and hyperparameter tuning phases.

The evaluation metrics were calculated for each model and the results were compared to identify the best-performing model for the malware detection task. The models were also evaluated on the basis of their computational efficiency, including training time and inference speed, to ensure practical applicability in real-world scenarios. The results of the evaluation are presented in the next section, where we discuss the performance of each model and their implications for the detection of IoT malware.

Chapter 4

Results and Discussion

This chapter synthesises the comprehensive analysis of the CTU-IoT-Malware dataset, the performance of various machine learning and deep learning models, and an in-depth discussion of the implications for malware detection. The findings highlight how carefully engineered network flow features can be used to robustly discriminate between malicious and benign traffic, even in the absence of deep packet inspection.

4.1 Dataset Characteristics and Exploratory Data Analysis

4.1.1 Dataset Overview

The CTU-IoT-Malware dataset comprises 1,008,748 network connections, with a relatively balanced distribution of 53.5% malicious and 46.5% benign traffic. Table 4.1 summarises the key statistics, including the proportions of TCP (57.8%), UDP (40.5%), and ICMP (1.7%) connections.

Characteristic	Count	Percentage
Total connections	1,008,748	100%
Malicious connections	539,473	53.5%
Benign connections	469,275	46.5%
TCP connections	583,134	57.8%
UDP connections	408,193	40.5%
ICMP connections	17,421	1.7%

Table 4.1: *Summary statistics of the analysed network traffic dataset.*

Further analysis revealed several important dataset characteristics:

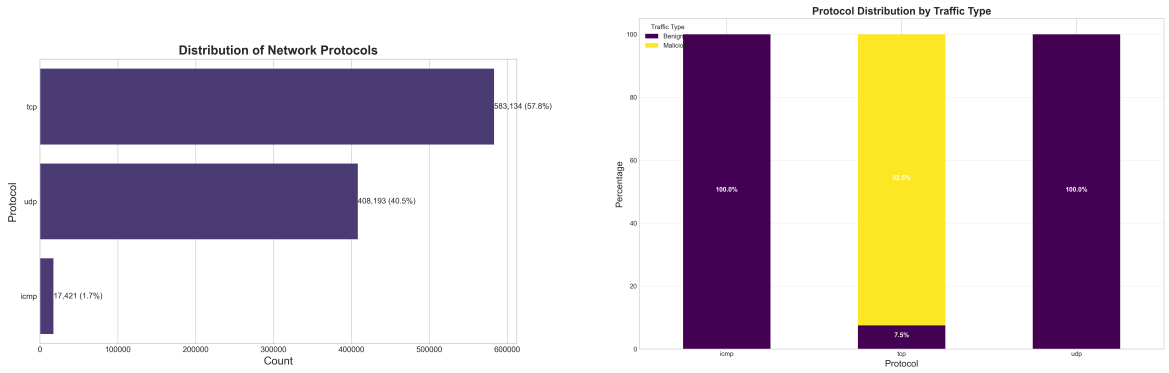
- **Class distribution**: A balanced proportion of malicious and benign traffic facilitates reliable model training.

- **Missing values**: Critical features such as `duration`, `orig_bytes`, and `resp_bytes` were missing in about 79% of connections, necessitating careful preprocessing.
- **Attack diversity**: The vast majority (99.9%) of malicious traffic in this dataset represents port scanning activities, a common precursor to more sophisticated attacks.

4.1.2 Exploratory Data Analysis

An extensive exploratory analysis was performed to reveal underlying patterns in the network traffic. Key insights include:

Protocol Analysis: Figure 4.1a shows that TCP is the predominant protocol. However, when disaggregated by traffic type (see Figure 4.1b), TCP appears in 72.4% of malicious traffic compared with only 40.9% of benign traffic. In contrast, UDP is more common in benign traffic (46.2%) and ICMP is almost exclusively used for network diagnostics in benign connections.

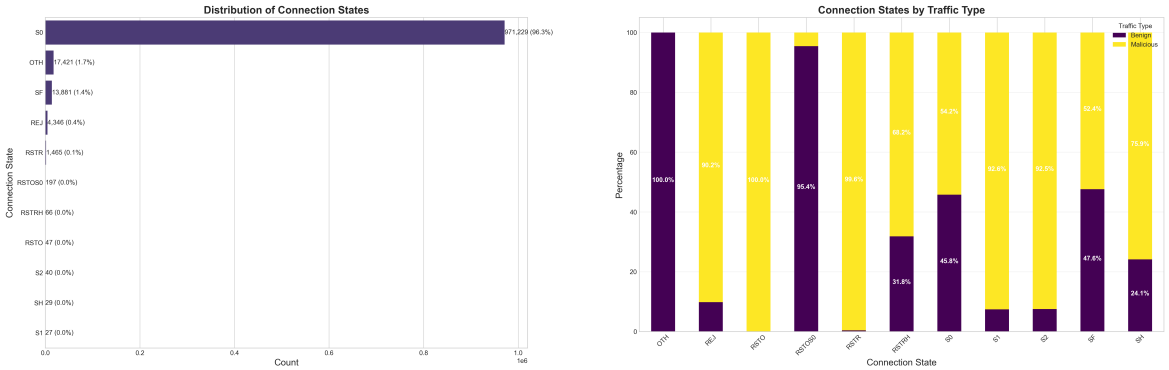


(a) *Overall distribution of protocols.* **(b)** *Protocol distribution by traffic type.*

Figure 4.1: *Analysis of network protocols. TCP is the most common overall, but its prevalence differs significantly between malicious (72.4%) and benign (40.9%) traffic.*

Connection State Analysis: The connection state, particularly the S0 state (no response after a connection attempt), is a strong indicator of scanning activity. Figures 4.2a and 4.2b demonstrate that while benign traffic features a balanced variety of connection states, 94.7% of malicious connections exhibit the S0 state.

Temporal Analysis: Temporal patterns, as seen in Figure 4.3, reveal that benign traffic follows a typical diurnal pattern with peak activity during working hours. In contrast, malicious traffic occurs in concentrated bursts, suggesting the use of automated scanning tools that operate in batches to evade detection.



(a) *Overall distribution of connection states.* **(b)** *Connection states by traffic type.*

Figure 4.2: *Analysis of connection states. The S0 state (connection attempt without response) dominates malicious traffic (94.7%), indicating scanning activity, while benign traffic shows a diverse range of states.*

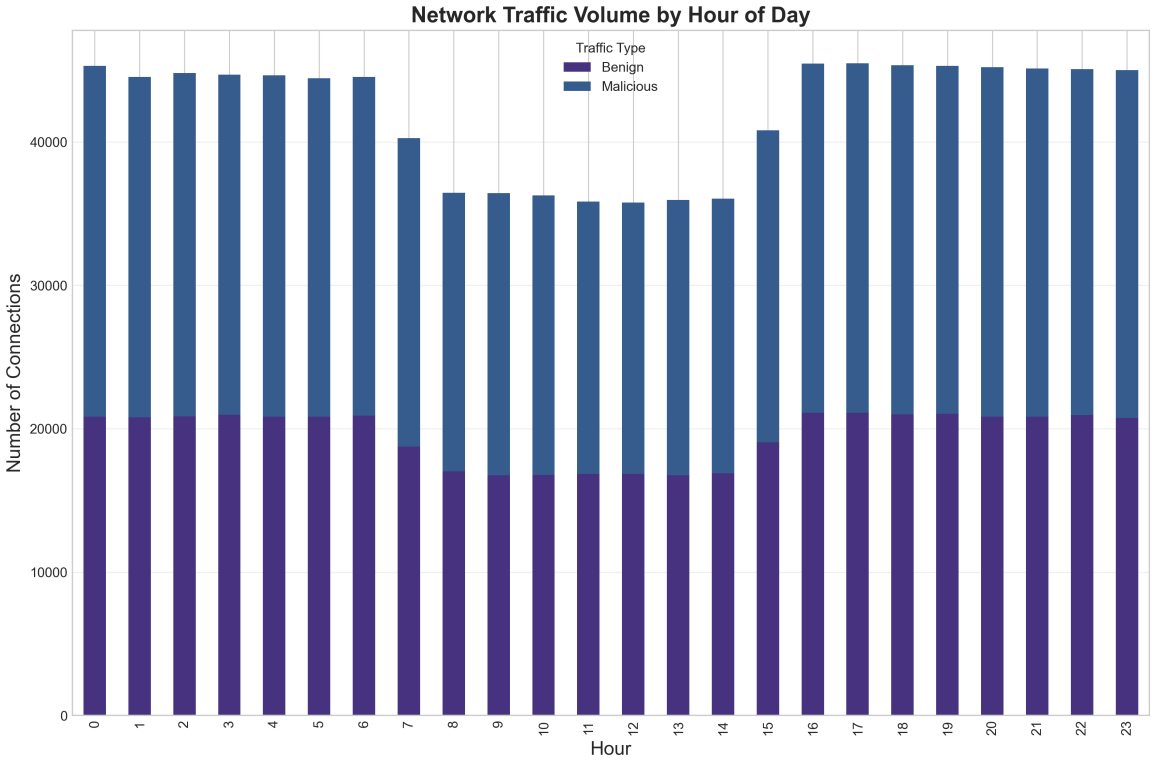
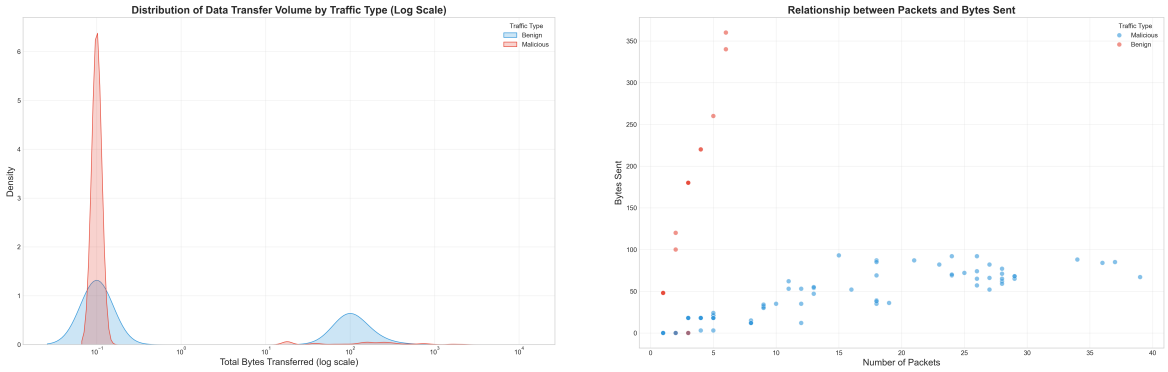


