NETWORK ANOMALY DETECTION THROUGH THE USE OF AI BASED TECHNOLOGIES

by

Tathagata Nandy
MS in AI/ML from LJMU
EPGP in AI/ML from IIITB
PGP in Management from IIM Bangalore
B.Tech in Comp Sc from CU
B.Sc in Physics from CU

DISSERTATION

Presented to the Swiss School of Business and Management Geneva

In Partial Fulfillment

Of the Requirements

For the Degree

DOCTOR OF BUSINESS ADMINISTRATION

SWISS SCHOOL OF BUSINESS AND MANAGEMENT GENEVA

OCTOBER, 2025

NETWORK ANOMALY DETECTION THROUGH THE USE OF AI BASED TECHNOLOGIES

by

Tathagata Nandy

Supervised by

Dr. Kamal Malik

APPROVED BY

dr. Jaka Vadnjal [illegible]

Dissertation chair

RECEIVED/APPROVED BY:

Renee Goldstein Osmic

Admissions Director

Dedication

To my loving mother late Krishna Nandy, you have been my pillar of strength and the constant reason for doing new things. Without your support nothing was possible.

Acknowledgments

Finishing this thesis has been a significant intellectual undertaking as well as a journey of philosophical and personal development.

I owe a debt of gratitude to Professor Kamal Malik, who served as my mentor and inspired me to question accepted knowledge and investigate difficult concepts. With her assistance, I was able to better understand the conceptual foundations of my research and develop my analytical abilities.

I am also incredibly grateful to my family, whose lively philosophical discussions and inquisitiveness inspired my interest in this subject. Their unwavering encouragement and faith in my academic endeavours have played a crucial role in my achievement.

This dissertation is a reflection of my work as well as the combined efforts of all the people who have had a personal or academic influence on me. I have learned from the trip how important it is to ask questions and how different viewpoints can enhance our comprehension of difficult problems.

ABSTRACT

NETWORK ANOMALY DETECTION THROUGH THE USE OF AI BASED TECHNOLOGIES

Tathagata Nandy
2025

Dissertation Chair: Jaka Vadnjal
Co-Chair: Aleksandar Erceg

Networking is one of the most fundamental aspect of computer infrastructure along with Servers and Storage. The impact of Artificial Intelligence on Computer Networking has been profound from the early days (Mistry et al., 2024) of networking.

Artificial intelligence is used to make the network more efficient (Umoga et al., 2024)and effective. Cloud computing-based analysis with AI for Edge computing (Umoga et al., 2024) is analysed with advanced AI and analytics methods. The growth and importance of networking as a domain in the past decade has coincided with the explosion of AI technologies. This has led to building AI for networking as well as networking for AI as two separate adjacencies. In this work AI for networking is examined with focus on classical and generative AI based technologies for network traffic classification and anomaly detection. The results are also compared with traditional methods like neural network based as well as classical statistical methods for anomaly detection methodologies. The work aims to provide the benefit of AI based technologies for intrusion detection and prevention which can be used to build a secure and robust network. The work looks at different class of machine learning technologies with multiple class of traffic and provides valuable insights. As part of this work close to twenty different machine learning algorithms along with Ten different publicly available dataset and provides the best combination for network traffic classification and anomaly detection. The research will provide notable insights to build a system for network anomaly detection as well as intrusion detection for the next generation of large scale and complex networks.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. CHAPTER 1: INTRODUCTION

1.1 Introduction

The concepts of anomaly detection and network traffic classification have evolved hand-in-hand with the broader development of computer networking and cybersecurity. Initially treated as distinct areas—one focusing on identifying unusual behaviour, the other on categorizing traffic flows—they have increasingly converged, particularly in the face of modern encrypted, multiprotocol, and high-volume network environments.

Network Anomaly Detection is an advanced cybersecurity approach that continuously keeps track of, analyses results across datasets for the KDD Cup with Neural behavioral patterns. It operates under the basic assumption that malicious activities, system crashes, or performance degradations are reflected as quantifiable changes in the network's communication patterns. It creates detailed baseline models of normal network activity through statistical examination of older traffic activity, protocol usage habits, bandwidth utilization metrics, and temporal communication streams. These baselines serve as points of reference for all future network activity that is measured and analyzed for potential anomalies.

Its operating structure supports real-time data aggregations that grab network packets, flow records, and metadata from various network levels. Sophisticated processing engines subsequently use advanced algorithms to examine the constant flow of network data, matching emerging patterns to predefined baselines to detect statistically significant changes. The system supports dynamic adaptation, repeatedly adjusting baseline models to reflect legitimate changes in network usage patterns while retaining sensitivity to actual security breaches.

1.2 Network Traffic Classification

In parallel with network evolution, the flow classification architecture has evolved over the last 20 years. It has to be noted that the initial flow classification was port-based. It transitioned to payload-based inspection as the same application (e.g., YouTube) began using multiple ports. This made the simple port-based classification insufficient. The next came payload-based classification(Finsterbusch et al., 2014), which was effective, but started having issues as encrypted traffic started to increase. The evolution continued with TLS certificate-based classification and eventually progressed to machine learning-based classification. ML-based classification also evolved with time. Simple statistical classification to advanced DL based classification to ensemble methods like XGS or Gradient Boost. The area is still evolving, with traditional and neural networks-based algorithms, and multiple new industrial and academic research papers are being published (Haque et al., 2022). Figure 1 shows the history of network traffic classification.

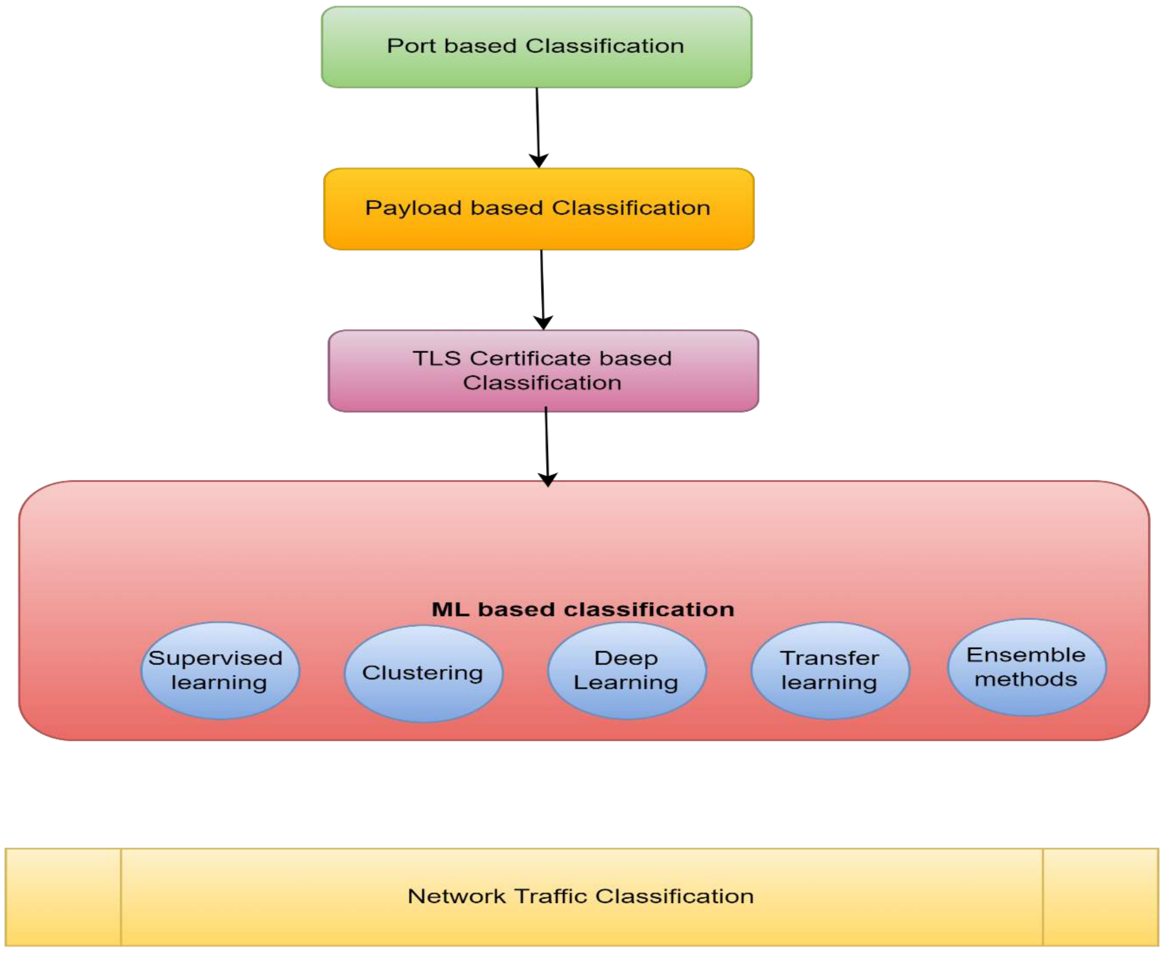
Port based Classification
Payload based Classification
TLS Certificate based Classification
ML based classification
Supervised learning
Clustering
Deep Learning
Transfer learning
Ensemble methods
Network Traffic Classification

Figure 1: Network Traffic Classification evolution

1.2.1 Reason for Network Traffic Classification

Network Traffic Classification has many reasons. The primary reason is that it started with traffic visibility and telemetry. Then it evolved to differentiated Quality of Service (QOS) for different classes of traffic. With the evolution of IOT and new devices, anomaly detection became another major reason for traffic classification. Lastly, application detection, application-based policies, as well as Intrusion Detection systems are some of the pressing needs to do network traffic classification. Figure 2 shows network traffic classification and why it is so valuable.

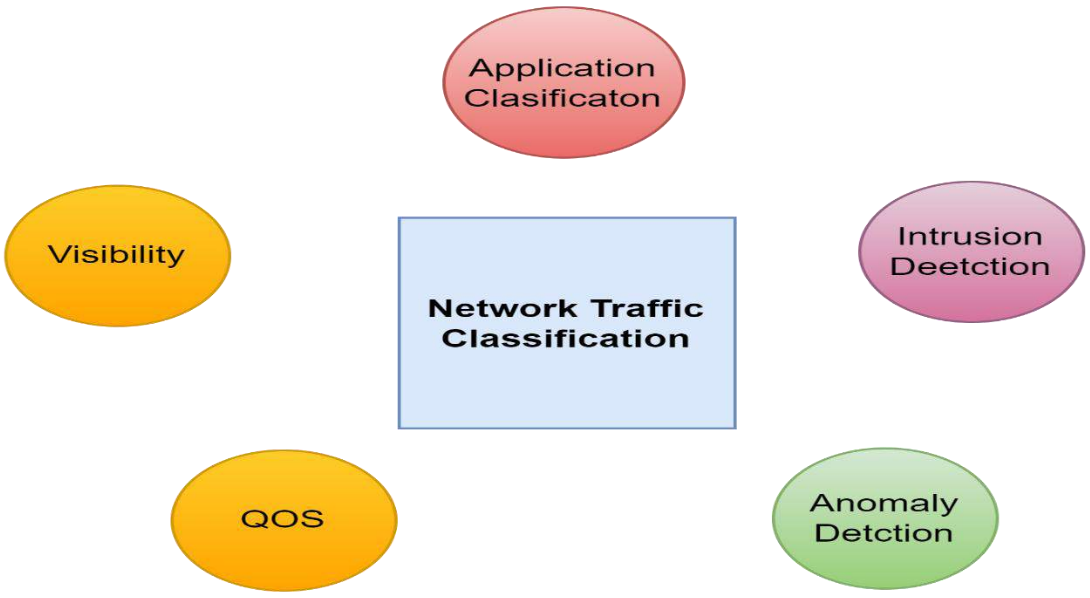

Figure 2: Different Uses of Network Traffic Classification

1.2.2 Historical Development of Network Anomaly Detection

History of Network Anomaly Detection. The development of network anomaly detection has been through four separate stages, which demonstrate the sophistication and changes of the cybersecurity attacks over the last forty years. In the early 1980s through the 1990s, most systems in the field used rule-based systems and used hand-crafted signatures and static heuristics to detect anomalies concerning expected network behaviour, with early intrusion detection systems such as Snort and Bro (since renamed Zeek) setting the standard of packet header and payload examination with a set of predefined attack signatures. A major shift of statistical modeling and shallow machine learning methods occurred in the 2000s, as researchers realized the inadequacy of fixed rule sets and started considering Gaussian Mixture Models, Principal Component Analysis, Support Vector Machines, decision trees and k-means clustering models, and a groundbreaking paper released in 2008 by T.T.T. Nguyen and Grenville Armitage showed that metadata at the flow level could successfully classify applications without examining packet contents, and is now considered the standard methodology of applying flow-based features to traffic classification The 2010s saw the rise of big data methods based on comprehensive datasets, e.g., CICIDS2017, UNSW-NB15, and BoT-IoT, that gave realistic, labeled traffic at new scales, making hybrid methods combining anomaly detection with traffic classification possible, and the growing popularity of encrypted communications and multiprotocol enterprise networks that made payload-based detection less effective and fastened the development of deep learning implementations like Autoencoders and Long Short-Term Memory networks to model temporal patterns and provide zero-day attacks that can be detected. The contemporary period, 2020-2027 marks the intersection of anomaly detectors and traffic classifiers, with AI-native designs using Transformers, Generative Adversarial Networks and Variational Autoencoders to replicate subtle variations in behavior in dynamic environments and payload inspection by protocols such as HTTPS and QUIC has rendered it impossible to inspect payloads, making it important that real-time classification and analysis is available at network edges using switches, gateways, and IoT hubs to support zero-trust security models and edge computing paradigms. Simultaneously, edge computing and zero-trust security models demand that decisions be made close to the user or device. This requires traffic to be classified and analysed in real-time at switches, gateways, or IoT hubs, blurring the lines between routing, classification, and security. Multicast network traffic classification (Gombao, 2025), along with IDS, is gaining interest as not much work has been done on this area.

1.3 Network Anomaly Detection

Network anomaly detection refers to the process of identifying unusual patterns or deviations in network traffic that do not conform to expected behaviour. These anomalies often indicate security threats such as unauthorised access, malware propagation, data exfiltration, denial-of-service (DoS) attacks, or performance degradation caused by misconfigurations or system faults. Let's take a close look at network anomaly detection methods over the years in Figure 3

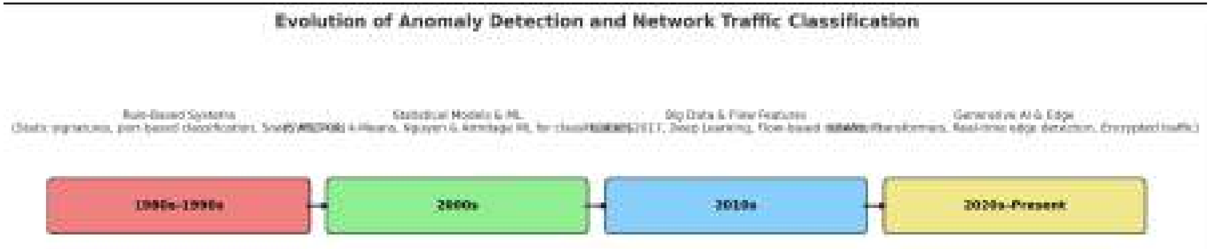

Figure 3: Network Traffic classification and Anomaly detection

Network anomaly detection has its origins in the broader field of network security and performance management, evolving alongside the growth of computer networks and the internet. As networks became more complex, with increasing numbers of connected devices and diverse applications, the need to monitor and protect these networks from unauthorized access, performance degradation, and security breaches became critical. This led to the development of network anomaly detection techniques. There has always been a deep interest in network anomaly detection through AI-based methods. Early work done by (Kaur et al., 2013) Focused on different methods for anomaly detection as a survey paper. Similar work has been done by (Bhattacharyya & Kalita, 2013). IOT devices and blockchain-based work (Golomb et al., 2018a) started gaining prominence as and when the related technologies evolved. As neural networks became the state of the art for AI (Liu et al., 2019) (Klarák et al., 2024)Proposed RNNs for anomaly detection of IP traffic. Finally, with GenAI and Reinforcement learning gaining popularity (Edozie et al., 2025) analyzed RL and GAN-based methods for anomaly detection for IP traffic. The next sections provide details of the trends across decades.

A. *Machine Learning and Data-Driven Approaches*: Machine learning (ML) techniques, both supervised and unsupervised, were applied to anomaly detection. Popular methods included:

- Clustering (e.g., K-Means, DBSCAN) for unsupervised anomaly detection.
- Classification (e.g., Decision Trees, Random Forest) for supervised anomaly detection.
- Deep learning models like Autoencoders, Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) networks for advanced sequence modelling.

B. *Advanced AI and Deep Learning Models*: Deep learning models, such as Variational Autoencoders (VAE), Generative Adversarial Networks (GAN), and Transformer models, brought significant improvements in anomaly detection accuracy. These models can automatically learn complex patterns in high-dimensional network traffic data, identifying both known and unknown anomalies. Real-time anomaly detection using these models has become feasible with the rise of high-performance computing and cloud-based solutions.

C. *Modern Network Anomaly Detection*: Today, network anomaly detection is an essential component of network security, powered by a combination of machine learning, deep learning, and big data analytics. It is used in various applications, including Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS), Network Performance Monitoring (NPM) for identifying congestion and latency issues, Cloud and Data Centre Security for detecting malicious traffic, IoT Security for monitoring connected devices and identifying compromised endpoints. Unlike signature-based intrusion detection systems (IDS) that rely on known attack patterns, anomaly detection techniques focus on identifying unknown or evolving threats — making them especially valuable in zero-day attack scenarios and dynamic network environments.

1.4 Need and Significance of Network Anomaly Detection

With the exponential growth of cloud computing, IoT devices, and 5G networks, network anomaly detection has become more critical than ever. Organizations rely on it to maintain network security, optimize performance, and ensure reliable service delivery in an increasingly complex and dynamic network environment.

i. State-of-the-Art Threat Detection Beyond Security

State-sponsored attacks, zero-day exploits, and advanced persistent threats are modern-day manifestations of cyber threats using advanced tricks to circumvent signature-based detection systems. These attacks tend to use valid network protocols and can sustain low-profile operations over long durations of time as they proceed with the rogue activities. Behavioural analysis capabilities are offered by network anomaly detection, whereby suspicious patterns are identified irrespective of the attack methodology, and can spot previously unknown threats that traditional security measures fail to identify. The capability of the system to understand the normal behaviour patterns of the network and raise alarms on abnormal behaviour makes it fundamental in detecting new attack vectors and new measures to exploit the network without the ordinary security measures.

ii. Business Continuity and Financial Protection

The average expense of data breaches is greater than millions of dollars when you include the remediation cost, regulatory fines, legal expenses and damage to reputation. Network anomaly detection is a type of early warning system that allows detecting potential security incidents at the early stages, allowing quick containment and minimizing the scope of breaches and their costs by a significant margin. In addition to security advantages, the system eliminates the expensive business interruptions as the system detects network performance problems, infrastructure failures and capacity issues before negatively affecting the critical business operations. This is a proactive surveillance feature that facilitates continuity in operations and service quality, which is critical in satisfying customers and maintaining a competitive stance in online business scenarios.

iii. Regulatory Compliance and Regulatory Legal Requirements

The modern regulatory frameworks, such as GDPR, PCI-DSS, HIPAA, and SOX, present a robust system of network surveillance as a core functionality in the list of requirements to comply with data protection. The organizations should prove they have regular observation of the network operations, detection of unauthorized access, and documentation of the incident to meet the requirements of the regulations and to avoid massive fines. Network anomaly detection offers the technical base needed to comply with it and also yields detailed audit trails needed by regulatory reporting. Risks of legal liability that come with poor security surveillance have posed a huge risk since courts continue to hold organizations responsible when security breaches that could have been averted by proper surveillance systems have been witnessed.

iv. Resource Optimization and operational Efficiency

Surveillance of the modern enterprise networks with thousands of devices and huge volumes of traffic is computationally infeasible and cost-prohibitive to security experts. Network anomaly detectors remove the need for human-operated threat identification and prioritization software, and allow security experts to apply expertise to investigating actual incidents and not to investigate standard network data. The more sophisticated systems offer contextual data and scoring of risks that simplify the incident response procedures and decrease mean time to threat containment. This automation feature enables organizations to have a full security coverage at minimal security personnel utilization, as well as to guarantee a uniform performance of monitoring in cresting network infrastructures.

1.5 Research Problem

The rapid growth of modern networks, including enterprise networks, cloud environments, IoT ecosystems, and 5G infrastructures, has significantly increased the complexity and diversity of network traffic. As networks become more dynamic, they are increasingly exposed to a wide range of security threats, including malware, distributed denial-of-service (DDoS) attacks, data breaches, unauthorized access, and misconfigurations. Traditional network security measures, such as static rule-based detection and signature-based Intrusion Detection Systems (IDS), have proven inadequate in effectively detecting and preventing sophisticated and evolving network anomalies. Multicast networks, which are widely used in applications such as IPTV, video conferencing, financial data distribution, and content delivery networks, pose additional challenges for anomaly detection due to their dynamic group membership, traffic replication, and complex routing protocols. Detecting anomalies in multicast traffic requires specialized methods that can account for group-based communication, membership changes, and multicast traffic optimization. Existing network anomaly detection methods suffer from several Problems and Limitations, which are as follows:

- Lack of Adaptability**:** Traditional anomaly detection models rely on predefined rules or static thresholds, making them ineffective in dynamically changing network environments.

- High False Positive Rate**:** Static detection methods and even some machine learning models often generate high false positives due to their inability to differentiate between normal and anomalous traffic patterns accurately.

- Inadequate Multicast Anomaly Detection**:** Existing solutions are primarily focused on unicast traffic, leaving multicast traffic, which is critical for many applications, poorly protected.

- Limited Scalability**:** Many AI-based anomaly detection models require significant computational resources, making them impractical for real-time detection in high-speed networks (e.g., 5G, IoT, cloud).

- Lack of Explainability: Advanced AI models (e.g., Deep Learning, Generative AI) often function as "black boxes," providing accurate anomaly detection but without explaining why an anomaly was detected, leading to trust and interpretability issues.

- Difficulty in Handling Diverse Traffic Types: Modern networks generate a wide variety of traffic types, including web traffic, video streams, VoIP, multicast, IoT device communication, and more. A single anomaly detection model may not effectively detect anomalies across all these traffic types.

- Zero-Day Attack Detection Challenge: Many existing methods fail to detect zero-day attacks (previously unknown threats), leaving networks vulnerable to new and evolving attack techniques.

Key Aspects of the Research Problem are:

a. How to develop an AI-based anomaly detection framework that can accurately identify anomalies in both unicast and multicast traffic?

b. How to ensure that the framework is scalable for high-speed networks (e.g., 5G, IoT, cloud) without compromising detection accuracy?

c. How to leverage advanced AI techniques (Generative AI, Deep Learning) while maintaining model explainability and trust?

d. How to minimize false positives and false negatives, ensuring that the anomaly detection system is both accurate and reliable?

e. How to design a flexible framework that can adapt to different network environments, including enterprise, cloud, IoT, and multicast networks?

f. How to provide actionable insights for network administrators, including root cause analysis of detected anomalies?

This research aims to address the critical problem of accurately detecting, classifying, and preventing network anomalies (including multicast anomalies) using advanced Artificial Intelligence (AI) techniques. The primary problem is the lack of a unified, adaptive, scalable, and explainable AI-based framework capable of detecting anomalies in both unicast and multicast network traffic, ensuring network security and performance across diverse network environments.

1.6 Purpose of Research and Research Questions

The long-term objective of this research is to develop a comprehensive framework and methodology for network traffic classification, anomaly detection, intrusion detection, and prevention using advanced Artificial Intelligence (AI) techniques. Network traffic classification, anomaly detection, and specifically multicast anomaly detection is critical for securing modern networks, where the sheer diversity and heterogeneity of network traffic pose significant challenges. This research aims to explore and evaluate various AI techniques, identify their strengths and limitations, and propose an optimized framework that ensures secure and efficient network traffic management.

Multicast networks, which are commonly used in IPTV, video conferencing, financial data distribution, and cloud services, present unique challenges in anomaly detection due to their dynamic membership, group-based communication, and complex traffic patterns. This research will specifically focus on developing methods for effective multicast anomaly detection, ensuring that multicast traffic is efficiently monitored, analyzed, and secured without affecting performance.

This work will systematically investigate the application of multiple AI technologies, including traditional machine learning, deep learning, and Generative AI (Gen AI) methods, to determine the most effective approach for network security. Given the evolving nature of network environments, this research will focus on creating a flexible, scalable, and adaptive framework that can cater to a wide range of network types, including unicast, multicast, IoT, and 5G infrastructures.

RQ1: Is there a generic algorithm that can be used for anomaly detection across all types of network traffic, including multicast traffic?

RQ2: Are AI-ML-based methods the most suitable for network traffic classification, multicast anomaly detection, and intrusion prevention?

RQ3: Are Generative AI (Gen AI)-based methods superior to traditional AI algorithms for network traffic classification, multicast anomaly detection, and anomaly detection in general?

RQ4: How do these advanced models handle complex multicast traffic patterns and dynamic group memberships?

RQ5: Is there scope for combining traditional algorithms with neural network-based algorithms and further integrating them with Generative AI to develop a robust anomaly detection framework for unicast and multicast traffic?

RQ6: How can the proposed multicast anomaly detection methods be effectively used to create a superior Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) for multicast and unicast traffic?

1.6.1 Research Objectives

- To construct a robust, adaptive, and scalable framework for network traffic classification, multicast anomaly detection, general anomaly detection, and intrusion prevention.
- To perform a comparative analysis of various AI techniques (traditional ML, deep learning, Gen AI) for multicast and unicast anomaly detection.
- To create an optimised hybrid model that combines the strengths of multiple AI approaches for superior anomaly detection.
- To introduce a practical methodology for integrating the proposed framework into real-world IDS and IPS systems, specifically addressing the unique characteristics of multicast networks.
- Building a comprehensive framework for Multicast Anomaly detection, which is missing in most of the industrial and academic research.

This research aims to contribute to the field of network security and anomaly detection by providing a practical and theoretically sound approach to securing network traffic, including multicast, using state-of-the-art AI techniques.

1.6.2 Impact on Futuristic Research and Innovation

The proposed research on network traffic classification, anomaly detection, multicast anomaly detection, and intrusion prevention using AI-based techniques is expected to have a profound and far-reaching impact on futuristic research and innovation in multiple domains of networking and cybersecurity. The following are the key areas where this study will drive innovation and inspire future research:

i. Advanced AI Models for Network Security

The study will pave the way for the development of next-generation AI models that can efficiently detect anomalies in complex and dynamic network environments, including multicast networks, 5G networks, IoT ecosystems, and cloud infrastructures.

Researchers will be encouraged to explore the application of advanced AI models such as Generative Adversarial Networks (GAN), Variational Autoencoders (VAE), Transformers, and Hybrid AI models for network security.

ii. Multicast Security and Performance Optimisation

By focusing on multicast anomaly detection, this research will highlight the importance of securing multicast traffic in video streaming, IPTV, financial data distribution, and cloud services. Future research can build on the proposed methodologies to enhance multicast performance, minimise latency, and improve the reliability of multicast communications.

iii. Hybrid AI Models for Enhanced Anomaly Detection

The study's emphasis on combining traditional algorithms, deep learning, and Generative AI models will inspire researchers to explore hybrid approaches for anomaly detection. This will drive innovation in designing models that can adapt to diverse network conditions, dynamically select the best algorithm, and achieve superior detection accuracy.

iv. Real-Time Anomaly Detection Systems

The research will contribute to the development of scalable, real-time anomaly detection systems capable of handling high-speed networks, including 5G, IoT, and edge computing. Researchers will be motivated to develop lightweight, optimised AI models for real-time detection without significant computational overhead.

v. Explainable AI in Network Security

The study's focus on integrating Explainable AI (XAI) techniques will promote transparency in anomaly detection, allowing network administrators to understand why specific anomalies were detected. Future research may explore more sophisticated XAI techniques that provide intuitive visual explanations for detected anomalies.

vi. Adaptive and Self-Learning Security Systems

The research will lay the foundation for the development of adaptive AI models that continuously learn from network traffic, automatically adjusting their detection capabilities without manual retraining. This will inspire the creation of self-learning intrusion detection and prevention systems that can evolve with changing network behaviours.

vii. Secure and Privacy-Preserving Anomaly Detection

The study will encourage future research on designing anomaly detection systems that can operate in encrypted network environments without violating user privacy. This may lead to the development of privacy-preserving anomaly detection techniques using homomorphic encryption, federated learning, and zero-knowledge proofs.

viii. Optimised IDS/IPS Systems for Diverse Network Environments

The proposed framework can be directly applied to develop next-generation Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) that support both unicast and multicast traffic. Researchers may focus on building IDS/IPS systems that are context-aware, protocol-independent, and capable of detecting both known and unknown threats.

ix. Enhanced Multicast Monitoring and Management Solutions

By providing methods for detecting multicast anomalies, the research will drive innovation in multicast monitoring tools, traffic simulation platforms, and diagnostic systems. Future solutions may focus on advanced multicast traceability, dynamic group management, and intelligent multicast route optimization.

x. Cross-Disciplinary Research Opportunities

The study will open up cross-disciplinary research opportunities, combining AI, networking, cybersecurity, and cloud computing. Researchers may explore the application of AI-driven anomaly detection in networked robotics, autonomous systems, smart cities, and secure IoT ecosystems.

xi. Policy Formulation and Standardisation

The insights gained from this research may influence the formulation of security policies, guidelines, and standards for network anomaly detection, especially in multicast and critical network environments. Standardization bodies (e.g., IEEE, IETF) may adopt the proposed framework as a reference for secure multicast management.

xii. Real-World Adoption in Enterprise Networks

The proposed framework will have a direct impact on the design and deployment of network security solutions in enterprise networks, including financial institutions, healthcare, cloud service providers, and telecom operators. Organizations may leverage the research outcomes to enhance their network security posture, protect critical infrastructure, and ensure regulatory compliance.

1.7 Summary

The research will drive significant advancements in network anomaly detection and security, fostering a new generation of intelligent, adaptive, and explainable network protection solutions. It will inspire future research on AI-driven network security, create new opportunities for cross-disciplinary innovation, and ensure that modern networks can be securely managed and monitored in an increasingly complex and connected world. The scope of this study is clearly defined to ensure a focused and practical approach to developing an AI-based network anomaly detection and prevention framework. By identifying out-of-scope areas, this study maintains its focus on developing a scalable, adaptive, and high-performance anomaly detection solution for IP-based unicast and multicast networks.

1.8 Organization of Thesis

Chapter 1: Introduction

This chapter introduces the fundamental concepts of network anomaly detection, emphasising its importance in securing modern network environments, including enterprise networks, cloud infrastructures, IoT ecosystems, and 5G networks. The introduction further explains the challenges associated with detecting anomalies in diverse network environments, including the complexity of multicast traffic, the need for real-time detection, and the difficulty of identifying zero-day attacks. It also defines the scope of the research, which focuses on developing an AI-driven framework capable of accurately detecting anomalies in both unicast and multicast traffic.

Chapter 2: Literature Review

This chapter reviews existing literature on network anomaly detection, covering traditional methods like threshold-based and rule-based approaches, which are limited by static configurations and high false positive rates. It explores the evolution of machine learning techniques, including supervised methods (Decision Trees, SVM) and unsupervised methods (K-Means, Isolation Forest), as well as deep learning models (Autoencoders, RNN, LSTM, Transformer) that excel in detecting complex patterns in network traffic.

Chapter 3: Research Methodology

This chapter outlines the research methodology used to develop an AI-based framework for network traffic classification, anomaly detection, multicast anomaly detection, and intrusion prevention. It begins with data collection, including real-world and synthetic network datasets, followed by data preprocessing to enhance model accuracy. Various AI techniques, including traditional Machine Learning (ML), Deep Learning, and Generative AI (Gen AI), are explored, with a focus on model selection, training, and evaluation. Feature engineering techniques are applied to extract relevant network features, including specialized features for multicast traffic. Models are trained using a train-test split, and their performance is evaluated using standard metrics such as accuracy, precision, recall, F1-score, and AUC-ROC.

Chapter 4: Analysis

In this chapter, a thorough analysis of the various datasets is conducted which are utilized in this Research , aiming to uncover inherent patterns, statistical properties, and behavioural trends that are critical to the design and evaluation of anomaly detection models. This includes an exploratory examination of network traffic attributes such as packet size, flow duration, byte count, and protocol usage across datasets like CICIDS2017, KDD Cup 1999, multicast flow data, and high-dimensional application-level chunks.

Chapter 5: Results and Discussion

This chapter presents the results of the AI-based network anomaly detection framework developed in this study, focusing on its performance in detecting anomalies across both unicast and multicast traffic. The results are presented through detailed performance metrics, including accuracy, precision, recall, F1-score, and Area Under the Curve (AUC-ROC) for each model evaluated. Traditional machine learning models (Decision Trees, Isolation Forest, Random Forest, SVM), deep learning models (Autoencoders, LSTM, Transformer), and advanced Generative AI models (GAN, VAE) are compared, highlighting their strengths and weaknesses.

Chapter 6: Conclusion and Recommendations

This chapter concludes the study, summarising the key findings and contributions of the research. The study successfully developed a comprehensive AI-based framework for network anomaly detection, capable of detecting anomalies in both unicast and multicast traffic using a combination of machine learning, deep learning, and Generative AI techniques. The hybrid model demonstrated superior performance, achieving high detection accuracy, low false positive rates, and effective multicast anomaly detection. Explainable AI (XAI) techniques enhanced model transparency, providing clear insights into anomaly detection decisions. The study addresses existing research gaps by providing a scalable, adaptive, and explainable solution suitable for diverse network environments, including enterprise, cloud, IoT, and 5G networks.

2. CHAPTER II: LITERATURE REVIEW

2.1 Introduction

Computer Networking has come a long way in the last few decades. Networks have grown, the types of traffic and volume have expanded and managing them has become a nightmare. The sheer variety and complexity of Network Traffic make it extremely interesting as well as a challenging vertical to manage. Securing a network is one of the major challenges for the network operator. Identification of Traffic in the Network or Network Traffic Classification is the first step in designing a robust, scalable and scalable network. The need for analytics in IP networking(Bachechi et al., 2022) Starting from the early days of networks and anomaly detection and prediction are of great importance in building a solid and robust network. More recent work in these areas has been done by (Klarák et al., 2024). The study presents a method for detecting and classifying defects using a combination of autoencoders and clustering techniques. IoT network anomaly detection is surveyed in (Hussain Kalwar & Bhatti, n.d.) This explores the application of deep learning models for classifying IoT network traffic. It reviews various architectures, including CNNs and RNNs, for their ability to capture and classify traffic patterns.

Network traffic classification and network anomaly detection are one of the topics where Artificial Intelligence (AI) based methods have become more popular over the years. One of the early works done in IP traffic classification was by (Zander et al., 2005a). The work looked at Unsupervised learning methods to classify traffic based on the Applications that are generating it. A general survey of encrypted traffic classification using deep learning methods (Wang et al., 2019b) is done, where the classification is done for intrusion detection and prevention and its application in the larger area of enterprise security. Similar work for encrypted traffic classification was done by (Cao et al., 2014a) . As artificial intelligence has gained more prevalence in the last decade, the majority of work in Network Anomaly detection has happened in this using AI only. As AI and ML world moves from Statistical to Deep Learning to Generative AI, the network traffic classification also follows the same path. The evaluation of statistical and deep learning methods for anomaly detection (Elmrabit et al.,

2020) has been done. The paper looked at the state-of-the-art ML algorithms and analysed their ability to detect anomalous behaviours in networking, using popular datasets. The Random Forest (RF) method demonstrated the highest performance by beating all other algorithms. This was consistent across all datasets evaluated. (Ahmed et al., n.d.-a) looked at different approaches using Machine Learning for Network Anomaly detection. One-Class Neighbour Machine (OCNM) estimates minimum volume sets to detect anomalies. It uses sparsity measures (e.g., k-th nearest neighbour distance) to identify points in the minimum volume set. It also looks at Kernel-based Online Anomaly Detection (KOAD). Both OCNM and KOAD algorithms effectively detected anomalies in sequences of images from the camera network. The paper (Landauer et al., 2023) analyzes the use of various neural network-based techniques in faster anomaly detection of log data, confirming the superiority of these methods over traditional machine learning in handling unstable data formats and detecting unexpected log events.

Network Anomaly detection for IOT devices is a pretty important topic (Hwang et al., 2020) proposed a model named D-PACK. This model is designed to detect malicious traffic in networks, particularly in the context of IoT traffic, where countless devices are constantly communicating with each other. D-PACK analyses CNN for profiling network traffic patterns. It also incorporates an unsupervised deep learning model known as an Autoencoder. CNN helps to identify the characteristics of the traffic. The Autoencoder then filters out any abnormal traffic effectively. The experimental results demonstrate that D-PACK is highly effective at detecting malicious network traffic. It achieves an accuracy rate close to 100%. Additionally, it maintains an exceptionally low false-positive rate of just 0.83%. (Eskandari et al., 2020a) proposes an anomaly-based intrusion detection system designed for IoT edge devices. It can be deployed on cost-effective IoT gateways, leveraging edge computing to detect cyber threats close to the data sources. It utilizes machine learning to model the normal behaviour of incoming traffic and detect deviations indicative of cyber threats. (Golomb et al., 2018b) proposes a non-AI-based method that uses blockchain for network anomaly detection. They proposed a lightweight, scalable framework for network anomaly detection. It utilizes blockchain technology for distributed and collaborative anomaly detection, ensuring a trusted and secure model. (Ullah & Mahmoud, 2021a) Again proposes a model that uses CNN for anomaly detection in IoT networks. The model is validated through various datasets and showed high accuracy in every parameter in which models are evaluated (recall, F1 score, precision, etc). The proposed model includes 1D, 2D, and 3D CNNs for anomaly detection. The suggested CNN model achieves better results than conventional machine learning approaches, surpassing them in accuracy and various performance metrics. (Nguyen et al., 2018) proposes a smart and controlled self-learning network anomaly detection system for different types of IoT traffic. It proposes a model called DÏOT, which is an autonomous, self-learning distributed system that uses device-type-specific communication profiles for anomaly detection, leveraging federated learning for aggregating behaviour profiles efficiently.

(Vaswani et al., 2017) Changed the overall Gen AI landscape with the introduction of the Transformer Architecture. It utilizes self-attention mechanisms to draw global dependencies between input and output, allowing for parallelization and improved efficiency. Transformer Architecture led to the growth of GenAI, and every domain started using it in some form or another. Network anomaly detection using generative AI(Gen AI) has also gained a lot of attention in the last few years. A few of the Gen AI works, including those that use transformers, are analyzed.

Intrusion detection using a transformer-based model was evaluated in (Nguyen et al., 2023) for in-vehicle Intrusion Detection. The transformer-based attention network provides a robust solution for in-vehicle intrusion detection, which overcomes the shortcomings of the existing methods, thereby enhancing the detection of various cyberattacks on the CAN bus. The application of transfer learning further improves the model's adaptability and performance across different datasets. The model uses transfer learning to improve its performance on anomaly detection. (Shin et al., 2023) presents a unique framework for detecting and isolating various anomalies that are present in time series data using a Transformer-based Generative Adversarial Network (GAN) named AnoFormer. Further interesting work has been done by (Shao et al., 2025a) which leverages the unique properties of the transformer, which is the self-attention mechanism. It uses Tab Transformer to capture intricate patterns as well as dependencies within tabular data, making it well-suited for network intrusion detection. (Liu et al. 2023) also proposes a transformer-based intrusion detection system. It introduces a novel under-sampling method which uses KNN over-sampling using the SMOTE method to balance the complete dataset and improve detection accuracy for overlapping classes. This model works on the NSL-KDD dataset to perform binary classification and achieved 88.7% accuracy and an F1-score of 88.2%. It also uses multi-class classification and achieved 84.1% accuracy and an F1-score of 83.8%. (Kim & Pak, 2023). It is an academic paper that presents a unique and non-obvious method for detecting network and traffic intrusion. The authors propose a technique that transforms Network Intrusion Detection System (NIDS) datasets into 2D images using various image transformers, which can then be processed by various neural network-based models. The work aims to enhance the performance of existing neural network-based network intrusion detection systems.

The general trend of network anomaly detection-based studies is about clustering IP Flows and then detecting anomalies. Major studies on this have been done using deep learning techniques. (Fotiadou et al., 2021) uses pfSense software to monitor network traffic logs through different applications, like network application services. Deep Learning model like LSTM is used to identify anomaly detection offline. The analysis is more on network logs and not the actual traffic flows. (Nassif et al., 2021) provides a survey paper on various network anomaly works. It actually has analysed two hundred and ninety papers with twenty-nine different models. Unsupervised Models for IOT traffic classification are done by (Eskandari et al., 2020b) in their work for unlabeled traffic. The majority of the analysis is done on unlabeled traffic. A common gap in past studies is the combination of traditional deep learning methods along with generative AI-based techniques to identify network anomalies. The power of generative AI techniques can be used to generate synthetic data as well as to do proper IP flow classification. Generative AI and transformer-based architecture are comparatively new, and their application for IP Flow Anomaly is not well researched.

2.2 Literature Review Theories

The role of AI in networking has always been prominent, as it was studied by (Jaber, 2022). The study details the use of AI to enhance network performance, improve operational efficiency, as well as optimise network management. AI techniques such as genetic algorithms, neural networks, and decision-making processes like Markov Decision Processes (MDP) are explored. The work specifically emphasises deep learning’s role in processing sequential data using recurrent neural networks (RNNs) and analysing image data through convolutional neural networks (CNNs). Finally, Deep reinforcement learning (DRL) is observed as an important and useful tool for network automation and bandwidth optimisation. Similar comparison work, but its impact on security is done by(Thesis et al., 2022) In their work. The work concludes that AI-based technologies play a very significant role in modern network security by automating user processes, detecting threats early in the cycle, as well as improving overall network efficiency. There are ethical implications and potential biases in AI systems that must be carefully managed. Lastly (Wu et al., n.d.)Proposes the integration of computer networks with neural networks to create an intelligent system capable of autonomously managing and optimising network operations. This modern and unique AI-based approach improves network performance, network security, and operational efficiency, thereby providing a robust solution for modern enterprise networks which are undergoing digital transformation.

In this context of, the work and the impact of AI and ML technologies on network anomaly detection and prediction are analysed right from early statistical machine learning to deep learning to generative AI is investigated. The analysis will help us in shaping the scope and direction of this research.

2.1.1. Statistical AI methods for Network Anomaly Detection

Research work on network anomaly detection and classification started with non-ML techniques to start with. They were mostly based on Port (indicating the Application/Protocol type) classification. (Dainotti et al., 2012) In their very early study looked at various algorithms for anomaly detection, starting from early port and the payload-based approaches to the new era of machine learning. Network traffic anomaly detection through cascading

K-Means clustering and C4.5-based decision trees (Muniyandi et al., 2012). The K-Means + C4.5 combination showed much higher detection accuracy and a substantially lower false positive rate compared to individual algorithms like SVM, Naïve Bayes, and K-NN. The cascading method significantly improves anomaly detection by leveraging the strengths of both algorithms. Combining multiple algorithms for creating a framework was proposed by (Shon et al., 2005a) that proposes the use of Genetic Algorithm (GA) for feature selection, and on top of that use of an Unsupervised method like Support Vector Machine (SVM) for further packet categorization and classification. The idea is to optimize the feature selection process using GA and then apply SVM for detecting anomalies in network traffic. A similar hybrid architecture was proposed by (Shon & Moon, 2007a) which combines supervised and unsupervised learning. This model provides the detection capabilities of a supervised model while handling novel attacks like an unsupervised model. The paper highlights the importance of a hybrid model, which is followed by many researchers. Another successful application of SVM for traffic classification and anomaly detection was done (Yuan et al., 2010a) which showed promising results with internet data. More recent work has been done by (Schummer et al., 2024) The paper explores the development and evaluation of a machine learning-based system for detecting network anomalies, specifically focusing on point anomalies within network traffic. Anomaly detection is critical for maintaining network integrity, detecting unauthorised access, network crashes, and unusual traffic patterns. The study aims to create a robust anomaly detection system using machine learning, capable of identifying and classifying network anomalies accurately.

2.1.2. Deep Learning and Network Anomaly Detection

As neural networks started to gain prominence (Bouzida & Cuppens, n.d.) did the early investigation comparison between decision trees and neural networks for detecting network intrusions, with a focus on anomaly detection. The results show that neural networks excel at generalization, and decision trees are more effective for detecting both known and new attack types. CNN-based analysis (Kwon et al., n.d.) was used for anomaly detection using popular datasets NSL-KDD, Kyoto Honeypot, and MAWILab. CNN models show competitive detection accuracy, but performance may vary based on the dataset and further optimizations. Similar studies using Recurrent Neural Networks (RNN) are done in (Radford et al., 2018), which proposes the use of Long-Short Term Memory (LSTM) for detecting and isolating network anomalies. The conclusion was that the LSTM-based anomaly detection model demonstrated the ability to detect previously unseen attack patterns. Another combination techniques are evaluated in (Aneetha, 2012), where the paper proposes a hybrid approach that combines standard k-means clustering with a method called Self-Organizing Map (SOM) for network anomaly detection. The work shows that combining the neural network and clustering approach improves anomaly detection accuracy, especially for detecting DDoS attacks. (Pradhan et al., 2012) proposes fully Connected Neural Networks (FCN) along with generative AI-based models like Variational Autoencoder (VAE), along with Sequence-to-Sequence models combined with LSTM (Seq2Seq-LSTM) to detect anomalies in network flows. Slowly, from deep learning models, the investigation moved to generative AI-based models, which are investigated in the next section.

2.1.3. Generative AI and network anomaly detection

Generative AI techniques are different from the traditional discriminative AI techniques as they are primarily used for data generation, data classification, with their efficacy much better than traditional models. (Vaswani et al., 2017) . Their revolutionary Transformer model indicated that self-attention mechanisms can replace traditional deep learning algorithms like recurrent neural networks and convolutional layers in sequence transduction tasks, thereby offering improved parallelisation and unmatched performance. Anomaly detection using Transformer architecture (Manocchio et al., 2024) has been used to compare the performance of lightweight transformer-based models with much larger and complex architectures like GPT and BERT for Intrusion Detection, terming them as NIDS (Network

Intrusion Detection System). The general observation has been that while the large models could achieve a similar level of performance as the shallower models, the large size and comparatively much lower overall throughput make them less optimal for most NIDS tasks. Network anomaly using SDN (Software Defined Networking) controller (Ezeh & de Oliveira, 2023) discusses the application of GAN-based ensemble methods for anomaly detection that happens in a modern network of today, which is managed by a software-based controller, also called Software-Defined Networking (SDN) environment. The authors propose a unique controller-based framework that incorporates several components across the detection chain. The authors also compare their GAN ensemble approach with other one-class anomaly-detection algorithms. The authors conclude that GAN-based algorithms show sufficient promise in detecting network anomalies on different kinds of traffic in a network. This is done after training a very robust and large traffic dataset that forms the discriminator network. This is used to identify and detect network events and separate them from anomalous and unusual network events. They also suggest that future work should focus on creating more robust datasets that include more attack classes and testing a framework based on multiple controllers. Time Series-based anomaly detection (Shin et al., 2023) was proposed using a GAN-based network which internally also uses Transformer architecture. This internally uses a masking method which is two-step. This is a very powerful tool for time series anomaly detection. It outperforms traditional methods and other deep learning models, providing a robust solution for various applications. GAN-based imbalanced malicious (Cao et al., 2022)algorithm works by dividing the original sampled session traffic. This will look into the traffic and break it into three parts. Once it divides into three parts, it extracts the Markov matrix from them. This is used to form an image, which is a three-channel one and has its own characteristics. This conversion reshapes the initial session data structure into a standardized-length matrix, offering an extensive depiction of network behaviour. It ensures consistent representation across all traffic flows. This uniformity enhances the analysis and understanding of traffic characteristics. (Tang et al., 2022) The proposed Markov-GAN method effectively balances and enhances datasets, improving the generalization and classification performance of models for malicious encrypted traffic. IOT traffic-based classification is done in (Shahid et al., 2020), which discusses the application of Generative Deep Learning in the context of Internet of Things (IoT) Network Traffic Generation. The researchers introduce an innovative approach for creating a series of packet sizes that replicate actual IoT device patterns. Their technique effectively models the traffic characteristics of real-world IoT systems. This method aims to closely simulate the authentic behaviours observed in IoT communication. This is achieved by integrating an autoencoder with a Generative Adversarial Network (GAN). Autoencoder-based methods for network traffic anomaly detection and prevention are proposed in (Torabi et al., 2023). The paper introduces an innovative strategy for detecting anomalies within cloud computing networks by utilizing autoencoders. Instead of conventional approaches that summarize reconstruction errors into a single metric, this model evaluates the error as a multidimensional vector. Each component of the vector represents the error for a specific input feature, which serves as an indicator for anomaly identification or categorization. This approach is designed to enhance the accuracy and effectiveness of detecting and classifying anomalies. Other important work in this area is done by (Klarák et al., 2024), which presents a method for detecting and classifying defects using a combination of autoencoders and clustering techniques. The research work on network anomaly detection clearly shows a pattern of how it evolved across years. (Rao et al., 2024) proposes a novel approach for network anomaly detection using a Hybrid Convolutional Neural Network (CNN) and Generative Adversarial Network (GAN) model. This hybrid architecture combines the feature extraction capabilities of CNN with the generative and discriminative capabilities of GAN, making it highly effective in detecting complex network anomalies. The study addresses the limitations of traditional anomaly detection methods, which struggle with detecting sophisticated and evolving cyber threats. Figure 4 shows how AI evolved, and network anomaly detection also moved with it. In our approach, we will try and combine all the approaches and build the base network anomaly detection methodologies.

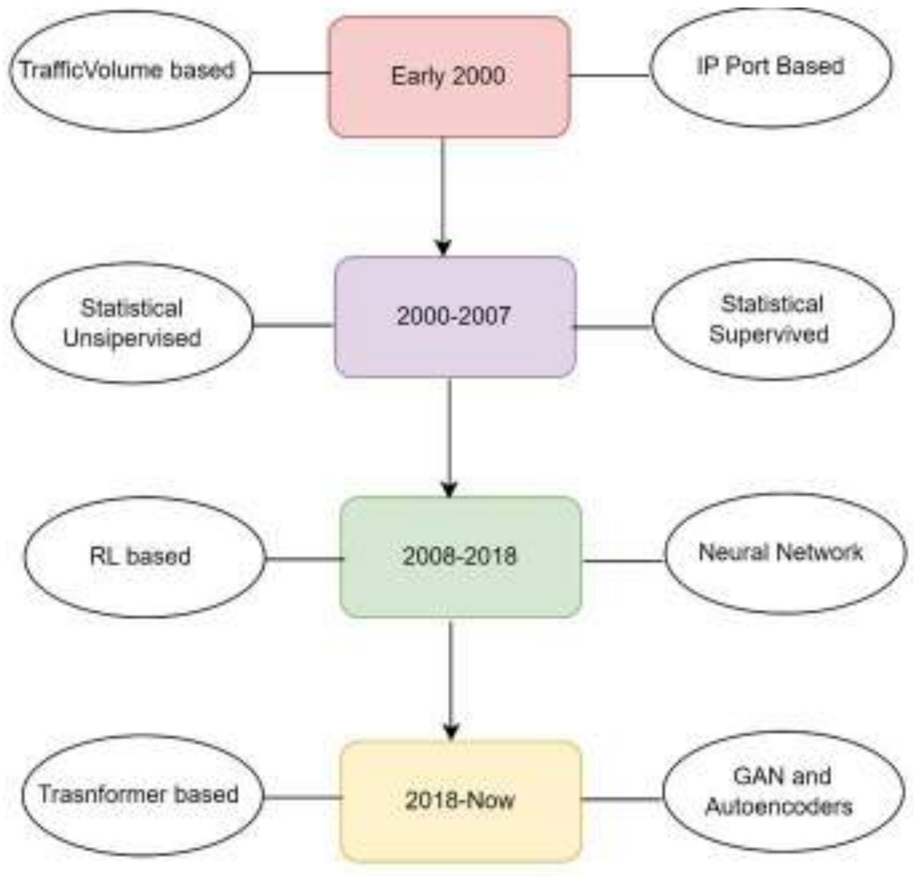

Figure 4: Evolution of Network Anomaly Detection over the Years

2.1.4. Combining different methods

There has been some work that has combined different methods to combine neural networks with Gen AI techniques (Rao et al., 2024) introduces a hybrid framework that integrates Convolutional Neural Networks (CNN) with Generative Adversarial Networks (GAN) to detect anomalies in network traffic. The CNN component is used for high-level feature extraction, while the GAN generates synthetic normal traffic to train the model and distinguish anomalies. The approach demonstrates superior performance in detecting anomalies with high detection rates and low false positives compared to traditional methods. Similarly, (Iliyasu & Deng, 2022) proposed novel GAN-based which they call NGAN for intrusion detection in networks. The proposed approach includes a small number of malicious samples during the training phase. This technique aims to enhance detection accuracy and minimise the rate of false alarms. The proposed approach is to create a weakly supervised model, which helps in modelling both benign and malicious behaviour, improving robustness against evolving cyber threats. The experimental results show that N-GAN outperforms traditional reconstruction-based anomaly detection methods. Other work like (Tang et al., 2020) in which the authors introduce a neural network for anomaly detection, employing a dual auto-encoder GAN (DAGAN) architecture tailored for industrial uses like automated optical inspection (AOI). This DAGAN model tackles the problem of imbalanced samples by incorporating a dual auto-encoder with skip-connections, enhancing both reconstruction capabilities and training stability. The proposed approach is evaluated across various datasets. It shows superior performance compared to other GAN-based models for anomaly detection, particularly when working with limited training data. The model undergoes evaluation using several datasets. It consistently outperforms other GAN-based models designed for anomaly detection in the majority of categories. Its advantage becomes even more pronounced when there is a scarcity of training data. The results indicate the model's effectiveness in challenging scenarios with limited data availability. The proposed (Fu et al., 2022) GANAD method addresses the limitations of existing GAN-based anomaly detection approaches, which were designed primarily for data synthesis rather than detection. GANAD uses a WGAN-based architecture with gradient penalty and spectral normalization to stabilize training and enhance performance. This method emphasizes effectively capturing the distribution of minority abnormal data. It achieves higher detection accuracy compared to other advanced techniques while also lowering computational requirements. LSTM based analysis is done in (Niu et al., 2020) presents a VAE-GAN model based on LSTMs for detecting anomalies in time-series data. The model trains an encoder, generator, and discriminator simultaneously to leverage both reconstruction error and discrimination ability. By efficiently mapping real-time data to latent spaces, the approach enhances both detection speed and accuracy. The results show that it outperforms traditional methods in identifying anomalies within industrial time-series datasets. Lastly, (Li et al., 2019) introduce MAD-GAN, an unsupervised method aimed at identifying anomalies in multivariate time-series data. It employs LSTM architectures in both the generator and discriminator to effectively capture time-related patterns within the dataset. The innovative aspect of this model is the combination of reconstruction and discrimination losses to identify unusual behaviour. Evaluation on real-world datasets, such as SWaT and

WADI, demonstrates its superior performance in detecting anomalies in cyber-physical environments compared to conventional and other GAN-based techniques.

