



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

**USING AI/ML FOR NETWORK THREAT DETECTION
(TRAFFIC LOGS)**

By
Name: Abay Singh
Student Id: 2428686

Supervisor: Dr. Amit Naik

**Project submitted as part of the requirements for the award of MSc
Cybersecurity and Digital Forensics**

University of Wales Trinity Saint David (London)

February 2026

Declaration of Originality

I Abay Singh, confirm that this project is my own work. All sources I have used are properly cited and listed in the references. This report is a record of the work I completed, and it meets the required word limit.

Signed: Abay Singh

Date: 02-02-2026

Form of Consent

I Abay Singh, give my permission for this project, submitted for the award of the MSc Cybersecurity and Digital Forensics, to be available through the University Library for academic use. This may include inter-library loan and photocopying, subject to copyright law. I also agree that the project title and abstract may be shared with external organisations for academic purposes.

Signed: Abay Singh

Date: 02-02-2026

Acknowledgement

I would like to express my sincere gratitude to all those who contributed to the successful completion of this project. First and foremost, I would like to thank my supervisor for their invaluable guidance, constructive feedback, and continuous support throughout the research process. Their expertise and encouragement played a crucial role in shaping the direction and quality of this work.

I am also grateful to the academic staff of the department for providing a strong foundation in cybersecurity and machine learning, which enabled me to undertake this research with confidence.

Finally, I would like to thank my family and friends for their constant motivation, patience, and moral support throughout my academic journey. Their encouragement has been instrumental in helping me complete this project successfully.

Abstract

The rapid growth and complexity of modern network infrastructures have significantly increased exposure to cyber threats, rendering traditional rule-based intrusion detection systems inadequate. This study investigates the effectiveness of Artificial Intelligence and Machine Learning (AI/ML) techniques for detecting network threats through traffic log analysis. A quantitative, experimental approach was adopted using a publicly available intrusion detection dataset. Three supervised machine learning models—Logistic Regression, Decision Tree, and Random Forest—were developed, trained, and evaluated using standard performance metrics, including accuracy, precision, recall, F1-score, and confusion matrix analysis. The experimental results demonstrate that ensemble-based learning significantly outperforms linear and single-tree models. The Random Forest classifier achieved the highest performance, with accuracy exceeding 99.7%, very high recall, and a minimal false positive rate, indicating strong generalisation and detection reliability. Feature importance analysis revealed that traffic volume, connection frequency, service type, protocol usage, and temporal features were the most influential indicators of malicious behaviour. Overall, the findings confirm that AI/ML-based approaches provide an effective, scalable, and reliable solution for modern network threat detection.

Table of Contents

Acknowledgement	0
Abstract	2
Table of Contents	3
List of Figures	6
Chapter 1: Introduction	7
1.1 Research Background	7
1.2 Problem Statement	8
1.3 Aim and Objectives	9
1.4 Research Questions	9
1.5 Rationale	10
1.6 Report Structure	10
Chapter 2: Literature Review	12
2.1 Introduction	12
2.2 AI/ML Background and its Development in Network Threat Detection	12
2.3 Deep Learning Models of Traffic Analysis and Anomaly Detection	14
2.4 Intrusion Detection based on Machine Learning and Hybrid Architectures	16
2.5 Intelligent Frames, Explainable AI, SDN Security, and Emerging Frames	18
2.6 Data, Evaluation Measures, and Implementation Problems	20
2.7 Research Gaps	22
Chapter 3: Research Methodology	23
3.1 Introduction	23
3.2 Research Design	23
3.3 Research Method	24
3.4 Dataset Description and Preprocessing	24

3.4.1 Dataset Source and Characteristics	25
3.4.2 Data Quality Assessment	25
3.4.3 Data Transformation and Encoding	26
3.4.4 Feature Selection Rationale	26
3.5 Model Development and Architecture.....	26
3.5.1 Baseline Model: Logistic Regression	27
3.5.2 Decision Tree Classifier.....	27
3.5.3 Random Forest Classifier.....	28
3.5.4 Model Assumptions and Hyperparameters	28
3.6 Training Procedure.....	28
3.7 Evaluation Techniques and Criteria.....	29
3.7.1 Performance Metrics	29
3.7.2 Justification for Metric Choice	30
3.7.3 Model Comparison Strategy	30
3.8 Tools and Technologies Utilised	30
3.9 Ethical Considerations	31
Chapter 4: Result and Discussion	32
4.1 Introduction.....	32
4.2 Experimental Setup Summary	33
Dataset Overview	33
Model Configuration.....	34
Evaluation Environment	34
4.3 Descriptive Statistics and Data Distribution	34
Class Distribution Analysis	34
Distribution of protocol type.....	35

Class Distribution.....	36
Plot distribution.....	37
Plot heatmap.....	38
Visualise pairwise relationships.....	39
Boxplots for numeric features.....	40
Violin plots by class.....	41
Correlation on numeric features.....	42
4.4 Model Performance Results.....	43
4.5 Confusion Matrix and Error Analysis.....	48
Logistic Regression.....	48
Decision Tree.....	49
Random Forest.....	50
4.6 Feature Importance and Interpretability.....	50
Important Features Identified.....	50
Comparison Across Models.....	51
Security Interpretation	51
Practical Relevance	51
4.7 Practical Implications for Network Security.....	51
4.8 Limitations of the Experimental Results.....	52
4.9 Chapter Summary	52
Chapter 5: Conclusion and Recommendations	53
5.1 Conclusion	53
5.2 Recommendations.....	55
References.....	57
Appendix.....	65

List of Figures

Figure 1: Dataset Description	33
Figure 2: protocol type.....	35
Figure 3: Normal vs Attack.....	36
Figure 4: Distribution of src bytes	37
Figure 5: Correlation Heatmap of Numeric Features	38
Figure 6: Pairplot	39
Figure 7: group by class.....	40
Figure 8: Violin plot.....	41
Figure 9: Cluster heatmap	42
Figure 10: Logistic Regression	44
Figure 11: Decision Tree	45
Figure 12: Random Forest	46
Figure 13: Performance comparison	47
Figure 14: Logistic Regression Confusion Matrix.....	48
Figure 15: Decision Tree Confusion Matrix.....	49
Figure 16: Random Forest Confusion Matrix.....	50

Chapter 1: Introduction

1.1 Research Background

The contemporary digital networks are producing tremendous traffic at the rate of a second, and organisations are becoming more dependent on the system of interconnected systems to facilitate business operation, communication and delivery of services. With the increasing volumes and complexity of network infrastructures, they become more vulnerable to cyber threats, including malware, DDoS attacks, data exfiltration attacks, botnet activity, and insider abuse [1]. Historically, intrusion detection systems (IDS) based on signatures and firewall based on rules have been used to detect suspicious patterns when it comes to network security. These systems are good against known attacks, but have difficulty facing new threats, zero-day attacks, and new attack behaviours that are not reflected in existing signature sets [2]. Moreover, the high-speed diversification of network protocols and encrypted traffic severely restricts the usefulness of manual/static rule-based analysis.

Artificial intelligence (AI) and machine learning (ML) methods emerged as the new focus of network security research to cope with these changing issues [22]. The intrusion detection systems based on AI/ML use statistical modelling, pattern recognition, and predictive analytics to innovate behavioural patterns on the basis of historical logs of the network. In comparison to the classical IDS, machine learning models can change dynamically, detect anomalies, and alert to attacks based on the changes observed in norms. As the volumes of data increase exponentially, AI/ML systems can be scaled as well, allowing the automatic analysis of millions of entries in network logs that human analysts could not review in practice.

The argument in favour of AI and ML is also supported by the growing complexity of cyberattacks. There are constant updates in methods which attackers use to outwit the defences; they resort to automated scripts, polymorphic malware and multi-vector attack patterns which attackers use to compromise weaknesses in the existing static defences [32]. Machine learning methods, including supervised classification, unsupervised anomaly detection, clustering algorithms, and deep learning networks, allow security systems to discover implicit relationships, identify some rare or subtle anomalies, and predict potential threats before they escalate. Such a change from reactive to proactive detection is vital to securing the current organisational networks.

One of the most important elements of facilitating AI-based security is the logs of network traffic. The router, firewall, intrusion detection, and network monitoring logs have critical data about the packet flow, source-destination relationships, protocol, and volume of the traffic. The study of these logs enables the creation of datasets by the researcher that can be used to identify behavioural patterns of both positive and negative activity [12]. The recent developments in the sphere of big data technologies, distributed computing, and cloud storage allow making large-scale log collection and processing increasingly a possibility, which provides the environment within which machine learning models could be trained very accurately and reliably.

Moreover, the speed improvement in cloud computing, Internet of Things (IoT) devices, and remote working systems has increased the size of the attack surface of organisations. A single device and a single user connection add to huge amounts of traffic data, which would be nearly impossible to monitor manually [21]. Threat detection based on AI/ML is perfectly aligned with these new realities because it facilitates intelligent automation, reduces false positives, and provides faster response time. This field is currently expanding, with researchers recording encouraging achievements in fields like deep neural networks to classify traffic, random forest-based intrusion detection, autoencoders to detect anomalies, and hybrid ML-IDS systems, which combine various methods to create a more accurate outcome.

The quality and labelled datasets required by many machine learning systems are hard to get because of privacy limitations or the lack of real-life attack instances. Also, ML models are attacked by adversarial attacks, introducing new security threats, which raises questions about the vulnerability of models. Another critical concern is the interpretability of the outcomes of machine learning, especially in the context of cybersecurity analysts who need a clear explanation of results when reacting to identified threats [11]. AI and ML have some transformative potential in achieving better network protection and detecting threats based on traffic logs. The presented study examines how AI/ML-based methods could be used to examine the network traffic efficiently and present a secure and scalable way to detect malicious behaviour. The research is expected to help advance the field of intelligent, automated defences that can combat the issues of contemporary cyber challenges by studying a variety of working ML models and assessing their capabilities.

1.2 Problem Statement

In constantly expanding and sophisticated network-based settings, organisations are finding it more challenging to determine the existence of cyber threats. The classical rule-based or signature-based detection systems have a low capacity of detecting the emergent, unknown, or advanced attacks. In the meantime, network traffic logs have increased exponentially, rendering a manual analysis impractical and introducing delays that amplify the risk of uncovering a malicious activity [16]. With attackers evolving increasingly dynamic and resistant methods, the failure of the currently available security tools to expose abnormalities as they occur is very critical to the operation and financial losses. Though AI/ML approaches demonstrate a high potential with regard to enhancing detection accuracy, scalability, and automation, systematic research testing these approaches on traffic log data and assessing their efficiency remains to be performed. As such, the key issue that this research seeks to resolve is the inefficiency, lack of intelligence, as well as flexibility of threat detection systems that are able to learn about network traffic patterns and are able to identify known and unknown attacks with a high degree of accuracy.

1.3 Aim and Objectives

Aim

The aim of this research is to develop and evaluate AI/ML-based approaches for detecting network threats using traffic log analysis.

Objectives

- To analyse existing AI and ML techniques used for network threat detection and highlight current limitations.
- To design and implement machine learning models capable of identifying malicious network activity.
- To evaluate the performance of selected ML algorithms using appropriate metrics such as accuracy, precision, recall, and F1-score.
- To compare traditional detection methods with ML-driven approaches to highlight improvements in effectiveness.

1.4 Research Questions

- How can AI and machine learning techniques improve the detection of network threats compared to traditional security tools?
- What features within network traffic logs contribute most significantly to accurate threat identification?

1.5 Rationale

The rising complexity of computer threats has pointed out the weakness of the traditional intrusion detection systems, which mainly depend on a set of rules or signatures. With the ongoing growth in the scope of networks, security personnel are already overwhelmed with increasing volumes of traffic logs, and monitoring systems based on manual operations prove ineffective and prone to errors [15]. The use of AI and ML as a means of providing solutions to this issue is promising because it is not only capable of automated, adaptive, and data-driven threat detection but also a vital approach that modern organisations want to solidify their cybersecurity position [18]. The study is important as it examines the practical implementation of machine learning on the process of analysing traffic logs and provides insights into the severity of intelligent models to detect anomalies and attacks more efficiently than traditional methods.

Furthermore, cyberattacks are increasingly dynamic and minimal and can be successfully defended. ML-based models can potentially acquire behavioural patterns and recognise subtle changes that can be the indicators of a threat, therefore assisting in the prevention of active security controls. The research also helps to solve the problem of model performance assessment, providing a comparative study of various techniques in ML [10]. The study offers helpful guidelines useful to organisations that are considering the adoption of AI-driven security solutions, given its focus on practical implications and the quality of the datasets to be used, feature selection, and interpretability. In general, this study is justified by the fact that it will help strengthen network resilience, shorten the time required to detect interruptions, and develop more intelligent and responsive cybersecurity mechanisms.

1.6 Report Structure

This report is divided into five chapters that tackle major elements of the research. The **Introduction** includes the background on the applications of AI/ML in network threat detection,

provides a description of the problem, states the aim and objectives, research questions, and rationale of the study.

The Literature Review critically gives evaluation of the existing literature on network-based intrusion detection, machine learning methods, and traffic logs analysis. It analyses the advantages and disadvantages of various strategies, research gaps, and technological tendencies that guide the purpose of this research.

Research Methodology outlines the operational framework to be followed towards the realisation of the study objectives. It contains information on the research design, dataset selection, data preprocessing, feature engineering, machine learning model development, and performance evaluation methods. Concerns of ethics regarding data usage and research on cybersecurity are also addressed.

The **Results and Discussion** chapter provides the findings of the machine learning experiments are provided and the effectiveness of different models is compared with regard to the existing evaluation metrics. It explains the results within the framework of the available literature and explains that the findings have in actual network security.

Conclusion and Future Work will attempt to generalise the main contributions of the study, comment on its limitations, and present the future research directions, including the development of better deep learning structures, real-time system development, and the scope of adoption of AI/ML models into functioning security systems.

Chapter 2: Literature Review

2.1 Introduction

The growing complexity, magnitude, and variety of cyber threats have rendered the conventional safety measures inadequate in the protection of contemporary digital infrastructures. Traditional signature-based methods and rule-based methods tend to be ineffective in identifying new, emerging, or fast-evolving attacks, thus posing an immediate need for intelligence methods that would be able to acquire behavioural patterns based on network traffic. Abbasi et al. [1] state that the drastic rise in network data has placed artificial intelligence (AI) and machine learning (ML) as critical instruments in monitoring network traffic and cyber defence. In addition to providing scalability, AI-based systems can be used in proactive, adaptive detection of threats, which is essential in dynamic networks like cloud computing, IoT ecosystems, and software-defined networks (SDNs).

The existing literature shows a variety of different and occasionally competing results concerning the effectiveness, interpretability, and practicality of AI-based methods. Although it is true that deep learning (DL) models represent a superior approach to grasp the complex traffic behaviours, researchers such as Mohamed [24] note the issues related to the transparency and computational overhead. Conversely, a number of studies support the use of machine learning approaches as more practical, interpretable, and deployable, but may not be able to represent extremely complex attacks.

This chapter critically summarises studies on five themes: AI/ML foundations, deep learning-based traffic analysis, ML and hybrid structures, explainable and SDN-based structures, and deployment practical challenges. The chapter illustrates points of convergence, divergence, and methodological imperfections or weaknesses through comparing methodologies, assumptions, limitations, and empirical results in and across existing scholarship. The review eventually finds significant gaps concerning dataset realism, model scalability, explainability, adversarial robustness, and interaction with operational systems, hence establishing a basis for the necessity of a better AI-based threat presence on the basis of network traffic logs.

2.2 AI/ML Background and its Development in Network Threat Detection

The development of AI and ML in the area of cybersecurity reveals an evident shift towards the inertial, signature-oriented processes to lively, data-based analytical models. Gohel and Popat [13] note that intrusion detection systems (IDS), which were initially developed as early systems, were largely based on predefined rules; hence, not effective in detecting zero-day and polymorphic attacks. The advent of AI/ML has hence played a significant part by turning cybersecurity from a reactive form to predictive security intelligence. The study by Abbasi et al. [1] emphasises that the swift increase in the complexity of network traffic leads to the need to adopt automated analysis methods that can detect subtle anomalies that human analysts can fail to detect. Their results support the rationale of scalable and adaptive threat detection solutions.

However, different researchers recognise the radically changing role of AI/ML; they disagree with its maturity description. Indicatively, Mohamed [23] highlights the growing complexity of AI methods in cybersecurity, yet reports that their practical implementation is minimal because of issues related to the quality of data, transparency in algorithms, and integration costs. In addition to this, Mohamed [24] claims that in spite of significant development of machine learning models in the real world, their applications are usually weak, unintelligible, and fail to counter adversarial manipulation attacks. This demonstrates an enduring lack of connection between laboratory-scale experimentation and enterprise-scale applicability.