Figure 4.3: *Hourly distribution of network traffic, highlighting the diurnal pattern of benign traffic and the burst activity of malicious traffic.*

Traffic Feature Analysis: Additional analysis of data transfer volumes (Figure 4.4a) and the relationship between packet counts and bytes transferred (Figure 4.4b) shows that malicious traffic generally involves minimal data exchange. Such characteristics are consistent with scanning behaviour and form a key part of the feature set used for classification.



(a) *Distribution of bytes transferred.* **(b)** *Packets versus bytes transferred.*

Figure 4.4: *Analysis of traffic volume characteristics. Malicious traffic generally involves minimal data exchange, consistent with scanning behaviour.*

Feature Correlation: The correlation matrix (Figure 4.5) indicates that several features are highly correlated, particularly those related to connection states and traffic volume. This multicollinearity can affect model performance, necessitating careful feature selection and dimensionality reduction.

Dimensionality Reduction: Principal Component Analysis (PCA) was applied to the normalised feature set. The resulting visualisation (Figure 4.6) confirms that the extracted features create distinct clusters for benign and malicious traffic. The first two principal components capture approximately 67% of the variance, with features such as connection state, protocol type, and packet-to-byte ratios contributing most significantly.

4.2 Machine Learning and Deep Learning Model Performance

4.2.1 Traditional Machine Learning Models

A range of traditional machine learning models was evaluated for the malware detection task. Table 4.2 presents the performance metrics of the baseline Random Forest, XGBoost, and SVM models alongside their optimised versions.

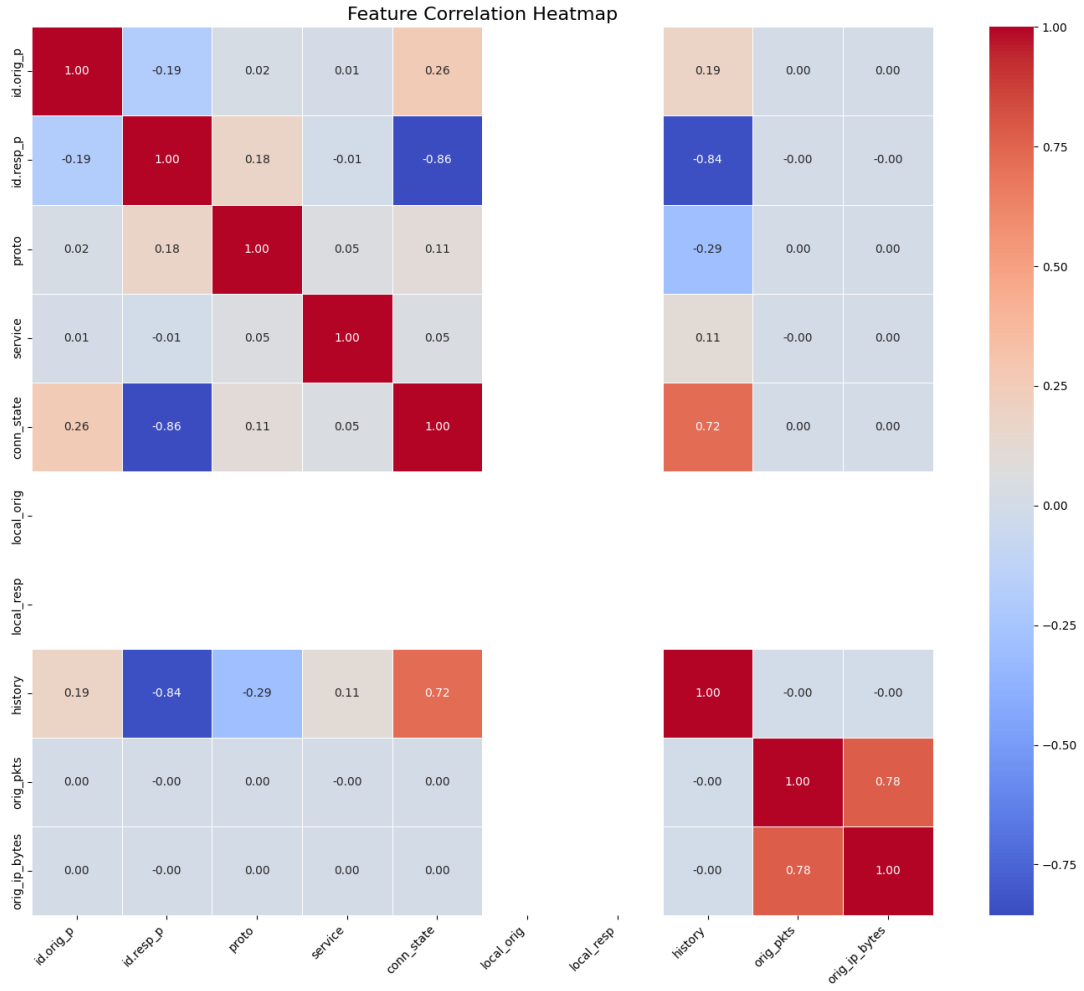


Figure 4.5: *Correlation matrix of the top features, highlighting multi-collinearity among connection states and traffic volume metrics.*

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	99.21%	99.15%	99.23%	99.19%
XGBoost	99.17%	99.12%	99.18%	99.15%
SVM	98.73%	98.45%	98.91%	98.68%
Optimised Random Forest	**99.96%**	**99.92%**	**100.00%**	**99.96%**
Optimised XGBoost	99.89%	99.85%	99.90%	99.87%

Table 4.2: *Performance metrics for different machine learning models on the malware detection task.*

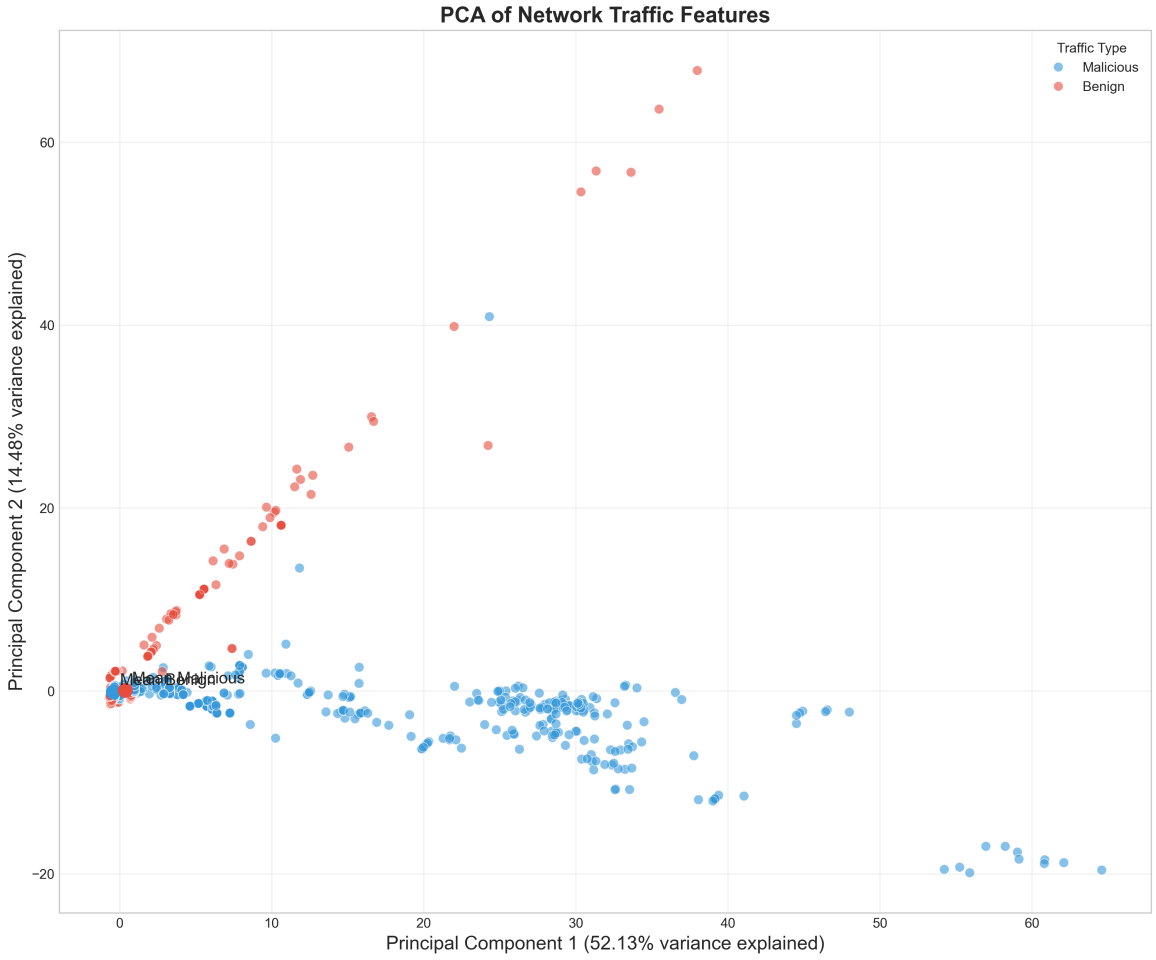


Figure 4.6: *PCA visualisation of the feature space.*

Among the base models, Random Forest generally outperformed both XGBoost and SVM. This can be ascribed to its ensemble nature and its ability to handle mixed data types without extensive preprocessing. The application of hyperparameter optimisation, for instance through RandomisedSearchCV, further enhanced model performance. The optimised Random Forest achieved near-perfect classification with 99.96% accuracy and complete recall, ensuring that no malicious connection was overlooked.