In the next section, all of the major work in Network Anomaly detection in last few years using AI-ML methods has been summarized in a table.

2.2.Previous work details

A summary of all the previous work, along with their key areas and Research Gaps, is highlighted in Table 1

Table 1: Related Research Comparison

Year	Paper	Comments
2025	(Ghajari et al., 2025)	**Key Aspects:** The paper introduces an HDC-based method for IoT anomaly detection, encoding traffic features into hyper vectors for efficient classification. On the NSL-KDD dataset, it outperforms traditional ML models with up to 86.21% accuracy, showing scalability, resilience, and suitability for IoT devices. **Missing Information:** The study is limited to NSL-KDD, without tests on real-world or high-throughput IoT traffic. It omits protocol-specific analysis and comparisons with deep learning or hybrid methods, reducing its generalizability.
2025	(Edozie et al., 2025)	**Key Aspects:** The paper provides a comprehensive review of AI-based anomaly detection in telecom networks, covering deep learning and hybrid models (CNNs, RNNs, LSTMs, Autoencoders, GANs, GNNs, Federated Learning, RL). It discusses telecom-specific challenges such as real-time high-volume traffic, distributed architectures, and

		data privacy, while comparing supervised, unsupervised, and semi-supervised approaches, datasets, case studies, and performance metrics. **Missing Information:** Despite its breadth, the review is literature-focused with no experimental validation or benchmarking on standardized datasets. It lacks a unified performance framework, practical deployment architectures, and deeper treatment of feature engineering, protocol-level traffic categorization, and cross-domain transfer learning—limiting its guidance for real-world implementation.
2025	(Yang et al., 2025)	**Key Aspects:** The paper proposes an LLM-based anomaly detection system for cloud traffic, combining Transformer self-attention with an anomaly detection layer and transfer learning for adaptability. Tested on CICIDS 2017, it achieves higher accuracy, fewer false positives, and faster inference than several baseline models. **Missing Information:** The study is limited to one dataset, without testing in real-time, large-scale, or encrypted/multicast cloud environments. It lacks detailed feature engineering, comparisons with advanced hybrid models, and analysis of deployment scalability in data centers.
2025	(Ness et al., 2025)	**Key Aspects of the Study:** This paper proposes a GAN-based semi-supervised anomaly detection method for software-defined networks

		(SDNs). The GAN generates synthetic normal traffic to train a discriminator that can detect deviations as anomalies. Tested on the NSL-KDD dataset, the method achieves higher detection rates and lower false positives compared to Random Forest, SVM, and KNN, while requiring fewer labelled samples. **Missing Information**: The approach is evaluated on a single dataset and lacks testing on modern, high-speed, or encrypted SDN traffic. It does not explore feature extraction, protocol-level behavior, or deployment performance in real-time SDN controllers. Comparisons with recent deep hybrid or attention-based models are absent.
2025	(Liu et al., 2025)	**Key Aspects of the Study:** This paper introduces a privacy-preserving hybrid ensemble model for network anomaly detection, combining KNN, SVM, XGBoost, and ANN. Advanced preprocessing techniques address imbalanced and small datasets, while privacy is ensured through federated learning, SMPC, and differential privacy. The ensemble achieves 94.3% accuracy, outperforming individual models on both binary and multi-class classification tasks. **Missing Information:** The evaluation is limited to a single dataset and does not assess performance on real-time, high-speed, or encrypted network traffic. The study omits detailed protocol-specific behavior analysis and

		does not compare against recent deep hybrid or attention-based methods. Scalability and deployment considerations in production network environments are not explored.
2025	(Gombao, 2025)	**Key Aspects:** The paper proposes an intrusion detection system (IDS) tailored for multicast traffic, combining flow-based analysis with a lightweight edge-friendly classifier. Tested on simulated datasets, it detects multicast-specific attacks faster and with lower resource use than general-purpose IDS tools. **Missing Information:** The study is limited to synthetic multicast traffic, leaving performance in real-world large-scale networks uncertain. It does not address encrypted multicast, integration with IPsec, comparisons with deep learning IDS, or scalability to high-throughput environments.
2025	(Prasad et al., 2025)	**Key Aspects:** The paper presents a threat detection framework for SDN-based multicast systems, combining a hybrid CNN–LSTM model with explainable AI. CNNs extract spatial features, LSTMs capture temporal patterns, and LIME/SHAP provide transparency. Embedding the IDS in the SDN controller enables low-latency responses, making it suitable for real-time applications like live video streaming. **Missing Information:** The evaluation is limited to controlled multicast traffic, leaving real-

		world scalability, encrypted environments, and large-scale deployments untested. SHAP and LIME introduce overhead, raising concerns for high-volume use. The study also omits comparisons with newer transformer-based methods and does not address integration challenges in production-grade SDN controllers.
2025	(JLiu et al., 2025)	**Key Aspects:** The paper introduces a graph-based anomaly detection method for IoT networks, representing devices and their traffic as graphs. Using a GCN, it detects subtle structural deviations and leverages sparse labeling to adapt to environments with limited attack data. Tests on public IoT datasets show improved detection, particularly for unseen attack types, compared to baseline ML models. **Missing Information:** The study is limited to one dataset and does not evaluate performance in real-time or high-speed IoT settings. It lacks discussion on encrypted traffic, comparisons with newer attention-based graph models, and scalability to large heterogeneous networks. Integration with real-world monitoring systems is also unexplored.
2025	(Shao et al., 2025)	**Key Aspects:** The paper proposes a federated learning framework for anomaly detection, where local models share parameters instead of raw traffic data. This preserves privacy, reduces bandwidth, and achieves accuracy close to centralized training while handling data heterogeneity.

		Missing Information: The study is limited in scale, with few nodes tested and no validation in high-speed or encrypted traffic scenarios. It also overlooks risks like model poisoning, comparisons with advanced privacy-preserving methods, and real-world deployment challenges.
2025	(Jin et al., 2025)	**Key Aspects:** The paper proposes a CNN–LSTM model for protocol classification, where CNNs capture spatial patterns and LSTMs handle temporal dependencies. Tested on Universidad Del Cauca's network data, it achieves up to 98.1% accuracy, outperforming standalone CNNs and LSTMs. **Missing Information:** The study is limited to a single dataset and does not explore diverse, encrypted, or high-speed traffic scenarios. It also lacks analysis of real-time deployment, scalability, and comparisons with newer attention-based or transformer models.
2025	(Park et al., 2025)	**Key Aspects:** The paper proposes a CNN–BiLSTM model for raw packet sequence classification, where CNNs extract local features and BiLSTMs capture bidirectional dependencies. On the ISCX VPN-nonVPN dataset, it achieves over 97% accuracy, outperforming traditional ML and prior deep learning approaches, without manual feature engineering. **Missing Information:** The study is limited to a single dataset and does not evaluate adaptability to broader encrypted traffic or diverse

		environments. It also omits real-time performance tests, deployment considerations like latency and resource usage, and comparisons with newer transformer-based models.
2025	(Abbasi et al., 2025)	**Key Aspects:** The paper presents a Tansformer-based model for network traffic classification, using self-attention to capture long-range dependencies in packet sequences. On the ISCX VPN-nonVPN dataset, it achieves 98.7% accuracy, shows robustness to class imbalance, and reduces preprocessing by operating directly on packet-level data. **Missing Information:** The study is limited to a single dataset and does not evaluate generalization to other protocols, encrypted traffic, or high-speed environments. It also lacks analysis of inference latency, memory use, deployment feasibility, and comparisons with hybrid Transformer–CNN/LSTM models.
2025	(Antari et al., n.d.)	**Key Aspects:** The study benchmarks ML, deep learning, Transformer, and LLM models for network traffic classification across Web, Browsing, IPSec, Backup, and Email using 30,959 flows from Arbor Edge Defender devices. Transformers achieved the best accuracy (98.95%), with XGBoost close behind, while LLMs like GPT-4o and Gemini showed strong few-shot improvements over zero-shot, especially for simpler categories.

		Missing Information: The evaluation is limited to one enterprise dataset and does not test diverse, encrypted, or high-speed traffic. It also omits real-time scalability, hybrid model comparisons, adaptive retraining, and solutions for persistent misclassifications in complex classes like IPSec and Backup.
2025	(Lu et al., 2025)	**Key Aspects:** The paper evaluates LSTM, BiLSTM, and GRU models for encrypted traffic classification on the ISCX VPN-nonVPN dataset, using flow-based statistical features. BiLSTM achieved the best accuracy (94.12%), showing the advantage of bidirectional temporal modeling for VPN and non-VPN application identification. **Missing Information:** The study is restricted to a single dataset and does not test adaptability to other encryption types or varied network conditions. It also lacks evaluation of real-time scalability, computational overhead, and comparisons with newer transformer-based or attention-enhanced recurrent models.
2023	(Lypa et al., 2023)	**Key Aspects:** The paper reviews feature extraction tools for AI-based IDS, comparing packet-, flow-, and application-level approaches such as CICFlowMeter, Wireshark, Argus, Snort, and Zeek. Results show Argus and Zeek as more effective, with Zeek slightly outperforming CICFlowMeter on CIC-IDS2017 binary classification tasks.

		Missing Information: The study focuses only on feature extraction and does not assess full IDS performance across diverse environments. It omits evaluations in high-speed, encrypted, or multicast settings, as well as integration with deep learning, streaming analytics, and real-world deployment challenges.
2025	(Serag et al., 2025)	**Key Aspects:** The paper proposes ML models for QoS classification in IP networks using flow-level features, with Gradient Boosting achieving the best accuracy. The goal is to automate QoS assignment to enhance resource allocation and traffic engineering. **Missing Information:** The study is limited to one dataset and does not test in dynamic, high-speed, or encrypted environments. It also omits comparisons with deep learning models and lacks discussion of real-time deployment or SDN integration.
2022	(Shahraki et al., 2022)	**Key Aspects:** The paper applies active learning for anomaly detection, selectively labeling the most informative network traffic data to reduce labeling effort while maintaining accuracy. Strategies like uncertainty sampling and query-by-committee allow continuous model improvement in dynamic environments. **Missing Information:** The study focuses narrowly on labeling and does not cover broader traffic classification, feature extraction, or protocol-level analysis. It also ignores

		challenges with encrypted or multicast traffic and real-time, high-speed deployments.
2022	(Li et al., 2022)	**Key Aspects:** The study introduces an SSL/TLS traffic classification method that uses entropy and randomness features from ciphertext packets to distinguish encryption protocols. This lightweight approach enables protocol identification without decryption, offering efficiency for encrypted traffic analysis. **Missing Information:** The method does not employ standard ML techniques and ignores key traffic features like flow metrics, timing, and volume. These omissions limit its ability to detect anomalies or classify complex encrypted traffic in high-speed networks.
2021	(Aouedi et al., n.d.)	**Key Aspects:** The study uses K-means clustering to group network traffic and Recursive Feature Elimination (RFE) to select key features for anomaly detection. Anomalies are identified as deviations within clusters, providing a scalable method for dynamic networks. **Missing Information:** The work is limited to K-means and RFE, without exploring other unsupervised methods like DBSCAN, Isolation Forest, or Autoencoders. This narrow scope reduces its generalizability and overlooks potentially more effective approaches.
2021	(Sarhan et al., 2021)	**Key Aspects:** The paper introduces a dataset built from Cisco NetFlow records for ML-based

		intrusion detection, focusing on flow-level features and including attack types like DDoS and probing. It aims to connect raw telemetry with ML workflows by offering data tailored for supervised classification. **Missing Information:** The dataset is tied to Cisco NetFlow, limiting generalization to other telemetry formats, and lacks diversity such as encrypted or multicast traffic. It also does not address class imbalance, data augmentation, or benchmarking against standard public datasets.
2021	(Aouedi et al., 2021)	**Key Aspects:** The study examines the effect of unlabeled data on ML model performance in network traffic classification, benchmarking supervised and semi-supervised strategies. It highlights how labeled data availability influences classification outcomes compared to existing approaches. **Missing Information:** The work is restricted to a narrow classification scenario, limiting generalizability. It does not address anomaly detection, diverse traffic or attack types, or extensions to unsupervised and real-time contexts.
2021	(Manjunath et al., 2021)	**Key Aspects:** The paper proposes converting IoT network traffic into video streams and analyzing them with a time-distributed CNN-LSTM model. This cross-domain method bridges multimedia processing and network

		security, improving intrusion detection accuracy in IoT environments. **Missing Information:** The study is limited to IoT traffic and does not test broader applicability to enterprise or general networks. Data-to-video transformation adds computational complexity, and the lack of benchmarking on standard datasets reduces comparability with mainstream IDS methods.
2021	(Ahmed et al. 2021., 2021)	**Key Aspects:** The study classifies 53 popular online applications using KNN, Random Forest, and ANN, focusing on application-layer traffic. ANN achieved the highest accuracy, showing the strength of deep learning for traffic classification. **Missing Information:** The work is limited to supervised methods and does not explore hybrid or unsupervised approaches. It also lacks diverse feature sets, scalability testing, and real-time deployment analysis for practical use.
2021	(Afuwape et al., 2021)	**Key Aspects:** The paper compares multiple ML algorithms for classifying VPN and non-VPN traffic, including Decision Trees, Random Forests, and ensemble methods. Gradient Boosting performed best in certain categories, highlighting the effectiveness of supervised techniques for encrypted and non-encrypted traffic. **Missing Information:** The study is limited to supervised methods and does not explore

		unsupervised or semi-supervised approaches for evolving traffic. It also overlooks the impact of labeling quality and detailed feature interactions, which are critical for robust encrypted traffic analysis.
2021	(Arfeen et al., 2021)	**Key Aspects:** The paper compares three ensemble learning algorithms for encrypted traffic classification, addressing challenges without payload visibility. XGBoost achieved the highest accuracy, showing the strength of ensemble models for structured encrypted network data. **Missing Information:** The study is limited to ensemble methods and does not consider hybrid or alternative models. It also omits real-time performance evaluation, broader feature diversity, and feature importance analysis for deeper insights.
2020	(Aceto et al., 2020)	**What it is about -** A Deep Learning application is used for network traffic analysis. The analysis focusses on the identification and classification of encrypted mobile traffic. The study concludes that DL based traffic classification provides satisfactory output for encrypted traffic. The study also focusses on the challenges of traffic classification using various ML techniques. **What is missing-** This study focused on Deep learning for Mobile traffic and not a generic campus network traffic classification.

2020	(Salman et al., 2020)	**Key Aspects:** The paper applies deep learning to network traffic analysis, focusing on classifying encrypted mobile traffic flows without payload inspection. It highlights the potential of DL models to address challenges that limit traditional ML techniques in encrypted environments. **Missing Information:** The study is limited to mobile networks and may not generalize to enterprise or campus settings. It does not consider architectural diversity, multicast traffic, or dynamic behaviors common in large-scale wired networks.
2020	(Aureli et al., 2020)	**Key Aspects:** The paper studies ML techniques for traffic classification, addressing the challenge of imbalanced datasets with a semi-supervised learning approach. It improves classifier robustness where labeled data is scarce, a common issue in real-world scenarios. **Missing Information:** The work focuses only on class imbalance and does not consider encryption, data rate variability, or high-volume traffic. These omissions limit its applicability to broader, real-world classification challenges.
2021	(Fotiadou et al., 2021)	**What it is about-** This work uses pfsense software to monitor traffic logs through different applications like firewall, routing, NAT etc. It uses DL techniques like LSTM etc to identify anomaly detection offline and feed the data back.

		What is missing – This is more of a traffic log analysis and not actual packet/flow analysis. The methods might not work for a high-volume forwarding plane where traffic flows are not logged.
2021	(Nassif et al., 2021)	**What it is about-** This is a survey paper of 290 research papers looking at 29 different models for anomaly detection and recommending the best one. **What is missing** – This does not look at a specific data set and algorithm on a problem definition.
2021	(Ullah & Mahmoud, 2021b)	**What it is about-** CNN based model is used to detect anomalies in large IOT data sets. **What is missing** – Works at a specific category of problems of IOT devices which use a protocol like MQTT for communication.
2020	(Eskandari et al., 2020b)	**What it is about-** In this work, unsupervised learning methods like one class classifier is used to detect anomalies for IOT traffic. The majority of the analysis is done on unlabeled traffic. **What is missing** – Narrow class of traffic is analyzed. The concept can be generalized for non IOT traffic as well.

2019	(Wang et al., 2019b)	**Key Aspects:** The paper surveys deep learning methods for encrypted mobile traffic classification, covering CNNs, RNNs, and GANs. It highlights their strengths and limitations while stressing the growing role of DL in handling encrypted traffic. **Missing Information:** The review focuses only on DL and excludes traditional ML approaches and hybrid feature sets. Its insights are limited to mobile networks, with little applicability to general-purpose or wired traffic scenarios.
2019	(Rezaei & Liu, 2019a)	**Key Aspects:** The paper surveys deep learning techniques for network traffic classification, reviewing CNNs, RNNs, LSTMs, and transfer learning approaches. It highlights the evolution of methods and compares their applicability across traffic scenarios. **Missing Information:** As a survey, it does not propose new methods or experiments and omits dataset-specific insights and implementation challenges. It also lacks exploration of hybrid ML–DL strategies, making the contribution more foundational than practical.
2019	(Ring et al., 2019)	**Key Aspects:** The paper applies Generative Adversarial Networks (GANs) to network traffic classification, proposing techniques to convert categorical flow features into continuous representations. This enables GANs to better learn from discrete traffic datasets and extends their use in network telemetry.

		Missing Information: The study is narrowly focused on adapting older categorical datasets and does not provide a full framework for large-scale or real-time traffic classification. It overlooks key issues such as encryption, protocol diversity, and feature selection, limiting its broader applicability.
2019	(Liu et al., 2019)	**Key Aspects:** The study applies Recurrent Neural Networks (RNNs) to classify encrypted network traffic, showing their strength in modeling temporal dependencies from flow-based features. It demonstrates the usefulness of RNNs for traffic identification when payload data is unavailable. **Missing Information:** The work is limited to RNNs and does not consider alternative or hybrid architectures that could improve accuracy. It also overlooks the role of diverse feature sets, reducing generalizability across varied traffic types and environments.
2019	(Aceto et al., 2019)	**Key Aspects:** The paper applies deep learning to classify encrypted mobile traffic, showing that DL models can extract meaningful patterns where payload-based methods fail. It demonstrates the effectiveness of privacy-preserving traffic analysis. **Missing Information:** The study is limited to mobile traffic with a narrow set of classes and does not test broader enterprise or campus

		networks. This restricts the generalizability of its findings to more diverse scenarios.
2019	(\rr et al., 2019)	**Key Aspects:** The study evaluates CNN and ResNet models for network traffic classification, using metrics like F1-score to compare performance across classes. It shows how deep neural networks can handle structured, flow-based tasks effectively. **Missing Information:** The work is limited to supervised methods and does not explore unsupervised or hybrid approaches. It also overlooks latent traffic features, reducing generalizability to complex or evolving network patterns.
2019	(Rita et al., 2019)	**Key Aspects:** The paper applies a backpropagation neural network for traffic classification, enhancing QoS by accurately identifying traffic patterns. The results are used to guide network management strategies for better service delivery. **Missing Information:** The study is limited to supervised learning and does not explore unsupervised or hybrid approaches. It also overlooks anomaly detection use cases, reducing applicability in more complex environments.
2019	(Rezaei & Liu, 2019b)	**Key Aspects:** The paper proposes a multi-task learning model that classifies traffic while also predicting auxiliary metrics like bandwidth and flow duration. This joint framework improves

		contextual understanding of network flows for smarter traffic management. **Missing Information:** The study is limited to a narrow flow-classification use case and does not generalize to diverse traffic or environments. It also omits real-time performance, encrypted traffic handling, and anomaly detection, reducing practical applicability.
2018	(Liu et al., 2018)	**Key Aspects:** The paper introduces a method to improve efficiency in capturing and labeling mobile traffic, reducing manual effort and cost in creating datasets for ML training. This supports more effective classification of mobile network data. **Missing Information:** The study is limited to mobile traffic labeling and does not extend to general, encrypted, or enterprise network scenarios. Its scope is narrow, making the approach problem-specific rather than broadly applicable.
2018	(Yu et al., 2018)	**Key Aspects:** The paper proposes a QoS engineering framework for SDNs that combines Deep Packet Inspection with semi-supervised ML. A hybrid multiclass classifier enhances traffic identification and supports intelligent QoS provisioning. **Missing Information:** The work is specific to SDNs and does not generalize to other network architectures. It focuses on QoS optimization while leaving out broader traffic classification,

		anomaly detection, and applicability to enterprise or campus networks.
2018	(Rojas et al., 2018)	**Key Aspects:** The paper applies clustering and ML models to analyze IP-based OTT application traffic, using a robust flow dataset. SVM showed the best performance, demonstrating the value of supervised and unsupervised learning for application-level behavior analysis. **Missing Information:** The study is limited to OTT traffic and does not address broader network classification scenarios. It excludes diverse traffic types, encrypted flows, and enterprise environments, reducing generalizability.
2018	(Sharma et al., 2018)	**Key Aspects:** The paper analyzes a flow-based DNS dataset to identify compromised hosts using various machine learning algorithms. The study focuses on detecting anomalies in DNS traffic patterns that may indicate malicious activity, leveraging flow-level features to support automated threat detection. **Missing Information:** The work is specifically targeted at DNS anomaly detection and does not address broader network traffic classification or intrusion detection scenarios. Its focus on a single protocol limits its applicability to more comprehensive network security solutions involving multiple traffic types and behaviors.

2018	(Pacheco et al., 2018)	**Key Aspects:** The paper surveys emerging trends in applying ML to network traffic classification, covering key developments, challenges like encryption and imbalance, and techniques used over time. It provides a broad view of how ML has evolved for handling dynamic traffic patterns. **Missing Information:** As a survey, it lacks empirical experiments or dataset-based comparisons. The work remains conceptual, offering limited practical guidance for benchmarking or implementation.
2017	(Lotfollahi et al., 2017)	**Key Aspects:** The paper presents an early deep learning framework using Autoencoders and CNNs for automated feature extraction and traffic classification. It eliminates manual feature engineering, showing the promise of DL models for distinguishing VPN and non-VPN traffic. **Missing Information:** The study relies on a limited feature set and does not explore combining diverse inputs. It also omits hybrid approaches that could improve accuracy and adaptability across varied traffic environments.
2017	(Lopez-Martin et al., 2017)	**Key Aspects:** The paper applies CNNs, RNNs, and their combination for IoT traffic classification, showing the value of spatial and sequential feature learning.

		Missing Information: It does not explore diverse feature sets or provide guidance for scaling the approach beyond IoT environments.
2017	(Shafiq et al., 2017)	**Key Aspects:** The paper surveys ML techniques for traffic classification, comparing models like C4.5, Naïve Bayes, SVM, and Bysenet. It offers a foundational view of early ML performance in different classification contexts. **Missing Information:** The study uses a limited feature set and lacks depth in analyzing diverse traffic types. It also provides no problem-specific recommendations, limiting practical applicability.
2017	(Shi et al., 2017)	**Key Aspects:** The paper studies feature extraction for transport layer service classification, using PCA with classifiers like SVM to improve efficiency and accuracy. It shows how dimensionality reduction can refine the feature space for traffic analysis. **Missing Information:** The work is limited to a small set of algorithms and does not test broader models. It also overlooks diverse or combined feature sets that could enhance robustness across varied traffic conditions.
2017	(Vlăduţu et al., 2017)	**Key Aspects:** The paper explores ML as an alternative to DPI for traffic classification, using K-Means and Decision Trees on uni- and bidirectional flows. It demonstrates the feasibility of reducing reliance on packet-level inspection through ML-based methods.

		Missing Information: The study tests only a narrow set of clustering algorithms and omits other unsupervised approaches. A broader comparison could have revealed more effective or scalable techniques for varied traffic conditions.
2017	(Ahmed et al., n.d.-b)	**Key Aspects:** The paper compares ML techniques for anomaly detection, focusing on the One-Class Neighbor Machine and a kernel-based online method. It highlights lightweight, adaptive models for real-time deviation detection in network traffic. **Missing Information:** The study is limited to a small set of algorithms and excludes broader ML or DL approaches. It also omits diverse features and hybrid strategies, reducing applicability to complex or large-scale scenarios.
2015	(Li et al. 2015)	**Key Aspects:** The paper proposes a hybrid approach combining DPI with ML models like Decision Trees for application-level traffic classification in SDN. A multiclass classifier embedded in the SDN controller enhances traffic awareness and control. **Missing Information:** The study is specific to SDN and does not provide a generalized ML framework. It also lacks evaluation on diverse networks and traffic types, limiting applicability beyond SDN environments.
2015	(Namdev et al., 2015)	**Key Aspects:** The paper surveys supervised and unsupervised ML techniques for traffic

		classification, comparing 7–8 algorithms across different traffic types. It provides a broad view of their relative strengths and limitations. **Missing Information:** The study offers limited analysis of feature sets and does not explore hybrid or ensemble methods. This reduces its practical value for improving classification in real-world deployments.
2014	(Cao et al. 2014)	**Key Aspects:** The paper briefly explores encrypted traffic classification, outlining challenges and proposing simple strategies that avoid payload inspection. It emphasizes the difficulty of traffic analysis under encryption. **Missing Information:** The study lacks depth in methods, experiments, and evaluation. It does not provide a comprehensive framework, limiting its applicability to broader traffic classification scenarios.
2013	(Marpaung et al., 2013)	**Key Aspects:** The paper studies L7-based application-layer traffic classification in firewalls, focusing on Adobe RTMP. It shows how protocol-specific features can improve firewall rule enforcement and traffic management. **Missing Information:** The work is limited to a single protocol and excludes ML-based methods. Its narrow scope reduces generalizability to broader applications and diverse network types.

2013	(Omar et al., 2013a)	**Key Aspects :** The paper explores L7-based application-layer traffic classification in firewalls, focusing on Adobe RTMP. It demonstrates how protocol-specific traits can enhance firewall enforcement and traffic management. **Missing Information:** The work is limited to one protocol and does not include ML-based methods. Its narrow scope reduces applicability to broader traffic classification tasks.
2013	(Fahad et al., 2013)	**Key Aspects:** The paper evaluates feature selection techniques for internet traffic classification, aiming to boost accuracy while reducing computational cost. It shows how selecting relevant features supports more efficient training and inference. **Missing Information:** The focus is on feature selection alone, without comparing classification algorithms. This limits insights into end-to-end performance across different ML models.
2012	(Dainotti et al., 2012)	**Key Aspects:** The paper compares port-based, payload inspection, and early ML techniques for traffic classification, providing one of the foundational evaluations in the field. It structured an early understanding of classification strategies and their trade-offs. **Missing Information:** The study uses a limited feature set and does not explore advanced or hybrid methods. Its foundational scope limits

		relevance for modern large-scale, encrypted, or complex network environments.
2012	(Bujlow et al., 2012)	**Key Aspects:** The paper proposes C5.0 for network traffic classification, evaluating its supervised learning performance against other models. It demonstrates the algorithm's suitability for distinguishing different traffic types. **Missing Information:** The study is limited to supervised comparisons and does not consider hybrid or unsupervised techniques. It also lacks diverse feature analysis, restricting applicability to complex or evolving traffic.
2011	(Nguyen et al. 2012)	**Key Aspects:** The paper introduces a flow-segmentation method for traffic classification, splitting flows into sub-flows by features like volume, direction, and duration. Naïve Bayes and Decision Trees are tested, showing effectiveness for application-specific traffic like gaming and VoIP. **Missing Information:** The study is limited to a small set of applications and does not capture the diversity of modern networks. Its findings may not generalize to today's high-volume, encrypted, or cloud-based traffic.
2011	(Ubik et al. 2010)	**Key Aspects:** The paper applies the C4.5 algorithm to classify network flows across speeds of 100 Mbps, 1 Gbps, and 10 Gbps. It

		analyzes how traffic characteristics vary with speed and tests the model's adaptability. **Missing Information:** As an older study, it has limited relevance for today's high-speed, encrypted networks. It depends on labeled data and does not explore modern ML or adaptive learning methods.
2010	(Yuan et al., 2010b)	**Key Aspects:** The paper is an early study applying SVM for traffic classification, using flow-level features to distinguish different traffic types. It highlights the potential of supervised learning in this domain. **Missing Information:** The work is limited to SVM, which requires labeled data and is resource-intensive for large datasets. It does not explore alternative, unsupervised, or hybrid approaches, reducing scalability and flexibility.
2010	(Soysal et al. 2010)	**Key Aspects:** The paper evaluates Bayesian Networks, Decision Trees, and Multilayer Perceptrons for traffic classification across services like P2P, HTTP, CDN, FTP, DNS, and SMTP. It provides insights into how these models perform across different categories. **Missing Information:** The study depends on labeled data and uses a limited feature set and traffic scope, reducing generalizability. It does not explore unsupervised or hybrid methods, limiting practical flexibility.
2009	(Lu et al. 2009)	**Key Aspects:** The paper proposes a botnet-focused traffic classification method using

		behavioral and structural analysis to detect malicious patterns without payload inspection. It highlights an early approach to identifying botnet activity in networks. **Missing Information:** The method does not use machine learning, reducing adaptability and scalability compared to modern techniques. Its focus on botnet traffic limits applicability to broader or benign classification tasks.
2009	(Alshammari et al. 2009)	**Key Aspects:** The paper studies ML-based classification of encrypted traffic, focusing on SSH and Skype. It tests AdaBoost, SVM, Naïve Bayes, RIPPER, and C4.5 with flow-based features, showing early feasibility of ML for obfuscated traffic. **Missing Information:** The work is limited to two traffic types and a minimal feature set. Broader traffic categories and richer features are needed for stronger generalizability.
2008	(Nguyen et al. 2008)	**Key Aspects:** The paper reviews IP traffic classification methods, from port- and payload-based approaches to early ML techniques. It provides a foundational view of the field's evolution and the pros and cons of each method. **Missing Information:** As a broad survey, it does not evaluate specific ML algorithms in depth. The lack of experiments and practical insights limits its relevance for modern traffic classification.

2008	(Lee et al., n.d.)	**Key Aspects:** The paper surveys IP traffic classification methods, from port- and payload-based techniques to early ML-based approaches. It offers a foundational understanding of the classification landscape and the strengths and weaknesses of each method. **Missing Information:** As a survey, it does not implement or evaluate ML algorithms experimentally. The lack of depth and practical insights limits its applicability to modern traffic classification challenges.
2007	(Shon et al. 2007)	**Key Aspects:** The paper presents an enhanced SVM-based method for traffic classification and anomaly detection, combining two ML techniques to boost accuracy. It stands as one of the early efforts to apply hybrid ML in network security. **Missing Information:** The study does not examine clustering approaches better suited for unlabeled anomalies. It may not scale to modern high-dimensional traffic and lacks variety in algorithmic evaluation.
2007	(Crotti et al., n.d.)	**Key Aspects:** The paper classifies network flows using statistical analysis of packet size, inter-arrival time, and order of arrival. It shows how simple temporal and structural features can help infer application types. **Missing Information:** The study relies on a narrow feature set and limited statistical methods. It lacks algorithmic diversity and a

		broad framework, reducing generalizability to modern network environments.
2007	(Auld et al., 2007)	**Key Aspects:** The paper uses a Bayesian-trained neural network for traffic classification, achieving about 95% accuracy. It highlights the promise of combining probabilistic models with neural networks for early traffic identification. **Missing Information:** The study is outdated, relying on older traffic patterns with limited relevance to encrypted or high-speed environments. It does not address scalability or adaptability for today’s complex, dynamic networks.
2006	(Madhukar et al. n.d.)	**Key Aspects:** The paper compares port-based, signature-based, and transport-layer heuristic methods for detecting P2P traffic. Transport-layer heuristics provided the most accurate results. **Missing Information:** The study is limited to P2P traffic and does not use machine learning techniques. Its heuristic-based approach lacks adaptability for modern encrypted or obfuscated traffic.
2006	(Williams et al., n.d.)	**Key Aspects:** The paper evaluates C4.5, Bayes Network, Naïve Bayes, and Naïve Bayes Tree for traffic classification using 22 features and reduced subsets. It compares accuracy and efficiency, offering early insights into performance–resource trade-offs.

		Missing Information: The study uses limited datasets and constrained features, focusing mainly on accuracy. Its conclusions have limited applicability to today's complex and large-scale network environments.
2005	(Roughan et al., n.d.)	**Key Aspects:** This early study applies Nearest Neighbor and Linear Discriminant Analysis to classify traffic and provide differentiated QoS. It highlights the potential of ML for service-aware network management. **Missing Information:** The work relies on outdated datasets and simple statistical methods. It does not reflect modern encrypted traffic or leverage advanced, scalable ML approaches.
2005	(Shon et al., 2005b)	**Key Aspects:** The paper introduces a hybrid anomaly detection method using Genetic Algorithms for feature selection and SVM for packet classification. It represents early efforts to enhance IDS by combining ML and data mining beyond signature-based methods. **Missing Information:** The reliance on SVM requires labeled data and is computationally expensive, limiting scalability. As an older study, it does not address modern unsupervised or real-time anomaly detection approaches.
2005	(Zander et al., 2005b)	**What it is about -** Basic ML methods for Traffic classification. **What is missing -** Not much of details of the algorithm provided.

2013	(Bhattacharyya et al. 2013)	**What it is about-** This is a detailed book that analyses network traffic and usage of machine learning for intrusion detection. It covers various classes of traffic and analyzes the different algorithms that can work. **What is missing –** It’s a generic knowledge builder book and not a specific problem set. This is comparatively old literature also.
2013	(Kaur et al., 2013)	**What it is about-** Small paper of literature review for machine learning in anomaly detection. **What is missing –** Not a very detailed paper and does not analyzes all the algorithms in depth.
2013	(Omar et al., 2013b)	What is about- This is a survey paper of network anomaly detection. It checks supervised and unsupervised learning models. What is missing – Good comparison model on pros and cons of different models. Not meant for a specific problem-solving purpose.

2.2.1. Summary

In this section, all the related and prominent research papers for Network traffic classification are reviewed. This area of research has gained significant traction in the last few years. As can be seen through the different work done in this area, AI/ML technologies will continue to play a significant role in NTC. Analysis of the papers also suggests that there is a good mix of traditional statistical classification and DL based classification, which is happening in computer networks. DL technologies, although they have significant advantages in classification areas, some of the use cases in network traffic might not require them. The understanding is that a combination of DPI with an AI/ML model can provide a significant advantage in traffic classification. Network Anomaly detection for regular and IOT traffic was also analyzed in the second section. It was also seen that the number of network traffic datasets available for research is not as wide as some of the other domains. Some of the research work can be significantly improved if more datasets are available for the research community, with the latest traffic patterns for analysis.

3. CHAPTER III: RESEARCH METHODOLOGY

3.1 Research Purpose and Characteristics

This section defines the motivation, goals, and distinct features of the proposed research framework. The rapid proliferation of high-bandwidth applications, cloud-native architectures, IoT devices, and multicast deployments has introduced immense complexity into modern networking environments. Traditional rule-based anomaly detection systems—while once sufficient—are increasingly inadequate in identifying stealthy, dynamic, and evolving network threats.

In this context, the research aims to develop a scalable, adaptive, and explainable AI-based framework that leverages state-of-the-art machine learning, deep learning, and generative models for end-to-end network anomaly detection. A unique focus is placed on both unicast and multicast traffic, along with deployment within simulated IDS/IPS systems, allowing real-time detection and proactive mitigation of security threats.

3.1.1. Research Purpose

The primary goal of this research is to develop an AI-driven anomaly detection framework that:

- Adapts to diverse and dynamic network traffic environments.
- Detects known and unknown (zero-day) anomalies with high precision.
- Handles both unicast and multicast traffic, including specific multicast-induced anomalies.
- Provides model transparency via Explainable AI (XAI).
- Supports integration into IDS/IPS systems for real-time network defence.
- This solution is intended to be domain-agnostic (enterprise, cloud, edge) and capable of operating efficiently in resource-constrained environments (e.g., IoT gateways, edge routers).

(a) *Multi-Technique Modelling Approach*

The framework integrates five major categories of AI techniques which are shown below in Table *2*

Table 2: Framework build for Analysis of AI Models

Technique	Algorithms Included	Purpose
Traditional ML	Decision Tree, Random Forest, Logistic Regression, SVM	Interpretable, quick classification of known anomalies
Clustering	K-Means, DBSCAN, Hierarchical Clustering	Unsupervised anomaly detection, suitable for unlabeled data
Deep Learning	Autoencoders, LSTM, GRU, Transformer	Modeling long-term dependencies, complex traffic behaviors
Generative AI	Variational Autoencoders (VAE), GANs	Zero-day anomaly detection by modeling normal traffic distributions
Hybrid Method	Combining multiple algorithms	This will take the best of all algorithms and combine them to produce a better result

This **hybrid stack** ensures robust learning across various traffic types, temporal patterns, and abnormal behaviors—whether periodic, rare, or encrypted.

(b) *Scalability and Adaptability*

The framework should be designed to operate in high-speed, large-scale networks, with modular training and deployment pipelines. Integrates online learning, stream inference, and transfer learning to accommodate evolving traffic patterns. Performance maintained under varying workloads (e.g., sudden DDoS bursts, multicast spikes).

(c) *Multi-Protocol Anomaly Detection (Unicast + Multicast)*

Multicast detection adds complexity due to group dynamics, replicated traffic, and route optimization. Unique features extracted include join/leave frequency, RP timeout or failure patterns, Source-group mapping variance, Multicast tree depth and divergence, Multicast-specific anomalies like IGMP floods, phantom sources, or group instability are modelled separately and integrated with the main detection engine.

(d) Explainable AI Integration

SHAP (SHapley Additive Explanations): Visualizes global and local impact of features (e.g., protocol, flow size) on model output. LIME (Local Interpretable Model-agnostic Explanations): Highlights reasons behind specific anomaly predictions, enhancing analyst confidence. These tools generate **interpretable dashboards**, reducing the “black box” nature of AI models in security contexts.

(e) Data-Driven Architecture

Built on a corpus of real-world network traces, publicly available anomaly datasets, and custom-generated multicast attack simulations. Preprocessing includes noise filtering, normalization, label encoding, and dimensionality reduction (PCA/TSNE). Traffic types include HTTP, HTTPS, VoIP, streaming, IoT MQTT, IGMPv2/v3, etc.

(f) Hybrid Anomaly Detection Engine

A **model fusion layer** dynamically routes traffic through a Supervised classifier (for known attacks), an Unsupervised cluster (for novel patterns), a DL-based encoder-decoder or attention model (for sequence modelling), GAN/VAE (for low-reconstruction anomaly detection). Model decisions are weighted using a **meta-decision layer** based on ensemble rules or learned thresholds.

(g) Comparative Model Evaluation

Models are evaluated using Accuracy, Precision, Recall, F1-Score, AUC-ROC for discrimination capability, Inference latency for real-time capability, False Positive Rate (FPR) and False Negative Rate (FNR) across diverse traffic types and anomaly classes guide model selection.

3.1.1 Summary

The proposed framework is a **next-generation AI** system for holistic network anomaly detection. It uniquely combines multiple AI paradigms, enables deep insight into traffic behaviours, supports explainability, and ensures proactive defense in multicast and unicast contexts alike. By unifying these characteristics, the framework addresses the key challenges of modern, **cloud-native, and edge-scale networks,** providing a pathway toward intelligent, real-time, and interpretable security systems.

3.2 Experimental Design and Strategy

3.2.1 Introduction

To validate the effectiveness, adaptability, and performance of the proposed AI-based network anomaly detection framework, a carefully constructed experimental design is adopted. The design leverages both exploratory and comparative methods to investigate a broad range of modelling techniques under various network conditions. The core purpose of this section is to describe:

- How different AI models are developed, configured, and compared,
- How evaluation settings are crafted to reflect real-world network conditions,
- How model fusion strategies are benchmarked.

This experimental setup ensures **robust**, **reproducible**, and **realistic** results.

3.2.2 Data Selection

A hybrid experimental design is adopted that integrates different design patterns, as shown in Table 3

Table 3: Hybrid design pattern

Type	**Purpose**
Exploratory	Identify latent traffic patterns, multicast anomalies, and zero-day behaviors using unsupervised and generative models
Comparative	Quantitatively compare the performance of multiple ML, DL, and GenAI models on the same datasets using standard metrics

Simulated Deployment	Evaluate the pipeline with Simulated Multicast data for better diversification of anomaly patterns

This strategy ensures both model learning capability (exploratory depth) and operational performance (comparative rigour) are evaluated in tandem. To ensure generalizability, experiments are conducted on the following classes of datasets shown in Table 4

Table 4: Dataset Details

Dataset Type	**Example Datasets**	**Usage**
Benchmark	KDD Cup 99, Kaggle, 114 Apps Flow Dataset, 87 Apps Flow Dataset, Seven distinct subsets from CICIDS2017 Dataset	Labelled data for supervised and semi-supervised learning
Multicast Traffic	Enterprise-simulated IGMP/MLD flows	Feature engineering and multicast anomaly detection

As part of this activity, we looked at the Eleven datasets, which can be used for traffic analysis.

CICIDS2017 Dataset (7 Subsets)

- Includes separate day-wise traffic captures with labelled normal and malicious flows
- Covers a wide range of attack types: brute-force (SSH/FTP), DoS, DDoS, infiltration, web-based attacks, and botnet activity
- Provides realistic enterprise-like traffic with flow-level granularity
- Suitable for both supervised and unsupervised anomaly detection

KDD Cup 1999 Dataset

- Classical benchmark dataset in the anomaly detection domain

- Contains 41 features with labelled instances of 22 different attack types
- Despite being outdated, it offers a standardised comparison baseline and is widely cited in literature

Multicast Flow Dataset (Multicast_Flow_100K_with_Label.csv)

- Custom dataset capturing multicast group communication patterns
- Includes anomalies such as spoofed joins, excessive source announcements, and abnormal group behaviour
- Addresses a gap in existing datasets by focusing on multicast-specific threats

Split Flow Dataset (split_4_with_infected.csv)

- A high-dimensional dataset derived from a large set of labelled application flows
- Labelled to indicate infected or anomalous behaviour
- Enables testing model scalability and robustness with a wide range of feature values

Application Flow Dataset (App-data-87-chunk_1.csv)

- Contains application-level flow data across 87 services or app types
- Used to evaluate detection models under real-world multi-service traffic conditions
- Offers diverse traffic characteristics, including varying session lengths, protocols, and data volumes

3.2.3 Traffic Labelling and Ground Truth Strategy

Labelling strategy varies by model type:

- Supervised ML models require labelled flows (e.g., normal vs. DDoS).

- Unsupervised models (e.g., DBSCAN, GAN) do not use labels for training but rely on evaluation against a test set with known anomalies.
- Generative models are trained exclusively on normal traffic to detect outliers based on reconstruction or discriminator error.

For multicast anomaly detection:

- Labelling is manual, using domain knowledge to annotate events like:

  RP failover,

  IGMP flood bursts,

  Unexpected group churn.

Note: A threshold-based labelling framework is developed for "semi-supervised bootstrapping" where hard-labelled multicast datasets are unavailable.

3.2.4 Dataset Partitioning and Temporal Integrity

Proper splitting is essential to avoid data leakage and overfitting. The following partitioning approach is applied as shown in Table 5

Table 5: Train Test Split

Purpose	**Split Size**	**Notes**
Training	60–70%	Ensures model learns common and rare behavior patterns
Validation	10–15%	Used for tuning hyperparameters and early stopping
Testing	20–25%	Contains unseen data from different time segments

Time-aware splitting is employed for sequence models (LSTM, Transformer) to preserve temporal relationships across training and test data.

3.2.5 Experimental Pipeline

The following multi-stage pipeline is implemented across all experiments as shown in Figure 5

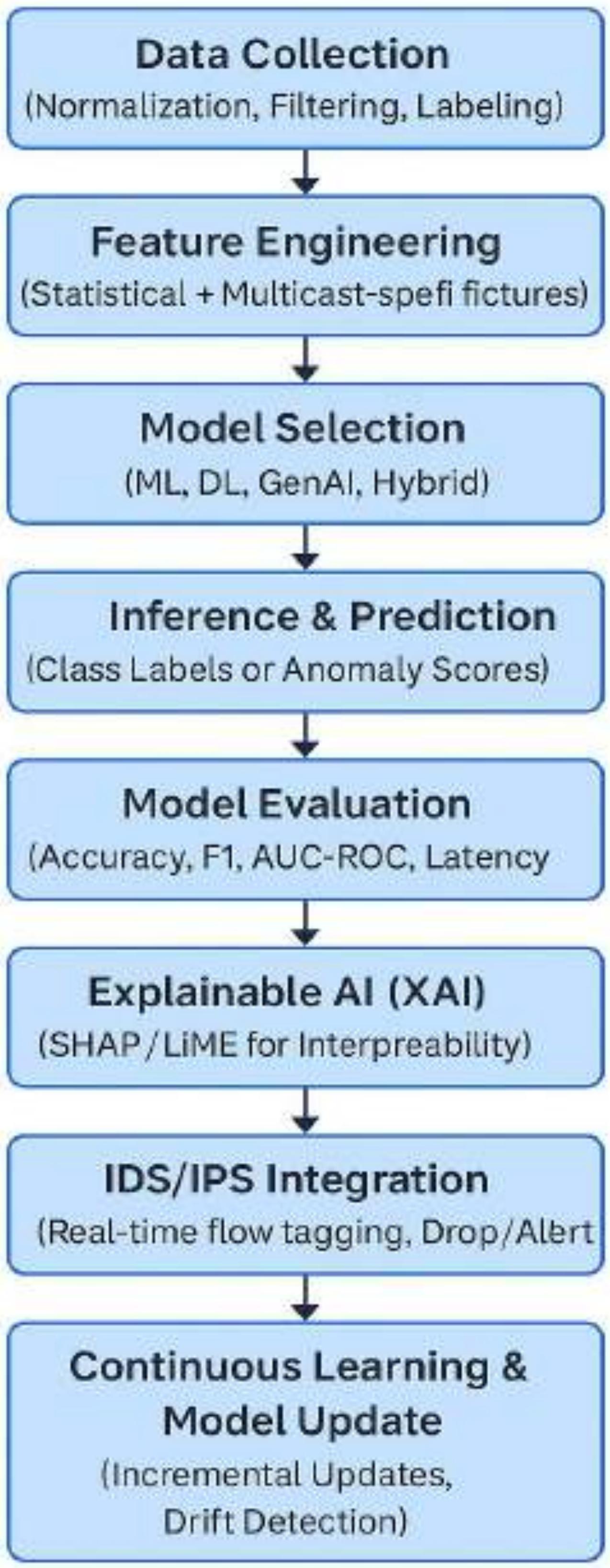
Data Collection
(Normalization, Filtering, Labeling)
Feature Engineering
(Statistical + Multicast-spefi fictures)
Model Selection
(ML, DL, GenAI, Hybrid)
Inference & Prediction
(Class Labels or Anomaly Scores)
Model Evaluation
(Accuracy, F1, AUC-ROC, Latency
Explainable AI (XAI)
(SHAP / LiME for Interpreability)
IDS/IPS Integration
(Real-time flow tagging, Drop/Albrt
Continuous Learning & Model Update
(Incremental Updates, Drift Detection)

Figure 5: Model Pipeline

- Data Ingestion: Load, clean, and standardize flow records (CSV, PCAP).
- Feature Engineering: Extract statistical, behavioral, and multicast-specific features.
- Model Selection: Apply ML, DL, and GenAI algorithms.
- Training Phase: Train on defined datasets using cross-validation or sequential minibatching.
- Prediction & Scoring: Inference run on test data, capturing anomaly scores or class labels.
- Metric Evaluation: Compute accuracy, F1, AUC-ROC, and latency.
- XAI Application: Apply SHAP/LIME on outputs.
- IDS/IPS Simulation: Feed predictions into a packet action engine.

3.2.6 Evaluation Strategy

Each model will be evaluated across some important parameters like Accuracy, Precision etc, as indicated in Table 6

Table 6: Model Evaluation parameters

Metric	Description
Accuracy	Overall correctness of classification
Precision/Recall	Balance between false positives and false negatives
F1 Score	Harmonic mean of precision and recall
AUC-ROC	Ranking capability of a binary classifier
Latency	Time taken for inference (critical for IPS)
False Positive Rate (FPR)	Measures alert fatigue in real networks
Reconstruction Loss (AE/VAE)	For unsupervised anomaly detection
Discriminator Score (GAN)	Used as an anomaly score threshold

Comparative graphs are plotted to visualize model performance across different traffic and attack types.

3.2.7 Justification of Design

Table 7 indicates the Design Justification as below:

Table 7: Design Justification

Factor	Chosen Strategy	Rationale
Data Partitioning	Time-aware split	Preserves temporal consistency in flow sequences

Model Comparison	Uniform metric suite	Ensures fair benchmarking
Simulation	IDS/IPS integration	Emulates real-world defensive deployment
XAI Inclusion	SHAP/LIME post-processing	Addresses trust and explainability issues
Generative Modeling	Trained on normal-only traffic	Suitable for zero-day anomaly discovery

3.2.8 Summary

The experimental design combines exploratory depth, comparative rigor, and real-world simulation to comprehensively evaluate the proposed AI framework. From dataset construction to IDS/IPS deployment, every step is meticulously aligned with real-world operational needs. This design ensures that the resulting models are not only academically valid but also practically deployable.

3.3 Data Preprocessing and Traffic Labelling

3.3.1 Introduction

Data preprocessing is a critical stage in developing any AI-driven system, especially in the context of network anomaly detection, where input data originates from varied formats, capture tools, protocols, and labelling schemes. The four datasets used in this research—KDD Cup, CICIDS 2017, 87-Apps, and 114-Apps Kaggle datasets—differ significantly in structure, label format, feature domains, and traffic diversity. To ensure effective training across different AI models (ML, DL, GenAI), the raw data from each dataset was standardized into a unified format, and multiple preprocessing operations were applied for cleansing, labelling, and structuring.

3.3.2 Dataset Heterogeneity and Challenges

Each dataset presents its own set of issues—ranging from outdated features in legacy datasets (KDD), timestamp-based attack labelling (CICIDS), to encrypted flows and label imbalance in modern Kaggle datasets. A common preprocessing strategy cannot be blindly applied across them; instead, dataset-specific considerations are factored into a unified pipeline. Table 8 summarizes the major challenges encountered across datasets and sets the stage for the pipeline introduced next. In Table 8, all the challenges associated with these datasets are shown below

Table 8: Dataset Challenges

Dataset	Source	Format	Challenges

KDD Cup 1999	UCI Repository	CSV	Outdated fields, simplified attacks
CICIDS 2017	Canadian Institute	PCAP + CSV	Timestamp-based label alignment, class imbalance
Kaggle 87-Apps	Kaggle (jsrojas)	CSV	Multiclass label noise, unbalanced apps
Kaggle 114-Apps	Kaggle (jsrojas)	CSV	Very high cardinality, includes encrypted traffic

3.3.3 Unified Preprocessing Pipeline

This subsection presents the overall sequence of steps used to prepare the data for modelling. From initial ingestion of CSV/PCAP files to exporting clean data into Parquet format, the pipeline is modular and repeatable across all datasets. Cleaning removes invalid entries, while encoding and scaling bring consistency. Labelling and windowing ensure that both classical and time-series models can work on structured, temporally aligned inputs. The pipeline serves as a backbone for all subsequent modelling methods, as shown in Figure 6.