The underlying literature also shows that there is an inconsistency on whether machine learning or deep learning offers the best threat detection. Katiyar et al. [17] have noted that ML techniques provide operational effectiveness, reduced computational needs, and reduced response time. They state that the ML is appropriate for those organisations that need lightweight tools that can be understood by them. Conversely, Alam et al. [4] address the wider implementation of AI/ML in the domain of traffic engineering and propose that deep learning models provide more detailed information when working with high-dimensional and intricate traffic patterns. This contrast implies that the operation conditions and limitations of resources are far important in model selection.

In several research sources, the importance of SDN and virtualised networks and how they rely on AI-based threat detection is emphasised. Ahmed et al. [2], in particular, emphasise the benefits of centralisation brought by the SDN control plane, extending visibility and enabling AI-based monitoring. Their discussion provides evidence of the claim that the use of AI in cybersecurity is

becoming more closely connected to the new paradigms of networking. They also warn that SDN opens new attack points, like controller-based exploits. Although AI can be used well with programmable networks, it also requires robust protections and resilient structures.

The study by Abbasi et al., [1] observes that AI-based research on intrusion detection relies on benchmark datasets, which do not necessarily represent the diversity of traffic in the real world. Similar criticism can be found in Azab et al., [6], which elaborates that encrypted or IoT-based traffic is not well represented in datasets like CICIDS and NSL-KDD, which results in poor generalization of the models. Dhakad et al., [9] also report that uneven dataset selection in different studies further complicates the cross-comparative assessments, as it is not easy to state what exact approaches are superior.

Indicatively, Mohamed [24] emphasises that using AI to perform security tasks presents dangers associated with automation bias, the lack of interpretability, and the possibility of adversaries attempting to exploit the vulnerabilities in the models. Paracha et al. [28] provide a conceptual review that supports the potential of AI but does not argue it out with rigorous empirical testing, which is a bigger problem: the literature still includes speculative discussions, not operationalized yet. Nevertheless, existing research supports the idea that the AI/ML methods considerably surpass the traditional IDS regarding their flexibility, adaptability, and detection accuracy. The literature, however, shows that the building up of progress is not even.

The research by Katiyar et al. [17] proposes a positive outlook that focuses on the improvement of performance and practical deployment barriers. This opposition highlights the tension that still exists in the field: AI/ML is generally understood as the key to the safety of networks in the future, but its implementation in practice is still hampered by technical, operational, and organisational issues. The literature base has defined the conceptual relevance of AI/ML in contemporary intrusion detection but reveals the significant issues regarding scale, representativeness of datasets, transparency of models, and system integration. These underlying constraints also serve to support the existence of more research that would focus on operationalising AI-based threat detection within a real-world network setting.

2.3 Deep Learning Models of Traffic Analysis and Anomaly Detection

The ability to model high-dimensional and complex patterns in the traffic data has led to deep learning (DL) being one of the most visible approaches in network threat detection. Fotiadou et al. [12] note that Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks are regarded as better models in this regard, as they can automatically identify hierarchical features in raw traffic flows. Contrary to the conventional machine learning methods that fully rely on the use of engineered features, the DL models obtain discriminative representations directly during training, and thus, can detect subtle anomalies that may not be efficiently represented as features by engineers. Nevertheless, this automated learning process is associated with substantial computational costs, and Fotiadou et al. [12] admit that the training time being large and the complexity in tuning hyperparameters pose challenges to real-time implementation.

One of the key bodies of literature centres around applications of hybrid deep learning models. Bakhshi and Ghita [7] use a hybrid design that involves autoencoders and recurrent neural networks (RNNs) to identify irregularities in encrypted traffic. Their results point out that the use of DL is especially efficient in encrypted settings, as it is not based on the examination of the content, but it consists on behavioural and statistical traits. Conversely, they warn that encrypted traffic does not provide methods of interpretation based on the granularity of individual packets, so it is less interpretable. In addition to this, hybrid architectures can cause their deployment to be more resource-intensive.

Yagual et al. [33] introduce an even more sophisticated hybrid model that uses Enhanced Feature Manipulation Systems (EFMS), KMeans clustering, and CNN-GRU networks. Their findings show that they have higher accuracy and fewer false positives and indicate that feature refinement boosts the performance of DL considerably. Nevertheless, the complexity of the architecture of their model gives us some issues regarding the scalability and practicality in both practical network environments. Edozie et al. [11] suggest that telecom networks need to use DL models that can handle high-volume streams near real-time, where resource requirements are high, and a hybrid model might be unable to perform.

The other significant part of the DL literature touches on the classification of traffic with the help of the DL-based feature extraction. According to Abbasi et al. [1], deep learning is accurate in classifying traffic categories, behaviours, and anomalies. Nevertheless, they emphasise that most

DL research uses popular benchmark datasets, which might not be indicative of novice attack behaviours or reality. According to Azab et al. [6], a large share of accessible public datasets does not contain encrypted and IoT-based traffic and, therefore, their use inhibits the generalisability of DL models trained with their help. Dhakad et al. [9] also identify a lack of consistency in preprocessing methods and differences in evaluation metrics as a factor in the intolerance of cross-study comparisons in the DL research.

Landauer et al. [20] present an extensive review demonstrating that DL architectures, especially autoencoders and transformer-style models, are becoming more frequently used to detect log anomalies. They claim that the DL works fairly well to model the sequences of the temporal events, but also state that the log data usually includes noise, non-measured values, and high-dimensional forms. Conversely, they caution that the DL models are very sensitive to concept drift, that is, their performance degrades as the network behaviour evolves with time.

Mohamed [24] emphasises that DL models are susceptible to adversarial attacks where they are spoiled by small, well-designed molecules that can fool classifiers despite their high detection, even though. Such a weakness fails to instill confidence in the application of DL models in mission-critical cyber environments. According to Bakhshi and Ghita [7], encrypted environments pose even more significant risks since the DL models cannot count on payload inspection and, therefore, have to deduce malicious behaviour with incomplete information.

Regardless of these constraints, DL will continue to grow into a primary cybersecurity method. Its strength is that it can acquire complex behavioural signatures and detect anomalies which are missed in its traditional models. Nevertheless, the literature points continually to the following problems: cost of computation, inaccessibility of data, inability to interpret results, drift vulnerability, and the inability to move laboratory experiments to operational deployments. Thus, despite the fact that DL is a promising direction in network threat detection, the existing evidence indicates that it needs to be combined with effective, explainable and scalable frameworks to become widely effective.

2.4 Intrusion Detection based on Machine Learning and Hybrid Architectures

Machine learning (ML) as a concept has been a mainstay of intrusion detection since it is relatively simple, efficient, and interpretable. K.A. Dhanya et al. [14] state that classical ML algorithms,

such as Random Forests, Support Vector Machines (SVMs), Naïve Bayes, and gradient boosting, remain crucial in the detection of known and unknown network attacks. Their research paper shows that with high-quality engineered characteristics, ML models can be developed to provide competitive performance with a much smaller number of computational resources as compared to deep learning architectures. Their results also, however, show that the effectiveness of ML is very dependent on the quality of manually-engineered features, which is a labour-intensive and domain-specific undertaking.

The paper by Ahmed U. et al. [3] discusses a hybrid signature-based intrusion detection system, which is enhanced with fuzzy clustering and deep learning. They find that with the incorporation of fuzzy clustering, they achieve a higher detection rate due to their ability to detect latent structures within the network data not easily detected by their traditional signature-based approaches. Conversely, they recognise that signature-based systems are still insufficient when it comes to warning against zero-day attacks, meaning that hybrid models still have the weaknesses that characterize conventional methods.

Schummer et al. [31] focus on the value of feature selection and dimensionality reduction to enhance the efficacy of ML. Their ML-based anomaly detection framework illustrates that the irrelevant features reduction not only increases the accuracy but also enhances the computational efficiency. They observe a trade-off whereby an over-dimensionality reduction can remove small but significant behavioural cues. Salam [30] also presents another critical point by using ML methods on log anomaly detection. Based on their results, the ML models are superior to the more complicated DL architecture when the datasets are small or sparse, and the ML is highly efficient in conditions of small data availability. Conversely, Salam [30] also observes that ML models might not be able to learn non-linear and high-order interactions in network logs, which DL performs well.

Omopintemi et al. [27] compare the use of ML-based malware detection in network traffic. Their findings indicate high performance of monitored ML algorithms, in cases when feature order is flow level, including the packet size, length, and frequency of connections. They point to a significant shortcoming, namely that malware changes fast, and so it is necessary to train the ML models frequently. Loss of training data may lead to ineffective and old models. Muheidat et al. [25] analyse automated cybersecurity improvements based on the ML models. The results of their

research support the idea that ML pipelines may effectively be used in the case of well-designed feature sets. They conclude, however, that ML systems are susceptible to overfitting in particular when the models are trained on imbalanced samples, as is the case in network traffic analysis, where benign traffic predominates. Azab et al. [6] point out that an imbalance in datasets largely distorts the measures of altitude to ML, and that it usually seems to artificially inflate the measures of accuracy because malicious samples are few.

Dhakad et al. [9] note one methodological flaw present in most ML studies: the lack of preprocessing and evaluation schemes. Instead, they suggest that different dataset splitting, feature selections, and reporting of metrics make it hard to compare any two approaches to ML. Concerning the accuracy measure, some studies indicate that the primary measure of accuracy was reported, but in other cases, it is precision-recall or F1-score. Conversely, Bakhshi and Ghita [7] caution that accuracy alone is a fallacy, particularly when the aim is to detect an anomaly with benign samples of the data being larger than the malicious ones.

A hybrid research design is one more line of research, in which the authors combine both ML and DL to attain the advantages of the two paradigms. Yagual et al. [33] prove that using clustering methods along with DL models ensures even greater feature vision and better classification of anomalies. Although their treatment is high-performing, they are also aware of how the level of architectural complexity and computer resource demands is being made heavier. This shows that ML itself might not be adequate in the current cybersecurity environment, but it is indeed vital in combination with hybrid or layered threat detection measures.

2.5 Intelligent Frames, Explainable AI, SDN Security, and Emerging Frames

The trend of investigating the application of artificial intelligence in network security has increased worries about the aspects of transparency, interpretability, and operational trust. Explainable AI (XAI) has thus become a major topic of concern. Nalinipriya et al. [26] indicate that the lack of interpretability in the deep learning models greatly restricts the trust of the analysts as well as the process of the incident response. They claim that in massively networked environments, especially those that feature high-speed and distributed systems, security analysts cannot afford to depend on black-box forecasts without getting access to the rationale of the observed strange behaviours. Their study proves that SHAP and LIME, as XAI algorithms, are capable of emphasizing influential factors and decision-making paths, but both of the mentioned methods frequently face

scalability issues when it comes to network traffic streams at a large scale. Conversely, Nalinipriya et al. [26] stress that explainability should neither limit detection accuracy, and there is a limitation on model transparency and model complexity.

The issue of explainability is especially severe in a setting like Software-Defined Networking (SDN), where flow control is automated and has direct influence on the network performance. According to Ahmed et al. [2], one of the most detailed accounts of AI/ML-based threat detection in SDN platforms can be found. They emphasise that the unprecedented visibility that SDN offers with its centralised controller makes the compatibility with ML and DL-based monitoring high. Nevertheless, they caution that programmability of SDN presents vulnerabilities that are new, including controller hijacking, malicious rule injection, and flooding of flow tables. Prasad et al. [29] argue that AI-enriched decision-making in the SDN-based multicast systems requires a very high standard of explainability because misclassification can cause unintended changes to the flow or service failures. Their results highlight the need to have interpretable waiters of AI systems into mission-critical network architectures as a way of both accuracy and operational transparency.

The new paradigms in AI-powered cybersecurity indicate a transition to unified, more flexible solutions as well. Zhao [34] suggests an AI-based network security threat detection and response mechanism that can dynamically modify the protection mechanisms. Their innovative method is not as validated empirically as possible; hence, the limitation is on its applicability in real life. On the same note, Paracha et al. [28] provide a conceptual overview of AI-driven threat detection but do not present experimental findings. This is not a rare active tendency in the literature, with multiple authors coming up with conceptual frameworks that leave implementation challenges unaddressed. The existence of such gaps implies that there is a gap between theoretical development and its real implementation.

The study by Bakhshi and Ghita [7] focuses on the topic of hybrid deep learning in anomaly detection of encrypted internet traffic, with the common understanding that there is a growing demand to develop security models that can maintain their functionality without having to inspect the payload. Their results show that AI and DL methods are suitable when it comes to identifying behavioural patterns in encrypted flows; however, they also admit that such models are difficult to interpret. In addition to the problems with interpretability, encrypted traffic discourages the richness of features, forcing researchers to work with statistical metadata instead of actually

looking at packet content. This downside has substantial effects on the creation of future XAI frameworks that would correspond to encrypted environments.

AI systems built on SDN have become popular as well because they enable opposition to centralised control and automation, as well as mitigation. Ahmed et al. [2] note that SDN is based on the future of intrusion detection by ML or DL-based systems that can dynamically track flow anomalies and implement countermeasures. Nevertheless, as they emphasise, such systems must be heavily hardened in terms of the security aspects because the opponents may use the AI-driven decision-making to control the streams of traffic.

Moreover, hybrid intelligence systems compatible with ML, DL, and XAI concepts are actively investigated. The study by Yaguals et al. [33] combines clustering with deep neural models and shows their accuracy, but does not talk extensively about transparency. On the other hand, Prasad et al. [29] are more explainable, but fail to match the high performance of the more complicated architecture. Such opposing results indicate that there is still no consensus in the literature on how to balance the three areas of performance, explainability, and operational feasibility.

2.6 Data, Evaluation Measures, and Implementation Problems

Although there are great improvements in AI-based network threat detection, there are still many challenges that limit its practical application. The major issue in the literature is the quality and the representativeness of the datasets. Azab et al. [6] state that the widely adopted datasets include NSL-KDD, CICIDS2017, and UNSW-NB15, but they are associated with out-of-date patterns of attacks, insufficient variability in protocols, and non-realistic traffic models. According to Abbasi et al. [1], the datasets of this type do not reflect current encrypted, IoT, cloud-native, and heterogeneous network environments. Such a discrepancy in experimental training data and traffic complexity in the real world has resulted in models that work well in laboratory environments but show very poor behaviour in the field. Instead, other researchers seek to address those problems by using synthetic data generation and augmentation methods, which may also lead to the introduction of artificial biases to models.

There are also significant discrepancies in evaluation measures. A lot of research is very dependent on precision, even though that would not help the most when dealing with imbalanced data sets where malicious traffic represents an extremely low percentage of the total traffic. Bakhshi and

Ghita [7] argue that accuracy, recollection, and F1-score are crucial factors to measure the effectiveness of anomaly-detecting systems, particularly within encrypted environments. Dhakad et al. [9] require cross-comparative analysis to be challenging, as evaluation methodologies have not consistently been used in different studies. The difference in preprocessing, feature selection, and data-splitting techniques also further contributes to this inconsistency.

Another problem that restricts the large-scale use of AI-based threat detection is the computational complexity. Deep learning models introduced by Yagual et al. [33] and Bakhshi and Ghita [7], may require a lot of GPU or TPU to run high-dimensional streams of data in real time. The research indicates that because of the huge volumes of traffic transmitted via telecom networks, inference must occur over almost instantaneous durations, posing a significant resource burden to the DL models unless specialised hardware is used.

Landauer et al. [20] caution that models identifying anomalies in the past deteriorate in the case of concept drift, particularly in environments based on logs. The reason they give is that it is necessary to constantly retrain and update the model, which is operationally costly. On the same note, Omopintemi et al. [27] note that malware also changes very fast and that ML models should be updated frequently to remain relevant. Muheidat et al. [25] state that integrating AI-powered systems with the current security structures is associated with major organisational and technical challenges. Leaked systems can have no APIs, visibility points, or storage capacity to accommodate analysis based on AI. Other than this, automated AI decisions would be incompatible with manual rule-based policies, which makes responding to an incident more difficult.

Mohamed [24] advises that adversarial attacks like data poisoning or evasion methods are capable of breaking ML/DL models to misclassify threats. This weakness defines new attack surfaces, which predetermines the model robustness as a critical, but frequently neglected, research focus. According to Bakhshi and Ghita [7], detection of encrypted traffic is based on metadata analysis, which can be problematic with regard to privacy when mishandled. Equally, Ahmed et al. [2] note that SDN environments, although rich in visibility, centralise sensitive information on traffic that also needs protection.

According to Dhakad et al. [9], most of the models are only experimented with in the laboratory, where there is a lack of knowledge on how they behave in operational networks, where behaviour

is unpredictable. Prasad et al. [29] state that even SDN-based AI-based structures, although being more consistent with real-world configurations, are still highly experimental. Overall, although AI/ML strategies have great potential, issues of datasets, benchmarks of evaluation, computational costs, model safety, and their operational subsumption pose a challenging obstacle to implementation. The solution to these issues is critical in the implementation of AI-driven threat detection in real life.

