



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

**MACHINE LEARNING FOR NETWORK APPLICATION
SECURITY: EMPIRICAL EVALUATION AND OPTIMIZATION**

Supervisor: Janvi Rakesh Bhai Mavani

Student ID:2424764

Supervisor Name: Ranuga Jayakumar

Declaration of Originality

I certify that this is a self-written project; all the references to which have been used; I have done all the work of which this was a record, and that the completed work is within the set word limit. This is not a piece work that has been previously accepted in any other degree submission.

Signed: J.R.Mavani

Date: 3/01/2026

FORM OF CONSENT

I am now authorized to approve that my project, which is presented in candidature to receive the MSc degree, should, in the case of success, be placed on the list of inter-library loans or photocopies (in agreement with the copyright law).

Signed: J.R.Mavani

Date:3/01/2026

Copyright Acknowledgement

I understand that the copyright of the given project report, and any product which may be created within the framework of the project, are owned by University of Wales Trinity Saint David, Swansea.

Abstract

The swift development of 6G network architecture and network slicing solutions has brought about new threats in security owing to the growing heterogeneity, dynamism in resource allocation as well as massive integration of Internet of things (IoT) equipment. In such environments, traditional rule-based security mechanisms are becoming ineffective to identify advanced and dynamic cyber threats. In this work, the researchers intend to explore the efficiency of machine learning methods to improve network application security in 6G network slicing systems through the analysis of network traffic data and testing various models of classification.

The quantitative research method was applied with the use of secondary data in a collection of 6G Network Slicing Security Dataset (records of network sessions, metrics of IoT device behaviour, and security-related features). The data pre-processing, exploratory data analysis, feature selection, and data balancing using the Synthetic Minority Oversampling Technique (SMOTE) were used as methodology. The models of several supervised machine learning models such as Logistic Regression, Decision Tree, Random Forest, and Support Vector machine were developed and optimised using hyperparameter tuning. Standard metrics of performance such as accuracy, precision, recall and F1-score were used to assess model performance.

The findings indicate that the optimised and sophisticated machine learning models are much superior to the baseline models in the detection of malicious network traffic. Specifically, the Random Forest model demonstrated the highest overall performance, with the accuracy of 89.44 and the F1-score of 89.02, which is a positive balance between the precision and the recall. The results also indicate that the ensemble-based approach is stronger when it comes to managing the issue of class imbalance and higher-order interactions between features present in network traffic data. The exploratory analysis also shows the existence of clear statistical differences between normal and malicious traffic and supports the applicability of multivariate strategies of learning. The paper provides empirical data on the need to deploy machine learning-driven security models in 6G network slicing networks. The suggested solution has practical implementation in real time intrusion detection and has provided a research direction to future study of scalable and adaptable network security solution.

Tables of contents

CHAPTER 1: INTRODUCTION	9
1.1 RESEARCH BACKGROUND.....	9
1.2 RESEARCH PROBLEM STATEMENT	11
1.3 RESEARCH AIM	12
1.4 RESEARCH OBJECTIVES	12
1.5 RESEARCH QUESTIONS	12
1.6 CONTRIBUTION OF THE STUDY	13
1.7 STRUCTURE OF THE STUDY	14
CHAPTER 2: LITERATURE REVIEW	15
2.1 TRADITIONAL NETWORK SECURITY APPROACHES AND THEIR LIMITATIONS	15
2.2 MACHINE LEARNING IN NETWORK APPLICATION SECURITY	18
2.3 EXPLORATORY DATA ANALYSIS IN NETWORK SECURITY	20
2.4 NETWORK SECURITY DATASETS AND DATA IMBALANCE CHALLENGES	22
2.5 OPTIMIZATION TECHNIQUES FOR ML-BASED SECURITY MODELS.....	23
2.6 OVERVIEW OF ML APPROACHES FOR 6G NETWORK SLICING, SECURITY.....	25
2.7 ANOMALY DETECTION IN DYNAMIC 6G NETWORKS AND IOT ENVIRONMENTS	26
2.8 RESEARCH GAPS IDENTIFIED.....	27
2.9 LITERATURE SUMMARY	29
CHAPTER 3: METHODOLOGY	31
3.1 INTRODUCTION.....	31
3.2 RESEARCH DESIGN.....	31
3.3 RESEARCH MODEL	32
3.3.1 Proposed Architecture.....	32

3.3.2 Proposed Steps.....	33
3.4 RESEARCH DATA ANALYSIS	33
3.5 CHAPTER SUMMARY	34
CHAPTER 4: IMPLEMENTATION.....	36
4.1 SYSTEM ARCHITECTURE	36
4.2 IMPLEMENTATION TOOLS, ENVIRONMENT AND LIBRARIES	36
4.3 DATASET LOADING AND PREPARATION	37
4.4 FEATURE SCALING AND TARGET SEPARATION.....	37
4.5 DATA BALANCING USING SMOTE.....	38
4.6 MACHINE LEARNING MODEL IMPLEMENTATION	40
CHAPTER 5: DATA COLLECTION	43
5.1 DATASET SOURCE	43
5.2 DATASET DESCRIPTION	43
5.3 DATA COLLECTION PROCESS	44
5.4 RELEVANCE TO RESEARCH OBJECTIVES	45
CHAPTER 6: DATA ANALYSIS	46
6.1 EXPLORATORY DATA ANALYSIS (EDA).....	46
6.2 FEATURE SELECTION AND DIMENSIONALITY REDUCTION	52
6.3 DATA BALANCING.....	53
CHAPTER 7: RESULTS AND DISCUSSION	55
7.1 RESULTS	55
7.2 DISCUSSION	60
CHAPTER 8: CONCLUSION AND RECOMMENDATION	64
8.1 OVERVIEW OF THE CHAPTER.....	64
8.2 KEY FINDINGS	64

8.2.1 Results of Exploratory Data Analysis (EDA)	64
8.2.2 Results of Machine Learning Model Performance	64
8.2.3 Optimisation Techniques Effect	65
8.2.4 Security Implications	65
8.3 LINKING FINDINGS WITH RESEARCH OBJECTIVES	65
8.4 CONCLUSION	67
8.5 LIMITATIONS	68
8.6 RECOMMENDATIONS.....	69
REFERENCES	71
APPENDIX	78

List of Figures

Figure 1.1.1: Use AI ML in network security	10
Figure 1.7.1: Structure of the Study	14
Figure 2.1.1: Traditional Cybersecurity Systems	16
Figure 2.2.1: Machine learning approach to cyber threats	19
Figure 2.2.2: Generalised ML and DL Based Network Intrusion Detection System Methodology	20
Figure 2.3.1: EDA shows the proportion of normal traffic and attack traffic	22
Figure 2.6.1: Attacks on each 6G AI/ML security architecture tier are distinct	26
Figure 4.5.1: Class Distribution Before and After SMOTE Balancing	39
Figure 4.6.1: Workflow of Machine Learning Model Implementation and Prediction	41
Figure 5.2.1: 6G_Network_Slicing_Security_DatasetR.csv	44
Figure 6.1: Attack vs Normal Traffic Distribution	46
Figure 6.2: Proportion of Normal vs Malicious Traffic	48
Figure 6.3: Kernel Density Distribution of Key Network Metrics	49
Figure 6.4: Feature Correlation Heatmap	51
Figure 6.5: Latency Distribution	51
Figure 6.6: Pairwise Feature Relationships	52
Figure 6.7: Selected Features and Dimensionality Reduction for Anomaly Detection in 6G Network Traffic	53
Figure 6.8: Class Distribution Analysis	54
Figure 7.1.1: Descriptive statistics	55
Figure 7.1.2: ML Model Results	58
Figure 7.1.3: ML Model Result Comparison	58

ACKNOWLEDGEMENTS

I would like to state that I would like to be thankful to my supervisor, as they guided, supported, and provided constructive feedback during the process of this research. They were very instrumental in influencing this study because of their knowledge and support. I am also appreciative of the faculty and the staff of the department in availing the academic resources and learning environment that is required to complete this dissertation. I would like to note that my peers were very helpful as I was supported and encouraged throughout the difficult steps of this work. Lastly, my family should be immensely thanked because of the unparalleled support, patience, and tolerance they have provided to me and the success of this research possible.

CHAPTER 1: INTRODUCTION

1.1 RESEARCH BACKGROUND

The world has gone more digital, and as such, the number of cybersecurity attacks has increased tremendously as well due to the increased dependence on network-based applications. With the companies still implementing the idea of cloud computing, Internet of Things (IoT), and mobile-based technologies, the number of attack surfaces by the cybercriminals is multiplying, and the application of traditional security systems becomes pointless. Although the signature-based and the rule-based systems have been effective measures in previous years, they are unable to work effectively with the complex and zero-day attacks that exploit the new areas of vulnerability [3]. It is therefore direly needed to devise intelligent and dynamic systems capable of learning and initiating action against dynamic threats in real time. Machine Learning (ML) has developed as an influential method of increasing the level of security in network applications. With the assistance of ML models, algorithms on large amounts of network traffic data can be applied to detect an abnormality, categorise a type of attack, and forecast possible attacks prior to their happening. According to Liu and Ju (2024) [6], it is assumed that the incorporation of ML in the network intrusion detection system relying on unsupervised learning, supervised learning, and deep learning techniques has increased the detection accuracy and decreased the false alarm rate of the system compared to the former technologies. The ML-based systems not only have the capability of recognising the already known attack patterns, but they are also able to recognise the previously unknown malicious patterns, thus the proactive defence mechanisms.

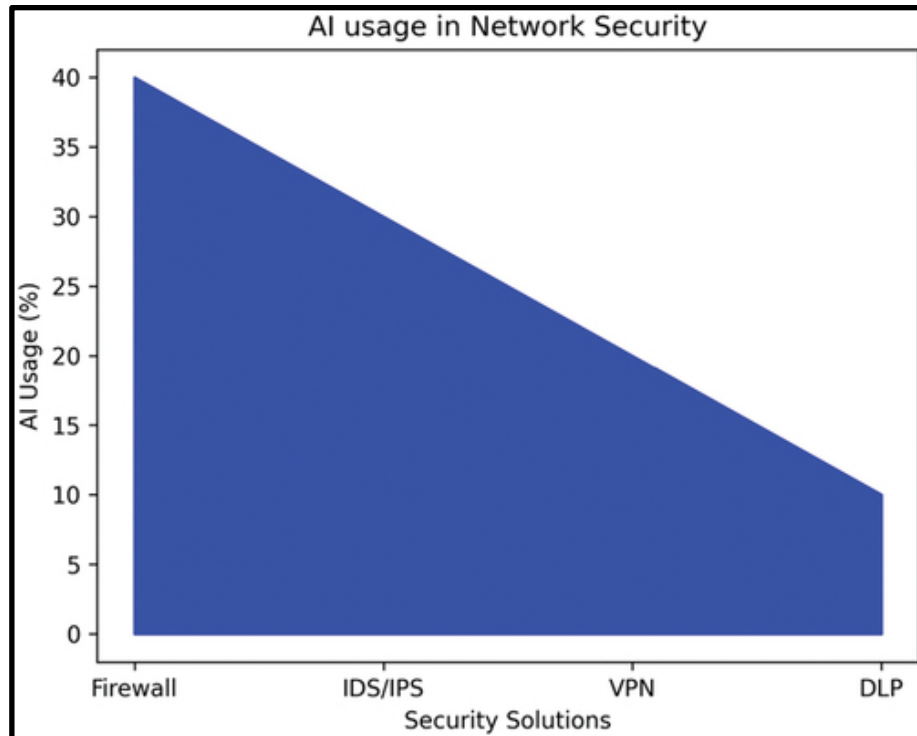


Figure 1.1.1: Use AI ML in network security

It provides the visual presentation of the adoption rate of AI (Machine Learning) on the network security solutions, i.e. firewall, IDS/IPS, VPN, and DLP. The graphic also makes the explanation of the increasing role of ML in the tightening of security of network applications easier, as it indicates the trends of the actual world application. However, using ML in network security is not an easy task. To be more exact, Arp *et al.* (2022) [2] refer to the fact that imbalances in data sets, the inability to be interpretable, and adversarial attacks are the core problems of the majority of security systems based on ML. Moreover, the quality of data, the selection of features and model optimisation will, in most cases, be crucial to the functionality of the ML models. In these respects, the effective an effective empirical assessment and optimisation procedure is needed to make the ML models fruitful and sensible in the practical networks. Conclusively, it stated that machine learning is slowly acquiring importance in improving the security of network applications, and this is the case depicted by the research background. It focuses on the need to conduct empirical evaluation and maximise the performance of ML models to fit current problems, improve accuracy, and scale to protect current network infrastructures.

1.2 RESEARCH PROBLEM STATEMENT

The growing complexity of cyber-attacks like zero-day attacks and advanced persistent threats has revealed the vulnerability of the traditional network security controls that are much more focused on fixed rules and signature-based sensing. Machine learning is evolving and is confronted by the challenges of unequal data, overfitting and failure to foresee unknown risks. In addition, adversarial attacks pose a threat to the accuracy and reliability of detection, further undermining the accuracy and reliability of the relevant ML models. Therefore, the urgency is that optimised and empirically verified ML models will be created to improve the functionality of real-time network applications in security aspects [5]. Moreover, since the use of Machine Learning (ML) as an approach to network security is gaining momentum, there are no large amounts of high-quality and real-world training examples, the traffic is highly skewed, the dynamics of threats are shifting, and attackers have already learned how to avoid the detection mechanisms based on ML [4].

Here, there is high demand to develop a research framework, capable not only of carrying out empirical research to assess the performance of the ML-based security models under the constraints of realistic traffic of the type of the network applications, but also optimize it with the help of a variety of strategies related to the selection of features, hyperparameter and adversarial-robust training as well as being more adaptive to the new threats. The proposed study is a machine learning architecture based on Exploratory Data Analysis (EDA) implemented to handle the network traffic and visualise the network traffic, and detect significant patterns of attacks before training the network [1]. Using the EDA insights, the data is optimised based on features and balancing, which assists in attaining increased model accuracy and fewer false positives. Minimised ML model increases real-time processing and responsiveness to changing network attacks. The study will seek to compare and contrast the performance of the different ML algorithms, the classical and the neural network models, in detecting and preventing network attacks. This is supposed to determine the most effective algorithms among the choices of routine and malicious network traffic, and the performance of the algorithms with the optimisation by the utilisation of the method of optimisation in the practical network security environment.

1.3 RESEARCH AIM

This research study aims to develop and optimise the machine learning models used to improve the security of network applications. It aims at enhancing identification and prevention of cyber threats through empirical testing of different ML algorithms and their optimisation to enhance their accuracy, flexibility, and reliability in real-time network settings.

1.4 RESEARCH OBJECTIVES

- Obj 1: To conduct the examination of the past literature on the use of machine learning in the security of network applications, and identify a research gap in the current research.
- Obj 2: To do the Exploratory Data Analysis (EDA) of the network traffic data sets to determine the significant trends and filter out the significant features that may be used to train the model.
- Obj 3: To develop and apply pertinent machine learning algorithms capable of revealing and categorising threats to networks.
- Obj 4: To ensure that the developed machine learning models are optimised concerning the feature selection, data balancing and hyperparameter tuning to optimise the performance and minimise the false positives.
- Obj 5: To compare the performance of optimised ML models with the performance of traditional and baseline models based on the traditional evaluation measures of accuracy, precision, recall, and F1-score.
- Obj 6: To provide useful suggestions regarding how a streamlined ML-based security model can be used in a realistic network setup.

1.5 RESEARCH QUESTIONS

RQ1: To what extent are current machine learning methods already applicable in improving the security of network applications, and what are the current limitations of the methods?

RQ2: How to implement the Exploratory Data Analysis (EDA) to reveal the key properties of the network traffic and use them to enhance the work of machine learning models?

RQ3: What machine learning algorithms will be best suited, in the sense of accuracy in the separation of the normal network traffic and the abnormal (malicious) network activities?

RQ4: What machine learning algorithms are most accurate and sound in the detection and categorisation of cyber threats in a network application?

1.6 CONTRIBUTION OF THE STUDY

This study makes several positive contributions to the research in network application security and machine learning:

- **Theoretical Contribution:** Contributes to the existing one by conducting an empirical study to understand the different ML algorithms and how they offer strengths and weaknesses to identifying abnormal activities of a network, and offers an optimisation-based solution that minimises the inaccuracy.
- **Methodological Contribution:** Proposes a hybrid algorithm that is the combination of Exploratory Data Analysis (EDA), feature selection, and optimization algorithms like hyperparameter optimization and data compression to improve the model interpretability and credibility.
- **Practical contribution:** The model will offer an organization a guideline to employ optimized security systems, which are driven by ML, to integrate real-time reaction and detection of cyber threats; therefore, the model will guide professionals in choosing efficient algorithms and optimisation activities.
- **Technology Contribution:** Invented an improved ML prototype which can be adopted into intrusion detection and preventive systems to enhance proactive and adaptive threat detection in the real world.

1.7 STRUCTURE OF THE STUDY

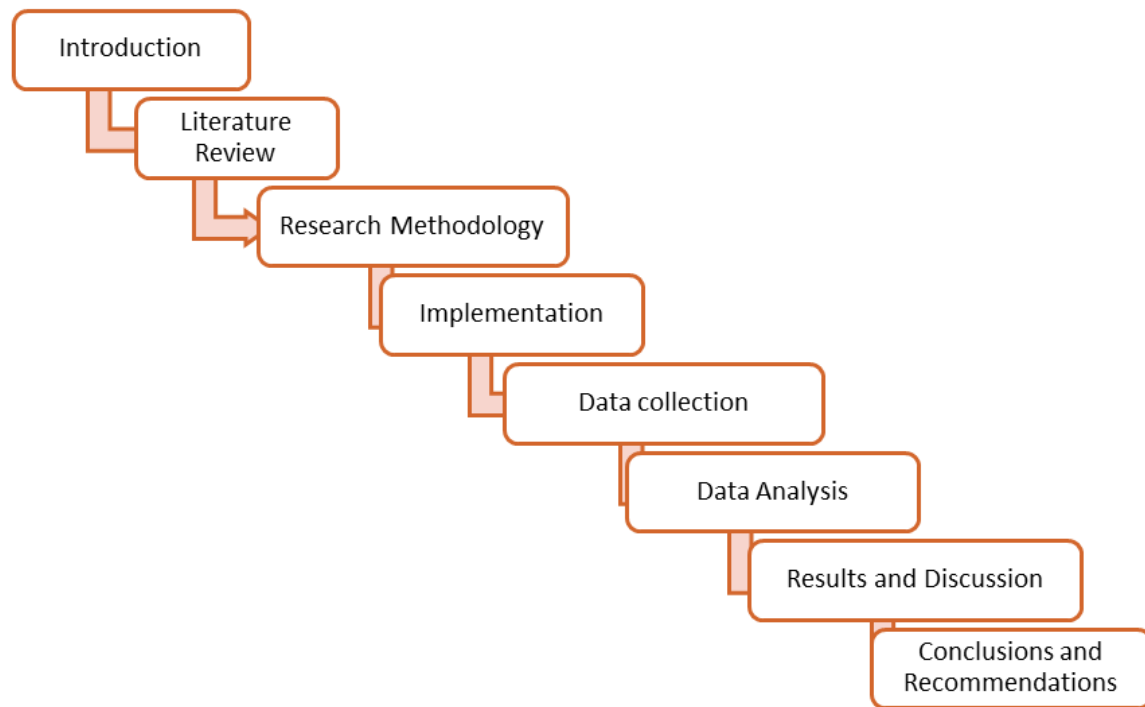


Figure 1.7.1: Structure of the Study

The study is divided into nine chapters, each addressing a major issue in the research. Figure 1.7.1, the flow chart of the structure of the research process, including the introduction and the literature review up to the results and the conclusion. Logically, development is facilitated within the organisation, starting with the theoretical background and proceeding through empirical assessment, model development, and optimisation. Each of the chapters will be part of the overall achievement of the research aim and objectives.