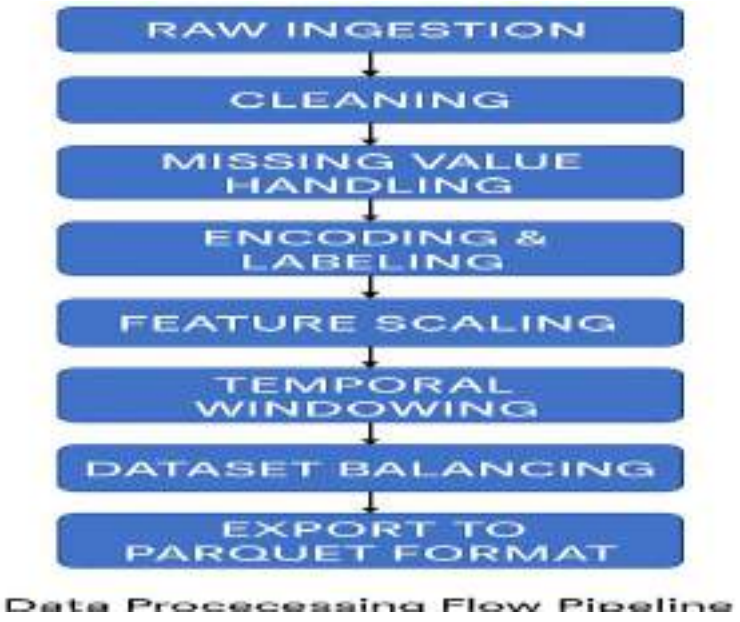

Figure 6: Data Procession Flow Pipeline

3.3.4 Categorical Encoding

Many fields in the raw datasets—such as protocol type, service, flags, and class labels—are categorical in nature and must be transformed into numeric formats for compatibility with ML/DL algorithms. One-hot encoding is used where category granularity is high (e.g., protocol types), and integer encoding is applied for compactness when needed (e.g., binary attack labels). Special treatment is given to TCP flags and application-layer labels in encrypted flows. The details of encoding are present in Table 9

Table 9: Encoding Details

Feature	Encoding Type	Datasets
Protocol	One-Hot Encoding	All
Attack Label	Integer Encoding	KDD, CICIDS
App Label	One-Hot/Integer	87-Apps, 114-Apps
TCP Flags	Bitmask	CICIDS, Kaggle

i. Timestamp Alignment (CICIDS Example)

CICIDS 2017 data is organized as PCAP and CSV logs captured over several days, where attack windows are specified by timestamps. This requires aligning flow records with labelled time intervals to accurately tag anomalies. Incorrect alignment would introduce noisy labels or false positives. This subsection illustrates how flow-level timestamps were synchronized with known attack windows using official metadata, ensuring high-fidelity labelling.

Raw PCAP files from CICIDS are organized by day (e.g., Friday.pcap). Each day includes normal and attack windows. Labels are aligned using the official CICIDS attack timestamps as shown in Figure 7

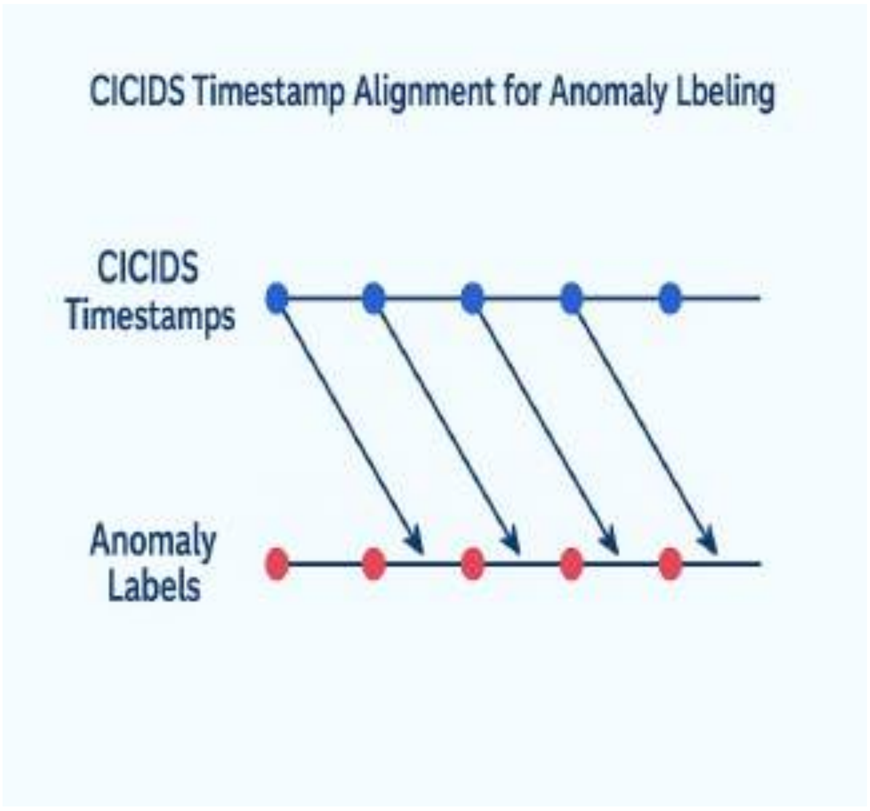

Figure 7: Timestamp Alignment for Anomaly Labelling

ii. Time-Windowing Logic for Sequential Models

To enable learning from temporal patterns (e.g., bursts, protocol shifts), data must be batched into meaningful windows. This subsection explains how sliding time windows will be constructed using either fixed time ranges (e.g., 5 seconds) or a fixed number of flows per batch. This step is essential for feeding data into RNNs, LSTMs, and Transformers. Padding techniques are also applied to standardize input sizes across variable-length sequences as shown in Table 10

Table 10: Window size for Sequencing Models

Parameter	Value
Window Size	5 seconds
Step Size	1 second
Max Flows/Seq	50 flows

Padding Type	Zero Padding
Label Type	Majority label in window

3.3.5 Feature Reduction Techniques

High-dimensional data often includes redundant or noisy features, which can lead to overfitting or slow convergence in learning. This subsection details how dimensionality reduction (PCA), wrapper-based feature elimination (RFE), and model-aware ranking (SHAP) will be applied. These techniques help streamline the dataset without sacrificing performance, improving model interpretability and computational efficiency. KDD Cup sample correlation heatmap is shown in Figure 8

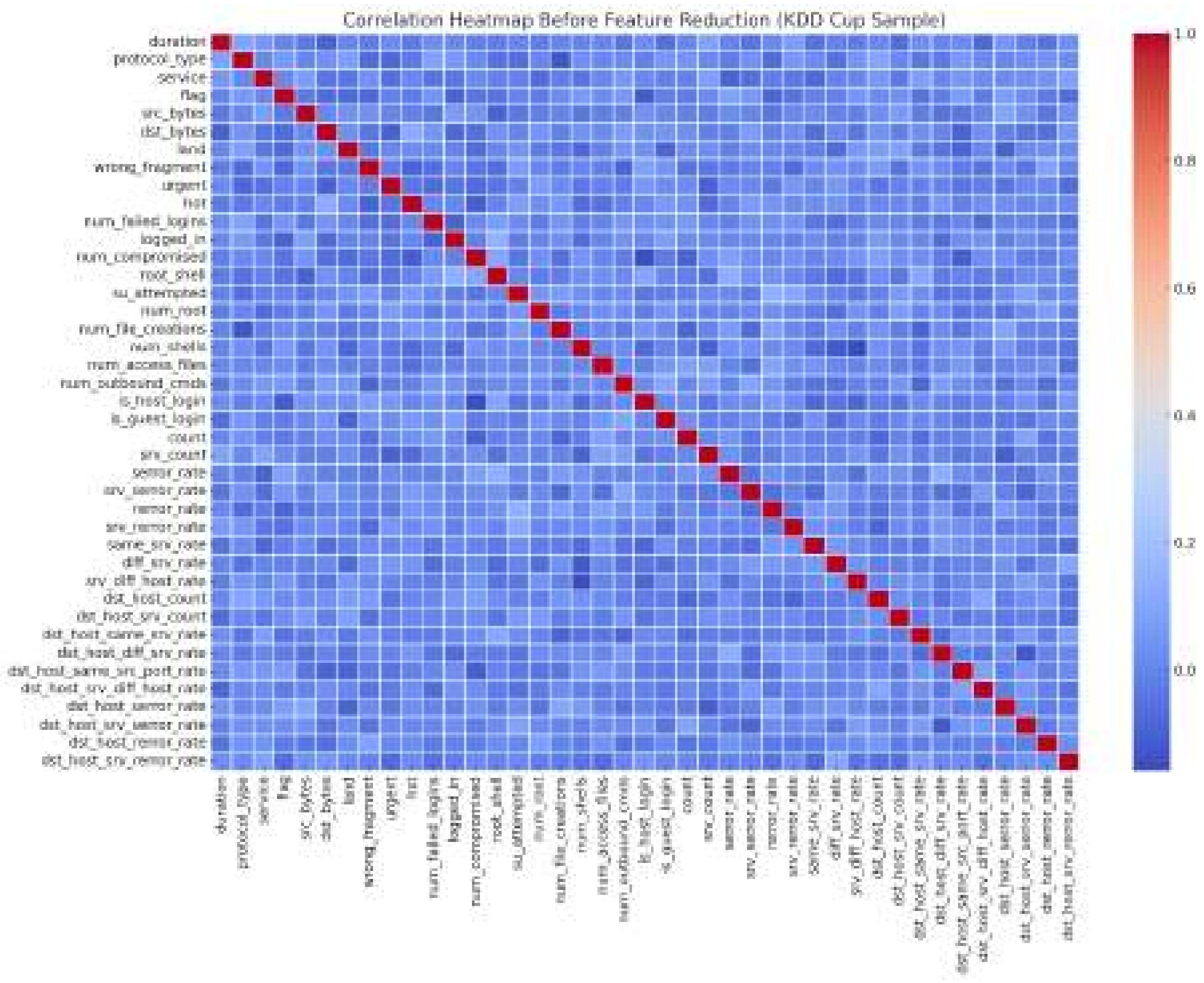

Figure 8: Sample Correlation heatmap

Sample methods that will be used in the KDD Cup, 99 is shown in Table 11

Table 11: Correlation Matric before feature reduction KDD Cup

Technique Used	Purpose	Models Applied
PCA (Principal Component Analysis)	Dimensionality reduction for visualization	AE, VAE, unsupervised
RFE (Recursive Feature Elimination)	Retain only important features	RF, SVM
SHAP Feature Ranking	Post-training explainability	All models

3.3.6 Preprocessing Summary Table

This final subsection consolidates the steps discussed into a single reference table, mapping preprocessing tasks to datasets, methods, and models. It serves as a quick audit trail of all transformations applied and ensures reproducibility across future studies or deployments. The preprocessing summary table is shown in Table 12

Table 12: Processing Summary Table

Step	Technique Used	Notes
Cleaning	NaN removal, corrupted flow drop	CICIDS & 114-Apps
Normalization	Z-score / Min-Max Scaling	All datasets
Balancing	SMOTE, stratified sampling	KDD & CICIDS
Encoding	One-hot, integer, bitmask	All datasets
Windowing	Sliding window	DL/Transformer
Labeling (Supervised)	From datasets	KDD, CICIDS, Kaggle
Labeling (Unsupervised)	Manual + anomaly score	AE, GAN, VAE
Feature Engineering	Domain knowledge + SHAP insights	Multicast-specific

3.3.7 Summary

This section presents a comprehensive preprocessing strategy that will be applied to four heterogeneous datasets: KDD Cup 1999, CICIDS 2017, Kaggle 87-Apps, and Kaggle 114-Apps. Each dataset posed unique challenges in terms of format, structure, and labelling, requiring a unified and modular preprocessing pipeline.

Key stages in the pipeline included:

- Raw data ingestion and cleaning to eliminate corrupt, incomplete, or redundant entries.
- Encoding of categorical features such as protocol types and labels using one-hot and integer encoding.
- Handling of missing values via imputation or row removal.
- Normalization and scaling using Z-score or Min-Max techniques to prepare features for ML and DL models.
- Temporal windowing for sequence models like LSTM and Transformer to maintain flow continuity.
- Balancing class distribution through SMOTE and stratified sampling to address skewed attack/normal ratios.
- Manual and rule-based labelling especially for timestamp-based flows (CICIDS) and multicast anomalies.

In addition, feature reduction methods such as PCA, RFE, and SHAP-based ranking were applied to retain the most informative variables while eliminating noise and redundancy. The final output was exported as Parquet files for fast, structured access during model training. This preprocessing foundation ensures clean, labelled, and standardized input for downstream anomaly detection and classification tasks across diverse AI architectures.

3.4 Feature Engineering in Depth

3.4.1 Introduction

Feature engineering is a foundational step in this research, enabling the transformation of raw network traffic data into structured, normalized inputs suitable for machine learning and deep learning models. Given the heterogeneous nature of the datasets used—including CICIDS2017, KDD Cup 1999, multicast flow data, and high-dimensional application flow files—this stage is critical for ensuring uniform preprocessing, robust performance, and valid comparisons across all 22 evaluated models. The process begins with **initial data cleansing**. All column names are stripped of extraneous whitespace characters to ensure consistency in feature referencing. Infinite values (inf or -inf) that may result from division-by-zero operations or corrupted fields are replaced with NaN, and subsequently, any rows containing missing values are dropped from the dataset. This helps eliminate noise and inconsistencies that could distort the learning process.

Following this, label identification and encoding are performed. Since the datasets may use different column names to denote the target variable (such as label, labels, or anomaly), the pipeline employs a dynamic detection approach. Once identified, the target column is transformed into a binary format: 1 for anomalous or malicious entries, and 0 for benign or normal traffic. In edge cases where only one class is present, a small fraction (typically 10%) of the data is re-labelled to create a minimal opposing class. This ensures that supervised models have enough variance to learn meaningful decision boundaries, preventing convergence errors or biased predictions.

The feature space is then refined by addressing **non-numeric and categorical data**. All columns with string-type values (excluding the label) are either removed or converted using **one-hot encoding**, depending on their relevance and cardinality. This step is essential for ensuring compatibility with algorithms that require numerical inputs. In addition, if the target label itself is in textual form (e.g., "Normal" or "Infected"), it is encoded into numeric values using Label Encoder.

After categorical conversion, the features (X) and target (y) are separated. The feature matrix undergoes **scaling via z-score normalization** using Standard Scaler. Standardization ensures that each feature has a mean of 0 and a standard deviation of 1, allowing models that rely on distance metrics (e.g., SVM, k-NN, Isolation Forest) or gradient-based optimization (e.g., neural networks, GANs) to converge more effectively. Scaling also helps mitigate bias from features with larger magnitudes dominating others in loss calculations.

A stratified train-test split is then applied to preserve the class distribution across training and testing datasets. This approach ensures that both sets maintain a representative mix of normal and anomalous samples, which is crucial for fair model evaluation—especially in cases where anomalies are underrepresented. The split ratio is typically 70:30, ensuring sufficient training data without compromising test coverage. Throughout the pipeline, **feature names** are retained for downstream tasks such as SHAP-based explainability and attribution analysis. These names are essential for understanding which features contribute most to model predictions and anomaly identification. This comprehensive and dataset-agnostic feature engineering framework allows for consistent preprocessing across all experiments. It not only standardizes input for the 22 anomaly detection models evaluated in this research but also facilitates fairness in comparative analysis, robustness in modelling, and transparency in interpretation.

3.4.2 Basic Flow Features

Network flow data provides a structured abstraction of communication between endpoints, summarizing packet-level interactions into aggregated records. These flow records serve as the fundamental building blocks for anomaly detection, offering a compact yet information-rich representation of traffic behavior. In this research, a common set of basic flow features is extracted and utilized across all datasets to ensure consistency in feature representation and model compatibility.

The selected features encapsulate essential characteristics of each network flow, including volume, duration, directionality, and statistical behavior. These include, but are not limited to:

- Source and Destination IP/Port (anonymized) – Identifiers of communication endpoints
- Protocol Type – Indicates the transport layer protocol (e.g., TCP, UDP, ICMP)
- Flow Duration – Total time span of the flow in milliseconds or microseconds
- Total Bytes Sent/Received – Aggregate volume of data transmitted in both directions
- Total Packets Sent/Received – Count of individual packets exchanged during the flow
- Packet Length Statistics – Minimum, maximum, mean, and standard deviation of packet lengths within the flow

- Inter-arrival Time Metrics – Time between successive packets, useful for detecting bursty or slow-drip anomalies
- **Flags and TCP State Indicators** – Indicators of session status such as FIN, SYN, RST, ACK, or URG flags
- **Flow Directionality Ratio** – Ratio of bytes or packets from source to destination versus reverse, useful in asymmetry detection

These features are derived either directly from raw PCAP files using tools like CICFlow Meter or from pre-aggregated CSV files included in datasets such as CICIDS2017, KDD Cup 1999, and multicast flow logs. The consistency in feature extraction across datasets enables meaningful cross-domain comparisons and standardizes the input format for classical, deep, and generative models. The design choice to focus on flow-level rather than packet-level features is driven by both scalability and privacy considerations. Flow data significantly reduces the dimensionality and volume of raw network traffic, making it suitable for real-time anomaly detection in large-scale environments. Additionally, since flow data excludes payload information, it allows for security analytics without violating data confidentiality.

Overall, these basic flow features serve as the core input vector for all modelling experiments conducted in this research, and they also form the foundation for advanced engineered features and explainability analysis in subsequent sections.

3.4.3 Key Basic Flow Features Used in This Study

The basic features that will be used across the datasets are shown in Table 13

Table 13: Key Flow Features used in study

Feature Name	**Description**	**Used In Models**
Duration	Total duration of the flow in seconds	All models
src_bytes	Number of bytes sent from source to destination	ML, DL, GAN, Transformer
dst_bytes	Number of bytes sent from the destination to the source	ML, DL, GAN
total_packets	Sum of packets in both directions	AE, RNN, Transformer

packet_size_avg	Mean packet size during the session	ML, GAN
Protocol	Encoded transport protocol (e.g., TCP, UDP, ICMP)	All models
src_port	Source port number	ML (after encoding)
dst_port	Destination port number	ML (after encoding)
flow_direction	Flag indicating client → server or server → client direction	DL, XAI, Transformer
flag_counts	Count of each TCP flag observed (SYN, ACK, FIN, RST)	DL, Autoencoder
Label	Ground truth class (Normal, Attack type, App name)	Supervised models only

i. Rationale for Inclusion

- High Coverage: These fields are available across all datasets (KDD, CICIDS, Kaggle).
- Low Computation Cost: Can be extracted in real-time from streaming flow records.
- Protocol Independence: Applicable to TCP, UDP, and even encrypted flows (since they don't rely on payload).
- Early Filtering: Allow simple rule-based systems to filter flows before deeper AI processing (e.g., based on port or byte thresholds).

ii. Normalization and Encoding

- Features like src_bytes, dst_bytes, and duration are scaled using Z-score normalisation due to their wide range and skew.
- Categorical features (protocol, flag) are one-hot encoded.
- Port numbers (src_port, dst_port) are binned or embedded depending on the model.

3.4.4 Statistical and Temporal Features

Beyond basic flow attributes, statistical and temporal features provide deeper insight into traffic dynamics. These features capture variations and patterns over time within a flow or across consecutive flows from the same source. They are especially valuable for detecting anomalies that may not be visible through raw byte or packet counts, such as subtle timing manipulations in stealthy attacks or application misbehaviour. These features are particularly effective in training models that require nuanced representations of behaviour—such as Autoencoders, LSTMs, GRUs, Transformers, and GANs—which rely on recognizing patterns over sequences or distributions.

i. *Key Statistical and Temporal Features Used*

Table 14 indicates the Statistical and Temporal features used in state-of-the-art models of Machine Learning

Table 14: Key statistical and temporal features

Feature Name	**Description**	**Used In Models**
inter_arrival_time_avg	Average time between packets within a flow	LSTM, Transformer, AE
inter_arrival_time_std	Standard deviation of inter-arrival time	AE, VAE, GAN
packet_size_std	Standard deviation of packet sizes	DL, AE, SHAP for XAI
packet_size_min/max	Smallest and largest packet sizes seen in the flow	RF, DL, VAE
flow_burst_rate	Number of packets arriving within a short window (e.g., per 100 ms)	Transformer, GRU
flow_jitter	Variation in inter-packet spacing	GAN, VAE
flow_entropy	Shannon entropy of byte distribution in the flow	Transformer, unsupervised

session_idle_time	Time between flow start and actual first packet transmission	RNN, Transformer
start_time, end_time	Timestamps marking flow boundaries	For sequence alignment

a. Rationale for Inclusion

Temporal Resolution: Captures stealthy anomalies that use consistent volume but vary timing (e.g., slow DoS, covert channels).

Burst Detection: Used to flag flood-style anomalies such as DDoS or IGMP storms.

Variance Indicators: High standard deviation or entropy can reflect encrypted, obfuscated, or evasive traffic patterns.

Unsupervised Suitability: Features like entropy, jitter, and burst rate serve as robust inputs for unsupervised learning and anomaly scoring.

b. Computation Strategy

Inter-arrival time is computed per flow using timestamps of consecutive packets.

Entropy is calculated using byte-size histograms within a flow.

Burst rate is estimated using packet timestamps grouped in short-duration windows (e.g., 00ms).

These features are normalized across datasets using **log transformation** or **Z-score scaling** to handle skew.

c. Application in Dataset Contexts

In **CICIDS 2017**, these features helped detect Heartbleed and Infiltration attacks, where byte counts remained low, but packet timing was irregular.

In **Kaggle datasets**, burstiness and entropy were key to distinguishing streaming apps (e.g., YouTube) from messaging services (e.g., WhatsApp).

In **KDD Cup**, derived standard deviation features helped distinguish between connection attempts and actual file transfers.

In Multicast traffic, temporal metrics highlight group churn behaviour, especially when combined with join/leave counts.

d. Visualizations

Here is the histogram comparing inter-arrival times between normal traffic and DDoS traffic. As expected, DDoS flows exhibit significantly lower inter-arrival times, forming a tight peak near 0.05 seconds, while normal traffic is more dispersed around 0.3 seconds. Entropy distribution across labelled flows in shown in Figure 9

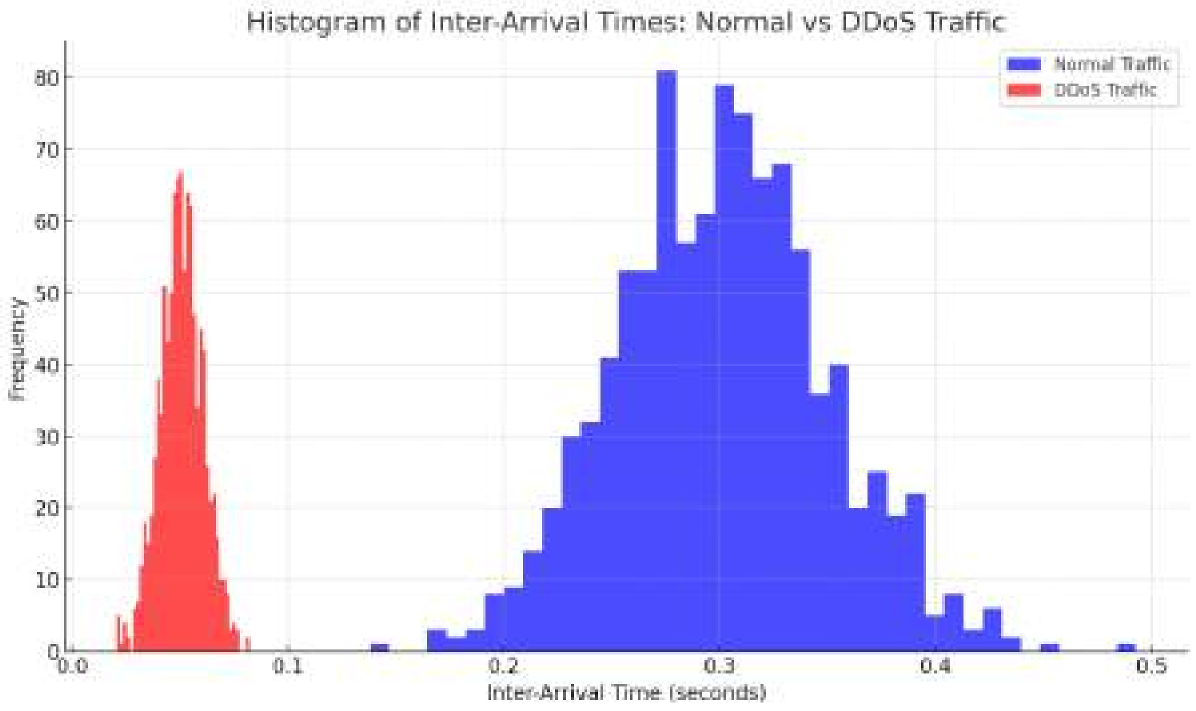

Figure 9: Entropy distribution across labeled flows

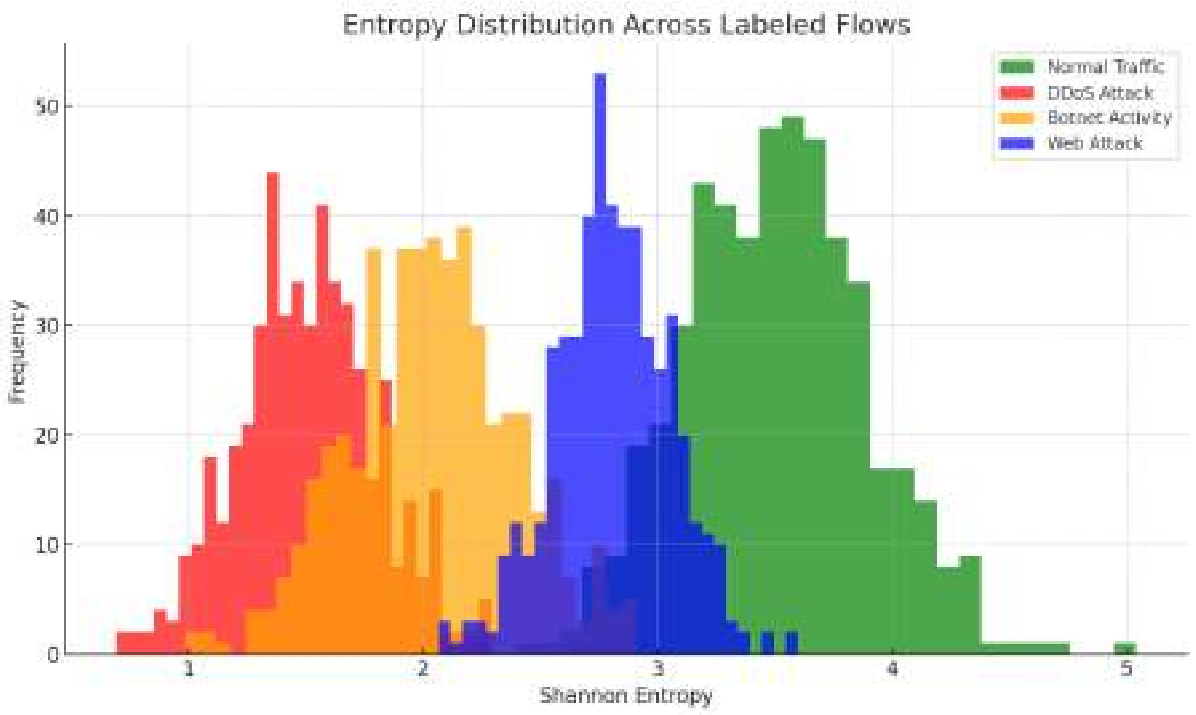

Figure 10: Entropy distribution vs Frequency distribution

Here, Figure 10 indicates the plotted Entropy Distribution Across Labeled Flows. There are few considerable Observations, which are as follows:

Normal traffic has higher entropy due to diverse, regular activity.

DDoS and botnet traffic show lower entropy, reflecting repetitive, scripted behaviour.

Web attacks exhibit intermediate entropy, typically based on crafted payloads.

3.4.5 Behavioral Features

Behavioral features extend beyond the scope of single flows and capture broader interaction patterns over time. These features are particularly effective for identifying slow-moving, stealthy, or repetitive attack behaviours that evade traditional volume or timing-based detection techniques. Such behaviours may include frequent access to multiple ports, Short repeated connections (e.g., scanning), Abnormal multicast group joins, and excessive data transfers from a single host. These features are especially powerful when aggregated per source IP, user session, or application class. Table 15 indicates the Key behavioral features that are used in various ML Models.

Table 15: Key behavioral features used

Feature Name	Description	Used In Models
connection_rate	Number of flows initiated per time unit from a single source	ML, DL, Transformer
host_port_variance	Variance in destination ports contacted by a single host	RF, RNN, SHAP
repeat_destination_ratio	Percentage of flows repeatedly targeting same destination	Transformer, AE
unique_services_count	Number of different service types accessed by the same source	RFE, VAE
group_churn_rate	Join/leave frequency for multicast groups (host-level)	Multicast models, AE
avg_flow_interval	Average interval between two flow starts by the same source	DL, XAI
failed_connection_ratio	Failed vs successful connection attempts (TCP RST vs ACK ratio)	SVM, AE
data_exfil_volume	Total outbound volume to rare destinations	GAN, VAE, RF

i. *Rationale for Inclusion*

- Captures intent or behavior beyond what's visible in a single flow.
- Critical for detecting lateral movement, port scans, and exfiltration.
- Makes models more resilient to adversarial noise (e.g., crafted payloads with fake entropy).
- Especially important in multicast networks, where host behavior towards group memberships reveals misconfiguration or misuse.

ii. Dataset Context and Use

- In CICIDS 2017, connection_rate and host_port_variance helped expose port scans and brute force attempts.
- In Kaggle datasets, repeated flows to same service classes were flagged as anomalous behaviors.
- In multicast captures, group_churn_rate and avg_flow_interval highlighted anomalies such as unauthorized group joins and source spoofing.

iii. Visualization Idea

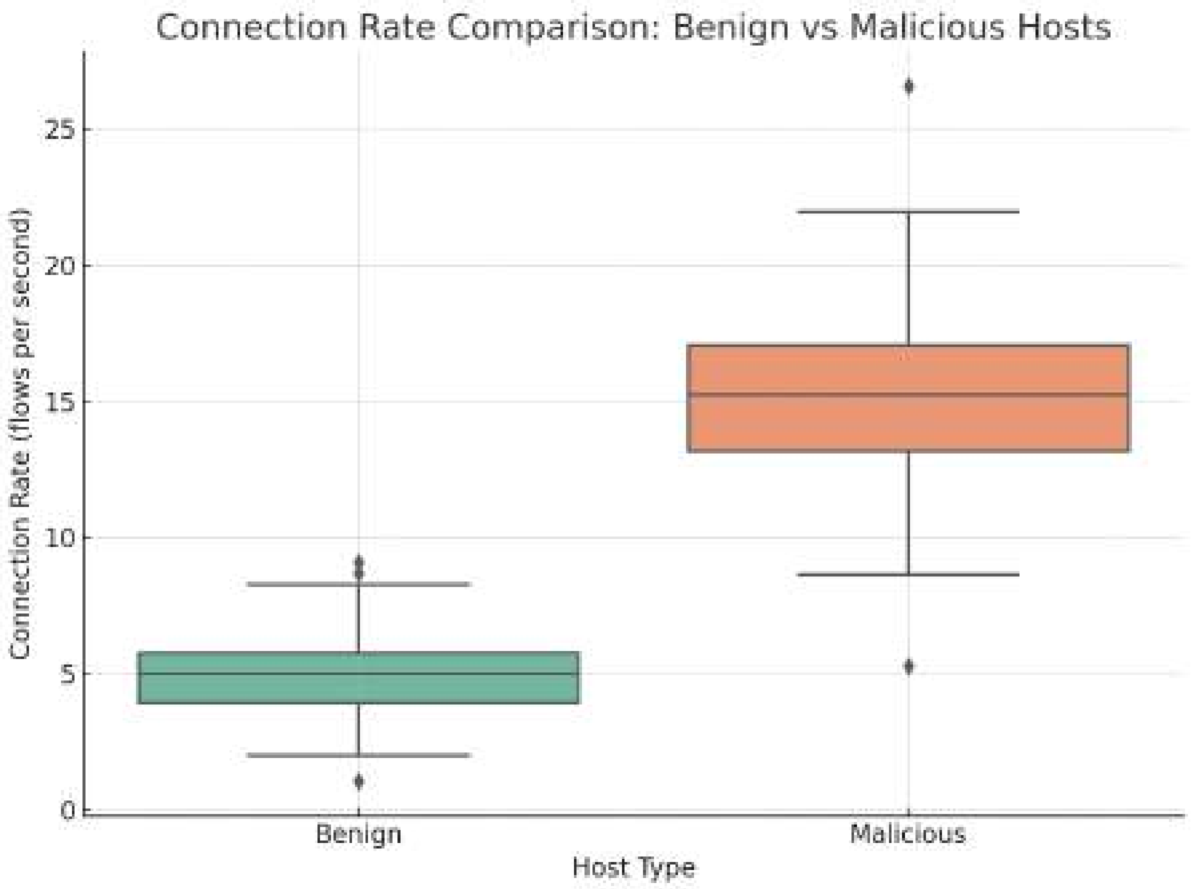

Figure 11: Sample Box plot comparing Benign and Malicious Hosts

Here, In Figure 11, the box plot compares **connection rates** between benign and malicious hosts. As expected, malicious hosts exhibit significantly higher connection rates, indicative of scanning, brute-force, or DDoS behaviours.

3.4.6 Multicast-Specific Features

Multicast communication is essential for scalable content distribution in IP networks, such as IPTV, conferencing, or real-time telemetry. However, multicast networks introduce unique behaviors and failure conditions not observed in traditional unicast flows. These include group dynamics, source replication, RP responsiveness, and IGMP-based control message volatility. Therefore, anomaly detection models must incorporate multicast-specific features that reflect the unique structure and behavior of multicast routing protocols like PIM-SM and IGMPv3.

This section presents features engineered explicitly to detect anomalies in multicast environments, such as RP failures, unauthorised group joins, group churn storms, and multicast flooding attacks. The key multicast features used are shown in Table 16

i. *Key Multicast-Specific Features Used*

Table 16: Multicast-specific features used

Feature Name	**Description**	**Relevance**
group_join_rate	Number of IGMP Join messages per second per host	Detects join floods, instability
group_leave_rate	Frequency of Leave messages per group	Useful in identifying churn
rp_response_delay	Time between Join/Prune message and RP response (e.g., Register or Data)	Indicates RP sluggishness/failure
multicast_tree_depth	Hop distance from source to group receivers (from routing metadata)	Measures efficiency and routing loops
multicast_replication_factor	Number of receivers per multicast group replicated at switch/router level	Can expose flooding or rogue sources
group_membership_variation	Change in active members for a group over time	Behavioral feature of multicast volatility

phantom_group_ratio	Proportion of traffic targeting non-existent groups	Indicates spoofing or scanning
source_flap_frequency	Frequency with which the multicast source switches or restarts	Helps flag unstable or spoofed source behavior

ii. Rationale for Inclusion

Traditional anomaly detection models ignore multicast control planes, assuming symmetric, request-response traffic patterns. Many multicast-specific anomalies are triggered at the control level (IGMP, PIM) but result in subtle data-plane effects like RP blackholing or bandwidth spikes. These features are especially useful for identifying layer-3 issues in multicast trees, routing instabilities, and group-level abuse from edge clients.

iii. Feature Extraction Strategy

Control-plane parsing: IGMP and PIM control messages are parsed from PCAP using custom filters (e.g., tshark -Y "igmp").

Source-tracking: Flow records are correlated with group join/leave logs to estimate churn and replication rates.

RP response time: Measured as the delta between Join request and actual multicast data delivery.

Replication factor: Extracted from NetFlow logs or multicast forwarding counters in switches.

These features were integrated into the same preprocessing pipeline and normalized using Min-Max scaling to maintain consistency with the unicast feature set.

3.4.7 Dataset Application and Insights

In synthetic multicast testbeds, high group_join_rate and delayed rp_response_delay were effective in detecting simulated RP failovers and join storms.

In real-world enterprise logs, multicast_replication_factor spiked during unintended group misuse (e.g., SSDP amplification).

phantom_group_ratio and group_membership_variation helped identify IoT misbehaviour and rogue IGMP hosts.

Flowchart in action

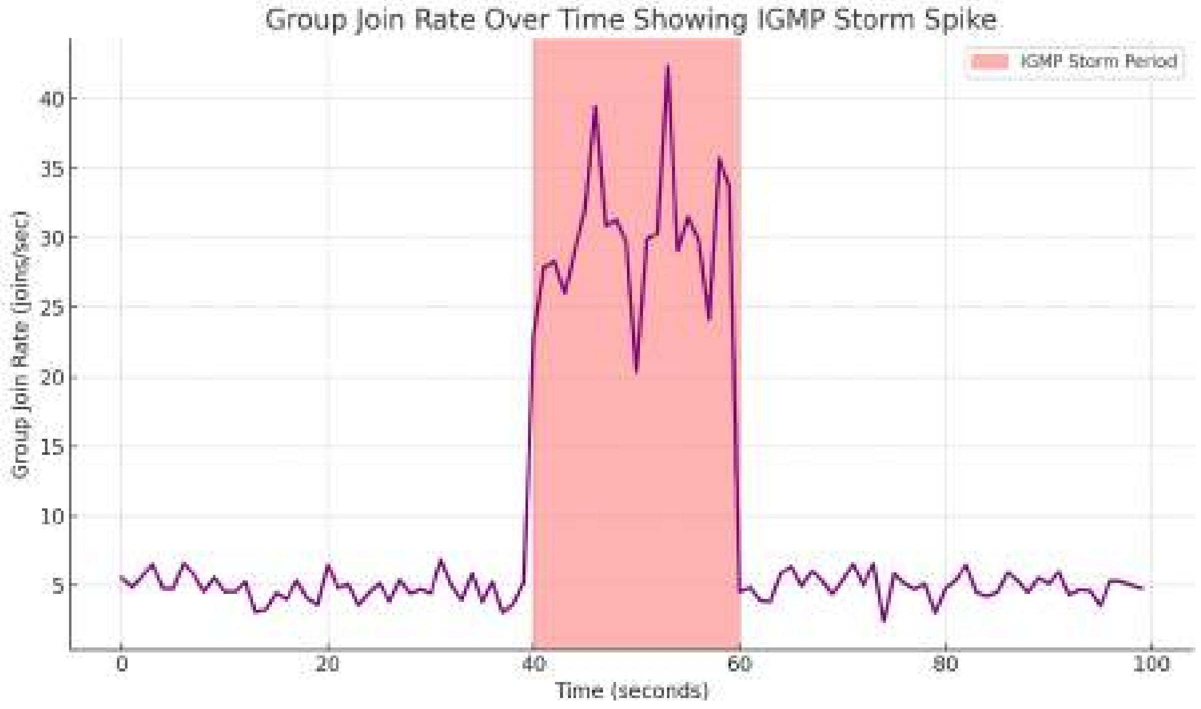

Figure 12: Group Join Rate vs IGMP Storm Spike

Here, in Figure 12, is the line chart showing a spike in group_join_rate during an IGMP storm period. The shaded red area (between time 40 and 60 seconds) highlights the abnormal burst of join messages—an indicator of multicast control-plane abuse.

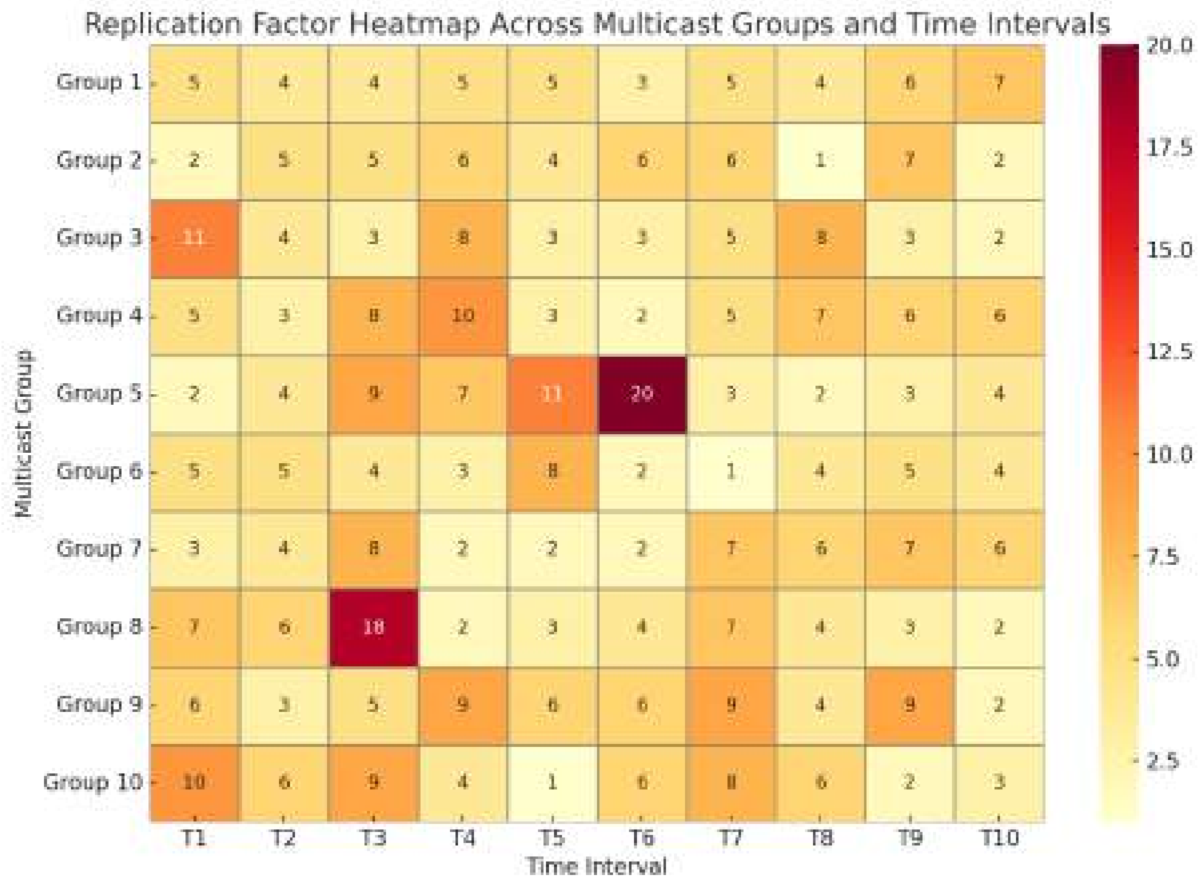

Figure 13: Heatmap of replication factor across multicast groups

Figure 13 is the heatmap of the replication factor across multicast groups and time intervals. The brighter (orange/red) regions indicate higher replication, with visible spikes simulating multicast flooding or unauthorized joins.

3.4.8 Feature Importance and Selection Methods

As the volume and dimensionality of engineered features grow, it becomes essential to identify the most relevant subset for each modelling strategy. High-dimensional data can lead to overfitting, increased computational cost, and reduced model interpretability. This section presents the techniques used to assess feature importance and perform feature selection across various model architectures—ensuring optimal learning, generalization, and explainability. The following techniques are used for feature importance.

i. SHAP (SHapley Additive ExPlanations)

SHAP is a Model-agnostic method based on game theory. It computes the contribution of each feature toward individual predictions and is used for deep learning models, random forests, and hybrid ensemble models. It utilizes visualisations like summary plots, force plots, and waterfall plots to interpret decisions.

ii. LIME (Local Interpretable Model-Agnostic Explanations)

LIME interprets individual predictions by approximating the model locally with a linear interpretable model. It is especially useful for black-box models like autoencoders or GAN-based detectors.

iii. Recursive Feature Elimination (RFE)

RFE is a wrapper-based method that recursively removes least important features. It is effective with models like SVM and logistic regression. It is usually combined with cross-validation for robustness.

iv. Feature Importance from Tree-based Models

Random Forests and Gradient Boosting are the tree-based models which provide native importance scores. Scores are derived from Gini impurity or information gain during training. These are used as a quick filter for identifying and selecting dominant features.

v. Principal Component Analysis (PCA)

PCA is an unsupervised dimensionality reduction that projects data into fewer dimensions, capturing maximum variance. It is used for visualization and noise reduction, especially before GAN and VAE input.

3.4.9 Application Across Models

Different Models and their application is shown in Table 17

Table 17: Models and their application

Model Type	**Selection Technique Used**	**Outcome**
Random Forest	Tree-based importance, SHAP	Reduced noise, faster training
SVM	RFE, PCA	Improved margin generalization

Autoencoder, VAE	PCA (preprocessing), SHAP (post-training)	Reduced reconstruction error
GAN	PCA + manual filtering	Enhanced discriminator precision
Transformer/LSTM	SHAP (attention-weight validation)	Focus on temporal + behavioral features
Hybrid Ensemble	SHAP + Tree-based voting	Adaptive model-specific features

3.4.10 Feature Engineering and Selection

Feature engineering is one of the most critical stages in the machine learning pipeline for network anomaly detection. This process involves transforming raw network traffic data into a structured and meaningful representation that facilitates efficient and accurate anomaly detection by AI models. Poor feature selection can significantly degrade model performance, increase false positives, and reduce interpretability, especially in high-dimensional network data involving complex unicast and multicast flows.

i. *Objectives of Feature Engineering*

- The main objectives of this stage are:
- To capture the essence of network traffic behavior through quantifiable metrics.
- To reduce dimensionality and remove redundant or irrelevant attributes.
- To improve model convergence and prediction accuracy.
- To ensure adaptability to dynamic network environments such as IoT, 5G, and multicast ecosystems.

ii. *Raw Feature Extraction from Network Flows*

Initial features are extracted from raw packet captures (pcaps), NetFlow/sFlow logs, or telemetry records using open-source tools (e.g., Wireshark, nfdump, Zeek). For this study, features were extracted using a combination of Python-based parsers and tools like CICFlowMeter, which generate bi-directional flow features such as:

Basic flow features: source IP, destination IP, source port, destination port, protocol, timestamp.

Statistical features: flow duration, total bytes and packets sent in both directions, average packet size, inter-arrival times.

Behavioral indicators: packet rate, burstiness, flow entropy, and session frequency.

iii. Derived Features for Multicast and IoT Flows

Given the unique behavior of multicast and IoT traffic, additional features were designed specifically for those patterns:

Group Join Rate: frequency at which IGMP/MLD joins are observed.

Replication Factor: number of outgoing interfaces per multicast flow.

Group Lifetime: duration for which a multicast group remains active.

Device Stability Index: variability in traffic behavior from the same source over time.

For IoT-specific flows, device fingerprints, protocol ratios (e.g., MQTT vs HTTP), and periodicity metrics were introduced to distinguish benign vs anomalous devices.

iv. Temporal Feature Engineering

Temporal attributes were engineered to capture evolving behavior over time. Time-based windows (e.g., sliding and tumbling windows) were used to extract time-series aggregates such as:

- Mean packet size over 5-second intervals.
- Peak inter-arrival times during congestion bursts.
- Frequency of anomalous protocol flags (e.g., FIN, URG).

Such temporal features were crucial in training models like RNNs and Transformers that rely on sequence modeling.

3.4.11 Data Preprocessing Steps

All datasets underwent multiple preprocessing operations to improve model input quality as shown in Table 18

Table 18: Data processing Steps

Step	**Technique Used**	**Purpose**
Null Handling	Drop or Impute (mode/mean)	Eliminate missing data
Outlier Detection	Z-score, IQR filtering	Remove noise and extreme values
Encoding	One-hot encoding (protocols, ports)	Convert categorical to numerical features
Normalization	Min-Max scaling, Standard scaling	Standardize feature ranges
Time Formatting	UNIX timestamp conversion, time-window creation	Prepare for RNN and Transformer input

i. Multicast Traffic Simulation and Annotation

Multicast flows are typically underrepresented in public datasets. To bridge this gap:

A **custom multicast traffic simulator** was built using Mininet and PIM-SSM protocol scripts.

IGMP Join/Leave patterns, group replication trees, and flow membership dynamics were programmatically generated.

Anomalies such as **IGMP spoofing**, **register floods**, and **cross-VLAN leaks** were injected at specific intervals. Sample multicast tree is shown in Figure 14

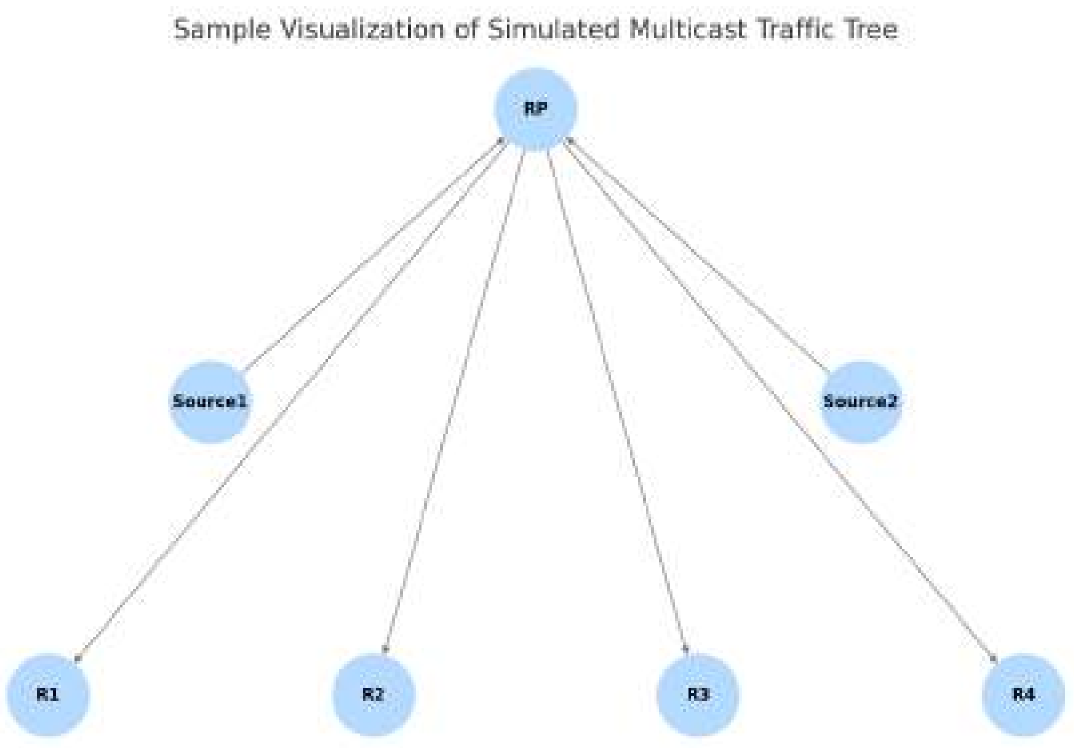

Figure 14: Sample Visualization of Simulated Multicast Tree

ii. Data Splitting and Partitioning

Each dataset was partitioned into training, validation, and test sets. Special consideration was given to time-based splits to simulate real-world progression: The ration of the train test split is shown in Table 19

Table 19: Train test split

Split Type	**Ratio**	**Use Case**
Stratified Split	70/15/15	General model training
Time-based Sequential	Chronological window	Real-time detection scenario
Cross-Validation Blocks	5-fold	Model robustness testing

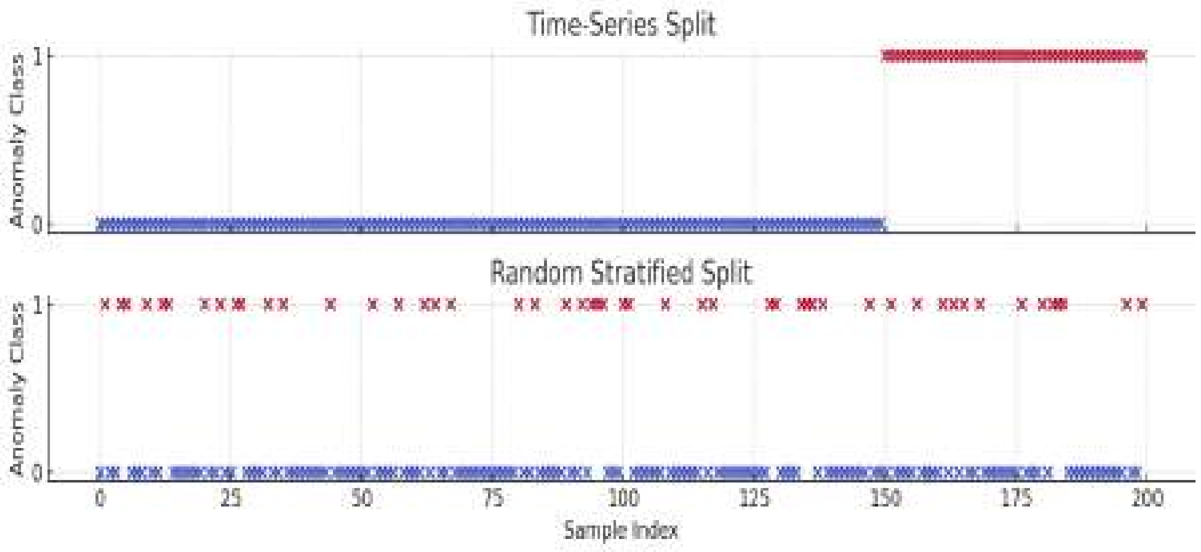

Figure 15: Sample Split for the data

A simple split is shown in Figure 15

iii. Handling Imbalanced Classes

Anomalies typically represent <5% of total records. To address this imbalance:

SMOTE: Synthetic Minority Over-sampling Technique was applied to boost minority class samples. Handling class imbalance is mentioned

Random Undersampling: Removed excess normal flows to balance the dataset.

Class Weights: Applied during model training to penalize false negatives more heavily.

Different method of handling class imbalance is shown in Table 20

Table 20: Handling class imbalance

Technique	**Before (Imbalance Ratio)**	**After (Balanced Ratio)**
No Adjustment	98:2	N/A
SMOTE Only	98:2	65:35
SMOTE + Undersample	98:2	50:50

iv. Data Quality Checks

To ensure consistency, the following checks were applied:

Schema Validation: Ensured uniform column structures across datasets.

