Arab American University
Faculty of Graduate Studies
Department of Natural, Engineering and Technology Sciences
Master Program in Cybersecurity

Improving Network Security-Based Anomaly Detection Using Machine Learning And Deep Learning

Marah Radi Hawa
202316926

Supervision Committee:

Dr. Amani Yousef Owda

Dr. Huthaifa Ashqar

Dr. Mohammed Hussein

This Thesis Was Submitted in Partial Fulfilment of the Requirements for the Master Degree in Cybersecurity

Palestine, December/2025

Arab American University
Faculty of Graduate Studies
Department of Natural, Engineering and Technology Sciences
Master Program in Cybersecurity

Thesis Approval

Improving Network Security-Based Anomaly Detection Using Machine Learning And Deep Learning

Marah Radi Hawa
202316926

This thesis was defended successfully on ..30/12/2025... and approved by:

Thesis Committee Members:

Name	Title	Signature
1. Dr. Amani Yousef Owda	Main Supervisor	
2. Dr. Huthaifa Ashqar	Member of Supervision Committee	
3. Dr. Mohammed Hussein	Member of Supervision Committee	

Palestine, December/2025

Declaration

I declare that, except where explicit reference is made to the contribution of others, this thesis is substantially my own work and has not been submitted for any other degree at the Arab American University or any other institution.

Student Name: ·Marah Radi Hawa.·····

Student ID: ···202316926·····

Signature: 

Date of Submitting the Final Version of the Thesis: 19.1.2026

Dedication

To my family and friends for their ongoing support during my research I dedicate this thesis.

My parents' support and direction has been invaluable.

Finally, I dedicate this manuscript to my husband and children: your patience and understanding is what got me through.

To my brothers and sisters, thank you for the way you have held me up in prayer.

I am grateful to my friends and colleagues who collaborated with me along this educational journey.

And finally, I would like to thank the Arab American University for providing such a scientific environment where this work has been accomplished.

.Marah Radi Mohammed Hawa

Acknowledgments

I would like to thank the staff doctors Dr Amani Owda, Dr Huthaifa Ashqar and Dr Mohammed Hussein for their supervision and direction in this research. Their comments and questions during several talks about the thesis were useful and have lead to an improvement of this final version.

I am also grateful to Arab American University, in particular the faculty of graduate studies and engineering and technology sciences department for providing chances and academic follow up, so that I finish my master study.

Improving Network Security-Based Anomaly Detection Using Machine Learning and Deep Learning

Prepared By: Marah Radi Hawa

Supervision Committee: Dr. Amani Yousef Owda, Dr. Huthaifa Ashqar, Dr. Mohammed Hussein

Abstract

Developing technology has enabled greater potential for delivering services, while at the same time exposing new threats in cyberspace and online as digital capabilities proliferate. There are a number of drawbacks to typical intrusion detection systems (IDSs). Signature-based methods are unable to deal with zero-day and stealthy attacks, and anomaly-based approaches tend to produce a large number of false positives. These difficulties are magnified in resource-limited settings like Palestine where there is limited access to state-of-the-art security resources and representative local datasets. In this study, a hybrid intrusion detection system utilizing machine learning (ML) and deep learning (DL) is developed to overcome the above-mentioned limitations. The proposed method incorporates a two-stage system. In the first stage, several predefined models are trained and validated with global benchmark datasets to determine which architecture is appropriate. Second, the chosen model is implemented and fine-tuned based on a newly created Palestinian network traffic database. This hybrid of both global and local data ensures that the system can have a general detection capability while being sensitive to region-specific traffic scenarios. Experimental results show its superiority to the classical IDSs in both stability, adaptability to real network traffic and false positive rate. The model obtained global benchmark datasets exceeding 99% and the Palestinian dataset above 98.8%. Although similar state-of-the-art works have reported high classification performance on global datasets, none have considered the performance of IDS with real Palestinian traffic. Therefore, the reported local accuracy is a first known baseline for intrusion detection in this regional domain rather than comparative measures. We also show that errors and host activity are very important for separating bad traffic from good traffic. This shows that behavior characteristics that are specific to a domain are more useful than general features for finding anomalies.This work provides practical guidelines for developing scalable and cost-effective context-aware IDS solutions to address resource-constrained environments. The results reveal that the combination of global and local datasets could introduce a robust and interpretable intrusion detection system, which can be generalized to Palestine or similar cybersecurity contexts.

Keywords: Digital Technologies, Cyber Threats, Intrusion Detection Systems, Machine Learning, Deep Learning.

Table of Contents

List of Tables

List of Figures

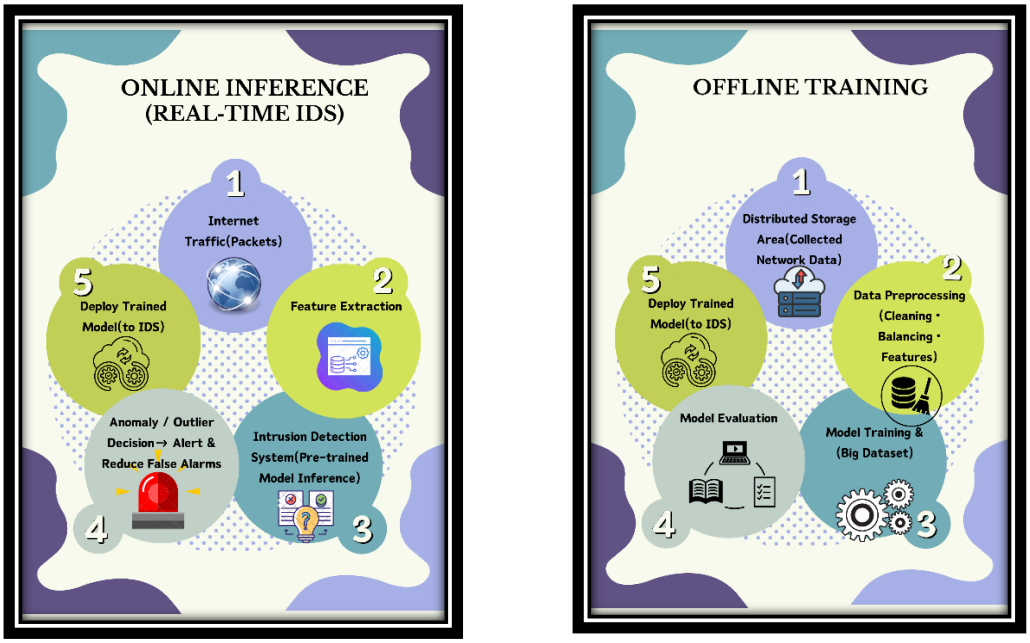
List of Definitions of Abbreviations

ML	Machine Learning
AI	Artificial Intelligence
APTs	Advanced Persistent Threats
IDS	Intrusion Detection System
CNNs	Convolutional Neural Networks
WBANs	Wireless Body Area Networks
NAD	Network Anomaly Detection
SVMs	Support Vector Machines
DL	Deep Learning
RNNs	Recurrent Neural Networks
LSTM	Long Short-Term Memory
DNN	Deep Neural Network
SDN	Software-Defined Networking
DoS	Denial of Service
U2R	User to Root
R2L	Remote to Local
MIMIC-II	Medical Information Mart for Intensive Care
DHMs	Deep Hybrid Models
GAN	Generative Adversarial Network
XGBoost	Extreme Gradient Boosting
SAE	Sparse Autoencoder
RF	Random Forest
MLP	Multi-Layer Perceptron
IoT.	Internet of Things
SIEM	Security Information and Event Management
SOAR	Security Orchestration, Automation, and Response
SOC	security operation center

Chapter One: Introduction to the Study

1.1 Introduction

The fast spread of digital networks has changed modern life in a big way. They support important services, including bank transfers, national utilities, and government functions. Admittedly, reliance on this connectivity also exposes networks to an increasing range of cyber threats—from low-level cybercriminals through to highly resourced state actors (Hindy et al., 2020). The emergence of sophisticated forms of attacks such as polymorphic malware, advanced persistent threats (APTs) (Soliman et al., 2021), and AI-aided intrusions has exposed the weak points in the standard signature-based form of defenses (Soltani et al., 2021). Figure 1.1 illustrates the architecture of an intrusion detection system(IDS).



(A) (B)

Figure 1.1: Network-Based IDS: (A) Online Inference (Real-Time); (B) Offline Training pipeline.

Figure 1.1 (A) Online inference (real-time): Internet packets are processed for feature extraction and evaluated with a pre-trained IDS model-method to identify anomalies and minimize false alarms in this architecture. 1.1 (B). Offline training

pipeline : establishes collected network data are purified, balanced and feature-engineered followed by model trained at large scale datasets and is deployed on the IDS.

Intrusion Detection Systems (IDS) are still one of the fundamental part in network security. Conventional signature-based IDSs can be successful at detecting known threats, but are largely ineffective against unknown or deliberately obfuscated attack behaviors such as zero-day threats (Gopireddy, 2018). Although anomaly-based IDSs can identify unusual patterns of network activity that raise suspicion, the false-positive rates associated with them are high enough to swamp security managers. These constraints—more particularly in resource-deprived settings with scarce access to sophisticated tools and available localized data—necessitate more flexible, intelligent, and context-aware detection approaches.

In the era of artificial intelligence(AI) techniques, intrusion detection has shifted from rule-based approaches towards data-driven and adaptive learning strategies. More importantly, ML and DL techniques have enabled IDSs to form complex flow behaviors based on large network data instead of applying static rules obtained by manual development and updating these models with the change of attacks (Fu et al., 2022; Aziz et al., 2023). This study extends these concepts by developing an artificial intelligence powered intrusion detection system that addresses the specific cybersecurity requirements of Palestine, and other low-resource environments.

Improvements in artificial intelligence (AI) have changed how intrusion detection works, moving it toward data-driven and adaptive learning models. Machine learning (ML) and deep learning (DL) methods are especially useful for IDSs because they can find complex traffic patterns in large amounts of network data and respond to new threats without having to rely on static, hand-written rules (Fu et al., 2022; Aziz et al., 2023). This study creates an AI-based intrusion detection framework that builds on these skills to help with the cybersecurity problems in Palestine and other places with limited resources.

Nevertheless, a critical gap remains: in under-resourced, region-specific environments, there is limited evidence on how global versus locally collected datasets affect IDS performance and which traffic features most strongly drive anomaly-detection accuracy. Addressing this gap motivates the research questions stated at the end of this chapter.

1.2 Significance of the Study

The significance of this work is investigating when AI-based IDS can make use of models trained on global benchmarks, compared to locally captured traffic and what are the most dominant features in traffic that affect anomaly-detection accuracy. To clarify the contribution, significance is presented as follows into two parts: (i) Theoretical Significance, which considers generalization under varying dataset sources and feature importance; and (ii) Practical (applied) Significance, to offer evidence for regional institutions and security operation center(SOC) teams for making selection of datasets and features that maximize accuracy yet minimize cost and privacy risk.

1.2.1 Theoretical Significance

In doing so, this work contributes to the theoretical understanding of AI-driven intrusion detection by exposing how dataset provenance,globally-leveled benchmark corpora or locally-collected network traffic—impacts a model's out-of-distribution generalization. Systematically comparing these sources, the study explores when and why models learned from global traffic may not characterize region-specific adversarial behaviors and traffic patterns. Concurrently, the study discovers a principled subset of traffic features that are most influential for accuracy in anomaly detection and thus adds to feature-engineering literature in IDS and insights that inspire learner model designs with higher interpretability. Last but not least, it presents a reproducible experiment to compare datasets/feature sets that starts from data preparation and cleansing, through data balancing and feature extraction, and up to training and evaluation, such that future researchers can reproduce the obtained results with methodological reproducibility.