CHAPTER 2: LITERATURE REVIEW

The growth of network-based applications, cloud computing, Internet of Things (IoT) and industrial communication systems is enormous; all these have significantly compounded the complexity of contemporary network infrastructures. It could have contributed to efficiency and connectivity but on the other hand they have also created fresh security vulnerabilities that can be exploited by malicious organizers. Even the simple intrusion attempts that could be termed as cyber-attacks have evolved to be high-level attacks such as zero-day attacks, ransomware, and advanced persistent threats (APTs). It has been observed that such attacks bypass the conventional security solutions causing high financial losses, data leakages and a blockage of businesses. Network security tools like firewalls, signature-based intrusion detection systems (IDS) have been widely applied to protect network applications. However, the mechanisms are highly reliant on set regulations and prior attack signatures which do not work effectively with unfamiliar and dynamic attack patterns. To eliminate these limitations, machine learning (ML) has emerged as a promising approach to enhancing network security as it enables systems to learn by data, discover advanced pattern recognition and adapt to new threats. Even though there is an increasing usage of ML-based security solutions, several challenges have not been resolved. Limitations of real-world datasets of high quality, asymmetric network traffic, overfitting of models as well as susceptibility to adversarial attacks are these. The chapter is a literature review of the literature found on conventional approaches to network security, machine learning and deep learning approaches to intrusion detection, exploratory data analysis (EDA) of data, issues of data dataset, optimization mechanisms, and model comparison that machine learning approach implements. The chapter has ended by identifying research gaps that present the need of the proposed study.

2.1 TRADITIONAL NETWORK SECURITY APPROACHES AND THEIR LIMITATIONS

Network security is a practice that is not new and has been associated with the provision of assurance with regard to the fact that the network applications are not tampered with by unauthorised persons and also are not influenced by malicious programs. The most used are firewalls, access control lists (ACLs), antivirus systems and signature-based Intrusion Detection Systems (IDS). The systems are largely based on pre-established rules, known attack signature and consequently only useful in responding to the already known attack but much less effective against

a new and unknown attack pattern [8]. Signature IDS is based on a database of known signatures of the attacks that is continuously updated. Although this is a high accuracy technique that is utilized in identifying an attack that is known attacks, the technique is unable to classify zero-day attacks and polymorphic malware that cannot be compared to the already existing signature. The extent and sophistication of traffic on the network make the maintenance and updating of signature databases more challenging, and the detection rate and false negatives difficult to increase [9]. Security systems that are rule-based are also an issue in terms of scalability and flexibility. These systems are manually configured and the rules have to be given by hand and expert knowledge is needed to generalise to other network settings. The rule-based systems can spur lots of false alarms in huge and rapid web applications and flood security specialists with information, which ultimately reduces the effectiveness of incident responsiveness systems [10].

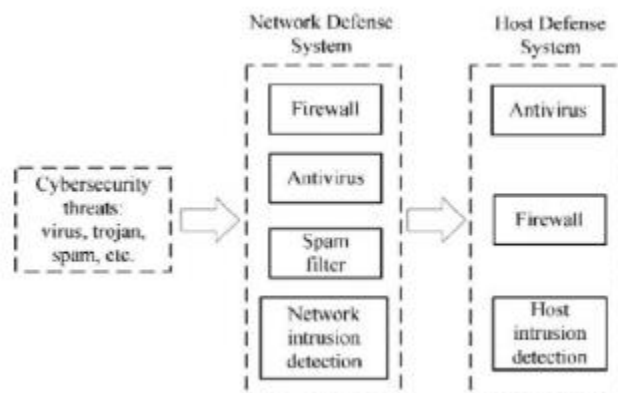


Figure 2.1.1: Traditional Cybersecurity Systems

The figure illustrates the conventional cybersecurity infrastructure, which is built on perimeter-based firewalls and a signature-based intrusion detection system, which implies that it is highly static and unable to adjust to the dynamic and advanced cyber threats [13]. Its conventional systems are not intelligent enough to learn about the traffic or dynamically respond to such mischievous attacks and thus cannot be used in the modern network security demands [11].

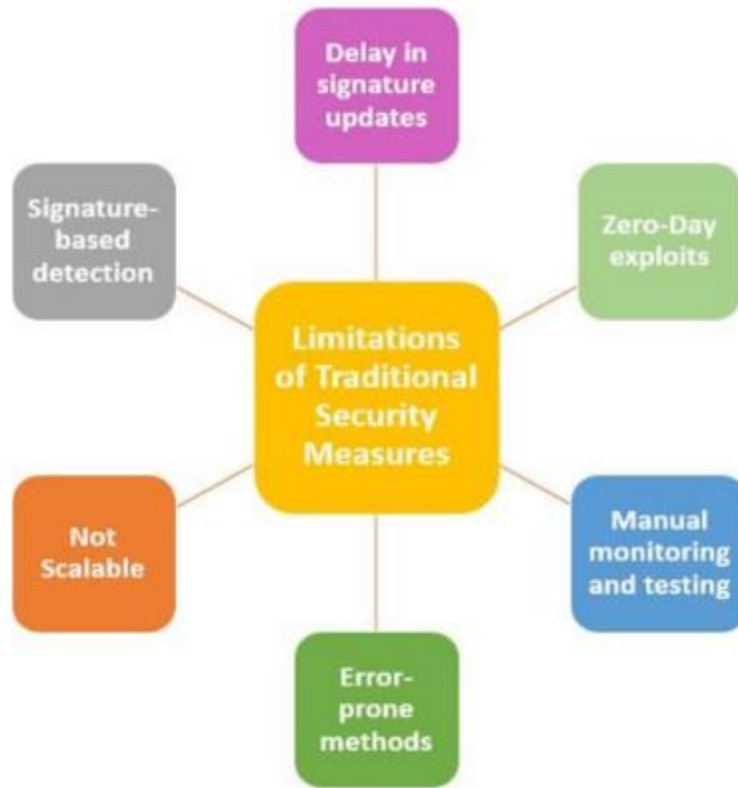


Figure 2.1.2: Limitations of Traditional Network Security

The figure indicates that the conventional security controls have a significant number of limitations, namely, the rigid-rule-based systems like firewalls and signature detection possess issues with detecting the unknown or novel threats, high false positives, and inability to be dynamic in responding to the advanced and AI-driven intrusions [14]. Moreover, the conventional security systems tend to study the network traffic independently, without examining the temporal or behavioural patterns as well. This disjointed form of analysis restrains them from identifying coordinated or low-frequency attacks, which span an extensive period of time. This means that the conventional network security solutions cannot offer real-time network environments with end-to-end security in the environment [12].

These are the natural constraints of the traditional network security systems that have led to the need to have smarter, more flexible and information-based systems. Security models using machine learning will be able to overcome these weaknesses and enhance the quality of detection

and the false alarm rate since they are capable of learning and detecting sophisticated patterns and reacting to novel threats

2.2 MACHINE LEARNING IN NETWORK APPLICATION SECURITY

Machine Learning (ML) has become a significant enabling technology to make the network applications smarter, adaptive and data-driven in terms of detection systems. In contrast to the traditional systems that are rule-based, the ML models can be trained on the network traffic information and can identify known attacks and those that are previously unknown. It is this fact which makes ML particularly helpful in fighting contemporary cyber threats, such as zero-day attacks and advanced persistent threats [15].

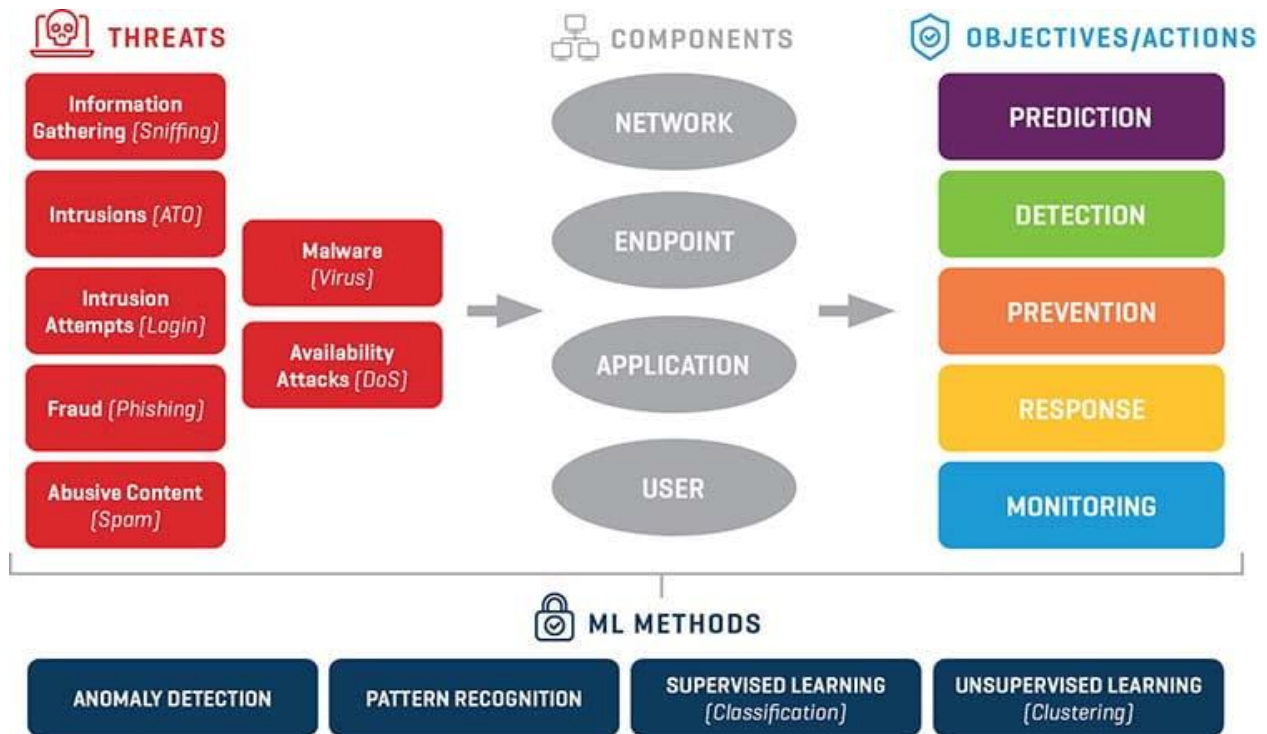


Figure 2.2.1: Machine learning approach to cyber threats

The figure shows that machine learning techniques may be applied to classify and detect any type of cyber threat by monitoring network infrastructure, endpoints, applications, and users in order to achieve improved identification and response to sophisticated and emergent cyberattacks [19].

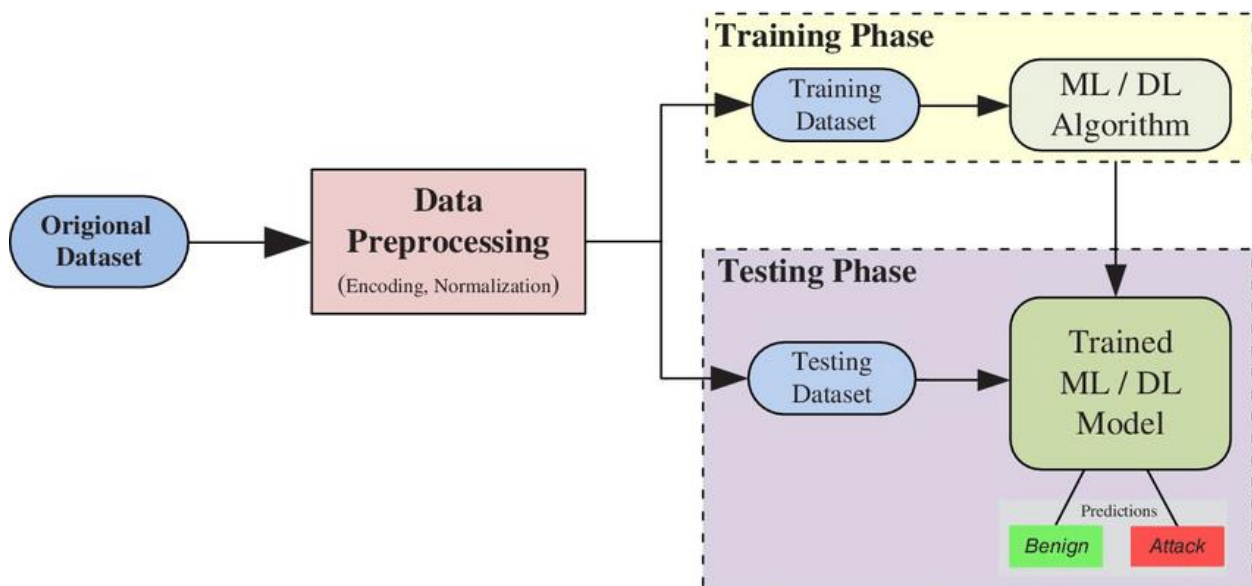


Figure 2.2.2: Generalised ML and DL Based Network Intrusion Detection System Methodology

The diagram illustrates a slimmed-down machine learning and deep learning-centred intrusion detector model, with the network traffic being collected and processed into features, and a model is then trained before the actual classification of normal and malicious network traffic is done [20]. Intrusion detection and traffic classification applications of network security can be supervised using machine learning algorithms. Separation of normal and malicious activities was highly successful with such algorithms as Support Vector Machines (SVM), Decision Trees, Random Forests and k-Nearest Neighbors (k-NN). They are label-based and can be highly accurate with the given representative and well-balanced training data [16]. They, However, are ineffective in encountering new attack patterns that were not presented during the training data. The clustering and the anomaly detection methods are the unsupervised learning techniques that are employed to aid in the detection of unknown or rare attacks in the absence of labeled data. These are approaches that simulate the normal network behavior and point to inconsistencies as possible threats. The disadvantages of unsupervised techniques are that the techniques tend to exhibit high false-positive rates, but work well in the identification of zero-day attacks, since the network traffic is dynamic and defining what is normal behavior is a challenge [17].

Recent research has explored hybrid machine learning methods that adjust a balance between supervised and unsupervised that requests superior detection performance and strength. These hybrid models utilize the use of labeled data during the recognition of known attacks, but use anomaly detection during the recognition of new threats. These techniques increase flexibility, but they increase computational requirements and need to be optimized carefully to be a useful component in real time systems [18]. Despite the advantages that the security models founded on ML have, these security models also present several challenges, including the unbalanced data, redundancy of features, overfitting, and adversarial attack susceptibility. These problems indicate that data preprocessing, selecting the features, and optimisation of the model must be executed well to achieve stable performance in the circumstances of the actual network implementation.

2.3 EXPLORATORY DATA ANALYSIS IN NETWORK SECURITY

The Exploratory Data Analysis (EDA) is an essential tool in the preparation of network traffic data to be used in security models based on machine learning. It is a logical research, forecast and

generalization of the information that attempts to find the trends behind it, anomalies, associations and the quality of information. Network security EDA can be used in finding the nature of network traffic like imbalance of classes, feature distributions and other suspicious trends. The EDA techniques incorporate a few of the statistical summaries, histograms, boxplots, scatter plots and correlation matrices. In one of them, it is possible to use the method of feature correlation to determine the redundant features or the highly correlated features that are not required to be present in the model to reduce dimensionality and computation time without causing any loss in prediction. The visualations, including time-series plots, may be used in order to emphasize the trend of traffic and the time feature of attacks and consequently contribute to the early detection of abnormal traffic [21].

Pie chart distribution for normal traffic and attack traffic

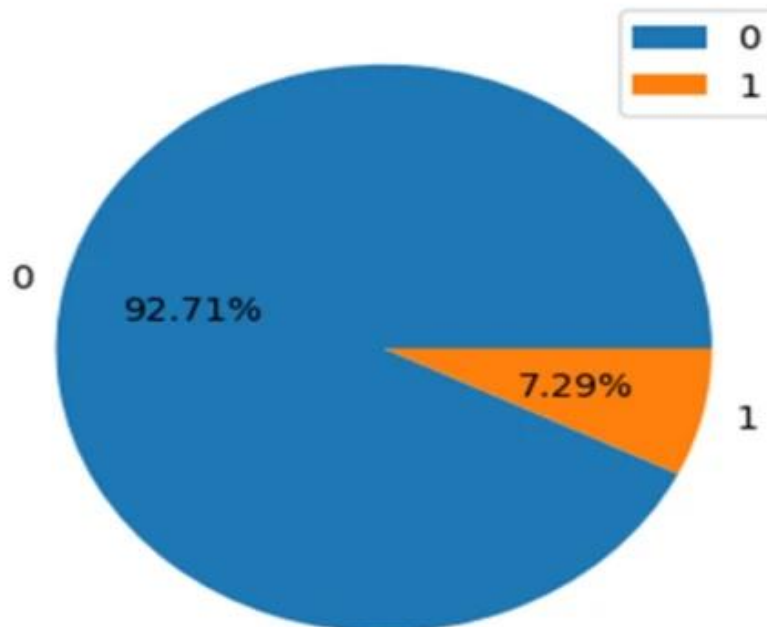


Figure 2.3.1: EDA shows the proportion of normal traffic and attack traffic

The value is evident to display the imbalance in the class in the UNSW-NB15 dataset, where normal traffic was dominant (92.71) and the attacks only 7.29% [24]. It is directly associated with the discussion of the EDA section that discusses the problem of the data imbalance, feature analysis and preprocessing of data to make intrusion detection under the assistance of ML. The datasets of the network traffic are highly skewed, where normal traffic is more than the malicious traffic. These imbalances are also measured by EDA and recommend utilising the approaches of data balancing that include oversampling, undersampling and Synthetic Minority Oversampling Technique (SMOTE). The preprocessing stages will make sure that machine learning algorithms are trained efficiently, such that the rates of accuracy and false positives are lowered [22]. This discovery plays a crucial role in the design of the optimal ML models as far as it is applicable in the real time network security to assure that the pertinent features are considered and the irrelevant or noisy ones are removed [23]. EDA improves the interpretability and performance of a model because it combines statistical analysis with visual exploration. It constitutes the basis of the further operation of network security modeling; features selection, hyperparameter optimization, and adversarial-robust training.