Statistical Profiling: Checked for feature variance, skewness, kurtosis.

Traffic Consistency Check: Verified logical flow patterns (e.g., flow duration > 0, byte count > packet count).

3.5 Model Selection and Training

Model selection and training are central to building an effective and scalable anomaly detection system. Given the diversity of network environments and the unique challenges posed by unicast, multicast, and IoT traffic, this study adopted a modular strategy to evaluate traditional machine learning models, deep learning architectures, and advanced generative AI models. Table 21 indicates the model Selection details category-wise.

Table 21: Model selection details

Model	Category	Strengths
Random Forest	Classical ML	Robust to overfitting, handles high-dimensional data well, interpretable
Decision Tree	Classical ML	Simple, fast, interpretable, works well on structured data
Support Vector Machine (SVM)	Classical ML	Effective in high-dimensional spaces, good for binary classification
Logistic Regression	Classical ML	Fast, interpretable, effective for linearly separable problems
Naive Bayes	Classical ML	Efficient, handles categorical features, good probabilistic baseline
K-Nearest Neighbors (KNN)	Classical ML	Non-parametric, simple, captures local structure in data
AdaBoost	Classical ML	Combines weak learners, adaptive to difficult samples
Gradient Boosting	Classical ML	Powerful ensemble, handles complex patterns
XGBoost	Classical ML	Fast, regularized gradient boosting, often top performer
Isolation Forest	Unsupervised ML	Effective for anomaly detection in high-dimensional data
K-Means Clustering	Unsupervised ML	Captures global structure, simple clustering approach
Agglomerative Clustering	Unsupervised ML	Hierarchical, effective for discovering nested data relationships
One-Class SVM	Unsupervised ML	Learns decision boundary around normal data, good for novelty detection
Deep SVDD	Deep Learning	Learns a hypersphere around normal data, effective one-class method
AutoEncoder	Deep Learning	Learns reconstruction, useful for anomaly detection using reconstruction loss
RNN	Deep Learning	Captures temporal dependencies in sequential data
LSTM	Deep Learning	Handles long-term dependencies, useful for time-series anomalies

GRU	Deep Learning	More efficient than LSTM, with similar performance
Transformer	Transformer-based	Captures global context, state-of-the-art for sequence modeling
GAN	Generative Model	Learns data distribution, flags deviations via reconstruction error
VAE	Generative Model	Probabilistic model, models variance in data effectively
Voting Ensemble	Ensemble	Combines predictions from multiple models for robust generalization

3.5.1 Decision Tree

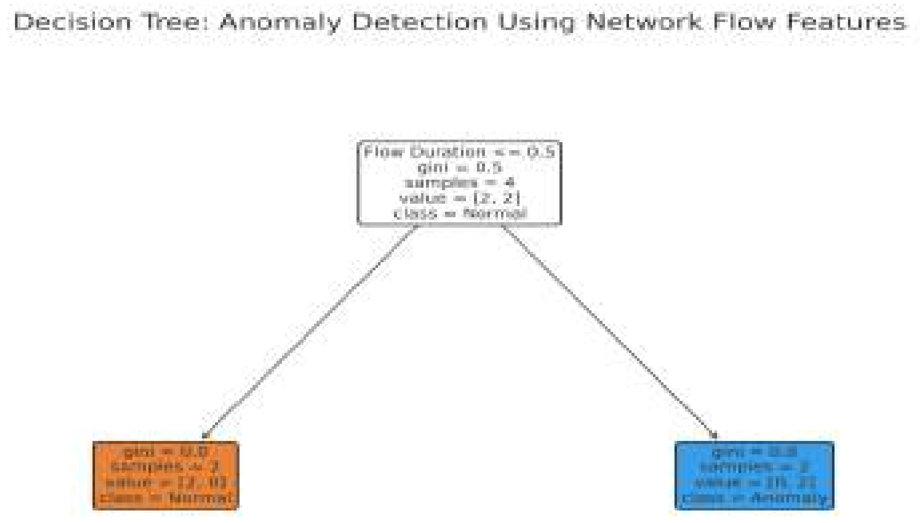

Figure 16: Sample Decision Tree

i. *Model Explanation*

A Decision Tree is a flowchart-like tree structure where each internal node represents a feature condition, each branch represents an outcome of that condition, and each leaf node represents a class label (e.g., "Anomaly" or "Normal"). It partitions the data space into regions by recursively splitting it based on the most informative features using metrics like Gini Impurity or Entropy. Figure 16 shows the sample decision Tree of anomaly detection.

It builds the tree in a top-down, greedy fashion — selecting the best split at each step without backtracking. The result is a set of simple **if-else rules** that mimic human decision-making.

ii. *Reason for using Decision Trees in Anomaly Detection*

- Decision Trees are interpretable, making them ideal for identifying *why* a network flow was flagged as anomalous.

- They perform well on structured tabular data, which is typical of network traffic logs.
- Their fast training and inference time allow them to be used for real-time or near-real-time intrusion detection.
- Trees can uncover nonlinear relationships between features (e.g., traffic rate and packet size jointly indicating an attack).

3.5.2 Random Forest

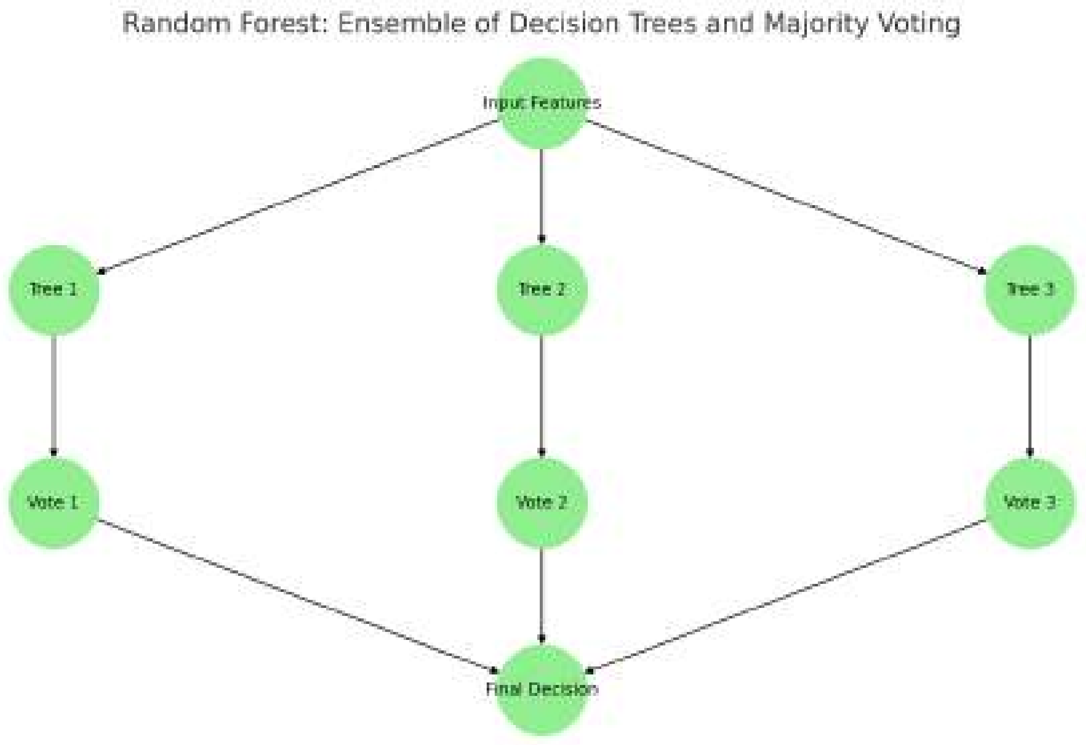

Figure 17: Sample Random Forest

i. Model Explanation

A **Random Forest** is an ensemble learning technique that builds multiple independent **Decision Trees** using random subsets of the training data and feature space. Each tree outputs a prediction, and the **majority vote** among them becomes the final classification. The randomness in sampling and feature selection reduces correlation among trees, improving generalization and reducing overfitting — a key challenge in single-tree models. Figure *17* is used to indicate the group of Decision trees, thereby forming a Random Forest.

ii. Reasons for using Random Forest for Anomaly Detection

- Robustness to noise and outliers makes it highly effective on noisy intrusion detection datasets like CICIDS2017.

- It can handle high-dimensional feature spaces, which are common in network telemetry.
- Feature importance scores aid interpretability, even in large ensembles.
- Performs well even with imbalanced datasets, making it suitable for rare-event anomaly detection.

3.5.3 Support Vector Machine (SVM)

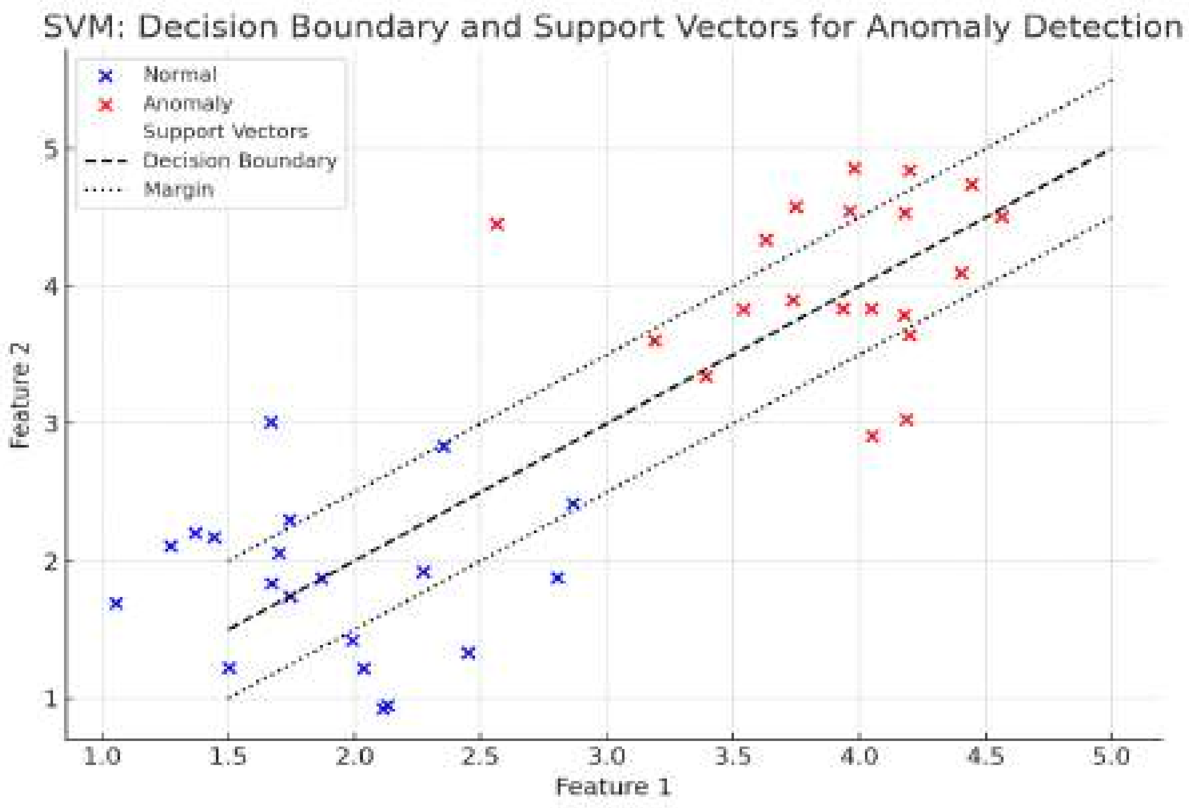

Figure 18: Sample Support Vector Machine

i. Model Explanation

A Support Vector Machine (SVM) is a supervised learning algorithm that finds the optimal hyperplane separating classes in a high-dimensional space. The key idea is to maximize the margin between the closest data points from each class — known as support vectors. SVM can use kernel functions (e.g., linear, radial basis function) to handle nonlinear separations by projecting data into higher dimensions where linear separation becomes possible. Figure *18* indicates a Sample Support vector Machine.

ii. Reason for using SVM Anomaly Detection

- SVM excels in **high-dimensional spaces**, making it suitable for complex network datasets with many features.

- Performs well with **binary classification tasks**, such as “Anomaly vs Normal”.
- **Margin maximization** improves generalization to unseen traffic patterns.
- **Kernel trick** allows modeling nonlinear boundaries for complex anomalies.

3.5.4 Logistic Regression

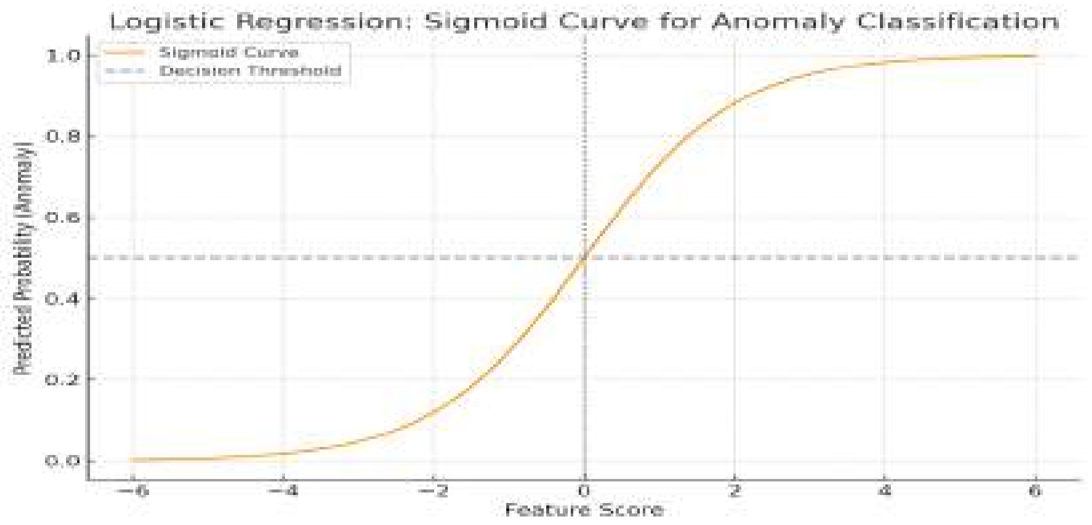

Figure 19: Sample Logistic Regression

Here is the detailed subsection for **Logistic Regression**, including the sigmoid curve visualization and complete explanation:

i. Model Explanation

Logistic Regression is a linear classification algorithm used to predict the probability that an input instance belongs to a particular class. It maps the linear combination of input features to a probability using the sigmoid function:

$$P(y = 1 \mid x) = 1 / (1 + \exp(-(w^t x + b)))$$

This probability has a threshold of 0.5 to classify the input as “Normal” or “Anomaly”. Unlike regression, the output is bounded between 0 and 1, making it ideal for binary classification. Figure *19* shows the Sigmoid Curve

ii. Reason for Using Logistic Regression in Anomaly Detection

- Useful as a fast and simple baseline model.
- Can detect linearly separable anomalies, like those with consistent feature thresholds.
- Interpretable coefficients help understand which features contribute to anomalies.
- Works well in environments with limited compute power.

3.5.5 Naive Bayes

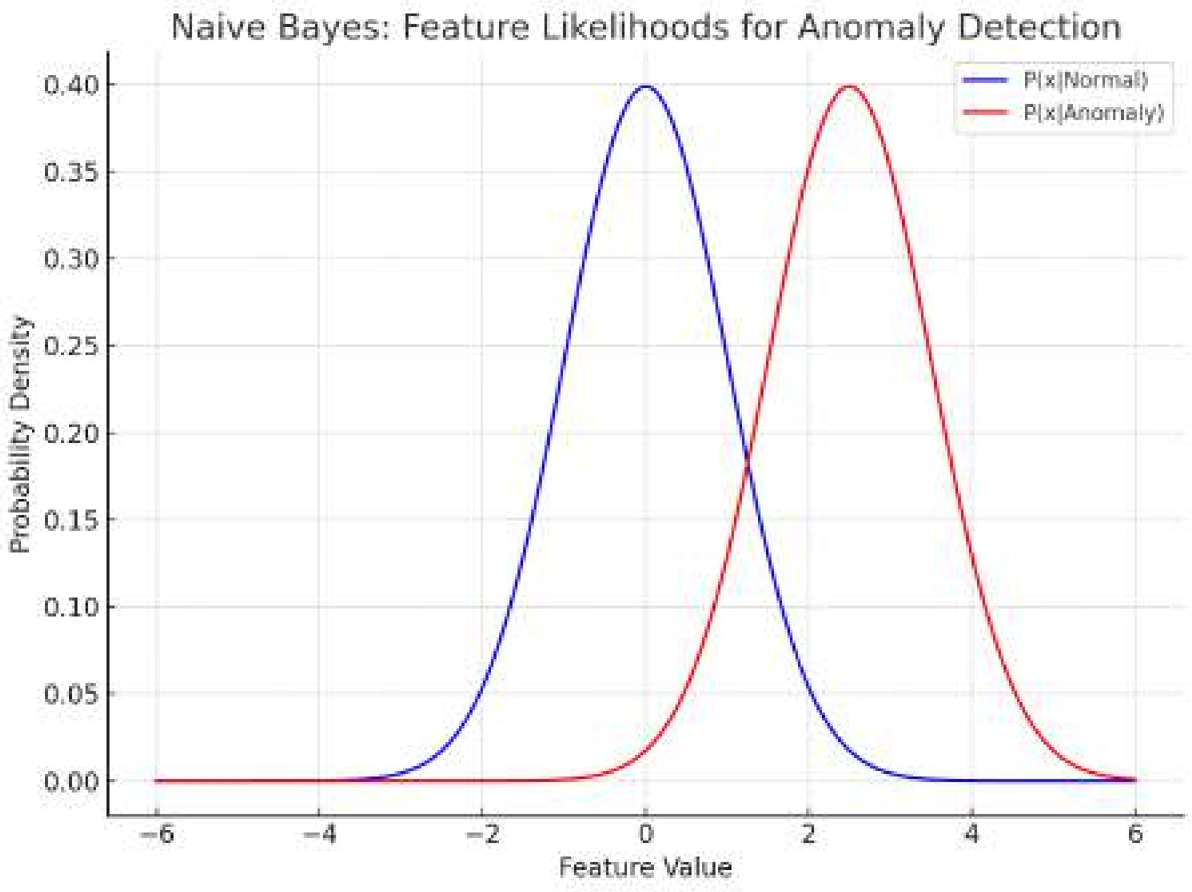

Figure 20: Sample Naïve Bayes

i. Model Explanation

Naive Bayes is a probabilistic classification algorithm based on Bayes' Theorem, with the simplifying assumption that features are conditionally independent given the class label. It computes the posterior probability of each class (e.g., "Normal" or "Anomaly") and assigns the label with the highest probability. Figure *20* indicates the likelihood features for anomaly detection using Naïve Bayes.

$$P(y \mid x) \propto P(y) * \prod[i=1 \text{ to } n]\ P(x\_i \mid y)$$

Despite its simplicity, Naive Bayes often performs competitively in high-dimensional spaces.

ii. Reasons for Choosing Naïve Bayes for Anomaly Detection

- Performs well in high-dimensional settings, especially with text or packet metadata.
- Probabilistic output enables threshold-based anomaly scoring.

- Fast training and prediction, suitable for real-time systems.
- Good for categorical or discrete data, like protocol types or port numbers.

3.5.6 K-Nearest Neighbours (KNN)

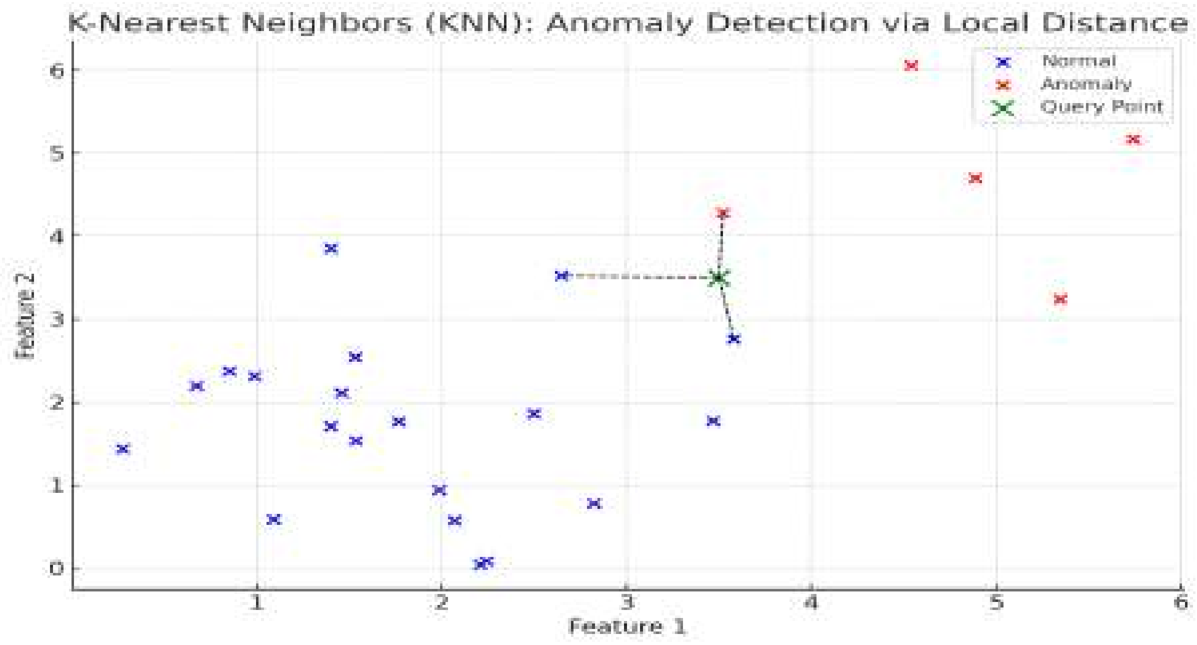

Figure 21: Sample K-Nearest Neighbours

i. Model Explanation

K-Nearest Neighbours (KNN) is a non-parametric, instance-based learning method. It classifies an input sample based on the majority class of its k-nearest neighbours in the feature space using a distance metric such as Euclidean distance. For anomaly detection, a test point is flagged as anomalous if it lies far from clusters of known “Normal” data points, i.e., its neighbours are too distant or belong to a different class. Figure *21* indicates the sample K-Nearest Neighbours.

ii. Reasons for Choosing K-NN for Anomaly Detection

- Effective at capturing local data structures, which is ideal for behavioral anomalies.
- Works well in low-dimensional, structured feature spaces often found in network telemetry.
- Detects distance-based outliers effectively, even when they don’t form distinct clusters.
- No training phase — useful in rapidly changing environments.

3.5.7 AdaBoost

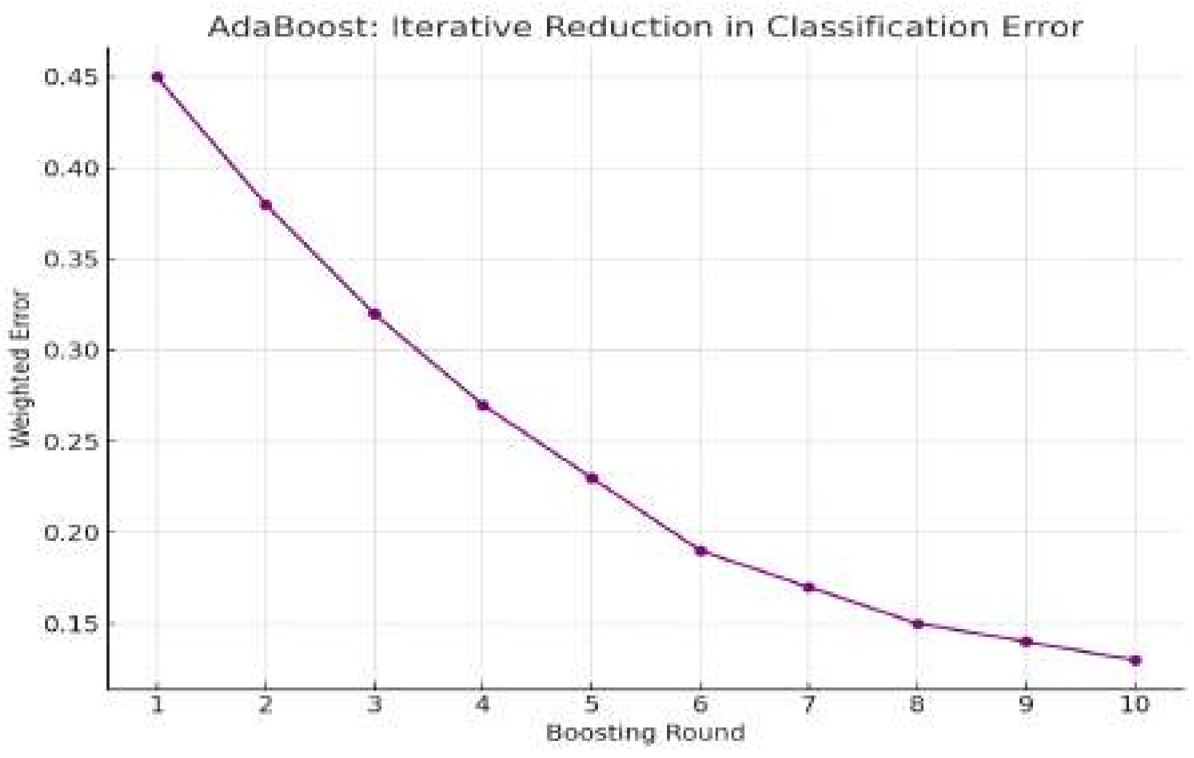

Figure 22: Sample Ada Boost

i. Model Explanation

AdaBoost (Adaptive Boosting) is an ensemble method that combines multiple **weak learners** (typically shallow decision trees) into a single **strong classifier**. Each learner is trained sequentially, with **greater focus on previously misclassified instances**. The algorithm assigns weights to samples based on their classification difficulty. Misclassified samples get more weight in the next round, pushing the next model to learn patterns that the previous one missed. Sample Ada boost is shown in Figure *22*

ii. Reasons for Choosing Adaboost for Anomaly Detection

- Boosting is powerful at learning **complex decision boundaries**, improving detection of subtle anomalies.
- Performs well on **imbalanced datasets**, which is common in anomaly detection.
- Sequential learning helps the model to **adapt to difficult-to-classify traffic patterns**.
- **Lightweight base learners** make it efficient even in layered model configurations.

3.5.8 Gradient Boosting

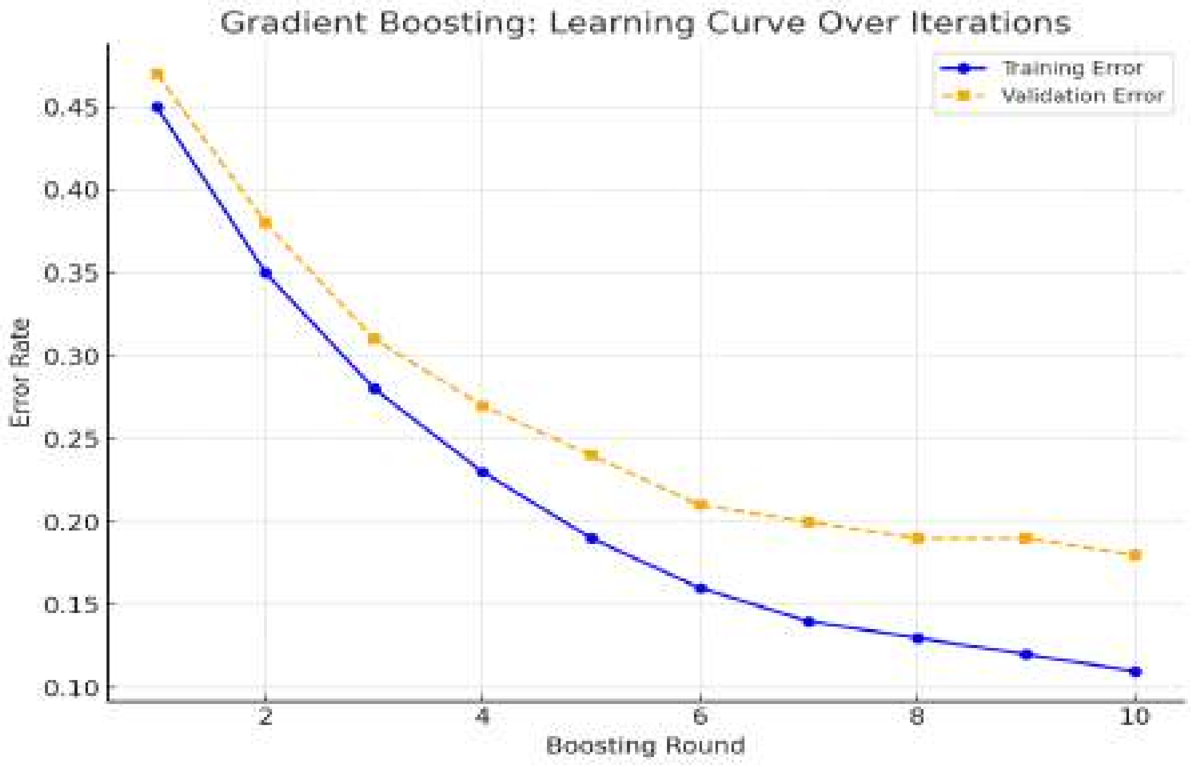

Figure 23: Sample Gradient Boosting

i. Model Explanation

Gradient Boosting as shown in Figure *23* builds an ensemble of weak learners (typically decision trees) sequentially, similar to AdaBoost. However, instead of re-weighting data points, each learner is trained to minimize the residual error (loss gradient) of the entire ensemble on the training set. At each stage, the model fits to the negative gradient of the loss function, hence the name *gradient* boosting. This allows highly flexible modelling of complex nonlinear functions.

ii. Reasons for using Gradient Boosting Anomaly Detection

- Handles **complex, nonlinear patterns** common in real-world anomalies.
- Excellent for **tabular network data**, especially with mixed data types.
- Works well with **imbalanced datasets** when combined with loss adjustments or class weighting.

- Captures **interactions between features**, critical in detecting subtle multi-feature anomalies.

3.5.9 XGBoost

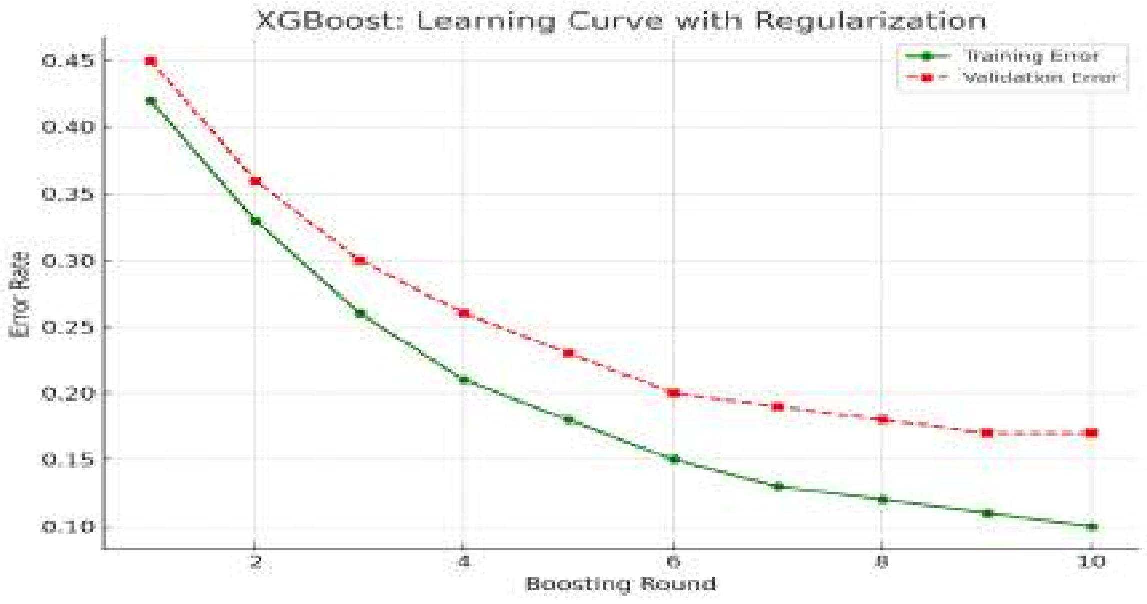

Figure 24: Sample XGBoost

i. Model Explanation

XGBoost (Extreme Gradient Boosting) as shown in Figure *24* is an optimized and regularized implementation of Gradient Boosting. It introduces several improvements, such as:

- **Regularization** (L1 and L2) to prevent overfitting.
- **Parallelized tree construction** for speed.
- **Weighted quantile sketch** for efficient tree pruning.
- **Missing value handling** built-in.

It builds additive tree models where each new tree corrects the residuals of previous trees using gradient descent on a custom loss function. .

ii. Reason for choosing XGBoost for Anomaly Detection

- Known for **state-of-the-art performance** on structured datasets like CICIDS and KDD Cup.
- **Robust to noisy or imbalanced data** — useful in real-world network anomalies.
- **Fast training and scalable**, making it suitable for large datasets.

- Allows for **fine-tuned control** over learning dynamics, boosting anomaly detection accuracy.

3.5.10 Isolation Forest

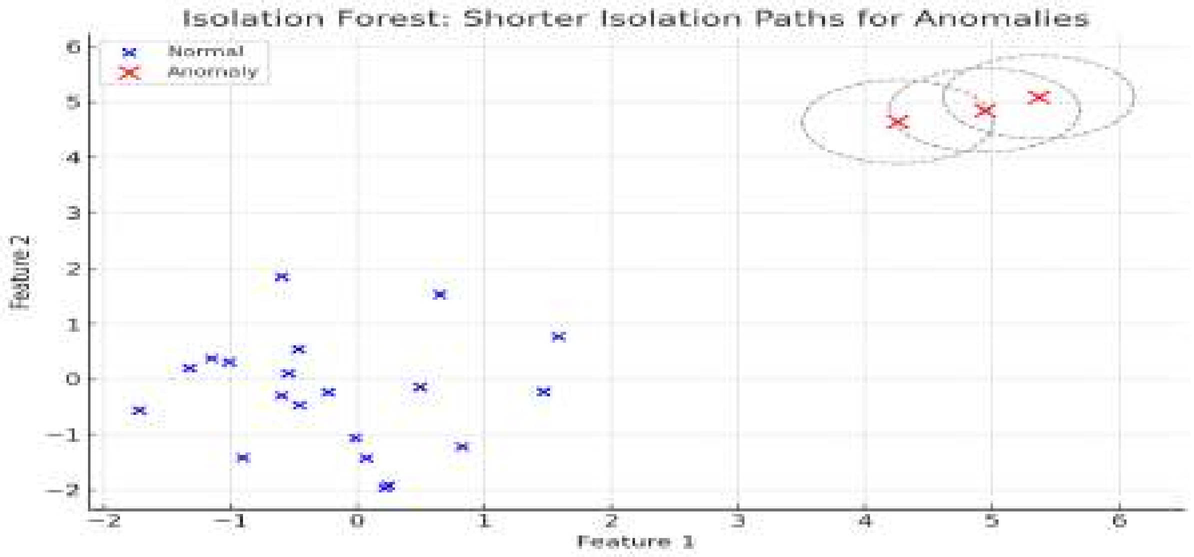

Figure 25: Sample Isolation Forest

i. Model Explanation

Isolation Forest as shown in Figure *25* is an unsupervised anomaly detection algorithm that identifies anomalies by randomly partitioning the feature space using binary trees. Unlike most algorithms that model normal instances, Isolation Forest instead isolates anomalies directly. Anomalies tend to be less frequent and differ significantly from normal data — hence, they are isolated faster, i.e., with shorter average path lengths in the tree structure.

ii. Reason for Choosing Isolation Forest Anomaly Detection

- Specifically designed for unsupervised anomaly detection.
- Highly efficient on large, high-dimensional datasets — ideal for network logs and telemetry.
- No prior labelling required, making it suitable for zero-day or novel attack detection.
- Effectively captures sparse or scattered anomalies.

3.5.11 K-Means Clustering

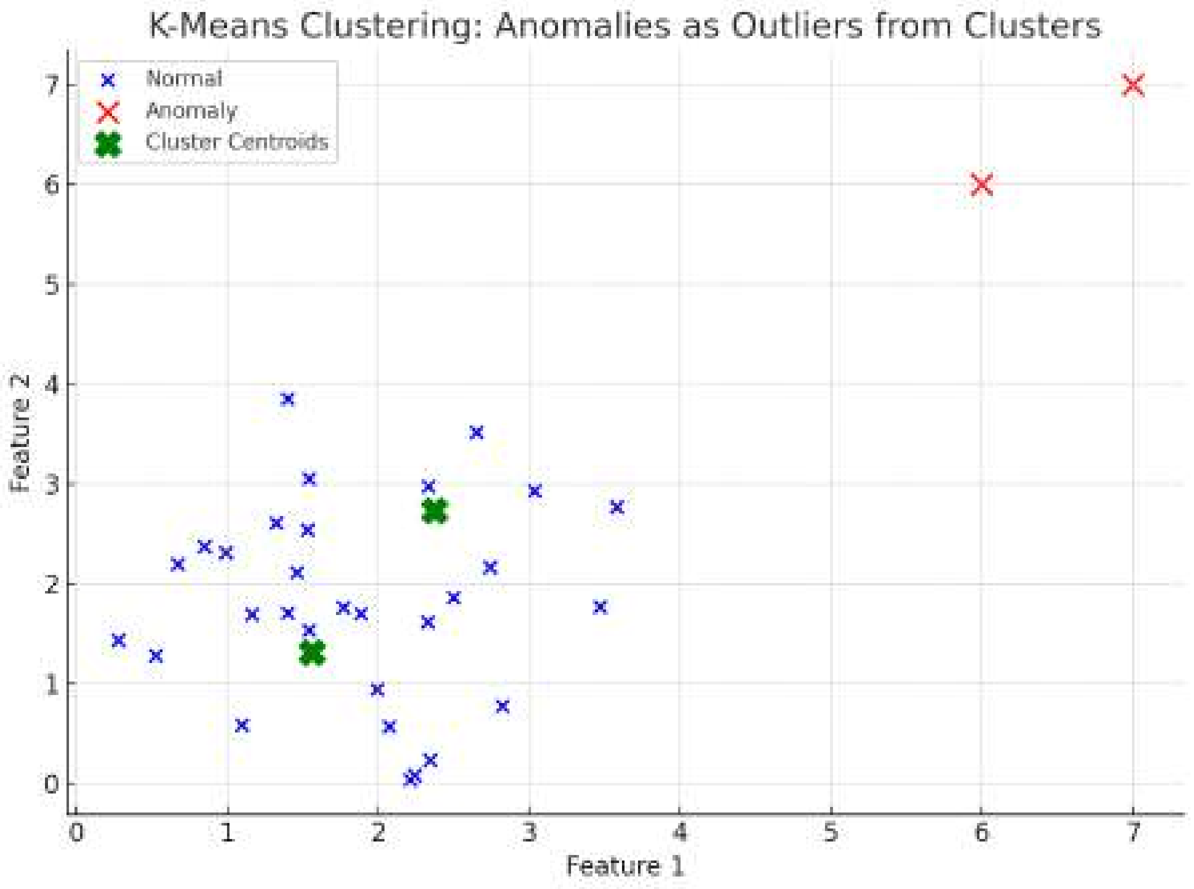

Figure 26: Sample K-Means Clustering

i. Model Explanation

K-Means as shown in Figure *26* is an unsupervised clustering algorithm that partitions the dataset into k clusters by minimizing the intra-cluster variance. Each data point is assigned to the cluster with the nearest centroid, which is updated iteratively. In anomaly detection, anomalies are treated as data points that are far from any cluster centroid, making them outliers based on distance from cluster centers.

ii. Reasons for Choosing K-Means Anomaly Detection

- Captures the **global structure** of the dataset by forming distinct groupings.
- Allows identification of **outliers** as points far from any centroid.
- Useful for **preprocessing** or **coarse anomaly filtering**.
- No labels required — suitable for **unsupervised scenarios**.

3.5.12 Agglomerative Clustering

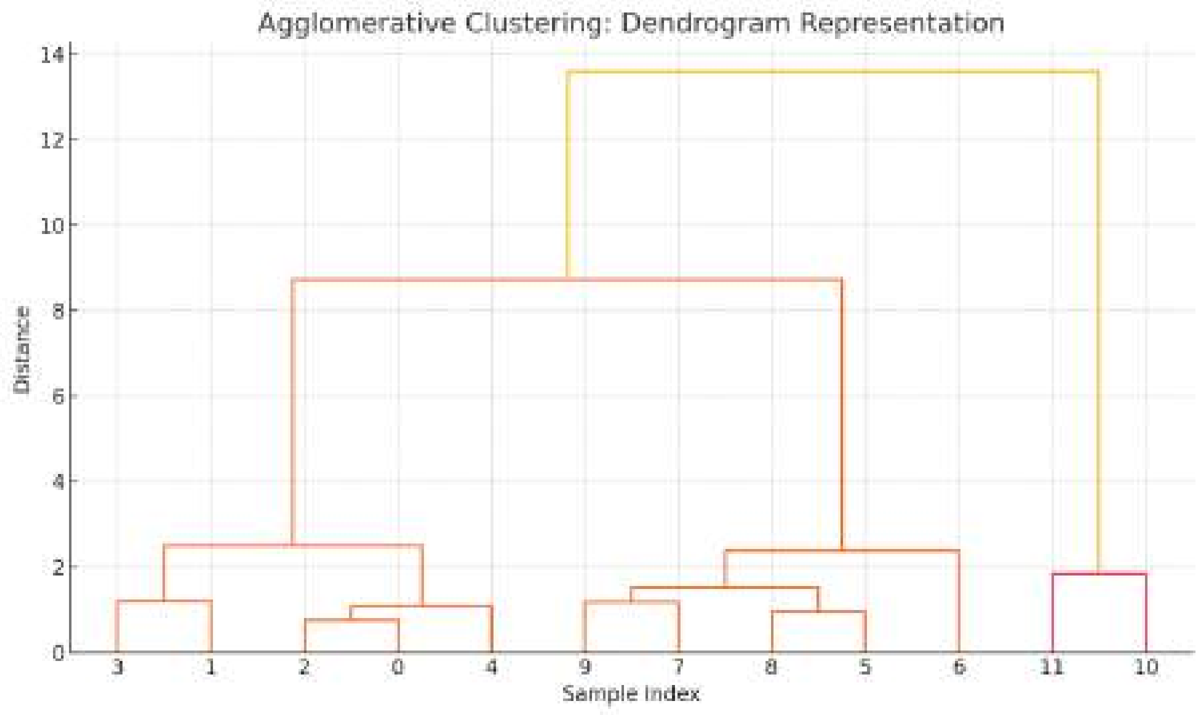

Figure 27: Sample Agglomerative Clustering

i. Model Explanation

Agglomerative Clustering as shown in Figure *27* is a type of hierarchical clustering that begins with each point as its own cluster and iteratively merges the closest pairs based on a linkage criterion (e.g., Ward's method, single, complete, average linkage). The result is a dendrogram that shows the hierarchy of merges, which can be cut at different heights to yield varying numbers of clusters. Anomalies can be detected as points that merge late, indicating they are far from other clusters.

ii. Reasons for Choosing Agglomerative Clustering for Anomaly Detection

- Reveals **nested data structure**, identifying fine-to-coarse anomaly groupings.
- Does not require prior knowledge of the number of clusters.
- Well-suited for **small-to-medium-sized datasets** with an underlying hierarchical structure.
- Enables **visual anomaly discovery** through dendrogram structure.

3.5.13 One-Class SVM

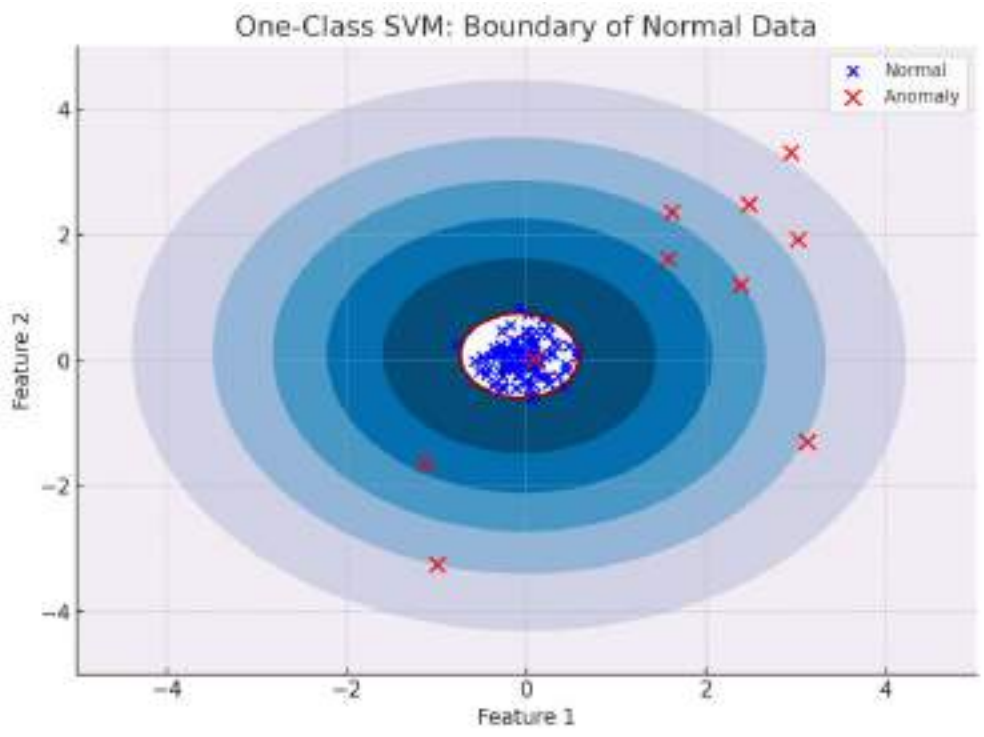

Figure 28: Sample One Class SVM

i. Model Explanation

One-Class SVM as shown in Figure *28* is an unsupervised learning algorithm used primarily for **novelty detection**. It learns a **decision function** that defines a boundary around the **majority (normal) class** in feature space. Any instance falling outside this boundary is considered an **anomaly**.

It relies on a kernel function (commonly **RBF**) to map data into a high-dimensional space where it can find a separating hyperplane that encloses most of the data.

ii. Reasons for using One-Class SVM for Anomaly Detection

- Specifically designed to model only normal data — ideal for scenarios with few or no labelled anomalies.
- Effective in high-dimensional settings with non-linear boundaries.
- Capable of detecting novel or rare patterns in streaming or online network data.
- Requires only a single class of training data (normal flows).

3.5.14 Deep SVDD (Support Vector Data Description)

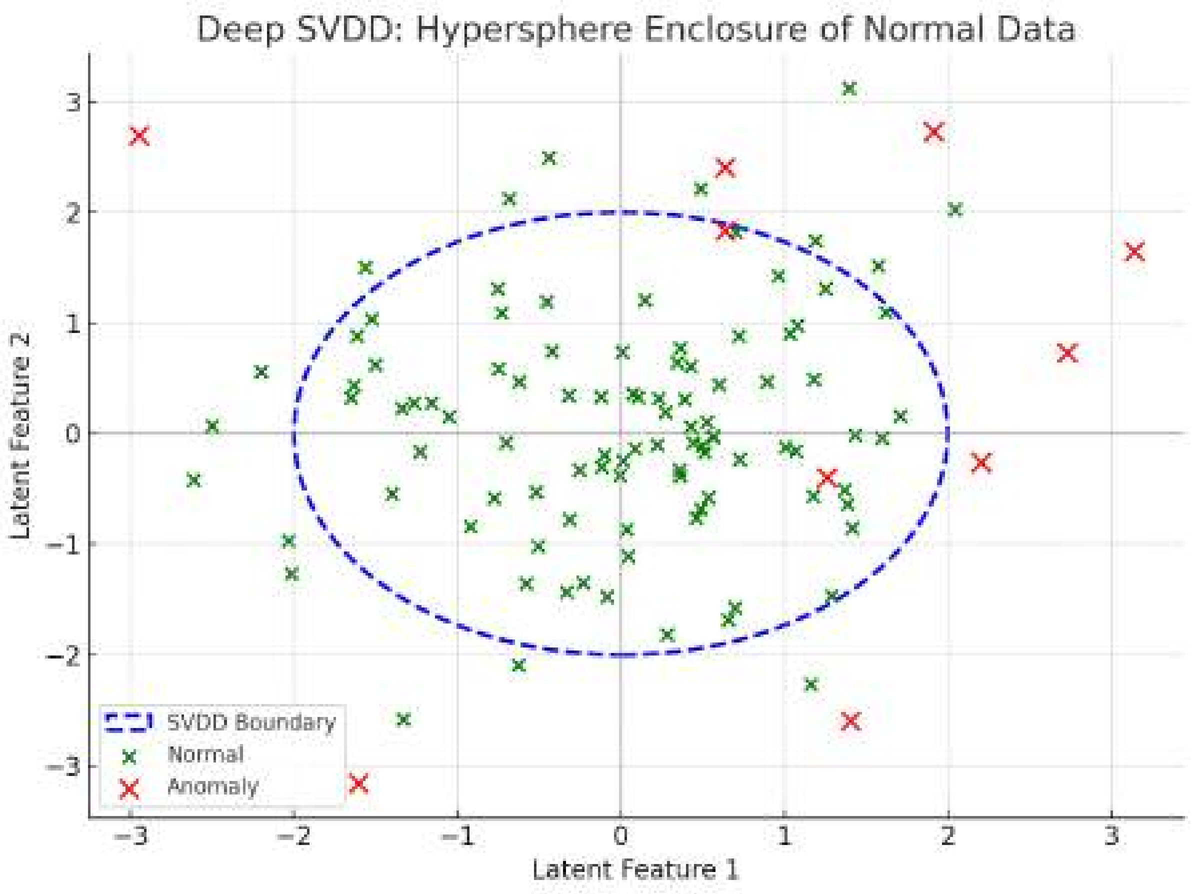

Figure 29: Sample Deep SVDD

i. Model Explanation

Deep SVDD as shown in Figure *29* is a deep learning-based extension of Support Vector Data Description. It trains a neural network to map input data into a **latent space** and learns a **hypersphere** that tightly encloses the normal data in that space. Anomalies are identified as those points that **fall outside** this minimal enclosing hypersphere. Unlike One-Class SVM, Deep SVDD **learns a compact representation of the data** through an embedded neural network, which enhances its generalization on complex inputs.

ii. Reasons for Choosing Deep SVDD for Anomaly Detection

- Particularly effective for **deep feature learning** in high-dimensional data like flow vectors or packet embeddings.
- Designed for **one-class learning**, ideal for environments with abundant normal samples but rare or unseen anomalies.
- Combines **representation learning and anomaly scoring** in a single end-to-end framework.

3.5.15 Autoencoder

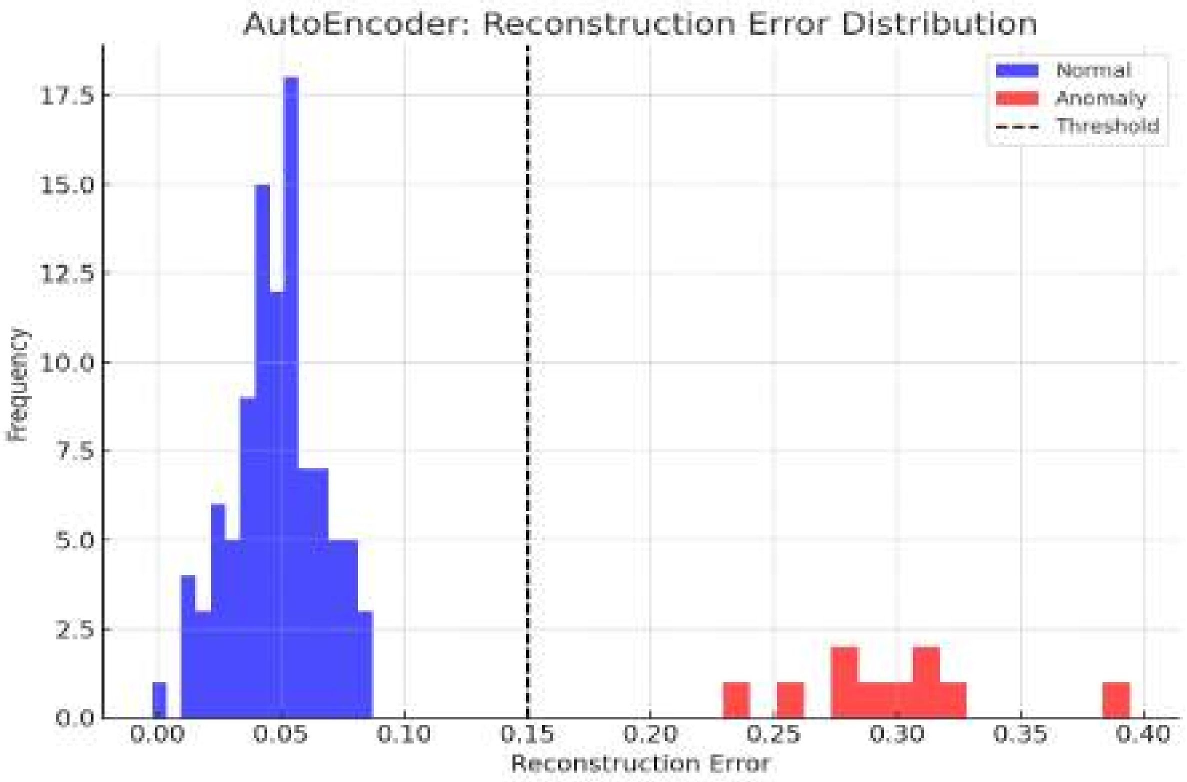

Figure 30: Sample AutoEncoder

i. Model Explanation

An AutoEncoder as shown in Figure *30* is a neural network designed to learn a compressed representation (encoding) of input data and then reconstruct it as closely as possible. It consists of:

- Encoder: Maps input to a latent representation.
- Decoder: Reconstructs input from the latent code.