1.2.2 Practical (Applied) Significance

Operationally, such an IDS model can be directly implemented into real operational network in ministries, university, ISP and government data center. The model can be deployed in an operational Security Operations Center (SOC) by interfacing with

live network traffic via a packet-capture sensor (e.g., Zeek, Suricata or SPAN/mirror port on a core switch). In this scenario, the model can process real time traffic and use the optimal features set discovered in this study to generate alerts onto anomalous or region-specific attacks which global models might not be able to detect.

It can also be running on edge routers of ministries or university campus, the hybrid model is operated as a light weight detection module, and alert suspicious flows before aggregating into internal servers. This supports use cases like identifying early brute-force attempts, ransomware command and control traffic, scanning activities or attacks (well known to target organizations in Palestine).

Furthermore, the model can be built into local SOC team SIEM(Security Information and Event Management), SOAR(Security Orchestration, Automation, and Response) (e.g., ELK/Splunk), etc. systems to reduce and sort features required by the model found by the study that reduces cost of collecting and storing data while keeping high detection rate. This enables security analysts to use the model in operational scenarios, such as automated alerting, dashboard visualization and incident-response workflows.

These deployment use-cases show cases how the proposed IDS scheme can be actually justified and deployed operationally within actual implementation of real networks further to enhance the threat visibility, prioritize local attack patterns and offer a cost-effective security decision-making mechanism in the Palestinian scenario

1.3 Research Problem

In resource-challenged environments like Palestine, an IDS performance deteriorates as most of its machine learning models are trained on international corpora that will not reflect regional traffic distribution and attack types, while a few localized datasets/studies exist (Javaid et al., 2016; Aziz et al., 2023). Signature based IDSs are mostly ineffective in detecting zero-day/APT attacks and Anomaly based systems generate a moderate or high number of false positives, which contributes to the operational overhead. (Soliman et al., 2021).This study addresses that gap by establishing a strong baseline via a hybrid ML/DL IDS trained and evaluated on global benchmarks, and then testing the best model on a newly collected Palestinian dataset (binary: normal vs. abnormal) to measure how global-local transfer affects accuracy and false positives. To our knowledge, no prior work has reported end-to-end results that integrate localized

Palestinian traffic within a hybrid ML/DL IDS framework (Fu et al., 2022; Shivhare et al., 2023).

1.4 Research Objectives

The group is fundamental that this research focuses on a specific aspect as follows:

1. Create and put into action a hybrid IDS model that uses both ML and DL algorithms to better find both normal and abnormal traffic.

Make a hybrid IDS that uses both supervised and unsupervised ML/DL. It should be trained on network traffic to find both normal and abnormal attacks. Look at the strengths of the other to make the results more accurate.

2. Performance with respect to global and local data

To understand how well the IDS model generalizes to other parts of the world, use global and local data sets (e.g., Palestine) to see how network traffic and attack types differ between regions. Data localisation is in place to be able to detect more accurately and avoid being susceptible to local cyber threats.

3. Find which network features matter.

Apply feature selection and engineering to engage the most significant differences between normal and attack traffic. It helps make models more accurate and efficient, and reduces false positives by focusing on it. most relevance featuresUse feature selection and engineering to find the most important differences between normal and malicious traffic. It improves the accuracy and efficiency of the model, reduces false positives, by concentrating on important features. the most important parts

1.5 Research Questions

To develop and evaluate an AI-based IDS to enhance the capability of detecting cyber threats, particularly in resource-constrained environments. The following work we have carried out will help answer the following research questions:

1. How can such a fusion of machine learning and deep learning models help in increasing the accuracy and reliability of intrusion detection systems, as well as reducing the false positives?
2. What is the effect of using global versus locally collected network datasets on the detection performance of AI-based intrusion detection systems?

3. Which network traffic characteristics have the greatest impact on the accuracy of anomaly detection in AI-driven IDS models?

1.6 Limitations of the Study

This study focuses on a network-based IDS using flow/session data with binary labels (normal vs. abnormal). We do not inspect packet content. The local dataset covers a limited time and a subset of Palestinian public-sector networks, so some seasonal or institution-specific patterns may be missing. Privacy, logging policies, and limited compute restrict the number of features and the size of models we can use. Global benchmark datasets may contain artifacts that differ from local traffic. Evaluation is offline in batch mode using recorded files. we do not emulate real-time traffic or deploy inline on a live network. To reduce these risks, we use clean data splits, report confidence intervals, keep the model as simple as possible with the most useful features, and outline plans to expand local data and replicate results in future work.

1.7 Conceptual and procedural Definitions

This section presents the conceptual and procedural definitions of the key terms used throughout the thesis. The goal is to achieve clarity, internal consistency and a common understanding of the key terms and constructs. Each definition is accompanied by a standard reference from the literature. Certain terms are defined in Table 1.1.

Table 1.1: Definitions of Major Concepts Used for This Study

Term	Definitions	Reference
Intrusion Detection System	looks for known, suspected, or unauthorized activity by using monitor network traffic .	(Shivhare et al., 2023)
AI-Based IDS	An intrusion detection system that utilizes AI or ML algorithms to automatically detect and categorize network anomalies.	(Aziz et al., 2023)
Anomaly Detection	The process focuses on analyzing data to uncover irregular patterns that differ from expected behavior and could signal malicious cyber activities.	Chandola et al. (2009)
Global Dataset	Global standard data used to detect network traffic(e.g., NSL-KDD, CIC-IDS2017)	(Maseer et al., 2023,Moustafa, et al. (2015))
Local Dataset	Data collected from the Palestinian Ministry of Education	

Feature Engineering	The science of deriving new network properties based on existing properties.	(Maseer et al., 2023)
Machine Learning Models	Different learning models, including Isolation Forest, XGBoost, and others.	(Aziz et al, 2023, Koorsen et al. (2023)
Evaluation Metrics	There are several quantitative evaluation metrics for measuring how well a model works. These are Accuracy, Precision, Recall and F1-score as well as threshold-independent ones such as ROC-AUC and PR-AUC.	(Ahmad, et al. 2021)
Binary Classification	The process of categorizing network traffic into two classes: Normal and Abnormal(Attack)	(Javaid et al., 2016)

1.8 Thesis Structure: A Roadmap for the Research Journey

This thesis is organized by way of a number of chapters, which are intended to aid in addressing the research aims and research questions.

presents the research background, which includes introducing the study's background and identifying the research problem. It introduces the research aims and questions, discusses the reasons for undertaking this research and motivates the work, and highlights its importance with respect to all of intrusion detection or more generally cybersecurity related issues.

Chapter 2 - Literature Review: Introduces the current state of lore concerning IDSs, including both ML and DL. It provides an indispensable overview of the relevant literature, highlights gaps in previous research, and describes how the current study fills these theoretical openings.

Chapter 3 – Proposed Methodology: This chapter outlines the research design and elaborates on the research execution. It talks about the datasets used (both global and local), how the data were collected, and the steps taken before the data were used, such as cleaning the dataset, normalizing it, and creating new features.

Chapter 4: Results introduces the gained results from all experiments, including performance comparison of models and effects on dataset origin (global vs. local), as well as selected features' properties.

Chapter 5: Discussion of results and Recommendations. The discussion relates to existing research and discusses the implications of the findings for deploying an IDS and

limitations affecting the results. concludes, makes recommendations for practitioners and future research in resource-poor settings such as Palestine.

References: Includes all related literature references mentioned in the thesis following APA reference format.

Appendix: It includes numerous additional material that are supportive of the main findings but is not suitable for embedding within the flow text. A visual "roadmap" of the structure of the thesis is shown in Figure 1.2:

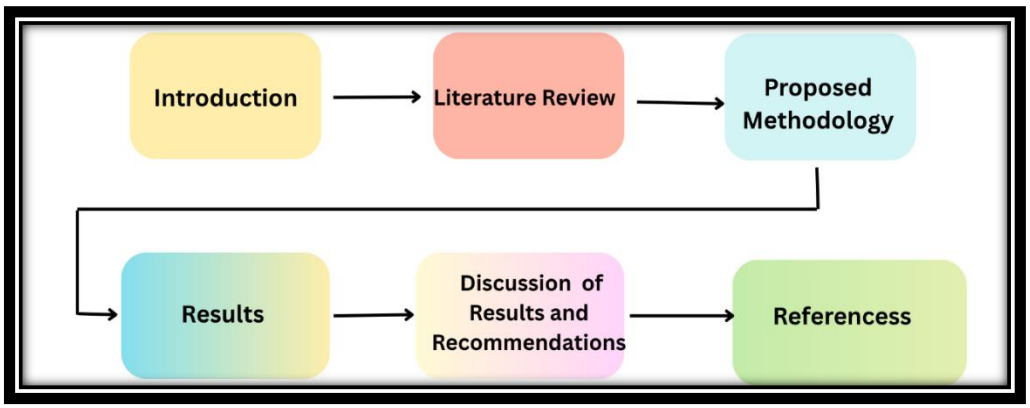


Figure 1.2: Overview of the Thesis Structure.

1.9 Chapter Summary

The background to the research problem was discussed in this chapter, and the rationale for using AI-based techniques of intrusion detection was provided. It described the primary objective of the study and defined the secondary qualitative research questions examined in this dissertation. The research questions were then presented, with a focus on comparing global datasets to locally gathered network traffic and identifying the most important traffic characteristics. Furthermore, we presented its practical implications and defined the scope of interest with a focus on the main limitations. Concepts and definitions of procedures employed in the research were also included. Finally, the chapter presented a brief outline of the thesis in general and Chapter 2 in particular, which is the Literature Review.

Chapter Two: Theoretical Framework and Previous Studies (Literature Review)

2.1 Introduction

Network anomaly detection is an essential operation in the modern age of cybersecurity that provides for detecting abnormal behaviors and potential threats concealed within network traffic. The traditional detection methods (such as the signature-based) can be not effective to deal with the increasingly diverse attacks brought forth by the escalating net intensity, sophisticated network layout being widely applied. But on the other hand this limitation has encouraged more recent ML techniques, especially in DL architecture to optimize anomaly detection performance in relation to accuracy and efficiency. (Rao et al., 2024).

In recent years, it has become apparent that applying deep-learning models (Yang et al., 2022) including Convolutional Neural Networks (CNNs), autoencoders, and isolation forests, is highly beneficial for network anomaly detection, whereas hybrid models are also being used. Such models are also particularly powerful in learning automatic features from massive amounts of network traffic data, and not rely on heavy human-intervention or deep domain-knowledge. They are capable of learning intricate patterns and identifying subtle deviations, even when considering high-dimensional data points that could be overlooked by human analysts (Sánchez et al., 2024).

The novelty of such advanced methods is that they can recognize both normal and abnormal attacks. The emergence of IoT (Zolfagharipour et al., 2023), Software-Defined Networking (SDN) (Alzahrani et al.,2021) and large-scale data centers has caused the network size and structure to explode. With the explosive growth of network traffic in the last years, there is an increasing need for efficient anomaly detection techniques. Furthermore, the publication of different contemporary datasets (e.g. NSL-KDD, CICIDS-2017), BoT-IoT has played a significant role in expediting advancements in employing DL models by providing researchers with proper benchmarks to improve and compare their forecasting performance in the development and evaluation of deep-learning models (Sánchez et al., 2024).

In this study, we present a comprehensive survey of recent advances on network anomaly detection, especially focusing on deep learning- based techniques in cybersecurity. Focusing on the principal models, tools, sets of data and challenges

emerging from this problem, we discuss how network security has evolved over time, as well as the new possibilities brought by machine learning to improve detection systems.

2.2 Background of Network Anomaly Detection

Network Anomaly Detection (NAD) is a promising research area in cybersecurity for identifying abnormal/suspicious activities in network traffic, which could possibly indicate an attack or unauthorized activity. As the sophistication of network infrastructures has dramatically increased, traditional signature-based detection approaches that are based on known attack signatures to detect malicious activities in case we face with continue poisoning attacks can no longer guarantee the capability to identify a novel or modified threat (Dasgupta et al., 2022).