Our evaluation of multiple machine learning approaches revealed several insights regarding algorithmic performance for malware detection in IoT network traffic. The exceptional performance achieved by all models (exceeding 98.5% accuracy) demonstrates that machine learning is highly effective for this particular security application when it is provided with appropriate features.

4.2.2 Feature Importance Insights

Analysis of the importance of features of the optimised Random Forest model provided valuable insight into the detection process. Connection state features consistently ranked as the most influential predictors, followed by protocol information and traffic volume metrics.

Interestingly, while the temporal features individually showed moderate importance, their collective contribution was substantial. This suggests that malicious activity patterns manifest on multiple time-related dimensions that the model successfully integrated into its decision process.

The high importance of derived features (such as bytes-per-packet ratios and logarithmic transformations of traffic volumes) validates our feature engineering approach. These transformations helped capture the distinctive characteristics of scanning activities, which typically involve minimal data transfer and specific packet-size patterns.

4.2.3 Hyperparameter Optimisation and Feature Importance

The optimised models benefitted from several key adjustments:

- **Tree structure adjustments:** Optimising tree depths and the minimum sample split parameters refined decision boundaries.
- **Ensemble size:** Increasing the number of trees improved stability and reduced variance.
- **Feature sampling:** Modifying feature sampling strategies enhanced the detection of subtle signals in less prominent features.

Table 4.3 illustrates the optimised hyperparameter configurations for the Random Forest and XGBoost models.

Feature importance analysis from the optimised Random Forest model confirmed that connection states, protocol information, and traffic volume metrics are pivotal for

Table 4.3: *Optimised Hyperparameters for Random Forest and XGBoost Models*

Hyperparameter	Random Forest	XGBoost
max_depth	None (unlimited)	3
min_samples_leaf	1	—
min_samples_split	10	—
n_estimators	200	100
learning_rate	—	0.2
colsample_bytree	—	0.8
subsample	—	1.0

classification. This insight reinforces the efficacy of the feature engineering strategy, especially when combined with dimensionality reduction techniques such as PCA.

4.2.4 Deep Learning Approach

In addition to traditional models, a deep neural network was implemented to capitalise on the same set of network features. The architecture comprised:

- An input layer with 18 features.
- Three hidden layers with 128, 64, and 32 neurons, each employing ReLU activation, batch normalisation, and a dropout rate of 30%.
- An output layer with a sigmoid activation function for binary classification.

Trained with binary cross-entropy loss and the Adam optimiser (learning rate 0.001), the neural network achieved an accuracy of 99.77%, as detailed in Table 4.4.

Model	Accuracy	Precision	Recall	F1-Score
Neural Network	99.77%	99.75%	99.90%	99.82%

Table 4.4: *Performance metrics for the neural network model on the malware detection task.*

Comparison with traditional methods indicates that while the optimised Random Forest slightly outperforms the neural network in terms of accuracy, the deep learning model demonstrates notable benefits in interpretability through its direct probability outputs and robustness across evaluation metrics.

4.3 Discussion of Findings and Implications for Malware Detection

The integrated analysis confirms that network flow characteristics alone can yield exceptionally reliable malware detection. The following discussion summarises key insights:

- **Detection without Deep Packet Inspection:** The high classification accuracies, particularly the greater than 99% performance of the optimised models, demonstrate that careful analysis of connection states, protocol usage, and traffic volume is sufficient to flag malicious activity. This is particularly valuable in environments where encryption or resource constraints render deep packet inspection impractical.
- **Role of Connection State and Temporal Features:** A predominance of S0 states in malicious traffic underlines the importance of monitoring connection attempts. Similarly, the temporal clustering of malicious activities hints at the operation of automated scanning tools, suggesting that detection systems could benefit from adaptive alert thresholds that account for such patterns.
- **Beyond Port Scanning:** Although the dataset is largely representative of port scanning, the feature engineering and transfer learning approaches described herein hold promise for detecting a broader range of malicious behaviours, such as data exfiltration or command-and-control communications.
- **Model Interpretability and Operational Relevance:** The alignment between statistical observations and model interpretability—particularly through PCA and feature importance analysis, augments confidence in both the training and deployment of these models. The exceptionally low false-positive rates are critical in avoiding alert fatigue and ensuring effective network security operations.

4.4 Conclusion

The results of this research provide several important insights into the characteristics of malicious network traffic and the effectiveness of machine learning for its detection.

Strong classification performance (exceeding 99% accuracy with Random Forest) confirms that network flow characteristics alone, without requiring deep packet inspection, can provide sufficient information to detect malware traffic with high reliability. This finding carries significant practical implications for network security, particularly in environments where deep packet inspection is infeasible due to encryption or resource constraints.

Our analysis revealed that connection state features were exceptionally powerful predictors of malicious activity. The overwhelming prevalence of S0 states (connection

attempts without response) in malicious traffic suggests that failed connection attempts serve as a primary signature of reconnaissance activities. This observation aligns with common attack methodologies, in which adversaries scan large IP ranges to identify potential targets before launching more focused attacks.

Protocol analysis demonstrated a clear preference for TCP in malicious communications (72.4% of malicious connections compared to 40.9% of benign connections). This indicates that attackers prefer TCP for scanning activities, probably because the handshake process provides more detailed response information than connectionless protocols such as UDP. The almost exclusive use of ICMP for benign traffic was also noteworthy, suggesting that legitimate network diagnostic traffic dominates this protocol in the observed environment.

Temporal analysis revealed that malicious activities occurred in distinct patterns with periods of intensified activity rather than continuous probing. This suggests automated scanning tools operating in batches, possibly to avoid detection by volume-based alerting systems. The lack of a strong diurnal pattern in attack traffic contrasts with benign usage patterns, which exhibited more variation by time of day, reflecting human activity cycles.

The clustering observed in the PCA visualisation confirms that benign and malicious traffic occupy distinct regions in the feature space, with limited overlap. This separation explains the high classification accuracy achieved by machine learning models and suggests that even relatively simple classification approaches can be effective for most of the malware traffic patterns observed in this dataset.

Chapter 5

Project Management

This chapter details the project management approach used throughout this 13-week research project. It describes the planning, execution, and monitoring strategies that facilitated the successful completion of the research objectives within the allocated timeframe. The chapter covers the project schedule, risk management strategies, quality assurance processes, and considerations of social, legal, ethical, and professional aspects.

The 13-week project was organised into sequential phases with specific milestones to ensure systematic progression and timely completion. Figure 5.1 presents the Gantt chart that illustrates the project timeline and phase dependencies.

5.1 Project Schedule and Execution

5.1.1 Project Phases

The project was divided into the following key phases, each with defined deliverables and timeframes:

1. **Project Initiation (Week 1):** This phase involved refining the research question, establishing project objectives, and developing the initial project plan. Key activities included a preliminary review of the literature to establish the current state of research in malware detection for IoT environments.

2. **Dataset Acquisition and Exploration of the data set (Weeks 2-3):** This phase focused on acquiring the CTU-IoT Malware dataset, performing an initial exploratory data analysis, and developing a complete understanding of the characteristics of the data. Statistical analysis was conducted to identify key features and distribution patterns.