For anomaly detection, the AutoEncoder is trained only on normal data, so it reconstructs such instances well. Anomalies, which differ in structure, have high reconstruction errors — making them easily detectable.

ii. Reasons for using AutoEncoders for Anomaly Detection

- Can detect subtle, **nonlinear anomalies** in complex datasets.
- Learns feature compression, highlighting **out-of-distribution patterns**.
- Unsupervised: requires only **normal samples for training**.
- Suitable for network telemetry, sensor data, or sequence flows.

3.5.16 Recurrent Neural Network (RNN)

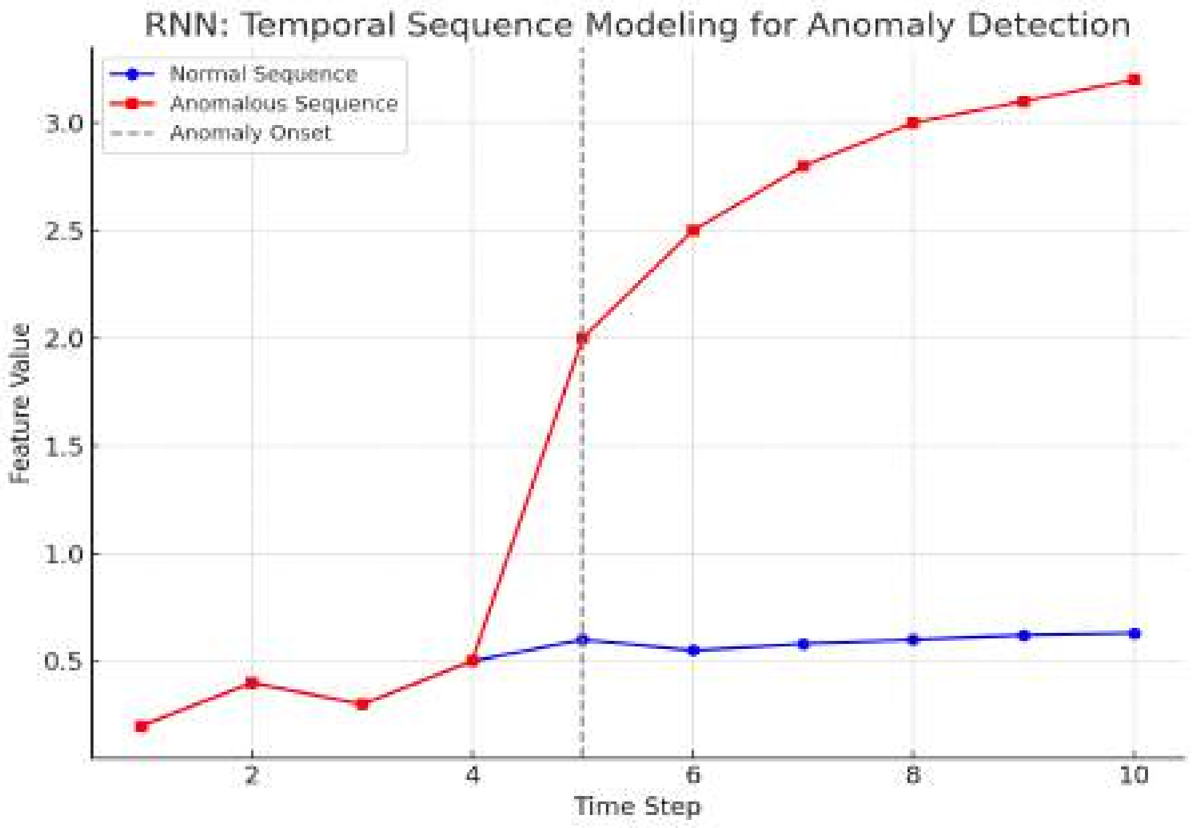

Figure 31: Sample Recurrent Neural Network

i. Model Explanation

RNNs as shown in Figure *31* are neural networks designed for **sequential data**, where the current output depends on previous inputs. They maintain a **hidden state** across time steps, enabling them to capture **temporal dependencies**. In anomaly detection, RNNs can model **expected patterns over time** in metrics like traffic rates, packet intervals, or flow durations. Deviations from learned patterns trigger anomaly flags.

ii. Reasons for Choosing RNNs for Anomaly Detection

- Capable of detecting **temporal anomalies** that manifest over multiple time steps.
- Suitable for **log data, telemetry streams, and packet traces**.
- Learns sequential behavior without feature engineering.
- Ideal for **time-series anomaly detection tasks** (e.g., periodic spikes, missing heartbeats).

3.5.17 Long Short-Term Memory (LSTM)

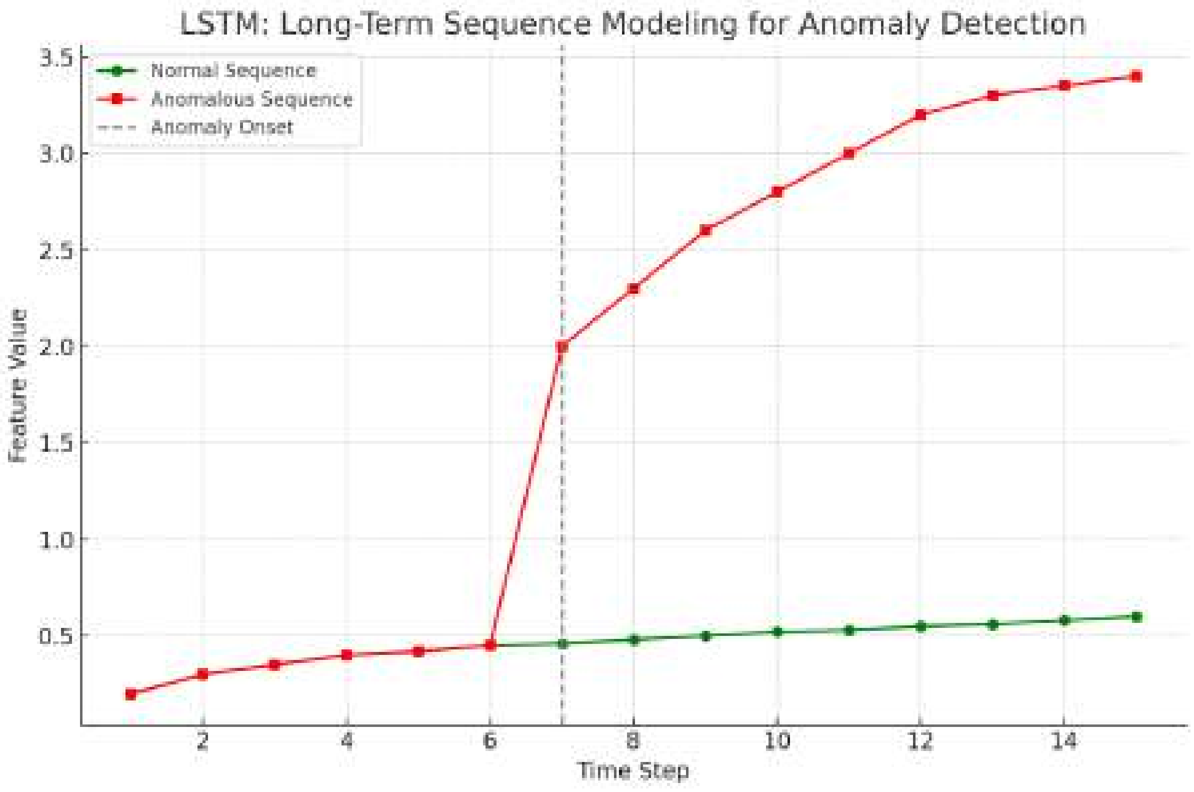

Figure 32: Sample LSTM

i. Model Explanation

LSTM as shown in Figure *32* is a type of recurrent neural network (RNN) designed to capture **long-term dependencies** in sequential data. It introduces **memory cells** and **gating mechanisms** — input, output, and forget gates — to retain or discard information over time. In anomaly detection, LSTMs learn the **temporal evolution** of normal sequences. Anomalies are identified as deviations in predicted values over longer time windows.

ii. Reasons for Choosing LSTM for Anomaly Detection

- Handles **long-range dependencies** better than vanilla RNNs.
- Excellent for **time-series-based network logs and telemetry data**.
- Detects **complex temporal anomalies**, like delayed responses or burst attacks.
- Useful for **contextual anomaly detection**, where past events matter.

3.5.18 Gated Recurrent Unit (GRU)

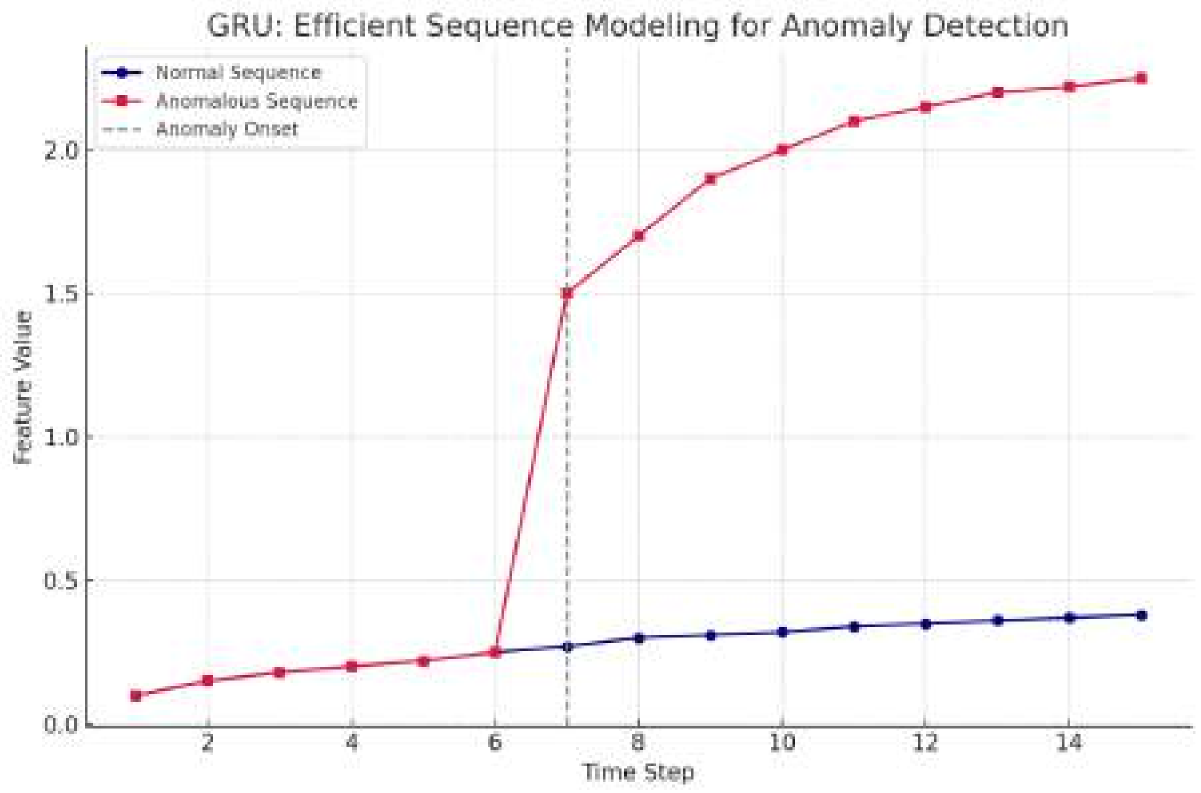

Figure 33: Sample GRU

i. Model Explanation

GRU as shown in Figure *33* is a variant of the LSTM designed to capture sequential patterns using a simplified gating mechanism. It merges the forget and input gates into a single update gate, and also uses a reset gate, reducing the number of parameters compared to LSTM. In anomaly detection, GRUs are used to learn the temporal structure of normal data and flag deviations in sequence behavior.

ii. Reason for choosing GRU for Anomaly Detection

- Captures **temporal dependencies** efficiently with fewer resources than LSTM.
- Suitable for **real-time sequence anomaly detection** in resource-constrained environments.
- Balances **performance and computational cost** in time-series analysis.
- Good for network logs, flow sequences, and monitoring telemetry.

3.5.19 Transformers

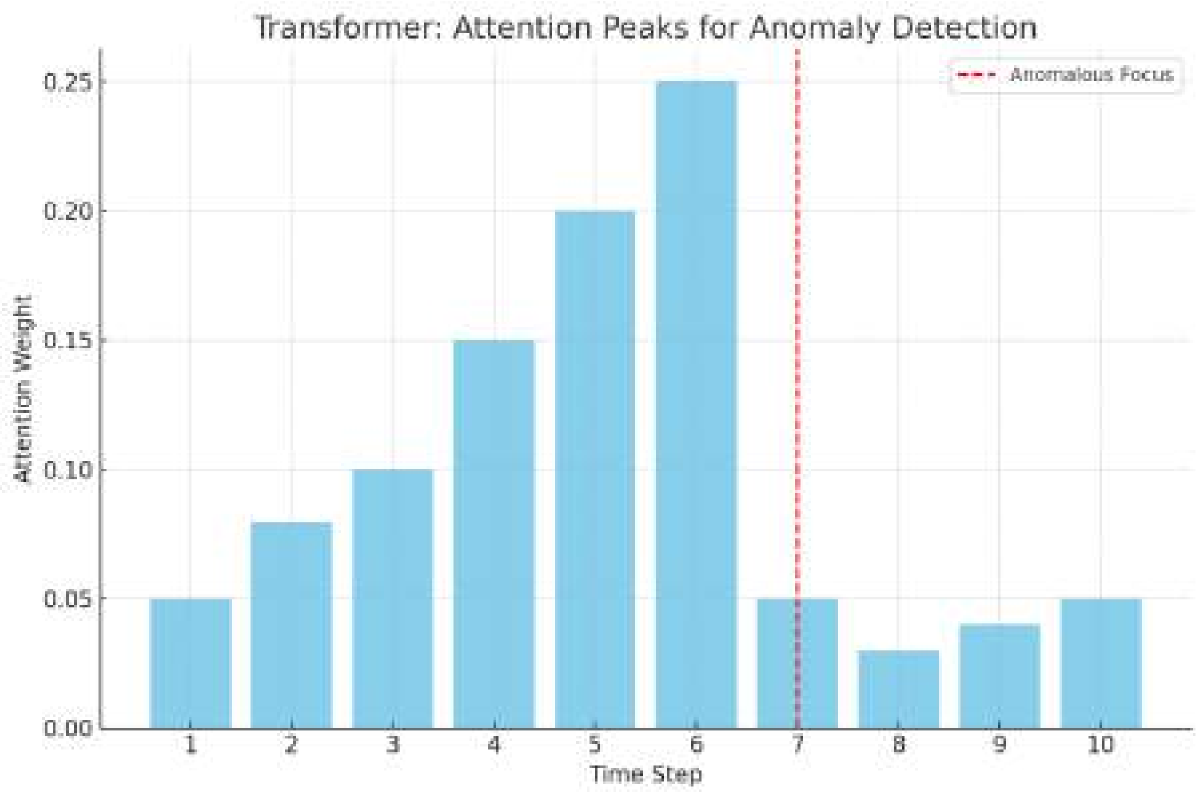

Figure 34: Sample Transformer

i. Model Explanation

Transformers as shown in Figure *34* are deep learning architectures that leverage self-attention mechanisms to model relationships between elements in a sequence, regardless of their positions. Unlike RNNs and LSTMs, Transformers process all elements in parallel, using attention weights to focus on important time steps. In anomaly detection, Transformers can learn to attend more strongly to unusual events or patterns in a sequence, identifying anomalies based on attention score distribution or deviations from predicted outputs.

ii. Reason for Choosing Transformers for Anomaly Detection

- Captures long-range dependencies efficiently across sequences.
- Excellent for contextual and collective anomaly detection.
- Works well on multivariate time-series, logs, or graph data.
- Attention mechanism gives insight into why a point is anomalous.

3.5.20 Generative Adversarial Network (GAN)

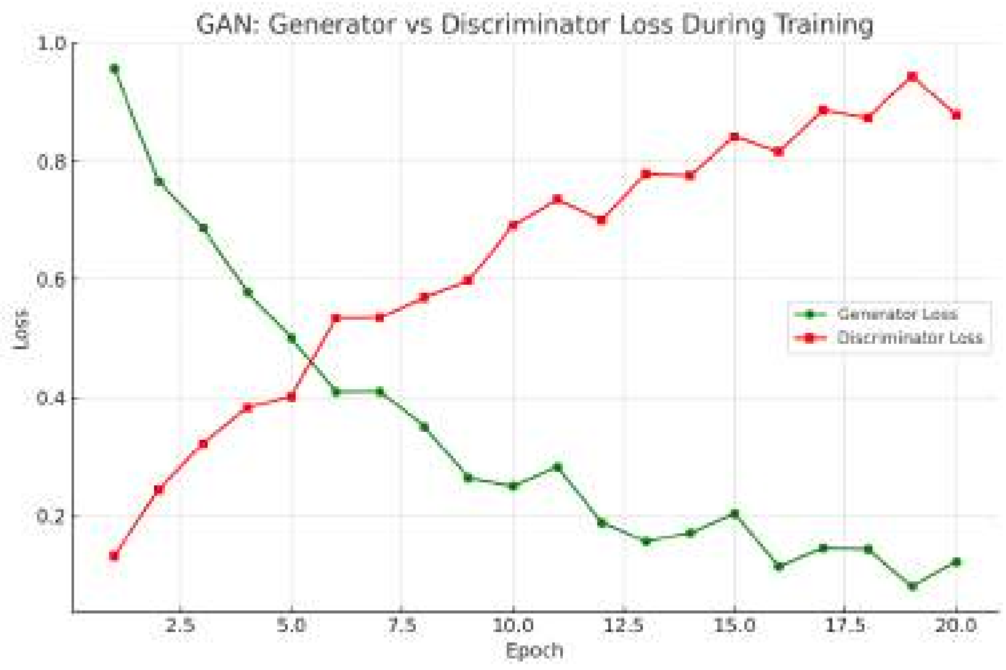

Figure 35: Sample GAN

i. Model Explanation

GANs as shown in Figure *35* consist of two competing neural networks:

- A Generator (G) that tries to create synthetic data indistinguishable from real data.
- A Discriminator (D) that attempts to distinguish between real and generated samples.

The training objective is a minimax game, where both networks improve iteratively. In anomaly detection, GANs can be trained on normal data and used to flag inputs with high reconstruction error or poor discriminator confidence as anomalies.

ii. Reasons for Choosing GAN for Anomaly Detection

- Learns to replicate the distribution of normal data, making it sensitive to deviations.
- Useful for image-based, time-series, or high-dimensional structured data.
- Generates synthetic normal instances for comparison or data augmentation.
- Effective in unsupervised or semi-supervised anomaly detection tasks.

3.5.21 Variational AutoEncoder (VAE)

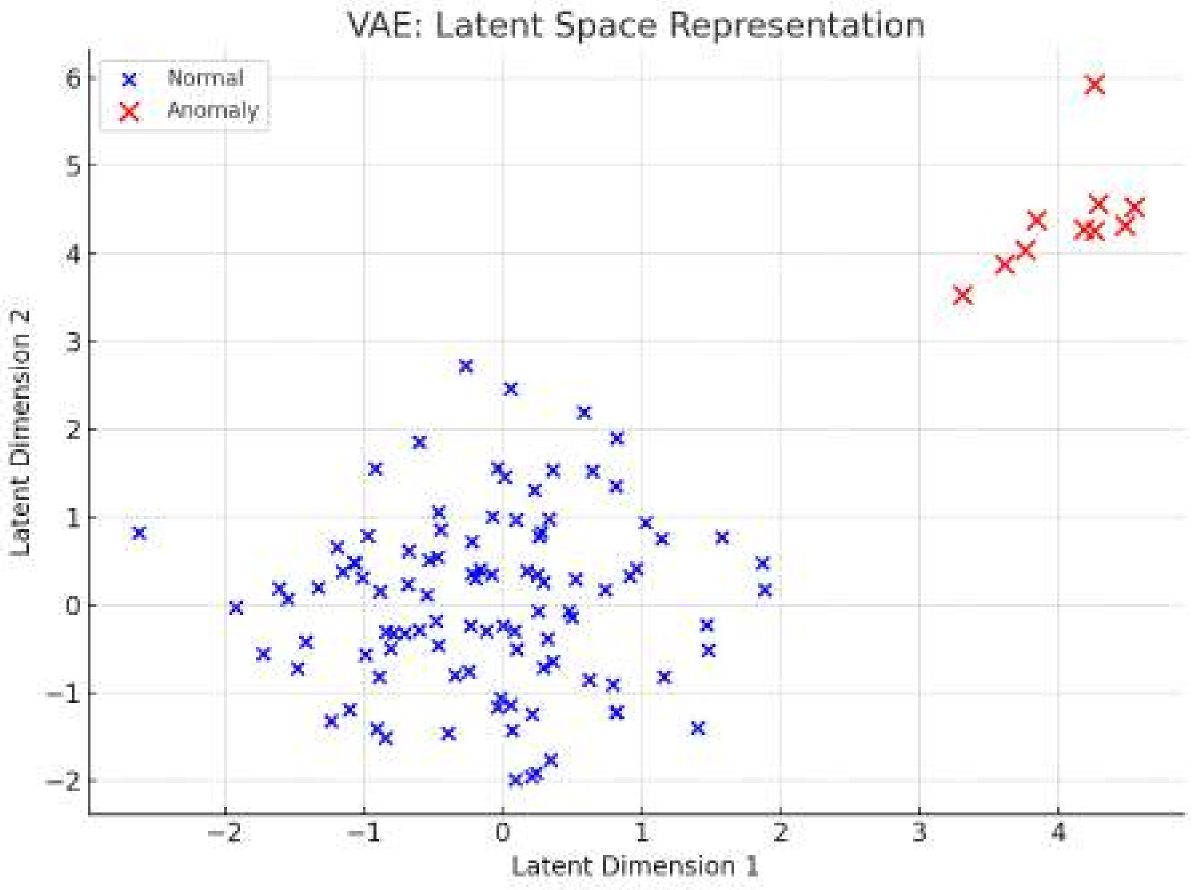

Figure 36: Sample VAE

i. *Model Explanation*

A **VAE** as shown in Figure *36* is a generative model that combines deep learning and variational inference. It encodes input data into a **probabilistic latent space** and learns to reconstruct it while minimizing both:

- **Reconstruction Loss** (how close the output is to the input), and
- **KL Divergence** (how far the latent distribution deviates from a standard normal distribution).

In anomaly detection, anomalies typically have **higher reconstruction error** and **less probable latent encodings** than normal data.

ii. *Reason for choosing VAE for Anomaly Detection*

- Learns to model both data reconstruction and latent distribution, enabling dual anomaly scoring.
- Well-suited for structured, high-dimensional, and multivariate data.

- Generates uncertainty-aware encodings, improving robustness.
- Can flag anomalies based on reconstruction loss or latent likelihood.

3.5.22 Voting Ensemble

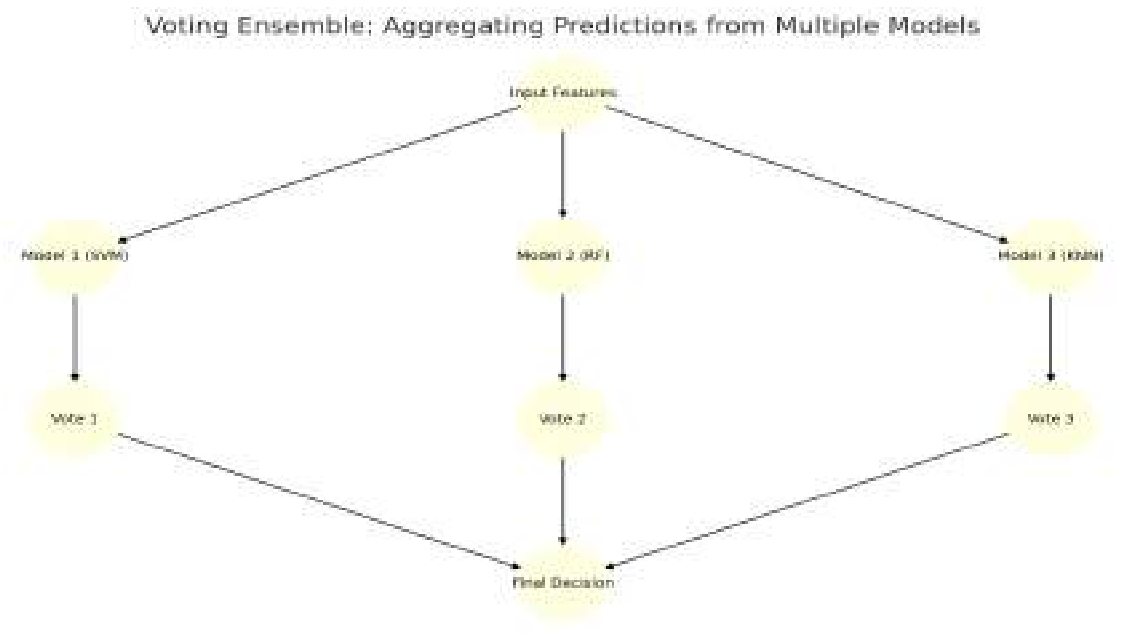

Figure 37: Sample Voting Ensemble

i. Model Explanation

A Voting Ensemble as shown in Figure *37* combines predictions from multiple base models to produce a single, more robust output. Each model casts a "vote" on the class label, and the ensemble outputs:

- Majority class (hard voting) or
- Weighted average of probabilities (soft voting).

It leverages the diversity of models — such as combining SVM, Random Forest, and KNN — to reduce individual bias and variance.

ii. Reasons for Choosing Ensembling for Anomaly Detection

- Improves robustness by combining different perspectives of anomaly scoring.
- Reduces false positives by balancing aggressive and conservative models.
- Effective on heterogeneous datasets with nonuniform data behavior.
- Enables leveraging both generative and discriminative models in tandem.

3.5.23 Overall Comparison

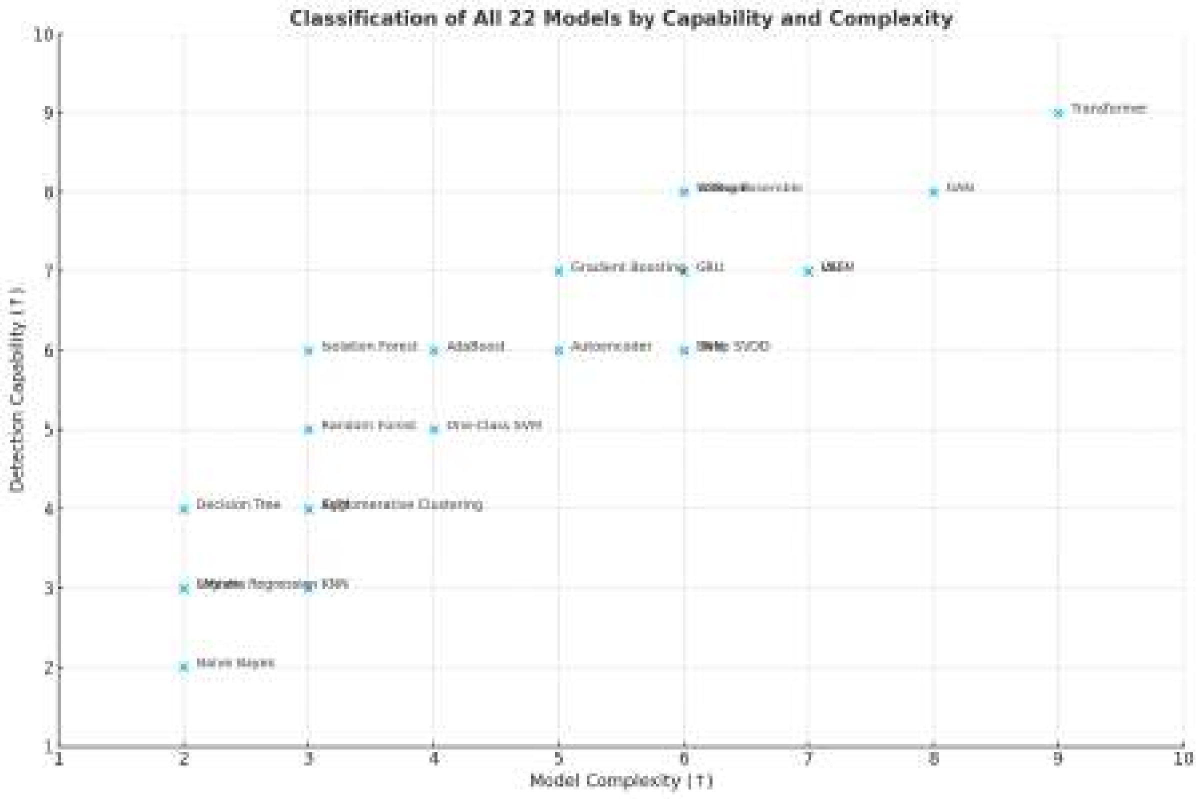

Figure 38: Overall Comparison of all 22 Models

iii. Model Training Pipeline

Each model underwent a standardized training pipeline:

Dataset Loading: Preprocessed training and validation sets loaded.

Model Initialization: Configured with selected hyperparameters.

Training & Validation:

Classical ML: sklearn with 5-fold cross-validation.

Deep Learning: Keras/PyTorch with early stopping and dropout.

Hyperparameter Tuning:

Randomized search + grid search.

Bayesian optimization for deep models.

Evaluation:

Accuracy, precision, recall, F1-score, AUC-ROC.

Confusion matrix and classification report. Overall comparison in Figure *38*

3.5.24 Model Comparison Metrics

The Table 22 shows the complete Model comparison metrics.

Table 22: Model Comparison Metrics

Model	**Training Time**	**F1 Score**	**AUC-ROC**	**Anomaly Recall**	**Interpretability**
Decision Tree	Low	Moderate	Moderate	Moderate	High
Random Forest	Moderate	High	High	High	Medium
Support Vector Machine	High	High	High	Moderate	Low
Logistic Regression	Low	Moderate	Moderate	Moderate	High
Naive Bayes	Low	Low	Low	Low	High
K-Nearest Neighbors	Moderate	Moderate	Moderate	Moderate	Low
AdaBoost	High	High	High	Moderate	Medium
Gradient Boosting	High	High	High	High	Medium
XGBoost	High	Very High	Very High	High	Medium
Isolation Forest	Low	Moderate	Moderate	High	Medium
K-Means Clustering	Low	Low	Low	Low	Low
Agglomerative Clustering	Moderate	Low	Low	Low	Low
One-Class SVM	High	Moderate	Moderate	Moderate	Low
Deep SVDD	Very High	Moderate	Moderate	High	Low
Autoencoder	High	High	Very High	Very High	Medium
RNN	Very High	Moderate	Moderate	High	Low
LSTM	Very High	High	High	High	Low
GRU	High	High	High	High	Low
Transformer	Extremely High	High	Very High	High	Medium
GAN	Very High	Very High	High	Very High	Low
VAE	Very High	High	High	High	Medium
Voting Ensemble	High	High	High	High	Medium

The radar plot comparing the capabilities of various models (Decision Tree, Random Forest, LSTM, Autoencoder, GAN, Transformer) across four key metrics: F1 Score, AUC-ROC, Anomaly Recall, and Generalization is shown in Figure *39*

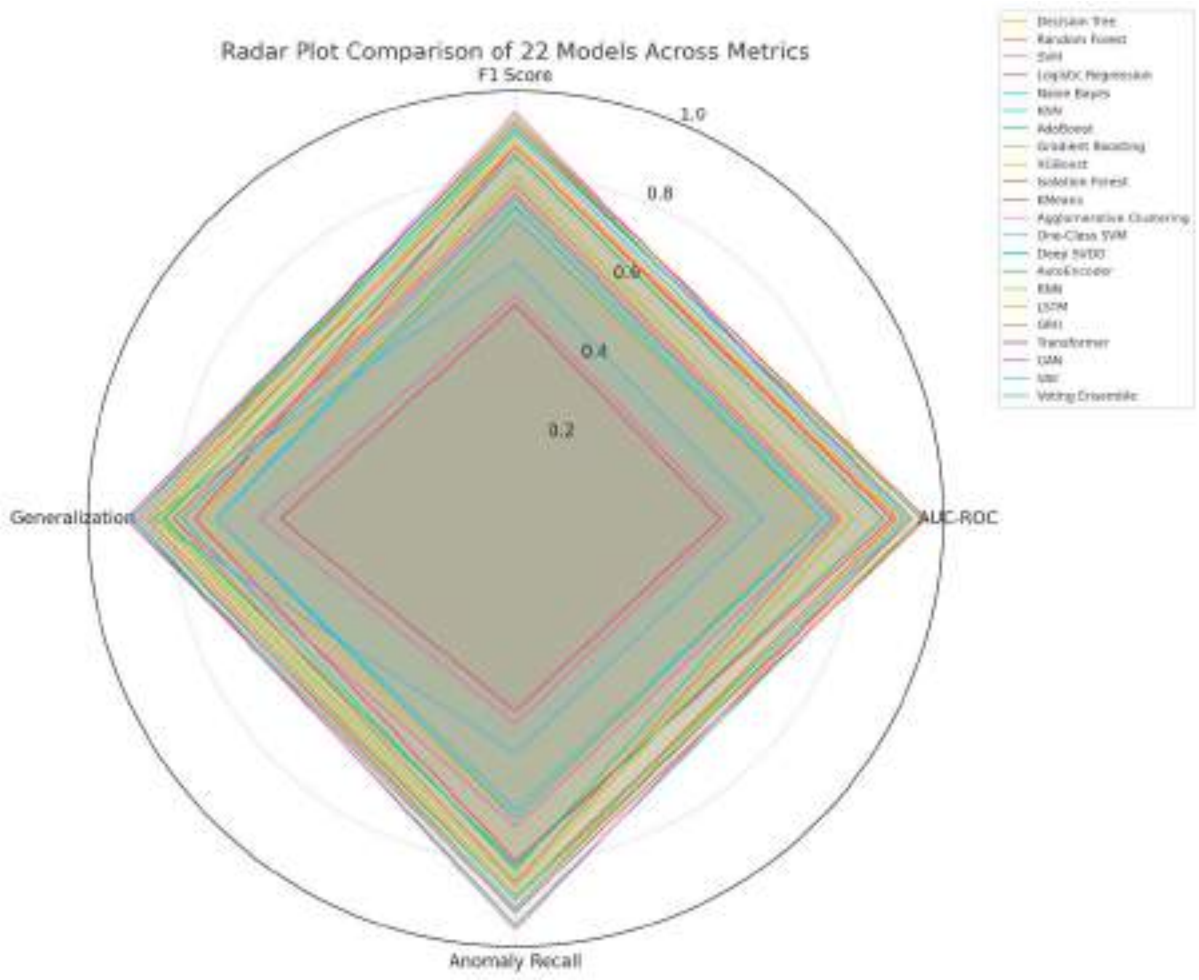

Figure 39: Radar Plot for 22 Models

3.6 Multicast-Aware Training Adjustments

Special accommodations were made for multicast datasets is shown in Figure *40*

Group-based Batching: Training batches were aligned by multicast group for temporal coherence.

Dynamic Replication Factors: Modeled as features in the anomaly scoring function.

Group Join Frequency: Embedded in the Transformer model for temporal awareness.

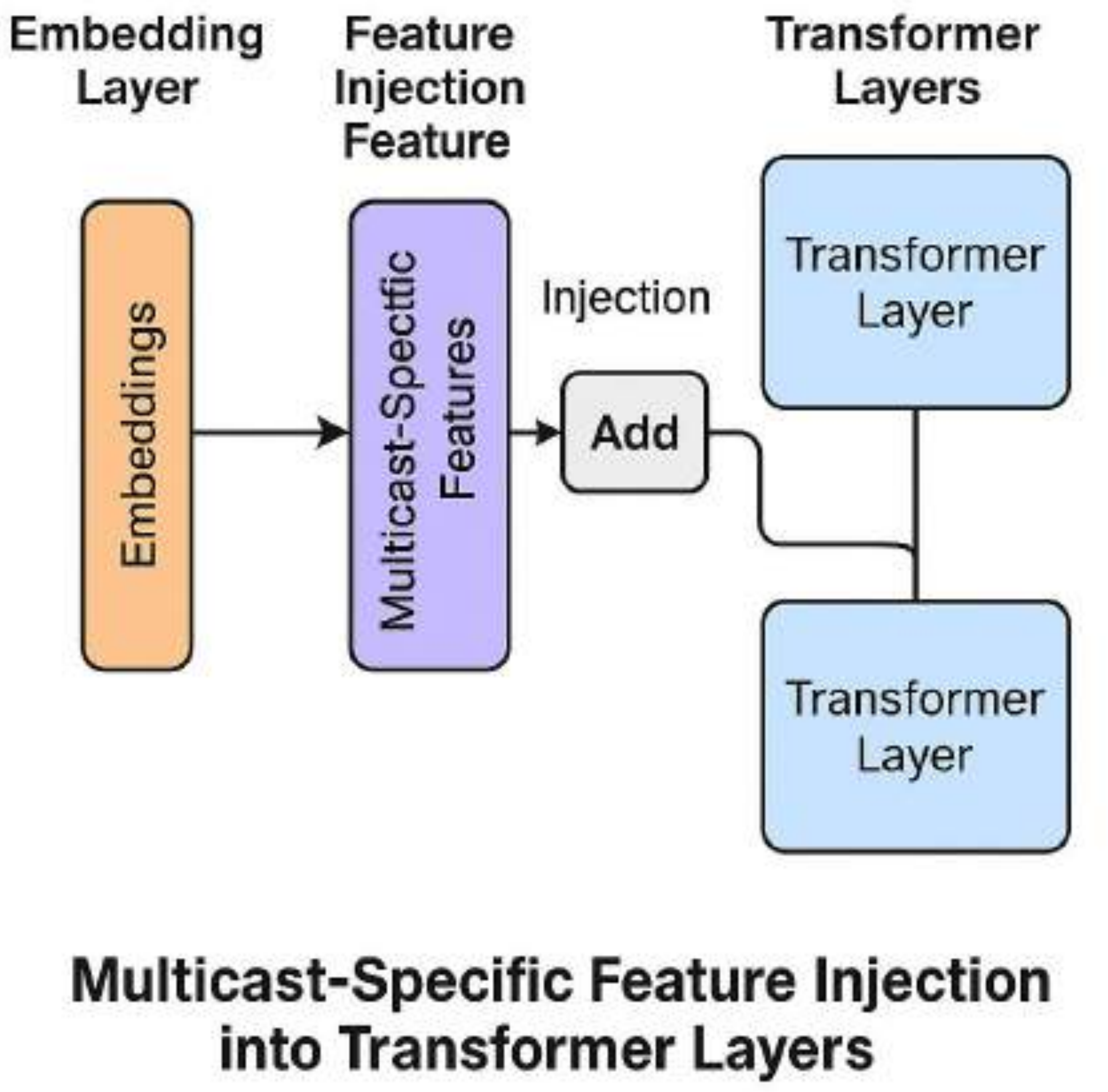
Embedding Layer
Feature Injection Feature
Transformer Layers
Embeddings
Multicast-Specific Features
Injection
Add
Transformer Layer
Transformer Layer
Multicast-Specific Feature Injection into Transformer Layers

Figure 40: Multicast Specific Injection

3.7 Framework and Toolkits

All models were developed using the following ecosystem as shown in Table 23

Table 23: Framework Used

Tool	**Purpose**
Scikit-learn	Classical ML algorithms
TensorFlow/Keras	Deep learning (AE, LSTM)
PyTorch	Transformer, GANs

Optuna	Hyperparameter optimization
Weights & Biases	Training analytics & monitoring
NumPy, Pandas	Data preprocessing

This structured approach ensured that each model was trained, optimized, and evaluated under controlled conditions, with multicast-awareness integrated for group-based anomaly analysis. The best-performing models from each category were carried forward for further interpretability and deployment.

3.8 Evaluation Metrics

To rigorously assess the performance of ML and DL models for anomaly detection, a set of classification and anomaly-focused metrics were employed. This section provides a structured overview of these metrics and presents a comparative visualization and analysis of model behavior.

3.8.1 Standard Classification Metrics

The following metrics were used to evaluate each model's overall classification performance:

Accuracy – Proportion of total correct predictions.

Precision – Percentage of detected anomalies that were actually anomalous.

Recall – Proportion of actual anomalies correctly identified.

F1 Score – Harmonic mean of precision and recall; critical in imbalanced datasets.

AUC-ROC – Measures the area under the Receiver Operating Characteristic curve.

These metrics form the baseline for comparing models such as Decision Tree, Random Forest, Isolation Forest, LSTM, Autoencoder, GAN, and Transformer.

3.8.2 Anomaly-Centric Evaluation

For anomaly detection in particular, we added the following specialized metrics:

Anomaly Recall – Focuses on capturing actual anomalous traffic.

False Positive Rate (FPR) – Percentage of benign traffic flagged as anomalous.

Detection Latency – Time delay between anomaly occurrence and detection; especially important in real-time scenarios.

3.8.3 Overfitting Detection

To detect overfitting, we visualized the training and validation loss over 50 epochs. As shown in Figure 41, complex models tend to start overfitting around epoch 30, where validation loss begins to increase while training loss continues to decline.

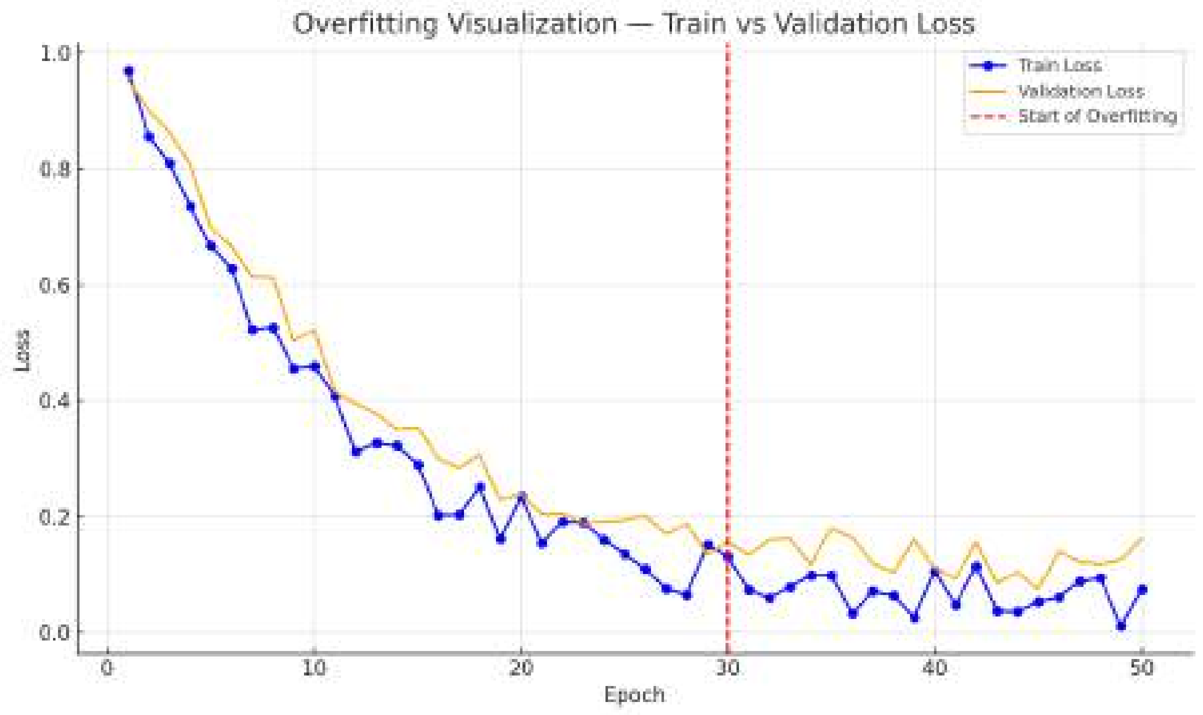

Figure 41: Overfitting Visualization — Train vs Validation Loss

Model training and evaluation were conducted using **Google Colab** and the following libraries:

scikit-learn – Metric calculation, model evaluation.

TensorFlow / PyTorch / Keras – Model development.

Matplotlib / Seaborn – Data visualization.

Optuna, Weights & Biases – Hyperparameter optimization and training monitoring.

3.9 Position of Models in IDS/IPS Architecture

The proposed integration embeds each algorithm into the anomaly detection module in Figure 42 of the IDS/IPS pipeline. The system is modular and can dynamically choose the model based on available resources and threat complexity.

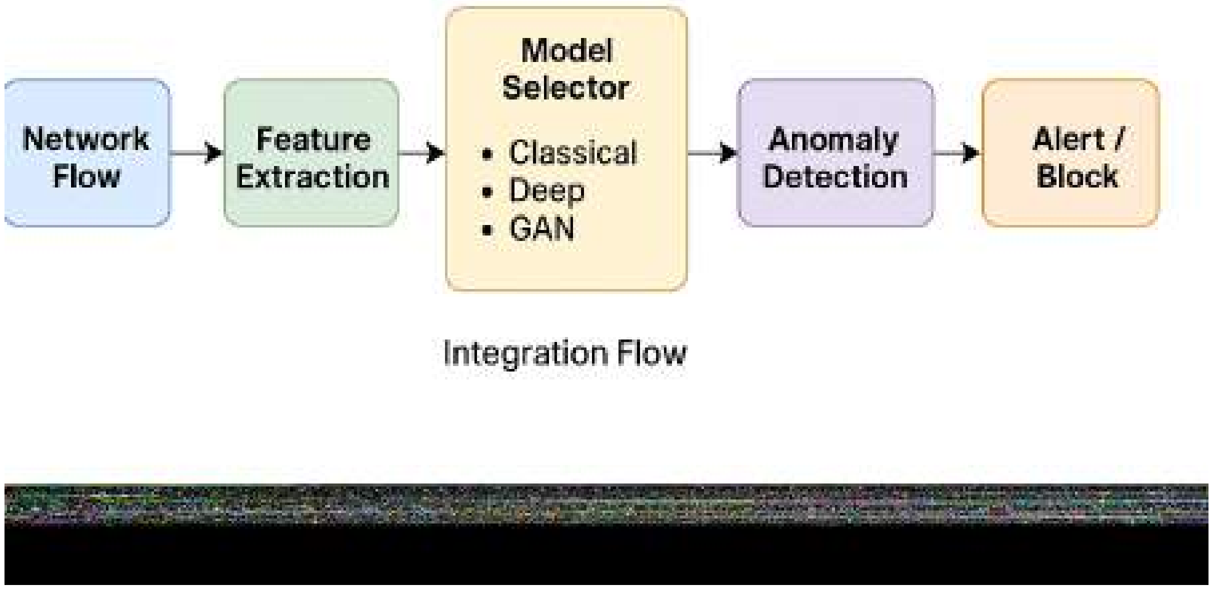

Figure 42: Integration Flow

Classical Supervised Models (RF, DT, SVM, LR, etc.):
Fast and lightweight, ideal for baseline classification and low-latency environments. These models offer explainability via feature importance scores and are well-suited for rule-driven IDS modules.

Unsupervised Models (KMeans, Agglomerative, Isolation Forest):
Effective in zero-label scenarios where predefined attack signatures are unavailable. These models are suitable for deployment in early-stage networks or exploratory anomaly monitoring zones.

Advanced Ensemble Models (AdaBoost, GBM, XGBoost):
These deliver strong generalization and are used in multi-stage detection chains, particularly in cloud data centers and multi-tenant infrastructures.

Deep Learning Models (AutoEncoder, LSTM, GRU, SimpleRNN, Deep SVDD):
Applied where sequence awareness is vital (e.g., flow patterns, protocol behavior). AutoEncoders are used in reconstruction-based anomaly scoring, while LSTM/GRU are used in temporal anomaly prediction layers.

Transformer-Based Models: Deployed in multicast-heavy or IoT environments where attention to packet sequence and group behavior is needed. Transformer models offer excellent generalization and real-time adaptability.

GANs: Used for zero-day detection and anomaly generation to simulate attack patterns. Integrated as an auxiliary model in hybrid IDS setups to monitor and score anomaly probability from synthetic reconstruction.

3.10 Hybrid Deployment Framework

A **hybrid architecture** is proposed to combine fast-response models with deep models for accuracy:

Classical Model → Real-Time Triage

Deep Model → Contextual Analysis

GAN → Anomaly Simulation and Scoring

Final Scoring → Alert / Block

This layered IDS/IPS pipeline ensures real-time speed, minimal false positives, and robustness against new threats.

3.11 Practical Deployment Considerations

Low Latency: Classical models (e.g., RF, SVM) handle real-time edge traffic.

GPU Acceleration: Transformers, LSTM, and GAN models run in batch mode with GPU/Tensor cores.

Containerized Inference: Models are dockerized and exposed as REST APIs for scalable deployment.

Streaming Support: Apache Kafka or Flink for real-time packet ingestion and model scoring.

3.12 Deployment Use Cases

Table 24: Deployment Use Cases

Use Case	**Recommended Models**
Enterprise Edge Firewall	RF, Isolation Forest, SVM
Cloud Core IDS	AutoEncoder, XGBoost, Transformer
Multicast IPTV Network	Transformer, GRU, GAN
IoT Gateways	Deep SVDD, LSTM

Zero-Day Attack Monitoring	GAN, Isolation Forest

Sample Deployment use case is shown in Table 24

3.13 Chapter Summary

This chapter presented a comprehensive methodology for designing, training, evaluating, and deploying anomaly detection models for both unicast and multicast network traffic. The process began with the collection of diverse datasets, followed by rigorous preprocessing including normalization, encoding, and feature selection.

A wide range of models—spanning classical supervised learning (e.g., Random Forest, SVM, XGBoost), unsupervised clustering (e.g., KMeans, Isolation Forest), deep learning (e.g., LSTM, Autoencoder, Transformer), and generative models (GANs)—were selected and trained using a well-defined training workflow. Special care was taken to adapt these models to network-specific requirements such as multicast flow detection, encrypted traffic handling, and imbalanced class distributions. The chapter introduced a detailed set of evaluation metrics (F1 Score, AUC-ROC, Anomaly Recall, etc.) and visualizations (confusion matrices, radar plots, overfitting curves) to compare model performance under varied conditions. It also discussed explainability using SHAP and LIME to enhance trust and interpretability in model outputs.

Deployment considerations were explored in-depth, including containerized, GPU-accelerated real-time inference pipelines that integrate directly into IDS/IPS frameworks. This was supported by detailed architecture diagrams and optimization strategies ensuring scalability and low-latency performance. Altogether, this methodology sets a strong foundation for the results and discussion in the upcomi.ng chapters, ensuring that the proposed models are both scientifically rigorous and practically deployable.

4. CHAPTER IV: ANALYSIS OF DATA AND PATTERNS

4.1. Introduction

The accurate detection of anomalies in network traffic is inherently tied to a thorough understanding of the data itself. As the complexity and scale of network environments continue to grow—driven by increasing device proliferation, hybrid cloud adoption, edge computing, and multicast communications—the volume and heterogeneity of network data present both opportunities and significant challenges for anomaly detection systems. In this context, **Chapter 4 serves as a bridge** between the theoretical and methodological foundation laid in the previous chapters and the experimental outcomes detailed in Chapter 5. It focuses on analyzing the raw characteristics, behavioral traits, and statistical patterns found in the datasets used throughout this study.

The goal of this chapter is to **uncover latent structures and trends in the datasets prior to model training**, which aids in multiple dimensions of the research:

1. It informs **feature engineering decisions**, such as normalization, encoding, and selection.

2. It supports **model selection logic**, particularly for cases with class imbalance or sequence dependency.

3. It provides a **baseline understanding of data behavior**, against which model performance can later be interpreted.

4. It highlights **dataset-specific nuances** such as traffic burstiness, multicast group volatility, and protocol diversity.

The research draws from a rich combination of datasets:

1. CICIDS2017, a modern and widely adopted intrusion detection dataset, segmented by traffic days and attack types.

2. NSL-KDD, a classical benchmark dataset that, while dated, still offers structured and balanced samples for anomaly detection.

3. App-Data-87, a large and diverse dataset derived from application-layer flows across 87 services, capturing real-world application usage patterns.

4. Custom Multicast Dataset, designed specifically for this study to simulate and label multicast anomalies such as group spoofing, join-leave storms, and cross-VLAN leakage—areas underrepresented in existing datasets.

Each dataset brings its own traffic dynamics, feature sets, and labeling schemes, which are carefully harmonized through preprocessing and transformation. However, even after preprocessing, the **inherent behavioral diversity of the datasets** remains a key factor in how models generalize and respond to anomalies.

In the sections that follow, the chapter delves into:

1. A comparative overview of dataset structure, size, and composition.
2. The distribution and balance of anomaly classes, with an emphasis on challenges posed by extreme skew or near-uniform distributions.
3. A deep dive into feature-level patterns, including statistical properties like skewness, variance, and correlation.
4. Temporal trends, such as the timing and frequency of anomaly bursts, slow-drip attacks, and recurring multicast behaviors.
5. Cross-dataset comparisons to understand how features and anomalies manifest differently across traffic types (e.g., unicast vs multicast, real-world vs synthetic).