In order to overcome these limitations, several authors started adopting Machine Learning (ML) approaches in IDSs. Earlier ML approaches (e.g., decision trees and SVMs) enhanced adaptability by training on labeled data to identify known threats. However, with the advance of networks, because it is necessary to detect novel attacks as well, unsupervised learning methods (Liu et al., 2023), such that clustering and statistical anomaly-based change-point detection, determine differences from a given data set without using annotated labels.

Great strides have been made in intrusion detection using Deep Learning methods (Chalapathy et al., 2022). Architectures like Convolutional Neural Networks(CNN), Recurrent Neural Networks(RNN), etc., including Long Short-Term Memory(LSTM) models, can directly learn complex and hierarchical patterns from network traffic data. This feature increases detection power as well as reduces dependence on hand-engineered features (Ayeni et al., 2023).

Elapsed since up to today the release of benchmark datasets such as NSL-KDD, CICIDS-2017 , or BoT-IoT can be accessed by the public, which contributed in accelerating NAD research through providing standardized labeled examples pertaining to different attack types, such as DoS, and DDoS attacks with botnets (Tanim et al., 2024). These are the datasets that have allowed to construct more general models, and performance comparison between different works.

Nevertheless, there are still some obstacles that remain, such as the problem of class imbalance in network traffic data and very high false positive rates for detection, not to mention the fact that it is computationally very expensive when DL architectures become

large (Dasgupta et al., 2022). In addition, scarcity of labels still restricts supervised methods and hence hybrid frameworks become more attractive. More recent works have investigated the fusion of models(e.g., CNN–GAN or autoencoder architectures) to improve robustness and generalization (Rao et al., 2024).

In short, NAD has undergone a lot of change over the last ten years - from manual, signature-based solutions to intelligent AI/hybrid AI-based systems. Nevertheless, the existing problem of data imbalance, model interpretability and computational efficiency are still open research challenges that drive our work.

2.3 Theoretical Framework

This work is based on a combination of theories from anomaly detection, machine learning and deep learning. These both provide the theoretical underpinning for use of pipeline and hybrid ML/DL models, inspire attention to dataset provenance, and validate our feature-importance analyses.

2.3.1 Statistical & Model-based Anomaly Detection

Statistical anomaly detection assumes that normal network activity obeys measurable mathematical laws and any major deviations can reflect unusual activities. In this study, we translate such objective into mathematical feature engineering: we extract significant characteristics from bytes_ratio, network_activity, and dst_host_activity. These learnable representations capture deviations from frequent traffic behaviors, and are used as more informative inputs to the ML/DL models, contributing to their robustness and consistency across different sets of data (Dasgupta et al., 2022).

2.3.2 Machine Learning and Deep Learning Theory

Machine learning provides the fundamental ability to classify or cluster network traffic through learning statistical patterns of benign and malicious practices. Yet, DL greatly expands on this engine by autonomously learning hierarchical features from un- or preprocessed data. This has been pointed out by Chalapathy et al. (2022) due to the reason that deep learning models have shown great performance in modeling nonlinear complex spatial dependencies among features and temporal dynamics within time-series traffic flow, which can hardly be modeled based on much hand-crafted feature engineering by traditional ML methods. By reducing dependence on hand-engineered

features and allowing the model to generalize beyond available attack signatures, DL fortifies identification of subtle, evolving and previously unknown intrusions. This synergy helps DL emerge as a potent supplement of ML for better robustness and adaptability in the next-generation ID environments.

2.3.3 Hybrid Learning Theory

Hybrid learning scheme aims at utilizing the complementary strengths of classical machine learning and deep learning to provide models that are accurate and explainable. In this formulation, "hybrid" represents the combination of domain-motivated feature engineering and a deep neural network (DNN) classifier. The computed features offer a clear or atasensitive representaton of network behavior relevant to security for the interpretability of why certain trac is malicious. In the mean time, DNNs embed complex non-linear interaction that traditional ML models can not be well represented on learned representations and lead to enhanced detection performance. This organization provides a useful compromise in terms of the interpretability and accuracy balance without needing extra stacked ML layers—like, for example, XGBoost or RF ensembles—that frequently do not add advances consistent with the model complexity (Bizzarri et al., 2024).

2.3.4 Relevance to the Study

These theoretical consideration also guide the present study. The study builds a hybrid intrusion detection model using machine learning and deep learning concepts, considering domain-specific feature engineering with deep neural architectures. This theoretical alignment supports a comprehensive testing over both global benchmarking dataset and a locally acquired Palestinian public sector dataset, to guarantee the robustness of the model under different network conditions. Moreover, in contrast with existing work, the use of interpretable hand-crafted features together with deep, non-linear learning allows for an in-depth feature-importance analysis such that the system is able to not only have high detection accuracy but also to explain why a series of patterns are seen as malicious. Based on these theories, the research work is proceeding to an applicable and interpretable IDS that can be scaled for use in actual operation in Palestinian governmental networks.

2.4 Top Five Techniques in Network Anomaly Detection

Conventional signature-based and statistical detectors are vulnerable to new adaptive attacks. In such case, ML and DL technologies have been extensively studied to construct more intelligent and flexible IDSs, specifically in large-scale complicated networks for near real-time applications.

Here, we consider five prototypical types of state of the art algorithms that illustrate "modern" and "classical" branches: CNN vs autoencoders in learning traffic representations, Isolation Forest to detect anomalies in high-dimensional spaces with small computational effort (sampled during processing time), hybrid ML/DL models capturing spatial and temporal dependencies, the complete anomaly detection pipeline following a generic multi-stage process –ranging from data preprocessing to domain-informed feature selection, model training, and evaluation– where each analysis applies its own set of techniques/configurations at every stage. We evaluate these pipeline-based solutions with respect to detection performance, runtime efficiency, scalability and operational deployability, which reveals the strengths and weaknesses of each method together with its applicability in real-world network defense.

2.4.1 Enhanced Network Anomaly Detection Using CNN in Cybersecurity Operations

Deep learning, especially CNNs, has been extensively embraced in the new and advancing field of network security for improving intrusion and attribute detection technologies. The surveyed works suggest CNN-based and hybrid models that employ recent datasets as well as smart feature selection to provide high detection accuracy while being realistic in real-time.

A major study by Ayeni et al. (2023) have used CNNs on CICIDS-2017 dataset to train an intrusion detection system. DoS, DDoS and brute force attacks as well as SQL injection, botnet and other attack types that are only very limitedly represented or not at all in previous datasets such as the KDD Cup 1999 dataset. CICIDS-2017 includes detailed flow-level features (total 78) over five days, which are determined to be normal or attacks. To facilitate the learning of mapping, data in the subset are split into its training-testing parts, and we used CNNs for features extraction to have minimal human intervention. The accuracy of the model was 0.997, a value higher than both classical machine learning methods in addition to other deep learning action potentials models. While the results indicated CNN outperformed other models in terms of learning features

and detecting fever, the study also reported that hyperparameter tuning and transfer-learning to a real-world dynamic network has room for improvement.

Tanim et al. (2024) proposed a CNN- Anomaly detector trained with BoT-IoT – a mixed dataset which includes normal traffic and complex attacks such as DDoS, DoS keylogging, and data exfiltration. with billions of flow records collected, this dataset is large enough to support deep-learning techniques for training and evaluation. Data pre-processing involved scaling, normalizing and mathematical feature expansion. With a view to improving interpretability and less redundancy, mutual information-based entropy feature selection and Pearson correlation were utilized. The employed CNN model consisted of 3×3 and 5×5 convolutional filters, pooling layers as well as fully connected (FC) layers. The offline-trained model performed 0.96 accuracy in online testing situations. Nevertheless, the authors identified challenges concerning requirements of abundant labeled data, interpretability, and processing cost.

Rao et al. (2024) introduced a hybrid intrusion detection model including a CNN-GAN model to make the anomaly detection process possible more effective. CNNs extracted complex features and GANs generated synthetic normal traffic for data augmentation as well as generalisation purposes, to identify subtle/anomalous events that were never seen before. The model was trained on the NSL-KDD dataset, which is an enhancement of the KDD Cup 1999 to handle duplication and class imbalance. The hybrid framework of the detector obtained satisfactory detection rate as well as low false alarms, and had a high potential for practical applications. The study however recognized the issue of scalability and the difficulty in tuning hybrid deep-learning architectures.

Overall, these studies emphasize the necessity of recent data such as CICIDS2017, BoT-IoT and NSL-KDD to train CNN-based IDSs. The techniques leverage automatic feature generation, intelligent pre-processing and mixed modeling to improve model performance. The reported detection accuracies varied between 0.96 and 0.997, highlighting the usefulness of these techniques for protecting sophisticate networks. So, There is a many challanges still appearing – like the requirement for large labeled datasets, computational efficiency and interpretability persist– especially in the context of deploying models in rapidly evolving real-world settings.

2.4.2 Evaluating the Isolation Forest Method for Anomaly Detection in Software-Defined Networking Security

Growing size and complexity of modern networks demand more effective means for anomaly detection, particularly to protect infrastructures from advanced cyber attacks. In the same line, isolation forest (IF) was evaluated in order to detection anomalies on network traffic, mainly based on Software-Defined Networking(SDN) scenarios.

Sri Lakshmi et al. (2023) explored the introduction of IF to SDN, also inspired by the fusion of cloud computing and IoT. Due to the absence of publicly available labeled datasets from a real operational networks, synthetic traffic traces were simulated for normal and malicious behaviours, issuing a multivariate Normal distribution to capture the realistic rates of containment. Before training the IF model, data preprocess includes normalization and standardization, where the latter isolates anomalies of shorter path lengths under the tree structure. The framework was also implemented along with the SDN controller to automatically change network policies based on detected anomalies. Although the model has the capability of attack detection, performance metrics such as precision, recall and f1-score, AUC, etc are relatively poor even in more complicated network conditions. However, the work provided some perspective on IF's capability to scale and support real-time SDN anomaly detection.

Chua et al. (2024) utilized IF on a real web traffic dataset of 10 million Nginx access logs collected from an Iranian e-commerce website. The dataset had 3% class imbalance which is highly imbalanced, it is also a typical problem in anomaly detection. The preprocessing consisted of ingestion, the elimination of NULL values and extraction of length URIs, User-Agent features and encoding categoricals using one-hot. The Python-based IF model was written with the scikit-learn package, and parallelized to run for various contamination rates. The accuracy, precision, recall and F1-score of the model was 0.93, 0.95, 0.90 and 0.92 respectively. Although they achieved strong performance, their limitations are that they were adversely affected by false positives (swamping) and false negatives (masking), particularly when handling rare or obfuscated attack signatures, emphasizing the necessity for better feature engineering and threshold adjustment.

Mykhaylova et al. (2024) applied IF to detect brute-force login attacks by profiling such behavior with decision tree ensembles. Login attempts were analyzed for the presence of features such as IP address, username, and timestamp patterns. The brute-force attacks were identified with good accuracy and false positives rate through the

model. But the researchers pointed out some limitations: models need regular retraining to be effective, are sensitive to class imbalance and can potentially be bypassed by savvy attackers who act normally. The work stressed out the opportunity of using IF along with explainable AI toward enhanced transparency and robustness in real-time intrusion detection systems.

2.4.3 Autoencoder-Based Network Anomaly Detection

In this subsection, the surveyed studies concentrate on applying autoencoder as an unsupervised way to perform anomaly detection for high-dimensional data. The main gist is to train the model so that it learns how to reconstruct "normal" samples, then check for anomaly by measuring the reconstruction error.