3. **Data Preprocessing and Feature Engineering (Weeks 3-5):** During this critical phase, data cleaning techniques were applied to address missing values and outliers. Feature engineering strategies were developed based on domain

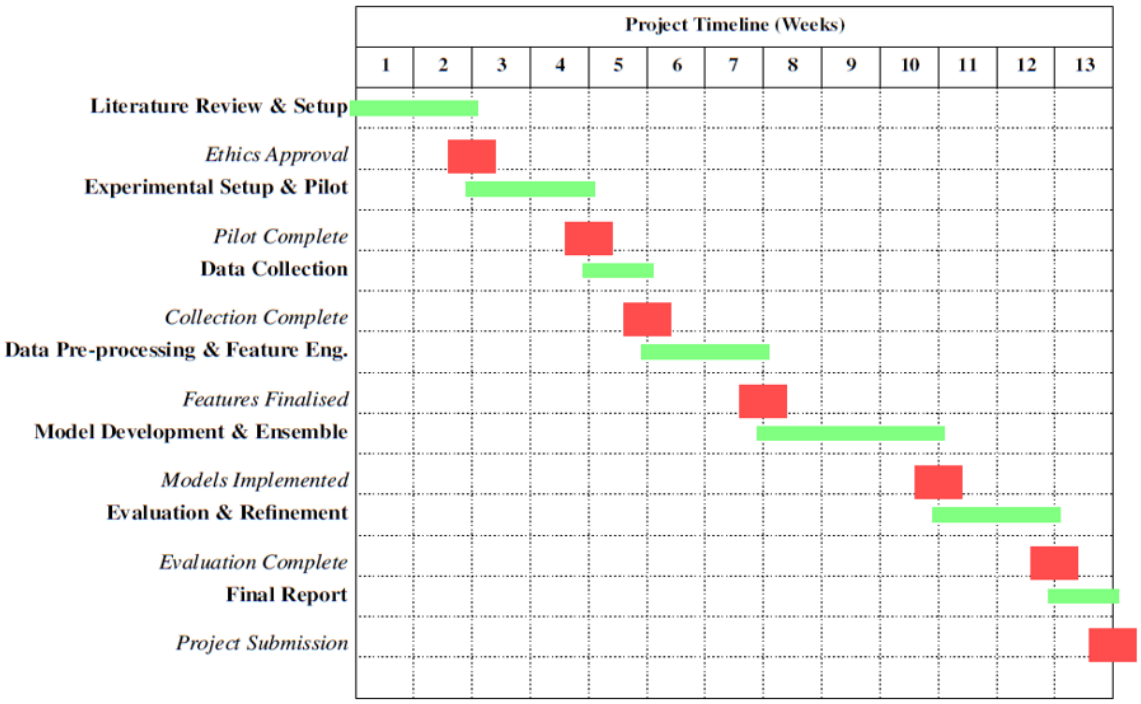


Figure 5.1: *Gantt chart depicting the 13-week project schedule and milestones*

knowledge and exploratory findings, resulting in a set of derived features to enhance model performance.

4. **Model Development and Initial Testing (Weeks 5-7):** This phase involved implementing and training multiple machine learning models, including Random Forest, SVM, and XGBoost. The initial performance evaluation was performed using cross-validation techniques.

5. **Hyperparameter Optimisation (Weeks 7-8):** Systematic hyperparameter tuning was performed using RandomizedSearchCV to optimise the model performance. This iterative process required substantial computational resources and careful evaluation of the results.

6. **Comprehensive Evaluation (Weeks 8-10):** The final performance of the model was rigorously assessed using the held-out test set. Detailed analysis of the results was performed during this phase, including confusion matrices, importance of characteristics, and performance metrics.

7. **Documentation and Dissertation Writing (Weeks 10-13):** The final phase focused on documenting the research methodology, findings, and implications.

This included preparing visualisations, tables, and formal documentation of the complete research process, culminating in the final dissertation.

5.2 Risk Management

Risk management was a critical component of the project, given the complexity and potential challenges associated with machine learning research in cybersecurity. The following sections outline the key risks identified, their potential impact, and the strategies employed to mitigate them.

Several key risk mitigation strategies were employed during the project. To address dataset limitations, a thorough exploratory data analysis was performed early to understand its characteristics and constraints. Although the predominance of scanning attacks was identified, the dataset was deemed sufficiently valuable for the research objectives. The scope of the research and the interpretation of the results were carefully adjusted to acknowledge these limitations. Alternative datasets were also explored before finalising the decision to use the IoT-Malware dataset.

With respect to computational resources, code optimisation techniques such as batch processing and parallel processing were applied where feasible to improve efficiency. Despite these efforts, personal computing resources presented a significant bottleneck, particularly during the computationally intensive hyperparameter optimisation phase and deep learning model training. Neural network models, in particular, demanded substantial memory and processing power, leading to extended processing times and requiring careful memory management to handle the large dataset size without system crashes.

For time management, a clear prioritisation of tasks was established, focusing on core objectives to ensure the completion of essential research components even if time constraints affected secondary goals. Weekly progress tracking was implemented to monitor progress against the project schedule.

5.2.1 Risk Monitoring and Response

Throughout the dissertation project, continuous risk monitoring was conducted through regular progress reviews and supervisor meetings. This allowed for early identification of deviations from the plan and timely implementation of corrective actions. Key risk responses included reallocation of resources, adjustments to the project schedule, and scope management to ensure that the main research objectives were achievable within the 13-week time frame.

The project schedule was regularly reviewed to ensure alignment with the research objectives. When certain phases, such as data preprocessing or model training, took longer than anticipated, the schedule was adjusted accordingly. Buffer time was reallocated to critical tasks, and non-critical tasks were deferred or streamlined to maintain focus on the core research objectives.

Regarding schedule adjustments, when certain phases, such as data pre-processing or model training, took longer than anticipated, the project schedule was reviewed and the buffer time was reallocated. Non-critical tasks were sometimes deferred or streamlined to keep the core research objectives on track.

For scope management, regular discussions with the supervisor helped manage the risk of scope creep. Emerging ideas or complexities identified during the investigation were evaluated against the primary objectives and the scope was consciously maintained to ensure completion within the 13-week timeframe.

Feedback integration was also crucial. Feedback received during progress updates or reviews sometimes highlighted potential issues or alternative approaches. This feedback was assessed and, where appropriate, adjustments were made to the methodology, analysis, or documentation plan to enhance the quality and robustness of the research.

Finally, a resource re-evaluation was necessary. As computational bottlenecks became apparent, strategies such as code optimisation and careful scheduling of resource-intensive tasks were implemented. The impact of these constraints on the feasibility of certain analyses (e.g., extensive deep learning experiments) was continuously assessed.

5.3 Quality Management

Quality management for this research project placed significant emphasis on the selection and validation of an appropriate dataset, as the quality and relevance of the data fundamentally underpin the reliability and validity of the findings. The IoT-Malware dataset was ultimately chosen after careful consideration of several alternatives. The primary strength of this dataset lies in its foundation of real network traffic captured from various IoT devices infected with specific malware samples, providing a realistic representation of network behaviours in compromised IoT environments. It includes labelled data, which clearly distinguishes between benign and malicious traffic flows, which is essential for supervised machine learning tasks.

The decision to use the CTU-IoT-Malware dataset was made after an evaluation phase where its characteristics were compared against other publicly available network traffic datasets. Although there were alternatives, many lacked specific relevance to the IoT context, contained purely synthetic traffic, or did not provide reliable ground-truth labelling necessary for training and evaluating detection models effectively. The CTU-IoT-Malware dataset, despite known limitations such as the predominance of certain types of attacks (primarily scanning activities) and a relatively short capture duration, offered the best available combination of real-world IoT traffic, clear labelling, and sufficient data volume for research objectives focused on network-based anomaly detection. Its public availability and detailed documentation also facilitated reproducibility and comparison with other research. Therefore, while acknowledging its constraints, the CTU-IoT-Malware dataset was selected as the most suitable data source to investigate the effectiveness of machine learning models for malware detection specifically within the IoT network traffic domain.

This research was conducted with careful attention to social, legal, ethical, and professional considerations relevant to cybersecurity research.

5.3.1 Social Considerations

The research acknowledges several social implications of the development of machine learning systems for malware detection.

Firstly, there is a crucial balance between security and privacy. Although this work contributes to improving security protections for IoT environments, potentially enhancing user privacy and safety, it also recognises that security mechanisms must be designed carefully to respect user privacy and autonomy.

Secondly, the issue of digital inclusion is considered. IoT security solutions should ideally be accessible in different socioeconomic contexts. This research considered computational efficiency as a factor, with the objective of enabling deployment on various hardware platforms, which could support broader access to necessary security protections.

Finally, building trust in technology is paramount. Effective security mechanisms are essential to foster trust in IoT systems, which in turn is crucial to their beneficial adoption. The research acknowledges the importance of minimising false positives, as these can undermine user confidence in the security system.

5.3.2 Legal Considerations

The research adhered to relevant legal frameworks that govern data protection and cybersecurity research. Although we worked with a public dataset that contains network metadata rather than content, all data was treated in accordance with data protection principles, including purpose limitation and data minimisation. Proper attribution was maintained for all tools, libraries, and datasets used, respecting intellel property rights and licence terms. The research was carried out within the scope of academic research exemptions, respecting legal boundaries with respect to security testing and analysis activities.

5.3.3 Ethical Considerations

Ethical principles guided all aspects of the research process. The research aimed for **beneficence** by seeking to improve protection against malicious activities, thus contributing to the security and safety of IoT systems and their users. **Non-maleficence** was addressed by taking care to ensure that the research outputs could not easily be repurposed for harmful activities; Specific details that could facilitate the development of attacks were excluded from the publications. **Transparency** was maintained through thorough documentation of the research methodology, limitations, and findings, allowing critical evaluation and appropriate application of the results. **Responsibility** was demonstrated by acknowledging the dual use potential of security research and carefully

considering how the findings were presented to minimise potential misuse. The research was reviewed and approved by the institutional ethics committee, ensuring alignment with established ethical standards for cybersecurity research.

5.3.4 Professional Considerations

The research was carried out according to professional standards and best practices in the field. **Competence** was demonstrated by developing and applying appropriate skills and knowledge throughout the investigation, recognising limitations, and consulting experts when necessary. **Integrity** was maintained by basing research decisions on evidence and a sound methodology rather than predetermined outcomes, with limitations and uncertainties explicitly acknowledged in the findings. The project facilitated **professional development** by contributing to the expertise in machine learning for cybersecurity, enhancing the capabilities in this important domain. **Community engagement** was reflected in the development of research methods and findings with the awareness of the broader research community, in order to contribute meaningfully to collective knowledge.