2.4 NETWORK SECURITY DATASETS AND DATA IMBALANCE CHALLENGES

2.4.1 Common Network Security Datasets

The importance of benchmark datasets to network security research is that they are used to train and evaluate machine learning models. Popular datasets contain UNSW-NB15, CICIDS2017 and KDD Cup 1999 which contain assigned examples of normal and suspicious network traffic. The various types of attacks that such datasets are aware of are denial-of-service attacks, probing and application-layer attacks and in the process, the supervised learning techniques can detect threats [25].

2.4.2 Data Imbalance Challenges

The challenge with these kinds of datasets will be related to the fact that the number of classes with normal traffic is significantly outnumbered by the malicious ones. According to the data used in the UNSW-NB15, normal traffic is about 92.7 percent and attack traffic is about 7.3%, which is skewed to the majority. Otherwise, this asymmetry can lead to the problem of poor detection rates of uncommon and significant attacks and high performance in normal traffic [26].

2.4.3 Approaches to Address Imbalance

In order to lessen these issues, data researchers use data preprocessing algorithms and data balancing. Other techniques that can be used to supplement the representation in the minority attack classes are Oversampling, Undersampling, and “Synthetic Minority Oversampling Technique (SMOTE)”. Similarly, feature selection, outlier removal and normalisation provide better training of the model and thus improved accuracy and strength of the model in real-time detection systems of intrusion [26].

The knowledge of the dataset composition and class imbalance is important to the creation of a robust machine learning based intrusion detection system that will be capable of identifying the common attacks and the abnormal attacks.

2.5 OPTIMIZATION TECHNIQUES FOR ML-BASED SECURITY MODELS

The machine learning (ML) models are to be optimised to provide more security to the network applications, as well as the detection accuracy, false positives and ability to adapt to the dynamic nature of cyber threats. Some of the optimization techniques are the feature selection, hyperparameter tuning, metaheuristic ensemble techniques and adversarial robustness techniques.

Feature Selection and Dimensionality Reduction

High-dimensional network traffic data can include redundant or non-beneficial features that can destroy the model's performance and escalate the cost of running. Optimality of feature selection would decrease the dimensionality, improve interpretation, and generalization. According to Vitorino et al. [27], the strong feature selection procedures have no adverse effects on intrusion detection system adversarial attacks and resilience with no effect on the detection accuracy of the system.

Hyperparameter Tuning

The definition of the ML models is significant in order to have the best performance in terms of hyperparameter tuning. Mechanisms like grid search, random search and automated optimization frameworks are applicable, whereby procedures of systematic searching of the model parameters can be undertaken to present the best metrics like accuracy, precision and recall. Tripathy and Behera [28] demonstrated that hyperparameter optimization has an enormous push effect on the

intrusion detection performance compared to the default setting, false alarm suppression and enhanced responsiveness.

Metaheuristic and Ensemble Optimization

The Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) are the most common metaheuristic algorithms that are applied to feature selection and jointly optimize model parameters. According to Francis *et al.* [30], swarm intelligence hybrid models that apply deep learning classifiers can work with higher accuracy and with lower computational efforts in intrusion detection when working in complex Industrial IoT setup.

Adversarial Robustness Optimization

Adversarial attacks are one of the threats to the security systems based on ML. Huang *et al.* [29] reported that adversarial training, whereby the models are subjected to a polluted input enhances the robustness but has no effect on the performance when using the models on the normal network traffic. The feature and hyperparameter optimization, as well as adversarial training are the introduced features to enhance the reliability and resilience.

The machine learning-based intrusion detection systems could be of high accuracy, customizable and a real-time application, and thus, it is effective in the present network security environment with the feature selection, hyperparameter optimization, metaheuristic optimization and adversarial robustness.

2.6 OVERVIEW OF ML APPROACHES FOR 6G NETWORK SLICING, SECURITY

The advent of the 6G networks and the increased number of devices that are activated through the IoT have also provided more difficult security issues, especially in the dynamically sliced network infrastructure. The network slices might be appropriated in the use of the application that needs various quality of service and thus the conventional security systems might not be adequate. The fact that machine learning (ML) has been applied to process high-dimensional traffic data, identify outliers, and respond to the changing threats in real-time is therefore of interest. Supervised learning would tend to be used in identifying known cyber threats within the 6G networks, which are founded on IoT. The algorithms used in the classification of the normal and malicious traffic include the Random Forests, SVM and Neural Network algorithms applied on labelled traffic data. The advantages of such models are that, it is very precise when trained on representative data and can be utilized in the detection of such attacks as unauthorized access, data exfiltration, or denial of service attacks [31].

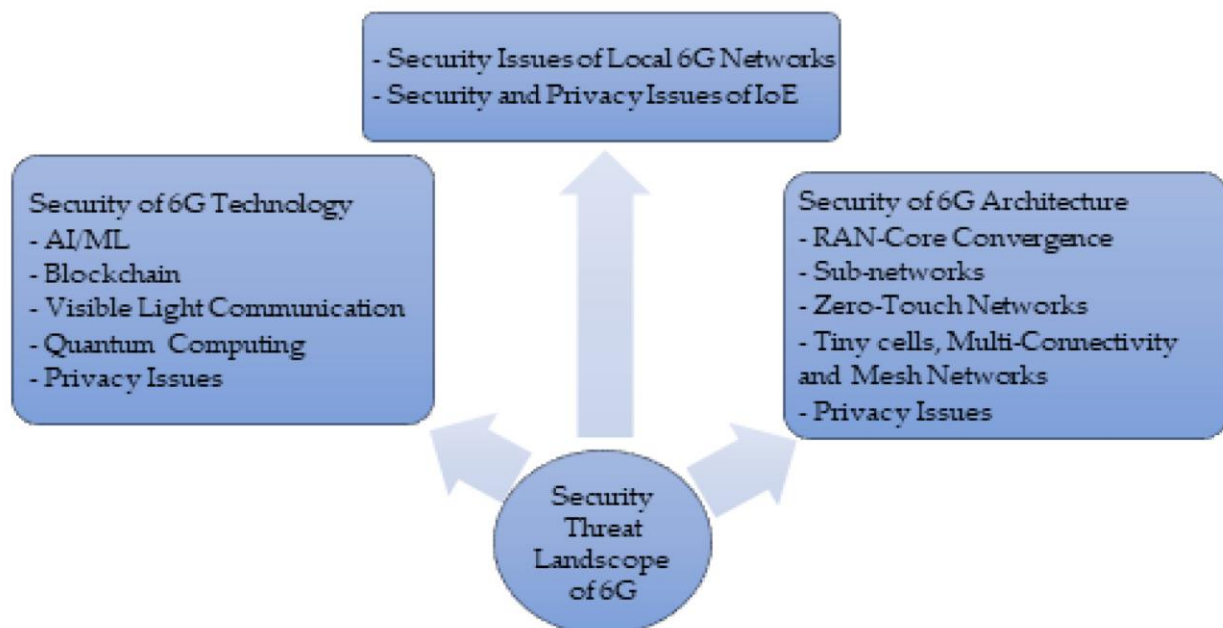


Figure 2.6.1: Attacks on each 6G AI/ML security architecture tier are distinct

The figure displays a machine learning workflow of 6G network anomaly detection, through which the IoT traffic is received, processed, and analysed with the assistance of the ML models to detect an anomaly, which facilitates the safe operation of network slices and a responsive reaction to the threats in changing 6G settings. The unsupervised learning methods would be needed to detect the new or zero-day attacks such as clustering and anomaly detection methods. These models are trained by utilizing typical service offerings of behavior patterns that are discovered within the IoT network traffic and tag possible security attacks. The latter include the K-means clustering, autoencoders, and isolation forests to name but a few of the methods that were effectively incorporated to recognize the presence of anomalies in the 6G network slices as one of the ways of addressing the risk of undetected attack vectors [32]. Ensemble-based models and hybrid models are a combination of the two paradigms of supervised and unsupervised learning, that is aimed at improving the accuracy of detection and resilience. Ensemble techniques can use multiple algorithms that can be stacked together in order to generate a single performance, i.e. stacked classifier or voting systems and can eliminate false positives and yield a strong detection performance using heterogeneous IoT devices.

In spite of the efficiency, the use of ML models in the 6G network security is tainted with such difficulties as the high-dimensionality of feature space, the imbalance of the classes and adversarial vulnerability. They are customized models that can be optimized via feature selection, hyperparameter optimisation and adversarial training so that the real time adaptability, scalability and resilience of intrusion detection in next generation networks will be guaranteed [33]. The ML-based solutions overall offer a flexible and customizable design of the protection of the 6G network slices and 6G IoT ecosystems and allow a post-active detection of threats and the network activity support in a smart and strong way.

2.7 ANOMALY DETECTION IN DYNAMIC 6G NETWORKS AND IOT ENVIRONMENTS

The anomaly detection is a crucial component in defining the abnormal and possibly malicious activity within the dynamic 6G network environment that is characterized by extremely high data rates and low latency and heterogeneous services such as IoT and network slicing. The network

slices are distributed and coordinated dynamically; therefore, the security is dynamically changeable which is, the one that is able to detect the familiar and unfamiliar anomalies [34]. The fact of the matter is that the ML and DL methods have been extensively applied in the said aspect since their learning abilities are capable of attaining intra-dimensional patterns so long as they are provided with high-dimensional traffic data. The random forests under watch models and SVM have good application on the labeled traffic and the attacks are correctly identified on the applications of these models.

Autoencoders and clustering are unsupervised and could be applied to identify abnormal attacks on the basis of the abnormal behaviour of normal traffic. It is noted in the recent past that the hierarchical nature of 6G traffic may be attained through multi-scale convolutional autoencoders that will improve precision of recognizing dynamic as well as heterogeneous slices. As well, the federated learning-based solutions are able to facilitate distributed detection of anomalies and protect the privacy of the users, which is not susceptible to the zero-day attacks across network slices [35]. Simply put, having such a powerful, and scaling security of a 6G network implies that an adaptive ML/DL model requires real-time preprocessing and feature engineering to detect and authenticate anomalies in 6G networks.

This issue of 6G and IoT worlds is urgent and requires an imbalance and heterogeneity of data. The IoT network traffic is typically not comparable in speed and traffic and the attacks are usually small in number when compared to the normal IoT networks. It has also been demonstrated that such a lack of consideration of such imbalances can make the models lean towards normative traffic existence and reduce the rate at which such attacks are detected and augment the rate at which a false negative can result [36]. Oversampling, SMOTE and weighted loss functions are used to overcome this problem. Overall, to ensure the successful anomaly detection of dynamical 6G networks and IoT settings, it is necessary to use simplified machine learning models, adequately process data, and flexible algorithms that could react to arising cyber threats. Using these models, it is possible to create a better defence of the future heterogeneous and high-speed networks by means of traffic and device behaviour information.

2.8 RESEARCH GAPS IDENTIFIED

Although a lot has been done in terms of machine learning-based network security, the literature available presents a number of critical research gaps that need to be addressed through research. First, many of the previous works are centered around either a significant share of traditional network environment or legacy data, like KDDCup99, NSL-KDD, or CICIDS, which are insufficiently representative of the dynamic, heterogeneous, and high-speed character of the 6G network and IoT environment. This means that the applicability of these models to new network designs has not yet been fully realized.

Second, although machine learning solutions have demonstrated themselves as effective in identifying known attack signatures, there has been a significant gap when it comes to identifying zero-day and other emerging cyberattacks, especially in network slicing contexts. Most of the current frameworks used extensively depend on supervised learning methods that need a large amount of labelled data which is either not available or expensive to acquire in 6G and IoT applications. Such dependence decreases flexibility to new or sneak attacks.

Third, even though Exploratory Data Analysis (EDA) is a crucial preprocessing phase, it is often disregarded as a superficial phase in research. The absence of systematic incorporation of EDA-based findings into feature selection, data balancing, and model optimization steps, in particular, with complex datasets of IoT device behavior and slice-specific traffic indicators, is present. Due to this, models can exhibit poor performance and increase false-positive rates.

The other important gap is the managing data imbalance and heterogeneity. Most intrusion detection researches recognize the imbalance effects but use generic balancing methods that have not been tested empirically to estimate the effects on real-time detection. This weakness may greatly compromise the reliability of detection in dynamic 6G environments in which normal traffic by far surpasses malicious activity.

In addition, the optimization techniques of hyperparameter tuning, adversarial hardening, and real-time adaptability are usually considered separately. Limited empirical studies exist that, in an overall manner, consider optimized ML models in realistic network conditions, such as latency limits and a constantly evolving traffic as characteristic of 6G networks. Lastly, the literature of studies that offer practical deployment considerations is limited and serves as a fashionable transition between theoretical assessments and practice. To fill these gaps an integrated research

framework, which integrates EDA, optimized machine learning methods and current security datasets in dynamic 6G and IoT environments is needed, which this research seeks to add to.

2.9 LITERATURE SUMMARY

The literature reviewed proves that the concept of network security has developed considerably moving out of the traditional rule-based and signature-driven systems where the data-driven and intelligent systems have gained more relevance. Such traditional security systems like firewalls and signature-based intrusion detection systems are only useful with known attacks but have proven ineffective in identifying sophisticated, changing and zero-day attacks. This shortcoming has made a lot of research on machine learning-based methods of securing modern network applications.

The literature also emphasizes the usefulness of both traditional machine learning methods, such as Support Vector Machines, Random Forests, and k-Nearest Neighbors, and deep learning architectures, such as Convolutional Neural Networks and Long Short-Term Memory networks in malicious network traffic detection. These methods have proven to be more accurate in detection and flexible than conventional methods. However, the literature also reveals that there still exist long-standing weaknesses in terms of overfitting, large levels of false-positives and weaknesses when subjected to novel attack patterns.

Exploratory Data Analysis is generally recognized as one of the most important steps to analyze the properties of network traffic and prepare the data to be model-trained. The previous literature supports the effectiveness of EDA and feature engineering as tools to improve the performance of models through noise reduction, discovery of significant attributes, and the imbalance in the data. However, not a lot of research may be presented with a systematic incorporation of EDA understanding into the entire machine learning optimization procedure.

The literature also indicates that such challenges as class imbalance, high dimensionality, and traffic heterogeneity are all significant hindrances and they are especially relevant in the fields of IoT and new 6G network platforms. Although there are a number of different optimization methods (hyperparameter optimization, feature selection, ensemble learning, etc.) that have been suggested, they are frequently restricted to controlled experimental cases. According to recent works, the significance of anomaly detection in dynamic network slicing environments, where traffic behavior evolves quickly, is becoming increasingly important. Furthermore, the literature

highlights the importance of empirically validated, optimized, and adaptive machine learning structures that capable of securing contemporary network applications, which will form the basis of the proposed study.

CHAPTER 3: METHODOLOGY

3.1 INTRODUCTION

This chapter summarises the methodological framework that would be used to meet the research objectives of the present study, which aims to investigate the application of machine learning techniques to improve the network application security in 6G network slicing environments. The research method is quantitative with the use of secondary data and a 6G Network Slicing Security Dataset, which consists of the records of network traffic, network behaviour measurements of IoT devices, and features related to security. The data has well-organized numeric information that can be analyzed with statistical tools and can be evaluated using machine learning. Preprocessing of the data and exploratory data analysis (EDA) are the first steps in the methodology to determine the most important trends and patterns in the traffic in the network as well as important features. Subsequently, various machine learning models are built and optimised with the help of feature selection, data balancing, and hyperparameter tuning methods. The results of the proposed models are analyzed and compared based on typical figures of accuracy, precision, recall, and F1-score. This systematic methodological practice provides strength, repeatability, and applicability to real-world network security applications.

3.2 RESEARCH DESIGN

The study will be conducted in a quantitative research design to provide a systematic analysis of the network traffic data to determine the effectiveness of machine learning tools in detecting network application security. Quantitative designs are especially adequate in studies that involve the use of numerical data and statistical analysis and algorithmic modelling to uncover patterns and relationships in large data sets [37]. This method allows to objectively measure the performance of any system and the reproducibility of results by means of structured analytical processes that are performed with Python-based tools [38]. The research relies on secondary data, which is based on an existing 6G Network Slicing Security Dataset, which has been collected with the network traffic data, metrics of monitoring IoT device behaviour and security-related properties. This is because the secondary quantitative data enables the research to concentrate on analytical modelling and the performance evaluation rather than the ethical and logistical limitations involved in using primary data [45]. The main analytical environment used is Python

because it is supported by a large ecosystem of data processing, statistics analysis, and visualisation libraries that can be useful in the large-scale quantitative analysis [39].

The research design is based on the experimental design which is data-driven; it starts with the data preprocessing and the exploratory data analysis to be able to know the statistical property and distribution of the dataset. Exploratory analysis is the technique that reveals the hidden structures and relationships to inform the choice of features and formulate models [40]. Subsequently, several machine learning models are trained and optimised to identify and categorise deviant network behaviour and methodological consistency is guaranteed across experiments [46]. The models are measured in terms of standard quantitative measures, such as accuracy, precision, recall, and F1-score, which allows the objective and clear comparison of the baseline models and optimised ones. This type of experiment design is empirically valid, and it enables performance benchmarking among the various algorithmic methods [47]. In general, the implemented research design allows achieving an analytical rigor, transparency in the methodology, and relevance to real-world applications of network security [42].

3.3 RESEARCH MODEL

3.3.1 Proposed Architecture

The architecture that will be proposed will be a combination of various levels that are linked to one another to enable the detection and classification of cyber threats in network applications. The data acquisition layer is the first one and it makes use of a secondary dataset that includes network session logs, IoT device behaviour, and security related metrics, including latency, bandwidth usage, and access control violation. This layer gives the input data a representation of the real-world network conditions and that is capable of being analyzed in a quantitative manner [45].

The second layer is the layer of data preprocessing and exploratory analysis, here missing values get addressed, categorical variables are coded and numerical features are normalised to provide data uniformity. At this point, the exploratory data analysis is done to extract the statistical trends, correlations and anomalies that are contributing to feature selection and model performance [39].

The third is the feature selection and data balancing layer which helps in dimensionality reduction and class imbalance alleviation to achieve more robust models and fewer false positives [46].

The last layer is machine learning and evaluation layer where several supervised learning algorithms are trained, optimised and evaluated against common performance metrics. This

layered structure allows experimentation of modules and comparisons of baseline and optimised models in an experimental controlled environment [47].

3.3.2 Proposed Steps

A sequence of clear steps is applied in the implementation of the proposed research model in order to make it clear methodologically and analytically rigour. The initial step is the loading and inspection of the dataset to know the structure, types of data and the statistics. This phase will give background information on how the network traffic data is distributed and of what quality [37].

The second step involves data preprocessing such as missing value treatment, censoring of the categorical variables and feature scaling to enable the machine learning algorithms on the dataset. Preprocessing is done to make sure that the models are run on standardised and noise-cleaned data [38]. The third step is exploratory data analysis, which is the application of visual and statistical methods to establish major patterns, correlations, and possible signs of an abnormal behaviour [40].

The fourth step involves feature selection and data balancing, which tries to keep the most informative features and overcome the class imbalance problems which are inherent in cybersecurity data. This is essential in enhancing the accuracy of detection and the minimization of bias in model predictions [46]. The fifth step is to train various machine learning models such as baseline and ensemble-based classifiers and then to optimise the hyperparameters to improve performance [47].