Ganesha et al. (2020) experimented with several AEs for network intrusion detection based on the NSL-KDD data set. They tried to learn new compact latent representations of normal traffic flows and identified anomalous demands through the reconstruction errors. Attack types in the dataset comprised of Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L) and probing, with a considerable class imbalance. After preprocessing (label encoding, normalisation and data cleaning), 122 features were used for modelling. Among the variants considered, the Sparse Deep Denoising Autoencoder reported one of its highest 5 accuracy, with a value of 0.893. where the models shows potential for treatment with imbalanced and partial labeled data, it is vulnerable to threshold selection and does not work well for finding attacks that have normal-like patterns. These limitations indicate the requirement for hybrid or ensemble techniques.

Similarly, Torabi et al. (2023) show a technique for cloud networks, a vector-based reconstruction error method with autoencoders was used and tested on the CIDDS-001 dataset. Unlike previous approaches that only collapse reconstruction errors into a single scalar, their approach accumulates error per feature and compares each with its own threshold. This improvement compensates for misclassification due to averaging and results in the better localisation of an anomaly. The categories of attacks are variadic and the procedure was about label encoding, normalization and dimension reduction. Their hierarchical multi-class model exceeded in accuracy, recall, FPR and F1-score others. Although it works well it is also computationally expensive and suffers from tuning of threshold.

Rassam (2024) utilized an unsupervised method which employed autoencoder–CNN structure for WBANs in clinical settings. The early detection of sensor malfunctions was explored within the multivariate physiological signals recorded in MIMIC-II. After preprocessing and normalization, the architecture was trained to reconstruct signals where high reconstruction error were detected as an anomalous pattern. It achieved the best F1-score (0.96) and low reconstruction error, making it better than previous unsupervised approaches both in adaptability and efficiency. But hyperparameter tuning and generalization to unobserved anomalies are the challenging issues, thus calling for the adaptive thresholding and real time streaming techniques.

Nisioti et al. (2018) where in a variety of intrusion-detection studies utilized an autoencoder model—deep autoencoder, sparse autoencoder, variational autoencoder (VAE)—to conduct unsupervised anomaly detection by reconstructing normal traffic and labeling deviations as attacks. These models were primarily tested on the datasets of NSL-KDD, KDDCUP99, and CICIDS2017 and performed well in terms of identifying unknown or zero-day attacks. Nevertheless, the survey highlights strong and weak points: In particular, autoencoders can generalize too much and might not be able to detect subtle attacks, it could overfit when training data of normal traffic are scarce and it does not perform well under concept drift in real networks.

2.4.4 Hybrid Models for Network Anomaly Detection

Modern studies has been focusing the advantages of hybrid deep models to further enhance the detection precision for network anomaly by combining ML and DL algorithms.Noor et al. (2025) illustrate hybrid modeling which consistes the advantages of ML and DL-based methods into one framework to handle challenges from imbalanced and high-dimensional data. They used an SAE for deep feature extraction, RF to select features and MLP for final classification and achieved 0.98 accuracy on NSL-KDD dataset .

Chalapathy et al. (2022) show the positive of hybrid modeling (CNN/LSTM-AE) architectures for joint learning of spatial and temporal features on network traffic data which leads to better generalization and reduced overfitting.

Chen et al. (2016), showed how the XGBoost system can be incorporated into hybrid pipelines as a strong ML model for feature ranking, regression and classification. Taken together, these works collectively demonstrate that hybridization (at the

feature/model/decisions level) improves detection performance by leveraging the interpretability of ML and the representational power of DL.

Zhang et al. (2025) provided an authoritative survey of the DL application in IDS, and they pointed out two problems that have not yet been thoroughly resolved: spatiotemporal feature extraction, class imbalance. They compared various architectures such as CNNs, RNNs, LSTMs, and GAN-based models in their review and also discussed how these approaches improve feature representation or tackle data imbalance by generating synthetic attacks and using adaptive loss functions. The authors found that although deep learning techniques have enhanced detection accuracy, the majority of models continued to experience challenges in terms of real-time efficiency, interpretability and generalization under dynamic network settings. These results also illustrate the necessity for hybrid, attention-based architectures that are able to summarize temporal–spatial correlations and generalize well in real-world, imbalanced environments.

Chen et al. (2024) shows a hybrid IDS which combines data-level resampling and spatial–temporal deep feature extraction. The model first mitigates class imbalance with random under-sampling and SMOTE, then captures spatial patterns with a CNN and global temporal dependencies by a Transformer encoder. The applicability of the proposed model on the NSL-KDD dataset was also demonstrated with strong multiclass performance [(0.992) accuracy] while surpassing all other methods in terms of FP rates for most categories. The hybrid model achieved (0.997) accuracy on the CICIDS2017 dataset, surpassing state-of-the-arts. However, the authors acknowledge several limitations: their model strongly relies on the quality of SMOTE, can add synthetic noise and its CNN-Transformer architecture makes it more computationally expensive. Moreover, the assessment only based on two datasets may weaken its applicability to high widespread real-world scenario (such as high-volume network environment).

Basit et al. (2022) presented a hybrid and multi-layer deep learning intrusion detection system using a combination of the statistical processing and deep neural architectures to address the low accuracy associated with the conventional IDS mechanisms. They propose to learn discriminative traffic features by extracting and selecting network traffic data based on a multilayer CNN, and classifying the discovered network flows with a Softmax classifier. To improve its classification ability, the authors additionally insert a multilayer DNN as a further decision layer. The performance of the model was tested out on two standard IDS datasets NSL-KDD and KDDCUP’99,

in terms of accuracy, recall, precision and F1-score. It is experimentally verified that their combination scheme is effective and attains an accuracy of about 0.99, which also outperforms a number of existing intrusion detection systems. This integration shows the effectiveness of combining CNN-structured feature extraction with deeper multilayers classifiers in terms of performance improvement for intrusion detection problem.

2.4.5 Pipelines for Anomaly Detection

Recently researchers have proposed to employ end-to-end machine learning pipelines for improving network anomaly detection systems. Such pipelines often include multiple stages, which includes data pre-processing (e.g., normalization and feature extraction) to prepare the input data for ingestion into and application by a model. Then as testing data arrives, the ensemble learning phase integrates the outputs from several ML models (such as XGBoost and Random Forests) to ensure generalization and avoid overfitting Chen et al., (2024).

These ensemble methods provide the detection system with a pool of models to have accurate predictions by benefiting from each algorithm. In the third stage, complex high dimensional data is mined for anomalies with deep learning such as CNNs and LSTMs and autoencoders.

as shown in Chen et al. (2024), where class-weight adjustments considerably boosted the detection on imbalanced datasets, especially when some types of attacks were not well represented.also CNN+LSTM models illustrate good performance as an end-to-end solution for real-time detection tasks due to their ability to combine the spatial and the temporal features.

Zhang et al. (2025) have used these advantages to show spatial and temporal patterns from traffic of networking, like the size of packet and type of protocol. When these hybrids are incorporated into the high-accuracy pipeline, a scale-invariant broad-spectrum (Ruiz et al., 2011) real time anomaly detection platform can be achieved (Chen et al., 2024; Zhang et al., 2025).

Chalapathy et al.(2022) discussed deep hybrid pipelines, where convolutional and recurrent layers are arranged sequentially (e.g., CNN → LSTM → Autoencoder) to capture both spatial and temporal characteristics in network traffic, thereby enabling multilevel anomaly detection.

Talukder et al. (2022) developed a complete processing pipeline that begins with preprocessing the KDDCUP'99 and CIC-MalMem-2022 datasets, followed by handling data imbalance using SMOTE and selecting optimal features through XGBoost feature selection. Their pipeline then evaluates multiple ML/DL models and reports extremely high performance, achieving 0.999 accuracy on KDDCUP'99 and 1.0 accuracy on CIC-MalMem-2022.

Also , Almuhanna and Dardouri (2025) show a multi-step hybrid ensemble pipeline that starts with preprocessing, feature engineering, and imbalance correction using SMOTE. their study trains multiple heterogeneous classifiers—including XGBoost, Random Forest, Graph Neural Networks (GNN), LSTM, and Autoencoders—and integrates their predictions through a weighted soft-voting mechanism. tested on more than 5.6 million real network traffic records, the ensemble achieved almost perfect performance, reaching close to 1.0 accuracy, precision, recall, and F1-score.

Moubayed (2024) probably shows an end-to-end pipeline for a 5G network that includes modules for EDA, data preprocessing, and feature selection. The last step in detection uses a deep learning-based classifier made for software-defined 5G environments.. On the 5G-NIDD dataset, the proposed pipeline achieved detection accuracy of more than 0.995, outperforming a good result compared to previous IDS models, and demonstrating the merits of structured, phased pipelines in contemporary intrusion detection.

2.5 Research Gap and Study Contribution

The section shows the most important research gaps, which were found in earlier studies in this area.A lot of gaps show how the results work in different situations, how generalizable they are, how different the datasets are, and how easy they are to understand.The study delineates its scientific and practical contributions aimed at addressing these deficiencies and advancing IDS research towards real-world applicability. The following subsections will first give a short summary of the research gaps and then go into more detail about the study's specific contributions.

2.5.1 Research Gap

As the size and complexity of network systems are rapidly moving toward the large scale and becoming ever-more sophisticated, current research in this area is shifting

focus towards advanced anomaly-detection methods. Conventional system cannot adapt to time-varying and non-homogeneous traffic. Despite the promising results of ML and DL models—specially, CNNs and autoencoders—for identifying known as well as unknown kinds of attacks (Torabi et al., 2023), some unresolved issues have been noted in the literature such as class imbalance, poor generalization, and lack of interpretability (Zhang et al., 2025; Liu et al., 2023; Sánchez et al., 2024; Dasgupta et al., 2022).

For example, Ayeni et al. (2023) used a CNN and trained it on CICIDS-2017, and obtained an accuracy of 0.997.Moreover their model depended only on the supervised learning task use to train it, being strongly associated with the distribution of the dataset. Focused only in training a detector for unseen traffic or for a specific region. Similarly, Tanim et al. (2024) applied entropy- and correlation-based feature selection with a CNN on BoT-IoT obtaining 0.96 accuracy, but they introduced heavy preprocessing in their pipeline and ignored per-feature interpretability aspect as well as did not test against heterogeneous or localized environments.

Hybrid networks such as the CNN-GAN model of Rao et al. (2024) tried to work around these data scarcity issues by synthetically generating traffic. While detection performance was enhanced, the system suffered from GAN-training instability, decreased recall and a high computational cost which precluded it from being used in real-time or resource-limited applications.

The CNN-Transformer hybrid model introduced in Chen et al. (2024) reached state of the art performance at significant expense: (1) they rely heavily on synthetic resampling (random undersampling and SMOTE), which risks the introduction of artificial noise and deformation of minority-class boundaries; and, (2) they propose an architecture that it is computationally expensive and validated only on two benchmark datasets, limiting scalability and real world relevance.

We tackle these limitations in my work, by proposing a more organized and computationally efficient pipeline that includes balanced feature engineering, controlled dimensionality reduction and optimized model combination. The proposed two-layer frameworks is actively supported by light-weight synthetic over-sampling techniques, which are only used moderately within the global process leading to limited computation complexity and more stable behavior across datasets and in detecting rarer or emerging attack patterns

Basit et al. (2022) (also with high inclusive precision), however, exclusively based their work on outdated benchmark datasets and spatial CNN features without taking into

consideration temporal dependency or data imbalance. The proposed pipeline addresses these challenges by integrating scalable approaches able to learn from modern traffic that exhibit both spatial and temporal patterns.

However, despite the reported performance being good in many pipeline-based IDS works, there are critical limitations: Talukder et al. (2022) The model didn't work well with noisy or operational traffic because it relied on old global datasets. Almuhanna and Dardouri (2025) gain near-perfect results, but their model depends on many heavy classifiers and is trained on structured, controlled data, rather than the more heterogeneous enterprise scenarios to keep it practical/cost effective.