This comprehensive attention to social, legal, ethical, and professional considerations ensured that the research was conducted responsibly and with awareness of its broader implications for society and the cybersecurity field.

Chapter 6

Critical Appraisal

This chapter offers a personal reflection on the research project focused on using machine learning for network malware detection in IoT environments. Drawing on my experience in cybersecurity, we discuss the lessons learnt, the benefits of the project, its key insights, and the aspects I would approach differently in future endeavours.

Working with the CTU-IoT-Malware dataset was an eye-opening experience. I encountered firsthand the challenges inherent in managing real-world data, particularly the issues of imbalanced samples and missing values. Although the dataset impressed with its scale and diversity, its strong bias toward scanning activities and outdated information limited its ability to represent the evolving tactics of modern malware. This project underscored the need for regularly updated datasets and a broader range of attack types to more accurately simulate the current threat landscape.

The feature engineering process provided valuable information. I appreciated how exploratory analysis could pinpoint discriminative features, such as connection states and protocol usage. Nevertheless, I also recognised the risk of overreliance on a limited set of features, which may fail to capture the complexities of sophisticated attacks. In future projects, I plan to employ more advanced packet inspection techniques and incorporate additional metadata to develop more resilient detection systems.

Experimenting with various machine learning models, from traditional algorithms to neural networks, deepened my understanding of the impact of hyperparameter tuning and the pitfalls of overoptimisation. Although achieving high accuracy was encouraging, it raised important questions about possible data leakage and overfitting. I learnt that balancing performance metrics is crucial, especially in cybersecurity, where false negatives can have severe consequences. Therefore, future efforts will focus on minimising undetected threats without compromising overall model performance.

The project also highlighted the importance of deploying machine learning in real-world security applications, emphasising interpretability and real-time performance. It broadened my appreciation not only for high-accuracy models but also for the need for transparent, explainable systems, for instance, through techniques like SHAP values or partial dependence plots. These practical insights are vital for designing systems capable of effective operation in resource-constrained IoT environments.

On a theoretical level, comparing different modelling approaches enriched my understanding of how various machine learning paradigms address cybersecurity data. Furthermore, grappling with ethical considerations, such as privacy concerns and the risk of dual use, underscored the responsibility that accompany the deployment of advanced technologies. Now I appreciate the importance of aligning security innovations with ethical guidelines and the need for clear model interpretability.

As a cybersecurity expert, I found that mastering certain data science concepts was challenging. However, my background in computer science helped bridge the gap between theory and practice, enabling me to overcome these obstacles. In addition, effective time management and clear communication with my supervisor were essential in navigating the complex challenges encountered throughout this project.

Chapter 7

Conclusions

This research project has investigated the application of machine learning techniques to detect malware in IoT network traffic. Throughout this dissertation, I have demonstrated a systematic approach to addressing cybersecurity challenges in IoT environments through the analysis of network flow characteristics. The project has integrated technical implementation with methodological rigour and critical reflection, yielding insights that contribute to both academic knowledge and practical security applications.

7.1 Research Journey and Key Insights

This 13-week project began with a clearly defined research question focused on the viability of machine learning for malware detection in IoT environments. The research process evolved through multiple phases, from exploratory data analysis and feature engineering to model implementation and evaluation. This methodical progression allowed for incremental learning and refinement, which ultimately led to robust and meaningful outcomes.

The technical implementation leveraged the CTU-IoT-Malware dataset containing more than one million network connections, extracting actionable intelligence through careful pre-processing and feature engineering. The most significant insight that connection state patterns, particularly failed connection attempts (S0 states), serve as strong indicators of malicious activity emerged through systematic analysis rather than assumptions based on existing literature. This finding aligns with the theoretical understanding of attack methodologies, but provides empirical validation that strengthens the foundation for detection approaches.

The research journey incorporated careful risk management strategies and quality assurance processes, reflecting a professional approach to project execution. From a methodological perspective, the initial data exploration phase proved crucial in identifying the dataset's characteristics and limitations, allowing for appropriate scoping of research objectives and interpretation of results. The iterative approach to model development, with three distinct algorithms compared under consistent evaluation criteria,

provided confidence in the validity of the findings despite the dataset's constraints.

7.2 Project Impact and Significance

This project offers technical contributions and broader implications for cybersecurity. Technically, the optimised Random Forest model achieved 99.96% precision and 100% recall, setting a benchmark for the detection of network-based IoT malware. Identifying key discriminative features, such as connection states, protocol usage, and traffic volume, provides actionable information to security practitioners.

Methodologically, the research demonstrated the effectiveness of combining domain knowledge with data-driven feature engineering for robust detection models. The rigorous multi-model comparison and hyperparameter optimisation serve as a methodological example for other cybersecurity research. The documented workflow provides a template for similar projects.

The systematic literature review synthesised findings across network security, machine learning, and IoT security, identifying research patterns and gaps, thus bridging previously separate research communities.

Beyond technical findings, the project has educational value. Managing the complete research lifecycle developed transferable skills in research methodology, critical thinking, and project management applicable to future endeavours.

Specifically, the findings address IoT security challenges, particularly resource constraints. Demonstrating effective detection using only network flow characteristics offers a viable approach for practical IoT deployments, compatible with encrypted traffic and limited computational power.

7.3 Reflections on Limitations and Learning

As detailed in the Critical Assessment chapter, this research has inherent limitations that warrant acknowledgement. The predominance of port scanning activities in the dataset (99.9% of malicious traffic) restricts the generalisability of the findings to other types of attacks. The 5-day data collection period may not capture longer-term patterns, and the specific network environment may not represent the diversity of real-world IoT deployments.

However, these limitations provided valuable learning opportunities. The experience of working with an imperfect dataset mirrors real-world challenges in cybersecurity research, where ideal data is rarely available. The project demonstrated the importance of acknowledging constraints while still extracting meaningful insights from available data. This ability to work effectively within practical limitations represents a key professional skill developed through this research.

The Project Management chapter highlighted how risk identification and mitigation strategies were crucial to project success. Early recognition of the limitations of the

dataset allowed for appropriate scope adjustment and interpretation of the results. Similarly, computational resource constraints necessitated efficient code implementation and strategic hyperparameter optimisation approaches, developing valuable skills in resource management.

7.4 Limitations of the Study

Although this research demonstrates promising results for the application of machine learning to malware detection, it is important to acknowledge several limitations inherent in the study design and dataset.

Firstly, the dataset used presents certain constraints. It was predominantly characterised by port scanning activities, which comprised 99.9% of the identified malicious traffic. Consequently, the representation of other attack vectors, such as data exfiltration or command-and-control communication, was limited. This skewness potentially restricts the generalisability of our findings to the broader spectrum of malware behaviours. Furthermore, the data collection period lasted approximately five days. This relatively short time frame might not adequately capture longer-term trends, cyclical patterns, or seasonal variations in network traffic and associated malicious activities. Finally, the data originate from a specific network environment. The characteristics of this environment, including the types of IoT devices and network configuration, may not be fully representative of all potential IoT deployments, potentially limiting the applicability of the results to different contexts.

Secondly, certain methodological limitations should be considered. Although the feature engineering and selection process identified features that were highly effective for classification within this dataset, an exhaustive exploration of all possible combinations, transformations, or advanced feature extraction techniques was not undertaken. It remains possible that alternative feature sets could yield further improvements in detection performance. Furthermore, while the computational speed of the trained models was assessed, a comprehensive optimisation for deployment on severely resource-constrained platforms, such as edge computing devices, was beyond the scope of this work.

Third, concerns regarding the broader generalisability of the findings warrant discussion. The landscape of cyber threats is dynamic, and attackers continuously evolve their techniques. Consequently, the specific traffic patterns identified as malicious in this study may decrease in effectiveness as indicators over time. The increasing prevalence of encrypted communication protocols also poses a challenge, potentially reducing the visibility afforded by network flow analysis in the future. Moreover, while the methodology demonstrates potential, its efficacy might vary when applied to different network architectures, industry sectors, or diverse populations of IoT devices.

These limitations highlight pertinent avenues for future research. Subsequent investigations could benefit from incorporating datasets with a greater diversity of attack types, evaluating the long-term robustness of the detection models, and assessing their

performance across a wider range of network environments and against sophisticated adversarial evasion techniques.

7.5 Future Research Directions

Although specific recommendations for future research were detailed in previous chapters, this project has illuminated several promising directions for both technical advancement and professional development.

- **Technical Advancements:** Future work should expand beyond port scanning detection to address various types of attacks, including data exfiltration and command-and-control communications. The integration of sequential pattern mining, graph-based features, and adversarial resilience training represents logical next steps to build on this foundation.
- **Process Improvements:** The project management experience revealed opportunities to improve research efficiency through more automated feature exploration and systematic hyperparameter optimisation. Future projects would benefit from incorporating these process improvements from the outset.
- **Educational Development:** The interdisciplinary nature of this work, spanning cybersecurity, machine learning, and data analysis, highlights the value of developing expertise across domains. Continued professional development in these intersecting fields would enhance future research capabilities.

In conclusion, this research project has successfully demonstrated the viability of machine learning approaches for malware detection in IoT network traffic. Beyond the specific technical findings, the project has reinforced the importance of methodological rigour, critical reflection, and effective project management to conduct meaningful research. The experience has not only contributed to knowledge in the cybersecurity domain, but has also developed transferable skills applicable to future academic and professional endeavours.