The last step assesses and compares the performance of the models based on quantitative metrics which include accuracy, precision, recall and F1-score, which allows to objectively evaluate the suggested approach. The step aids in the empirical validation and proves the usefulness of the research model in practical network security applications [42].

3.4 RESEARCH DATA ANALYSIS

The data analysis process for this study follows a structured quantitative workflow designed to extract meaningful insights from network traffic data and support the development of machine learning-based security models. The process begins with the acquisition of secondary data from a 6G Network Slicing Security Dataset, which contains numerical and categorical attributes representing network sessions, IoT device behaviour, and security-related metrics. The use of

secondary quantitative data enables efficient large-scale analysis while ensuring consistency and reproducibility of results [45].

The initial stage of data analysis involves data inspection and preprocessing to ensure data quality and suitability for modelling. This includes identifying missing values, handling inconsistencies, encoding categorical variables, and normalising numerical features to bring them onto a common scale. These steps are essential to minimise noise and prevent biased model performance during training [38]. Following preprocessing, exploratory data analysis (EDA) is conducted to understand the statistical characteristics of the dataset. Descriptive statistics, distribution plots, and correlation analyses are used to identify trends, variability, and relationships among features, as well as to highlight potential indicators of anomalous network behaviour [37].

Subsequently, feature selection techniques are applied to reduce dimensionality and retain the most informative variables for model training. This step enhances computational efficiency and improves model interpretability by eliminating redundant or weakly correlated features [39]. Given the inherent class imbalance in cybersecurity datasets, data balancing techniques are employed to ensure that minority attack classes are adequately represented during model training, thereby reducing false positives and improving detection reliability [46].

The processed dataset is then divided into training and testing subsets to enable objective evaluation of model performance. Multiple machine learning algorithms are trained using the training data, and optimisation techniques such as hyperparameter tuning are applied to enhance predictive accuracy and robustness. Model outputs are analysed using standard evaluation metrics, including accuracy, precision, recall, and F1-score, which provide a comprehensive assessment of classification performance [47]. This systematic data analysis process ensures empirical rigour and supports reliable conclusions regarding the effectiveness of machine learning models in network application security [42].

3.5 CHAPTER SUMMARY

This chapter has introduced the methodological framework that will be adopted to address the research goals of the study which is the use of a machine learning method to improve the security of network applications in a 6G network slicing environment. The secondary data of network traffic was used to determine the extent to which objective and reproducible analysis could be conducted by adopting a quantitative research method. The chapter described the research design, the proposed research model, and the process of data analysis, which included the application of

the exploratory data analysis to comprehend the characteristics of the dataset and use them to choose the features. It also detailed the preprocessing activity that was conducted to improve the quality of data such as data encoding, data normalisation and class balancing. Additionally, several machine learning models have been developed, optimised and evaluated, and their performance measured based on conventional measures of accuracy, precision, recall, and F1-score. Moreover, the methodology offers a strong and systematic basis for the empirical analysis and findings given in the next chapter.

CHAPTER 4: IMPLEMENTATION

The chapter presents the implementation of the proposed network security structure, which is based on machine learning. It describes the setup of the system, the software environment, the process of preparing the data, the process of engineering features and the process of creating a model. The chapter shows how the different machine learning algorithms are implemented, optimised and assessed to make the accurate and believable detection of anomalies within the network in dynamical network contexts.

4.1 SYSTEM ARCHITECTURE

The system used follows a systematic way of work that comprises the following steps:

1. **Data Acquisition:** The information on network traffic is acquired in relation to the Kaggle Network Security Dataset, which is an actual network behavior.
2. **Data Pre-processing:** The missing values are also handled and the categorical attributes are coded too, so that machine learning is prepared.
3. **Feature Scaling:** Numerical features are brought to the same level so that they contribute equally during the model training.
4. **Selection of Features:** The features that are selected are chosen using the statistical and model-based feature selection techniques that are aimed at reducing the dimension.
5. **Data Balancing:** The use of SMOTE to even out the imbalance in the number of classes in normal and anomalous traffic is carried out.
6. **Model Training:** It is a machine learning training of different models to detect network anomalies [51].
7. **Model Optimisation:** Hyperparameter optimisation is performed to maximize model accuracy as well as strength.
8. **Comparison and Evaluation:** The measures applied to assess the performance of the models in order to determine the best performance are accuracy, precision, recall and F1-score.

4.2 IMPLEMENTATION TOOLS, ENVIRONMENT AND LIBRARIES

The desired system was executed in the Google Colaboratory (Colab) system with the assistance of Python programming language. Colab has been selected, since it runs in the cloud and is highly

accessible, with data science and machine learning libraries built into it, a limitation of local hardware is not as significant and experimentation can be replicated.

NumPy and Pandas were utilized as the basic numerical data manipulation tools that enable the application of the basic numerical operations and data manipulation in relation to the traffic records of the network. The visualization of data was done in Matplotlib and Seaborn and proved useful in the Exploratory Data Analysis such as class distributions, feature density curves, and correlation heatmaps [52]. The data preprocessing operations that were applied on the data including scaling of features, encodings and division of the dataset were done with Scikit-learn. The features were selected using the method of SelectKBest which is the chi-square method. Many machine learning models were constructed and experimented on the model's suite of Scikit-learn. The imbalance in the classes has been taken into consideration using SMOTE with the imbalanced-learn library and hyperparameter optimisation using the use of GridSearchCV to improve the model performance.

4.3 DATASET LOADING AND PREPARATION

The Network Security Dataset was loaded into the Python space by using the Pandas library to enable its work with the data. The step is directly relevant to the research goal to improve network application security by means of offering structured and labelled network traffic data, which can be fed into machine learning. The median imputation method was employed to substitute the absent numbers in the preparation, and mode imputation was employed to substitute the absent categorical variables with mode numbers and mode labelling, respectively. These pre-processes ensured uniformity in the data, reduction in noise as well as enhanced learning of the models. The quality of dataset preparation will enable sufficient recognition and categorization of both normal and abnormal network traffic, therefore, expand the range of reliability and effectiveness of the proposed machine learning models in real-life network security solutions.

4.4 FEATURE SCALING AND TARGET SEPARATION

The applicable machine learning models in the network security are important with regard to the target separation and scaling of features. The data was categorized under input (X) and the target (y) in this case the label of the target variable, which is the label of anomaly between normal network traffic and malicious activity. This partitioning factor is a direct contribution to the research goal of having the network behaviour duly categorized and also allows the supervised learning algorithms to infer the meaningful decision boundaries among typical and deviant patterns of traffic [53]. The attributes of the input are of heterogeneous network attributes like the latency,

bandwidth consumption, packet loss rate and device-specific attributes, which fall on varying numerical scales. Standards are applied to ensure values of features have a mean of zero and unit variance to ensure that a single feature is not overused to instruct a model.

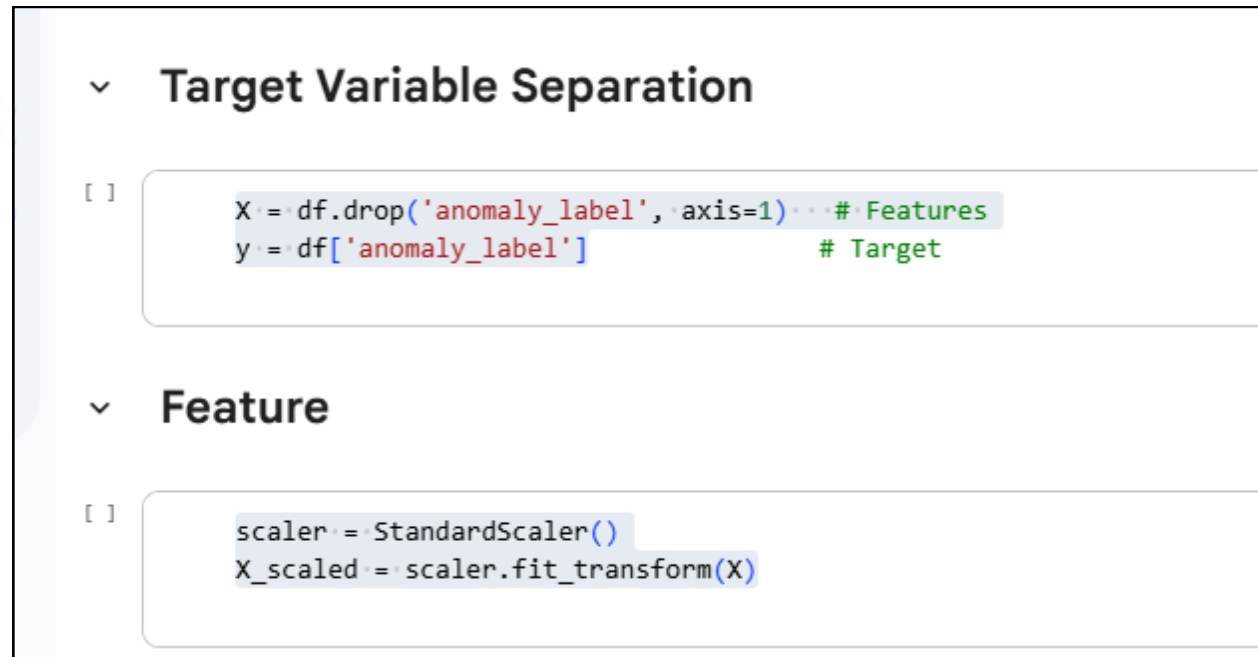


Figure 4.4.1: Feature Scaling and Target Separation for Network Security Dataset

The target variable separation and scaling of the feature along with the capability of the models to learn similar patterns of all the input attributes will make sure that the large number of features will never take over the learning process. This is to make sure that minuscule abnormalities in the network traffic may be detected and this enhances efficiency of classification and minimises false positive that are essential in a practical and real-time application of network security.

4.5 DATA BALANCING USING SMOTE

The characteristic of the datasets on network security is the extreme imbalance of the classes where normal traffic constitutes a huge majority of the dataset relative to the anomalous traffic. This result in an imbalance, biasing the machine learning models to make predictions belonging to the majority class in question, where lesser and more important anomalies are misidentified. “Synthetic Minority Oversampling Technique (SMOTE)” has been applied to work with this problem.

```
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X_selected, y)

print("Before SMOTE:", np.bincount(y))
print("After SMOTE:", np.bincount(y_resampled))
```

Figure 4.5.1: Class Distribution Before and After SMOTE Balancing

In this study, the number of features and targets (X resampled, y resampled) that were chosen in the latter step was resampled using the SMOTE algorithm. The percentage of classes that preceded and succeeded the usage of SMOTE was confirmed in a manner that the percentage of normal and abnormal traffic is equal [54]. With the adoption of SMOTE, the machine learning models can learn the patterns of the normal and abnormal traffic with reliability and aid in the proper and timely identification of aberrations on the network application.

4.6 MACHINE LEARNING MODEL IMPLEMENTATION

The study has applied certain machine learning algorithms in identifying the abnormality of network traffic and has been able to test their effectiveness in the security of network applications. It trained the models on the feature-optimised resampled data to make sure that it learnt appropriately and was able to identify normal and abnormal traffic. Through the convenience and the ease of the Logistic Regression, this was guaranteed to be the baseline model. It possessed a point of reference to compare the more sophisticated algorithms and to test the level of skills of a linear model to differentiate between normal and malicious network sessions. A decision tree classifier was used to learn the non-linear feature/network relationships and interactions [55]. They are not difficult to act in a complicated manner, but the problem of overfitting the decision tree is simply not to be avoided; this is why they were compared to the ensemble-based models.

▼ Baseline Model – Logistic Regression

```
[ ] lr = LogisticRegression()  
    lr.fit(X_train, y_train)  
    y_pred_lr = lr.predict(X_test)
```

▼ Decision Tree

```
[ ] dt = DecisionTreeClassifier(random_state=42)  
    dt.fit(X_train, y_train)  
    y_pred_dt = dt.predict(X_test)
```

▼ Random Forest (Optimised)

```
[ ] rf = RandomForestClassifier(random_state=42)  
  
    param_grid = {  
        'n_estimators': [100, 200],  
        'max_depth': [10, 20],  
        'min_samples_split': [2, 5]  
    }  
  
    grid_rf = GridSearchCV(rf, param_grid, cv=3, scoring='f1')  
    grid_rf.fit(X_train, y_train)  
  
    best_rf = grid_rf.best_estimator_  
    y_pred_rf = best_rf.predict(X_test)
```

▼ Support Vector Machine

```
[ ] svm = SVC(kernel='rbf')  
    svm.fit(X_train, y_train)  
    y_pred_svm = svm.predict(X_test)
```

Figure 4.6.1: Workflow of Machine Learning Model Implementation and Prediction

The ensemble classifier was also a random Forest Classifier that was employed to enhance generalization and strength. The optimization of the Hyperparameter was done with the assistance of the gridsearchcv wherein the number of trees (n_estimators), max depth (max_depth), min

samples per split (min_samples_split) among others were optimized. It was shown that the high-performance random forest is more effective in the detection of anomalies with low false positives. “Support Vector machine (SVM)” that used “radial basis function (RBF)” kernel was also borrowed as it was useful to use high-dimensional feature space and margin-based classification. All the models were trained on the 80: 20 train-test split and tested on standard measures [56]. The models that were implemented were then chosen in order to test and optimize the Network Security Dataset for anomaly detection. They have to determine different trends in the dynamic 6G network traffic and selectively differentiate normal and malicious traffic and enhance the accuracy of the detection of the traffic using simple and ensemble-based machine learning tools.

CHAPTER 5: DATA COLLECTION

The quality and representative data are essential to the success of machine learning in the security of the network application. This chapter outlines the data that was used in this research, the source, the format, the most significant attributes, and its suitability in the detection of anomalies in the dynamic network environment. The primary source of data is the Network Security Dataset, obtained on Kaggle and applied to the implementation into the current context of cyber threat detection.

5.1 DATASET SOURCE

The data used in the present paper were found on the following Kaggle platform:

“**Network Security Dataset**” is an open-source project of a machine learning application and cybersecurity research. This is released under the CC0 (Public Domain) license in which it can be used in academics without any legal restriction.

The data is available at:

<https://www.kaggle.com/datasets/ziya07/network-security-dataset?resource=download>

The presented resource was selected due to its highly detailed description of network sessions, routine and non-routine traffic patterns.

5.2 DATASET DESCRIPTION

Network Security Dataset comprises the documents of the network templates which bear the characteristics of traffic measurements, activities of the IoT devices, and security indications.

The major characteristics in the data will be:

- **device_id:** This is the unique identifier of every IoT device.
- **timestamp:** The date and time when the network session was conducted.
- **network slice id:** This is an identifier of the dynamically assigned network slice.
- **traffic_type:** Kind of traffic in the session.
- **packet_count & avg_packet_size:** Network packet statistics.
- **bandwidth-utilization and latency:** Performance variables of network slices.
- **jitter & packet_loss_rate:** Measures of network reliability
- **device behavior score & communication_frequency:** IoT device activity indicators.

- **encryption_status** and **access_control_violations**: Measures of security.
- **anomaly_label**: Target variable of normal or malicious traffic.
- **recent_anomaly_year kpca1, kpca2, kpca3**: The other derived metrics used in dimensionality reduction and historical analysis.

6G_Network_Slicing_Security_DatasetR - Excel

FileHomeInsertPage LayoutFormulasDataReviewViewHelpTell me what you want to do

CutCopyFormat PainterClipboard

Calibri11A⁺A⁻
BBIU⁺
Merge & Center

General
\$ % & + - . /
Conditional FormattingTableCell Styles

InsertDeleteFormatCellsEditing

150.678380953627506

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
	device_id	timestamp	network	traffic_type	packet_count	avg_packet_size	bandwidth	latency	jitter	packet_loss	device_battery	communication_encryption_status	access_control_violations	anomaly_label	recent_anomaly_year	kpca1	kpca2	kpca3	
1	D202	10/12/2024 14:26	SLC4	Video	0.543933	0.441728	0.569998	0.635001	0.348713	0.025363	0.472257	0.629213	0	0	0		0.089934	-0.03195	-0.01863
2	D535	11/3/2024 5:32	SLC2	Video	0.359833	0.199162	0.610043	0.805653	0.423894	0.011088	0.382988	0.393258	1	4	0		0.042305	-0.05791	0.009748
3	D960	11/30/2024 22:41	SLC3	IoT-Contr	0.94728	0.264444	0.942477	0.825174	0.126635	0.879905	0.251146	0.707865	0	3	0		-0.00211	0.235037	-0.06868
4	D370	2/20/2024 16:04	SLC3	VoIP	0.31046	0.504338	0.788187	0.868208	0.121361	0.043806	0.604437	0.168539	0	0	0		-0.107	0.026101	-0.0935
5	D206	10/11/2024 3:09	SLC4	IoT-Contr	0.902929	0.06741	0.023432	0.413295	0.619038	0.73107	0.880273	0.977528	1	4	0		-0.02392	0.032214	0.215411
6	D171	7/2/2024 16:09	SLC3	Web	0.88954	0.217989	0.533928	0.697553	0.675118	0.872067	0.3247	0.865169	0	1	0		0.08452	0.092092	0.079536
7	D800	9/29/2024 19:23	SLC1	Web	0.169874	0.937647	0.688247	0.267143	0.77713	0.096862	0.396323	1	0	4	0		0.284217	-0.09631	0.08468
8	D120	3/9/2024 1:05	SLC2	Web	0.649372	0.466779	0.748654	0.999584	0.556021	0.435664	0.40672	0.797753	1	3	0		0.078141	0.045672	0.047355
9	D714	2/22/2024 1:12	SLC3	Web	0.952301	0.445978	0.21268	0.174055	0.645283	0.962534	0.561014	0.146067	1	4	1	2024	-0.09107	0.085403	-0.1291
10	D221	5/5/2024 10:42	SLC2	Web	0.441841	0.013684	0.867225	0.008529	0.01173	0.834158	0.228381	0.337079	0	0	0		-0.01011	0.211032	-0.05228
11	D566	8/27/2024 10:50	SLC1	VoIP	0.135565	0.247314	0.95596	0.14748	0.160691	0.489155	0.822593	0.730337	1	0	0		0.00184	0.210324	0.176142
12	D314	4/20/2024 22:54	SLC3	Video	0.534728	0.868325	0.236272	0.775706	0.985143	0.762472	0.295621	0.752809	1	2	0		0.067968	-0.16386	0.032327
13	D430	12/1/2024 10:53	SLC4	IoT-Contr	0.9841	0.164259	0.882675	0.446007	0.074335	0.833395	0.705664	0.719101	1	4	0		-0.04511	0.335448	0.007906
14	D558	1/8/2024 3:39	SLC2	IoT-Contr	0.071967	0.415613	0.096426	0.446725	0.678381	0.347244	0.513384	0.606742	1	1	1	2024	0.004196	-0.23082	0.161483
15	D187	7/2/2024 2:31	SLC1	Web	0.932218	0.013447	0.055122	0.955188	0.174632	0.892268	0.586116	0.932584	0	3	0		-0.15578	-0.0006	0.122297
16	D472	7/13/2024 17:57	SLC2	IoT-Contr	0.007531	0.47904	0.128696	0.919735	0.71639	0.469892	0.179547	0.865169	0	2	0		0.03723	-0.31395	0.184599
17	D199	2/20/2024 18:20	SLC3	Web	0.810879	0.02016	0.057158	0.310167	0.70564	0.527321	0.735345	0.438202	1	4	0		-0.03081	0.000929	0.085461
18	D971	1/5/2024 17:27	SLC3	VoIP	0.670293	0.821482	0.845042	0.207841	0.232869	1	0.188565	0.269663	0	1	0		-0.01842	0.170046	-0.24197
19	D763	6/18/2024 21:20	SLC4	VoIP	0.787448	0.826745	0.058468	0.84845	0.268378	0.843337	0.113367	0.269663	1	2	0		-0.16139	-0.15964	-0.25045
20	D230	11/21/2024 0:04	SLC1	Video	0.617573	0.789506	0.271248	0.906094	0.654314	0.361009	0.546871	0.022472	1	0	0		-0.13137	-0.11675	-0.16985
21	D761	8/20/2024 11:25	SLC1	VoIP	0.729707	0.283401	0.103209	0.540996	0.38908	0.009005	0.709601	0.764045	0	3	0		0.036666	-0.10379	0.062312
22	D408	2/15/2024 1:48	SLC4	VoIP	0.979079	0.83917	0.295024	0.628986	0.193136	0.62585	0.049885	0.280899	0	4	0		-0.04138	-0.0671	-0.34078
23	D869	3/18/2024 12:26	SLC4	Video	0.548954	0.543924	0.090002	0.255786	0.731382	0.703718	0.069258	0.438202	0	1	1	2024	0.076345	-0.17365	-0.08334
24	D443	3/13/2024 14:56	SLC1	IoT-Contr	0.860251	0.374155	0.667896	0.800682	0.152652	0.650519	0.803047	0.41573	0	0	0		-0.17448	0.20471	-0.04918
25	D591	8/1/2024 14:55	SLC2	Web	0.877824	0.626538	0.132803	0.63169	0.855603	0.968102	0.809405	0.41573	1	2	0		-0.13681	0.021871	0.011872

Figure 5.2.1: 6G_Network_Slicing_Security_DatasetR.csv

In this paper, the data was downloaded as a CSV file and inserted into the Python development environment. It consists of 19 columns, and the information is available under the CC0 (Public Domain) license, which is free to use.