Similarly, Moubayed (2024) targets his pipeline only based on a dataset generated from 5G, limiting the applicability to other than softwarized 5G infrastructure.

Contrarily, the proposed pipeline involves sophisticated feature engineering, PCA-based dimensionality reduction and controlled imbalance treatment using SMOTE along with an efficient yet light-weight deep neural classifier. Most importantly, the model is tested not only on NSL-KDD but also on a real Ministry dataset with more realistic network background states, and it can produce more robust and generalization result and has better ability to detect the minority attack patterns in actual network conditions.

Unsupervised methods like Isolation Forest are label-free, however do not work well in high-dimensional or complex network (Sri Lakshmi et al., 2023; Mykhaylova et al., 2024). Autoencoder-based models, too (Ganisha et al., 2020; Torabi et al., 2023; Rassam et al., 2024), struggle with threshold sensitivity, computational overhead as well as anomalies that are not far from resembling normal traffic. The previous study show three primary gaps:

1. Generalization Gap: Most models just tested on global data, making this models unreliable for generalization in resource-limited environments. (CICIDS-2017, BoT-IoT, NSL-KDD).
2. Feature-Integration Gap: Past attempts are based on a strong dependence either of deep automatic feature extraction or manual entropy/correlation-based selection, without the ability to unify domain-informed feature engineering and deep learning in a unique pipeline.
3. Interpretability Gap: The lack of explainable models (CNN, LSTM, autoencoder) hinders applications in security-sensitive settings demanding explainability.

To fill these gaps, we propose in this study a High-Accuracy NSL-KDD Pipeline, which combines domain-oriented feature engineering with lightweight DNN classifiers,

providing an intermediate approach between the traditional characteristic-based methods and solely deep models.

2.5.2 Contribution of the Study

This work contributes with four main theoretical and practical contributions that fill gaps outlined by prior work:

1. First assessment of IDS models with Palestinian network real data

To the best of our knowledge, this is the first research to investigate how IDS models perform on real Palestinian network traffic. This study is the first localized method that gives baseline and novelty of attack behaviors which are not covered in global benchmark data set.

2. Hybrid pipeline of deep learning and domain-driven feature engineering

Unlike previous approaches, which only use (handcrafted or automatically learned) features [e.g., CNN], in this study we combined domain-informed feature engineering with a simple DNN classifier. This hybrid model increases interpretability, mitigates overfit and is more stable than using deep learning models alone.

3. A two-step evaluation process consisting of international standards then local validation

The research initially used standard algorithms on the global NSL-KDD dataset and compared them to select the best model.

This selected model was then applied and evaluated on a newly collected Palestinian dataset to examine its effectiveness in a real, region-specific environment.

This benchmarking-then-local-validation strategy fills a major gap in previous works, which typically validated models on a single global dataset without assessing cross-environment performance.

4. Interpretability and the usage in actual SOC environments

It leverages Feature-importance scoring and dynamic threshold calibration to present clear results to SOC analysts. It is less feature-rich and leaner so it’s easy to use in real time, but also in low-resource environments: political networks or academic networks like those found in Palestine. Table 2.1 Illustrates a Summary of Related Work and Limitations.

Table 2.1: Summary of Prior IDS Studies, Datasets, Methods, and Limitations

Paper	Focus Area	Techniques Used	Strengths	Limitations	Evaluation Metrics	Dataset Used	Contribution to Cybersecurity
Ayeni et al. (2023)	CNN-based IDS	CNN for feature extraction, accuracy evaluation	High accuracy, automated feature extraction	Hyperparameter tuning	Accuracy (0.997)	CICIDS-2017	CNN-based real-time IDS
Tanim et al. (2024)	CNN-based Anomaly Detection for IoT	CNN, feature selection	Real-time, interpretable	Large dataset, processing cost	Accuracy (0.96)	BoT-IoT	Real-time detection for IoT networks
Rao et al. (2024)	Hybrid CNN-GAN Anomaly Detection	CNN, GAN for data augmentation	High accuracy, low FP	Complexity, scaling	Accuracy: 98.71%,F1-Score: 96.58%	NSL-KDD	With robustness via GAN, less data is needed
Sri Lakshmi et al. (2023)	Anomaly Detection in SDN using Isolation Forest	Isolation Forest, SDN	Scalable anomaly detection	Low performance in complex networks	Accuracy≈ 0.93, F1-Score 0.89	Synthetic network traffic (normal and anomalous)	Isolation Forest for adaptive SDN
Chua et al. (2024)	Web Traffic Anomaly Detection	Isolation Forest, data preprocessing, feature extraction	High precision and recall in web anomalies	False positives/negatives, needs better features	Accuracy (0.93), F1-score (0.92)	Nginx log data (10M rows)	Supports web anomaly detection and monitoring

Mykhaylova et al. (2024)	Brute Force Attack Detection	Isolation Forest, decision tree	High accuracy, low false positives	Sensitive to imbalanced datasets	Accuracy (high)	Login attempts (IP, username, time)	Enhances brute force detection in real-time
Ganesha et al. (2020)	Autoencoder-based Anomaly Detection in Network Intrusion	Sparse/Denoising Autoencoders	Unsupervised, handles imbalanced data	False positives from threshold sensitivity	Accuracy (89.34), F1-score	NSL-KDD	Improves anomaly detection with unlabeled data
Torabi et al. (2023)	Cloud Network Anomaly Detection	Autoencoder, vector error, thresholding	Better accuracy & anomaly localization	High complexity, real-time challenges	Accuracy, recall, false positive rate, F1-score	CIDDS-001	Boosts cloud security with better thresholds
Rassam (2024)	Unsupervised Anomaly Detection in WBANs using Autoencoder CNN	Autoencoder CNN, reconstruction error for anomaly scoring	adaptive to dynamic data	Hyperparameter tuning is weak with unseen anomalies	High accuracy (F1-score 0.96),	MIMIC-II	Enhances early detection of abnormal vital signs in hospitals
Zhang et al. (2025)	Hybrid Network Intrusion Detection	CNNs, RNNs, LSTMs, and GAN-based models	Combines multiple techniques for feature selection and spatial and temporal learning	Complexity of integration, computation cost	Accuracy = (90.61) F1-Score = (92.26)	NSL-KDD	Proposed a hybrid model that improves anomaly detection accuracy by combining XGBoost, CNN, and LSTM.

Noor et al (2025)	pipeline	Autoencoder + MLP	Effectively addresses the class imbalance	Limited evaluation scope	(reported as "high accuracy,"	NSL-KDD	Proposes a hybrid framework that balances data, reduces irrelevant features, and enhances MLP-based intrusion detection accuracy.
Chalapathy et al (2022)	Hybrid Network Intrusion Detection	CNN-LSTM-AE architectures	Effective when labeled data is scarce, suitable for unsupervised detection	Struggles with high-dimensional data, prone to overfitting	Accuracy: (0.92) , F1-Score: (0.86),	KDD Cup 1999, NSL-KDD	Introduced autoencoders as an effective unsupervised anomaly detection method.
Chen et al. (2024)	Hybrid Network Intrusion Detection	CNN + Transformer	High accuracy with better rare-attack detection	Computationally heavy and depends on resampling quality	Accuracy: (0.997) and (0.992)	CICIDS2017 , NSL-KDD	Enhances network defense by reducing false alerts
Basit et al. (2022)	Hybrid Network Intrusion Detection	CNN + DNN with Softmax classifier	High accuracy with strong deep feature extraction	No imbalance handling and limited generalization	Accuracy: (~0.99)	NSL-KDD and KDDCUP'99	Enhanced intrusion detection through hybrid deep learning
Talukder et al. (2022)	Pipeline	SMOTE-based data balancing with XGBoost	Strong SMOTE+XGBoost pipeline boosts accuracy	Relies on old, pre-processed datasets	Accuracy: (0.99)	KDDCUP'99 , CIC-MalMem-2022	Improved feature selection and imbalance handling in IDS.
Almuhanna et al. (2025)	Pipeline	Hybrid ensemble pipeline	Near-perfect performance through a powerful hybrid ensemble	Computationally heavy and unsuitable for real-time deployment	Accuracy: (1.0)	5.6M real network records	Introduced a high-accuracy ensemble-based IDS for large-scale traffic.
Moubayed (2024)	Pipeline	Complete EDA-to-DL pipeline	Well-structured pipeline with high detection accuracy	Limited generalization targets only 5G	Accuracy: (0.99)	5G-NIDD	Advanced DL-based intrusion detection tailored for 5G networks.

2.6 Conceptual Framework

The macro analytical framework for this study is outlined in Figure 2.1. It breaks down the research by theme, articulating how the contributions address the major challenges found in the literature generalization, feature integration and model interpretability. The framework has three main components: Inputs – global (NSL-KDD) and local Palestine network datasets, which are in the form of the real-world challenges as following:class imbalance, noisy labels, distribution shift. Hybrid Pipeline: a unified workflow which uses domain-informed feature-engineering (e.g., bytes_ratio, error_rate_diff, host_activity indicators and temporal window aggregates), PCA-based dimensionality reduction, tailored handling of imbalanced classes, trimmed DNN classifier and post-hoc threshold calibration. Outputs: strong performance metrics (F1, ROC-AUC, PR-AUC), increased transferability from global to local traffic and improved operation interpretability via per-feature attributions.

The argument is that the engineered features regularize the learning to imbalance and changing traffic behavior, and that threshold calibration aligns the precision–recall curve of the classifier with security requirements in operations. This framework implements the contributions presented in Section 2.5.2 and serves as a foundation of the proposed intrusion-detection pipeline.

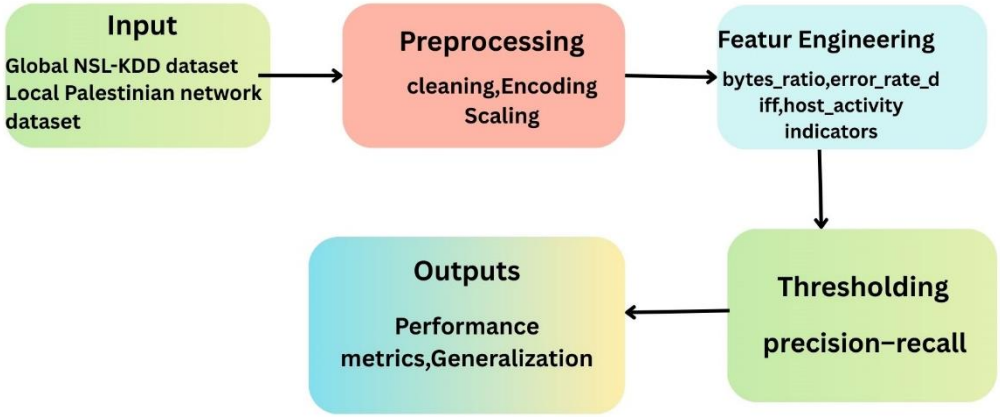


Figure 2.1: Block Diagram of the Hybrid High-Accuracy NSL-KDD Pipeline

Figure 2.1 shows a high-level diagram of the entire process followed in this work. The figure shows in sequence from input data to final outputs of the core processing steps and how the modules of the proposed pipeline are arranged inside our system.

2.7 Chapter Summary

The closing window of NAD technologies in a world with growing cybersecurity threats Hackers and malware use many methods to get onto a network and start causing problems. Conventional detection methods, including signature-based tools, are not able to follow the rate and diversity of modern cyber-attacks. As a result, machine learning and deep learning methods such as CNNs], autoencoders5, isolation forest, hybrid models have demonstrated good performance in identifying known and unknown network traffic anomalies. Despite the success of NAD, a few challenges remain, such as data imbalance, limited model interpretability and high computation complexity of deep learning algorithms. These problems prevent the NAD systems from being deployed and scaled in a resource-constrained environment, e.g., IoT. To overcome these obstacles, recent advances in hybrid modeling are utilised to improve model transparency, robustness and flexibility. We refine these methods with the use of existing tools and techniques, combined with understanding of a dataset that have been collected locally. This allows us to practically evaluate model scalability and performance under real traffic patterns and operational limitations.