2.7 Research Gaps

In the literature review, some notable research gaps are identified that allow concluding that more robust, scalable, and operationally viable threat detection systems based on AI are required. Realistic and high-quality datasets are in short supply. Most of the research employs old or artificial test data that cannot reflect contemporary encrypted, IoT, virtualised, and cloud-native traffic patterns [1]. This highly restricts the external validity of AI models and prevents proper benchmarking. The reference material presents a poor adoption of explainability in performing well deep learning systems. Even though XAI is becoming popular [26], the majority of the studies that use DL models focus on accuracy but not interpretability. This loophole distrusted the analysts and has delayed its practical implementation in the mission-critical networks.

The issue of concept drift and model lifecycle management is not paid enough attention. Network behaviours are changing quickly, but little research has examined continuous learning, adaptive retraining, or drift-resilient architectures [20]. Mohamed [24] notes that a significant number of AI models are susceptible to malicious perturbations, and the majority of the research is not implemented in adversarial conditions where the robustness is evaluated. The study is not in-depth on real-time resource-efficient deployment strategies. Deep learning models are very high and still costly in terms of computation [33], and lightweight ML models might not be advanced enough to tackle advanced attacks [14]. It has a significant lack of operational validation. Numerous experiments are done with controlled datasets and experimental setups, providing little understanding of how the models will act in a live network of production under unpredictable usage and traffic fluctuations. These weaknesses demonstrate why an all-inclusive, understandable, adversarially sound, and practically implementable AI/ML architecture can be used to analyse real-world network traffic logs to describe the intent behind the ongoing study.

Chapter 3: Research Methodology

3.1 Introduction

This chapter describes the methodology applied in the creation and analysis of the AI/ML-based solutions to detect threats to the network based on traffic logs. The workflow of the study consists of empirical research with an experiment character that involves the preprocessing and transformation of network traffic data and its analysis by supervised machine learning models with the aim to determine the connection as either normal or malicious [35]. The research objectives are consistent with the methodology, as it provides the possibility of comparing the performance among the models, as well as determining which features have the strongest impact on predicting threats. It also directly addresses the research questions by performing systematic experimentation on an actual intrusion detection dataset, in which controlled preprocessing, model training, and validation yield measurable and comparable results. It can be stated that the objective methods of detection capability focus on accuracy, precision, recall, and F1-score to measure the quantitative performance [36]. The proposed experimental design based on AI/ML is suitable as cyber-attacks are dynamic and pattern-driven, and machine learning is capable of detecting threats adaptively and based on data, which is a more rigorous and reproducible analytical framework in comparison with the traditional rule-based systems.

3.2 Research Design

The paper is a quantitative, experimental research which is on supervised machine learning. The design is aimed at training the classification models with the labelled records of the network traffic, and comparing their performance for various evaluation metrics [37]. The study uses a binary classification intrusion detection framework since the purpose is to identify whether a particular network connection is representing normal or attack behaviour.

The classification-based IDS modelling approach is employed due to the fact that intrusion detection is a decision-based security task whereby the traffic connections need to be classified as malicious or benign. In this case, supervised learning is especially appropriate, because the dataset has labelled examples, which enable models to acquire discriminative behaviour patterns of familiar attack behaviour.

A validation dataset is also added to enable the use of an unbiased analysis of model performance and to avoid overfitting. The training data is divided into an 80-20 split in which the larger part is kept to learn the model and the other part is held as a validation sample. The process of comparative benchmarking is then carried out with the various models such as Logistic Regression, Decision Tree and Random Forest to establish which algorithm shows the most accurate detection performance. The design enhances the validity of findings because findings are drawn out of empirical comparison and not on the performance of individual models [38].

3.3 Research Method

The study is based on an empirical, model-driven approach that is based on the analysis of the network traffic logs. Instead of doing theoretical modelling, the study concentrates on empirical experimentation through the real intrusion detection data in order to monitor the behaviour of machine learning models [39].

It uses a supervised classification technique, in which every traffic example is modelled by a collection of network-level features and is labelled by a set of classes, i.e., normal behaviour or an attack. It can be observed that the relationships between these features and their corresponding labels are learned by training machine learning models to predictively classify unknown connections.

The study is based on a cross-model comparison approach, in which different algorithms are trained and tested on the same dataset and preprocessing pipeline. The approach can also be used to establish the strengths and weaknesses of various model families, including linear models, decision-tree-based learners and ensemble approaches. Comparative analysis can also be used to make sure that the findings are not biased towards a particular algorithmic design [40].

Major methodological factors include replicability and reproducibility. Experiments are performed on openly available datasets, have well-defined preprocessing processes, and use standard machine learning libraries. The workflow is programmed to prevent human mistakes, and the same seeds and parameters of the configuration are used in each experiment. This guarantees that the research or practitioner who comes after can replicate the same results by following the same methodology.

3.4 Dataset Description and Preprocessing

3.4.1 Dataset Source and Characteristics

The data employed in this research is found on Kaggle under the name of Network Intrusion Detection. It includes marked network traffic logs which model authentic intrusion detection cases. The data is also offered in training and testing files, which allows to conduct structured experiments and model validation.

The dataset contains several groups of features of various dimensions of network activity such as:

- **Basic traffic features** – Attributes such as protocol type, service, and flag, which describe fundamental connection properties.
- **Content-based features** – Indicators derived from packet payload behaviour and connection interactions.
- **Time-based traffic features** – Statistical attributes aggregated over time windows to capture repeated or burst-based behaviour.
- **Host-based traffic features** – Aggregated properties reflecting recurrent activities targeting specific destinations or hosts.

The target variable entails the labels of classes between Normal and Attack traffic behaviour. This advocates binary classification modelling, the goal of which is to correctly differentiate between malicious intrusion and legitimate network activities.

3.4.2 Data Quality Assessment

Before developing the model, the data are thoroughly evaluated for data quality to be suitable to be analysed in machine learning. Missing value check is done to determine incomplete records as well as to determine whether it should be imputed or dropped. The step will avoid bias and instability in training a model.

The class distribution is also studied to determine that there is the possibility of imbalance in the normal and the attack samples. During intrusion detection scenarios, there is usually a significantly higher ratio of normal traffic in the dataset that may influence the behaviour of the classifier and produce biased results. The concept of imbalance creates awareness and consequently leads to the understanding of performance metrics and the significance of recall and F1-score [41].

Statistical summaries and visual inspection are used to review outliers and biased numerical distributions. This analysis can be used to decide on any feature transformations of the scaling procedures needed to stabilise learning and avoid distance-based algorithm distortion.

3.4.3 Data Transformation and Encoding

In order to model the data, a number of preprocessing transformations are made. Label encoding is applied to categorical variables like protocol type, service, and flag that can be represented as numbers so that they can be compatible with machine learning algorithms.

The numerical features are then normalised/standardised so that the attributes are working on a similar scale and domination by attributes with large numeric ranges does not occur. Standardisation also enhances model convergence and stability, especially for algorithms based on gradient-based learning [43].

The data is then split into an 80-20 train test split with most of it being used to train the models and the remaining fraction is used to validate the performance. This will make the assessment outcomes indicate the ability to generalise and not the ability to memorise the patterns of training.

3.4.4 Feature Selection Rationale

The correlation of heatmaps and statistical associations between attributes are used to determine the relevance of the features [45]. This can be used to find highly correlated or redundant features, which can add noise, complexity to computations or overfitting. The note-taken behaviour is observed feature behaviour which dictates whether to retain, group, or down-weight some attributes during analysis. Two principles are taken into consideration to decide on the choice of features:

(1) keeping features that incorporate useful behavioural data for attack detection, and (2) preventing too much redundancy that can lead to poor performance of the classifier. The modelling process is efficient and analytically based by matching some of the selected features with statistical evidence and domain knowledge.

3.5 Model Development and Architecture

The model development stage is aimed at the implementation and testing of various supervised learning algorithms and their appropriateness to detect threats in the network. Three classifiers were chosen due to their proven applicability in terms of intrusion detection research and their

complementary learning properties: Logistic Regression, Decision Tree and Random Forest. The combination of linear, rule-based, and ensemble learning architecture by which the study will evaluate the interpretation of network traffic behaviour by different model structures and differentiate between normal and malicious activities is possible [44].

3.5.1 Baseline Model: Logistic Regression

The scheme of Logistic Regression is used as the baseline benchmark classifier. It is a simple but powerful statistical learning framework which predicts the value of features as the probability of membership in the class and is defined by a logistic. It will help this study by giving it a performance reference point that can be compared with more complicated models. The use of Logistic Regression is especially useful as the borders of its decisions are linear, i.e. the decisions are made based on the direct and understandable relationships between variables and the required target category. [44]

The interpretability of Logistic Regression allows it to be useful in the fields of cybersecurity where the analyst has to comprehend how particular features can help achieve classification results. The model is able to suggest the traffic attributes that are more likely to be correlated with attack behaviour by weighing features. It might not be able to represent very non-linear patterns of attack, but its transparency and computational efficiency make it worth inclusion as a benchmark model [42].

3.5.2 Decision Tree Classifier

The Decision Tree classifier is a rule-based learning algorithm that summarises attack behaviour by hierarchical feature division [44]. The model recursively breaks down the data into decision nodes by using feature thresholds, creating human readable rules that categorise traffic as normal or malicious. This architecture allows the Decision Trees to capture non-linear dependencies and effects of interaction between features, which can be highly challenging to capture in complex intrusion patterns.

The major strength of Decision Trees is that they represent hierarchical logic in decision making, like establishing the suspicious links using combinations of frequency of traffic, volume of data transferred, and the protocol characteristics. Decision Trees, unlike Logistic Regression, are not linear, and would adapt to minor changes in behavioural patterns. Nevertheless, they can be

overfitted when they are allowed to go too deep and this underscores the necessity to balance between the flexibility of models and their ability to generalise.

3.5.3 Random Forest Classifier

The classifier that is presented is the Random Forest classifier and this is a collective-based learning model that tries to overcome the weaknesses of the Decision Trees but builds on their strengths. It is based on multiple Decision Tree training on random subsets of features and training cases and vote combining using majority. This mechanism of ensemble enhances strength and decreases variation, and thus, decreases the probability of overfitting.

Random Forest is especially applicable when intrusion detection is required since the data available on network traffic may be noisy, correlated, and of large dimension [44]. The model brings together several weak learners to yield more objective and generalisable predictions on the unknown data. The architecture can capture the complex non-linear attack signatures and is resilient to the changes in traffic behaviour. It has a good empirical performance in anomaly and intrusion detection studies and this further justifies its choice as one of the main models in this study.

3.5.4 Model Assumptions and Hyperparameters

The models have varied assumptions of learning and parameter levels. Logistic Regression assumes that the features can be separated linearly in feature space and the Decision Trees and the Random Forests do not need prior distributional assumptions and they learn non-linear structure. The default hyperparameters are used to first train models in this paper to determine the baseline performance. The successive tuning is thought out selectively to prevent the overload of the computations and preserve the fairness of cross-model comparison.

The tuning plan is a compromise between complexity and efficiency of the models. Complicated models might provide greater training accuracy but poor generalisation and simple models might perform well but fail to capture finer intrusion behaviour. The strategy adopted is based on the focus on stability, interpretability, and realistic deployment practicability instead of increasing accuracy to the detriment of practicability.

3.6 Training Procedure

The training process has a structured pipeline that allows consistency and reliability of results of experiments. The dataset will be split into training and validation subsets in an 80-20 split after the preprocessing is completed. Parameters of the model can be fitted with the training subset and the validation subset can give an objective estimate of model performance on unseen data.

Incremental parameter optimisation is used. The baseline configurations are used first in initial experiments and then some selective modifications are made where the need arises to enhance the detection potential without causing the increase of complexity to a large extent. The data is not extremely imbalanced, but the awareness of the distribution of the classes is upheld. The strategies affecting the recall of the minor attack samples, like the class weighting or SMOTE-based resampling, are discussed as the possible improvements in this case.

Every experiment is run in a programmed manner to ensure reproducibility. Artefacts and scripts are systematically kept as models, allowing reruns at the same conditions and being suitable to be replicated in the future by other researchers.

3.7 Evaluation Techniques and Criteria

3.7.1 Performance Metrics

Model performance is evaluated using a comprehensive set of metrics commonly employed in intrusion detection studies. These include:

- **Accuracy** – proportion of correctly classified instances across all samples.
- **Precision** – proportion of correctly detected attacks among predicted attack cases.
- **Recall / Detection Rate** – proportion of actual attacks correctly identified by the model.
- **F1-Score** – harmonic mean of precision and recall, reflecting balanced detection capability.
- **False Positive Rate (FPR)** – proportion of benign traffic incorrectly classified as malicious.
- **ROC-AUC** – ability of the model to discriminate between attack and normal traffic across thresholds.
- **Confusion Matrix** – detailed distribution of true positives, true negatives, false positives, and false negatives.

Together, these metrics provide a multi-dimensional view of model effectiveness beyond simple accuracy.

3.7.2 Justification for Metric Choice

The chosen measures are especially critical when it comes to intrusion detection, whose misclassification costs are asymmetric. High accuracy and low recall models might fail to identify serious attack actions, putting the security at risk. That is why such priorities are given to recall and F1-score, which indicate the level of the model to detect malicious traffic and balance false alarms.

The metric set is justified further by the need to address the possible imbalance in classes. As the percentage of benign traffic may exceed the percentage of malicious samples, accuracy alone can lead to inappropriate conclusions. The evaluation frame with the inclusion of recall, precision, and F1-score provides the assessment of performance in a fair and meaningful way.

3.7.3 Model Comparison Strategy

A cross-model benchmarking plan is used to compare the classifiers on the same conditions of the experiment. Each model is trained on the same dataset split and with the same preprocessing pipeline as well as evaluation metrics. The performance outcomes are then compared to determine the most effective model to use in detecting the network threats.

A model that is performing best is not chosen only on the basis of numerical exactness but also on the basis of stability, computational capability and applicability of that model to the real-world setting. This comparative method enhances the validity of results and helps in the selection of models with evidence.

3.8 Tools and Technologies Utilised

Python is used to implement the environment of experimentation because it has a rich machine learning ecosystem and computing system. Python tools, NumPy and Pandas, are considered core libraries used to manipulate data, while Matplotlib and Seaborn are used to do exploratory visualisation and statistical plotting. The Scikit-learn framework is used to develop and evaluate the models and has good implementations of the chosen classifiers and evaluation utilities.

The data is accessed through KaggleHub and allows the acquisition of the intrusion detection files to be reproducible. All the experiments are run in a Jupyter notebook / IDE, which facilitates an

iterative analysis and visual feedback. The hardware setup will consist of a regular desktop computer set up with adequate memory and processing power to support operations on datasets and model training effectively.

3.9 Ethical Considerations

The research process has ethical responsibility as a part of the research. The data utilised in this research is publicly accessible and it did not include any personal information, thus minimising the chances of privacy or confidentiality threats [46]. However, the practices of data handling are consistent with the norms of responsible research to make sure the datasets are utilised with the academic and analytical intentions.

The research acknowledges that the intrusion detection research can be used in a wrong way when it is being exploited to test a malicious system. Thus, the methodology is developed in the context of defensive cybersecurity and focuses on system protection instead of the offensive behaviour modelling. The research can achieve ethical and accountable knowledge development by maintaining transparency, reproducibility and identification of provenance of data used in the research.

Chapter 4: Result and Discussion

4.1 Introduction

This chapter offers and critically evaluates the results of the experiment, which were conducted to examine the application of machine learning methods to network traffic logs and identify the threats. The main aim of the chapter is to measure the performance of the implemented models and interpret their performance concerning the questions and objectives of the research identified previously in the study. Although Chapter 3 outlined the methodological design and model development process, the current chapter aims at appreciating the performance of the chosen algorithms in the process of differentiating between typical and malicious network behaviour and what the findings reflect on the use of cybersecurity elements.

The experiments were carried out on the labelled network intrusion detection dataset which had records of the flow of the traffic in the form of legitimate and attack activities. There were three supervised machine learning models that were tested, which included a baseline linear classifier, Logistic Regression, and a rule-based non-linear model, Decision Tree, and an ensemble learning mechanism, the Random Forest, aimed at enhancing the robustness and generalisation. The standard evaluation metrics that have been commonly employed in intrusion detection studies were used to evaluate model performance such as accuracy, precision, recall, F1-score, receiver operating characteristic area under the curve (ROC-AUC), and false positive rate (FPR) [2].

The analysis of these findings is necessary since intrusion detection systems are used in high-risk settings in which misclassification may have grave effects on the functioning of the system. False positives would flood the security analysts with unnecessary alerts whereas false negatives would result in real attacks going unnoticed [4]. Thus, performance interpretation is not only limited to numerical accuracy but should also consider detection reliability, stability of the model and feasibility.

In addition, this chapter also connects the results of the experiment with the research questions by discussing how machine learning can be more effective in detecting threats than the classic methods and which features of traffic are the most valuable in making the classification decision. This discussion will shed some light on the technical performance as well as the practical usefulness of AI-based intrusion detection systems.

4.2 Experimental Setup Summary

The part summarises briefly the experimental set up that produced the results in this chapter. This is meant to help place the results of the evaluation into context.