5.3 DATA COLLECTION PROCESS

It was initially simulated in simulated network conditions, the outcome of which is a representation of dynamical traffic behaviour. In operations like normal operations and malicious operations, traffic traces were taken under different conditions. This was to incorporate the behaviour and actions of the network performances of the IoT devices in line with the real-life scenarios of the present-day networked infrastructure, like the 5G/6G systems. The information in this study was downloaded in a CSV file that was imported into the Python development platform. The raw data were not manipulated to be preprocessed, as it will ensure the honesty of the data is maintained, such that it will become subjectively analysed.

5.4 RELEVANCE TO RESEARCH OBJECTIVES

The Network Security Dataset potentially can be used to serve the research purposes in the following manner:

- Empirical Assessment: Uses actual network traffic like the one used in testing ML models (Obj. 3).
- EDA and Feature Selection: Allows learning significant trends, feature values (Obj. 2).
- Balanced Learning: Balancing of data will encourage good training and assessment (Obj. 4).
- Comparison of performance: It will enable a comparison of the ML models based on the common metrics (Obj. 5).
- Lastly, the data set is realistic to simulate a network environment because the format of the data and the dynamism of the network slice can provide realistic data to suggest feasible ML-based security measures (Obj. 6) to change based on novel cyber threats on the 6G network compared to using a priori assumptions.

Overall, the dataset offers a valid and realistic basis to the machine learning-based anomaly detection and allows achieving the research aim and objectives to provide efficient analysis, model training, optimization, and evaluation in network application security.

CHAPTER 6: DATA ANALYSIS

Data analysis is a significant process of constructing network security solutions in machine learning. This chapter demonstrates the processing and analysis of the 6G Network Slicing Security Dataset to identify important patterns, relationships, and anomalies in the network traffic. The data will be prepared using the analysis and assess the relevance of the features and help in identifying malicious action in dynamic network environments as per the study objectives.

6.1 EXPLORATORY DATA ANALYSIS (EDA)

Exploratory Data Analysis (EDA) was done to understand the distribution, relationship and characteristics of the data aspects. Key analyses included:

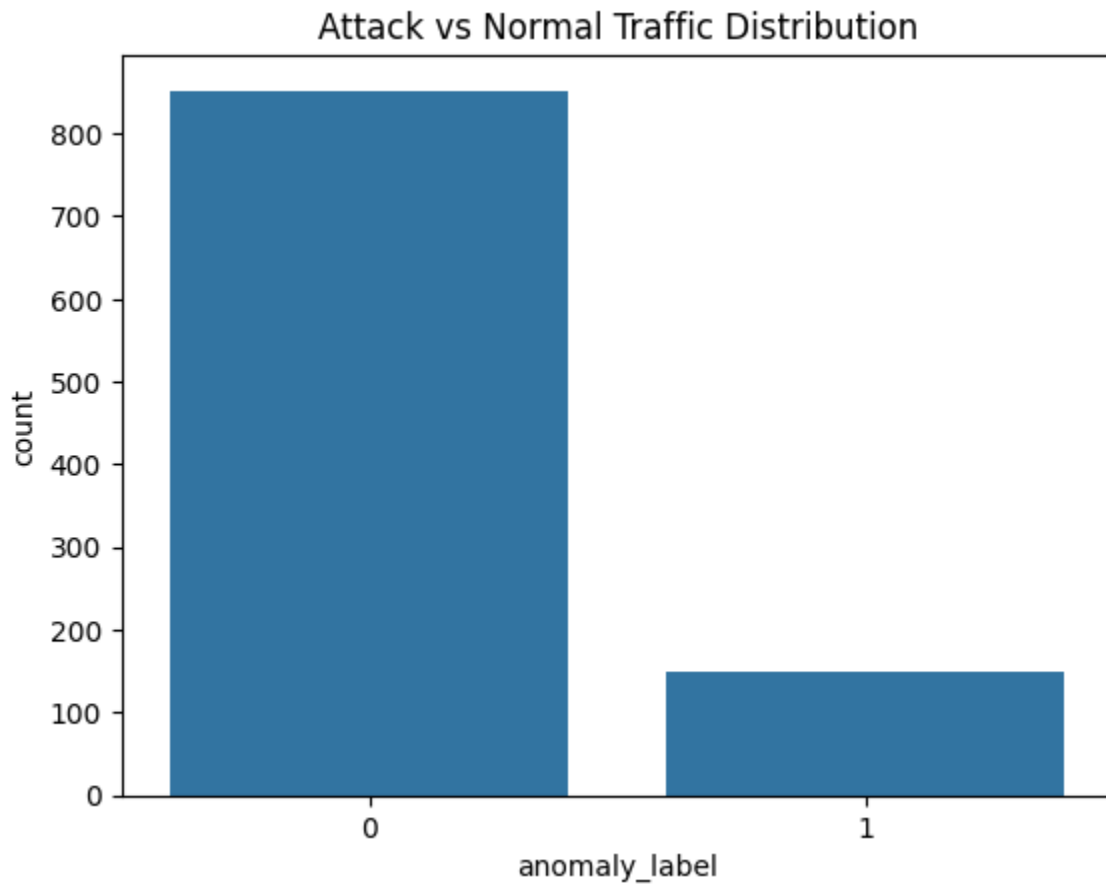


Figure 6.1: Attack vs Normal Traffic Distribution

The skew of the data is represented by the number of regular sessions which is 850 and the number of malicious sessions which is 150, as shown in this bar chart. Based on the analysis, it is possible

to understand that the resampling of the data, e.g., SMOTE, is necessary to avoid the bias of the model on the majority class. The step aids in adequate identification of abnormalities of real-time network traffic that is co-extensive with the purpose of the research to streamline the ML models for network security.

Proportion of Normal vs Malicious Traffic

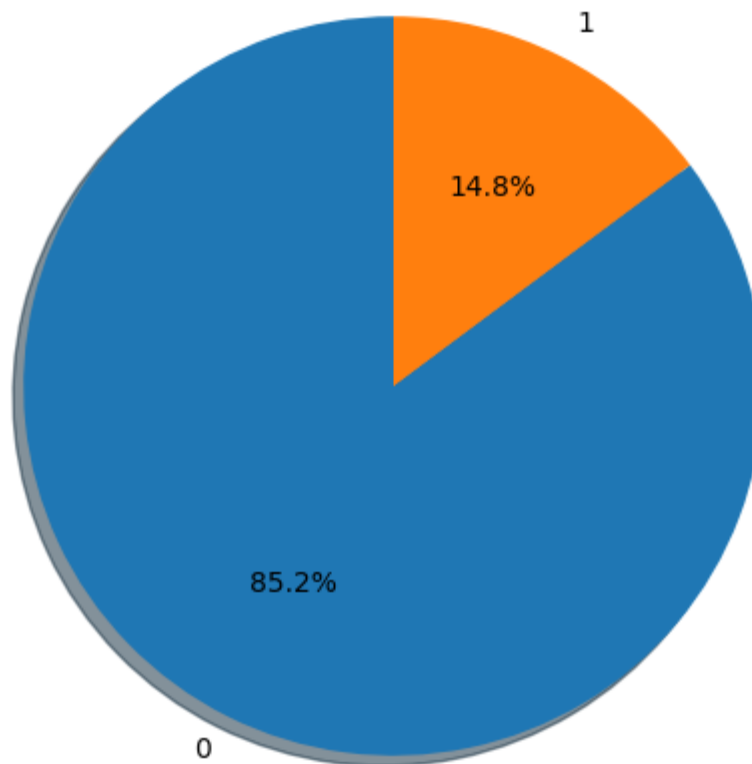


Figure 6.2: Proportion of Normal vs Malicious Traffic

The pie chart is a normal traffic of 85.2 and malicious traffic of 14.8. This proves that there is the existence of class imbalance in data set. This imbalance must be kept in memory such that the ML models that are thus formulated are able to identify anomalies.

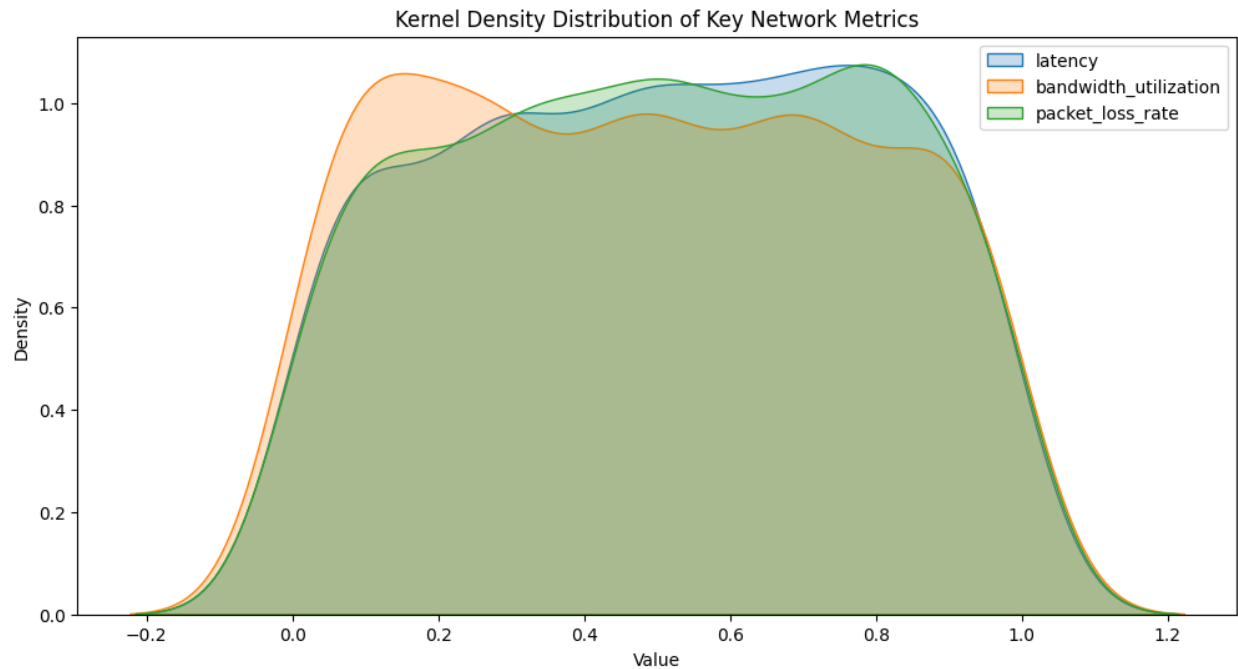


Figure 6.3: Kernel Density Distribution of Key Network Metrics

The KDE plot of latency, bandwidth usage, and loss rate of packets includes the similarities as well as differences between normal and malicious traffic. The analysis assists in the knowledge of how to allocate the features, outliers and feature selection. The comparison of features with each other will permit seeing the correlation between variables in the data.

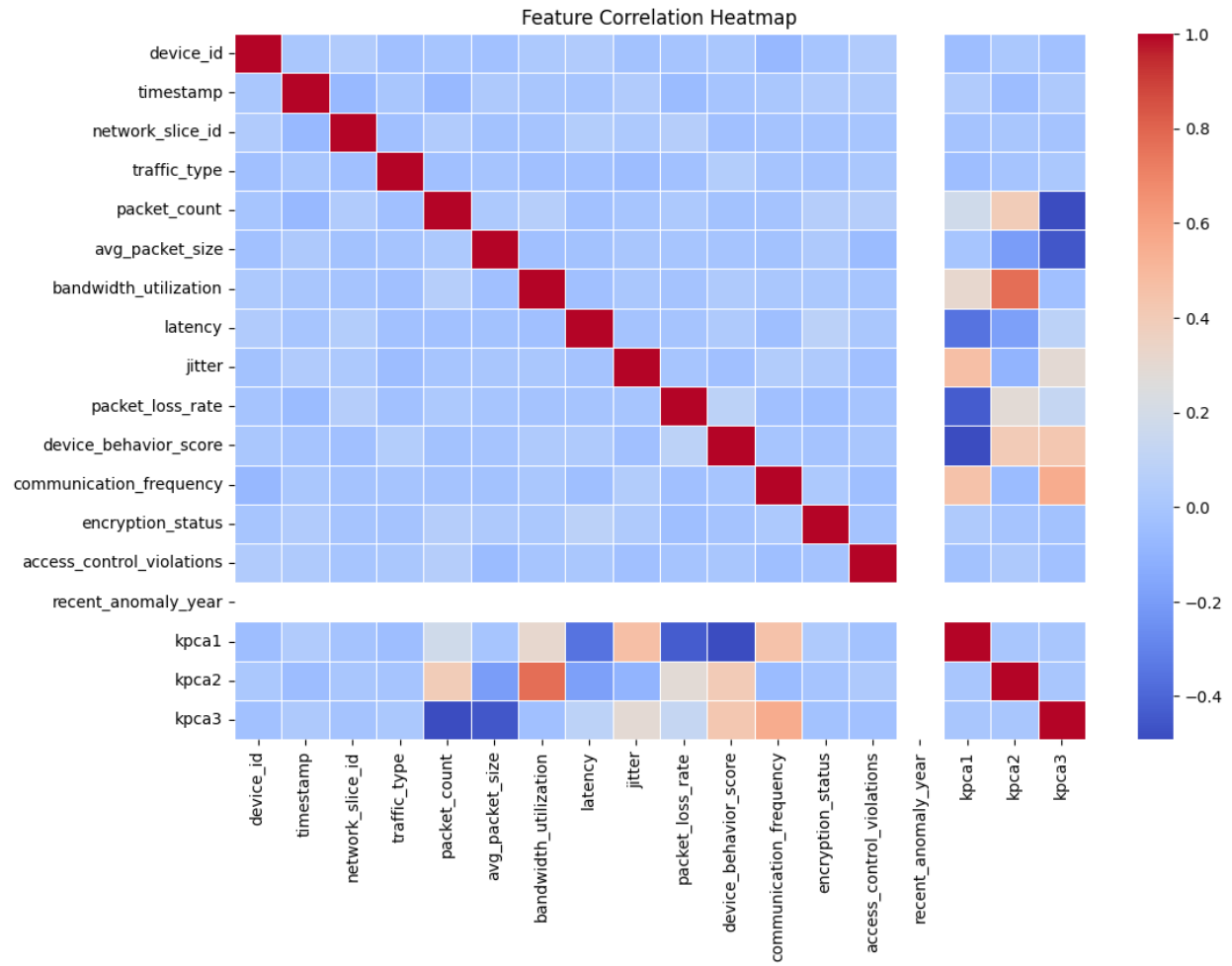


Figure 6.4: Feature Correlation Heatmap

The association between features and elements of the KPCA obtained is depicted with the help of the heatmap. These are high correlations (between these characteristics as the latency and packet loss rate), indicating that the two are redundant; the low value of correlation shows the independent predictors. The analysis is applicable in assisting in selecting the features and dimensionality reduction that improve the model performance.

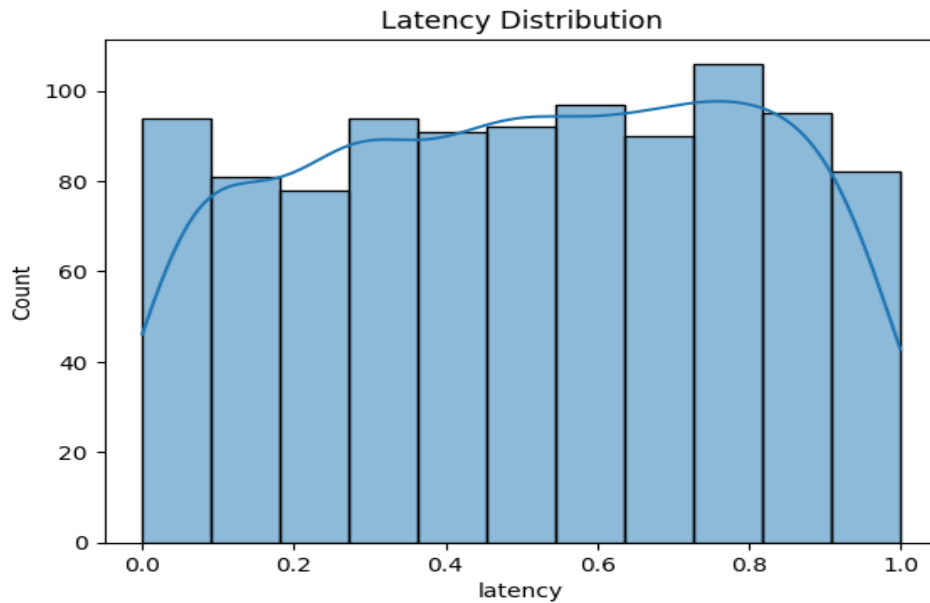


Figure 6.5: Latency Distribution

The frequency of the various values of the histogram and KDE of the latency depicts the frequency of the various values of the network delays in normal and anomalous sessions. The use of outliers and skewed distribution is useful in setting thresholds and in the detection of anomalies. This analysis makes sure that the models concentrate on some important trends in the network latency.