Chapter Three: Research Methodology

3.1 Introduction

Network anomaly detection is an essential approach toward assuring the integrity and security of today's networks. By now, with the rapid increase of internet traffic and higher level cyber attacks, traditional defense solutions are powerless to fight with cloaked or zero-day threats. Here's yet where you can have your deep learning models, hybrid methods (nice advanced machine learning tools). Those resources make the approach to network anomaly detection more able to detect not only better-known threats but also unknown threats on networks, by means of applying these brand-new technologies.

In this section, methodology of the improved anomaly detection is presented. In this study, we use a hybrid pipeline that combines feature engineering methods of the machine-learning community and a DNN classifier. The stage of machine learning is divided into data preparation and preprocessing, standardization, and feature engineering which provide features bytes_ratio, error_rate_diff and host_activity to enrich data representation. The deep learning stage, based on DNN architecture, extracts complex nonlinear representations in the processed data leading to enhancement of classification performance and generalization. These stages combined together constitute the HighAccuracyNSLKDDPipeline that provides a flexible and efficient mechanism for behavioral anomalies identification in network traffic.

3.2 Framework Applied to Network Anomaly Detection

As shown in Figure 3.1, our model for network anomaly detection is a hybrid pipeline using the best of both worlds of the advanced feature engineering and DNN classifier models, exploiting their complimentary benefits.

Figure 3.1 also shows the sequence assessment method introduced in this work. The model is initially trained and tested over the international knowledge-based network intrusion detection system (KBNIDS) with NSL-KDD dataset, whereas to test whether our approach performs well in a real-world context let us benchmark our best resulting network design to the local Palestinian scenario. This architecture enables the pipeline to

verify both predictive accuracy and cross-environmental robustness before making the final attack/non-attack decision.

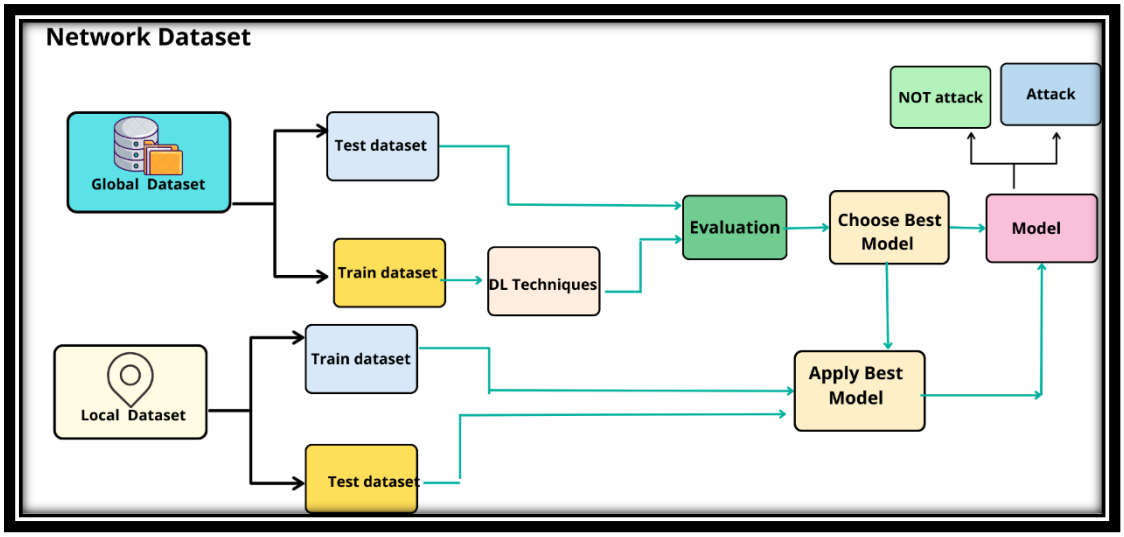


Figure 3.1: Proposed Workflow In General

In this chapter, we discuss the complete framework developed for detecting both normal and abnormal network behaviors using the NSL-KDD dataset and the local dataset. This framework is composed of several consecutive steps, from the data collection and preprocessing to feature engineering and selection, model's training phase and corresponding evaluation, see Figure 3.2.

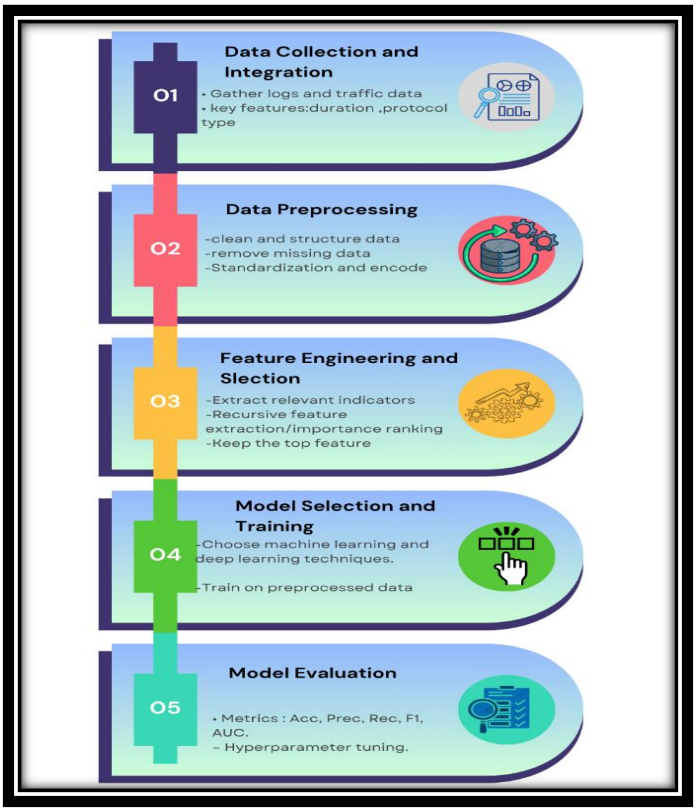


Figure 3.2: Stages of the Proposed Methodology

As shown in Figure 3.2, this process transforms raw network traffic data into meaningful input suitable for deep learning models, enabling accurate and efficient anomaly detection. The proposed framework synthesizes concepts taken from data mining, machine learning, and cybersecurity to result in a comprehensive loose-coupled scheme for the identification of potential intrusions and anomalous behaviour in large-scale networks.

3.2.1 Data Collection and Integration

The data gathering phase implies the generation of network traffic logs from different sources like routers, firewalls. Those logs carry important features, such as protocol type, length of communication and volume of packets that are crucial to recognize the pattern in network behavior. The analyzed flows contain normal and abnormal network activities and form the basis for training as well as testing. For the sake of credibility, all datasets are elaborately validated and combined into an integrated form for further preprocessing and feature engineering procedures.

- **Global Dataset**

  The NSL-KDD dataset(Kaggle, 2017) stands out as the most frequently used dataset in the fields of cybersecurity, particularly in relation to the development and analysis of IDS. It is the cleaned-up version of the original KDD Cup 1999 dataset (KDD Cup 1999, 1999). and involves a lot of preprocessing of collected data, and this includes the removal of redundant fields, and the classes are also balanced. This is labeled data where the normal and attack datasets can be used as the basis of comparison and evaluation of the performance of the machine learning algorithm in the detection of intrusions on network data (Tavallaee et al., 2009).

  Therefore, the NSL-KDD data set is significant because such a kind of data set can provide a clue on normal network traffic and abnormal network traffic to enable the development of systems that can effectively and autonomously identify and react to network security threats. With the growing trends of security attacks and vulnerable networks, datasets such as NSL-KDD are needed to support researchers or practitioners to improve IDS solutions. These datasets are rich in useful feature space that capture the semantics of network behaviors. As presented in Table 3.1, the NSL-KDD dataset includes several traffic features. Furthermore, Table 3.2. summarizes the key

characteristics of the NSL-KDD dataset used in this study, and Figure 3.3 illustrates the distribution of the dataset classes (Normal vs. abnormal) before preprocessing.

Table 3.1: Features of NSL-KDD Dataset

Category	Feature	Description
Connection Info	duration	Length of the connection in seconds.
	protocol_type	Type of protocol used (e.g., TCP, UDP, ICMP).
	service	Type of service (e.g., HTTP, FTP, DNS).
Traffic Volume	src_bytes	Bytes sent from the source machine.
	dst_bytes	Bytes sent to the destination machine.
	Count	Number of connections to the same host.
Error Indicators	wrong_fragment	Number of wrong fragments in the connection.
	Urgent	Number of urgent packets.
	error_rate	The rate of service errors in the connection.
Login Info	num_failed_logins	Number of failed login attempts during the session.
	logged_in	Binary feature indicating successful login (1 for success, 0 for failure)
Privilege Escalation	root_shell	Indicates if the root shell was used (1 if true).
	su_attempted	Number of attempts to switch to superuser (root).
File Access	num_file_creations	Number of files created during the session
	num_shells	Number of shell commands run during the session.
Attack Detection	label	Indicates if the traffic is normal or an attack (e.g., DoS, Probe, R2L, U2R).

Table 3.2: presents a Summarized Description of the NSL-KDD Dataset

Dataset	Source	Row	Column Loaded	Type	Highlights
NSl-KDD	Enhance KDD-Cup 1999	148.517	43	Labeled (Normal/Attack)	Duplicates removed; class balance improved; engineered features + one-hot encoding

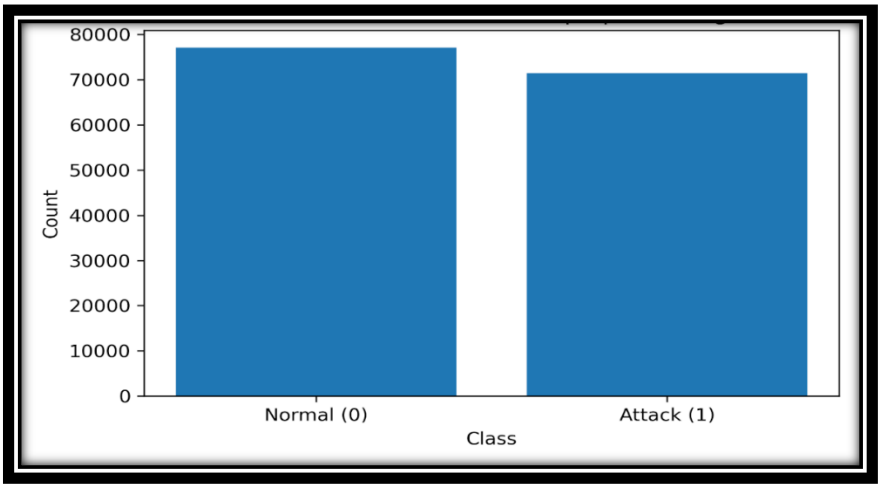


Figure 3.3: Class Distribution in NSL-KDD

- **Local Dataset**

The local dataset consists of 89,110 network connection records with 35 features, representing real-world traffic collected from the Ministry of Education environment. This dataset provides detailed host-activity metrics, error-rate indicators, byte statistics and session-behavior features that can be used to analyze normal as well as malicious traffic patterns. The acquisition of information for the local set is in accordance with existing institutional guidelines and privacy laws and has been approved by the institution's ethical review board. It is then essential to seek documentary permission for this data from the relevant agencies within the MoE (e.g. IT and Data Protection Officers). That's encryption that on all data gathering adheres to the laws and regulations about privacy in studies or research locally as well as other single standards. The manner in which data was collected and managed adheres to official approval from the Ministry (see ***Appendix D***).