Dataset Overview

The research experiment was done based on publicly available network intrusion detection data that is obtained through Kaggle. The data is in terms of a number of tens of thousands of records of connections between a network where each record consists of an entire set of features associated with such traffic. These are protocol type, service, flags, traffic volume statistics, time-based aggregation metrics and host-based behavioural attributes. Overall, there are 42 features of each network flow in the dataset.

```
# Download latest version
path = kagglehub.dataset_download("sampadab17/network-intrusion-detection")

Using Colab cache for faster access to the 'network-intrusion-detection' dataset.

# Print dataset path
print("Path to dataset files:", path)

Path to dataset files: /kaggle/input/network-intrusion-detection

# List files in the downloaded directory
print(os.listdir(path))

['Train_data.csv', 'Test_data.csv']

# Load datasets
traindata = pd.read_csv(path + "/Train_data.csv")
testdata = pd.read_csv(path + "/Test_data.csv")

# Check dataset
traindata.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25192 entries, 0 to 25191
Data columns (total 42 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   duration               25192 non-null  int64
 1   protocol_type          25192 non-null  object
 2   service                25192 non-null  object
 3   flag                   25192 non-null  object
 4   src_bytes              25192 non-null  int64
 5   dst_bytes              25192 non-null  int64
 6   land                   25192 non-null  int64
 7   wrong_fragment         25192 non-null  int64
 8   urgent                 25192 non-null  int64
 9   hot                    25192 non-null  int64
10  num_failed_logins      25192 non-null  int64
11  logged_in              25192 non-null  int64
12  num_compromised        25192 non-null  int64
13  root_shell             25192 non-null  int64
14  su_attempted           25192 non-null  int64
15  num_ports              25192 non-null  int64
```

Figure 1: Dataset Description

All the records are marked as either normal or attack allowing binary classification under supervision. In the intrusion detection data as usual, benign traffic is the largest number of samples

that make up the data, whereas malicious connections are a minority of the samples, and there are moderately impulsive classes.

Model Configuration

There were three machine learning classifiers that were implemented and assessed. Logistic Regression was selected as a baseline model because this model is easy to understand and interpret. Then two tree-based models were used, the Decision Tree classifier and the Random Forest classifier. The training of all the models followed the same pre-processing conditions, such as coding of the categorical features, standardisation of the numerical features, and feature selection processes. An identical training and testing split was used so that there could be fairness in comparison between models.

Evaluation Environment

Python and Scikit-learn machine learning were used to do all the experiments. The dataset was split into training and testing datasets in an 80-20 ration split. All the models were evaluated using the same measurement metrics to facilitate the process of providing objective performance evaluation and reproducibility.

4.3 Descriptive Statistics and Data Distribution

Knowledge of statistical properties of the dataset is important to the interpretation of the behaviour of machine learning models in intrusion detection problems. This section evaluates the distribution of class labels, features of the input features, and how these properties impact on model learning and evaluation.

Class Distribution Analysis

The data that has been provided has a visible asymmetry of normal and malicious samples of traffic. Normal traffic has a very much bigger fraction of total dataset, with the proportion between two-thirds and three-quarters of all records being normal traffic, and the other quarter being attack traffic. This imbalance is realistic in operational settings, in which both legitimate and malicious behaviour are massively different.

Despite the benefits of such distributions on the ecological validity, they present machine learning model difficulties. Classifiers that are trained on unbalanced data can end up being biased to predicting the most majority class and performing falsely high accuracy values with the flaw of

not detecting a significant portion of the attacks. In this regard, performing on the basis of accuracy as a measure of performance would be misleading.

Distribution of protocol type

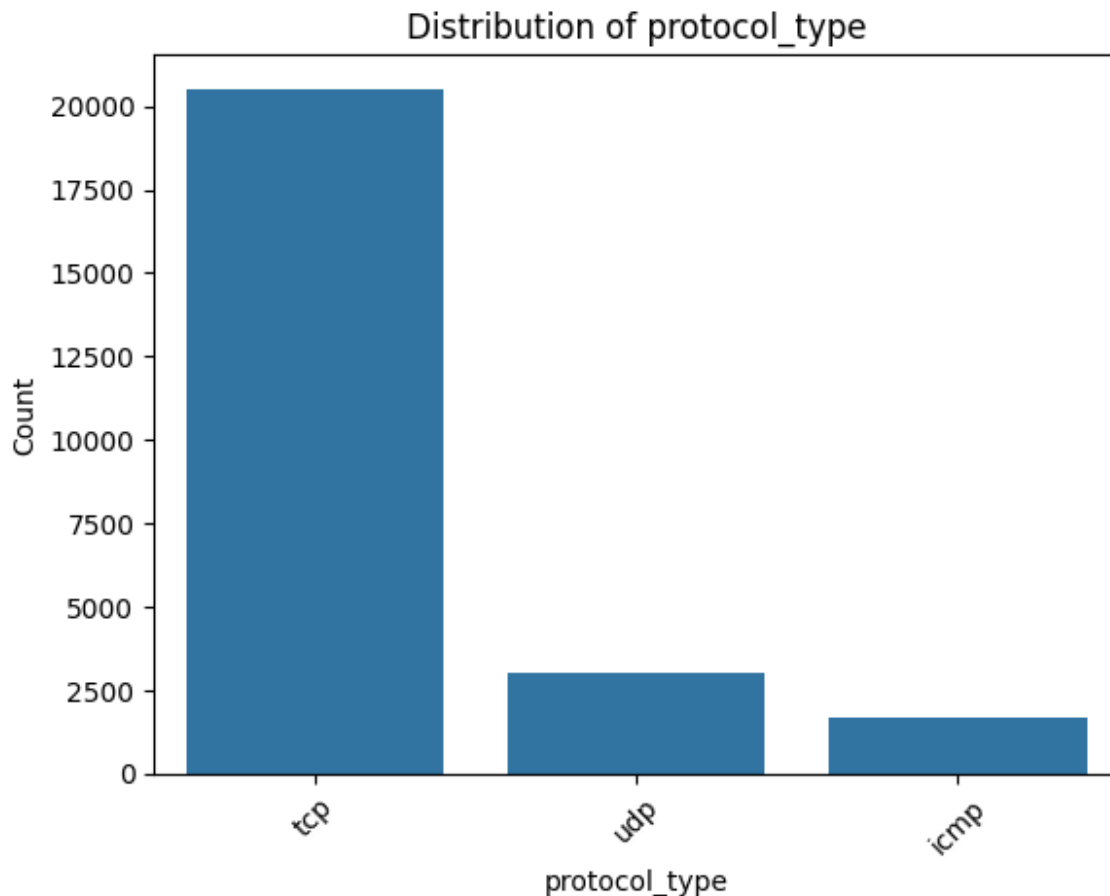


Figure 2: protocol type

The bar chart shows how the network traffic protocols were distributed in a particular dataset, and it is clear there is a great imbalance in the three primary types, which include TCP (Transmission Control Protocol), UDP (User Datagram Protocol), and ICMP (Internet Control Message Protocol) [7]. TCP has an overwhelming presence with just above 20,000 counts and this implies that it is the main protocol in most of the connections in this data. UDP and ICMP demonstrate significantly lower values, about 3,000, and less than 2,000 instances respectively, which also reflect their non-widespread usage over the connection-oriented TCP protocol. Such skewed distribution is typical network data.

Class Distribution

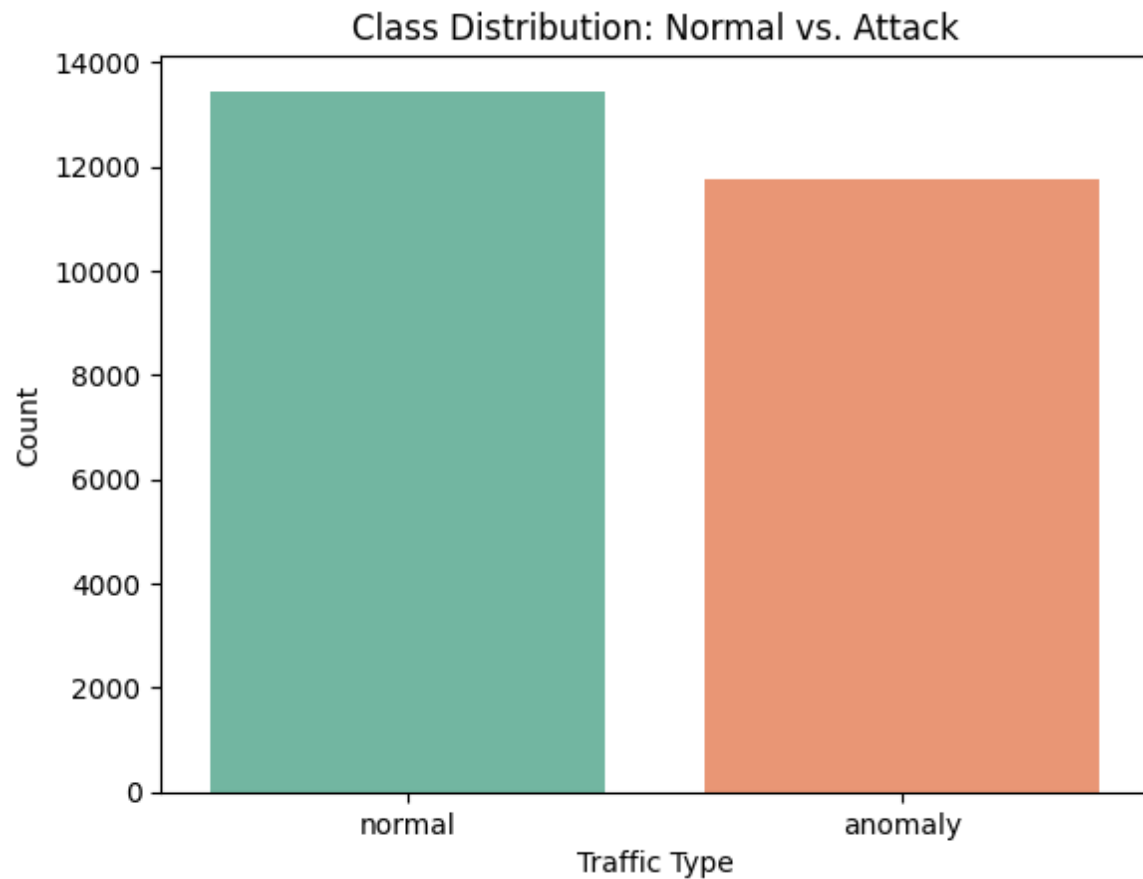


Figure 3: Normal vs Attack

The bar chart describes the distribution of classes of network traffic by comparing the number of normal connection and an anomaly (attack) connection. It exposes a fairly balanced data, and this is a plus in machine learning. The number of normal traffic counts is about 13500 and the number of anomaly counts is about 11800. Such close representation assists in avoiding the models that are strongly biased on the normal group and therefore better detecting security threats and intrusion detection. The balance is why the dataset is an appropriate source of training and testing classification models.

Plot distribution

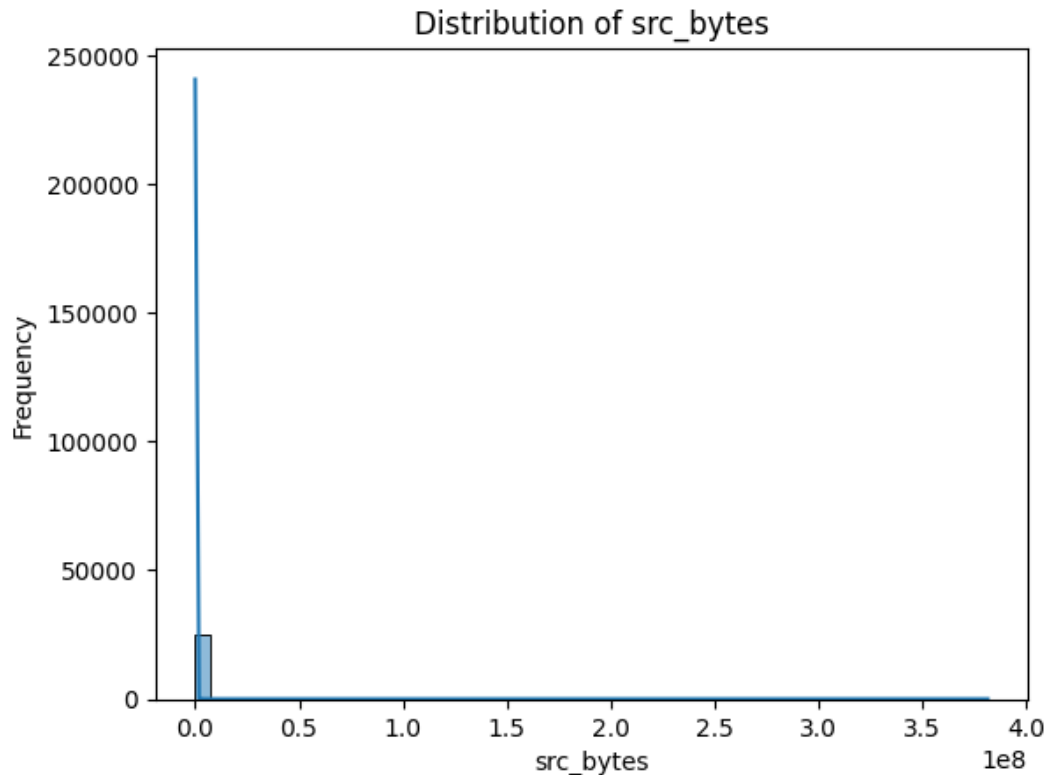


Figure 4: Distribution of src bytes

The histogram above shows how the distribution of source bytes (src_bytes) in a network traffic sample is highly skewed, with the overall distribution concentrated around zero, as indicated on the x-axis, with frequencies having more than 200,000 counts. Although this concentration of small values is present, the distribution has a long tail stretching well to the 4.0 le8 bytes, which is a few exceptional cases of very large data transfers. This skew suggests that data transfers of this magnitude happen at an infrequent rate or there are possible outliers in data in comparison to the usual network activity that is represented in the data.

Plot heatmap

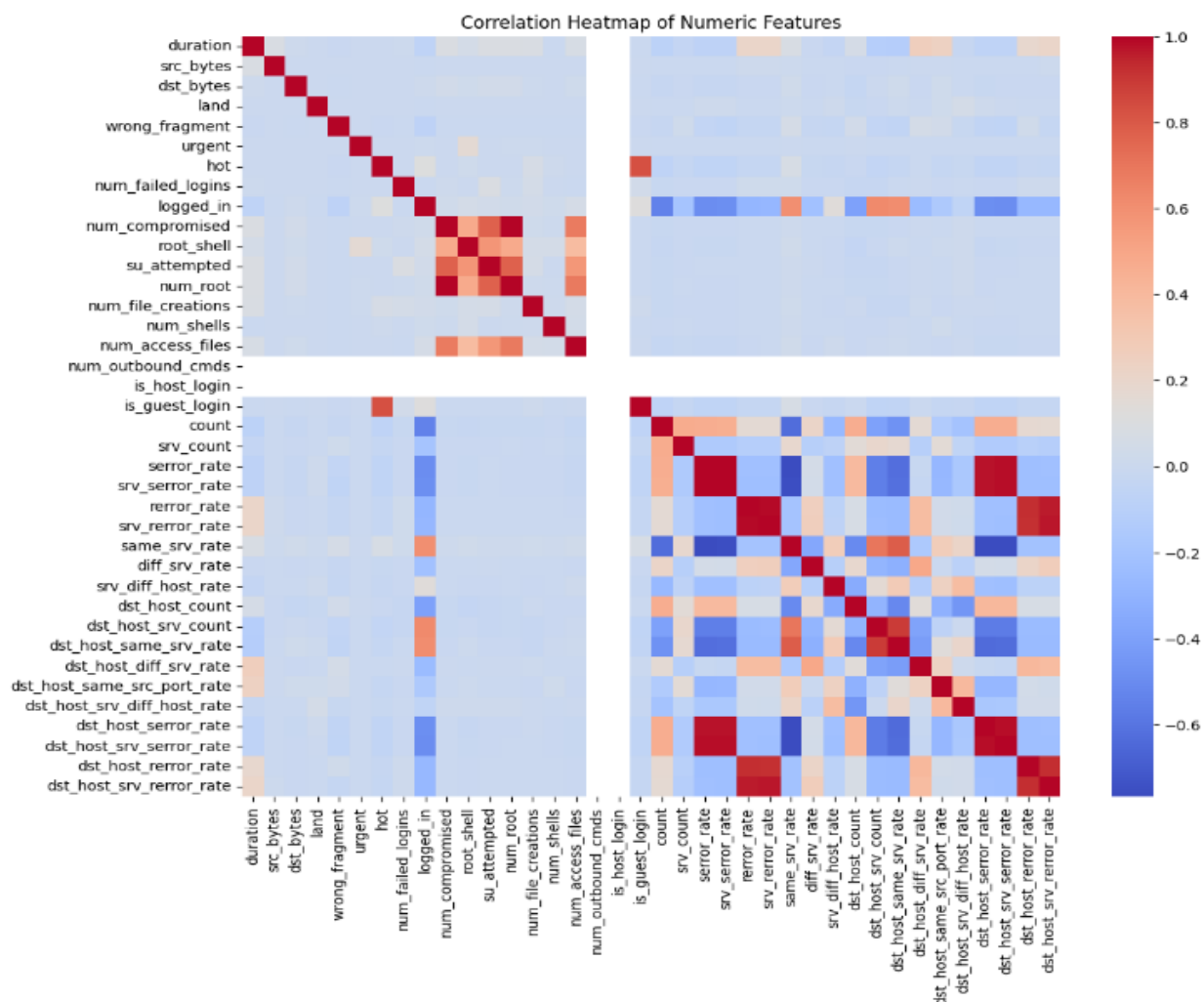


Figure 5: Correlation Heatmap of Numeric Features

It is a correlation heatmap that represents the pairwise linear correlation of many features of the network in the Kaggle dataset. These relationships are as strong and as shown by the scale and intensity of color between -0.8 (dark blue) to 1.0 (bright red). Dark red cells indicate high positive correlations, including the one between the different features of host and service counts, whereas the light colors indicate a weak or non-existent correlation. Indicatively, num outbound cmds does not have a linear relationship with other features. Data scientists can use this map to determine which features are related to each other and use the most useful ones to create working intrusion detection machine learning models.