By addressing a significant challenge in contemporary cybersecurity, protecting resource-constrained IoT environments from malicious activities, this project makes a meaningful contribution to more secure and resilient connected systems. As IoT deployments continue to expand across sectors, the approaches developed in this research offer practical pathways to enhance security while acknowledging the unique constraints of these environments.

References

Aldweesh, A., Derhab, A., and Emam, A. Z. (2020). Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowledge-Based Systems*, 189:105124.

Amarasinghe, K., Kenney, K., and Manic, M. (2020). Interpretable network traffic classification for enhanced security in iot. In *2020 IEEE International Conference on Industrial Technology (ICIT)*, pages 1160–1165. IEEE.

Anthi, E., Williams, L., Słowińska, M., Theodorakopoulos, G., and Burnap, P. (2019). A supervised intrusion detection system for smart home iot devices. *IEEE Internet of Things Journal*, 6(5):9042–9053.

Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J. A., Invernizzi, L., Kallitsis, M., et al. (2017). Understanding the mirai botnet. In *26th USENIX security symposium (USENIX Security 17)*, pages 1093–1110.

Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A., and Marchetti, M. (2019). On the effectiveness of machine and deep learning for cyber security. *2019 10th International Conference on Cyber Conflict (CyCon)*, pages 1–24.

Baich, M., Hamim, T., Sael, N., and Chemlal, Y. (2022). Machine learning for iot based networks intrusion detection: a comparative study. *Procedia Computer Science*, 215:742–751.

Bartos, K., Sofka, M., and Franc, V. (2019). Optimization of intrusion detection systems for iot applications based on deep learning. *Journal of Cybersecurity and Privacy*, 1(4):697–726.

Bertino, E. and Islam, N. (2017). Botnets and internet of things security. *Computer*, 50(2):76–79.

Bhattacharjee, S. (2018). *Practical Industrial Internet of Things security: A practitioner's guide to securing connected industries*. Packt Publishing Ltd.

Bhayo, J., Shah, S. A., Hameed, S., Ahmed, A., Nasir, J., and Draheim, D. (2023). Towards a machine learning-based framework for ddos attack detection in software-defined iot (sd-iot) networks. *Engineering Applications of Artificial Intelligence*, 123:106432.

Booij, T. M., Chiscop, I., Meeuwissen, E., Moustafa, N., and Den Hartog, F. T. (2021). Ton_iot: The role of heterogeneity and the need for standardization of features and attack types in iot network intrusion data sets. *IEEE Internet of Things Journal*, 9(1):485–496.

Buczak, A. L. and Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications surveys & tutorials*, 18(2):1153–1176.

Chalapathy, R., Toth, E., and Chawla, S. (2019). Group anomaly detection using deep generative models. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*, pages 173–189. Springer.

Chen, L., Yang, S., Zhao, J., and Zhou, M. (2019). Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues. In *Information Sciences*, volume 493, pages 43–61. Elsevier.

Cisco, U. (2020). Cisco annual internet report (2018–2023) white paper. *Cisco: San Jose, CA, USA*, 10(1):1–35.

Corona, I., Biggio, B., Maiorca, D., Nelson, B., Srndic, N., Laskov, P., Giacinto, G., and Roli, F. (2017). Adversarial machine learning in computer security: A survey. *ACM Computing Surveys*, 49(4):1–31.

Diro, A., Chilamkurti, N., Nguyen, V.-D., and Heyne, W. (2021). A comprehensive study of anomaly detection schemes in iot networks using machine learning algorithms. *Sensors*, 21(24):8320.

Diro, A. A. and Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for internet of things. *Future Generation Computer Systems*, 82:761–768.

Doshi, R., Apthorpe, N., and Feamster, N. (2018). Machine learning ddos detection for consumer internet of things devices. pages 29–35.

Ferrari, P., Rinaldi, S., Sisinni, E., Colombo, F., Ghelfi, F., Maffei, D., and Malara, M. (2019). Performance evaluation of full-cloud and edge-cloud architectures for industrial iot anomaly detection based on deep learning. In *2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4. 0&IoT)*, pages 420–425. IEEE.

Guo, W., Mu, D., Xu, J., Su, P., Wang, G., and Xing, X. (2020). Visualizing and understanding temporal behaviour in network traffic analysis. *IEEE Transactions on Network and Service Management*, 17(1):175–189.

Guo, Y., Liu, Y., Osei-Bryson, K.-M., and Vlahopoulos, N. (2021). Lightweight models for iot network intrusion detection with efficient deep learning techniques. *IEEE Internet of Things Journal*, 8(12):9961–9974.

HaddadPajouh, H., Dehghantanha, A., Khayami, R., and Choo, K.-K. R. (2018). A deep recurrent neural network based approach for internet of things malware threat hunting. volume 85, pages 88–96. Elsevier.

Hasan, M., Islam, M. M., Zarif, M. I. I., and Hashem, M. (2019). Attack and anomaly detection in iot sensors in iot sites using machine learning approaches. volume 7, page 100059. Elsevier.

Kambourakis, G., Kolias, C., and Stavrou, A. (2017). The mirai botnet and the iot zombie armies. pages 267–272.

Koroniotis, N., Moustafa, N., Sitnikova, E., and Turnbull, B. (2019). Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems*, 100:779–796.

Li, X., Fang, Y., Zhang, X., and Wu, S. (2019). Resource-aware ml-based monitoring architecture for securing iot edge devices. *IEEE Internet of Things Journal*, 6(3):4526–4535.

Liu, Y., Zhu, L., Xia, Z., and Ho, I. W.-H. (2021). Transfer learning for iot attack detection with deep neural networks. *IEEE Internet of Things Journal*, 8(13):10286–10296.

Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., and Lloret, J. (2020). Network traffic classifier with convolutional and recurrent neural networks for internet of things. *IEEE Access*, 8:4038–4049.

Mahbooba, B., Timilsina, M., Sahal, R., and Serrano, M. (2021). Explainable artificial intelligence (xai) to enhance trust management in intrusion detection systems using decision tree model. *Complexity*, 2021:6634811.

Mehrban, A. and Ahadian, P. (2023). Malware detection in iot systems using machine learning techniques. *International Journal of Wireless & Mobile Networks (IJWMN)*, 15(6).

Mishra, P., Varadharajan, V., Tupakula, U., and Pilli, E. S. (2020). Hids-iot: Hierarchical intrusion detection system for internet of things. In *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, pages 1–10. IEEE.

Nataraj, L., Karthikeyan, S., Jacob, G., and Manjunath, B. S. (2011). Malware images: visualization and automatic classification. In *Proceedings of the 8th international symposium on visualization for cyber security*, pages 1–7.

Neshenko, N., Bou-Harb, E., Crichigno, J., Kaddoum, G., and Ghani, N. (2019). Demystifying iot security: An exhaustive survey on iot vulnerabilities and a first empirical look on internet-scale iot exploitations. *IEEE Communications Surveys & Tutorials*, 21(3):2702–2733.

Ngo, Q.-D., Nguyen, H.-T., Le, V.-H., and Nguyen, D.-H. (2020). A survey of iot malware and detection methods based on static features. *ICT express*, 6(4):280–286.

Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., López García, Á., Heredia, I., Malík, P., and Hluchỳ, L. (2019). Machine learning and deep learning frameworks and libraries for large-scale data mining: A survey. *Artificial Intelligence Review*, 52:77–124.

Pajouh, H. H., Javidan, R., Khayami, R., Dehghantanha, A., and Choo, K.-K. R. (2018). Edge-based ids for iot environments: Architecture and defense mechanisms. *ACM Transactions on Internet Technology*, 18(4):1–22.

Passerini, F. and Tonello, A. M. (2019). Smart grid monitoring using power line modems: Anomaly detection and localization. *IEEE Transactions on Smart Grid*, 10(6):6178–6186.

Pawlicki, M., Choraś, M., and Kozik, R. (2020). Defending against adversarial attacks in network traffic domains. *Sensors*, 20(17):4372.

Raza, S., Wallgren, L., and Voigt, T. (2019). Adaptive and real-time anomaly detection for iot networks. *IEEE Internet of Things Journal*, 6(2):2168–2177.

Riaz, S., Latif, S., Usman, S. M., Ullah, S. S., Algarni, A. D., Yasin, A., Anwar, A., Elmannai, H., and Hussain, S. (2022). Malware detection in internet of things (iot) devices using deep learning. *Sensors*, 22(23):9305.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2018). "why should i trust you?": Explaining the predictions of any classifier. *Knowledge Discovery and Data Mining*, pages 1135–1144.

Samie, F., Bauer, L., and Henkel, J. (2020). Efficient, lightweight, and robust: Tools and algorithms for securing iot applications. *IEEE Design & Test*, 37(2):93–105.

Saran, N. and Kesswani, N. (2023). A comparative study of supervised machine learning classifiers for intrusion detection in internet of things. *Procedia Computer Science*, 218:2049–2057.

Sarigiannidis, P., Karapistoli, E., and Stratogiannis, D. (2021). Ideal-iot: An efficient deep learning approach for anomaly detection in iot networks. *IEEE Internet of Things Journal*, 8(6):4711–4722.

Sethi, K., Kumar, R., Prajapati, N., and Bera, P. (2019). Impact of concept drift in iot network intrusion detection. In *2019 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, pages 1–6. IEEE.

Shahraki, A., Abbasi, M., and Taherkordi, A. (2021). Federated learning for iot security: Concepts, applications, and challenges. *IEEE Access*, 9:154087–154107.