Critical network-analysis data are collected from a variety of sources (eg, routers, firewalls). These collect metadata about connections, including how long it is up, what protocol is used (TCP/UDP/ICMP etc — there are dozens of them), what service types are involved (web serving or FTP transferring or querying a DNS server) and the IP addresses and ports involved. You can also gather traffic volumes—the total amount of bytes each machine sends to and receives from the others—which I find useful for dissecting network behavior. These data attributes resemble with the global dataset used for complex network analysis.

The data that you gather quickly gets stored and formatted so that it maintains an order and stays usable for better analysis in the future. Anonymity is required to comply with the privacy of all participants and prevent the abuse of ultrasensitive information. so, the ethical standards are maintained. Figure 3.4 illustrates the workflow followed to collect, structure, and prepare the local dataset used in this study.

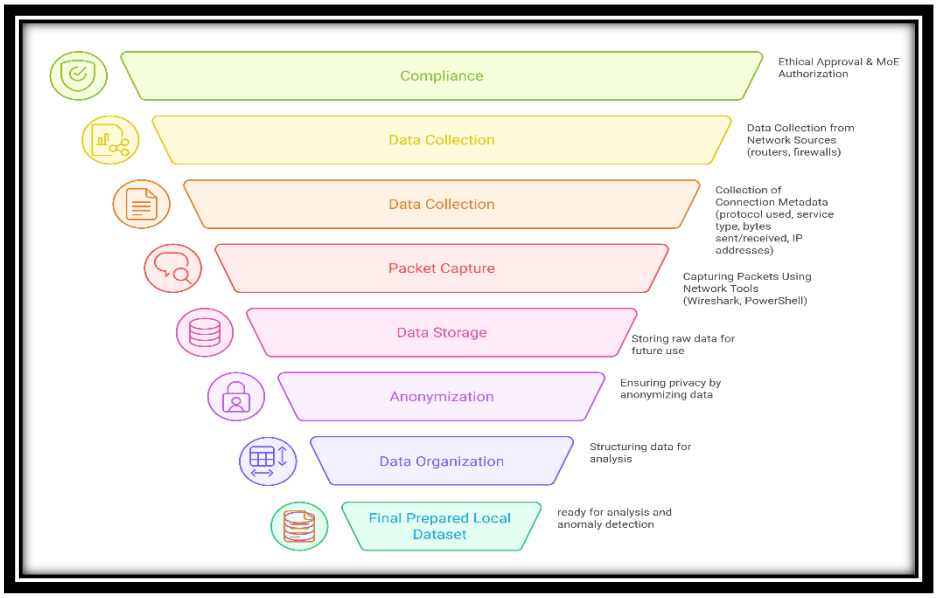


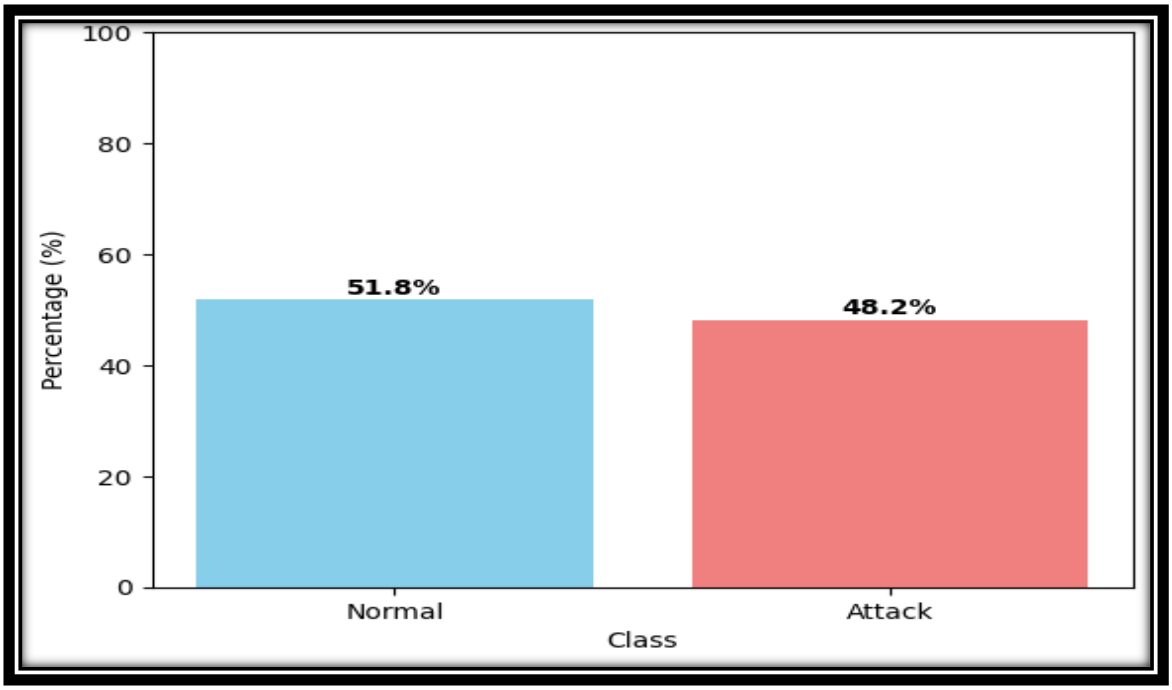
Figure 3.4:Work Flow to Collect Local Dataset

Different software tools are also used to improve the data collection and management process. However, to capture the packets on the network, you can use tools like Wireshark and PowerShell. To organize and structure the data that is obtained, Python libraries like Pandas and NumPy are employed, which in turn makes the data analysis easier. Table 3.3 shows the summary features of the local dataset.

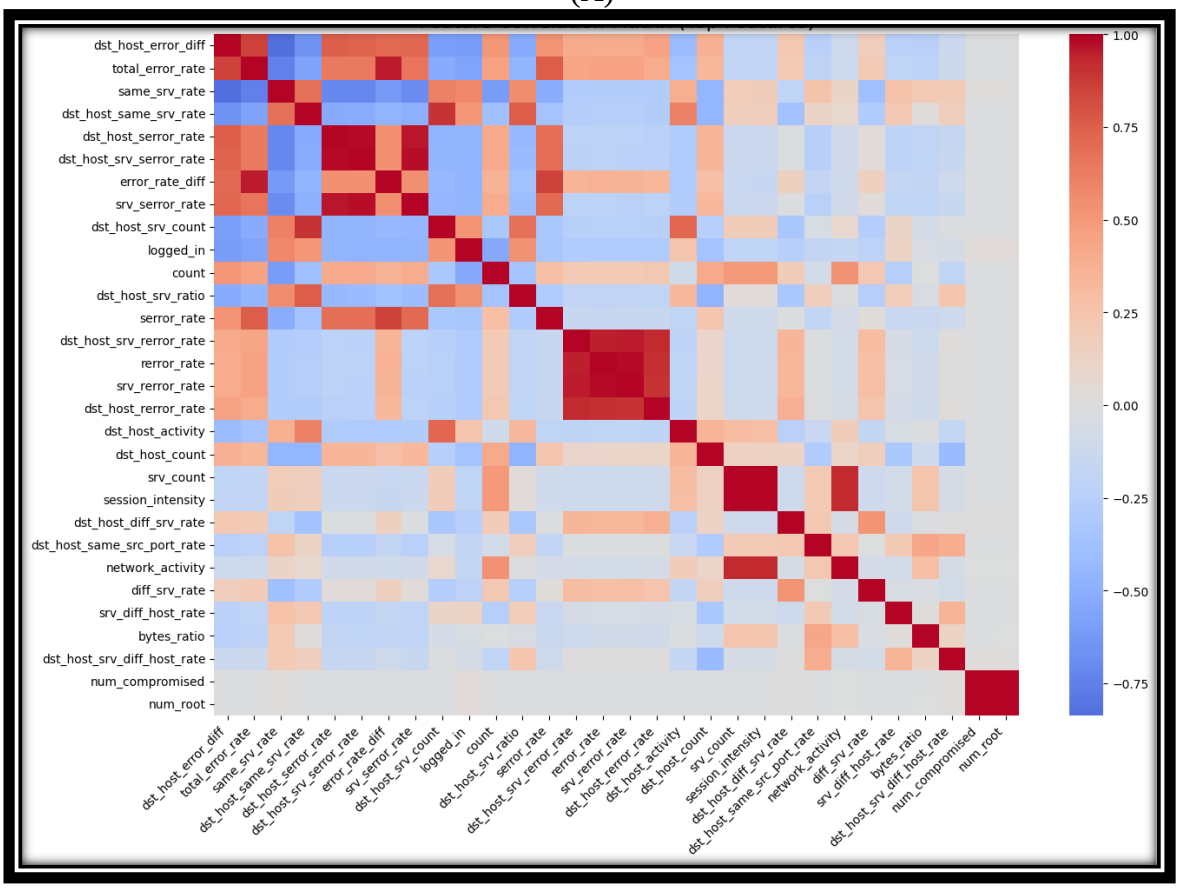
Throughout the whole process, great emphasis was placed upon keeping data accurate and respecting all privacy and security policies regarding the dataset made available by the Ministry of Education to be used for social network analysis and anomaly detection.Figure3.5. illustrates the distribution of the local dataset.

Table 3.3: Features of the local Dataset

Category	Feature	Description
Connection Info	Flag	Status flag of the connection (e.g., SF, S0, REJ)
	protocol_type	Type of protocol used (e.g., TCP, UDP, ICMP).
	service	Type of service (e.g., HTTP, DNS).
Traffic Volume	src_bytes	Bytes sent from the source machine.
	dst_bytes	Bytes sent to the destination machine.
	Count	Number of connections to the same host.
Error Indicators	wrong_fragment	Number of wrong fragments in the connection.
	Urgent	Number of urgent packets.
	serror_rate	percentage of connections that have "SYN" errors.
Login Info	num_failed_logins	Number of failed login attempts during the session.
	logged_in	Binary feature indicating successful login (1 for success, 0 for failure)
Privilege Escalation	root_shell	Indicates if the root shell was used (1 if true).
	su_attempted	Number of attempts to switch to superuser (root).
File Access	num_file_creations	Number of files created during the session
	num_shells	Number of shell commands run during the session.
Attack Detection	label	Indicates if the traffic is normal or an attack.



(A)



(B)

Figure 3.5:(A) Class Distribution in Local dataset,(B) Feature correlation Matrix.

Figure 3.5 (A) illustrates the class distribution of the local dataset used in this study. The dataset is relatively balanced, with 51.8% of the samples labeled as "Normal" and

48.2% labeled as "Attack“. Figure 3.5 (B) presents the feature correlation matrix for the top 30 most influential features in the local dataset after applying advanced preprocessing and feature engineering. The heatmap shows some strong positive correlations between error-rate-related features, such as serror_rate and srv_serror_rate, dst_host_serror_rate and dst_host_srv_serror_rate, suggesting that these metrics together capture similar behavior patterns which are usually related to attack traffic. Moreover, connection frequency and host activity-related features (e.g., count, srv_count, network_activity and session_intensity) also display significant correlation due to their correlated value in delineating frequent or repetition traffic pattern. By contrast, there is also a group of features including num_root and num_compromised with low correlations to the others, indicating they contribute different but nonfrequent information pattern in dataset.

3.2.2 Data Preprocessing

Data preprocessing is one of the critical first steps when processing network traffic data as well as applying it to machine learning models for anomaly detection. It leads to clean, well-structured and normalized data and optimizes the detection efficiency of systems (Alrayes et al. 2024). Preprocessing by tackling with missing information, irrelevant attribute and great inconsistency problem structures can improve the model to learn high accurate network intrusion incidents and anomalies. Below, we will describe all stages within the stage of data preprocessing for network anomaly detection.