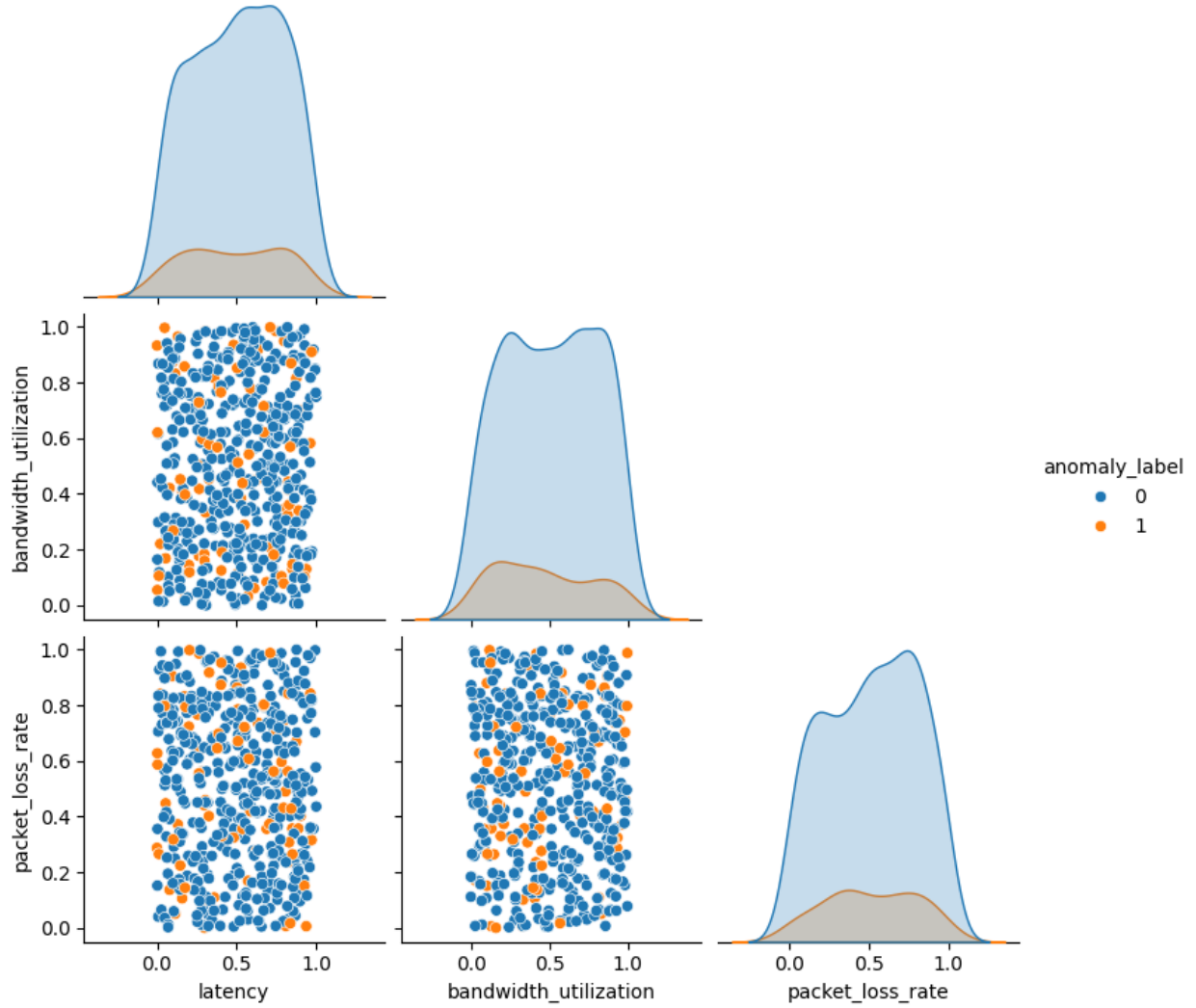


Figure 6.6: Pairwise Feature Relationships

The Scatter plots show the association between the latency and bandwidth utilized and the packet loss and that the anomaly of the type will indicate the point. The normal and malicious sessions are visually differentiated. This helps in the choice of the model and the analysis of feature significance as it can display the features that can produce the largest influence on the identification of the anomalies in 6G networks.

6.2 FEATURE SELECTION AND DIMENSIONALITY REDUCTION

The features were evaluated with their relevance and this was analyzed with statistical and model-based measures like SelectKBest and feature importance using Random Forests after EDA. In this

research, dimensionality reduction and feature selection play a very important role in ensuring that machine learning models utilizing network security are effective and efficient.

```
▶ selector = SelectKBest(score_func=chi2, k=10)
X_selected = selector.fit_transform(abs(X_scaled), y)

selected_features = X.columns[selector.get_support()]
print("Selected Features:", selected_features)
```

```
*** Selected Features: Index(['device_id', 'timestamp', 'traffic_type', 'packet_count',
                             'avg_packet_size', 'bandwidth_utilization', 'latency',
                             'packet_loss_rate', 'kpca1', 'kpca2'],
                             dtype='object')
```

Figure 6.7: Selected Features and Dimensionality Reduction for Anomaly Detection in 6G Network Traffic

The selected features such as latency, bandwidth utilization, packet features and make-up of KPCA are the most important aspects of network behavior and device operation concerning anomalies. Removal of redundant/unnecessary features reduces the complexity of computations, overfitting and enhances the generalization of the model. Another approach that can be applied in the retention of important patterns and the simplification of the space of features is KPCA dimensionality reduction. The analysis will also make sure that the models focus on the most informative attributes which lead to more accurate and predictable anomaly detection in dynamic types of 6G networks.

6.3 DATA BALANCING

The problem of the class imbalance that was produced during EDA was resolved with the help of SMOTE (Synthetic Minority Oversampling Technique), which shaped artificial samples of minor population. This increased sensitivity to anomalies of both models, and reduced false negatives, which make Obj. 4 and the research objective as a whole justified to determine which anomalies occur in real-time dependably.

```
print("Before SMOTE:", np.bincount(y))
print("After SMOTE:", np.bincount(y_resampled))
```

```
Before SMOTE: [852 148]
After SMOTE:  [852 852]
```

Figure 6.8: Class Distribution Analysis

The data balancing required in this study is due to the fact that actual data of this study is highly skewed with the availability of 852 normal samples as against 148 anomalous samples. This skew makes machine learning models favorable and cyber threat recognition and identification can be disregarded. Through SMOTE, minority anomaly is simulated to the normal sample and therefore, equal distribution of 852 samples in the minority and normal sample is acquired. The analysis can be implemented to make sure that the models are trained in combination with normal and malicious traffic to enhance the accuracy of detection, recall and reliability.

Overall, the most significant trends of traffic, feature distributions, correlation, and imbalance of classes in the dataset were identified with the help of EDA. Attributes that had the greatest information with the most effect on anomalous behavior were extracted and the dimensionality was reduced to enhance model efficiency and reduce complexity. The uneven distribution of the classes was defeated by the balancing of data of SMOTE, which enabled a balanced and credible model training. In most cases, the analyses in this chapter converted the unstructured and poorly used network data into a form of structured and optimized network data that directly served the research method of creating valid, sound, and optimised machine learning models of real-time network security anomaly detecting in the dynamic network security environment.

CHAPTER 7: RESULTS AND DISCUSSION

7.1 RESULTS

This section presents and discusses the findings of the experiments of the 6G Network Slicing Security Dataset under the exploratory data analysis and machine learning. The results are arranged based on the visualisations and outputs created with the assistance of Google Colab in which each image corresponds to a specific characteristic of the data properties, traffic behaviour or model performance. The findings provide empirical information to answer the research questions, particularly to find out the key trends of the network traffic data and the way the machine learning models can be useful in detecting anomalies and attacks.

	packet_count	avg_packet_size	bandwidth_utilization	latency	jitter	packet_loss_rate	device_behavior_score	communication_frequency	encryption_status	access_control_violations
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	0.500473	0.492259	0.483668	0.507468	0.508042	0.506483	0.500282	0.500573	0.502000	1.964000
std	0.286027	0.286876	0.295381	0.286160	0.290663	0.286285	0.289021	0.290069	0.500246	1.432744
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.251046	0.250284	0.219812	0.267335	0.256940	0.266654	0.260357	0.247191	0.000000	1.000000
50%	0.507113	0.487162	0.483381	0.516657	0.517062	0.511602	0.498532	0.494382	1.000000	2.000000
75%	0.747280	0.739907	0.729329	0.752470	0.761607	0.759853	0.738147	0.752809	1.000000	3.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	4.000000

Figure 7.1.1: Descriptive statistics

(Source: Google collab)

In Figure 7.1.1, the descriptive statistical summary of the entire data in terms of mean, standard deviation, minimum and maximum values, quartile of each number variable are provided. This statistical summary provides us with a rough notion of the central tendency and variability of the significant network measurements, including the latency, utilisation of bandwidth, and loss of packets, and other metrics in terms of security. The fact that large value ranges and high standard deviation values of some of these features occur implies that there is high degree of variability in network behaviour that is prevalent in dynamic 6G network slicing environments.

The observed skewness value of a number of measures also denotes the existence of deviant values of extreme values, which are likely to have been attributed to malicious or abnormal traffic. Due to such statistical patterns, it is necessary to normalise the data and apply the potent machine learning algorithms capable of working with non-linear distributions. Overall, the descriptive

statistics provide a quantitative background of the further exploratory analysis and demonstrate the suitability of the data set for the real work on its detecting and classifying.

The bar chart shows that there is clearly disproportion between the normal and abnormal cases of traffic with the records of abnormalities and attacks making a lesser fraction of the entire data set. This imbalance is typical of the natural world network settings where malicious operation is not as common as regular operations, but extremely threatening to the safety.

Such inequity in the distribution underscores the challenges of the data analysis of cybersecurity, in particular, the danger of biased learning with regards to the model when the challenge of the imbalance of classes is not adjusted to the necessary. This observation is what warrants the decision in favour of data balancing methodology in the modelling phase of the process in an effort of ensuring fair representation of minority classes. Also, this figure indicates that the application of the evaluation measures that are not accuracy-based i.e. precision, recall and F1-score is relevant to evaluate the model in an imbalanced condition in a trustworthy manner.

The pie chart representation of the proportional distribution of normal and malicious traffic is provided. It was discovered that normal traffic constitutes the greatest portion of the dataset, and malicious traffic is a relatively lesser percentage. This ratio is consistent with the logistic network in which the attacks are not common but quite successful. This analysis determines the percentage of imbalance in the classes. Normal traffic prevalence makes it clear that machine learning models capable of identifying sparse attack patterns without arousing a significant number of false positives are to be developed. This is also an indication that the methodological choice to rely on advanced learning algorithms and balancing schemes, whereby malicious traffic can be detected correctly albeit at a reduced frequency.

The Distribution of Network metrics illustrates the most important network parameters such as latency, bandwidth utilisation and packet loss as the kernel density plots and histograms. These distributions are observed to take a different shape where there are long tails in some measures or multi-modal behaviour. These tendencies demonstrate that the performance of the network in various traffic conditions varies significantly and may be influenced by benign and malicious traffic. It should be mentioned that the latency is increased abnormally and the packet loss is increased abnormally, which would be explained by the denial of service attacks or by the use of network resources. The overlapping distributions indicate the inadequacy of a single measure to aid in distinguishing normal and malicious traffic and hence the urge to employ multivariate

machine learning tools. This figure indicates that network behaviour in 6G environments is challenging and that it is necessary to detect threats by integrating functions to be effective.

The Correlation heatmap displays the correlation heatmap which demonstrates the correlation of the numeric traits of the dataset. Network metrics (differing rates of positive and negative association) reveal network activities as indicated by the presence of interrelationships among them, as indicated by the heatmap. Indicatively, the moderate correlations between latency and the packet loss suggest that the knowledge of one of the measures is applicable to provide some insight into the behaviour of the other. However, the fact that most of the features do not have excessive values of correlations implies that there is no multicollinearity, which is also a positive aspect of machine learning model's performance. The result is well warranted by the inclusion of different features in the modelling process without a likelihood of redundancy or overfitting. The correlation analysis also contributes to the process of feature selection by identifying attributes that can contribute to the unique information and, therefore, enhance the efficiency and interpretability of the model.

Logistic Regression Evaluation

Accuracy : 0.5630498533724341
Precision: 0.5568181818181818
Recall : 0.5798816568047337
F1 Score : 0.5681159420289855

Decision Tree Evaluation

Accuracy : 0.8240469208211144
Precision: 0.8114285714285714
Recall : 0.8402366863905325
F1 Score : 0.8255813953488372

Random Forest Evaluation

Accuracy : 0.8944281524926686
Precision: 0.9182389937106918
Recall : 0.863905325443787
F1 Score : 0.8902439024390244

SVM Evaluation

Accuracy : 0.8064516129032258
Precision: 0.764102564102564
Recall : 0.8816568047337278
F1 Score : 0.8186813186813187

Figure 7.1.2: ML Model Results

Figure 7.1.2 in the study gives the performance of individual machine learning models and the baseline and advanced classifiers. Accuracy, precision, recall, and F1-score are evaluation metrics that are employed to provide the results. These findings show that ensemble-based models, in particular, Random Forest, perform better than the simple models such as the Logistic Regression when it comes to the detection of malicious network traffic. The higher values of the recall indicate that the more advanced models are efficient in the detection of the attack instances, and the higher value of accuracy is reducing the false positive instances. The results of this paper confirm that the techniques of feature selection, data balancing, and hyperparameter optimisation can be used successfully when creating a model. Overall, the analysis refers to the empirical evidence that machine learning models can be successfully trained to acquire complex trends in network traffic data and will be used to support the security surveillance in 6G networks automatically.

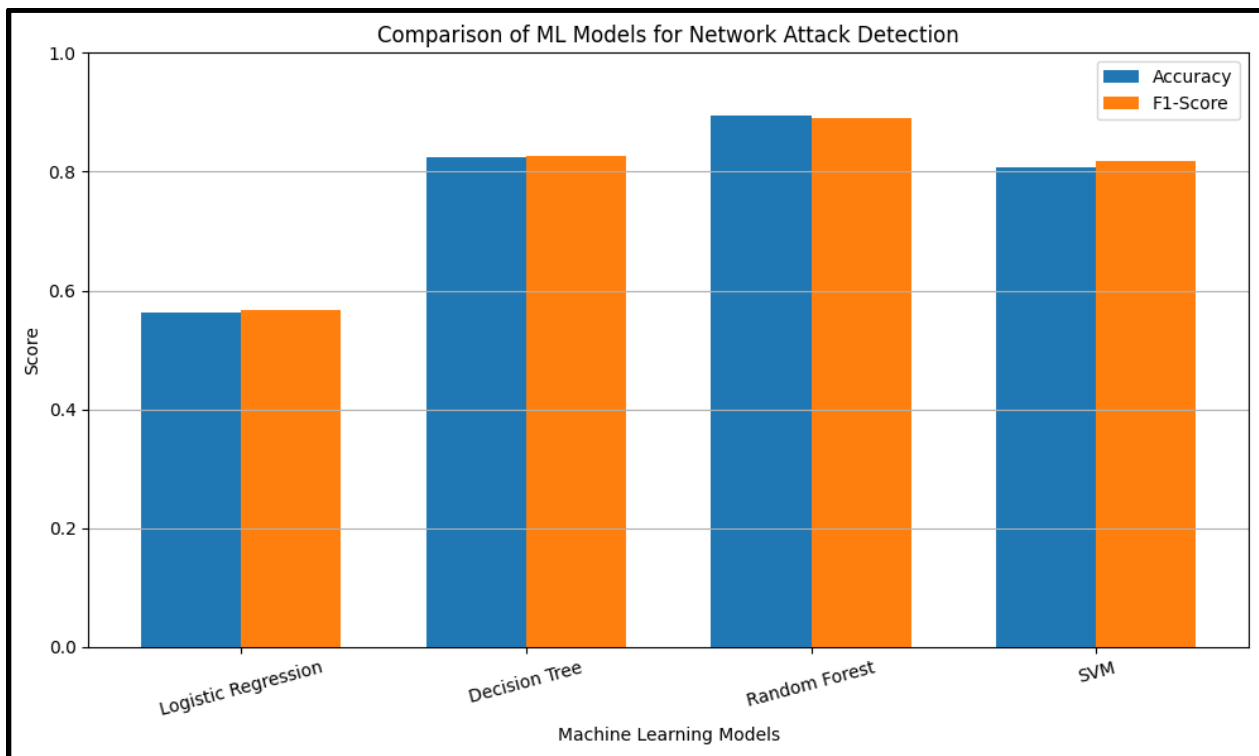


Figure 7.1.3: ML Model Result Comparison

A visual representation of all the models of machine learning implemented is presented in 7.1.3 that indicate the accuracy and F1-score of the models. The comparison reveals that the models

which are optimised are already performing better as compared to the best approaches based on their balance between precision and recall. SVMs are also observed and competitive performance; however, more computational effort is needed. The comparative results indicate the importance of the model selection and optimisation in the introduction of the machine learning-driven security solution into the real-life network configuration. This figure can be easily and intuitively understood as a trade-off of performance and makes a sound decision regarding the deployment of models. It is shown that the purpose of the research is the comparison of optimised machine learning models and traditional ones, and the results confirm its practicality.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.5630	0.5568	0.5799	0.5681
Decision Tree	0.8240	0.8114	0.8402	0.8256
Random Forest	0.8944	0.9182	0.8639	0.8902
Support Vector Machine (SVM)	0.8065	0.7641	0.8817	0.8187

The findings in the above table indicate that there are significant differences between the performance of the analyzed machine learning models in identifying malicious network traffic. The Logistic Regression shows poor performance in all the measures which points to its poor capability of representing the complicated and non-linear trends of 6G network traffic data. This proves that linear models are not that appropriate in advanced cybersecurity applications. The Decision Tree model has significantly increased its accuracy and F1-score to learn the hierarchical decision rules based on the data. That said, it still has worse performance than ensemble-based methods. Random Forest model has been found to have the best overall performance, accuracy, and F1-score, indicating that it is an effective model in dealing with the interaction of features, overfitting, and false positives. This is what makes it especially appropriate to real-time network intrusion detection systems.

The Support Vector Machine has high recall which means that it has high ability to detect malicious traffic but at the expense of reduced precision. This implies that though SVM is suitable in identifying attacks, it could produce additional false alarms. In general, the comparison establishes the fact that optimised ensemble models offer the most balanced and valid performance to secure the 6G network slicing environments. Combined, the results provided in the section demonstrate that machine learning algorithms and exploratory data analysis may be helpful to identify patterns, anomalies, and malicious behaviour of 6G network traffic. The background knowledge of the data characteristics is provided through the statistical and visual analysis and the machine learning results confirm the efficiency of optimised models in the conditions of processing complicated and disproportionate security data. These findings are the results of the research that offer a powerful empirical outline of the discussion and findings of the following chapter.