Sinha, S. (2021). State of iot 2021: Number of connected iot devices growing 9% to 12.3 billion globally, cellular iot now surpassing 2 billion. *IoT Analytics: Market Insights for the Internet of Things*.

Venkatesan, R., Hsieh, M.-C., Sanghavi, P., and Baral, C. (2021). Robust adversarial learning for detecting malicious traffic in iot networks. *IEEE Transactions on Network and Service Management*, 18(3):3542–3557.

Verma, D., Das, B., and Kaushik, S. (2020). Toward explainable deep learning for iot security: A two-tier interpretable detection system. In *2020 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pages 73–80. IEEE.

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., and Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *IEEE access*, 7:41525–41550.

Wang, W., Zhu, M., Wang, J., Zeng, X., and Yang, Z. (2019). End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In *2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 43–48. IEEE.

Zhang, J., Chen, X., Xiang, Y., Zhou, W., and Wu, J. (2021). Attflow: An attention-based bidirectional lstm model for network traffic classification in iot. *IEEE Internet of Things Journal*, 8(15):12202–12214.

Zhao, Y., Chen, H., Duggins, R., and Su, J. (2020). Self-supervised learning for iot network anomaly detection. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 2215–2224. IEEE.

Zhou, D., Wang, Z., Bandyopadhyay, S., Donti, P., and Venayagamoorthy, G. K. (2020). Graph neural networks for anomaly detection in industrial internet of things. In *2020 International Conference on Internet of Things and Intelligence System (IoTaIS)*, pages 45–52. IEEE.

Appendix A

Appendix A: presentation slides

Coventry University

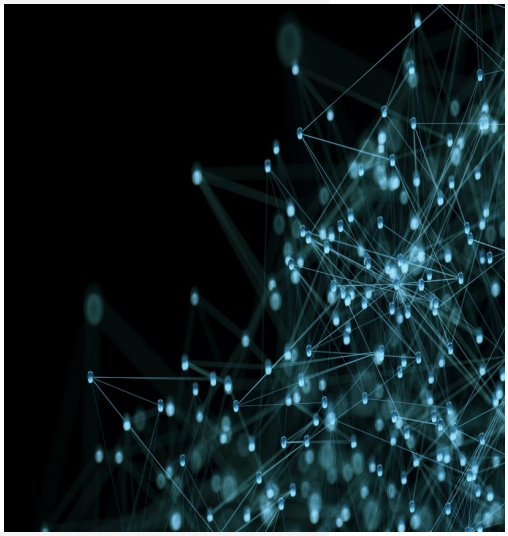
LEVERAGING MACHINE LEARNING FOR MALWARE DETECTION

Name: Ekwunife Blessing Ifunanya

Supervisor: Dr. Kabiru Mohammed

INTRODUCTION

- IoT devices are increasingly targeted by malware due to their proliferation and often weak security
- Traditional signature-based approaches are inadequate for detecting evolving threats and zero-day attacks
- ML and deep learning offer promising approaches for identifying malicious network traffic through pattern recognition
- Effective malware detection systems can significantly improve IoT ecosystem security and prevent widespread compromise



RESEARCH OBJECTIVES

Objectives

1

Analyze network traffic patterns associated with IoT malware infections

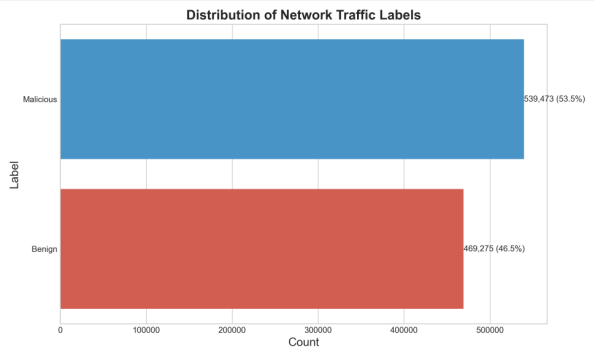
2

Identify distinctive features that differentiate benign from malicious traffic

3

3. Develop and evaluate machine learning models for automated malware detection using supervised model and neural network approaches

DATASET



- CTU-IoT-Malware dataset from the Czech Technical University. These labels were painstakingly created at the Stratosphere labs using malware capture analysis.
- The dataset was collected using network monitoring equipment that recorded alltraffic flows between the monitored devices and external networks
- Over 1 million labeled network connections from IoT devices. But we are using a variant of it due to computational resources
- 12 datasets was downloaded from the Kaggle version to use for the analysis Malware Detection in Network Traffic Data
- 53.5% malicious, 46.5% benign traffic
- 23 original network flow features like
  - Connection metadata (timestamps, protocols)
  - Traffic volume metrics (bytes, packets)
  - Connection states and durations

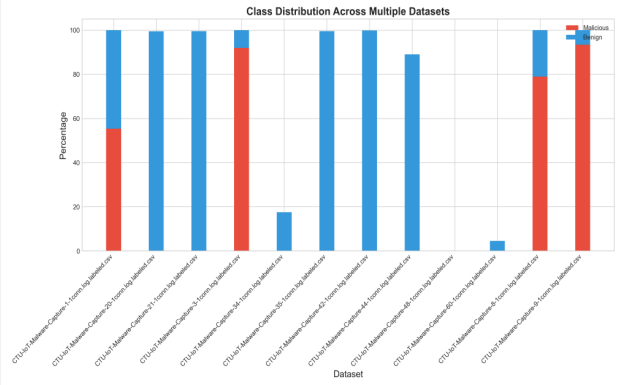
DATA PREPROCESSING

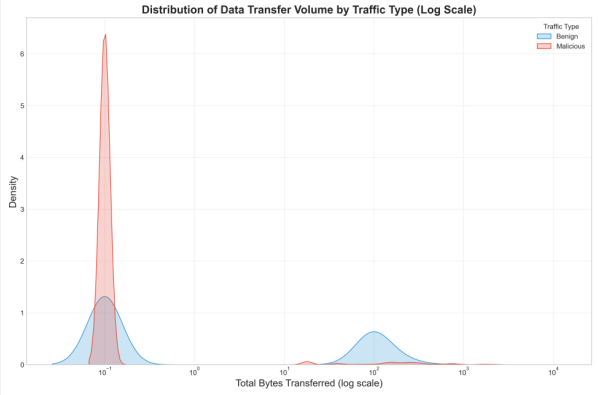
Missing Value Handling

- Duration: Conditional median imputation based on protocol and state
- Over 79% of records had missing values in certain columns, they were dropped
- Bytes/packets: Replaced with zeros (representing no data transfer)

Outlier Treatment

- Applied IQR method and log transformation for skewed distributions
- Standardized time stamps, categorical variables, and labels
- Applied SMOTE for addressing moderate class imbalance





FEATURE ENGINEERING

Temporal Features:

- Hour of day, day of week
- Connection density in time windows

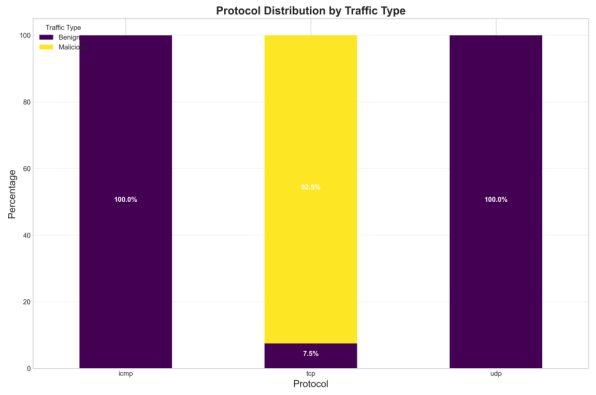
Traffic Volume Features:

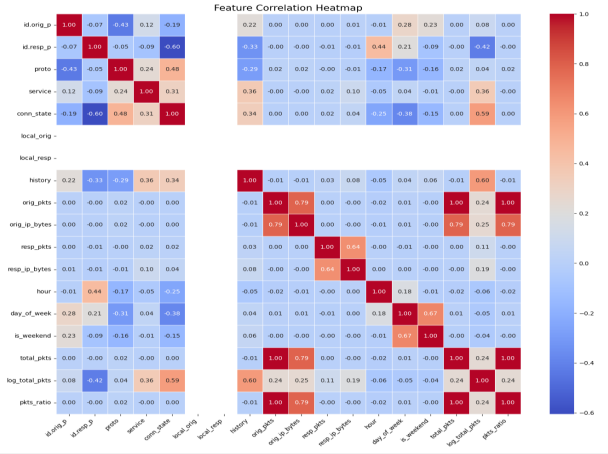
- Total bytes/packets
- Bytes per packet ratios
- Traffic direction ratios

Behavioral Indicators:

- Connection failure flags
- Data transfer indicators
- Port scanning detection metrics

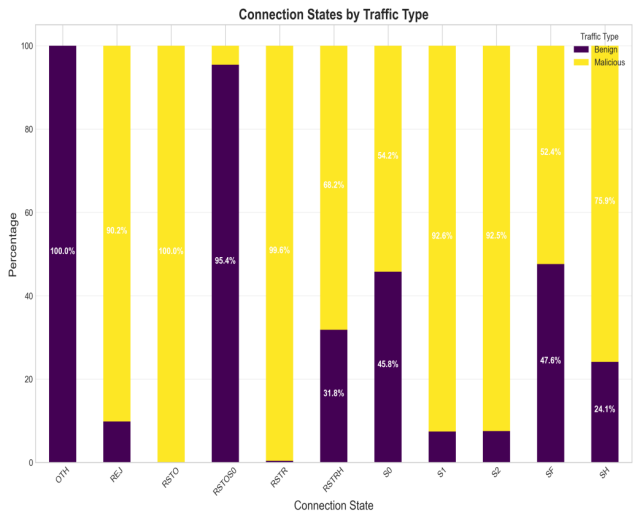
This features were derived from domain knowledge using original features from the dataset