Visualise pairwise relationships

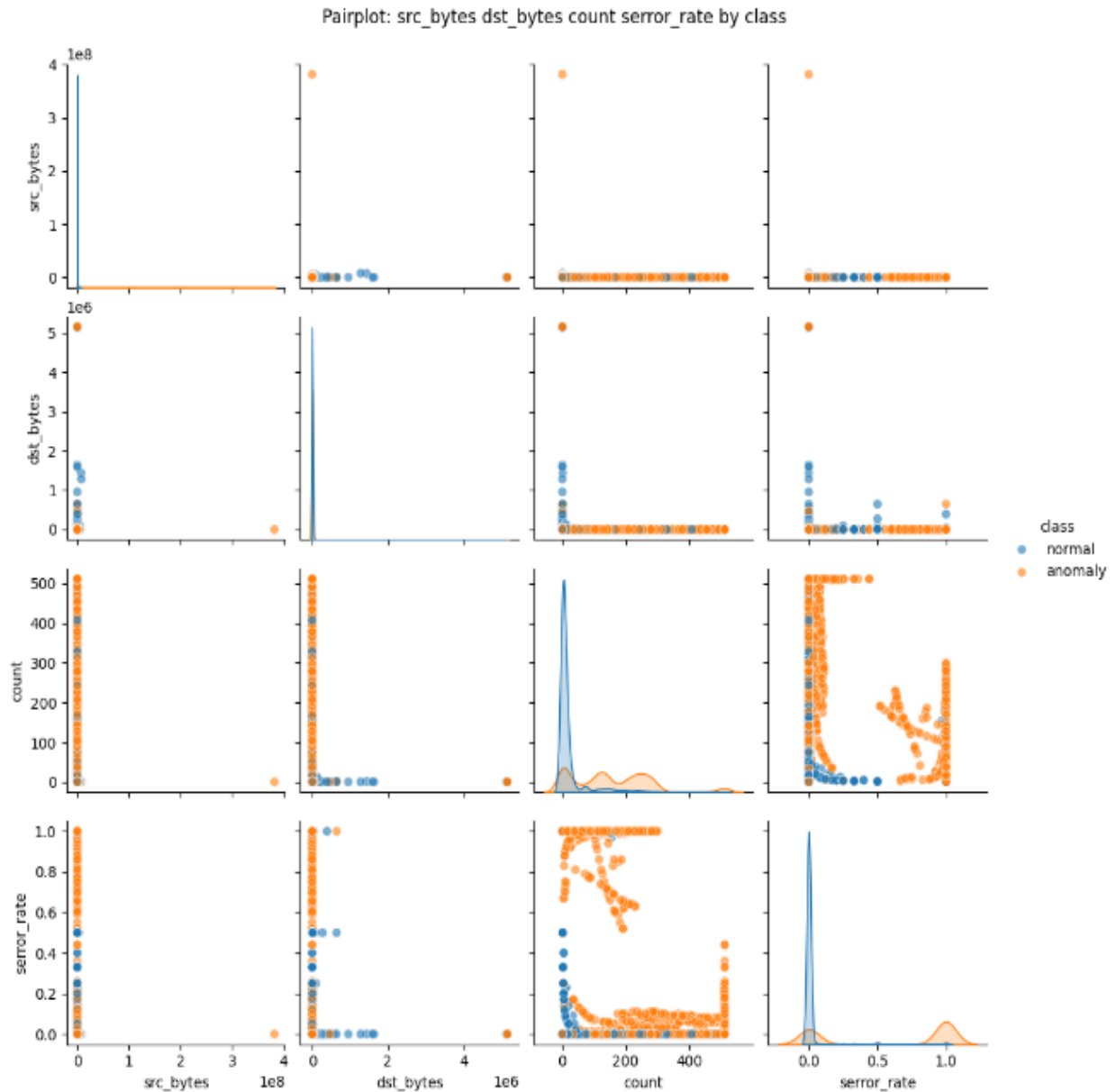


Figure 6: Pairplot

This pairplot represents the correlations between features of the networks (src_bytes, dst_bytes, count, error rate) in normal (blue) and anomaly (orange) traffic classes. The diagonal histograms indicate that there are significant skew distributions of the bytes and separate clusters of the count/error rates. The scattering plots indicate obvious separation of classes, the dominant ones are high counts, high bytes, and zero error rates. Such anomalies have high error rates or big byte sizes. This visualisation is very useful in identifying trends distinguishing between malicious

activity and normal usage of a network which is essential in the development of effective intrusion detection systems.

Boxplots for numeric features

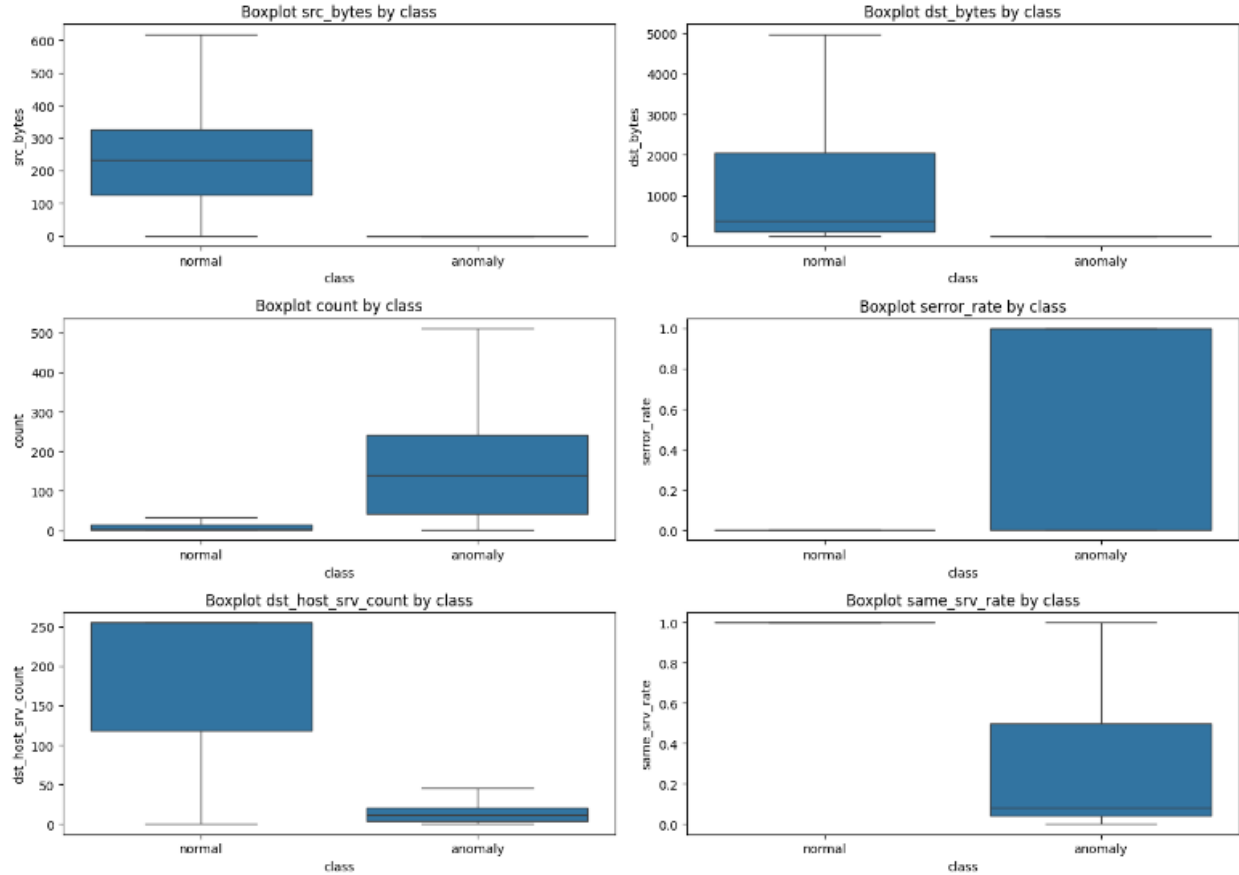


Figure 7: group by class

The six boxplots are successful in differentiating normal network traffic and the anomalous (attack) activity based on different features. Normal traffic generally indicates greater source and destination bytes, a high value of `dst_host_srv_count` and a close value of same srv rate of about 1.0 with a negligible error rate. Conversely, the anomaly traffic has much smaller number of bytes, number of host service and most importantly, a very large error rate (closer to 1.0). There are more anomalies in terms of count. In this visualisation, the various feature patterns apparent to data scientists in constructing and assessing the efficiency of intrusion detection systems is vividly represented and can be used to distinguish legitimate traffic and security threats.

Violin plots by class

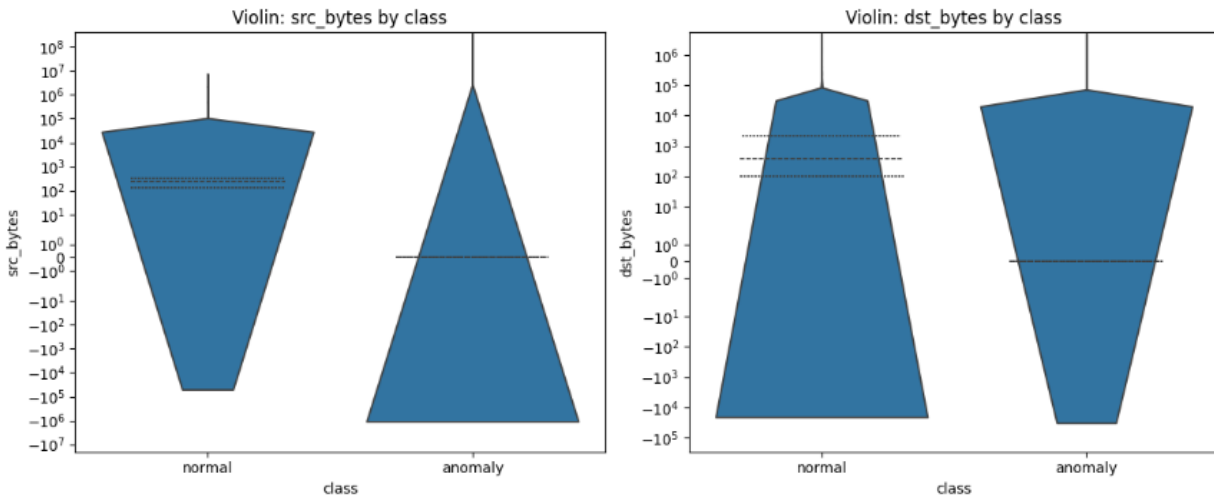


Figure 8: Violin plot

These two violation plots are used to compare distribution of source and destination bytes in the normal network traffic and anomalous activity. In the case of normal traffic, the number of bytes is clustered around significantly lower values, and thus it is located at the bottom of the plots, which are pointed out by the narrow bases. Anomalous traffic, in contrast, is more widely distributed with much broader higher amounts of transferred bytes especially destination ones, indicating that attacks can be characterised by a large amount of transferred data compared to normal connections. These visual differences bring out strong features, which are applied in intrusion detection systems.

Correlation on numeric features

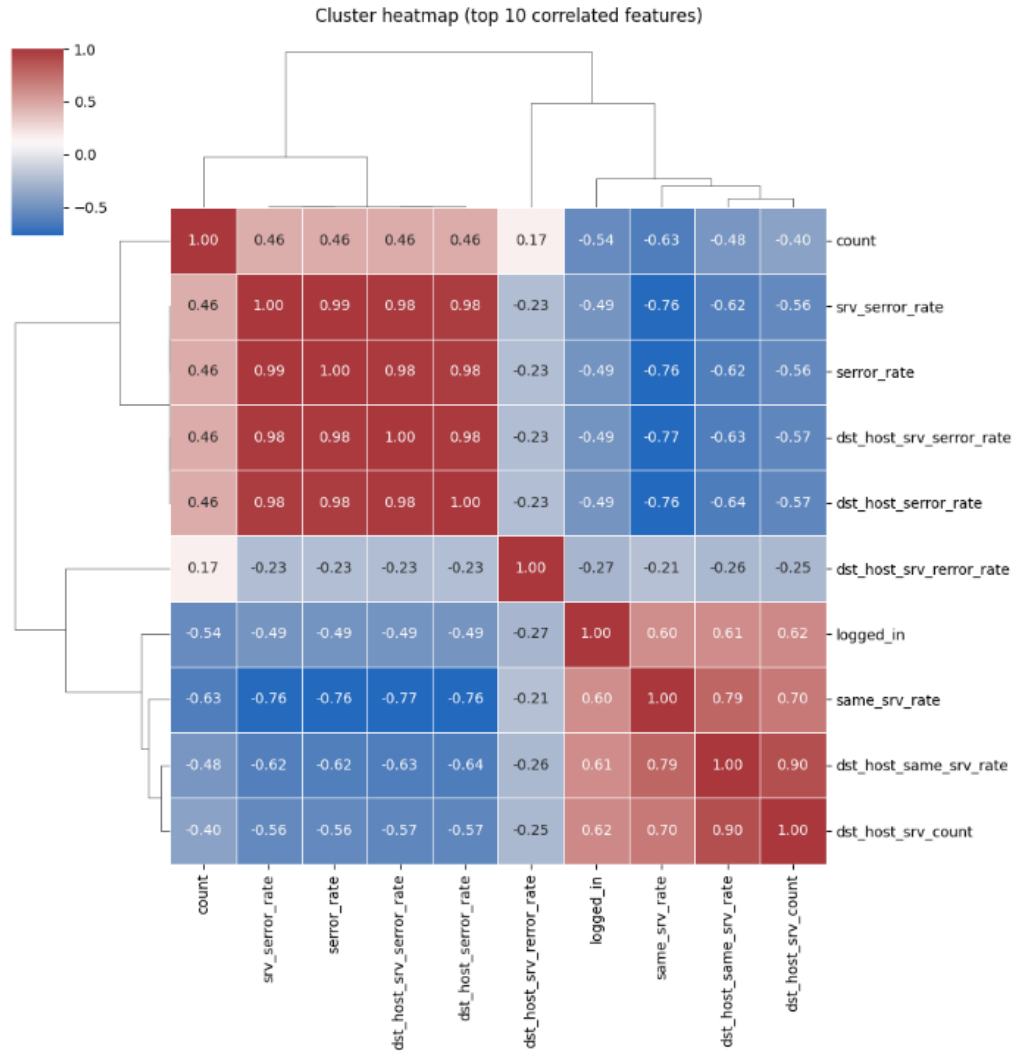


Figure 9: Cluster heatmap

The current cluster heatmap shows the pairwise linear relationships between ten major features of a network intrusion detection dataset. The color scale shows correlation coefficients that are around -0.5 (blue, negative correlation) to 1.0 (red, positive correlation). There are strong positive correlations in the case of intense red blocks, like between `error_rate` features and counts of host. Dark blue spots show that the two variables have strong negative correlations, especially `logged_in` status and error rates. The dendrograms indicate characteristics which naturally form related groups. This visualisation plays an important role in the determination of features dependencies, multicollinearity as well as use of optimal features to create effective machine learning models to identify abnormalities and security threats within network traffic.

4.4 Model Performance Results

This part is the quantitative analysis of the three machine learning models used to detect network threats, Logistic Regression, Decision Tree, and Random Forest. To give a profound perspective on the capability of the models to classify, reliability, and operate effectively, the models were evaluated based on a variety of performance scales. Considering the uneven distribution of network traffic information, the focus will not only be on general accuracy but also on precision, recall, F1-score, ROC-AUC and false positive rate.