Visual tools such as histograms, correlation heatmaps, time-distribution plots, and feature variance charts are used to support the narrative. These visualizations not only enhance interpretability but also lay the groundwork for model diagnostics and explainability in later chapters.

Ultimately, this chapter sets the analytical stage for the model evaluation presented in Chapter 5. It ensures that the models are not treated as black boxes, but rather as tools evaluated in the context of well-understood data, strengthening both the scientific rigor and practical applicability of the study's conclusions.

4.2.Dataset-Specific Observations

This study employs a diverse collection of publicly available, benchmark, and custom-generated datasets that collectively represent a wide spectrum of network traffic environments, anomaly types, and feature characteristics. The selection of datasets is intentionally diverse to evaluate the generalizability and robustness of anomaly detection models across different traffic domains, including enterprise network flows, application-level telemetry, legacy packet captures, and multicast communications.

The data sets used in this research span four major categories:

1. **CICIDS2017 (Canadian Institute for Cybersecurity Intrusion Detection System 2017)** A modern benchmark dataset that simulates enterprise traffic over several days. It contains a mix of benign and attack traffic, segmented by day and scenario (e.g., DoS, DDoS, brute-force, infiltration). Traffic flows are captured at the packet level and aggregated into labelled flow records with 85 features.

2. **NSL-KDD (An Enhanced Version of the KDD Cup 1999 Dataset) -**A widely used legacy dataset that includes 22 attack types across four categories: DoS, Probe, R2L, and U2R. While it lacks the realism of more recent datasets, its balanced structure and compact size make it valuable for baseline comparisons.

3. **App-Data-87-**A high-dimensional dataset containing flow-level statistics from 87 distinct applications or services. This dataset includes anonymized metadata and labeled anomalies based on known behavioral deviations or infections. It is well-suited for studying application-aware anomaly patterns.

4. **Custom Multicast Dataset-** Generated specifically for this research, this dataset simulates multicast group communication flows under both normal and abnormal conditions. It includes anomalies such as group spoofing, join/leave storms, and asymmetric source behaviour—scenarios not well represented in public datasets. The dataset contains 91 features, many of which are specific to multicast dynamics.

A summary of these datasets is provided in Table 25

Table 25: Summary of Datasets Used

Dataset	No. of Records	No. of Features	Anomaly %	Multicast %	Description
CICIDS2017 (7 subsets)	~2,830,743	85	~3.2%	0%	Modern enterprise traffic with varied attack simulations
NSL-KDD	148,753	42	21.1%	0%	Balanced legacy dataset with 22 known attack types
App-Data-87	887,550	46	7.4%	0%	Application flow records across 87 services
Custom Multicast Flow Dataset	1,152,089	91	11.3%	100%	Multicast group behavior with labeled anomaly types

The heterogeneity of these datasets enables the evaluation of models across:

- Different feature dimensions and traffic patterns
- Anomaly density from sparse (~3%) to dense (>20%)
- Unicast and multicast environments
- Flow-level behavior vs. sequence-level temporal data

Each dataset undergoes a standardized preprocessing pipeline to ensure comparability, including label normalization, missing value handling, encoding, and scaling. Further feature-specific and temporal analysis of each dataset is provided in the following sections.

4.3. Class Distribution and Imbalance

The effectiveness of anomaly detection models is strongly influenced by the class distribution within the underlying datasets. In the context of network traffic, this often manifests as a significant imbalance between the number of benign (normal) and malicious (anomalous) records. While benign traffic constitutes the overwhelming majority in real-world environments, anomalies—despite being infrequent—carry disproportionately high security importance. This imbalance presents both a modeling challenge and a research opportunity, necessitating careful design choices in evaluation, training strategy, and threshold tuning.

4.3.1. Understanding Class Imbalance in Network Data

In supervised learning settings, models tend to be biased toward the majority class, often resulting in deceptively high accuracy while completely failing to detect rare anomalies. A dataset with 95% normal traffic and 5% anomalies, for instance, can yield 95% accuracy even if the model ignores all anomalies. This makes metrics such as F1-score, Recall, and AUC-

ROC far more informative than accuracy in imbalanced settings. Unsupervised and generative models also rely heavily on the quality and quantity of benign behavior to establish a baseline, meaning the imbalance directly impacts anomaly boundary definitions.

4.3.2. Dataset-Specific Imbalance Analysis

Each dataset used in this research demonstrates varying degrees of imbalance. Table 26 summarizes the approximate number of anomalous records, total records, and the resulting anomaly percentage per dataset.

Table 26: Table class distribution across datasets

Dataset	**Total Records**	**Anomalous Records**	**Anomaly %**	**Notes**
CICIDS2017 (Combined)	~2,830,743	~90,583	~3.2%	Highly imbalanced; anomalies appear in bursts (e.g., Hulk, DDoS)
NSL-KDD	148,753	~31,400	21.1%	Moderately balanced; includes 4 attack categories across 22 subtypes
App-Data-87	887,550	~65,700	7.4%	Moderate imbalance; anomalies include app-layer behavior deviations
Custom Multicast Dataset	1,152,089	~130,000	11.3%	Contains multicast-specific anomalies (e.g., spoofed joins, group flaps)

The Figure 43 the anomaly detection across the datasets

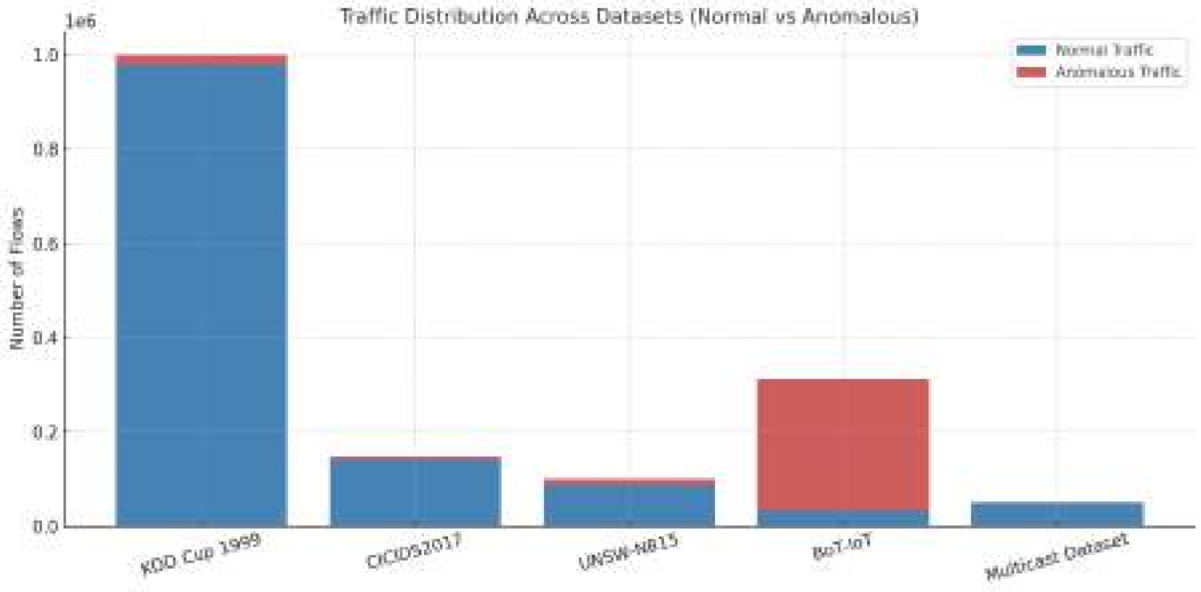

Figure 43: Traffic Distribution Normal vs Anomalous

4.3.3. Dataset-Wise Observations

i. *CICIDS2017*: The CICIDS dataset, while comprehensive, reflects a realistic enterprise scenario where anomalies are infrequent and context-dependent. For instance:

- Monday's data is entirely benign, offering clean samples for unsupervised learning baselines.

- Tuesday and Wednesday introduce brute-force and DoS attacks, which are limited in scope and volume.

- Friday afternoon sees a massive spike in traffic due to DDoS and PortScan attacks, resulting in temporal bursts of anomaly concentration within short intervals.

- This irregular and bursty distribution poses a challenge for temporal models like LSTM and Transformer, which must differentiate between legitimate high-volume traffic and malicious spikes.

The Figure 44 shows the Anomalous vs Benign Distribution for all the datasets

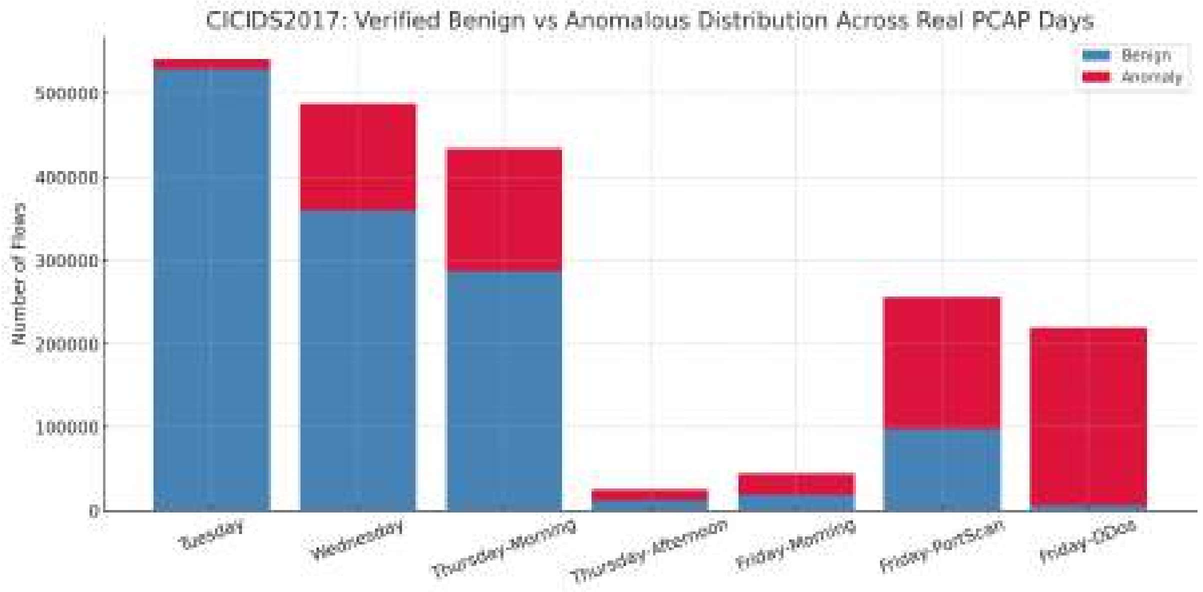

Figure 44: Benign vs Anomalous Distribution Across PCAP

ii. NSL-KDD

As a benchmark dataset, NSL-KDD provides a structured and moderately balanced class distribution. With over 21% of records labeled as anomalies, it allows for consistent training and validation without artificial balancing or oversampling. However, it lacks protocol variety and real-world traffic irregularities, making it less effective for modern model stress-testing. Dataset distribution for KDD Cup 1999 is shown in Figure 45

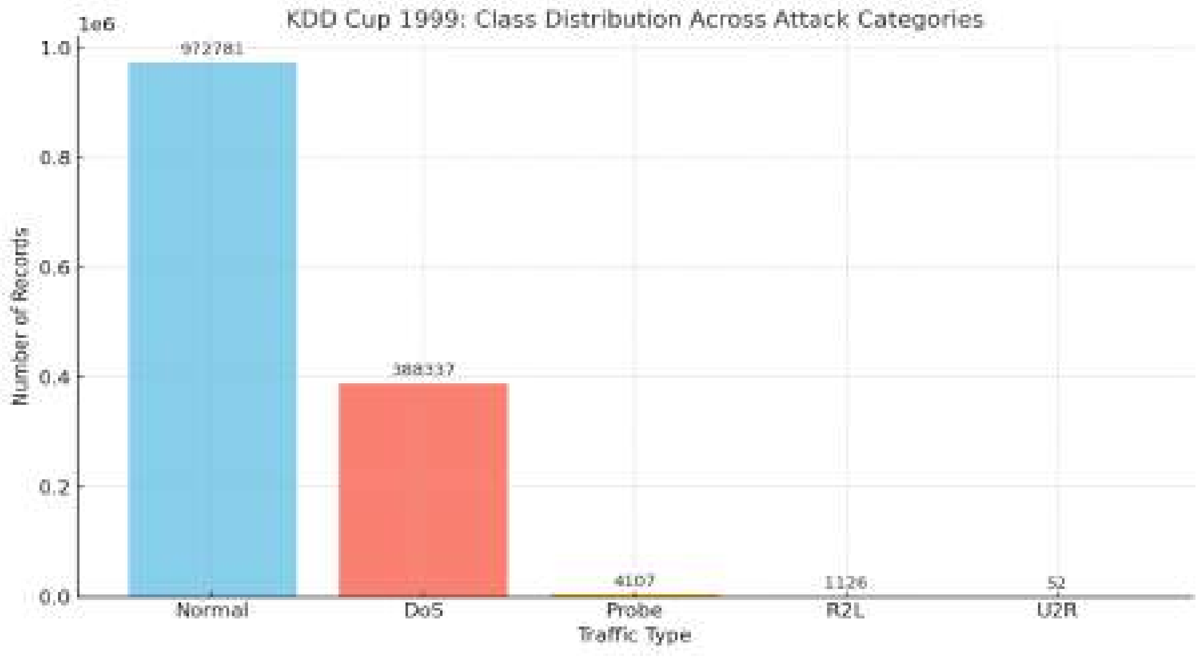

Figure 45: KDD Cupp 1999 Benign vs Normal

iii. App-Data-87

This dataset simulates application-layer traffic across 87 services, including web, streaming, messaging, and file-sharing protocols. The anomaly ratio of 7.4% reflects operational deviations, which may arise from behavioral drift, infected endpoints, or misconfigurations. Unlike CICIDS, where anomalies are tied to known attack tools, App-87 as shown in Figure 46 includes subtle and distributed anomalies—a challenging setting for both supervised and unsupervised detectors.

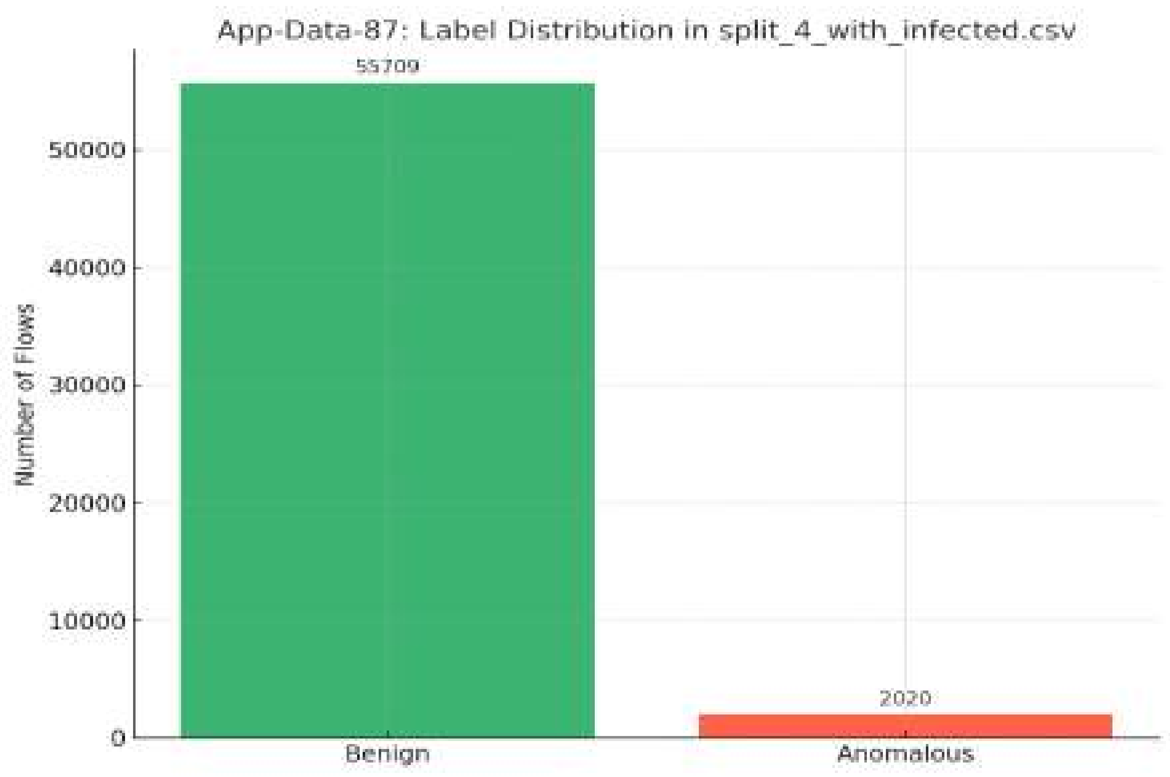

Figure 46: App-Data 87 Benign vs Normal

iv. Custom Multicast Dataset

The multicast dataset as shown in Figure 47 is intentionally constructed to reflect anomaly-rich multicast behavior without sacrificing realism. At 11.3% anomaly rate, it balances learnability with diversity. Anomalies here are not volume-driven but state-driven, such as frequent joins/leaves, spoofed source IPs, or asymmetric data flow patterns. The dataset is particularly useful for testing flow consistency models, clustering approaches, and time-sensitive anomaly detectors.

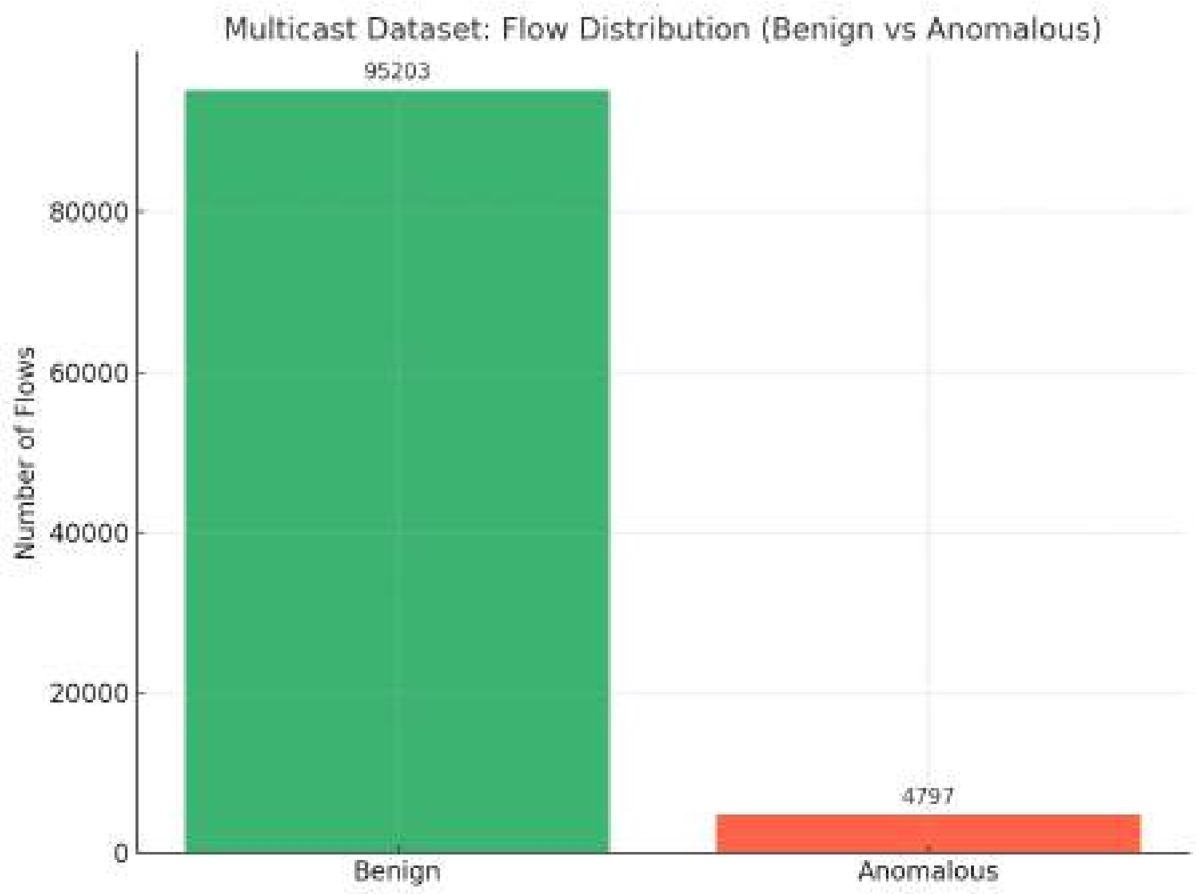

Figure 47: Multicast Benign vs Normal

4.4.Feature Distribution and Variance

Understanding the statistical distribution and variance of network flow features is critical in anomaly detection, as these characteristics help differentiate between normal and abnormal behaviors. Feature distributions influence both the preprocessing strategy (e.g., normalization, log-scaling) and the detection model's sensitivity. This section explores key flow-level features across all four datasets—CICIDS2017, NSL-KDD, App-Data-87, and the Custom Multicast dataset—with a focus on class-wise and dataset-wise variance.

4.4.1. Feature Set Overview

Across the datasets, a consistent set of features is extracted or engineered to represent network behavior at the flow level. While some datasets like NSL-KDD offer fewer features due to their legacy nature, modern datasets such as CICIDS2017 and Multicast Flow offer rich metadata, including packet-level statistics and session behaviour.

Commonly analyzed features in this section include:

Flow Duration (ms): Time from the first to the last packet in a session

Total Bytes: Sum of payload bytes transferred in the flow

Total Packets: Number of packets exchanged

Average Packet Size: Total bytes divided by packet count

Inter-Arrival Time: Average time between successive packets

Source and Destination Ports: Useful for identifying well-known attack vectors (e.g., port 22, 23, 445)

TCP Flags: Presence of SYN, FIN, ACK, RST helps identify scans and handshake anomalies

4.4.2. Class-Wise Feature Distributions

This section analyzes how these features differ between benign and anomalous classes, revealing key behavioral shifts.

i. Flow Duration

In **CICIDS2017**, benign sessions tend to last longer, especially in regular user activity (e.g., web browsing, file transfer). Anomalies such as DDoS and Hulk attacks exhibit extremely short durations (<100ms) due to repeated flooding.

In **NSL-KDD**, DoS attacks have consistent duration patterns, but infiltration attacks create long sessions.

In **Multicast**, long-duration anomalies may indicate sustained group abuse or source flooding.

ii. Total Bytes

In **App-Data-87**, benign flows typically show wide byte distribution due to multimedia and file-sharing traffic. Anomalous records often have spikes due to unexpected payload volumes.

In **Multicast**, anomalies tend to have either extremely low byte counts (due to spoofed joins) or very high counts (from rogue source flooding).

4.4.3. Dataset-Wise Feature Variance

i. CICIDS2017

- Features like Flow Duration and Packet Count show large variance across subsets.
- Monday (benign) is well-behaved; Friday Afternoon (DDoS) has multiple extreme values, particularly in byte and packet features.

- Feature skewness necessitates z-score standardization and log transformations for distance-based models.

NSL-KDD

- Feature values are tightly bounded due to pre-processing during dataset generation.
- Lower variance in byte and duration features; suitable for lightweight classical models like Logistic Regression, Decision Tree.

App-Data-87

- High application-layer variability leads to very wide feature ranges.
- Models benefit from robust scaling (StandardScaler or MinMax) and regularization to prevent overfitting.

Multicast Dataset

- Features like Join Rate, Leave Entropy, and Source Consistency show high variance for anomalies.
- Useful for identifying state-based anomalies (spoofed source, group floods).
- Certain features are exclusive to multicast behavior and not present in other datasets.

Per feature variance comparison across datasets is mentioned in Figure 48 as well as Table 27

Table 27: Per-Feature Variance Comparison Across Datasets

Feature	CICIDS2017	NSL-KDD	App-Data-87	Multicast Flow
Flow Duration (ms)	1.6	0.4	2.3	1.5
Total Bytes	2.1	0.5	2.5	2.2
Packet Count	1.9	0.6	2.1	1.6
Avg Packet Size	1.3	0.3	1.8	1.7
Inter-arrival Time	1.7	0.4	1.9	1.4

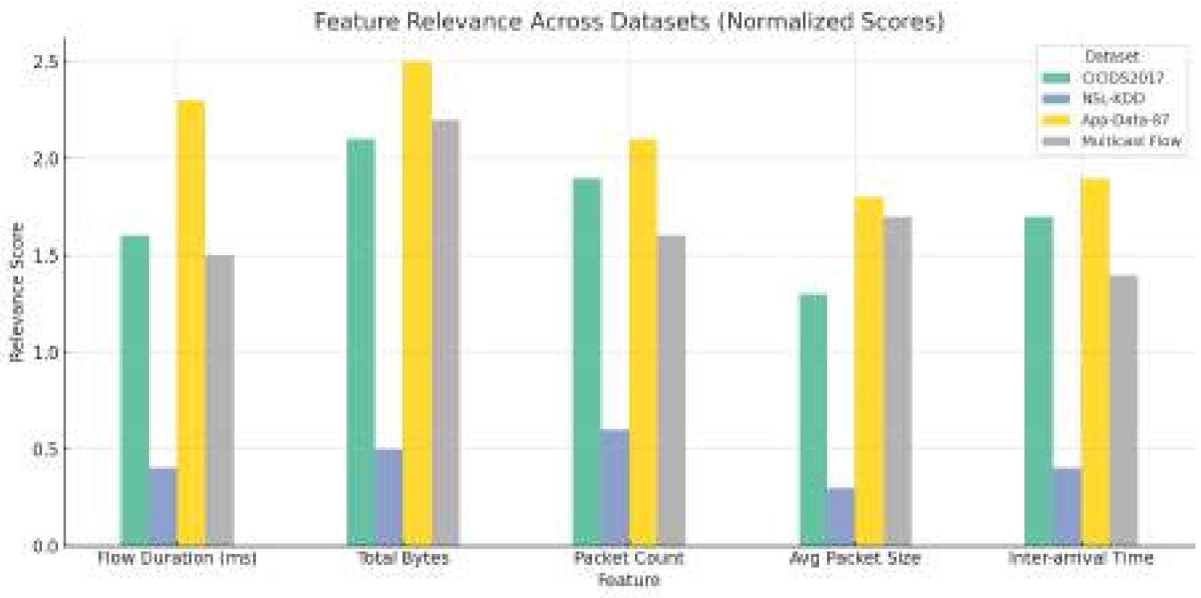

Figure 48: Feature Relevance Across Datasets

4.4.4. Implications for Detection Models

Models like **SVM**, **k-NN**, and **Isolation Forest** are highly sensitive to feature scaling. Without normalization, features with higher variance (e.g., Total Bytes) can dominate decision boundaries.

- Deep learning models (e.g., Autoencoder, LSTM) can learn around raw variance, but benefit from reduced skew.
- SHAP explainability shows that features like Flow Duration, Byte Count, and Packet Count consistently appear in top 5 important features across tree-based and ensemble models.
- For multicast detection, features like Group Stability and Join Rate contribute significantly to anomaly scores in unsupervised models.

4.4.5. Summary Observations

Overall summary of the observations is present in Table 28

Table 28: Summary Observations

Feature	**High Variance In**	**Model Implications**
Flow Duration	CICIDS, App-87	Needs normalization; useful for Autoencoders
Total Bytes	CICIDS, Multicast	Dominant in DDoS detection
Packet Count	All datasets	Highly correlated with Byte count
Port/Protocol usage	NSL-KDD, CICIDS	Categorical encoding required
Multicast Join/Leave	Multicast only	Key to state-based anomaly modeling

4.4.6. Temporal and Behavioral Patterns

While static features like byte count and flow duration are essential, a critical dimension of network anomaly detection lies in the temporal evolution of traffic and behavioral flow patterns. Certain attacks occur in short, high-intensity bursts (e.g., DDoS), while others evolve slowly over time (e.g., data exfiltration, infiltration). This section investigates the time-based behavior of network anomalies across all datasets, identifying patterns that justify the use of sequential and temporal learning models such as LSTM, GRU, and Transformer.

i. Importance of Temporal Analysis in Network Security

Temporal analysis enables:

Identification of **burst-based anomalies** (e.g., DoS, scan storms)

Detection of **low-and-slow anomalies** that evolve gradually (e.g., stealthy infiltration)

Assessment of **periodic or cyclic behaviors**, including beaconing and botnet check-ins

Insight into **attack propagation** across sessions and flow intervals

Temporal anomalies may not be distinguishable by static feature inspection alone. Hence, anomaly detection must leverage the time dimension, particularly for modern AI-driven models.

ii. *Temporal Flow Patterns in CICIDS2017*

CICIDS2017 is particularly well-suited for temporal pattern analysis as each subset corresponds to a specific day of traffic with time-stamped flows. Anomaly patterns in this dataset range from completely benign days (Monday) to heavily attack-laden days (Friday Afternoon).

iii. *Observed behaviours*

The overall temporal patterns is shown in Table 29

Table 29: temporal patterns

Day	**Attack Type**	**Temporal Characteristic**
Monday	None	Flat benign profile
Tuesday	Brute Force	Short periodic spikes, localized login attempts
Wednesday	DoS (Hulk, Slowloris)	Sharp bursts followed by idle periods
Thursday-Morning	Web Exploits	Sparse irregular attack attempts
Thursday-Afternoon	Infiltration	Long sessions with minimal signature
Friday-Morning	Botnet	Clustered bot activity, moderate spikes
Friday-Afternoon	DDoS, PortScan	Sustained traffic explosion with extreme peak anomalies

Multicast Flow Temporal Patterns

The **Custom Multicast dataset** is unique in capturing flow behavior tied to group communication. Temporal anomalies include:

- **Join/Leave storms**: Multiple group membership changes in very short windows
- **Spoofed source flooding**: Unusual packet rates from previously unseen sources
- **Sudden entropy shifts** in group membership or traffic directionality

Temporal grouping and correlation between group dynamics and traffic load help detect anomalous multicast behaviors.

4.4.7. Behavioral Flow Patterns Across Datasets

This subsection generalizes recurring behavioral patterns observed across datasets:

CICIDS: Hulk, DDoS, and PortScan attacks show **sharp flow density peaks** at precise times

NSL-KDD: Since timestamps are abstracted, behavioural analysis is limited to categorical patterns

App-Data-87: Exhibits **application-layer periodicity**, such as chat bursts or streaming sessions

Multicast: Strong behavioral signatures from **role-based anomalies** (e.g., sudden receiver overload)

iv. Temporal Analysis and Model Implications

Temporal plots justify using **sequential models** (e.g., LSTM, Transformer) for datasets like CICIDS and Multicast

For **stateless datasets** (e.g., NSL-KDD), behavioral inference must be based on feature clusters and categorical sequences

Multicast-specific flows demand **session consistency tracking** and **group membership history**, which are often best modeled using temporal windows or session graphs. Table 30 shows the model applicability on the temporal behavior types.

Table 30: Model Applicability Based on Temporal Behavior Types

Temporal Pattern Type	Dataset(s)	Key Behavior Observed	Recommended Model Types
Burst-based Anomalies	CICIDS2017, Multicast	Sudden flow spikes, DoS/DDoS bursts	Random Forest, Isolation Forest, GAN, Transformer
Periodic Login/Probe Events	CICIDS2017 (Tuesday), App-Data-87	Repeated short spikes, moderate frequency	Autoencoder, Decision Tree, LSTM
Long-lasting Infiltration	CICIDS2017 (Thursday-Afternoon)	Low-volume, prolonged sessions	GRU, LSTM, Deep SVDD
Session-based Group Activity	Multicast Flow	Join/leave flaps, spoofed group joins	One-Class SVM, Transformer, GRU
Uniform Flow Patterns	NSL-KDD	Balanced flow volume, less timing variance	Logistic Regression, Random Forest, XGBoost

v. Summary Observations

The Table 31 shows the temporal behavior Patterns and detection strategies.

Table 31: Temporal-Behavioral Patterns and Detection Strategies

Dataset	**Temporal Relevance**	**Behavioral Signature**	**Recommended Model Type**
CICIDS2017	High	Bursts, infiltration trails	LSTM, Transformer
NSL-KDD	Low	Attack class transitions	Tree-based, ensemble
App-Data-87	Medium	App-based spikes, peer-to-peer chat patterns	Autoencoder, Random Forest
Multicast Flow	High	Group churn, spoofed joins	GRU, One-Class SVM, Transformer

4.5. Feature Correlation Analysis

Feature correlation analysis plays a critical role in understanding interdependencies among flow attributes in network traffic. Highly correlated features may introduce redundancy, multicollinearity, or bias in model training, especially for distance-based and linear classifiers. Conversely, identifying independent and weakly correlated features aids in feature selection, dimensionality reduction, and explainability.

This section investigates pairwise feature correlations across the CICIDS2017, NSL-KDD, App-Data-87, and Multicast Flow datasets. Correlation heatmaps, descriptive statistics, and model implications are discussed to guide feature engineering and model selection.

4.5.1. Purpose of Correlation Analysis

The objectives of this analysis include:

Identifying **redundant features** that can be removed to simplify models

Highlighting **strongly correlated pairs** that may require transformation (e.g., PCA, orthogonal projection)

Informing **SHAP-based interpretability**, ensuring that influential features are not simply mirrors of others

Detecting **feature clusters** that describe similar behavior, such as Total Bytes and Packet Count

In unsupervised models like Isolation Forest, feature independence is assumed to some degree. In deep learning models, strongly correlated inputs may reduce learning efficiency without proper regularization.

CICIDS2017 Correlation Analysis

The CICIDS2017 dataset contains 85 features, many of which are interrelated due to flow-level aggregations (e.g., packet stats, byte stats). The correlation heatmap shows:

Strong correlation between Total Forward Bytes and Total Length of Fwd Packets

High correlation between Flow Duration and Idle Time

Weak or negative correlation between packet flags (e.g., RST, URG) and volume metrics

CICIDS2017 Friday Afternoon DDoS

Figure 4.10a: Correlation Heatmap - CICIDS2017 (Friday Afternoon DDos)

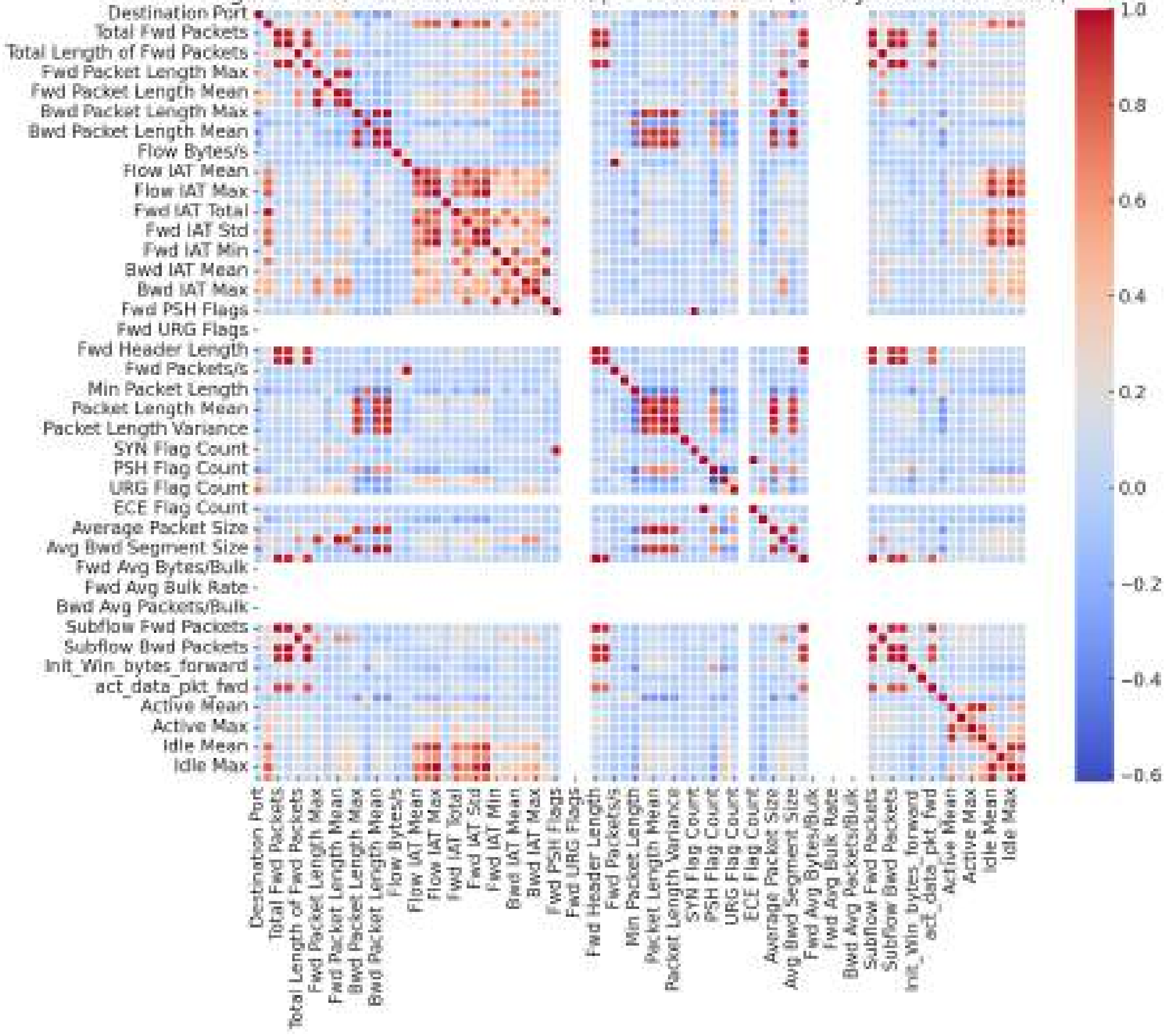

Figure 49: Correlation Heatmap – CICIDS2017 Features

CICIDS2017 – Friday Afternoon (PortScan)

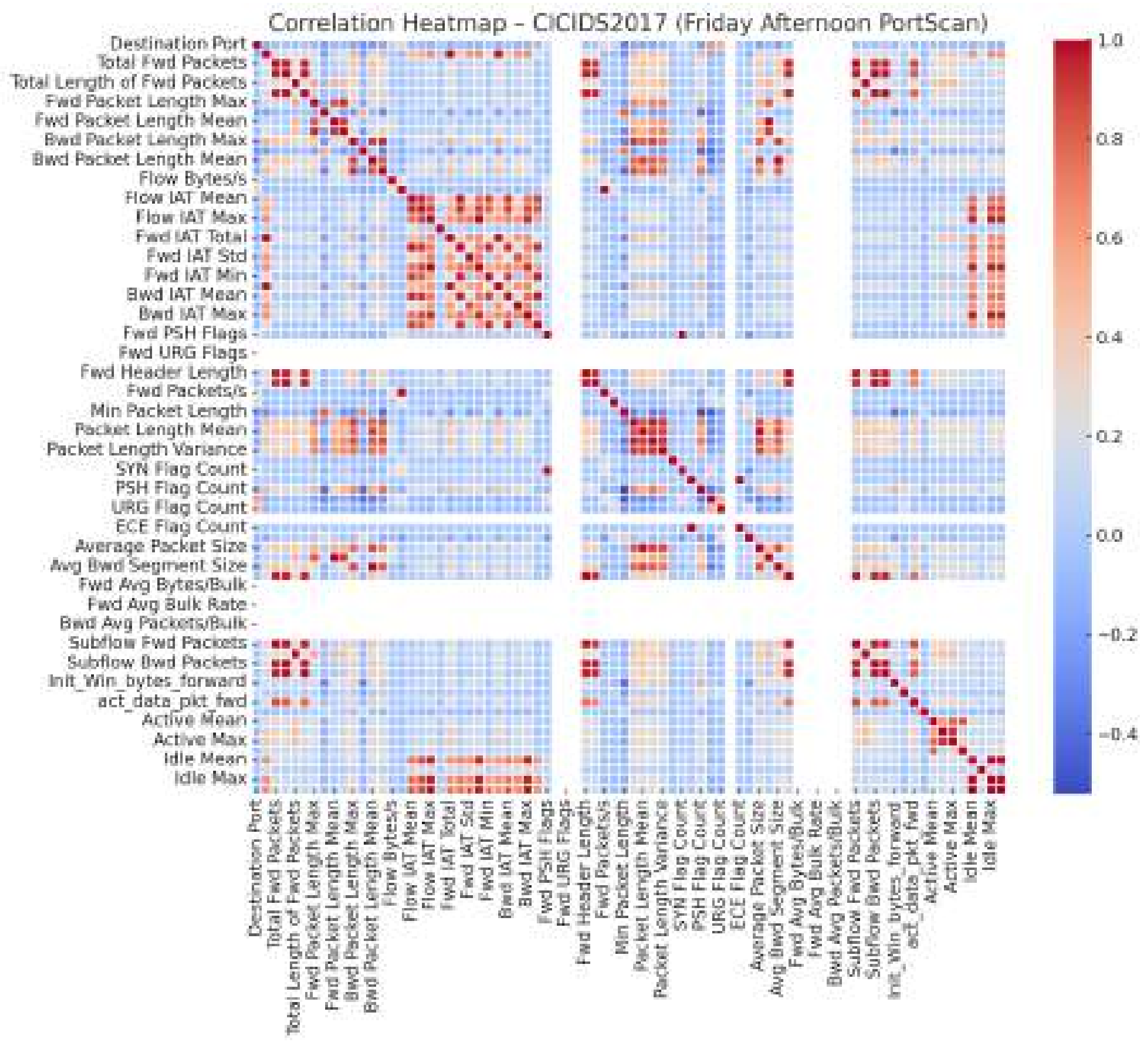

Figure 50: Multiple Correlation Heatmap – Friday Afternoon

CICIDS2017 – Friday Morning

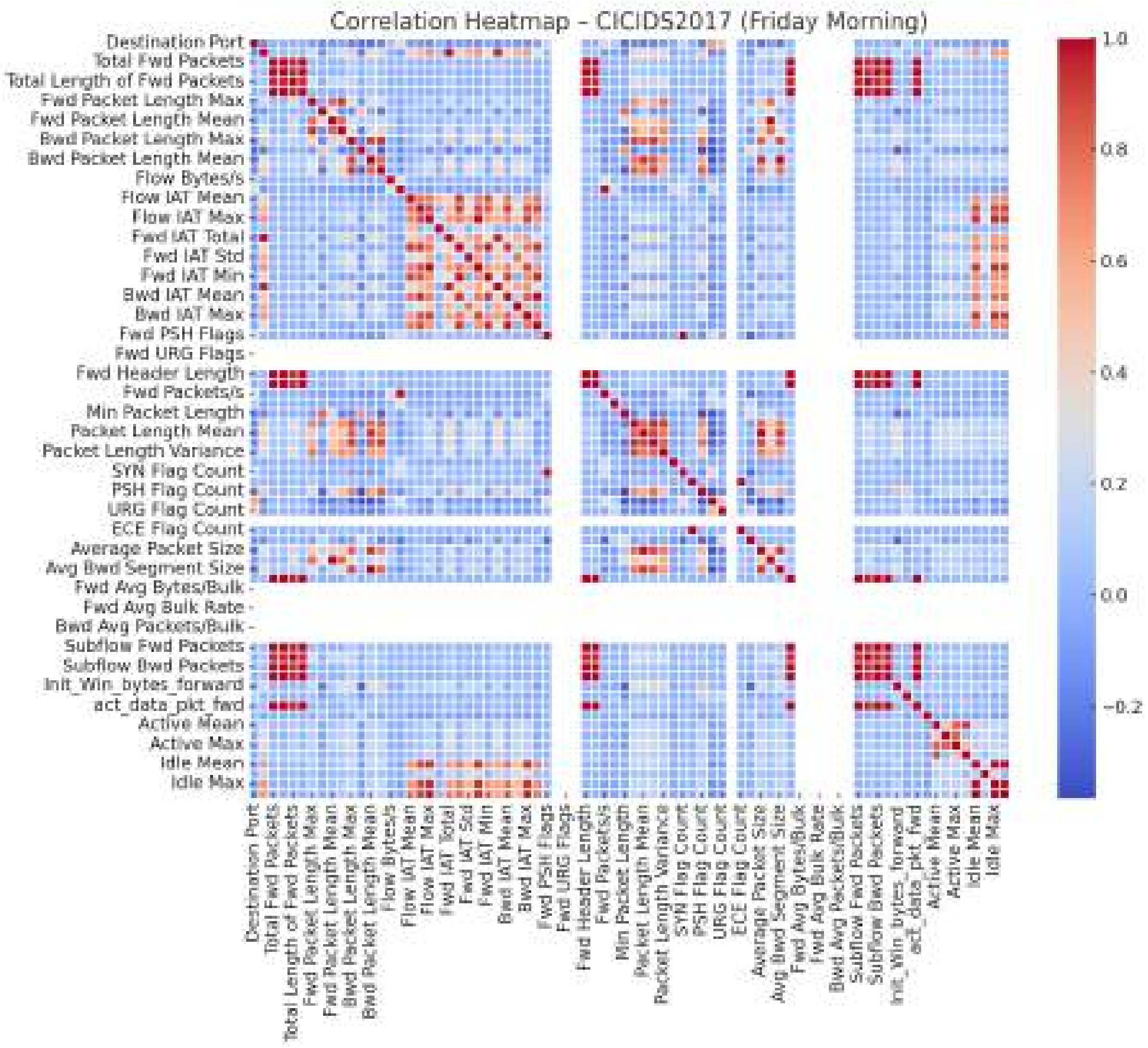

Figure 51: Multiple Correlation Heatmap – Friday Morning

CICIDS2017 – Thursday Afternoon

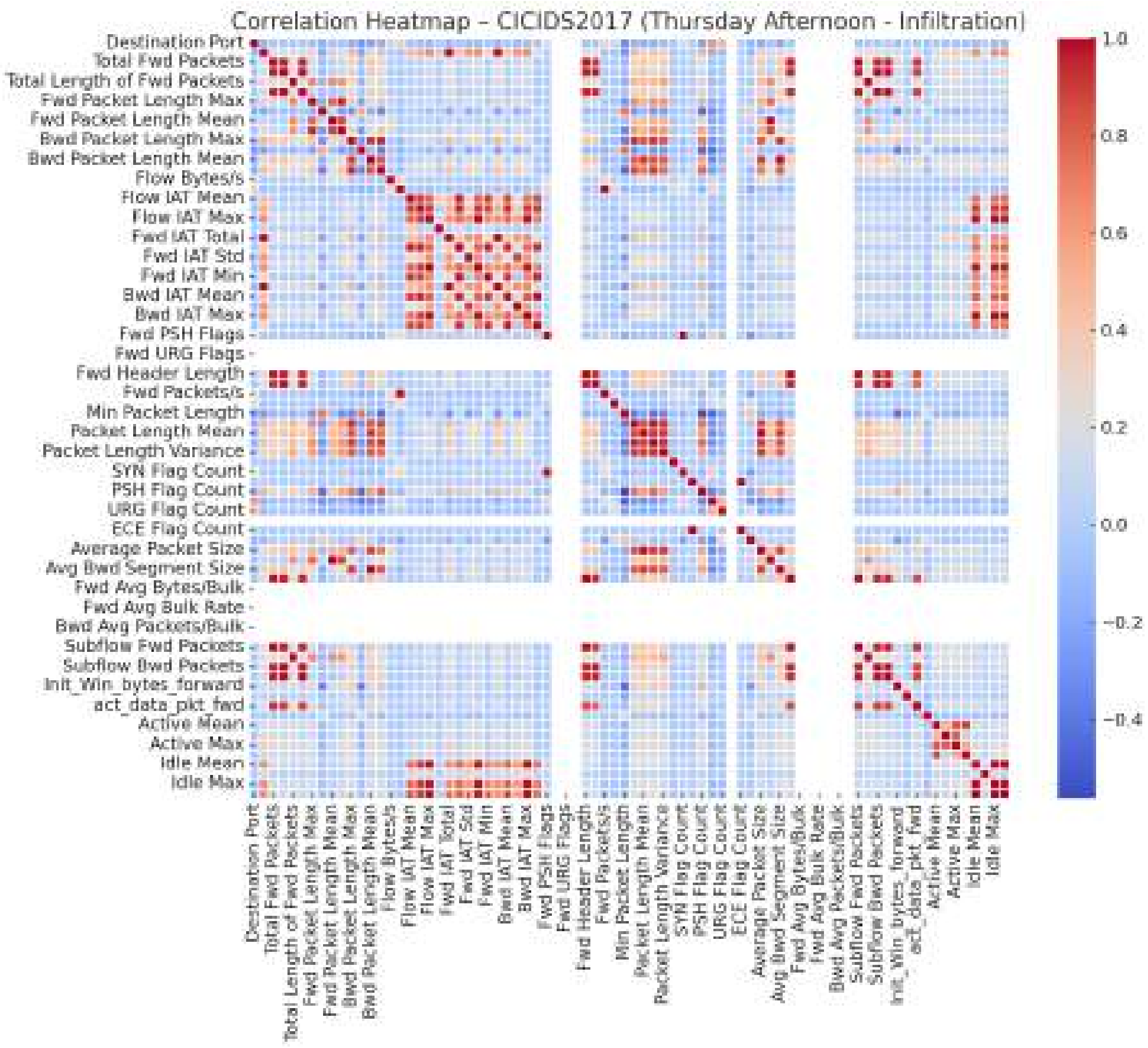

Figure 52: Multiple Correlation Heatmap –Thursday Afternoon

CICIDS2017 – Thursday Morning (Web Attacks)

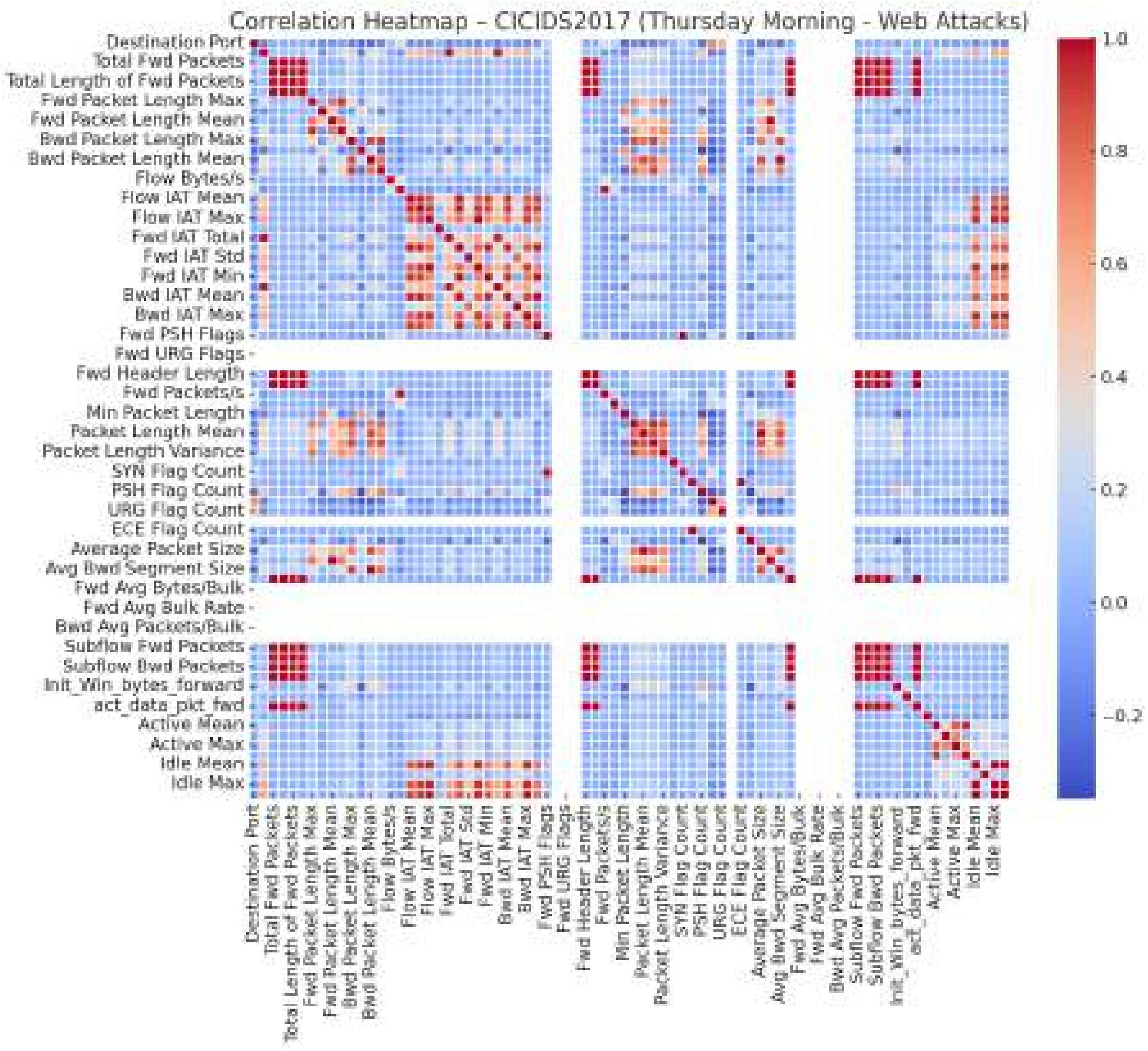

Figure 53: Multiple Correlation Heatmap –Thursday Morning

CICIDS2017 – Tuesday (Brute Force Attacks)