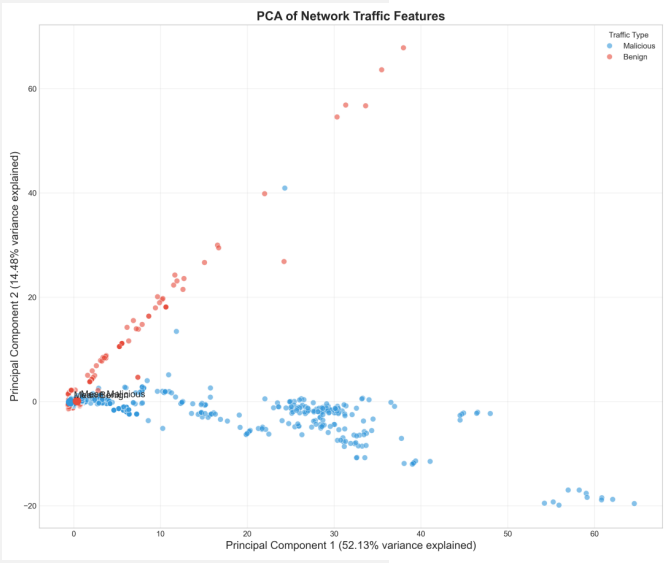




EXPLORATORY DATA ANALYSIS

- TCP dominated malicious traffic of 72.5%
- S0 state represented 54.2% of malicious connections which can be seen in the image below
- Malicious traffic showed concentrated bursts
- Malicious connections typically transferred minimal data
- Clear separation between classes in reduced dimensionality space





MACHINE LEARNING APPROACH

Model And Framework

Supervised Models Implemented:

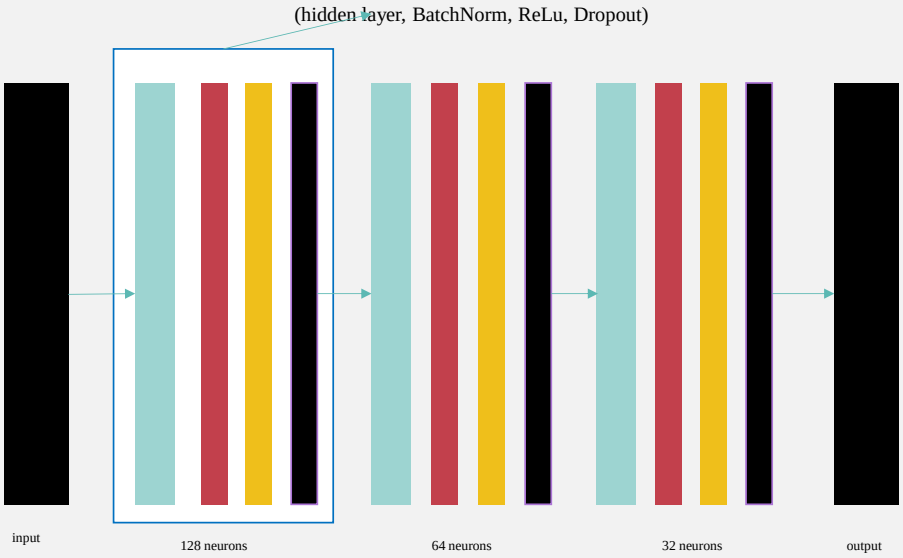
- Random Forest - Ensemble of decision trees, the decision is by majority voting
- SupportVector Machine (SVM) - Maximum margin classifier, which uses RBF kernel
- XGBoost - Gradient boosted trees, sequential correction

Evaluation Framework:

- Data split: 70% train, 15% validation, 15% test
- 5-fold cross-validation
- Hyperparameter optimization via RandomizedSearch CV
- Accuracy, precision, recall, F1 and confusion matrix were used to determine the model's performance

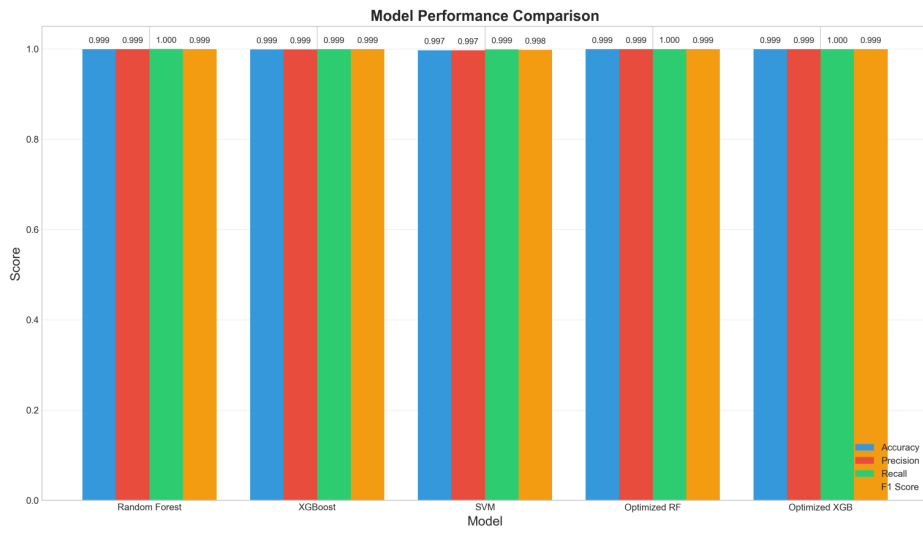
NEURAL NETWORK ARCHITECTURE

Deep learning model



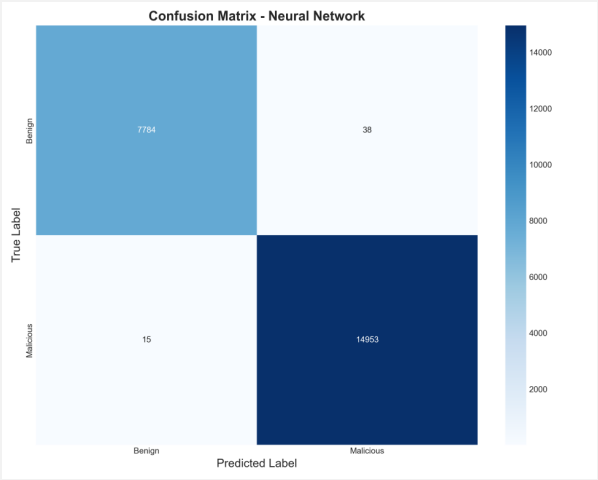
MACHINE LEARNING RESULTS

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	99.21%	99.15%	99.23%	99.19%
SVM	98.73%	98.45%	98.91%	98.68%
XGBoost	99.17%	99.12%	99.18%	99.15%
Optimized Random Forest	**99.96%**	**99.92%**	**100.00%**	**99.96%**
Optimized XGBoost	99.89%	99.85%	99.90%	99.87%
Neural Network	**99.77%**	**99.75%**	**99.90%**	**99.82%**



DEEP LEARNING RESULTS

- Achieved 99.77% accuracy and 99.90% recall
- rapid convergence with validation accuracy
- Greater than 99% within few epochs
- Performance comparable to best traditional models
- Slight trade-off between accuracy and computational requirements



INSIGHT

Results and Insights

Supervised Model Strengths:

- Optimized Random Forest achieved perfect recall(100%)
- Feature importance provides interpretability
- Lower computational requirements
- Less sensitive to hyperparameter tuning

Deep learning Advantages:

- Consistent performance across metrics
- High recall (99.90%)
- Inherent confidence measures
- Potential for scaling to more complex scenarios

Key Performance Insights:

- All models achieved greater 98.7% accuracy
- Statistical analysis showed significant difference between baseline and optimized models
- Optimized Random Forest slightly outperformed deep learning model
- The deep learning model showed strong performance with minimal feature engineering

REFERENCES

References

1. Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2023). Network traffic anomaly detection and prevention: concepts, techniques,and tools. Springer Nature.
2. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD Conference,785-794.
3. Garcia, S., Grill, M., Stiborek, J., & Zunino, A. (2019). An empirical comparison of botnet detection methods. Computers & Security,45, 100-123.
4. Stratosphere Laboratory. A labeled dataset with malicious and benign IoT network traffic. January 22th. Agustin Parmisano, Sebastian Garcia, Maria Jose Erquiaga. https://www.stratosphereips.org/datasets-iot23
5. 7.Sultana, N., Chilamkurti, N., Peng, W., & Alhadad, R. (2022). Survey on IoT security: Challenges and solution using machine learning, blockchain and post-quantum cryptography. Internet of Things, 100508.

THANK YOU

Appendix B

Ethics Approval Certificate

Leveraging Machine Learning for Advanced Malware Detection: P184895

Coventry University

Certificate of Ethical Approval

Applicant:	Blessing Ekwunife
Project Title:	Leveraging Machine Learning for Advanced Malware Detection:

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

Date of approval:	14 Feb 2025
Project Reference Number:	P184895

Blessing Ekwunife (7030CEM) Page 1 14 Feb 2025

Figure B.1: *Certificate of Ethical Approval*

project github repo link: https://github.com/BlessingEI/Msc.Dissertation-Blessing-

Ekwunife