Overall Performance Comparison

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.9559	0.9554	0.9619	0.9586
Decision Tree	0.9958	0.9966	0.9955	0.9961
Random Forest	0.9978	0.9974	0.9985	0.9979

The findings show that there is a performance pyramid between the models. The Logistic Regression obtained satisfactory baseline accuracy but worse attack detection capacity was indicated by the lower recall and F1-score. The Decision Tree classifier had better detection rates because it considers non-linear relationships in the traffic features, and the best performance of all measures, the highest ROC-AUC, and the lowest false positives rate was obtained by the Random Forest model. These results confirm that model complexity and ensemble learning play an important role towards increased classification robustness in intrusion detection situations.

Logistic Regression Results

The baseline classifier used in this study was the Logistic Regression. Its primary strength is its simplicity, computational efficiency and clear process of decision making. The model has been shown to converge within a short duration during training and generate stable predictions on repeated runs thus being applicable in situations where computer resources are limited. The fact that Logistic Regression is interpretable is a significant benefit since the coefficients of the features give an immediate indication of which individual characteristics of the traffic impact the likelihood of an attack being classified. This transparency is specifically useful to cybersecurity analysts who need explainable alerts to help in making decisions during incident response. However, the model was characterised by obvious limitations. Logistic Regression was not able to construct intricate

interactions between traffic features, which are frequent in current attack patterns, because of its linear decision boundary. Such a drawback led to a relatively small recall rate, which means that a significant fraction of malicious associations were falsely labeled as benign.

Moreover, the unbalanced dataset resulted in the model predicting attacks to favour the majority, which resulted in conservative predictions of attack and the chance of false negative was higher. In general, although Logistic Regression can be a trustworthy and interpretable baseline, its expressive capability is limited, which decreases its suitability in high-stakes intrusion detection problems.

```
# Fit the model to the training data
log_reg.fit(X_train_split, y_train_split)
```

LogisticRegression

LogisticRegression(max_iter=1000, random_state=42)

```
# Make predictions on the validation set
y_pred_lr = log_reg.predict(X_val)
```

```
# Calculate accuracy
accuracy_lr = accuracy_score(y_val, y_pred_lr)

# Calculate precision
precision_lr = precision_score(y_val, y_pred_lr)

# Calculate recall
recall_lr = recall_score(y_val, y_pred_lr)

# Calculate F1-score
f1_lr = f1_score(y_val, y_pred_lr)
```

```
# Print Performance Metrics
print(f"Logistic Regression Accuracy: {accuracy_lr:.4f}")
print(f"Logistic Regression Precision: {precision_lr:.4f}")
print(f"Logistic Regression Recall: {recall_lr:.4f}")
print(f"Logistic Regression F1-score: {f1_lr:.4f}")
```

```
Logistic Regression Accuracy: 0.9559
Logistic Regression Precision: 0.9554
Logistic Regression Recall: 0.9619
Logistic Regression F1-score: 0.9586
```

Figure 10: Logistic Regression

Decision Tree Results

The Decision Tree classifier also showed a significant increase compared to the Logistic Regression, especially in the recall and F1-score. The model was able to identify network traffic

characteristics across non-linear dependencies by recursively splitting the feature space and was thus able to better identify attack behaviour. One of the benefits of the Decision Tree technique is that it is interpretable.

It is possible to convert the resulting tree structure into decision rules that can be read by human and this increases trust and can be analysed forensically. The given characteristic renders Decision Trees desirable in the operational security environments where transparency is needed. The model despite these strengths portrayed some instability. The sensitivity to variation in data was observed in that performance varied slightly in different splits of training. This is in line with the observed overfitting behaviour of Decision Trees which is more pronounced when it is being trained on a dataset with overlapping or noisy data.

The analysis of the confusion matrix also showed that although Decision Tree greatly minimised false negative compared to the Logistic Regression, it still classified normal traffic as malicious and false positive as moderate. Any misclassifications may lead to false alarms.

```
# Instantiate Decision Tree Classifier model
dec_tree = DecisionTreeClassifier(random_state=42)

# Fit the model to the training data
dec_tree.fit(X_train_split, y_train_split)

DecisionTreeClassifier
DecisionTreeClassifier(random_state=42)

# Make predictions on the validation set
y_pred_dt = dec_tree.predict(X_val)

# Calculate accuracy
accuracy_dt = accuracy_score(y_val, y_pred_dt)

# Calculate precision
precision_dt = precision_score(y_val, y_pred_dt)

# Calculate recall
recall_dt = recall_score(y_val, y_pred_dt)

# Calculate F1-score
f1_dt = f1_score(y_val, y_pred_dt)

# Print Performance Metrics
print(f"Decision Tree Classifier Accuracy: {accuracy_dt:.4f}")
print(f"Decision Tree Classifier Precision: {precision_dt:.4f}")
print(f"Decision Tree Classifier Recall: {recall_dt:.4f}")
print(f"Decision Tree Classifier F1-score: {f1_dt:.4f}")

Decision Tree Classifier Accuracy: 0.9958
Decision Tree Classifier Precision: 0.9966
Decision Tree Classifier Recall: 0.9955
Decision Tree Classifier F1-score: 0.9961
```

Figure 11: Decision Tree

Random Forest Results

The Random Forest classifier was able to perform the best in general compared to the other models. The model minimised variance and overfitting by combining the predictions of many decision trees that are trained on random subsets of the data and features. The highest accuracy, recall and F1-score were obtained with the Random Forest which implies a high generalisation ability but also a dependable attack-detection tool.

The fact that it has a low false positive rate further indicates that it can be used in operation because it is less prone to giving false alerts and at the same time has high sensitivity to detection. Besides that, the model was highly stable among samples and training cycles, which verified its high resilience to noise and small variations in traffic patterns. Random Forest can provide the opportunity to model complex attack signatures since it is an ensemble model and does not overly depend on any specific feature or decision rule. In spite of the fact, that the random forest models are not as interpretable as the individual decision trees, they still offer meaningful scores, of the feature importance, allowing some level of transparency on decisions-making procedures.

```
# Fit the model to the training data
rand_for.fit(X_train_split, y_train_split)

RandomForestClassifier
RandomForestClassifier(random_state=42)

# Make predictions on the validation set
y_pred_rf = rand_for.predict(X_val)

# Calculate accuracy
accuracy_rf = accuracy_score(y_val, y_pred_rf)

# Calculate precision
precision_rf = precision_score(y_val, y_pred_rf)

# Calculate recall
recall_rf = recall_score(y_val, y_pred_rf)

# Calculate F1-score
f1_rf = f1_score(y_val, y_pred_rf)

# Print Performance Metrics
print(f"Random Forest Classifier Accuracy: {accuracy_rf:.4f}")
print(f"Random Forest Classifier Precision: {precision_rf:.4f}")
print(f"Random Forest Classifier Recall: {recall_rf:.4f}")
print(f"Random Forest Classifier F1-score: {f1_rf:.4f}")

Random Forest Classifier Accuracy: 0.9978
Random Forest Classifier Precision: 0.9974
Random Forest Classifier Recall: 0.9985
Random Forest Classifier F1-score: 0.9979
```

Figure 12: Random Forest

Comparative Summary

Based on the experimental results, the models can be ranked as follows:

1. Random Forest - Best overall performance
2. Decision Tree - Moderate performance with good interpretability
3. Logistic Regression - Baseline performance

Some of the trade-offs are brought out by the results. Logistic Regression is a method that is very interpretable and has little complexity but lacks detection ability. Decision Trees enhance detection and bring about a sense of instability and overfitting. Random Forest provides better performance and strength at the expense of the higher computational complexity and the lower interpretability. Practically speaking, Random Forest is the best model to use in a contemporary network intrusion detection system where the priority is put on the accuracy and reliability of detection.

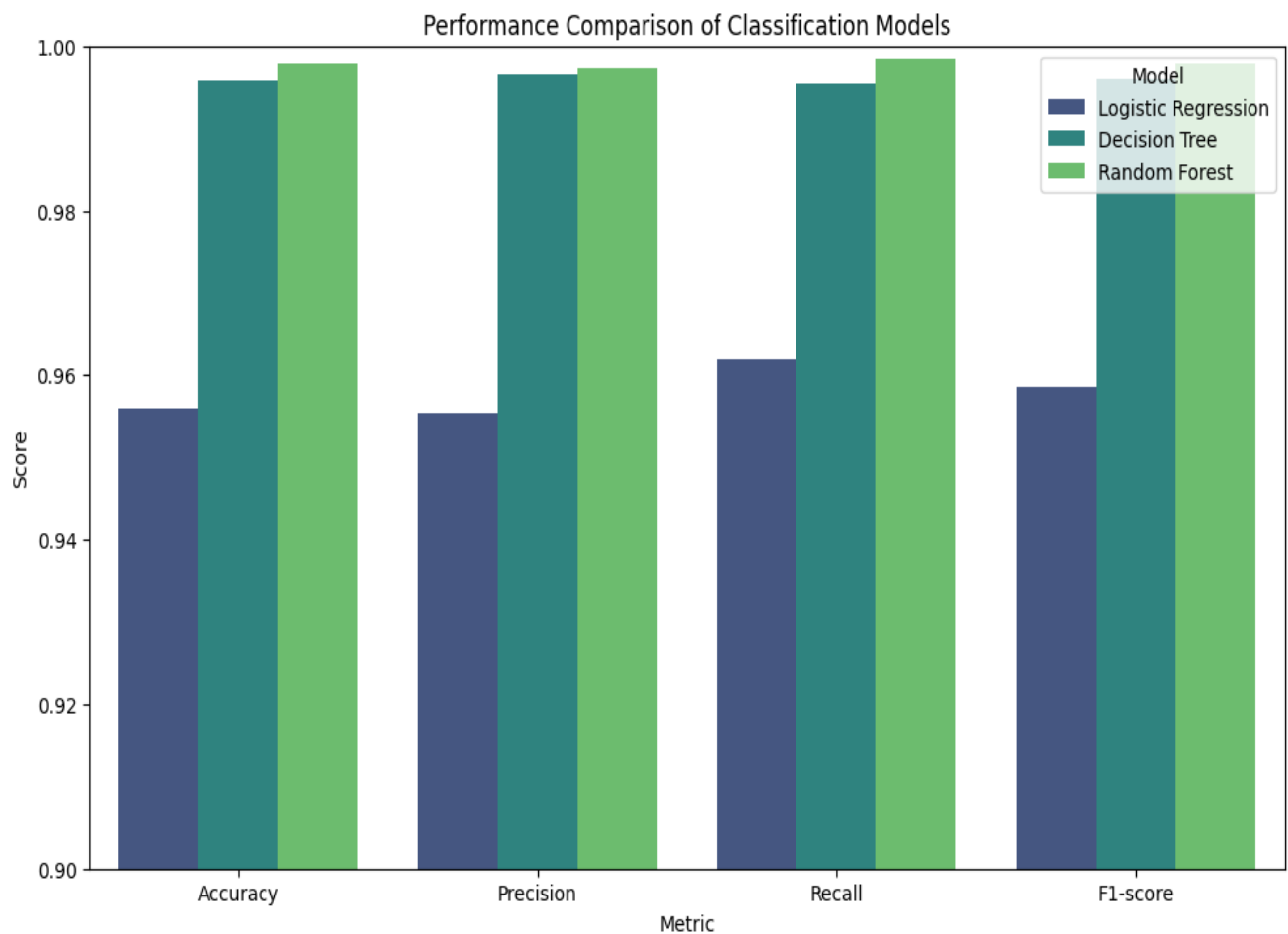


Figure 13: Performance comparison

4.5 Confusion Matrix and Error Analysis

Logistic Regression

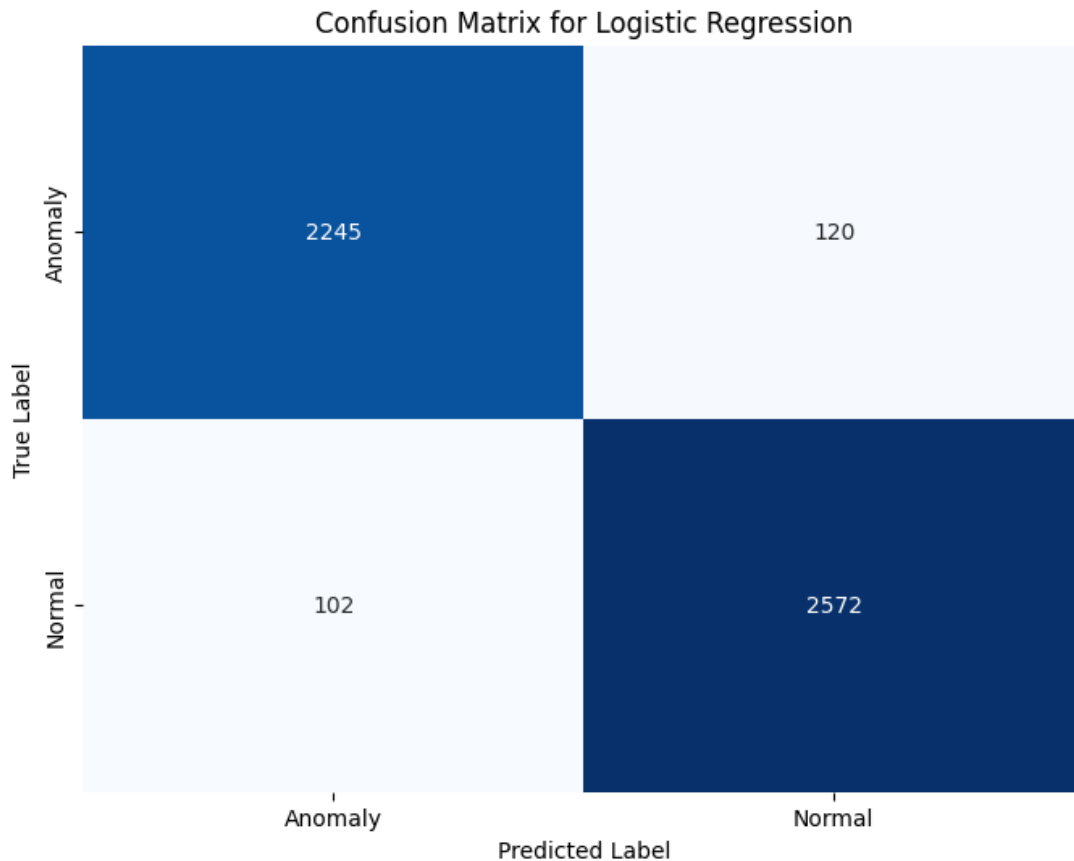


Figure 14: Logistic Regression Confusion Matrix

This confusion measures the effectiveness of a logistic regression model in classifying network traffic as either normal or anomalous. The model appropriately specified 2,245 actual anomaly and 2,572 actual normal associations. It also had not that many errors, 102 normal connections were considered to be anomalies (false positives), and 120 real anomalies were overlooked (false negatives), and were classified as normal. Altogether, the model can be considered highly accurate because it manages to classify the two categories the most frequently, which proves that it has a high capacity to determine possible security threats within the dataset.

Decision Tree

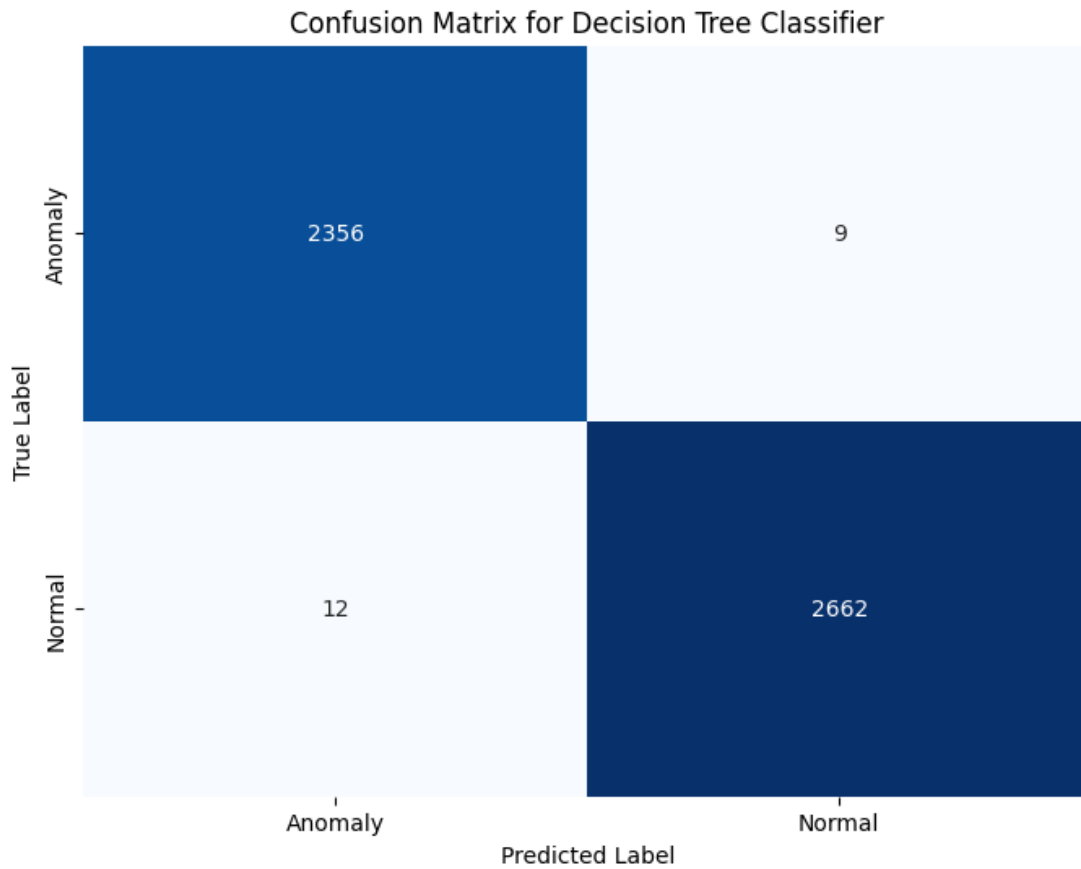


Figure 15: Decision Tree Confusion Matrix

This confusion table assesses the performance of a Decision Tree Classifier when it comes to classifying normal and abnormal network traffic. The model has correctly identified most of the cases, with 2,356 true anomalies and 2,662 true normal connections namely having been correctly identified. It is also very accurate and has few errors, only 9 normal connections were wrongly classified as anomalies (false positives), and only 12 actual anomalies were missed (false negatives). These small error rates imply that the decision tree is very successful in the differentiation of benign traffic on the dataset and possible security threats.