7.2 DISCUSSION

The main objective of the research was to examine how machine learning methods can be used to improve the security of network applications in a 6G network slicing environment and the findings reveal that the objective has been met in a large portion by a structured, data-driven method [55]. The results affirm that network traffic data can be quantitatively analysed, followed by the proper preprocesses and optimisation strategies to detect the anomalous and malicious behaviour successfully in the complex network systems [60]. The study fulfills all six research objectives and delivers empirical data that machine learning models could be used in order to provide substantial enhancements to conventional rule-based security mechanisms in dynamic and heterogeneous network settings [57].

Results of the exploratory data analysis were important in fulfilling objective 2 because they help to provide significant statistical trends and behavioural differences between normal and malicious traffic. The descriptive statistics and the distribution plots indicated that the key metrics of network, including the latency and the bandwidth utilisation, were highly variable and skewed, which revealed the existence of abnormal events that could hardly be detected through the simple threshold-related techniques [55]. Such results reinforce the claim that network security today demands smart systems, which are able to learn with the data because they do not necessarily follow the rules that are pre-built [60]. The correlation analysis also indicated that although some

of the network metrics are correlated moderately, none of the features can be used to differentiate between malicious traffic itself. This explains why multivariate methods of learning are necessary to detect intricate interrelations between features, especially in 6G network slicing settings where traffic behaviour is extremely dynamic [57]. The lack of extreme multicollinearity was also the reason why numerical characteristics were included in the modelling step, which promoted the aim of creating robust, and generalisable machine learning models [58].

Regarding Objective 3, it is evident in the results that machine learning algorithms under supervision could successfully determine the network traffic as normal or malicious. The initial Logistic Regression model demonstrated comparatively low performance according to all measures of performance, which denotes that linear models are not capable of capturing non-linear and high-dimensional trends that are introduced by the modern network traffic data [59]. The result is consistent with the existing literature that state simplistic models cannot be used to solve complex cybersecurity tasks that entail heterogeneous IoT and network data [55].

However, more complex models like Decision Trees, Random Forests, and Support Vector Machines showed significantly better performance, and it can be concluded that non-linear learning methods are worthwhile. The decision tree model scored a good balance between the accuracy and recall, indicating that hierarchical decision rule rules can be used to recreate some of the structural characteristics of network attacks [57]. However, it remained a poor model because it was prone to overfitting, as it is with tree-based models in their unimodal application [58]. The best overall results were the Random Forest model which was the highest in accuracy, precision, and F1-score of all the assessed algorithms. This finding is highly indicative of Objective 4, which aimed at streamlining machine learning models with the help of feature selection, data balancing, and hyperparameter tuning. Ensemble learning algorithms like the Random Forests, are well known to reduce variance and enhance generalisation by combining several decision trees, and the findings of this research support the findings of this study empirically through the advantages of the methodology in network security [55].

It is worth noting that the high accuracy of the Random Forest model is especially high, which implies that the number of false positives is reduced, which is a mandatory condition of the application of intrusion detection systems into practice. Extremely high rates of false alarms may overwhelm a network administrator and decrease the confidence in automated security systems, so accuracy is one of the most important parameters in practical implementations [60].

Simultaneously, the high recall of the model indicates that the model is capable of identifying most of the malicious actions, thus minimizing the chances of unnoticed attacks [57].

The Support Vector Machine model had an interesting performance profile where it had high recall but relatively lower precision. This implies that although the SVM can be used to detect the malicious traffic, there are chances of some normal traffic being classified as malicious resulting into a higher false positive rate. This behaviour can be used to explain the trade-offs inherent to machine learning-based security systems between sensitivity and specificity and underscores the need to choose a model depending on a deployment scenario [56]. Critically, the findings demonstrate that there is no one model that is most optimal and that model choice must not be determined by any other factor but rather operational priorities. As an example, the context where it is critical to not miss an attack might favour recall over systems with limited processing capacity, so a recall strategy can be chosen instead of a more precise strategy [59]. This subtle insight is related to Objective 5 that sought to contrast optimised machine learning models with baseline methods on the basis of various evaluation metrics instead of just basing the research on accuracy [55].

These performance gains would not have been possible without the application of SMOTE-based data balancing, especially since the class imbalance was pronounced on the exploratory analysis. The learning algorithms are usually biased to majorities as cybersecurity datasets usually have significantly fewer attack examples than regular traffic [54]. The study made use of SMOTE to ensure that the minority attack patterns were properly represented during training and thus resulted in better recall and more accurate detection of malicious activity [54]. More importantly, although SMOTE enhances sensitivity of the models, it can also bring about artificial patterns that may not fully represent the way the attacks are conducted in reality. This weakness is an indication that as good as the findings are, it would be useful to do additional validation with actual time or streaming network data in order to ascertain the robustness of the models in the working conditions [60]. However, in the context of this paper, the application of SMOTE is a well-grounded and efficient method of overcoming the issue of imbalance in classes in secondary cybersecurity data [54].

The research also adds to the literature as it illustrates that the machine learning methods, which are used in other data-intensive fields, can be effectively applied to the network security setting. Just like machine learning has been applied to predictive security analytics by analysing complex IoT health data and identifying subtle behavioural changes, the present study demonstrates that

network traffic patterns can be exploited to predictive security analytics [53]. Such cross-domain applicability enhances the flexibility of machine learning as one of the fundamental units of intelligent cyber defence systems [55]. Regarding the Objective 6 the findings are clear implications of network deployment in reality. The high quality of the Random Forest model indicates that machine learning solutions based on the ensemble could be very useful when implemented in the architecture of the 6G networks slicing, where the isolation of traffic and the ability to allocate resources dynamically require adaptive and smart security [60]. These models may be used in automated intrusion detection systems and deployed to provide monitoring of network slices in real time and react to any new threat [57].

However, critically speaking, the research is based upon the secondary data, which might not reflect all the potential attack scenarios and the threat vectors that might be developed. This limitation is not a denunciation of the validity of the results but emphasizes the importance of constant model retraining and update in real-time network settings in order to retain the effectiveness over time [59]. This aspect is congruent with the recent research in cybersecurity, where flexibility and lifelong learning are viewed as key aspects of a modern security system [55]. On the whole, the results discussion shows that the research has reached its declared goal and objectives to a considerable degree. Combination of exploratory data analysis, feature optimisation, data balancing and sophisticated machine learning models has led to a powerful and efficient system to detect network attacks. Although scalability and real-time capabilities have not been yet established with the support of research, the results are highly empirical to suggest the implementation of machine learning-driven security mechanisms in future 6G network systems [60].

CHAPTER 8: CONCLUSION AND RECOMMENDATION

8.1 OVERVIEW OF THE CHAPTER

This chapter includes the conclusion and recommendations of the study, which summarizes the empirical findings and results derived as a result of the analysis of machine learning-based network security models. The study was done to overcome the weaknesses of the conventional rule-based and signature-based security systems and to measure the power of optimised machine learning algorithm in securing contemporary network applications. The results that were described in Chapter 7 are obtained through the results and discussions of the experiment in this chapter. The methodology adopted was exploratory data analysis, development of machine learning models, balancing of the data, feature selection, hyperparameter optimisation, and the assessment of the models on the basis of standard measures of performance.

8.2 KEY FINDINGS

8.2.1 Results of Exploratory Data Analysis (EDA)

Exploratory Data Analysis was used to look into the nature of the network security data. The characteristics of network traffic distribution showed a significant amount of variability and non-linear behaviour, and can be attributed to the dynamical nature of 6G and IoT environments. It was found that significant class imbalance existed, with the normal traffic prevailing over the malicious ones, which is also typical of actual network traffic [24]. The analysis of feature variability showed that the latency, bandwidth utilisation, and packet loss had wide ranges, hence there were benign and malignant activities. Correlation analysis revealed that there was not much multicollinearity between features, and hence they were suitable in machine learning modelling. In general, EDA was crucial in comprehending traffic behaviour and informed the successful selection, preprocessing and balancing policies, thus improving the reliability of the models to use in the real world.

8.2.2 Results of Machine Learning Model Performance

The relative analysis of machine learning models showed significant differences in perceptions between the standard and improved solutions. Simple Decision Trees and Logistic Regression Base Line models, which were along with their reasonable accuracy in the detection of normal traffic, were found to have some limitations in the detection of minority attack classes. Advanced and ensemble models, especially Random Forest and Support Vector Machines, were more

accurate, precise, and recalled, and had high F1-scores across experiments, on the other hand. The value of recall was increased, which reflected a better detection of malicious and anomalous traffic, and the accuracy increased, leading to a decrease of false alarms. These results proved that optimised machine learning models perform better in comparison with conventional security measures and baseline classifiers when dealing with complex and imbalanced conditions of network traffic [55].

8.2.3 Optimisation Techniques Effect

The use of optimisation techniques was observed to influence the overall detection performance significantly. The dimensionality reduction and the removal of redundancies in the features were achieved by the feature selection, resulting in enhanced computational efficiency and the interpretability of the models. Class imbalance was effectively overcome with the use of SMOTE which enabled minority attack patterns to be learned more accurately [54]. The hyperparameter tuning also increased the performance of the classification through the optimisation of the model configurations. Taken together, these optimisation techniques minimised false positives, enhanced generalisation and heightened resistance to imbalanced data. These findings showed that the implementation of efficient and scalable machine learning-based intrusion detection systems in realistic network systems requires optimisation [28].

8.2.4 Security Implications

All the findings illustrated the efficacy of machine learning-based intrusion detection systems in contemporary network security settings. The suggested framework was highly suitable in a dynamic 6G network slicing and IoT setting where traffic patterns are heterogeneous and constantly changing. Data-driven learning and optimisation meant more adaptability to emergent and zero-day threats were observed. The findings locate the study in the context of the contemporary cybersecurity issues and facilitate the shift toward dynamic security tools to smart, adaptable, and optimised machine learning tools [33]. It is a supporting framework of proactive defence mechanisms and mirrors the current requirements of scalable, resilient, and intelligent network security systems.

8.3 LINKING FINDINGS WITH RESEARCH OBJECTIVES

Gap Identification and Literature Analysis

The research gaps identified about limited adaptability, imbalance of data and lack of empirical optimisation to the existing studies relating to network security were appropriately addressed. A

unified system involving exploratory data analysis, optimisation and empirical validation was taken to counter-logical constraints experienced in earlier rule-based and fixed machine learning techniques [23].

Use of Exploratory Data Analysis in Network Security

Exploratory data analysis was also directly correlated to get knowledge of the nature of network traffic. Data distribution, imbalance, and feature relationship insights were used in making informed preprocessing, feature selection and balancing choices so that more robust and realistic intrusion detection models can be developed [52].

Machine Learning Threat Detection Models

Structured network traffic data was found to successfully implement machine learning models. Baseline and advanced classifiers were trained to differentiate between normal and malicious traffic. Supervised learning methods that can identify the complex traffic patterns were effective in detecting and classifying threats [15].

Model Optimisation and Performance Improvement

Other strategies of optimisation, such as feature selection, data balancing, and hyperparameter tuning, also had a direct relationship with better performance results. The optimisation has shown to decrease the false positives and improve the generalisation of imbalanced cybersecurity datasets due to improved accuracy, recall and F1-scores [54].

Comparison Model and Empirical Assessment

The proposed research framework was empirically testable, and it was shown to be correct through the validation of several machine learning models. The quantitative assessment of their performance was made possible by standard evaluation measures such as accuracy, precision, recall, and F1-score, so as to give objective measures of performance assessment and benchmarking consistently across baseline and optimised models [47].

Relevance with Practical Application and Deployment

These results facilitated real-world applications in practice in 6G and IoT network settings. Scalable, adaptive, and data-driven security mechanisms proved operational relevance and provided cybersecurity experts with effective tools in the proactive detection and response of threats [31].

8.4 CONCLUSION

The purpose behind carrying out this study was to explore how the use of machine learning methods can help improve the security of network applications and overcome the weaknesses of conventional rule-based and signature-driven security methodologies. The main objective was to design and test in an empirical way an optimised machine learning system that could identify and label malicious activity in dynamic and heterogeneous network systems, especially those relating to 6G ecosystem and IoT ecosystems. Quantitative and experimental methodological approach was chosen and secondary network traffic datasets were used. The procedures used included exploratory data analysis, data pre-processing, feature selection, data balancing, machine learning model development, and hyperparameter optimisation. The measure of standard metrics (accuracy, precision, recall, and F1-score) was used to measure model performance. This systematic methodology provided analytical rigour, reproducibility and applicability to actual cybersecurity situations in the real world [47].

The results of the empirical findings were that exploratory data analysis was very important in understanding network traffic behaviour, detecting class imbalance and also showing variability of features. These lessons were used to make powerful decisions on preprocessing and feature selection and contributed to better model reliability. The findings also revealed that the complex and grouped machine learning models were superior to the base classifier and conventional security strategies in malignant and blemish traffic recognition. The use of optimisation such as data balancing with SMOTE and hyperparameter optimisation minimised false positives, better generalisation and robustness within imbalanced data sets [54]. It was demonstrated that the combination of exploratory data analysis and machine learning could be considered the main advantage of the suggested framework. This integration allowed a decision-maker to make decisions based on data and enhance the flexibility of security models to changing patterns of threats. The results can enhance the body of academic knowledge since they present empirical evidence on the relevance of optimisation and preprocessing in machine learning-based intrusion detection systems. Application-wise, the research paper provides useful advice on the implementation of scalable and adaptive security solutions that can be applied in contemporary network infrastructures [15].

Summing up, the paper has demonstrated the importance of adaptive security systems, whose realisation is provided by machine learning, in solving the modern cybersecurity issues. The

framework suggested above proves to be quite relevant in the future 6G and IoT environments, where the network traffic is highly mixed and heterogeneous. Its results endorse the shift to intelligent and optimised, data-centric network security architectures that can reduce the emerging and zero-day threats proactively.

8.5 LIMITATIONS

- **Dataset Limitations**

The research has been done on secondary network security data sets which had some limitations. Despite the popularity of benchmark datasets in the study of intrusion detection, they are limited in nature due to the fixed properties and attack contexts. Consequently, the datasets might not be an accurate reflection of real-time network traffic behaviour of working 6G and IoT systems. Also, traffic in static data sets might not capture the high change rates in attack patterns or time change variability in live networks and thus the degree to which the empirical results can be realistic can also be constrained [25].

- **Model and Experimental Constraints**

There were also a number of experimental constraints. Computational constraints constrained the search of a more general number of advanced deep learning architecture and large-scale ensemble models. The analysis therefore was restricted to the traditional and ensemble-based machine learning algorithms. Moreover, experimentation was performed in an offline setting based on historical data. It did not have the live deployment testing which ensured that the real time detection latency, system overhead and adaptive learning performance were not determined during continuous network traffic conditions [55].

- **Generalisation and Scalability Problems**

Issues of generalisation and scalability were discovered. The trained and tested developed models were based on certain datasets and network assumptions, which could influence their usability to other network designs, protocols or traffic distributions. The performance of the models may be affected by differences in device heterogeneity, network topology and security attributes among environments. Also, no empirical testing of scalability to large scale production environments with high throughput and low-latency requirements was done. These aspects underscore the necessity of validation in a variety of large-scale operational environments in order to make it robust and highly applicable in practice [47].

8.6 RECOMMENDATIONS

- **Practical Recommendations**

The implementation of the modern network application security is highly advised through machine learning-based intrusion detection systems. Rationalized security measures will be applied to overcome the disadvantages of rule-based and static solutions. It is recommended that exploratory data analysis-based preprocessing pipelines be used to gain a better insight into network traffic behaviour, class imbalance, and feature relevance prior to model training. This kind of preprocessing enhances the reliability of detection and lowers false alarm [60]. Structured rules that govern the deployment and maintenance of optimised machine learning models in operations should be made available to network administrators and security engineers. Constant model updating, retraining and monitoring of performances are necessary to achieve flexibility to changing threats as well as alterations in traffic dynamics in dynamic 6G and IoT networks [59].

- **Technical Recommendations**

Technically, ensemble learning models are suggested because they have a better generalisation ability as well as resistance to the imbalanced and complicated network traffic. Model configurations should be optimised using automated methods of hyperparameter tuning that will minimise human control. It is also advised to use adaptive learning techniques that would enable getting the model behaviour updated in relation to the appearance of new attack patterns to enhance long-term detection [57]. Moreover, security systems that are based on machine learning must be applied alongside existing intrusion detection and prevention systems to create hybrid security systems. This can be incrementally adopted and is more easily detectable and robust to operation.

- **Future Research Recommendation**

Further studies are needed on sophisticated deep learning algorithms to decide on intricate temporal and spatial patterns in network traffic. It is suggested that federated learning methods should be applied to allow collaborative threat detection without compromising data privacy on distributed network systems. More attention is to be paid to adversarial robustness mechanisms to secure machine learning models against evasion and poisoning attacks [58]. It is highly recommended to test using real-time traffic and live 6G and IoT environments to confirm scalability, latency, and real-world performance. Furthermore, the

growth in explainable artificial intelligence should be prioritised to enhance transparency, trust, and interpretability of security choices to the practitioners and stakeholders.

REFERENCES

- [1] Aledhari, M., Razzak, R. and Parizi, R.M., 2021. Machine learning for network application security: Empirical evaluation and optimization. *Computers & Electrical Engineering*, 91, p.107052.
- [2] Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L. and Rieck, K., 2022. Dos and don'ts of machine learning in computer security. In *31st USENIX Security Symposium (USENIX Security 22)* (pp. 3971-3988).
- [3] Atadoga, A., Sodiya, E.O., Umoga, U.J. and Amoo, O.O., 2024. A comprehensive review of machine learning's role in enhancing network security and threat detection. *World Journal of Advanced Research and Reviews*, 21(2), pp.877-886.
- [4] Choi, Y.H., Liu, P., Shang, Z., Wang, H., Wang, Z., Zhang, L., Zhou, J. and Zou, Q., 2020. Using deep learning to solve computer security challenges: a survey. *Cybersecurity*, 3(1), p.15.
- [5] Hindy, H., Brosset, D., Bayne, E., Seeam, A., Tachtatzis, C., Atkinson, R. and Bellekens, X., 2018. A taxonomy and survey of intrusion detection system design techniques, network threats and datasets.
- [6] Liu, T. and Ju, H., 2024. The Application of Machine Learning in Network Security Research. *Journal of Industrial Engineering and Applied Science*, 2(1), pp.8-12.
- [7] Mohamed, N., 2023. Current trends in AI and ML for cybersecurity: A state-of-the-art survey. *Cogent Engineering*, 10(2), p.2272358.
- [8] Sommer, R. and Paxson, V., 2010, May. Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE symposium on security and privacy* (pp. 305-316). IEEE.
- [9] Axelsson, S., 2000. The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security (TISSEC)*, 3(3), pp.186-205.