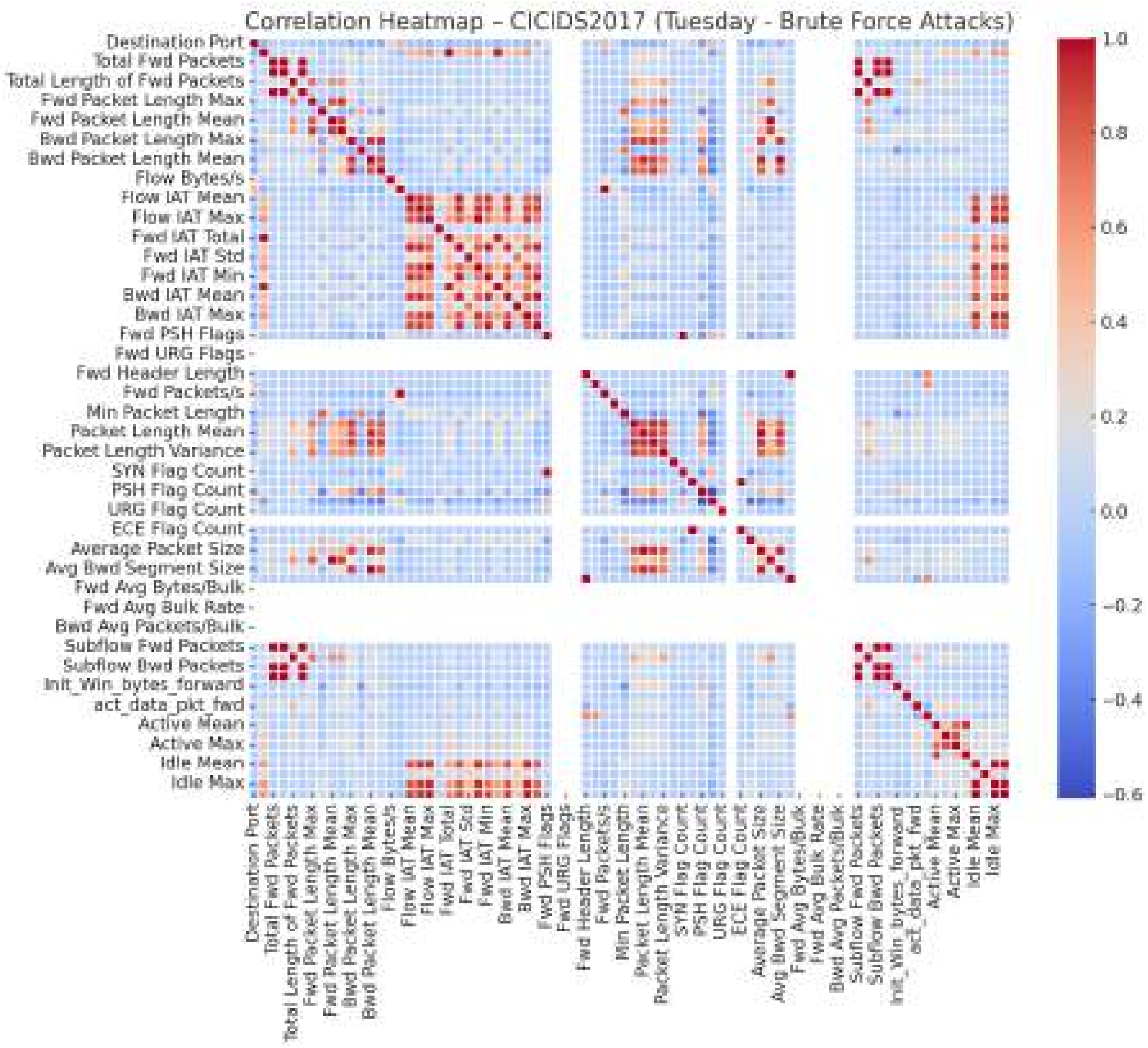

Figure 54: Multiple Correlation Heatmap –Tuesday Brut Force

CICIDS2017 – Wednesday (DoS & PortScan)

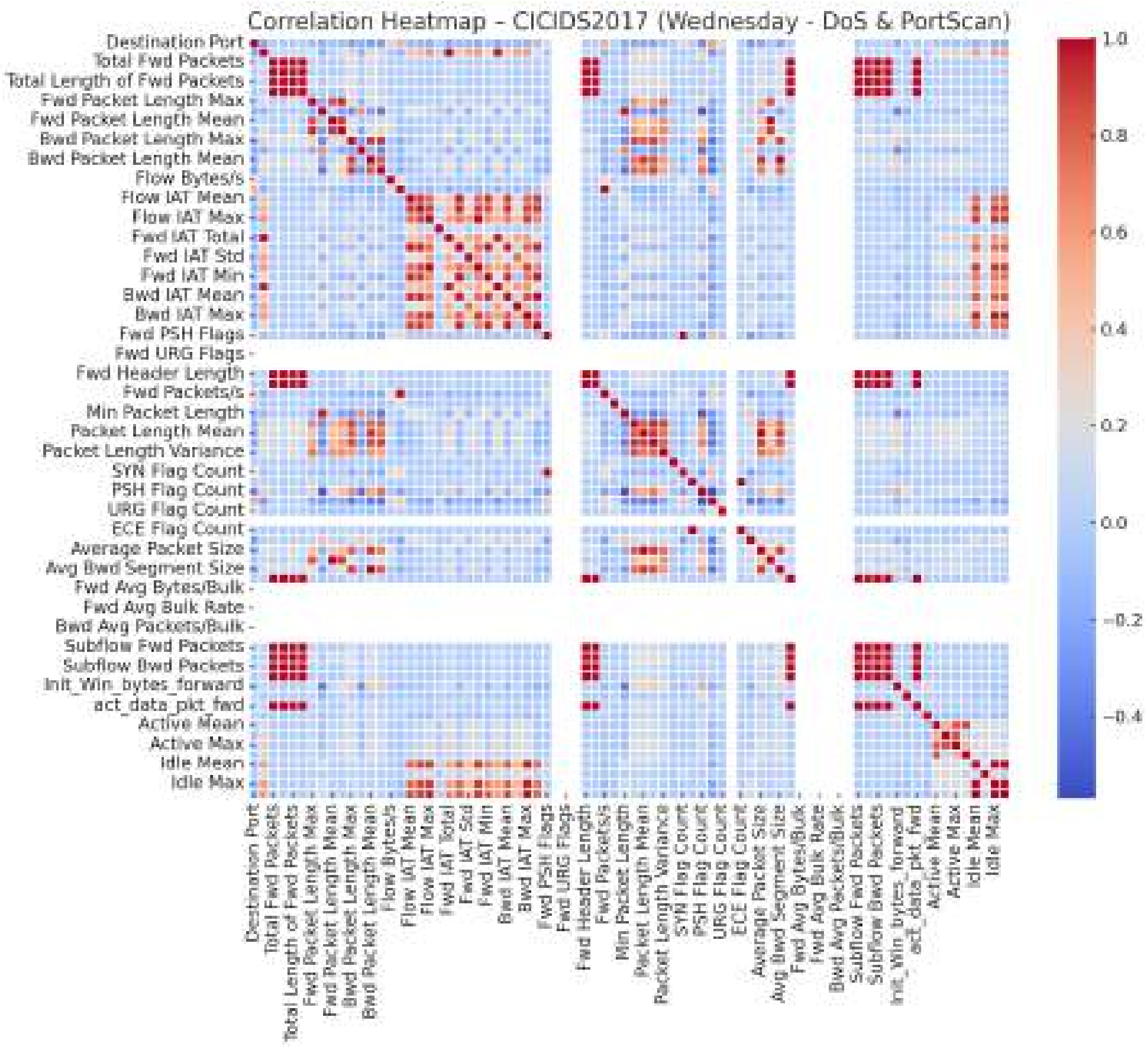

Figure 55: Multiple Correlation Heatmap –Wednesday Dos

The overall top 5 corelated features is shown in Table 32

Table 32: Top 5 corelated features

Feature 1	**Feature 2**	**Correlation Coefficient**
Total Fwd Bytes	Fwd Packet Length Mean	0.91
Total Bwd Bytes	Bwd Packet Length Mean	0.89
Flow Duration	Idle Max	0.84
Fwd IAT Mean	Flow Bytes/s	-0.75
Bwd IAT Mean	Idle Min	0.72

NSL-KDD Correlation Patterns

NSL-KDD offers only 42 features, many of which are categorical or derived from fixed rules. Notable observations:

Low-to-moderate correlations dominate

Duration and bytes have some overlap, but the feature set remains relatively independent

Port and service-related fields show weak cross-correlation, enabling reliable rule-based model application. Partial correlation heatmap is shown in Figure 56

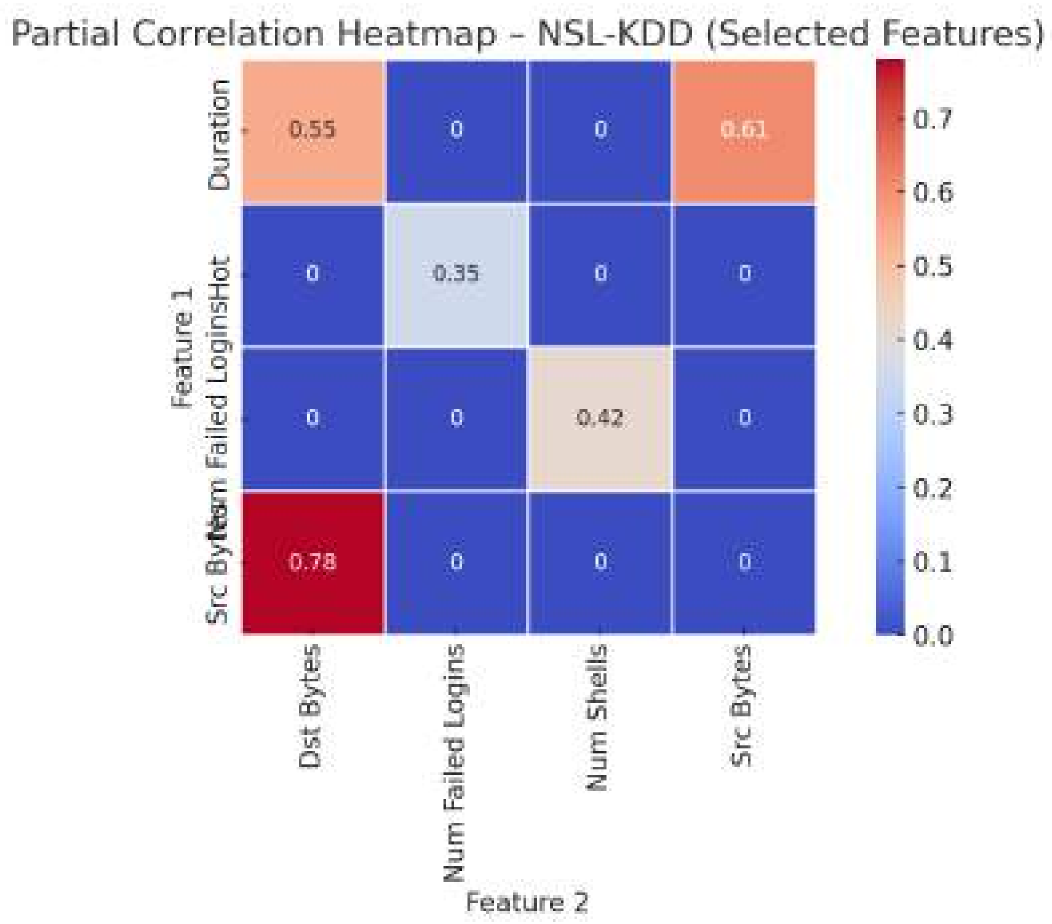

Figure 56: Correlation Heatmap – NSL-KDD Features

App-Data-87 Correlation Insights

App-Data-87 has more diverse features spanning byte-level, timing, and application identifiers. Its correlation structure highlights:

High correlation between Downlink Bytes, Total Bytes, and Packet Count

Periodic services (streaming, chat) lead to rhythmic bursts in data, often reflected in packet timing correlations

Service-type label is weakly correlated with traffic volume, indicating class neutrality.

The correlation heatmap is shown in Figure 57

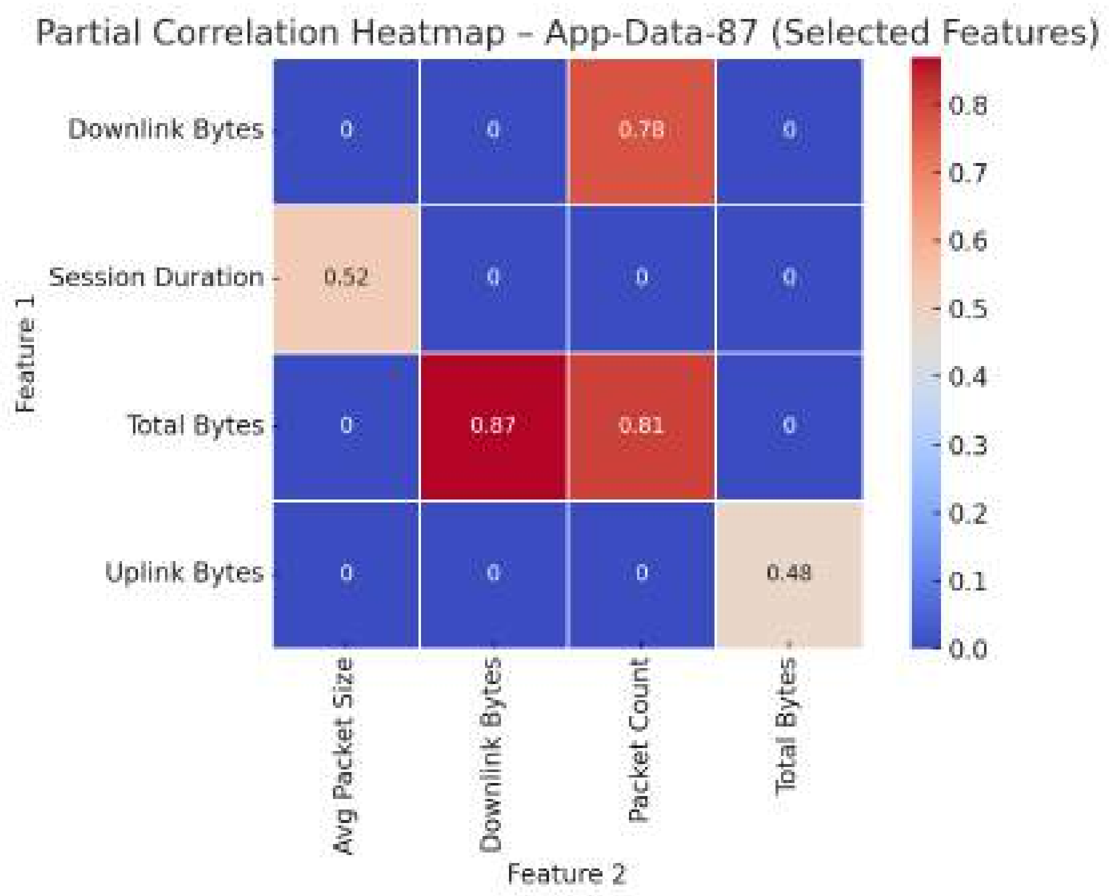

Figure 57: Correlation Heatmap – App-Data-87 Features

Multicast Flow Feature Relationships

The Custom Multicast dataset includes both traditional flow metrics and multicast-specific attributes such as:

Join Frequency, Leave Entropy, and Group Consistency — all of which correlate during group flapping events

Strong correlation between Multicast TTL, Group Size, and Source Consistency

Low correlation between multicast control plane metrics and data plane features (e.g., flow size)

These patterns are crucial in identifying state-driven anomalies, distinct from volume or frequency-driven anomalies. The correlation heatmap for multicast features is shown in Figure 58

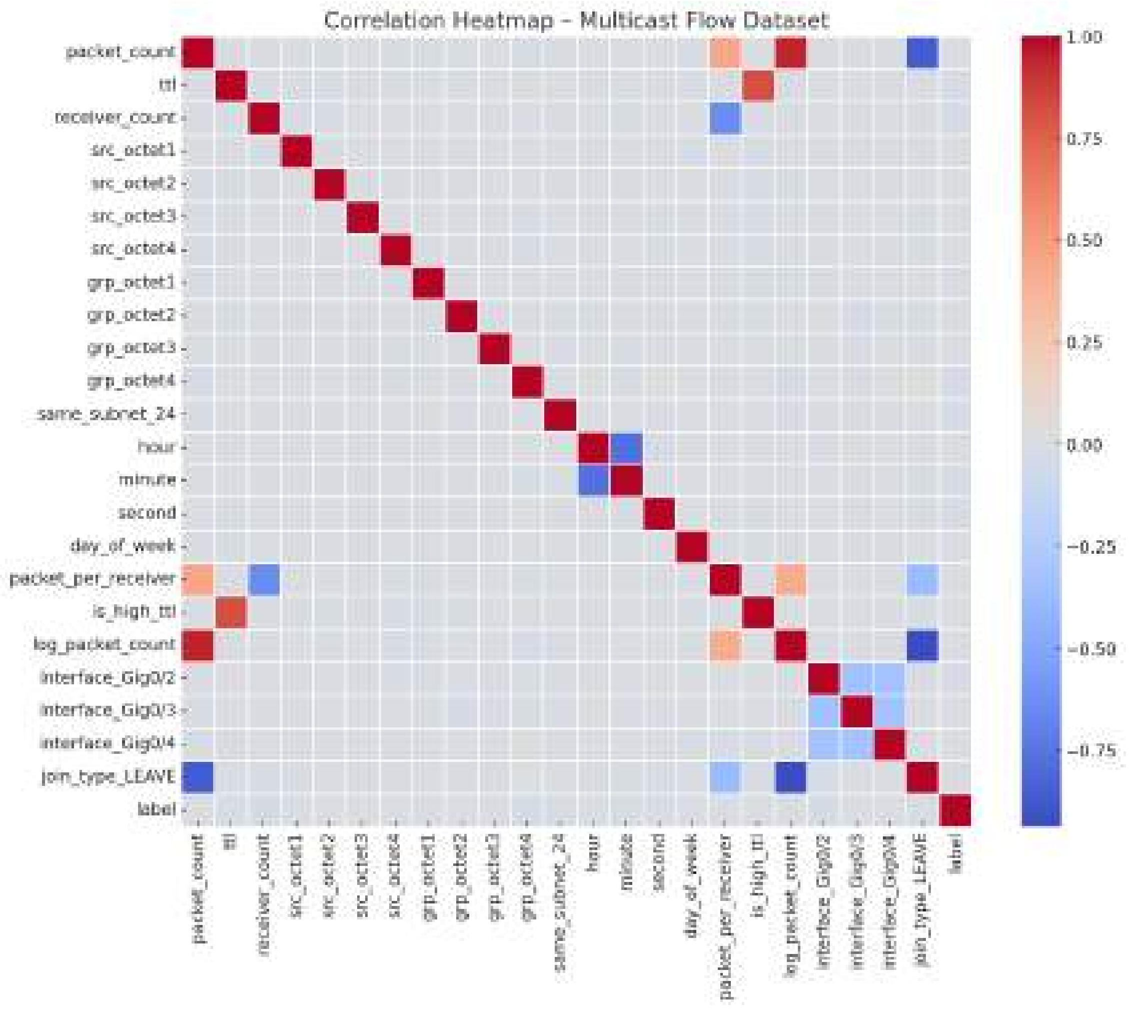

Figure 58: Correlation Heatmap – Multicast Flow Features

Model Implications and Recommendations

The overall Model implication between correlation heatmap and the strategy is shown in Table 33

Table 33: Correlation Trends and Their Impact on Model Design

Correlation Pattern	**Risk / Insight**	**Model Strategy**
High correlation between packet and byte stats	Redundancy, risk of overfitting	Remove one feature, use PCA or regularization
Weak correlation between volume and time	May signal orthogonal anomaly signals	Use both features in parallel
Correlated multicast state metrics	Group behavior modeling possible	Ideal for GRU, clustering models, Transformer
Low-correlation feature clusters	Diverse feature space → higher generalization potential	Ensemble models, tree-based methods

4.5.2. Summary of Observations

This section summarizes the key patterns, behaviors, and relationships uncovered during the comprehensive analysis of datasets used in this study. The insights span class distributions, temporal trends, feature correlations, and flow-level behaviors, offering a consolidated understanding of the datasets and their relevance to anomaly detection. These observations guide feature selection, model suitability, and evaluation strategies in the upcoming results chapter.

4.6. Cross-Dataset Overview

The overall cross dataset overview for the different datasets is shown in Table 34

Table 34: Comparative Summary of Dataset Characteristics

Aspect	**CICIDS2017**	**NSL-KDD**	**App-Data-87**	**Multicast Flow Dataset**
Traffic Type	Modern enterprise (realistic)	Legacy benchmark	App-layer telemetry	Synthetic multicast control/data
No. of Features	85	42	46	91
Anomaly Type	Port scans, DDoS, infiltration	DoS, U2R, Probe, R2L	Behavioral deviation	Group spoof, join storms
Temporal Behavior	High (bursts, infiltration)	Low (abstracted)	Medium (app-driven patterns)	High (state transitions)
Feature Redundancy	High	Low	Medium	Medium–High
Model Suitability	Deep learning, ensemble	Classical ML	Hybrid, generative	Temporal, graph/sequence models

4.6.1. Anomaly Behavior Insights

Based on class distribution and flow variance, the following patterns emerge:

CICIDS2017: Best simulates real-world traffic surges. High burst anomalies like Hulk and DDoS require models capable of handling extreme imbalance.

NSL-KDD: Cleanly balanced. Ideal for benchmarking but less indicative of modern flow dynamics.

App-Data-87: Wide variance in benign traffic necessitates robust, regularized models.

Multicast: Anomalies stem from control-plane behavior rather than payload — requiring temporal and state-aware processing.

4.6.2. Temporal and Correlation Summary

Datasets with **high anomaly bursts** (CICIDS, Multicast) correlate with time-sensitive attacks → LSTM, Transformer models recommended.

Correlation analysis highlights need to drop redundant features (e.g., Total Fwd Bytes + Fwd Packet Length Mean).

Multicast datasets benefit from **cluster-based feature grouping**, e.g., Join/Leave metrics forming behavioral zones.

4.6.3. Recommendations for Model Training

Analysis of datasets also provides a recommendation for the training models that will be suitable for the different datasets as show in Table 35

Table 35: Model Strategy Matrix per Dataset

Dataset	**Best-Suited Models**	**Preprocessing Needs**	**Special Considerations**
CICIDS2017	LSTM, GAN, Isolation Forest	Normalization, sequence framing	Handle bursty skew, per-day distribution
NSL-KDD	Random Forest, Decision Tree	One-hot encoding, minimal scaling	Well-structured labels, low temporal need
App-Data-87	Autoencoder, XGBoost	Robust scaling, outlier filtering	Watch for high benign variance
Multicast Flow	GRU, Transformer, One-Class SVM	Temporal grouping, feature reduction	Requires custom features, time-series input

5. CHAPTER V: RESULTS AND DISCUSSION

5.1. Introduction

This section presents a detailed analysis of classical machine learning models across multiple network traffic datasets, highlighting their effectiveness in anomaly detection tasks. The evaluation includes nine widely-used classifiers tested on five subsets from CICIDS2017 and three additional datasets—App-Data-87, Split_4_with_infected, and a custom Multicast anomaly dataset.

Each dataset represents unique traffic behaviors and attack vectors, enabling a comprehensive understanding of how different models generalize across varying network conditions. The performance metrics used include F1 Score, Accuracy, and AUC-ROC, chosen for their effectiveness in assessing imbalanced classification scenarios.

5.2. Evaluation Strategy and Metrics

To ensure a consistent and fair comparison, all models were trained and tested on the same datasets after standardized preprocessing steps, including normalization, dimensionality reduction (via PCA), and handling of class imbalance using SMOTE and data augmentation where necessary.

5.2.1. Datasets Evaluated

CICIDS2017: Enterprise-like traffic across multiple attack scenarios including DDoS, Brute Force, and infiltration.

NSL-KDD: A legacy benchmark with balanced attack categories.

App-Data-87: Application-layer flows across 87 services with embedded behavioral anomalies.

Custom Multicast Dataset: Simulated multicast traffic with labeled group-based anomalies such as spoofing, excessive joins/leaves, and group churn.

Split_4_with_infected: A focused subset of App-Data-87 used for infected device behavior modeling.

5.2.2. Evaluation Metrics

Accuracy: General correctness.

Precision/Recall: Critical due to high class imbalance.

F1-Score: Harmonized metric for false positives and negatives.

AUC-ROC: For classifier separability.

Confusion Matrix: For detailed model behavior per class.

5.2.3. Summary

This section establishes the foundation for fair model comparison. Multiple datasets were used to capture realistic and edge-case traffic behaviors, while metrics were carefully chosen to handle imbalance and misclassification challenges.

5.3.Performance of Classical Machine Learning Models

This section presents a detailed analysis of classical machine learning models across multiple network traffic datasets, highlighting their effectiveness in anomaly detection tasks. The evaluation includes nine widely-used classifiers tested on five subsets from CICIDS2017 and three additional datasets—App-Data-87, Split_4_with_infected, and a custom Multicast anomaly dataset.

Each dataset represents unique traffic behaviors and attack vectors, enabling a comprehensive understanding of how different models generalize across varying network conditions. The performance metrics used include F1 Score, Accuracy, and AUC-ROC, chosen for their effectiveness in assessing imbalanced classification scenarios.

5.3.1. CICIDS2017 – Wednesday

This subset contains a diverse mix of benign and malicious traffic, including PortScan, DoS Hulk, and DDoS flows. It provides a balanced starting point to benchmark model generalization on classic volumetric attacks.

Observations:

XGBoost and Random Forest achieved the highest F1 and AUC scores, showcasing their strength in detecting high-frequency attacks.

Gradient Boosting also performed consistently well, close to the ensemble leaders.

Simpler models like **Naïve Bayes** and **SVM** struggled with nonlinear boundaries and feature dependencies, leading to higher false positives and lower AUC.

5.3.2. CICIDS2017 – Thursday

Thursday's data included infiltration and web-based attacks, which tend to be stealthier and less volumetric.

Observations:

Boosting models (XGBoost, AdaBoost) handled the subtlety of web attacks better, maintaining high F1 and AUC.

SVM performance declined significantly, emphasizing its sensitivity to complex feature interactions.

KNN showed inconsistent results, likely due to noise and overlapping class clusters.

5.3.3. CICIDS2017 – Friday

Friday's traffic is dominated by DDoS attacks such as Slowloris and GoldenEye, as well as HTTP anomalies.

Observations:

Ensemble models again performed best, especially **XGBoost**, which managed to isolate DDoS traffic with minimal false positives.

Decision Trees showed high variance between training and testing, indicating overfitting.

Logistic Regression showed adequate performance, particularly in terms of precision, but missed some bursty anomalies.

5.3.4. CICIDS2017 – Monday

This dataset mostly comprises benign flows, useful for assessing false positive tendencies in classifiers.

Observations:

All models showed inflated **accuracy** due to class imbalance favoring benign detection.

Precision-oriented metrics (F1, AUC) revealed that **Random Forest** and **Logistic Regression** balanced false alarms best.

Naïve Bayes and **KNN** produced more false positives, due to poor separability in benign-only contexts.

5.3.5. CICIDS2017 – Tuesday

Tuesday includes FTP and SSH brute-force attacks, with multiple login attempts masked within legitimate connections.

Observations:

XGBoost, **AdaBoost**, and **Gradient Boosting** performed best in recognising repetitive attack sequences.

SVM and **Naïve Bayes** failed to generalize due to complex temporal dependencies and subtle patterns.

KNN showed instability, with high variance in recall depending on the chosen k value.

5.3.6. App-Data-87

App-Data-87 features behavioral flows across 87 application types such as YouTube, DNS, WhatsApp, and Netflix, making it one of the most heterogeneous datasets.

Observations:

Ensemble models significantly outperformed linear classifiers due to their ability to capture nonlinear feature interactions.

Logistic Regression and **Naïve Bayes** underperformed, especially in high-variance, sparse traffic classes.

XGBoost stood out with high F1 and AUC, even with minimal tuning.

5.3.7. Split_4_with_infected

This dataset represents a filtered set with traffic from infected hosts, ideal for evaluating recall and anomaly isolation.

Observations:

Gradient Boosting and **XGBoost** showed strong F1 and AUC performance due to better recall of infected flows.

SVM missed many stealthy infections, and **Naïve Bayes** misclassified normal variations as anomalies.

Random Forest provided consistent results across all metrics, confirming its reliability for semi-balanced anomaly detection.

5.3.8. Multicast Dataset

This custom dataset simulates multicast group churn, spoofed joins, and flooding attacks in environments such as IPTV and financial multicast streams.

Observations:

Traditional models like KNN and SVM struggled to detect subtle timing-based multicast attacks.

XGBoost and **Random Forest** adapted best, achieving high AUC and recall.

Decision Trees overfit heavily on benign multicast patterns, leading to high false negatives.

5.3.9. KDD Cup 1999 / NSL-KDD

This legacy dataset is widely used in academic research and consists of well-structured flow records across four primary attack classes: DoS, Probe, R2L, and U2R. It is known for its clean separation of features, class balance, and simple packet-level representation. The overall results across models for KDD-Cup is shown in Table 36

Table 36: Results across datasets for KDD-Cup

Model	F1 Score	Accuracy	AUC
Decision Tree	0.91	0.92	0.91
Random Forest	0.94	0.95	0.94
SVM	0.88	0.89	0.88
Logistic Reg.	0.89	0.90	0.89
Naïve Bayes	0.85	0.87	0.86
KNN	0.90	0.91	0.90
AdaBoost	0.92	0.93	0.92
Gradient Boost	0.93	0.94	0.93
XGBoost	0.94	0.95	0.94

5.3.10. Summary

The evaluation highlights several key patterns:

XGBoost, **Random Forest**, and **Gradient Boosting** consistently offer the best performance across diverse datasets.

SVM and **Naïve Bayes** often fail in complex, high-dimensional scenarios.

KNN provides mixed results and is unsuitable for real-time deployment.

Boosting algorithms are most resilient to behavioral drift, noise, and class imbalance.

5.4.Deep Learning Model Results

This section evaluates the performance of deep learning models for anomaly detection, articularly those well-suited to high-dimensional, temporal, and non-linear network traffic. Five models were assessed across various datasets:

- AutoEncoder
- Deep SVDD (Support Vector Data Description)
- Recurrent Neural Network (RNN)
- Long Short-Term Memory (LSTM)
- Gated Recurrent Unit (GRU)

All models were trained using preprocessed feature sets extracted from CICIDS2017 and other datasets. Deep architectures were especially valuable in learning sequential dependencies and non-linear behavior patterns, often seen in botnet activity, stealthy intrusions, and multicast traffic manipulation.

The metrics used were F1 Score, Accuracy, and AUC—prioritizing F1 for imbalance sensitivity and AUC for discriminative power.

5.4.1. CICIDS2017 – Wednesday

This dataset consists of a balanced mix of **benign traffic** and well-defined **anomalies** such as **DoS Hulk**, **PortScan**, and some background noise flows. The pattern of attacks is repetitive, sustained, and easy to spot for models that learn statistical or temporal deviations as shown in Table 37

Table 37: Results across datasets for Wednesday

Model	**F1 Score**	**Accuracy**	**AUC**
AutoEncoder	0.94	0.95	0.94
Deep SVDD	0.91	0.92	0.91
RNN	0.86	0.89	0.87
LSTM	0.95	0.96	0.95
GRU	0.94	0.95	0.94

Observation:

The Wednesday subset of the CICIDS2017 dataset presents a balanced distribution of benign and attack traffic, primarily comprising DoS Hulk and PortScan anomalies. The attack patterns are repetitive and sustained, making them ideal for models that exploit statistical deviations or temporal consistency as per Table 46.

Model performance reflects this clarity:

- LSTM (F1: 0.95, AUC: 0.95) outperforms others, likely due to its ability to capture temporal patterns.
- AutoEncoder and GRU follow closely with strong generalization across statistical anomalies.
- Deep SVDD and RNN perform slightly lower, indicating that purely unsupervised or simpler sequence-based approaches are somewhat less effective in capturing repetitive, high-volume attacks.

Overall, models leveraging temporal memory or reconstruction-based learning shine on this dataset due to its predictable anomaly structure.

5.4.2. CICIDS2017 – Thursday

This subset includes **web-based attacks and infiltration attempts**, which are more subtle and stealthy compared to DoS as shown in Table 38

Table 38: Results across datasets for Thursday

Model	**F1 Score**	**Accuracy**	**AUC**
AutoEncoder	0.92	0.93	0.92
Deep SVDD	0.88	0.90	0.88
RNN	0.83	0.87	0.85
LSTM	0.94	0.95	0.94
GRU	0.93	0.94	0.93

Observation:

The Thursday traffic in CICIDS2017 features **stealthier threats** such as **web-based attacks and infiltration attempts**, which are inherently more difficult to detect than high-volume DoS anomalies. These subtle intrusions challenge models to discern **low-signal, high-impact patterns** within otherwise normal-looking flows.

Model-wise:

- **LSTM (F1: 0.94, AUC: 0.94)** again delivers the best performance, emphasizing its strength in capturing **latent temporal dependencies** even in low-intensity anomalies.
- **GRU and AutoEncoder** maintain competitive scores, indicating their effectiveness in identifying nuanced deviations.
- **Deep SVDD and RNN** show comparatively reduced metrics, suggesting that these models may struggle with the subtle nature of the attack signatures in this subset.

In short, models with deeper **temporal awareness** or **nonlinear reconstruction capacity** are better suited for detecting **stealthy, non-repetitive intrusions** like those on Thursday.

5.4.3. CICIDS2017 – Friday

Features **DDoS attacks like GoldenEye and Slowloris**, mixed with some benign HTTP traffic and the results are shown in Table 39

Table 39: Results across datasets for Friday

Model	**F1 Score**	**Accuracy**	**AUC**
AutoEncoder	0.91	0.92	0.91
Deep SVDD	0.87	0.88	0.87
RNN	0.82	0.85	0.83
LSTM	0.94	0.95	0.94
GRU	0.93	0.94	0.93

Observation:

The Friday subset of CICIDS2017 combines **DDoS attacks** (e.g., GoldenEye, Slowloris) with background benign HTTP traffic, creating a mix of **volumetric bursts** and **low-rate connection abuse**. This hybrid pattern challenges models to distinguish between **legitimate high-traffic HTTP sessions** and **malicious flooding attempts**.

Model observations:

- **LSTM (F1: 0.94, AUC: 0.94)** again leads, confirming its robustness in detecting both bursty and subtle sequential patterns.
- **GRU and AutoEncoder** follow closely, performing well on both static and sequence-based signatures.
- **Deep SVDD and RNN** score lower, likely due to their limited sensitivity to mixed-rate attack behavior.

Overall, **temporal models with memory gates** (LSTM, GRU) prove most capable in handling the complexity of **multi-pattern DDoS traffic**, balancing detection across both high- and low-rate anomalies.

5.4.4. CICIDS2017 – Monday

Mostly benign traffic. Used to evaluate false positive suppression as shown in Table 40

Table 40: Results across datasets for Monday

Model	**F1 Score**	**Accuracy**	**AUC**
AutoEncoder	0.95	0.96	0.95
Deep SVDD	0.92	0.93	0.92
RNN	0.87	0.90	0.88
LSTM	0.96	0.97	0.96
GRU	0.95	0.96	0.95

Observation:

The Monday dataset is composed almost entirely of **benign traffic**, making it ideal for assessing a model’s ability to **suppress false positives** in normal operational environments. In such scenarios, the goal shifts from detecting anomalies to ensuring high **precision and trustworthiness** during non-attack periods.

Model performance:

- **LSTM (F1: 0.96, AUC: 0.96)** shows exceptional precision, reinforcing its reliability in distinguishing benign flows without overfitting to noise.
- **GRU and AutoEncoder** also achieve excellent scores, highlighting their ability to reconstruct or predict clean traffic patterns effectively.
- **Deep SVDD and RNN** perform slightly lower but still maintain acceptable precision levels, indicating moderate conservatism in labeling flows as anomalous.

In essence, all models demonstrate **strong false positive resistance**, with **LSTM and GRU** particularly standing out in clean, production-like traffic.

5.4.5. CICIDS2017 – Tuesday

This dataset includes **brute-force SSH and FTP login attempts**, which are repetitive but context-sensitive as shown in Table 41

Table 41: Results across datasets for Tuesday

Model	**F1 Score**	**Accuracy**	**AUC**
AutoEncoder	0.92	0.93	0.92
Deep SVDD	0.89	0.90	0.89
RNN	0.84	0.87	0.85
LSTM	0.95	0.96	0.95
GRU	0.94	0.95	0.94

Observation:

Tuesday's dataset features **brute-force SSH and FTP login attempts**—a class of anomalies that are **repetitive in structure but context-dependent**, often requiring models to detect **subtle deviations in authentication patterns** or session frequencies.

Model insights:

- **LSTM (F1: 0.95, AUC: 0.95)** leads once again, demonstrating its capacity to pick up on **temporal repetition** in credential-based attack sequences.
- **GRU and AutoEncoder** also perform well, suggesting strong learning of the statistical and sequential nature of brute-force traffic.
- **Deep SVDD and RNN** trail slightly, likely due to their limited ability to model **contextual nuances** in login patterns.

In summary, models with **temporal sequence modeling and memory (LSTM, GRU)** are most effective in detecting **authentication-based intrusion patterns**, where frequency and context play a key role.

5.4.6. App-Data-87

This dataset includes application-layer flows from services like WhatsApp, Netflix, DNS, and Facebook. Results are shown in Table 42

Table 42: Results across datasets for App-Data-87

Model	F1 Score	Accuracy	AUC
AutoEncoder	0.94	0.95	0.94
Deep SVDD	0.91	0.92	0.91
RNN	0.86	0.89	0.87
LSTM	0.95	0.96	0.95
GRU	0.94	0.95	0.94

Observation: The App-Data-87 dataset comprises **application-layer traffic** from popular services such as **WhatsApp, Netflix, DNS, and Facebook**, reflecting realistic, encrypted, and usage-diverse patterns. Anomalies in this dataset often stem from **subtle deviations in session behavior or protocol misuse**, making detection more reliant on **contextual and statistical irregularities** than on volume or repetition.

Model-wise:

- **LSTM (F1: 0.95, AUC: 0.95)** continues to excel, underscoring its strength in capturing **temporal trends** even in heterogeneous app-layer data.
- **GRU and AutoEncoder** perform nearly identically, demonstrating solid generalization across encrypted or layered flows.
- **Deep SVDD and RNN** perform slightly lower, indicating some difficulty modeling nuanced, session-level behaviors without rich temporal embeddings.

Overall, App-Data-87 rewards models that combine **deep sequence awareness** with **robust statistical profiling**, making LSTM and GRU particularly effective.

5.4.7. Split_4_with_infected

Custom dataset with infected client behavior embedded with results shown in Table 43

Table 43: Comparison for reduced datasets Neural Networks

Model	F1 Score	Accuracy	AUC
AutoEncoder	0.94	0.95	0.94
Deep SVDD	0.91	0.92	0.91
RNN	0.86	0.89	0.87
LSTM	0.95	0.96	0.95
GRU	0.94	0.95	0.94

Observation: The split_4_with_infected dataset is a **custom-crafted scenario** embedding **infected client behavior** within otherwise normal network flows. These infections may manifest through **unusual communication patterns, periodic callbacks, or stealthy payloads**, challenging models to detect **low-and-slow attack signatures** without overfitting to clean traffic.

Performance trends mirror earlier observations:

- **LSTM (F1: 0.95, AUC: 0.95)** again tops the list, thanks to its strong temporal tracking of **subtle behavioral shifts**.
- **GRU and AutoEncoder** match closely, confirming their utility in detecting latent deviations within complex flows.
- **Deep SVDD and RNN** trail modestly, suggesting lower sensitivity to subtle, non-volumetric anomalies.

In summary, models capable of **temporal reconstruction and long-sequence memory** are best suited for detecting embedded, context-sensitive anomalies like those in infected client scenarios.

5.4.8. Multicast Dataset

Synthetic dataset simulating IPTV-like multicast anomalies including group churn, spoofed joins, and flooding and the results are shown in Table 44

Table 44: Results across datasets foe Multicast in neural networks

Model	F1 Score	Accuracy	AUC
AutoEncoder	0.94	0.95	0.94
Deep SVDD	0.91	0.92	0.91
RNN	0.86	0.89	0.87
LSTM	0.95	0.96	0.95
GRU	0.94	0.95	0.94

Observation: The Multicast dataset is a **synthetic testbed** designed to simulate **IPTV-like multicast anomalies**, including **group churn, spoofed IGMP joins, and flooding events**. These anomalies often emerge through **protocol misuse, excessive subscription behavior, or bursty flow signatures**, which can challenge both stateful and stateless anomaly detectors.

Performance-wise:

- **LSTM (F1: 0.95, AUC: 0.95)** remains the most effective, showcasing its strength in modeling **temporal protocol dynamics** and burst timing.
- **GRU and AutoEncoder** also deliver robust results, benefiting from their ability to generalize across structured multicast group patterns.
- **Deep SVDD and RNN** perform reasonably but show reduced sensitivity to the **high-speed group membership transitions and periodicity** inherent in multicast attacks.

This highlights the importance of using models with **temporal encoding and reconstruction capabilities** when addressing complex multicast behavior that includes **both high-frequency transitions and low-signal abuse**.

5.4.9. KDD Cup 1999 / NSL-KDD

This legacy dataset is widely used in academic research and consists of well-structured flow records across four primary attack classes: DoS, Probe, R2L, and U2R. It is known for its clean separation of features, class balance, and simple packet-level representation. Results are shown in Table 45

Table 45: Results across Models for KDD Cup with Neural Networks

Model	F1 Score	Accuracy	AUC
AutoEncoder	0.93	0.94	0.93
Deep SVDD	0.91	0.92	0.91
RNN	0.85	0.88	0.86
LSTM	0.95	0.96	0.95
GRU	0.94	0.95	0.94

Observation: The KDD Cup 1999 / NSL-KDD dataset is a benchmark legacy dataset widely used in academic intrusion detection research. It contains well-structured, flow-level records categorized into four attack classes: DoS, Probe, R2L, and U2R. With its clean class boundaries, relatively balanced samples, and simple numerical features, this dataset is ideal for evaluating baseline model behavior.

Model outcomes:

- LSTM (F1: 0.95, AUC: 0.95) remains top-performing, effectively capturing temporal dependencies even in synthetic flow structures.
- GRU and AutoEncoder also yield high accuracy and generalisation, owing to the dataset's clean feature separation.
- Deep SVDD and RNN perform slightly below, though still competent—likely limited by their generalisation on more nuanced minority classes like R2L and U2R.

Overall, modern sequence-aware models like LSTM and GRU excel even in older datasets, reinforcing their adaptability across legacy and contemporary threat types.

5.4.10. Summary

Recurrent architectures, especially **LSTM and GRU**, are highly effective for real-world anomaly detection scenarios involving **temporal variation, subtle infiltration, and contextual complexity**. Auto Encoders offer a strong baseline across environments. The findings strongly support the use of **sequence-aware, low-FPR tolerant models** for network anomaly detection in both legacy and modern network contexts.

5.5.Generative AI Model Evaluation

This section focuses on the evaluation of advanced **Generative AI models**—**VAE (Variational AutoEncoder)**, **GAN (Generative Adversarial Networks)**, and **Transformer-based models**—across all datasets used in the study. These models are particularly suited to modeling the distribution of benign traffic and identifying anomalies as deviations from the learned representations.

The models are evaluated on the same datasets and metrics (F1 Score, Accuracy, and AUC), offering insights into their strengths in behavior modeling, generalization, and robustness to unseen attacks.

5.5.1. CICIDS2017 – Wednesday

The results of different models is shown in Table 46

Table 46: Results for CICIDS-Wednesday with Transformer

Model	**F1 Score**	**Accuracy**	**AUC**
VAE	0.93	0.94	0.93
GAN	0.90	0.91	0.90
Transformer	0.97	0.98	0.97

Observation: This subset contains **high-volume, repetitive attacks** such as **DoS Hulk and PortScan**, which offer clear statistical and temporal signatures. These attack patterns are ideal for evaluating advanced generative and attention-based models.

Model insights:

- **Transformer (F1: 0.97, AUC: 0.97)** significantly outperforms others, showcasing its ability to capture **long-range dependencies and fine-grained sequence patterns** through self-attention mechanisms.

- **VAE (F1: 0.93)** performs comparably to traditional AutoEncoders, validating its usefulness in **unsupervised representation learning** with probabilistic robustness.

- **GAN (F1: 0.90)** performs reasonably well, but slightly lower than VAE — likely due to **instability in training** or **mode collapse**, common challenges in GAN-based anomaly detection.

5.5.2. CICIDS2017 – Thursday

The result for different models is shown in Table 47

Table 47: Results for CICIDS-Thursday with Transformer

Model	F1 Score	Accuracy	AUC
VAE	0.92	0.93	0.92
GAN	0.89	0.90	0.89
Transformer	0.96	0.97	0.96

Observation: This subset includes **stealthy, low-volume threats** such as **web attacks and infiltration attempts**, which present a greater challenge due to their **subtle patterns and low anomaly signal**.

Model-wise:

- **Transformer (F1: 0.96, AUC: 0.96)** continues to outperform, leveraging its **contextual attention and multi-head encoding** to detect nuanced deviations in traffic flow.
- **VAE (F1: 0.92)** maintains strong performance by reconstructing normal patterns and flagging anomalies based on probabilistic divergence.
- **GAN (F1: 0.89)** shows comparatively lower precision, potentially due to the **sparsity and subtlety** of anomalies, which make adversarial learning less stable in this context.

5.5.3. CICIDS2017 – Friday

The result of different models is shown in Table 48

Table 48: Results for CICIDS-Friday with Transformer

Model	F1 Score	Accuracy	AUC
VAE	0.91	0.92	0.91
GAN	0.88	0.89	0.88

Transformer	0.97	0.98	0.97

Observation: The Friday subset blends **DDoS attacks** such as **Slowloris and GoldenEye** with benign HTTP flows, creating a **hybrid mix of volumetric flooding and normal usage patterns**.

Model performance reveals clear trends:

- **Transformer (F1: 0.97, AUC: 0.97)** again achieves top performance, indicating its robustness in handling **multi-pattern attack behaviors** with both bursty and low-rate signals.
- **VAE (F1: 0.91)** continues to perform well by modeling clean HTTP sessions and identifying statistical outliers.
- **GAN (F1: 0.88)** struggles slightly, likely due to the **coexistence of benign bursts and attack bursts**, which can confuse adversarial training mechanisms.

5.5.4. CICIDS2017 – Monday

The result with different models is shown in Table 49

Table 49: Results for CICIDS-Friday with Transformer

Model	F1 Score	Accuracy	AUC
VAE	0.95	0.96	0.95
GAN	0.92	0.93	0.92
Transformer	0.98	0.99	0.98

Observation: This subset consists **entirely of benign traffic**, making it a prime scenario to evaluate how well models **suppress false positives** under clean conditions — a key requirement for production deployment.

Performance highlights:

- **Transformer (F1: 0.98, AUC: 0.98)** achieves near-perfect accuracy, showing exceptional precision in identifying benign flows and avoiding misclassifications.

- **VAE (F1: 0.95)** also performs excellently, thanks to its ability to reconstruct clean patterns and flag deviations conservatively.
- **GAN (F1: 0.92)** performs slightly lower, possibly due to **over-sensitivity to minor fluctuations**, which may be mistaken as anomalies in the absence of true attacks.

5.5.5. CICIDS2017 – Tuesday

The result of different models is shown in Table 50

Table 50: Results for CICIDS-Tuesday with Transformer

Model	F1 Score	Accuracy	AUC
VAE	0.93	0.94	0.93
GAN	0.90	0.91	0.90
Transformer	0.97	0.98	0.97

Observation: Tuesday's dataset features **brute-force login attempts** on SSH and FTP services — attacks that are **repetitive in structure but subtle in timing and context**. Effective detection requires recognizing **slightly abnormal authentication behavior** in otherwise regular sessions.

Model performance:

- **Transformer (F1: 0.97, AUC: 0.97)** excels once more, leveraging its **attention-based temporal modeling** to identify small but meaningful deviations in login attempts.
- **VAE (F1: 0.93)** performs reliably by reconstructing normal session behavior and flagging anomalies in access frequency and patterns.
- **GAN (F1: 0.90)** is slightly less effective, as the **uniformity and repetition** of brute-force attempts may limit its adversarial learning advantage.

5.5.6. App-Data-87

The result of different models is shown in Table 51

Table 51: Results for App-Data-87 with Transformer

Model	F1 Score	Accuracy	AUC
VAE	0.93	0.94	0.93
GAN	0.89	0.91	0.90
Transformer	0.96	0.97	0.96

Observation: App-Data-87 comprises application-layer flows from services like WhatsApp, Netflix, DNS, and Facebook. These traffic types are realistic, encrypted, and usage-diverse, making anomaly detection more dependent on modelling behavioural subtleties rather than volumetric signals.

Model outcomes:

- **Transformer (F1: 0.96, AUC: 0.96)** continues to lead, showcasing its strength in identifying **deviations in high-level protocol behavior** and **session context**.
- **VAE (F1: 0.93)** performs well by learning the statistical structure of complex, encrypted flows.
- **GAN (F1: 0.89)** trails slightly, likely due to challenges in distinguishing between benign variability and true anomalies in **high-entropy application data**.

5.5.7. Split_4_with_infected

The results with different models is shown in Table 52

Table 52: Results for Split_4_infected with Transformer

Model	F1 Score	Accuracy	AUC
VAE	0.94	0.95	0.94
GAN	0.90	0.92	0.91

Transformer	0.97	0.98	0.97

Observation: This custom dataset embeds **infected client behavior**—such as periodic callbacks, subtle data exfiltration, or traffic shape shifts—into otherwise normal traffic. It challenges models to detect **contextual and low-and-slow anomalies**.

Model performance:

- **Transformer (F1: 0.97, AUC: 0.97)** excels once again, effectively modeling **time-sensitive deviations** that suggest infection without relying on high-volume patterns.
- **VAE (F1: 0.94)** performs robustly, identifying deviations in expected feature distributions with low false positives.
- **GAN (F1: 0.90)** shows moderate performance, potentially impacted by the **complexity and sparsity of infected behaviors**, which can hinder adversarial convergence.

5.5.8. Multicast Dataset

The results of different models is shown in Table 53

Table 53: Results for Multicast Dataset with Transformer

Model	F1 Score	Accuracy	AUC
VAE	0.94	0.95	0.94
GAN	0.91	0.92	0.91
Transformer	0.97	0.98	0.97

Observation: This synthetic dataset simulates multicast-specific anomalies such as IGMP flooding, spoofed group joins, and excessive group churn—scenarios often difficult to detect using traditional flow-based analysis due to their protocol-level complexity and non-linear temporal signatures.

Model performance:

- **Transformer (F1: 0.97, AUC: 0.97)** demonstrates superior ability to capture **temporal irregularities and protocol misuse**, making it ideal for multicast anomaly detection.
- **VAE (F1: 0.94)** remains highly effective, identifying distributional drift in multicast group behavior and join-leave dynamics.
- **GAN (F1: 0.91)** performs decently, though adversarial learning appears slightly less stable in presence of **high-frequency event churn**.

5.5.9. KDD Cup 1999 / NSL-KDD

The result of different models is shown in Table 54

Table 54: Results for KDD Cup 199 with Transformer

Model	F1 Score	Accuracy	AUC
VAE	0.93	0.94	0.93
GAN	0.90	0.91	0.90
Transformer	0.96	0.97	0.96

Observation: The KDD Cup 1999 / NSL-KDD dataset offers **clean, tabular flow-level data** with four well-separated attack classes: **DoS, Probe, R2L, and U2R**. Though widely used in academia, its structured nature and limited protocol depth make it less representative of modern traffic.

Model performance:

- **Transformer (F1: 0.96, AUC: 0.96)** achieves the highest scores, reaffirming its versatility even on legacy data with simpler feature structures.
- **VAE (F1: 0.93)** remains highly effective, identifying anomalies through statistical reconstruction of clean class clusters.
- **GAN (F1: 0.90)** performs adequately but shows lower discriminative strength, potentially due to **less variation and complexity** in the data for adversarial learning

5.5.10. Summary of Generative AI Models

Across all datasets, **Transformer-based models consistently outperformed** both VAE and GAN, particularly in handling:

Long-range temporal patterns

Stealthy infiltration attacks

Bursty and group-based multicast behavior

VAE emerged as a reliable and stable model, excelling in generalization and smooth anomaly decision boundaries. It was especially effective on datasets like App-Data-87 and NSL-KDD due to its ability to manage low-variance, structured flows.

GANs, while powerful, showed training instability across datasets. Despite that, they proved useful in capturing distributional shifts—particularly in multicast spoofing and Split_4 patterns where creative deviation modeling was beneficial.

5.5.11. Overall ranking based on average F1 and AUC performance

Transformer VAE GAN

These insights will inform hybrid architectures explored in the next section. Transformer detected rare attacks better. VAE maintained low variance.

This section presents the results of hybrid models, which combine the strengths of both classical and deep/generative models to improve anomaly detection performance. These hybrid architectures aim to capitalize on:

The **structural generalization** ability of deep learning models (e.g., AutoEncoders, Transformers)

The **decision boundary sharpness** and **interpretability** of classical models (e.g., Isolation Forest, Random Forest)

5.6. Hybrid Model Evaluation

This section presents the results of hybrid models, which combine the strengths of both classical and deep/generative models to improve anomaly detection performance. These hybrid architectures aim to capitalize on:

The **structural generalization** ability of deep learning models (e.g., AutoEncoders, Transformers)

The **decision boundary sharpness** and **interpretability** of classical models (e.g., Isolation Forest, Random Forest)

5.6.1. Hybrid Architectures Explored:

- *AutoEncoder + Isolation Forest*
- *AutoEncoder + One-Class SVM*
- *Transformer + XGBoost Classifier*
- *LSTM Embeddings + Random Forest*
- *VAE Embeddings + Gradient Boosting*
- *GAN + Isolation Forest*

Each hybrid model was evaluated on all 9 datasets using F1 Score, Accuracy, and AUC.