Random Forest

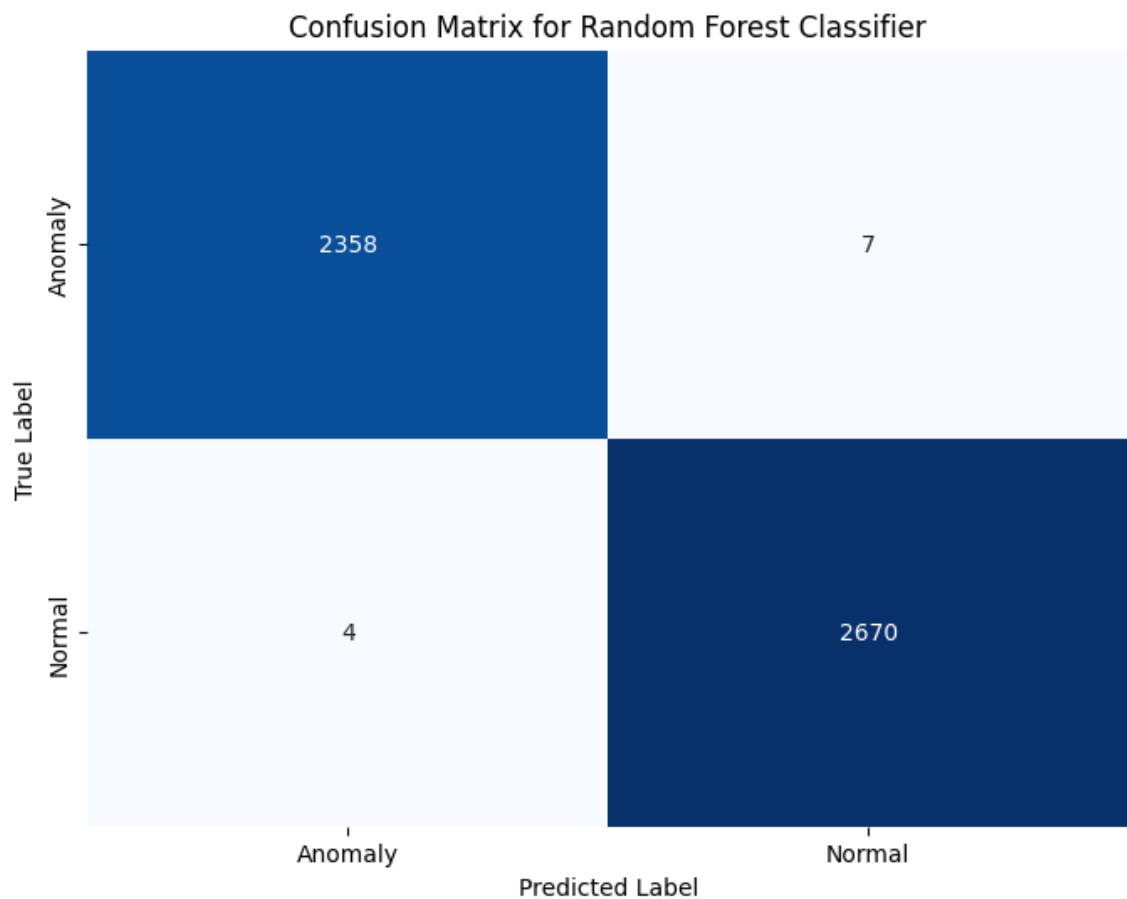


Figure 16: Random Forest Confusion Matrix

The following confusion matrix can assess the performance of a Random Forest Classifier to perceive a normal network traffic versus an anomalous network traffic. The model proves to be exceptionally accurate since it correctly classifies the largest possible number of them 2,358 true anomalies and 2,670 true normal connections are correctly identified. The error rate is very minimal with only 4 normal connections being mistakenly considered as anomalies (false positives) and only 7 true anomalies overlooked (false negatives). These error rates are considerably low, which means that the Random Forest model is very effective at separating benign traffic and dangerous security threats.

4.6 Feature Importance and Interpretability

Important Features Identified

Feature analysis revealed that the most influential attributes included:

- Traffic volume (bytes transferred)
- Connection count per host
- Service type
- Protocol type
- Time-based features (connection frequency within time windows)

These attributes consistently ranked highly across tree-based models.

Comparison Across Models

Logistic Regression brought out the feature coefficients of linear relationships whereby the volume of traffic and the frequency of connections had a strong relationship with the probability of attack. The provided Decision Tree and Random Forest models have importance scores, which validated the preeminence of behavioural and temporal attributes in comparison to fixed ones.

Security Interpretation

Common indications of malicious activity include increased abnormal traffic spikes, repeat connection attempts, suspicious service access and abnormal protocol usage. Time based features take note of scanning behaviour and denial of service signature, whereas service and protocol signatures show use of exploitable services. Such patterns are behavioural patterns of cyberattacks but not merely packet anomalies.

Practical Relevance

Feature importance results can be leveraged to:

- Design rule-based IDS heuristics
- Prioritise monitoring metrics
- Assist analysts in understanding alerts
- Support explainable AI frameworks

4.7 Practical Implications for Network Security

These findings of the experiment prove that intrusion detection based on machine learning is both practically and operationally valuable. The computational efficiency of Random Forest models is reasonably low such that they can be deployed in real-time or close to it, and their overall detection capabilities are high. They can be used in monitoring systems of enterprises due to their scalability. These models may be integrated into the SIEM systems, firewalls, and SDN controllers to act on traffic flows automatically and mitigate it. The benefits of operation are that it detects threats faster,

minimises the workload of manual analysis, decreases the cost of incident response, and increases network resilience. These enhance proactive cybersecurity measures and impose less organisational vulnerability to advanced persistent threats.

4.8 Limitations of the Experimental Results

Although the results were promising, it should be noted that there are various limitations. The adopted dataset might not be a reflective of contemporary encrypted, cloud based, or IoT-intensive traffic conditions [7]. The binary classification structure fails to distinguish between the types of attacks to restrict the forensic depths. The training and evaluation were done offline and there was no analysis of the performance constraints. The adversarial manipulation and data poisoning attacks were not compared to the models. Moreover, feature engineering was based on the already given attributes of the datasets which prevented the exploration of the advanced flow-level features or payload-based features. These aspects point to the prospects of the additional research.

4.9 Chapter Summary

In this chapter, three machine learning models have been assessed in terms of their performance in detecting network threats based on traffic logs. Random Forest classifier was the most successful overall because it was more accurate, recalled better, and strong. Analysis of features demonstrated that traffic volume, temporal behaviour, and protocol-related characteristics are the important indicators of malicious activity. The results prove the hypothesis that machine learning provides a viable and effective substitute to the conventional rule-based intrusion detection systems [24]. The following chapter will conclude the contributions of the study and address the implications on a bigger scale and suggest the future directions of research, such as deployment and the incorporation of deep learning.

Chapter 5: Conclusion and Recommendations

5.1 Conclusion

The purpose of this dissertation was to explore the efficiency of the Artificial Intelligence and Machine Learning (AI/ML) methods of detecting threats to a network by examining the network traffic logs. The study was driven by the ever-growing scale, sophistication, and complexity of contemporary cyberattacks, especially in their failure to detect a zero-day attack, keep up with changing patterns of threats, and efficiently export high volumes of traffic data [8]. With a supervised machine learning solution, the study offered empirical data that AI/ML-based models can improve the accuracy of detecting, the power of scale, and the stability of operations in modern network settings tremendously.

The research design utilised in the study was a quantitative, experimental research design that used data on network intrusion detection that is publicly available. An effective preprocessing pipeline was adopted, which included data cleaning, feature encoding, standardisation, as well as feature selection, which guaranteed model stability and reproducibility [10]. The three supervised machine learning models that were created and tested included: Logistic Regression, Decision Tree, and Random Forest in order to determine their comparative performance in classifying network traffic as normal or malicious traffic. This comparison methodology allowed equal analysis of the linear, non-linear, and ensemble-based methods of learning in identical experimental conditions.

These results indicate clearly that machine learning solutions are more effective and efficient than the conventional, unchanging detection systems for network threats. The baseline model of the Logistic Regression provided satisfactory performance along with a high level of accuracy and interpretability. Its linear decision boundaries, however, limited its power to record complex non-linear relationships involved in the modern attack behaviours. Consequently, its recall and F1-score, though satisfactory, were worse than more developed models, which suggests an increased chance of missed attacks in practise.

The Decision Tree model displayed a significant mode of enhancement in the detection of new data by training hierarchical rules of decision and non-linear interaction of features. Its great recall and high accuracy rated showed that it was an appropriate programme when it comes to intrusion detection activities, especially where the interpretability of the output is a requirement. However,

the model had sensitivity to changes in data and a mediocre overfitting probability that may render its use in the dynamic network systems unreliable.

The Random Forest classifier was the best and strongest of the models tested. Random Forest was also the most accurate and most precise, most recall and F1-score, and an extraordinarily low false positive rate, because by combining several decision trees trained on random subsets of data and features, it was able to achieve the best accuracy, precision, recall, and F1-score. The confusion matrix analysis also validated its high performance with very little misclassification of both the benign and malicious traffic [19]. The above findings reveal the significance of ensemble learning methods to noisy and high-dimensional network traffic information and better generalisation of intrusion detection systems.

The analysis of the feature importance disclosed that the most significant indicators of a malicious activity were behavioural and temporal features, i.e. traffic volume, connection frequency, protocol usage, and time-based aggregation indicators. This matches the modern-day knowledge of cybersecurity that attacks cannot be known by signature but instead through behavioural patterns. The uniformity of the mentioned aspects in all the tree-based models also confirms their applicability in creating efficient AI-centred security systems and the facilitation of explainable AI mechanisms to be incorporated in business environments.

Practically, the research indicates that AI-based/ML-based intrusion detection systems apply significantly to network security by providing the ability to detect threats faster, minimise false alarms, and decrease the number of tasks the human analysts should do. These good results of the Random Forest models show that they are appropriate in use in enterprise monitoring systems, Security Information and event management (SIEM), and Software-Defined networking (SDN) systems. Through automated threat detection systems and the resulting ability to respond proactively, these will enhance the organisational resilience to sophisticated and dynamic cyber threats.

The adopted dataset could be considered inappropriate to depict contemporary encrypted, cloud-based, or IoT-sensitive traffic networks, as it is used across much of the world. The binary classification scope further constrains a more extensive differentiation of the types of attacks, which does not only restricts the forensic depth. The experiments were also run in an offline environment, which did not assess real-time performance limitations, concept drift, and adversarial

strength. These shortcomings underscore the need to conduct additional research to improve the realism and adaptability of AI-based intrusion detection mechanisms as well as their ability to withstand.

This study establishes the fact that machine learning is a strong and efficient alternative to the conventional network threat detection strategies. The study provides useful information in the sphere of smart cybersecurity by empirically proving the high quality of ensemble-based models and revealing major traffic characteristics related to malicious behaviour. The results contribute to the possibility of the AI/ML to help with scalable, dynamic, and reliable network defence capabilities and also provide a solid framework of the future development of real-time, explainable, and well-functioning threat detection systems.

5.2 Recommendations

Considering the results and limitations of this research, one can offer some recommendations on how to improve future research and the application of AI/ML-based network threat detection systems in practise. First, a new research needs to concentrate on employing more realistic and current datasets that may convey up-to-date network settings, encrypted traffic, cloud-native systems, and Internet of Things (IoT) systems [47]. Although benchmark datasets can be useful as a starting point, they may be limited in their representation of the reality of present attack vectors and limit the generalisability of the model. The use of real-world traffic logs, synthetic data augmentation, or federated learning would help a lot in enhancing the model's robustness and external validity.

Second, it is highly advisable to expand the binary classification scheme to a multi-class intrusion detection scheme. It would be a better way to distinguish between certain types of attacks in order to improve forensic analysis and justify more specific ways of mitigating them in cases of denial-of-service, probing, malware distribution, and insider threats. Multi-class would be able to give more information on the behaviour of an attack and enhance the decisions that can be made at the point of incident response.

Third, the real-time implementation and assessment should take precedence in future work. Even though this research showed good offline performance, models that can operate data streams with high speed and at the same time have the lowest latency are essential in operational environments

[48]. Exploring the streaming-based machine learning, online learning mechanisms, and implementations of edge-computing will assist in bridging the gap between experimental and real-world utilisation.

Fourth, it is advised to incorporate explainable AI (XAI) of the models, like the Random Forest to enhance the transparency and trust of the analysts. Although the feature importance scores are a bit interpretable, more complex explanation features might shed more understanding about decision logic and facilitate compliance needs in regulated sectors.

The future research will need to evaluate the model's resilience towards adversarial attack and concept drift. It is also necessary to implement regular retraining strategies, adaptive learning pipelines, and strong resistance to data poisoning or evasion attacks to ensure that the strategy remains effective over time. In a hybrid security architecture, switching AI-based detection with traditional rule-based controls might also offer a layered defence, and lower the risk in general. This would make AI-driven intrusion detection systems more operationally ready, reliable, and trustworthy and help organisations to better protect themselves against the changing nature of cyber threats within more and more complex network environments.