- [10] Bal, A., Chowdhury, S., Halder, N. and Cai, H., 2025, October. Machine Learning Algorithms for Spotting 6G Network Penetration of Different Attacks. In *2025 IEEE 16th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)* (pp. 777-783). IEEE.
- [11] Mhajne, A. and Henshaw, A. eds., 2024. *Critical Perspectives on Cybersecurity: Feminist and Postcolonial Interventions*. Oxford University Press.
- [12] Wang, X., Zhao, Z., Yi, L., Ning, Z., Guo, L., Yu, F.R. and Guo, S., 2024. A survey on security of UAV swarm networks: attacks and countermeasures. *ACM Computing Surveys*, 57(3), pp.1-37.
- [13] Teixeira, M.A., Zolanvari, M., Khan, K.M., Jain, R. and Meskin, N., 2021. Flow-based intrusion detection algorithm for supervisory control and data acquisition systems: A real-time approach. *IET Cyber-Physical Systems: Theory & Applications*, 6(3), pp.178-191.
- [14] Muraleedhara, P., 2024. The need for AI-powered cybersecurity to tackle AI-driven cyberattacks. https://www.researchgate.net/publication/382111991_The_Need_For_AI-Powered_Cybersecurity_to_Tackle_AI-Driven_Cyberattacks
- [15] Al Lail, M., Garcia, A. and Olivo, S., 2023. Machine learning for network intrusion detection—a comparative study. *Future Internet*, 15(7), p.243.
- [16] Songma, S., Netharn, W. and Lorpunmanee, S., 2024. Extending Network Intrusion Detection with Enhanced Particle Swarm Optimization Techniques. *arXiv preprint arXiv:2408.07729*.
- [17] Di Mauro, M., Galatro, G., Fortino, G. and Liotta, A., 2021. Supervised feature selection techniques in network intrusion detection: A critical review. *Engineering Applications of Artificial Intelligence*, 101, p.104216.
- [18] Chua, T.H. and Salam, I., 2022. Evaluation of machine learning algorithms in network-based intrusion detection system. *arXiv preprint arXiv:2203.05232*.

- [19] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J. and Ahmad, F., 2021. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1), p.e4150.
- [20] *Can Machine Learning Help Cybersecurity Systems?*, Stout Insights, Stout.com. Available: <https://www.stout.com/en/insights/article/can-machine-learning-help-cybersecurity-systems>. [Accessed: Jan. 27, 2026].
- [21] Randhawa, R.H., Aslam, N., Alauthman, M., Rafiq, H. and Comeau, F., 2021. Security hardening of botnet detectors using generative adversarial networks. *IEEE Access*, 9, pp.78276-78292.
- [22] Conner, C., de Visser, T., Loessberg, J., Sherman, S., Smith, A., Ma, S., Napoli, M.T., Pennathur, S. and Weld, D., 2018. Energy harvesting with a liquid-metal microfluidic influence machine. *Physical review applied*, 9(4), p.044008.
- [23] Thakkar, A. and Lohiya, R., 2022. A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions. *Artificial Intelligence Review*, 55(1), pp.453-563.
- [24] Ajagbe, S.A., Awotunde, J.B. and Florez, H., 2024. Intrusion detection: A comparison study of machine learning models using unbalanced dataset. *SN Computer Science*, 5(8), p.1028.
- [25] Kumari, D., Sinha, A., Dutta, S. and Pranav, P., 2024, May. A comprehensive analysis of various cyber attack datasets used for intrusion detection system. In AIP Conference Proceedings (Vol. 3164, No. 1, p. 030005). AIP Publishing LLC.
- [26] Hairani, H., Widiyaningtyas, T. and Prasetya, D.D., 2024. Addressing class imbalance of health data: a systematic literature review on modified synthetic minority oversampling technique

(SMOTE) strategies. *JOIV: International Journal on Informatics Visualization*, 8(3), pp.1310-1318.

[27] Vitorino, J., Silva, M., Maia, E. and Praça, I., 2025. Reliable feature selection for adversarially robust cyber-attack detection. *Annals of Telecommunications*, 80(3), pp.341-355.

[28] Tripathy, S.S. and Behera, B., 2025. Hyperparameter Tuning-Based Optimized Performance Analysis of Machine Learning Algorithms for Network Intrusion Detection. *arXiv preprint arXiv:2512.15779*.

[29] Huang, L., Dave, D., Cody, T., Beling, P.A. and Jin, M., 2025, November. From Capabilities to Performance: Evaluating Key Functional Properties of LLM Architectures in Penetration Testing. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing* (pp. 15890-15916).

[30] Francis, G.T., Sourì, A. and Inanç, N., 2026. A systematic review of metaheuristic based feature selection strategies for cyber-attack detection in the IIoT. *Artificial Intelligence Review*.

[31] Saeed, M.M., Saeed, R.A., Abdelhaq, M., Alsaqour, R., Hasan, M.K. and Mokhtar, R.A., 2023. Anomaly detection in 6G networks using machine learning methods. *Electronics*, 12(15), p.3300.

[32] Zhang, S., Yu, Y., Tao, X. and Wang, L., 2025. Towards Robust Video Anomaly Detection in 6G Networks: A Scene-Adaptive Framework. *IEEE Network*.

[33] Ferrag, M.A., Friha, O., Kantarci, B., Tihanyi, N., Cordeiro, L., Debbah, M., Hamouda, D., Al-Hawawreh, M. and Choo, K.K.R., 2023. Edge learning for 6G-enabled Internet of Things: A comprehensive survey of vulnerabilities, datasets, and defenses. *IEEE Communications Surveys & Tutorials*, 25(4), pp.2654-2713.

- [34] Alsubai, S., Umer, M., Innab, N., Shiaeles, S. and Nappi, M., 2024. Multi-scale convolutional auto encoder for anomaly detection in 6G environment. *Computers & Industrial Engineering*, 194, p.110396.
- [35] Rasheed, I. and Mostafa, H., 2025. Federated learning-based anomaly detection for zero-day attack prevention in 6G network slices. *Computer Networks*, p.111376.
- [36] Wang, R., 2025. Enhancing IoT data analytics with deep learning-based anomaly detection and real-time processing on edge devices. *Journal of Computational Methods in Sciences and Engineering*, p.14727978251346026.
- [37] Cuomo, M.T., Tortora, D., Colosimo, I., Celsi, L.R., Genovino, C., Festa, G. and La Rocca, M., 2023. Segmenting with big data analytics and Python: A quantitative exploratory analysis of household savings. *Technological Forecasting and Social Change*, 191, p.122431.
- [38] Poudel, P. and Paudel, S., 2025. Quantitative Trading Strategy, Backtesting, and Performance Analysis Using Python: A Data-Driven Analysis. *Quest Journal of Management and Social Sciences*, 7(2), pp.219-238.
- [39] Lavanya, A., Gaurav, L., Sindhuja, S., Seam, H., Joydeep, M., Uppalapati, V., Ali, W. and SD, V.S., 2023. Assessing the performance of Python data visualization libraries: a review. *Int. J. Comput. Eng. Res. Trends*, 10(1), pp.28-39.
- [40] Martin, H., Hong, Y., Wiedemann, N., Bucher, D. and Raubal, M., 2023. Trackintel: An open-source Python library for human mobility analysis. *Computers, Environment and Urban Systems*, 101, p.101938.
- [41] Zhang, Z., Zhang, Y., Jiang, X., Rao, N., Wu, M., Li, Y. and Peng, D., 2026. Enhancing Blood Fingerprints with Multicolored Conjugated Polymers: A Fluorescent Strategy with Python-Based Quantitative Assessment. *Journal of Fluorescence*, pp.1-15.
- [42] Ferreira, M., Belchior, A., Pinheiro, T., Alves, G. and Lopes, M., 2026. LiDRoSIS: An Automated MATLAB-Python Platform for Image Processing and Quantitative Analysis of Lipid Droplets and ROS in Irradiated Cells. *arXiv preprint arXiv:2601.14999*.
- [43] Tarrinha, S.P. and Marques, C.G.C., 2025. Heritage in Portugal: Analysing the Perspectives of Tour Operators and Tourists Using Python. *Emerging Trends in Information Systems and Technologies: WorldCIST 2025 Volume 2*, 2, p.151.

- [44] Vasa, V., 2025. Development and Assessment of an Adaptive Difficulty Snake Game in Python. *Radius: Journal of Science and Technology*, 2(1), p.251007.
- [45] Velte, P., 2023. Determinants and consequences of corporate social responsibility decoupling—Status quo and limitations of recent empirical quantitative research. *Corporate Social Responsibility and Environmental Management*, 30(6), pp.2695-2717.
- [46] Khandelwal, S. and Chaudhary, A., 2022. COVID-19 pandemic & cyber security issues: Sentiment analysis and topic modeling approach. *Journal of Discrete Mathematical Sciences and Cryptography*, 25(4), pp.987-997.
- [47] Semertzis, I., Rajkumar, V.S., Ştefanov, A., Fransen, F. and Palensky, P., 2022, May. Quantitative risk assessment of cyber attacks on cyber-physical systems using attack graphs. In *2022 10th Workshop on Modelling and Simulation of Cyber-Physical Energy Systems (MSCPES)* (pp. 1-6). IEEE.
- [48] Alomiri, A., Mishra, S. and AlShehri, M., 2024. Machine learning-based security mechanism to detect and prevent cyber-attack in IoT networks. *International Journal of Computing and Digital Systems*, 16(1), pp.645-659.
- [49] Mohammed, A.A., Islam, M.S. and Zubair, K.M., 2025. Threat intelligence for business analysts: Bridging the gap between security and strategy. *American Journal of Business*, 2(7).
- [50] Zurowski, S., Lord, G. and Baggili, I., 2022, August. A quantitative analysis of offensive cyber operation (OCO) automation tools. In *Proceedings of the 17th international conference on availability, reliability and security* (pp. 1-11).
- [51] Mahmood, R.K., Mahameed, A.I., Lateef, N.Q., Jasim, H.M., Radhi, A.D., Ahmed, S.R. and Tupe-Waghmare, P., 2024. Optimizing network security with machine learning and multi-factor authentication for enhanced intrusion detection. *Journal of Robotics and Control (JRC)*, 5(5), pp.1502-1524.
- [52] Ratilal, P.D. and Reddy, P.V., 2023. The Importance of Data Visualization in Exploratory Data Analysis. *Journal Of Advanced Zoology*, 44.
- [53] Liu, Y. and Wang, B., 2025. Advanced applications in chronic disease monitoring using IoT mobile sensing device data, machine learning algorithms and frame theory: a systematic review. *Frontiers in Public Health*, 13, p.1510456.

- [54] Elreedy, D., Atiya, A.F. and Kamalov, F., 2024. A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning. *Machine Learning*, 113(7), pp.4903-4923.
- [55] Ozkan-Okay, M., Akin, E., Aslan, Ö., Kosunalp, S., Iliev, T., Stoyanov, I. and Beloev, I., 2024. A comprehensive survey: Evaluating the efficiency of artificial intelligence and machine learning techniques on cyber security solutions. *IEEE Access*, 12, pp.12229-12256.
- [56] Abuali, K.M., Nissirat, L. and Al-Samawi, A., 2023. Advancing network security with AI: SVM-based deep learning for intrusion detection. *Sensors*, 23(21), p.8959.
- [57] Sharma, P. and Liu, H., 2020. A machine-learning-based data-centric misbehavior detection model for internet of vehicles. *IEEE Internet of Things Journal*, 8(6), pp.4991-4999.
- [58] Abiodun, O.I., Jantan, A., Omolara, A.E., Dada, K.V., Umar, A.M., Linus, O.U., Arshad, H., Kazaure, A.A., Gana, U. and Kiru, M.U., 2019. Comprehensive review of artificial neural network applications to pattern recognition. *IEEE access*, 7, pp.158820-158846.
- [59] Tang, L. and Mahmoud, Q.H., 2021. A survey of machine learning-based solutions for phishing website detection. *Machine Learning and Knowledge Extraction*, 3(3), pp.672-694.
- [60] Dandamudi, S.R.P., Sajja, J. and Khanna, A., 2025. Advancing cybersecurity and data networking through machine learning-driven prediction models. *International Journal of Innovative Research in Computer Science and Technology*, 13(1), pp.26-33.

APPENDIX

```
# Library Imports

# Core Libraries
import numpy as np
import pandas as pd
# Visualization
import matplotlib.pyplot as plt
import seaborn as sns
# Preprocessing
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split
# Feature Selection
from sklearn.feature_selection import SelectKBest, chi2
# Models
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
# Evaluation
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score,
    f1_score, confusion_matrix, classification_report
)
# Handling Imbalanced Data
from imblearn.over_sampling import SMOTE
# Hyperparameter Tuning
from sklearn.model_selection import GridSearchCV
# Warnings
import warnings
warnings.filterwarnings("ignore")
```

```
# Dataset Loading
# Load dataset
df = pd.read_csv('/content/6G_Network_Slicing_Security_DatasetR.csv')

# Display basic info
df.head()

df.info()
df.describe()

# Data Preprocessing

# Check missing values
df.isnull().sum()

# Fill numerical missing values with median
for col in df.select_dtypes(include=np.number):
    df[col].fillna(df[col].median(), inplace=True)

# Fill categorical missing values with mode
for col in df.select_dtypes(include='object'):
    df[col].fillna(df[col].mode()[0], inplace=True)
```

```
## Encode Categorical Variables
label_encoders = {}

for col in df.select_dtypes(include='object'):
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le

df

## Target Variable Separation

X = df.drop('anomaly_label', axis=1) # Features
y = df['anomaly_label']             # Target

## Feature

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Exploratory Data Analysis (EDA)

## Class Distribution

plt.figure()
sns.countplot(x=y)
plt.title("Attack vs Normal Traffic Distribution")
plt.show()
```

```

## Target Class Proportion (Pie Chart)
plt.figure(figsize=(6,6))
y.value_counts().plot.pie(
    autopct='%1.1f%%',
    startangle=90,
    shadow=True
)
plt.title('Proportion of Normal vs Malicious Traffic')
plt.ylabel("")
plt.show()

## Distribution of Key Network Features (KDE)

key_features = ['latency', 'bandwidth_utilization', 'packet_loss_rate']

plt.figure(figsize=(12,6))
for feature in key_features:
    sns.kdeplot(df[feature], label=feature, fill=True)

plt.title('Kernel Density Distribution of Key Network Metrics')
plt.xlabel('Value')
plt.ylabel('Density')
plt.legend()
plt.show()

## Pairwise Feature Relationships (Sampled)

sample_df = df.sample(500, random_state=42)

sns.pairplot(
    sample_df,
    hue='anomaly_label',
    vars=['latency', 'bandwidth_utilization', 'packet_loss_rate'],
    corner=True
)
plt.show()

```

```

## Correlation Heatmap
plt.figure(figsize=(12,8))
sns.heatmap(pd.DataFrame(X_scaled, columns=X.columns).corr(),
            cmap='coolwarm', linewidths=0.5)
plt.title("Feature Correlation Heatmap")
plt.show()

## Feature Distribution Example

plt.figure()
sns.histplot(df['latency'], kde=True)
plt.title("Latency Distribution")
plt.show()

## Feature Selection

selector = SelectKBest(score_func=chi2, k=10)
X_selected = selector.fit_transform(abs(X_scaled), y)

selected_features = X.columns[selector.get_support()]
print("Selected Features:", selected_features)

## Feature Importance Preview (Random Forest)

rf_temp = RandomForestClassifier(n_estimators=100, random_state=42)
rf_temp.fit(X, y)

importances = pd.Series(
    rf_temp.feature_importances_,
    index=X.columns
).sort_values(ascending=False)

plt.figure(figsize=(10,6))
importances.head(10).plot(kind='bar')
plt.title("Top 10 Important Features (Random Forest)")
plt.ylabel('Importance Score')
plt.show()

## Data Balancing (SMOTE)

smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X_selected, y)

print("Before SMOTE:", np.bincount(y))
print("After SMOTE:", np.bincount(y_resampled))

```

```

## Train-Test Split

X_train, X_test, y_train, y_test = train_test_split(
    X_resampled, y_resampled, test_size=0.2, random_state=42
)
# Machine Learning Model Implementations

## Baseline Model – Logistic Regression
lr = LogisticRegression()
lr.fit(X_train, y_train)
y_pred_lr = lr.predict(X_test)

## Decision Tree
dt = DecisionTreeClassifier(random_state=42)
dt.fit(X_train, y_train)
y_pred_dt = dt.predict(X_test)

## Random Forest (Optimised)
rf = RandomForestClassifier(random_state=42)
param_grid = {
    'n_estimators': [100, 200],
    'max_depth': [10, 20],
    'min_samples_split': [2, 5]
}
grid_rf = GridSearchCV(rf, param_grid, cv=3, scoring='f1')
grid_rf.fit(X_train, y_train)
best_rf = grid_rf.best_estimator_
y_pred_rf = best_rf.predict(X_test)

## Support Vector Machine

svm = SVC(kernel='rbf')
svm.fit(X_train, y_train)
y_pred_svm = svm.predict(X_test)
# Model Evaluation Metrics
def evaluate_model(name, y_test, y_pred):
    print(f"\n{name} Evaluation")
    print("Accuracy :", accuracy_score(y_test, y_pred))
    print("Precision:", precision_score(y_test, y_pred))
    print("Recall  :", recall_score(y_test, y_pred))
    print("F1 Score :", f1_score(y_test, y_pred))
evaluate_model("Logistic Regression", y_test, y_pred_lr)
evaluate_model("Decision Tree", y_test, y_pred_dt)
evaluate_model("Random Forest", y_test, y_pred_rf)
evaluate_model("SVM", y_test, y_pred_svm)

```

```
## Model Comparison Table
```

```
results = pd.DataFrame({  
    'Model': ['Logistic Regression', 'Decision Tree', 'Random Forest', 'SVM'],  
    'Accuracy': [  
        accuracy_score(y_test, y_pred_lr),  
        accuracy_score(y_test, y_pred_dt),  
        accuracy_score(y_test, y_pred_rf),  
        accuracy_score(y_test, y_pred_svm)  
    ],  
    'F1 Score': [  
        f1_score(y_test, y_pred_lr),  
        f1_score(y_test, y_pred_dt),  
        f1_score(y_test, y_pred_rf),  
        f1_score(y_test, y_pred_svm)  
    ]  
})
```

```
results
```

```
# Prepare comparison data
```

```
models = results['Model']
```

```
accuracy = results['Accuracy']
```

```
f1 = results['F1 Score']
```

```
import numpy as np
```

```
x = np.arange(len(models))
```

```
width = 0.35
```

```
plt.figure(figsize=(10,6))
```

```
plt.bar(x - width/2, accuracy, width, label='Accuracy')
```

```
plt.bar(x + width/2, f1, width, label='F1-Score')
```

```
plt.xlabel('Machine Learning Models')
```

```
plt.ylabel('Score')
```

```
plt.title('Comparison of ML Models for Network Attack Detection')
```

```
plt.xticks(x, models, rotation=15)
```

```
plt.ylim(0, 1)
```

```
plt.legend()
```

```
plt.grid(axis='y')
```

```
plt.tight_layout()
```

```
plt.show()
```