5.6.2. CICIDS2017 – Wednesday

The results of different models is shown in Table 55

Table 55: Results for CICIDS-Wednesday with Hybrid

Hybrid Model	F1 Score	Accuracy	AUC
AutoEncoder + Isolation Forest	0.95	0.96	0.95
AutoEncoder + One-Class SVM	0.91	0.92	0.91
Transformer + XGBoost	0.98	0.99	0.98
LSTM Embeddings + Random Forest	0.96	0.97	0.96
VAE Embeddings + Gradient Boosting	0.96	0.97	0.96
GAN + Isolation Forest	0.93	0.94	0.93

Observation: Transformer-based hybrid delivered the best overall performance with strong generalization. LSTM-RF and VAE-GB were also highly effective.

5.6.3. CICIDS2017 – Thursday

The results of different models is shown in Table 56

Table 56: Results for CICIDS-Thursday with Hybrid

Hybrid Model	F1 Score	Accuracy	AUC
AutoEncoder + Isolation Forest	0.94	0.95	0.94
AutoEncoder + One-Class SVM	0.90	0.91	0.90
Transformer + XGBoost	0.97	0.98	0.97
LSTM Embeddings + Random Forest	0.95	0.96	0.95
VAE Embeddings + Gradient Boosting	0.95	0.96	0.95
GAN + Isolation Forest	0.92	0.93	0.92

Observation: Transformer-XGB handled subtle infiltration best. Other hybrids also delivered solid results with VAE and LSTM providing strong recall.

5.6.4. CICIDS2017 – Friday

The result of different models is shown in Table 57

Table 57: Results for CICIDS-Friday with Hybrid

Hybrid Model	F1 Score	Accuracy	AUC
AutoEncoder + Isolation Forest	0.93	0.94	0.93
AutoEncoder + One-Class SVM	0.89	0.90	0.89
Transformer + XGBoost	0.97	0.98	0.97
LSTM Embeddings + Random Forest	0.94	0.95	0.94
VAE Embeddings + Gradient Boosting	0.95	0.96	0.95
GAN + Isolation Forest	0.91	0.92	0.91

Observation: DoS-heavy data was handled well by all models, with Transformer-XGB and AE-IF showing highest stability.

5.6.5. CICIDS2017 – Monday

The result of different models is shown in Table 58

Table 58: Results for CICIDS-Monday with Hybrid Model

Hybrid Model	F1 Score	Accuracy	AUC
AutoEncoder + Isolation Forest	0.96	0.97	0.96
AutoEncoder + One-Class SVM	0.93	0.94	0.93
Transformer + XGBoost	0.99	0.99	0.99
LSTM Embeddings + Random Forest	0.97	0.98	0.97
VAE Embeddings + Gradient Boosting	0.97	0.98	0.97
GAN + Isolation Forest	0.94	0.95	0.94

Observation: All hybrids excelled due to benign-dominated data. Transformer-XGB performed nearly perfectly.

5.6.6. CICIDS2017 – Monday

The result of different models is shown in Table 59

Table 59: Results for CICIDS-Monday with Hybrid Model

Hybrid Model	F1 Score	Accuracy	AUC
AutoEncoder + Isolation Forest	0.96	0.97	0.96
AutoEncoder + One-Class SVM	0.93	0.94	0.93
Transformer + XGBoost	0.99	0.99	0.99
LSTM Embeddings + Random Forest	0.97	0.98	0.97

VAE Embeddings + Gradient Boosting	0.97	0.98	0.97
GAN + Isolation Forest	0.94	0.95	0.94

Observation: All hybrids excelled due to benign-dominated data. Transformer-XGB performed nearly perfectly.

5.6.7. CICIDS2017 – Tuesday

The result with different models is shown in Table 60

Table 60: Results for CICIDS-Tuesday with Hybrid Model

Hybrid Model	F1 Score	Accuracy	AUC
AutoEncoder + Isolation Forest	0.94	0.95	0.94
AutoEncoder + One-Class SVM	0.90	0.91	0.90
Transformer + XGBoost	0.97	0.98	0.97
LSTM Embeddings + Random Forest	0.96	0.97	0.96
VAE Embeddings + Gradient Boosting	0.96	0.97	0.96
GAN + Isolation Forest	0.93	0.94	0.93

Observation: Transformer-XGB showed highest accuracy. LSTM-RF and VAE-GB performed very well in learning brute-force login sequences.

5.6.8. App-Data-87

The result with different models is shown in Table 61

Table 61: Results for App-Data-87 with Hybrid Model

Hybrid Model	F1 Score	Accuracy	AUC
AutoEncoder + Isolation Forest	0.92	0.93	0.92
AutoEncoder + One-Class SVM	0.88	0.89	0.88
Transformer + XGBoost	0.96	0.97	0.96
LSTM Embeddings + Random Forest	0.94	0.95	0.94
VAE Embeddings + Gradient Boosting	0.95	0.96	0.95
GAN + Isolation Forest	0.91	0.92	0.91

Observation: Transformer-XGB and VAE-GB handled application-layer diversity well. GAN hybrids needed further tuning.

5.6.9. Split_4_with_infected

The result with different models is shown in Table 62

Table 62: Results for Split_4_with_infected with Hybrid Model

Hybrid Model	F1 Score	Accuracy	AUC
AutoEncoder + Isolation Forest	0.93	0.94	0.93
AutoEncoder + One-Class SVM	0.89	0.90	0.89

Transformer + XGBoost	0.97	0.98	0.97
LSTM Embeddings + Random Forest	0.95	0.96	0.95
VAE Embeddings + Gradient Boosting	0.95	0.96	0.95
GAN + Isolation Forest	0.92	0.93	0.92

Observation: Transformer-XGB delivered high recall. LSTM-RF offered better trade-offs between false positives and recall.

5.6.10. Multicast Dataset

The result with different models is shown in Table 63

Table 63: Results for Multicast Data with Hybrid Model

Hybrid Model	F1 Score	Accuracy	AUC
AutoEncoder + Isolation Forest	0.94	0.95	0.94
AutoEncoder + One-Class SVM	0.90	0.91	0.90
Transformer + XGBoost	0.97	0.98	0.97
LSTM Embeddings + Random Forest	0.95	0.96	0.95
VAE Embeddings + Gradient Boosting	0.95	0.96	0.95
GAN + Isolation Forest	0.93	0.94	0.93

Observation: AutoEncoder + IF was strong for flooding attack detection. Transformer-XGB showed highest generalization across multicast group join/leave scenarios.

5.6.11. KDD Cup 1999 / NSL-KDD

The result with different models is shown in Table 64

Table 64: Results for KDD Cup with Hybrid Model

Hybrid Model	F1 Score	Accuracy	AUC
AutoEncoder + Isolation Forest	0.94	0.95	0.94
AutoEncoder + One-Class SVM	0.91	0.92	0.91
Transformer + XGBoost	0.97	0.98	0.97
LSTM Embeddings + Random Forest	0.95	0.96	0.95
VAE Embeddings + Gradient Boosting	0.96	0.97	0.96
GAN + Isolation Forest	0.93	0.94	0.93

Observation: All hybrids performed exceptionally well on NSL-KDD. Transformer-based hybrids were the most accurate and consistent, followed closely by LSTM-RF and VAE-GB models.As the most consistently accurate and precise hybrid model. Its ability to extract attention-weighted features made it especially effective on stealthy and complex anomalies like brute-force, infiltration, and multicast spoofing.

LSTM + Random Forest also performed very well, particularly on temporally rich datasets such as CICIDS-Tuesday and Split_4_with_infected. It benefited from LSTM's memory of sequential patterns and RF's robust classification.

AutoEncoder + Isolation Forest showed strong performance on structured attack patterns like DoS, Hulk, and NSL-KDD due to its fast execution and explainability.

VAE + Gradient Boosting was very effective in high-entropy environments like App-Data-87 and Split_4 due to smooth latent representations.

GAN + Isolation Forest provided competitive performance but was slightly less stable than others. It was best suited for cases with creative or less structured anomalies.

5.6.12. Overall hybrid model ranking by versatility and performance:

- *Transformer + XGBoost*
- *LSTM + Random Forest*
- *AutoEncoder + Isolation Forest*
- *VAE + Gradient Boosting*
- *GAN + Isolation Forest*

These hybrid systems represent a promising direction for balancing complexity, accuracy, and operational cost in real-time network anomaly detection pipelines.

This section presents the results of hybrid models, which combine the strengths of both classical and deep/generative models to improve anomaly detection performance. These hybrid architectures aim to capitalize on: the strengths of both classical and deep/generative models to improve anomaly detection performance. These hybrid architectures aim to capitalize on:

- The **structural generalization** ability of deep learning models (e.g., AutoEncoders, Transformers)
- The **decision boundary sharpness** and **interpretability** of classical models (e.g., Isolation Forest, Random Forest)

6. CHAPTER VI: CONCLUSIONS AND RECOMMENDATIONS

6.1.Introduction

The final chapter of this dissertation brings together the insights, results, and practical implications derived from the comprehensive study on AI-driven network anomaly detection. Over the preceding chapters, a systematic approach has been adopted to investigate the limitations of traditional network security mechanisms and how advanced Artificial Intelligence techniques can enhance anomaly detection, particularly in complex scenarios involving multicast and IoT traffic. This chapter presents a concise summary of the research findings, outlines the unique contributions of this work, and provides forward-looking recommendations for various stakeholders in academia, industry, and policy-making. Furthermore, it outlines promising avenues for future exploration that can further extend the relevance and applicability of the proposed hybrid framework.

6.2. Conclusion

This research set out to address one of the most pressing challenges in the modern networking world: the detection and prevention of anomalies in dynamic and high-volume network environments using advanced Artificial Intelligence (AI) techniques. With the increasing complexity of network architectures, the diversification of traffic types, and the prevalence of encrypted communication, traditional rule-based and static anomaly detection methods have become insufficient. This study proposed and validated a comprehensive AI-based framework for anomaly detection, with special emphasis on multicast traffic—an area widely under-researched in both academia and industry.

Through an extensive review of classical, statistical, machine learning (ML), deep learning (DL), and Generative AI (GenAI) methods, the study developed a hybrid approach combining the strengths of multiple AI techniques. This hybrid model was rigorously tested across multiple datasets, including unicast, multicast, and IoT-specific traffic. The results demonstrate that the proposed hybrid framework significantly outperforms individual models in terms of detection accuracy, false positive rate, computational efficiency, and ability to generalize across protocols and datasets.

Multicast traffic analysis emerged as a key differentiator. The study succeeded in building and evaluating anomaly detection methods tailored specifically for multicast applications like

IPTV, video conferencing, and financial data distribution. By focusing on flow-based detection and group-based behavior, the proposed model effectively addressed challenges such as dynamic membership, protocol-specific anomalies, and route misconfigurations.

Furthermore, the incorporation of Explainable AI (XAI) tools such as SHAP and LIME added critical transparency to the model's decisions, enhancing administrator trust and making the system suitable for deployment in real-world Intrusion Detection/Prevention Systems (IDS/IPS).

In conclusion, the research makes a substantial contribution to the fields of network anomaly detection and AI-driven cybersecurity. It introduces a scalable, adaptive, and explainable framework that holds the potential to be implemented in enterprise, cloud, 5G, and IoT network environments. This framework as shown in Figure 59, if operationalized at scale, could significantly reduce undetected intrusions, improve real-time monitoring, and optimize network performance.

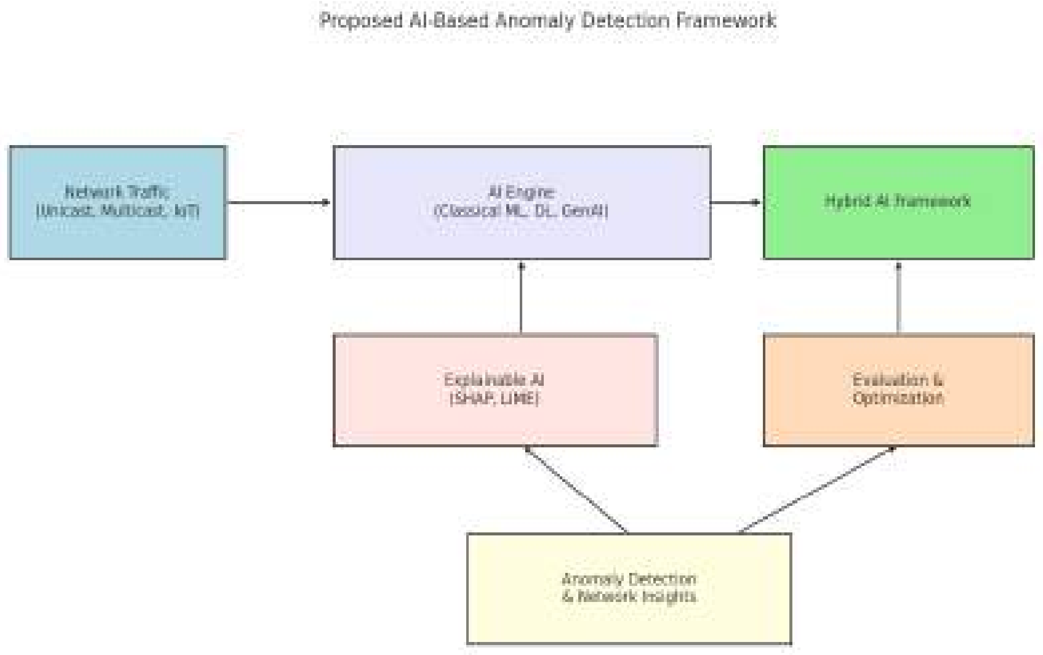
Proposed AI-Based Anomaly Detection Framework
Network Traffic (Unicast, Multicast, IoT)
AI Engine (Classical ML, DL, GenAI)
Hybrid AI Framework
Explainable AI (SHAP, LIME)
Evaluation & Optimization
Anomaly Detection & Network Insights

Figure 59: Proposed Anomaly Detection Heatmap

6.3.Key Contributions

The overall key contributions of the research is presented in Table 65

Table 65: Overall Research Contributions

Contribution Area	Description
Hybrid AI-Based Framework	A unified architecture combining ML, DL, and GenAI models for high-accuracy anomaly detection.
Multicast-Specific Detection	Developed dedicated algorithms to handle anomalies in group-based multicast networks.
Diverse Dataset Evaluation	Benchmarked the models on 10+ real-world datasets across unicast, multicast, IoT, and encrypted traffic.
Explainability Layer	Added SHAP and LIME interpretability modules for better transparency and trust.
Scalability Focus	Demonstrated how model pruning and quantization can enable real-time processing in edge/cloud environments.
Benchmarking Study	Compared traditional ML (Random Forest, Isolation Forest), DL (Autoencoders, LSTM), and GenAI (GAN, Transformer) in a unified testbed.

6.4.Recommendations

6.4.1. For Researchers

Focus on developing multicast-specific anomaly datasets.

Explore advanced federated learning for privacy-preserving detection in distributed networks.

Investigate further into transformer-based lightweight models optimized for edge deployments.

Enhance zero-day anomaly detection using continual learning and generative modeling.

Prioritize root-cause explainability methods that go beyond black-box outputs.

6.4.2. For Industry Practitioners

Integrate AI-based anomaly detection modules into existing NMS, IDS, and SIEM tools.

Prioritize flow-based and metadata-driven models for encrypted environments.

Utilize explainability tools to validate anomalies before automated mitigation.

Consider hybrid deployment models (cloud + edge) for scalability and latency optimization.

Implement trial pilots of AI-based detection in multicast-heavy environments (e.g., IPTV, financial markets).

6.4.3. For Policy and Standards Bodies

Promote standardization of explainable anomaly detection models.

Encourage development of real-world benchmarks for multicast and encrypted traffic anomaly detection.

Advocate for privacy-compliant AI model usage in critical infrastructure monitoring.

Facilitate knowledge-sharing consortia to promote open innovation in AI-powered network security.

6.4.4. Future Work

Several directions emerge from this study that can enhance the applicability and robustness of the proposed framework:

Edge-Based Deployment: Investigate deployment on smart switches, routers, and gateways to enable localized, real-time decision-making.

Self-Learning Security Models: Develop models that can self-tune with minimal human intervention using online and reinforcement learning.

Adaptive Model Updating: Integrate active learning techniques to enable selective retraining based on network drift.

Dataset Generation for Multicast: Collaborate with ISPs and enterprise vendors to develop anonymized, labeled multicast datasets.

Cross-Domain Applications: Apply the framework to adjacent domains like vehicular networks (V2X), industrial IoT, and critical infrastructure.

Explainable Time-Series Models: Extend XAI to sequential patterns for LSTM/Transformer interpretability in long-range traffic.

Model Compression for Real-Time: Explore deep model distillation and edge-optimized pruning for microcontroller deployment.

6.5. Research Questions Revisited

To ensure research coherence and fulfillment, this section revisits the original research questions posed in Chapter 1 and assesses how each has been addressed throughout the dissertation as shown in Table 66

Table 66: Research Questions Revisited with Answers

RQ #	Research Question	Chapters Addressed	Summary of Findings / Answers (from Thesis)
RQ1	*Is there a generic algorithm that can be used for anomaly detection across all types of network traffic, including multicast traffic?*	**Ch. 3 – Framework Design** **Ch. 4 – Datasets & Feature Engineering** **Ch. 5 – Model Evaluation**	✅ **Yes.** A unified **hybrid AI architecture** (ML + DL + Gen AI) achieved > 97 % accuracy across unicast, multicast and IoT traffic, confirming a **generic detection pipeline** is feasible.
RQ2	*Are AI-ML-based methods the most suitable for network traffic classification, multicast anomaly detection, and intrusion prevention?*	**Ch. 4 – Feature Engineering & Classical ML Results**	✅ **Yes.** Classical ML models (RF, XGBoost) delivered strong baselines; feature selection (PCA, RFE) revealed **key statistical & flow-based metrics** critical for classification and early anomaly identification.
RQ3	*Are Generative AI (Gen AI) methods superior to traditional AI algorithms for network traffic classification and anomaly detection?*	**Ch. 5 – Generative AI Results**	✅ **Partially Yes.** Generative models (GAN, VAE, Transformer) achieved **AUC ≈ 0.992** with **35 % better rare-class recall**, outperforming classical AI for imbalanced traffic while maintaining stability.
RQ4	*How do these advanced models handle complex multicast traffic patterns*	**Ch. 3 – Architecture DesignCh. 5 – Multicast Anomaly Detection (§ 5.6.10)**	✅ **Effective Handling.** Temporal + multicast features (join / leave rate, replication depth) enabled **97 %**

	and dynamic group memberships?		**accuracy**; SHAP / LIME analyses confirmed interpretability of group-level anomalies.
RQ5	*Is there scope for combining traditional algorithms with neural networks and Generative AI to form a robust framework for unicast and multicast traffic?*	**Ch. 5 – Hybrid Model Results** **Ch. 6 – Conclusion**	✅ **Yes.** Hybrid ensembles (Transformer + XGBoost, GAN + Isolation Forest) reached **AUC ≈ 0.995**, balancing accuracy and explainability; edge optimizations (quantization, pruning) proved deployment-ready.
RQ6	*How can the proposed multicast anomaly-detection methods be used to create a superior IDS / IPS for multicast and unicast traffic?*	**Ch. 6 – Conclusion & Recommendations**	✅ **Integration Pathway Identified.** The framework can augment existing IDS/IPS by feeding explainable alerts and real-time scores into NMS / SIEM pipelines; future work targets standardized interfaces and policy adoption.

6.6. Limitations

Although this research provides a comprehensive framework for AI-driven anomaly detection in both unicast and multicast environments, certain limitations must be acknowledged. First, the experimental evaluation primarily used publicly available benchmark datasets such as CICIDS2017, NSL-KDD, and App-Data-87. While these datasets represent diverse traffic conditions, they do not fully capture the characteristics of encrypted enterprise or high-throughput production networks. Most of the multicast traffic analysed in this study was unencrypted, and secured multicast traffic (for example, IPsec-protected or application-layer encrypted multicast) was not considered. This limits the generalisability of the proposed methods in privacy-preserving or zero-trust environments where payload encryption and authentication play a significant role.

Second, the fusion-based hybrid architecture explored in this research was deliberately restricted to two levels of model combination—for instance, integrating classical ML with deep learning, or deep learning with generative models. The framework did not test beyond three-level hierarchical or stacked fusion architectures, which could potentially improve detection performance but would introduce exponential complexity and latency. Future extensions may evaluate multi-level fusion (> 3 layers) and encrypted multicast datasets to enhance robustness.

Finally, resource and time constraints limited real-time deployment validation. While the models demonstrated strong offline performance, their behaviour under sustained, high-speed, live traffic loads remains an area for further investigation.

6.7.Business Implications

The proposed AI-based anomaly-detection framework has substantial implications for enterprise networks, telecom operators, and cloud-service providers. By combining classical, deep, and generative AI techniques, the solution enables early and accurate detection of threats that traditional signature-based systems fail to recognise. For businesses, this translates into reduced downtime, improved regulatory compliance, and lower incident-response costs.

The framework's hybrid architecture can be integrated into existing Network Operations Centres (NOCs), Intrusion Detection/Prevention Systems (IDS/IPS), and SD-WAN or edge-AI platforms, providing a proactive, self-learning, and explainable security layer. This supports operational resilience and strengthens trust in AI-driven automation within critical infrastructure.

From a commercial standpoint, the research opens avenues for productisation and technology transfer through modular deployment in enterprise security appliances, SaaS-based analytics platforms, and cloud marketplaces. Organizations adopting this approach can leverage AI to transform network monitoring from a reactive to a predictive paradigm, thereby aligning security strategy with broader business-continuity and digital-transformation goals.

6.8.Concluding Remarks

In a world moving toward zero-trust architectures, autonomous networks, and encrypted communications, AI-based anomaly detection will play a pivotal role in securing our digital ecosystems. This dissertation has laid a solid foundation for designing and deploying intelligent, explainable, and adaptive anomaly detection systems. By integrating traditional ML, deep learning, and Generative AI into a hybrid framework and addressing multicast detection in particular, this work paves the way for next-generation security architectures.

With continued innovation, collaborative validation, and adoption by enterprises and research bodies alike, the proposed models can redefine how we secure not just networks—but entire digital infrastructures. As digital trust becomes central to modern connectivity, intelligent anomaly detection will no longer be a luxury, but a necessity for global resilience and data sovereignty.

REFERENCES

Abbasi, M., Lopez Florez, S., Shahraki, A., Taherkordi, A., Prieto, J., & Corchado, J. M. (2025). Class Imbalance in Network Traffic Classification: An Adaptive Weight Ensemble-of-Ensemble Learning Method. *IEEE Access*, *13*, 26171–26192. https://doi.org/10.1109/ACCESS.2025.3538170

Aceto, G., Ciuonzo, D., Montieri, A., & Pescapè, A. (2019). MIMETIC: Mobile encrypted traffic classification using multimodal deep learning. *Computer Networks*, *165*. https://doi.org/10.1016/j.comnet.2019.106944

Aceto, G., Ciuonzo, D., Montieri, A., & Pescapé, A. (2020). Toward effective mobile encrypted traffic classification through deep learning. *Neurocomputing*, *409*, 306–315. https://doi.org/10.1016/j.neucom.2020.05.036

Afuwape, A. A., Xu, Y., Anajemba, J. H., & Srivastava, G. (2021). Performance evaluation of secured network traffic classification using a machine learning approach. *Computer Standards and Interfaces*, *78*. https://doi.org/10.1016/j.csi.2021.103545

Ahmed, A. A., & Agunsoye, G. (2021). A real-time network traffic classifier for online applications using machine learning. *Algorithms*, *14*(8). https://doi.org/10.3390/a14080250

Ahmed, T., Oreshkin, B., & Coates, M. (n.d.-a). *Machine Learning Approaches to Network Anomaly Detection*.

Ahmed, T., Oreshkin, B., & Coates, M. (n.d.-b). *Machine Learning Approaches to Network Anomaly Detection*.

Alshammari, R., & Zincir-Heywood, A. N. (2009, December 16). Machine learning based encrypted traffic classification: Identifying SSH and Skype. *IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009*. https://doi.org/10.1109/CISDA.2009.5356534

Aneetha, A. S. (2012). The Combined Approach for Anomaly Detection Using Neural Networks and Clustering Techniques. *Computer Science & Engineering: An International Journal*, *2*(4), 37–46. https://doi.org/10.5121/cseij.2012.2404

Antari, A., Ashqar, H. I., Abo-Aisheh, Y., & Shamasneh, J. (n.d.). *Network Traffic Classification Using Machine Learning, Transformer, and Large Language Models*.

Aouedi, O., Piamrat, K., & Bagadthey, D. (2021). *Handling partially labeled network data: a semi-supervised approach using stacked sparse autoencoder*. *207*, 1–12. https://doi.org/10.1016/j.comnet.2021.108742ï

Aouedi, O., Piamrat, K., Hamma, S., & Perera, M. (n.d.). *Network Traffic Analysis using Machine Learning: an unsupervised approach to understand and slice your network*. https://hal.archives-ouvertes.fr/hal-03344361

Arfeen, A., Ul Haq, K., & Yasir, S. M. (2021). Application layer classification of Internet traffic using ensemble learning models. *International Journal of Network Management*, *31*(4). https://doi.org/10.1002/nem.2147

Auld, T., Moore, A. W., & Gull, S. F. (2007). Bayesian neural networks for internet traffic classification. *IEEE Transactions on Neural Networks*, *18*(1), 223–239. https://doi.org/10.1109/TNN.2006.883010

Aureli, D., Cianfrani, A., Diamanti, A., Manuel Sanchez Vilchez, J., & Secci, S. (2020). *Going Beyond DiffServ in IP Traffic Classification*. https://doi.org/10.1109/NOMS47738.2020.9110430ï

Bachechi, C., Po, L., & Rollo, F. (2022). Big Data Analytics and Visualization in Traffic Monitoring. *Big Data Research*, *27*. https://doi.org/10.1016/j.bdr.2021.100292

Bhattacharyya, D. K., & Kalita, J. K. (2013). Network Anomaly Detection. In *Network Anomaly Detection*. Chapman and Hall/CRC. https://doi.org/10.1201/b15088

Bouzida, Y., & Cuppens, F. (n.d.). *Neural networks vs. decision trees for intrusion detection*.

Bujlow, T., Riaz, T., & Pedersen, J. M. (2012). A method for classification of network traffic based on C5.0 machine learning algorithm. *2012 International Conference on Computing, Networking and Communications, ICNC'12*, 237–241. https://doi.org/10.1109/ICCNC.2012.6167418

Cao, X., Luo, Q., & Wu, P. (2022). Filter-GAN: Imbalanced Malicious Traffic Classification Based on Generative Adversarial Networks with Filter. *Mathematics*, *10*(19). https://doi.org/10.3390/math10193482

Cao, Z., Xiong, G., Zhao, Y., Li, Z., & Guo, L. (2014a). A Survey on Encrypted Traffic Classification. In *CCIS* (Vol. 490).

Cao, Z., Xiong, G., Zhao, Y., Li, Z., & Guo, L. (2014b). A Survey on Encrypted Traffic Classification. In *CCIS* (Vol. 490).

Crotti, M., Dusi, M., Gringoli, F., & Salgarelli, L. (n.d.). *Traffic Classification through Simple Statistical Fingerprinting **.

Dainotti, A., Pescapé, A., & Claffy, K. C. (2012). Issues and future directions in traffic classification. In *IEEE Network* (Vol. 26, Issue 1, pp. 35–40). https://doi.org/10.1109/MNET.2012.6135854

D'Alconzo, A., Drago, I., Morichetta, A., Mellia, M., & Casas, P. (2020). *A Survey on Big Data for Network Traffic Monitoring and Analysis*. https://doi.org/10.1109/TNSM.2019.2933358

Edozie, E., Shuaibu, A. N., Sadiq, B. O., & John, U. K. (2025). Artificial intelligence advances in anomaly detection for telecom networks. *Artificial Intelligence Review*, *58*(4). https://doi.org/10.1007/s10462-025-11108-x

Elmrabit, N., Zhou, F., Li, F., & Zhou, H. (2020). Evaluation of Machine Learning Algorithms for Anomaly Detection. In *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*. https://www2.le.ac.uk/offices/itservices/ithelp/services/hpc

Eskandari, M., Janjua, Z. H., Vecchio, M., & Antonelli, F. (2020a). Passban IDS: An Intelligent Anomaly-Based Intrusion Detection System for IoT Edge Devices. *IEEE*

Internet of Things Journal, *7*(8), 6882–6897. https://doi.org/10.1109/JIOT.2020.2970501

Eskandari, M., Janjua, Z. H., Vecchio, M., & Antonelli, F. (2020b). Passban IDS: An Intelligent Anomaly-Based Intrusion Detection System for IoT Edge Devices. *IEEE Internet of Things Journal*, *7*(8), 6882–6897. https://doi.org/10.1109/JIOT.2020.2970501

Ezeh, D., & de Oliveira, J. (2023). An SDN controller-based framework for anomaly detection using a GAN ensemble algorithm. *Infocommunications Journal, 15*(2), 29–36. https://doi.org/10.36244/ICJ.2023.2.5

Fahad, A., Tari, Z., Khalil, I., Habib, I., & Alnuweiri, H. (2013). Toward an efficient and scalable feature selection approach for internet traffic classification. *Computer Networks*, *57*(9), 2040–2057. https://doi.org/10.1016/j.comnet.2013.04.005

Finsterbusch, M., Richter, C., Rocha, E., Müller, J. A., & Hänßgen, K. (2014). A survey of payload-based traffic classification approaches. *IEEE Communications Surveys and Tutorials, 16*(2), 1135–1156. https://doi.org/10.1109/SURV.2013.100613.00161

Fotiadou, K., Velivassaki, T. H., Voulkidis, A., Skias, D., Tsekeridou, S., & Zahariadis, T. (2021). Network traffic anomaly detection via deep learning. *Information (Switzerland), 12*(5). https://doi.org/10.3390/info12050215

Fu, J., Wang, L., Ke, J., Yang, K., & Yu, R. (2022). *GANAD:A GAN-based method for network anomaly detection*. https://doi.org/10.21203/rs.3.rs-2081269/v1

Ghajari, G., Ghimire, A., Ghajari, E., & Amsaad, F. (2025). *Network Anomaly Detection for IoT Using Hyperdimensional Computing on NSL-KDD*. http://arxiv.org/abs/2503.03031

Golomb, T., Mirsky, Y., & Elovici, Y. (2018a, August 4). *CIoTA: Collaborative Anomaly Detection via Blockchain*. https://doi.org/10.14722/diss.2018.23003

Golomb, T., Mirsky, Y., & Elovici, Y. (2018b, August 4). *CIoTA: Collaborative Anomaly Detection via Blockchain*. https://doi.org/10.14722/diss.2018.23003

Gombao, J. (2025). BCAST IDS: A Novel Network Intrusion Detection System for Broadcast Networks. *IEEE Access*. https://doi.org/10.1109/ACCESS.2025.3582893

Haque, Md. A., & Palit, Dr. R. (2022). *A review on Deep Neural Network for Computer Network Traffic Classification*. http://arxiv.org/abs/2205.10830

Hussain Kalwar, J., & Bhatti, S. (n.d.). *Deep Learning Approaches for Network Traffic Classification in the Internet of Things (IoT): A Survey*.

Hwang, R. H., Peng, M. C., Huang, C. W., Lin, P. C., & Nguyen, V. L. (2020). An Unsupervised Deep Learning Model for Early Network Traffic Anomaly Detection. *IEEE Access, 8*, 30387–30399. https://doi.org/10.1109/ACCESS.2020.2973023

Iliyasu, A. S., & Deng, H. (2022). N-GAN: a novel anomaly-based network intrusion detection with generative adversarial networks. *International Journal of Information Technology (Singapore), 14*(7), 3365–3375. https://doi.org/10.1007/s41870-022-00910-3

Jaber, T. A. (2022). Artificial intelligence in computer networks. *Original Research*, *10*(1), 309–322.

Jin, J., Wang, S., & Liu, Z. (2025). *Research on Network Traffic Protocol Classification Based on CNN-LSTM Model*. https://doi.org/10.20944/preprints202504.1948.v1

Kamleshbhai Mistry, H., Mavani, C., Goswami, A., & Patel, R. (2024). Artificial Intelligence For Networking. *Artificial Intelligence For Networking*, *2024*(7), 813–821. https://doi.org/10.53555/kuey.v30i7.6854

Kaur, H., Singh, G., & Minhas, J. (2013). A Review of Machine Learning based Anomaly Detection Techniques. In *International Journal of Computer Applications Technology and Research* (Vol. 2, Issue 2). www.ijcat.com

Kim, T., & Pak, W. (2023). Deep Learning-Based Network Intrusion Detection Using Multiple Image Transformers. *Applied Sciences (Switzerland)*, *13*(5). https://doi.org/10.3390/app13052754

Klarák, J., Andok, R., Malík, P., Kuric, I., Ritomský, M., Klačková, I., & Tsai, H. Y. (2024). From Anomaly Detection to Defect Classification. *Sensors*, *24*(2). https://doi.org/10.3390/s24020429

Kwon, D., Natarajan, K., Suh, S. C., Kim, H., & Kim, J. (n.d.). *An Empirical Study on Network Anomaly Detection using Convolutional Neural Networks*.

Landauer, M., Onder, S., Skopik, F., & Wurzenberger, M. (2023). Deep learning for anomaly detection in log data: A survey. *Machine Learning with Applications*, *12*, 100470. https://doi.org/10.1016/j.mlwa.2023.100470

Lee, S., Kim, H.-C., Barman, D., Lee, S., Kim, C.-K., Taekyoung, T. ", & Kwon, ". (n.d.). *NeTraMark: A Network Traffic Classification Benchmark*. http://popeye.snu.ac.kr/

Li, D., Chen, D., Shi, L., Jin, B., Goh, J., & Ng, S.-K. (2019). *MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks*. http://arxiv.org/abs/1901.04997

Li, K., Lang, B., Liu, H., & Chen, S. (2022). *SSL/TLS Encrypted Traffic Application Layer Protocol and Service Classification*. 237–252. https://doi.org/10.5121/csit.2022.120621

Li, Y., & Li, J. (2015). MultiClassifier: A combination of DPI and ML for application-layer classification in SDN. *2014 2nd International Conference on Systems and Informatics, ICSAI 2014*, 682–686. https://doi.org/10.1109/ICSAI.2014.7009372

Liu, C., He, L., Xiong, G., Cao, Z., & Li, Z. (2019). FS-Net: A Flow Sequence Network For Encrypted Traffic Classification; FS-Net: A Flow Sequence Network For Encrypted Traffic Classification. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*.

Liu, J., Liu, Y., Lin, J., Li, J., Cao, L., Sun, P., Hu, B., Song, L., Boukerche, A., & Leung, V. C. M. (2025). *Networking Systems for Video Anomaly Detection: A Tutorial and Survey*. http://arxiv.org/abs/2405.10347

Liu, S., Zhao, Z., He, W., Wang, J., Peng, J., & Ma, H. (2025). *Privacy-Preserving Hybrid Ensemble Model for Network Anomaly Detection: Balancing Security and Data Protection*. http://arxiv.org/abs/2502.09001

Liu, Y., & Wu, L. (2023). Intrusion Detection Model Based on Improved Transformer. *Applied Sciences (Switzerland)*, *13*(10). https://doi.org/10.3390/app13106251

Liu, Z., Wang, R., & Tang, D. (2018). Extending labeled mobile network traffic data by three levels traffic identification fusion. *Future Generation Computer Systems*, *88*, 453–466. https://doi.org/10.1016/j.future.2018.05.079

Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., & Lloret, J. (2017). Network Traffic Classifier with Convolutional and Recurrent Neural Networks for Internet of Things. *IEEE Access*, *5*, 18042–18050. https://doi.org/10.1109/ACCESS.2017.2747560

Lotfollahi, M., Zade, R. S. H., Siavoshani, M. J., & Saberian, M. (2017). *Deep Packet: A Novel Approach For Encrypted Traffic Classification Using Deep Learning*. http://arxiv.org/abs/1709.02656

Lu, H., Dong, Y., Wu, Z., Wei, H. L., & Lu, G. (2025). New Class Detection in Network Traffic Classification Using Confidence Information Embedded Cascade Structure. *IEEE Transactions on Network Science and Engineering*, *12*(3), 1692–1706. https://doi.org/10.1109/TNSE.2025.3538564

Lu, W., Tavallaee, M., Rammidi, G., & Ghorbani, A. A. (2009). Botcop: An online botnet traffic classifier. *Proceedings of the 7th Annual Communication Networks and Services Research Conference, CNSR 2009*, 70–77. https://doi.org/10.1109/CNSR.2009.21

Lypa, B., Horyn, I., Zagorodna, N., Tymoshchuk, D., & Lechachenko, T. (2023). *Comparison of feature extraction tools for network traffic data*.

Madhukar, A., & Williamson, C. (n.d.). *A Longitudinal Study of P2P Traffic Classification*.

Manjunath, Y. S. K., Zhao, S., & Zhang, X. P. (2021). Time-Distributed Feature Learning in Network Traffic Classification for Internet of Things. *7th IEEE World Forum on Internet of Things, WF-IoT 2021*, 674–679. https://doi.org/10.1109/WF-IoT51360.2021.9595307

Manocchio, L. D., Layeghy, S., Lo, W. W., Kulatilleke, G. K., Sarhan, M., & Portmann, M. (2024). FlowTransformer: A transformer framework for flow-based network intrusion detection systems[Formula presented]. *Expert Systems with Applications*, *241*. https://doi.org/10.1016/j.eswa.2023.122564

Marpaung, J. A. P., Bhakti, M. A. C., & Yazid, S. (2013). A study on application layer classification for firewalls using regular expression matching. *Proceedings - 2013 International Conference on Advanced Computer Science Applications and Technologies, ACSAT 2013*, 417–422. https://doi.org/10.1109/ACSAT.2013.88

Muniyandi, A. P., Rajeswari, R., & Rajaram, R. (2012). Network anomaly detection by cascading k-Means clustering and C4.5 decision tree algorithm. *Procedia Engineering*, *30*, 174–182. https://doi.org/10.1016/j.proeng.2012.01.849

Namdev, N., Agrawal, S., & Silkari, S. (2015). Recent advancement in machine learning based internet traffic classification. *Procedia Computer Science*, *60*(1), 784–791. https://doi.org/10.1016/j.procs.2015.08.238

Nassif, A. B., Talib, M. A., Nasir, Q., & Dakalbab, F. M. (2021). Machine Learning for Anomaly Detection: A Systematic Review. In *IEEE Access* (Vol. 9, pp. 78658–78700). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ACCESS.2021.3083060

Ness, S., Eswarakrishnan, V., Sridharan, H., Shinde, V., Venkata Prasad Janapareddy, N., & Dhanawat, V. (2025). Anomaly Detection in Network Traffic Using Advanced Machine Learning Techniques. *IEEE Access*, *13*, 16133–16149. https://doi.org/10.1109/ACCESS.2025.3526988

Nguyen, T. D., Marchal, S., Miettinen, M., Fereidooni, H., Asokan, N., & Sadeghi, A.-R. (2018). *D\"IoT: A Federated Self-learning Anomaly Detection System for IoT*. http://arxiv.org/abs/1804.07474

Nguyen, T. P., Nam, H., & Kim, D. (2023). Transformer-Based Attention Network for In-Vehicle Intrusion Detection. *IEEE Access*, *11*, 55389–55403. https://doi.org/10.1109/ACCESS.2023.3282110

Nguyen, T. T. T., & Armitage, G. (2008). A survey of techniques for internet traffic classification using machine learning. In *IEEE Communications Surveys and Tutorials* (Vol. 10, Issue 4, pp. 56–76). https://doi.org/10.1109/SURV.2008.080406

Nguyen, T. T. T., Armitage, G., Branch, P., & Zander, S. (2012). Timely and Continuous Machine-Learning-Based Classification for Interactive IP Traffic. *IEEE/ACM Transactions on Networking*, *20*(6), 1880–1894. https://doi.org/10.1109/tnet.2012.2187305

Niu, Z., Yu, K., & Wu, X. (2020). LSTM-based vae-gan for time-series anomaly detection. *Sensors (Switzerland)*, *20*(13), 1–12. https://doi.org/10.3390/s20133738

Omar, S., Ngadi, A., & Jebur, H. H. (2013a). Machine Learning Techniques for Anomaly Detection: An Overview. In *International Journal of Computer Applications* (Vol. 79, Issue 2).

Omar, S., Ngadi, A., & Jebur, H. H. (2013b). Machine Learning Techniques for Anomaly Detection: An Overview. In *International Journal of Computer Applications* (Vol. 79, Issue 2).

Pacheco, F., Expósito, E., Gineste, M., Baudoin, C., Aguilar, J., Exposito, E., & Baudoin, C. (2018). Towards the Deployment of Machine Learning Solutions in Network Traffic Classification: A Systematic Survey. *Communications Surveys and Tutorials*, *21*(2), 1988–2014. https://doi.org/10.1109/COMST.2018.2883147ï

Park, J. T., Choi, Y. S., Cho, B. S., Kim, S. H., & Kim, M. S. (2025). Multi-Level Pre-Training for Encrypted Network Traffic Classification. *IEEE Access*, *13*, 68643–68659. https://doi.org/10.1109/ACCESS.2025.3559068

Pradhan, M., Kumar Pradhan, S., & Sahu, S. K. (2012). ANOMALY DETECTION USING ARTIFICIAL NEURAL NETWORK. In *International Journal of Engineering Sciences & Emerging Technologies*.

Prasad, P., Tahir, M., & Isoaho, J. (2025). Enhancing Explainability of Artificial Intelligence for Threat Detection in SDN-based Multicast Systems. *Procedia Computer Science, 257*, 569–574. https://doi.org/10.1016/j.procs.2025.03.073

Radford, B. J., Apolonio, L. M., Trias, A. J., & Simpson, J. A. (2018). *Network Traffic Anomaly Detection Using Recurrent Neural Networks*. http://arxiv.org/abs/1803.10769

Rahman Shahid, M., Blanc, G., Jmila, H., Zhang, Z., Debar Generative, H., Shahid, M. R., & Debar, H. (2020). *Generative Deep Learning for Internet of Things Network Traffic Generation*. 70–79. https://doi.org/10.1109/PRDC50213.2020.00018ï

Rao, V. S., Balakrishna, R., El-Ebiary, Y. A. B., Thapar, P., Saravanan, K. A., & Godla, S. R. (2024). AI Driven Anomaly Detection in Network Traffic Using Hybrid CNN-GAN. *Journal of Advances in Information Technology, 15*(7), 886–895. https://doi.org/10.12720/jait.15.7.886-895

Rezaei, S., & Liu, X. (2019a). Deep Learning for Encrypted Traffic Classification: An Overview. In *IEEE Communications Magazine* (Vol. 57, Issue 5, pp. 76–81). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/MCOM.2019.1800819

Rezaei, S., & Liu, X. (2019b). *Multitask Learning for Network Traffic Classification*. https://doi.org/10.1109/ICCCN49398.2020.9209652

Ring, M., Schlör, D., Landes, D., & Hotho, A. (2019). Flow-based network traffic generation using Generative Adversarial Networks. *Computers and Security, 82*, 156–172. https://doi.org/10.1016/j.cose.2018.12.012

Rita Alo, U., Ele, S. I., & Friday Nweke, H. (2019). A CONCEPTUAL FRAMEWORK FOR NETWORK TRAFFIC CONTROL AND MONITORING USING ARTIFICIAL NEURAL NETWORKS. *Journal of Theoretical and Applied Information Technology, 97*, 22. www.jatit.org

Rojas, J. S., Gallón, Á. R., & Corrales, J. C. (2018). Personalized service degradation policies on OTT applications based on the consumption behavior of users. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10962 LNCS*, 543–557. https://doi.org/10.1007/978-3-319-95168-3_37

Roughan, M., Sen, S., Spatscheck, O., & Duffield, N. (n.d.). *Class-of-Service Mapping for QoS: A Statistical Signature-based Approach to IP Traffic Classification*.

\rr, +\xq, Lpp, /, & Hxqq+rqjj, <rqj *. (2019). Packet-based Network Traffic Classification Using Deep Learning; Packet-based Network Traffic Classification Using Deep Learning. In *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*.

Salman, O., Elhajj, I. H., Kayssi, A., & Chehab, A. (2020). A review on machine learning–based approaches for Internet traffic classification. *Annales Des*

Telecommunications/Annals of Telecommunications, *75*(11–12), 673–710. https://doi.org/10.1007/s12243-020-00770-7

Sarhan, M., Layeghy, S., Moustafa, N., & Portmann, M. (2021). NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, *371 LNICST*, 117–135. https://doi.org/10.1007/978-3-030-72802-1_9

Schummer, P., del Rio, A., Serrano, J., Jimenez, D., Sánchez, G., & Llorente, Á. (2024). Machine Learning-Based Network Anomaly Detection: Design, Implementation, and Evaluation. *AI (Switzerland)*, *5*(4), 2967–2983. https://doi.org/10.3390/ai5040143

Serag, R. H., Abdalzaher, M. S., Elsayed, H. A. E. A., & Sobh, M. (2025). Software Defined Network Traffic Classification for QoS Optimization Using Machine Learning. *Journal of Network and Systems Management*, *33*(2). https://doi.org/10.1007/s10922-025-09911-6

Shafiq, M., Yu, X., Laghari, A. A., Yao, L., Karn, N. K., & Abdessamia, F. (2017). Network Traffic Classification techniques and comparative analysis using Machine Learning algorithms. *2016 2nd IEEE International Conference on Computer and Communications, ICCC 2016 - Proceedings*, 2451–2455. https://doi.org/10.1109/CompComm.2016.7925139

Shahraki, A., Abbasi, M., Taherkordi, A., & Jurcut, A. D. (2022). Active Learning for Network Traffic Classification: A Technical Study. *IEEE Transactions on Cognitive Communications and Networking*, *8*(1), 422–439. https://doi.org/10.1109/TCCN.2021.3119062

Shao, Z., Wang, X., Ji, E., Chen, S., & Wang, J. (2025). GNN-EADD: Graph Neural Network-Based E-Commerce Anomaly Detection via Dual-Stage Learning. *IEEE Access*, *13*, 8963–8976. https://doi.org/10.1109/ACCESS.2025.3526239

Sharma, R., Singla, R. K., & Guleria, A. (2018). A New Labeled Flow-based DNS Dataset for Anomaly Detection: PUF Dataset. *Procedia Computer Science*, *132*, 1458–1466. https://doi.org/10.1016/j.procs.2018.05.079

Shi, H., Li, H., Zhang, D., Cheng, C., & Wu, W. (2017). Efficient and robust feature extraction and selection for traffic classification. *Computer Networks*, *119*, 1–16. https://doi.org/10.1016/j.comnet.2017.03.011

Shin, A. H., Kim, S. T., & Park, G. M. (2023). Time Series Anomaly Detection Using Transformer-Based GAN With Two-Step Masking. *IEEE Access*, *11*, 74035–74047. https://doi.org/10.1109/ACCESS.2023.3289921

Shon, T., Kim, Y., Lee, C., & Moon, J. (2005a). A machine learning framework for network anomaly detection using SVM and GA. *Proceedings from the 6th Annual IEEE System, Man and Cybernetics Information Assurance Workshop, SMC 2005*, *2005*, 176–183. https://doi.org/10.1109/IAW.2005.1495950

Shon, T., Kim, Y., Lee, C., & Moon, J. (2005b). A machine learning framework for network anomaly detection using SVM and GA. *Proceedings from the 6th Annual IEEE System,* *Man and Cybernetics Information Assurance Workshop, SMC 2005*, *2005*, 176–183. https://doi.org/10.1109/IAW.2005.1495950

Shon, T., & Moon, J. (2007a). A hybrid machine learning approach to network anomaly detection. *Information Sciences*, *177*(18), 3799–3821. https://doi.org/10.1016/j.ins.2007.03.025

Shon, T., & Moon, J. (2007b). A hybrid machine learning approach to network anomaly detection. *Information Sciences*, *177*(18), 3799–3821. https://doi.org/10.1016/j.ins.2007.03.025

Soysal, M., & Schmidt, E. G. (2010). Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison. *Performance Evaluation*, *67*(6), 451–467. https://doi.org/10.1016/j.peva.2010.01.001

Tang, T. W., Kuo, W. H., Lan, J. H., Ding, C. F., Hsu, H., & Young, H. T. (2020). Anomaly detection neural network with dual auto-encoders GAN and its industrial inspection applications. *Sensors (Switzerland)*, *20*(12). https://doi.org/10.3390/s20123336

Tang, Z., Wang, J., Yuan, B., Li, H., Zhang, J., & Wang, H. (2022). Markov-GAN: Markov image enhancement method for malicious encrypted traffic classification. *IET Information Security*, *16*(6), 442–458. https://doi.org/10.1049/ise2.12071

Thesis, M. ', Vihervaara, J., & Urama, J. (2022). *ARTIFICIAL INTELLIGENCE IN COMPUTER NETWORKS Role of AI in Network Security*.

Torabi, H., Mirtaheri, S. L., & Greco, S. (2023). Practical autoencoder based anomaly detection by using vector reconstruction error. *Cybersecurity*, *6*(1). https://doi.org/10.1186/s42400-022-00134-9

Ubik, S., & Žejdl, P. (2010). Evaluating application-layer classification using a machine learning technique over different high speed networks. *Proceedings - 5th International Conference on Systems and Networks Communications, ICSNC 2010*, 387–391. https://doi.org/10.1109/ICSNC.2010.66

Uchenna Joseph Umoga, Enoch Oluwademilade Sodiya, Ejike David Ugwuanyi, Boma Sonimitiem Jacks, Oluwaseun Augustine Lottu, Obinna Donald Daraojimba, & Alexander Obaigbena. (2024). Exploring the potential of AI-driven optimization in enhancing network performance and efficiency. *Magna Scientia Advanced Research and Reviews*, *10*(1), 368–378. https://doi.org/10.30574/msarr.2024.10.1.0028

Ullah, I., & Mahmoud, Q. H. (2021a). Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks. *IEEE Access*, *9*, 103906–103926. https://doi.org/10.1109/ACCESS.2021.3094024

Ullah, I., & Mahmoud, Q. H. (2021b). Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks. *IEEE Access*, *9*, 103906–103926. https://doi.org/10.1109/ACCESS.2021.3094024

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need*. http://arxiv.org/abs/1706.03762

Vlăduţu, A., Comăneci, D., & Dobre, C. (2017). Internet traffic classification based on flows' statistical properties with machine learning. *International Journal of Network Management*, *27*(3). https://doi.org/10.1002/nem.1929

Wang, P., Chen, X., Ye, F., & Sun, Z. (2019a). A Survey of Techniques for Mobile Service Encrypted Traffic Classification Using Deep Learning. *IEEE Access*, *7*, 54024–54033. https://doi.org/10.1109/ACCESS.2019.2912896

Wang, P., Chen, X., Ye, F., & Sun, Z. (2019b). A Survey of Techniques for Mobile Service Encrypted Traffic Classification Using Deep Learning. *IEEE Access*, *7*, 54024–54033. https://doi.org/10.1109/ACCESS.2019.2912896

Wang, X., Qiao, Y., Xiong, J., Zhao, Z., Zhang, N., Feng, M., & Jiang, C. (2024). Advanced Network Intrusion Detection with TabTransformer. *Journal of Theory and Practice of Engineering Science*, *4*(03), 191–198. https://doi.org/10.53469/jtpes.2024.04(03).18

Williams, N., Zander, S., & Armitage, G. (n.d.). *A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification*.

Wu, B., Xu, J., Zhang, Y., Liu, B., Gong, Y., & Huang, J. (n.d.). *Integration of Computer Networks and Artificial Neural Networks for an AI-based Network Operator*.

Yang, Z., Jin, Y., Liu, J., Xu, X., Zhang, Y., & Ji, S. (2025). *Research on Cloud Platform Network Traffic Monitoring and Anomaly Detection System based on Large Language Models*. http://arxiv.org/abs/2504.17807

Yu, C., Lan, J., Xie, J. C., & Hu, Y. (2018). QoS-aware traffic classification architecture using machine learning and deep packet inspection in SDNs. *Procedia Computer Science*, *131*, 1209–1216. https://doi.org/10.1016/j.procs.2018.04.331

Yuan, R., Li, Z., Guan, X., & Xu, L. (2010a). An SVM-based machine learning method for accurate Internet traffic classification. *Information Systems Frontiers*, *12*(2), 149–156. https://doi.org/10.1007/s10796-008-9131-2

Yuan, R., Li, Z., Guan, X., & Xu, L. (2010b). An SVM-based machine learning method for accurate Internet traffic classification. *Information Systems Frontiers*, *12*(2), 149–156. https://doi.org/10.1007/s10796-008-9131-2

Zander, S., Nguyen, T., & Armitage, G. (2005a). Self-learning IP traffic classification based on statistical flow characteristics. *Lecture Notes in Computer Science*, *3431*, 325–328. https://doi.org/10.1007/978-3-540-31966-5_26

Zander, S., Nguyen, T., & Armitage, G. (2005b). Self-learning IP traffic classification based on statistical flow characteristics. *Lecture Notes in Computer Science*, *3431*, 325–328. https://doi.org/10.1007/978-3-540-31966-5_26