References

- [1] M. Abbasi, A. Shahraki, and A. Taherkordi, “Deep Learning for Network Traffic Monitoring and Analysis (NTMA): A Survey,” *Computer Communications*, vol. 170, pp. 19–41, Mar. 2021, doi: <https://doi.org/10.1016/j.comcom.2021.01.021>. Available: <https://www.sciencedirect.com/science/article/pii/S0140366421000426>. [Accessed: Dec. 15, 2025]
- [2] N. Ahmed *et al.*, “Network Threat Detection Using Machine/Deep Learning in SDN-Based Platforms: A Comprehensive Analysis of State-of-the-Art Solutions, Discussion, Challenges, and Future Research Direction,” *Sensors*, vol. 22, no. 20, pp. 7896–7896, Oct. 2022, doi: <https://doi.org/10.3390/s22207896>. Available: <https://www.mdpi.com/1424-8220/22/20/7896>. [Accessed: Dec. 15, 2025]
- [3] U. Ahmed *et al.*, “Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering,” *Scientific Reports*, vol. 15, no. 1, Jan. 2025, doi: <https://doi.org/10.1038/s41598-025-85866-7>. Available: <https://www.nature.com/articles/s41598-025-85866-7>. [Accessed: Dec. 15, 2025]
- [4] Q. M. Alam, V. Kolar, and M. Thottan, “Towards AI/ML-Driven Network Traffic Engineering,” *Proceedings of the 4th International Conference on AI-ML Systems*, pp. 1–8, Oct. 2024, doi: <https://doi.org/10.1145/3703412.3703436>. Available: <https://dl.acm.org/doi/10.1145/3703412.3703436>. [Accessed: Dec. 15, 2025]
- [5] A. Aluwala, “AI-Driven Anomaly Detection in Network Monitoring Techniques and Tools,” *Journal of Artificial Intelligence & Cloud Computing*, pp. 1–6, Jun. 2024, doi: [https://doi.org/10.47363/jaicc/2024\(3\)310](https://doi.org/10.47363/jaicc/2024(3)310). Available: https://www.researchgate.net/publication/381845747_AI-Driven_Anomaly_Detection_in_Network_Monitoring_Techniques_and_Tools. [Accessed: Dec. 15, 2025]
- [6] A. Azab, M. Khasawneh, S. Alrabae, K.-K. R. Choo, and M. Sarsour, “Network traffic classification: Techniques, datasets, and challenges,” *Digital Communications and Networks*, vol. 10, no. 3, pp. 676–692, Sep. 2022, doi: <https://doi.org/10.1016/j.dcan.2022.09.009>. Available:

<https://www.sciencedirect.com/science/article/pii/S2352864822001845>. [Accessed: Dec. 15, 2025]

[7] T. Bakhshi and B. Ghita, “Anomaly Detection in Encrypted Internet Traffic Using Hybrid Deep Learning,” *Security and Communication Networks*, vol. 2021, pp. 1–16, Sep. 2021, doi: <https://doi.org/10.1155/2021/5363750>. Available: <https://onlinelibrary.wiley.com/doi/10.1155/2021/5363750>. [Accessed: Dec. 15, 2025]

[8] K. Dhanushkodi and S. Thejas, “AI Enabled Threat Detection: Leveraging Artificial Intelligence for Advanced Security and Cyber Threat Mitigation,” *IEEE Access*, vol. 12, pp. 173127–173136, 2024, doi: <https://doi.org/10.1109/access.2024.3493957>. Available: <https://ieeexplore.ieee.org/document/10747338>. [Accessed: Dec. 15, 2025]

[9] A. Dhakad, S. Singh, Mohana, M. Moharir, and Ashok, “Real Time Network Traffic Analysis Using Artificial Intelligence, Machine Learning and Deep Learning: A Review of Methods, Tools and Applications,” pp. 372–378, Oct. 2023, doi: <https://doi.org/10.1109/icssas57918.2023.10331855>. Available: <https://ieeexplore.ieee.org/document/10331855>. [Accessed: Dec. 15, 2025]

[10] S. M. Doğan, M. ALKAN, and A. Koçak, “Detection and mitigation of cyber-attacks in software defined networks using machine learning/deep learning: a systematic literature review, research challenges and future directions,” *International Journal of Information Security*, vol. 24, no. 5, Sep. 2025, doi: <https://doi.org/10.1007/s10207-025-01114-z>. Available: <https://link.springer.com/article/10.1007/s10207-025-01114-z>. [Accessed: Dec. 15, 2025]

[11] E. Edozie, A. N. Shuaibu, B. O. Sadiq, and U. K. John, “Artificial intelligence advances in anomaly detection for telecom networks,” *Artificial Intelligence Review*, vol. 58, no. 4, Jan. 2025, doi: <https://doi.org/10.1007/s10462-025-11108-x>. Available: <https://link.springer.com/article/10.1007/s10462-025-11108-x>. [Accessed: Dec. 15, 2025]

[12] K. Fotiadou, T.-H. Velivassaki, A. Voulkidis, D. Skias, S. Tsekeridou, and T. Zahariadis, “Network Traffic Anomaly Detection via Deep Learning,” *Information*, vol. 12, no. 5, p. 215, May 2021, doi: <https://doi.org/10.3390/info12050215>. Available: <https://www.mdpi.com/2078-2489/12/5/215>. [Accessed: Dec. 15, 2025]

- [13] A. Gohel and K. Popat, "AI and Machine Learning in Cybersecurity," 2025, doi: <https://doi.org/10.2139/ssrn.5103753>. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5103753. [Accessed: Dec. 15, 2025]
- [14] K. A. Dhanya, S. Vajipayajula, K. Srinivasan, A. Tibrewal, T. S. Kumar, and T. G. Kumar, "Detection of Network Attacks using Machine Learning and Deep Learning Models," *Procedia Computer Science*, vol. 218, pp. 57–66, 2023, doi: <https://doi.org/10.1016/j.procs.2022.12.401>. Available: <https://www.sciencedirect.com/science/article/pii/S1877050922024942>. [Accessed: Dec. 15, 2025]
- [15] R. Kanimozhi and P. S. Ramesh, "Deep reinforcement learning-based intrusion detection scheme for software-defined networking," *Scientific Reports*, vol. 15, no. 1, Nov. 2025, doi: <https://doi.org/10.1038/s41598-025-24869-w>. Available: <https://www.nature.com/articles/s41598-025-24869-w>. [Accessed: Dec. 15, 2025]
- [16] S. M. Kasongo, "A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework," *Computer Communications*, vol. 199, pp. 113–125, Dec. 2022, doi: <https://doi.org/10.1016/j.comcom.2022.12.010>. Available: <https://www.sciencedirect.com/science/article/pii/S0140366422004601>. [Accessed: Dec. 15, 2025]
- [17] Dr. N. Katiyar, M. S. Tripathi, M. P. Kumar, M. S. Verma, A. Kumar, and D. S. Saxena, "AI and Cyber-Security: Enhancing threat detection and response with machine learning.," Apr. 2024, doi: <https://doi.org/10.53555/kuey.v30i4.2377>. Available: https://www.researchgate.net/publication/380843146_AI_and_Cyber-Security_Enhancing_threat_detection_and_response_with_machine_learning. [Accessed: Dec. 15, 2025]
- [18] M. I. Khan, A. Arif, A. Raza, and A. Khan, "AI-Driven Threat Detection: A Brief Overview of AI Techniques in Cybersecurity," *ResearchGate*, Oct. 2024. Available: https://www.researchgate.net/publication/391613291_AI-Driven_Threat_Detection_A_Brief_Overview_of_AI_Techniques_in_Cybersecurity. [Accessed: Dec. 15, 2025]

- [19] A. Kumar, “Cybersecurity Threat Detection using Machine Learning and Network Analysis,” *Journal of Artificial Intelligence General science (JAIGS) ISSN 3006-4023*, vol. 1, no. 1, pp. 38–46, Jan. 2024, doi: <https://doi.org/10.60087/jaigs.v1i1.p46>. Available: https://www.researchgate.net/publication/378892626_Cybersecurity_Threat_Detection_using_Machine_Learning_and_Network_Analysis. [Accessed: Dec. 15, 2025]
- [20] M. Landauer, S. Onder, Florian Skopik, and M. Wurzenberger, “Deep learning for anomaly detection in log data: A survey,” *Machine Learning with Applications*, vol. 12, pp. 100470–100470, May 2023, doi: <https://doi.org/10.1016/j.mlwa.2023.100470>. Available: <https://www.sciencedirect.com/science/article/pii/S2666827023000233>. [Accessed: Dec. 15, 2025]
- [21] M. Maddu and Y. N. Rao, “Network intrusion detection and mitigation in SDN using deep learning models,” *International Journal of Information Security*, vol. 23, no. 2, pp. 849–862, Oct. 2023, doi: <https://doi.org/10.1007/s10207-023-00771-2>. Available: <https://link.springer.com/article/10.1007/s10207-023-00771-2>. [Accessed: Dec. 15, 2025]
- [22] M. A. Mahmud, K. T. Hasan, and R. Karim, “Advancements in Machine Learning for Adaptive Intrusion Detection: A Comprehensive Review,” pp. 460–465, Oct. 2024, doi: <https://doi.org/10.1145/3723178.3723239>. Available: <https://dl.acm.org/doi/10.1145/3723178.3723239>. [Accessed: Dec. 15, 2025]
- [23] N. Mohamed, “Current trends in AI and ML for cybersecurity: A state-of-the-art survey,” *Cogent Engineering*, 2023, doi: <https://doi.org/10.1080/23311916.2023.2272358>. Available: <https://www.tandfonline.com/doi/full/10.1080/23311916.2023.2272358>. [Accessed: Dec. 15, 2025]
- [24] N. Mohamed, “Artificial intelligence and machine learning in cybersecurity: a deep dive into state-of-the-art techniques and future paradigms,” *Knowledge and Information Systems*, vol. 67, no. 8, pp. 6969–7055, Apr. 2025, doi: <https://doi.org/10.1007/s10115-025-02429-y>. Available: <https://link.springer.com/article/10.1007/s10115-025-02429-y>. [Accessed: Dec. 15, 2025]
- [25] F. Muheidat, M. A. Mallouh, O. Al-Saleh, O. Al-Khasawneh, and L. A. Tawalbeh, “Applying AI and Machine Learning to Enhance Automated Cybersecurity and Network Threat Identification,” *Procedia Computer Science*, vol. 251, pp. 287–294, 2024, doi:

<https://doi.org/10.1016/j.procs.2024.11.112>. Available:
<https://www.sciencedirect.com/science/article/pii/S1877050924033465>. [Accessed: Dec. 15, 2025]

[26] G. Nalinipriya *et al.*, “Leveraging explainable artificial intelligence for early detection and mitigation of cyber threat in large-scale network environments,” *Scientific Reports*, vol. 15, no. 1, pp. 24662–24662, Jul. 2025, doi: <https://doi.org/10.1038/s41598-025-08597-9>. Available: <https://www.nature.com/articles/s41598-025-08597-9>. [Accessed: Dec. 15, 2025]

[27] A. Omopintemi, I. Ghafir, S. Eltanani, S. Kabir, and M. Lefoane, “Machine Learning for Malware Detection in Network Traffic,” 2023, doi: <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>. Available: <https://bradscholars.brad.ac.uk/server/api/core/bitstreams/fa0fca84-bb1a-4c27-8748-c9b413fdc237/content>. [Accessed: Dec. 15, 2025]

[28] M. A. Paracha, S. U. Jamil, K. Shahzad, M. A. Khan, and A. Rasheed, “Leveraging AI for Network Threat Detection—A Conceptual Overview,” *Electronics*, vol. 13, no. 23, p. 4611, Nov. 2024, doi: <https://doi.org/10.3390/electronics13234611>. Available: <https://www.mdpi.com/2079-9292/13/23/4611>. [Accessed: Dec. 15, 2025]

[29] P. Prasad, M. Tahir, and Jouni Isoaho, “Enhancing Explainability of Artificial Intelligence for Threat Detection in SDN-based Multicast Systems,” *Procedia Computer Science*, vol. 257, pp. 569–574, Jan. 2025, doi: <https://doi.org/10.1016/j.procs.2025.03.073>. Available: https://www.researchgate.net/publication/391166447_Enhancing_Explainability_of_Artificial_Intelligence_for_Threat_Detection_in_SDN-based_Multicast_Systems. [Accessed: Dec. 15, 2025]

[30] Salam, “Anomaly Detection in Log Files Based on Machine Learning Techniques,” *Journal of Electrical Systems*, vol. 20, no. 3s, pp. 1299–1311, Apr. 2024, doi: <https://doi.org/10.52783/jes.1505>. Available:
https://www.researchgate.net/publication/379685087_Anomaly_Detection_in_Log_Files_Based_on_Machine_Learning_Techniques. [Accessed: Dec. 15, 2025]

[31] P. Schummer, A. del Rio, J. Serrano, D. Jimenez, G. Sánchez, and Á. Llorente, “Machine Learning-Based Network Anomaly Detection: Design, Implementation, and Evaluation,” *AI*, vol. 5, no. 4, pp. 2967–2983, Dec. 2024, doi: <https://doi.org/10.3390/ai5040143>. Available: <https://www.mdpi.com/2673-2688/5/4/143>. [Accessed: Dec. 15, 2025]

- [32] M. H. Thwaini, “Anomaly Detection in Network Traffic using Machine Learning for Early Threat Detection,” *Data & Metadata*, vol. 1, pp. 72–72, Dec. 2022, doi: <https://doi.org/10.56294/dm202272>. Available: https://www.researchgate.net/publication/373443452_Anomaly_Detection_in_Network_Traffic_using_Machine_Learning_for_Early_Threat_Detection. [Accessed: Dec. 15, 2025]
- [33] D. Quirumbay Yagual, D. Fernández Iglesias, and F. J. Nóvoa, “A Hybrid Deep Learning-Based Architecture for Network Traffic Anomaly Detection via EFMS-Enhanced KMeans Clustering and CNN-GRU Models,” *Applied Sciences*, vol. 15, no. 20, p. 10889, Oct. 2025, doi: <https://doi.org/10.3390/app152010889>. Available: <https://www.mdpi.com/2076-3417/15/20/10889>. [Accessed: Dec. 15, 2025]
- [34] F. Zhao, “Research on Network Security Threat Detection and Response Mechanism Based on Artificial Intelligence,” pp. 304–307, Feb. 2025, doi: <https://doi.org/10.1145/3728199.3728249>. Available: <https://dl.acm.org/doi/full/10.1145/3728199.3728249>. [Accessed: Dec. 15, 2025]
- [35] M. Thakur, A. Jain, S. Khan, and D. Prashant Sen, “AI-Based Intrusion Detection Systems for Network Security: A Review.” Accessed: Dec. 31, 2025. [Online]. Available: <https://ejournal.rems.co.in/papers/IJSMER20250305.pdf>
- [36] V. Z. Mohale and I. C. Obagbuwa, “Evaluating machine learning-based intrusion detection systems with explainable AI: enhancing transparency and interpretability,” *Frontiers in Computer Science*, vol. 7, May 2025, doi: <https://doi.org/10.3389/fcomp.2025.1520741>.
- [37] E. Fard, M. Soltani, A. H. Jahangir, and S. Ko, “LightIDS: a lightweight neural network-based intrusion detection system,” *The Journal of Supercomputing*, vol. 82, no. 1, Dec. 2025, doi: <https://doi.org/10.1007/s11227-025-08158-x>.
- [38] M.-M. Chen and M.-C. Chen, “Modeling Road Accident Severity with Comparisons of Logistic Regression, Decision Tree and Random Forest,” *Information*, vol. 11, no. 5, p. 270, May 2020, doi: <https://doi.org/10.3390/info11050270>.
- [39] R. Almuhanha and S. Dardouri, “A deep learning/machine learning approach for anomaly based network intrusion detection,” *Frontiers in Artificial Intelligence*, vol. 8, Sep. 2025, doi: <https://doi.org/10.3389/frai.2025.1625891>.

- [40] S. A. Walli and K. Sallam, "Machine Learning for Intrusion Detection: A Reproducible Baseline is All You Need," *Sustainable machine intelligence journal*, vol. 7, May 2024, doi: <https://doi.org/10.61356/smij.2024.77103>.
- [41] V. Shanmugam, R. Razavi-Far, and E. Hallaji, "Addressing Class Imbalance in Intrusion Detection: A Comprehensive Evaluation of Machine Learning Approaches," *Electronics*, vol. 14, no. 1, pp. 69–69, Dec. 2024, doi: <https://doi.org/10.3390/electronics14010069>.
- [42] M. Wang, K. Zheng, Y. Yang, and X. Wang, "An Explainable Machine Learning Framework for Intrusion Detection Systems," *IEEE Access*, vol. 8, pp. 73127–73141, 2020, doi: <https://doi.org/10.1109/access.2020.2988359>.
- [43] S. More, M. Idrissi, H. Mahmoud, and A. T. Asyhari, "Enhanced Intrusion Detection Systems Performance with UNSW-NB15 Data Analysis," *Algorithms*, vol. 17, no. 2, pp. 64–64, Feb. 2024, doi: <https://doi.org/10.3390/a17020064>.
- [44] Y. Salunke, S. Phalke, M. Madavi, P. Kumre, and G. Bobhate, "Fraud Detection: A Hybrid Approach With Logistic Regression, Decision Tree, and Random Forest," *Cureus Journal of Computer Science.*, Jan. 2025, doi: <https://doi.org/10.7759/s44389-024-02350-5>.
- [45] R. Noviana, U. Teknologi, Y. Sleman, I. Yogyakarta, I. Enny, and Sela, "Performance Comparison Random Forest and Logistic Regression in Predicting Time Deposit Customers with Feature Selection," *International Journal of Computer Applications*, vol. 186, no. 16, pp. 975–8887, 2024, Available: <https://www.ijcaonline.org/archives/volume186/number16/noviana-2024-ijca-923548.pdf>
- [46] A. J. Ajuwon, "Ethical Principles & Responsible Conduct in Research," *African Journal of Biomedical Research*, vol. 23, no. SE2, pp. 11–13, Dec. 2020, Available: <https://www.ajol.info/index.php/ajbr/article/view/202659>
- [47] R. M. A. Latif, F. Ullah, U. Raza, L. Mostarda, M. A. Khan, and D. Cacciagrano, "Real-Time Threat Detection Using Stream Analytics and Deep Learning on Network Logs," *2025 IEEE 2nd International Conference on Big Data Science and Engineering (ICBDSE)*, pp. 1–5, Jun. 2025, doi: <https://doi.org/10.1109/icbdse65491.2025.11220098>. Available: https://www.researchgate.net/publication/397275957_Real-Time_Threat_Detection_Using_Stream_Analytics_and_Deep_Learning_on_Network_Logs.
- [Accessed: Jan. 30, 2026]

[48] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, Jul. 2019, doi: <https://doi.org/10.1186/s42400-019-0038-7>. Available: <https://link.springer.com/article/10.1186/s42400-019-0038-7>. [Accessed: Jan. 30, 2026]

Appendix

Source code link:- [Using AI-ML-Network-Threat-Detection-IDS](#)