- **Missing Data Handling and Data Cleaning**

The absence of data is also a frequent situation in network traffic datasets and the preprocessing pipeline begins with assessing the completeness and structure of data. The source of missingness can be attributed to incomplete recordings, sever fault in gathering data, and etc. In this article, the local dataset (local_dataset. csv) was first read and validated for existence of main label column and also ensured that valid binary values where present in it (0 for normal and 1 for attack). No imputation was performed on the data, as there were no missing values in the dataset. Besides, in case of necessity, the label column text to numeric conversion was done in order to make then consistent for the model fitting. Noisy or inconsistently organized data could lead to decreased model performance, and we were unable to assume that the data was already clean and well-organized in a prepared feature ready for scaling/engineering.

- **Standardization and scale feature**

Once we are satisfied that the data is clean and well formatted, the next preprocessing stage comprises of scaling features to ensure that all are numerical comparisons can be made for learning (Hosseini et al., 2023). Some in-network data for instance (e.g., packet size, connection duration, byte counts) can be on very different numerical scales where the magnitude of one given feature over-shadows other at training. To handle this, we standardized the features using the StandardScaler class, which scales input so that each feature has a mean value of zero and a standard deviation to one. This procedure normalizes feature importance in the training voice, which makes the model's learning become more reliable during the stabilizing learning process and stable test (Zhao et al., 2021).

- **One-Hot Encoding Categorical Features**

A significant amount of traffic data is based on categorical attributes, including protocol type (TCP/UDP/ICMP) and requested service (HTTP/FTP/DNS). Machine learning models accept only numerical data and therefore, categorical features are transformed into numeric formats (Azar et al., 2023). Categorical data was converted into binary vectors in this study, and one-hot encoding were used to transform the categorical columns excluding the target label column. Additionally, dimension …ality was reduced (and computation speed increased) by retaining only the top 5 most common categories per feature. This selective one-hot encoding helped the dataset to be compatible with the deep learning model but also preserved the most important categorical information resulting in a fair trade-off between feature richness and training performance (Bolikulov et al., 2024).

3.2.3 Feature Engineering

Feature engineering is a key component of learning models to have better discrimination power in detecting network anomaly detections. Feature engineering is done through domain-informed transformations in conjunction with data-driven selection methods to surface behavioral patterns not immediately present in raw network logs. This is particularly important for intrusion detection purposes, as slight variations (e.g., irregular byte transfer ratios or sudden increase in host activity) may be symptomatic of malicious activities.

In order to do this, multiple novel features were derived from the original traffic subprocesses. These also include bytes_ratio, bytes_diff and total_packets that account

for directionality of communication flows, transcendence and volume of data exchange instead.

Supplemental off error-related metrics, such as total_error_rate and error_rate_diff, point to mismatches between SYN vs. RST-based failure patterns (which frequently do indicate scanning or brute forcing). Host-centric attributes, such as dst_host_activity and dst_host_error_diff quantify service-specific repetition and error asymmetries at the destination host whereas time-behavior features like connection_speed, network_activity and session_intensity assist in capturing bursty or repeated session patterns that mark out several attack categories. ***Appendix A*** shows a representative selection of the derived features employed in the local step. Feature construction is followed by a two-stage dimensionality-reduction approach. First, the top-k features most associated with the attack label are retained by SelectKBest (ANOVA F-score). Next, PCA is used to maintain 99% variance for reducing the high-dimensional input into a compact form so that redundancy among the features can be minimized which in turn minimizes the computational cost. This combination of domain-driven feature construction and statistical selection achieves an expressive yet efficient final feature space, which allows the deep neural network classifier to learn decision boundaries that are more robust and generalisable. The definitions of top features selected in this work are available in ***Appendix B*.**

3.2.4 Handling Imbalanced Data

The network traffic datasets are commonly class-imbalanced, including normal behavior that occurs most frequently and a tiny fraction of anomalous or malicious behavior, such as intrusion or DDoS attack. Such an imbalance can skew the model to predict normal traffic, leading to miss rare but important anomalies.

To alleviate this problem, our pipeline uses SMOTE after the feature engineering stage (feature scaling, statistical feature extraction, SelectKBest and PCA). Then SMOTE is performed only on the training data to artificially create minority samples (anomalous traffic), keeping validation and test sets intact. This makes the model equally learn from normal and abnormal behaviors, effectively enhancing its performance to discover rare but high-impact attacks without leaking data.

3.2.5 Model Selection and Training

An architecture of a machine learning model represents how layers and pieces are connected to each other, and how they produce an output from input data. In network anomaly detection, architecture selection is important for obtaining well performance both in classification and generalization. this study represent deep models such as autoencoders, CNNs, isolation forests and hybrid models (CNN-LSTM-XGBoost), HighAccuracyNSLKDDPipeline . All experimental models were deliberately selected due to their ability to cope with the known features of Internet traffic data high dimensionality, non linearity and temporal dependence—guaranteeing a strong and smart detection system. The detailed hyperparameter configurations of all implemented models are summarized in ***Appendix C***. The chosen models are illustrated below.

- **Autoencoder**

Autoencoders are effective for network anomaly detection because they learn the normal behavior of traffic without requiring labeled attacks, then flag deviations via reconstruction error: if the model fails to accurately reconstruct an input, the high error indicates an anomaly (Hinton et al, 2006). In this study, we adopted an unsupervised Autoencoder that is trained only on normal traffic, then applied to mixed (normal/attack) traffic at test time; predictions are driven by a tuned error threshold rather than a supervised classifier. This choice aligns with recent literature advocating representation learning of benign behavior when labeled attacks are scarce and evolving.

The unsupervised autoencoder was applied using the NSL-KDD dataset to learn the normal behavior of network traffic. The data was preprocessed by standardizing numerical features using StandardScaler and encoding categorical attributes (protocol_type, service, and flag) using OneHotEncoder. PCA (95% variance) was employed for noise and dimensionless reduction before training. The architecture of the model was encoder–bottleneck–decoder with batch normalization, Dropout (0.2), and L2 regularization to enhance generalization. The network was trained with Nadam (0.0005) and MSE loss, against overfitting. EarlyStopping and ModelCheckpoint were applied.

The Autoencoder is trained on normal traffic alone to indicate anomalies using the error of reconstruction, i.e. high value implies abnormality. The optimal threshold was chosen between the 10th to 99th percentile optimizing F1 or G-mean score, with balanced precision and recall. This method is selected in that it does not need labeled attacks, fits for the imbalanced network data and can detect novel intrusions efficiently. When used

together with PCA and regularization, it offers a robust, interpretable and computationally effective anomaly detection approach for networks (Zhao et al., 2021). The autoloader models are depicted in Figure 3.6.

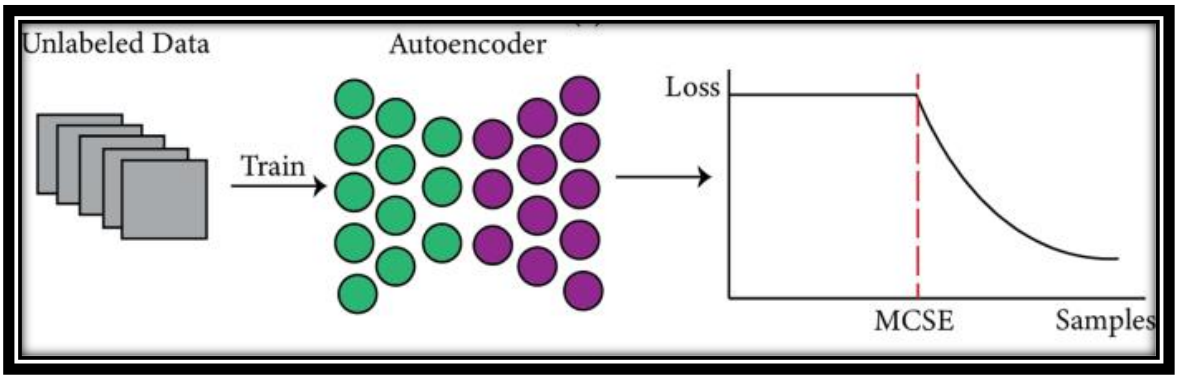


Figure 3.6: Autoencoder Model (Gulamali et al., 2022)

Figure 3.6 The training process of the autoencoder. The model is trained on unlabeled normal data to learn low-dimensional compact representations, measured by reconstruction loss over samples. Minimal cumulative squared error (MCSE) is the threshold to differentiate normal data pattern from possible anomalous .

- **Isolation Forest**

  The Isolation Forest (IF) an algorithm is a unsupervised machine learning method that has been optimized to make anomalies detection on high dimensions network traffic data. The difference is that IF does not learn the normal behaviourite is trying to find exceptions. It also presumes infrequent and distinct anomalies compared to the normal data points, enabling better separation in random division. . This is especially useful for outlier, intrusion or rare attack behaviour detection of large-scale network data. So, its multiple isolation trees (iTrees) can isolate the abnormal samples in smaller splits, which would make it very fast and scalable for real-time network anomaly detection. Figure 3.7 shows the design of an isolation forest.

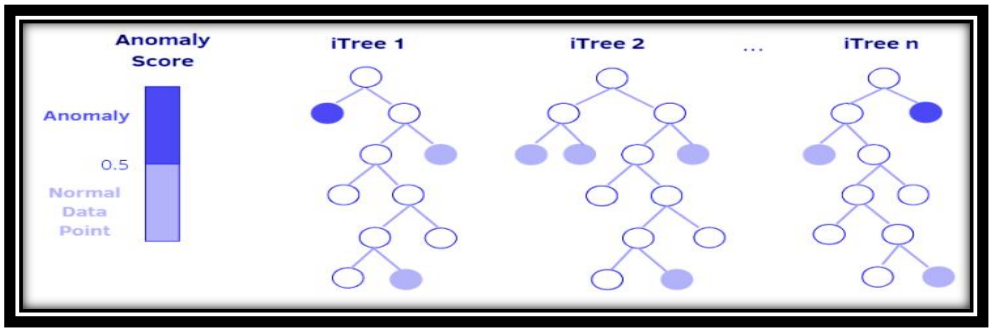


Figure 3.7: Isolation Forest with Random iTrees (Van Otten, 2024)

We apply Isolation Forest (IF) on NSL-KDD in an unsupervised manner. Categorical fields (protocol_type, service, flag) are encoded with LabelEncoder (not one-hot), then all features are standardized (StandardScaler). IF is trained only on normal training samples. The model uses n_estimators = 200, max_features = 0.8, contamination = 0.1, and random_state = 42. At test time, we compute anomaly scores = −score_samples(X_test) (higher = more anomalous), choose a data-driven threshold = 90th percentile of the scores, and predict attack = 1 if the score exceeds that threshold. We report Accuracy, ROC-AUC (on anomaly scores), classification report, show the confusion matrix, precision–recall curve with AP, and plot the score distribution with the chosen threshold.

IF isolates point via random splits, true anomalies get shorter paths, so their scores are higher. This fits our setting: high-dimensional traffic, few/unknown attacks, and a need for label-free detection with fast inference. Training on normal only avoids bias from imperfect labels; percentile-based thresholding (90th) gives a simple, reproducible operating point we can retune to trade precision/recall as needed. These decisions collectively make IF a scalable unsupervised baseline that aligns well with our deep model in terms of capturing inexpensive minority/unseen behaviors.

- **Convolutional Neural Networks**

Convolutional neural networks CNN is one of the best performing deep learning models for network ANM since it automatically learns unique representations from raw data without having to rely heavily on manual feature engineering. The Convolutional layers learn local patterns about features (packet size relationships, error rates and communication frequencies in this case), while the pooling layers decrease dimensionality and noisy signals and enhance generalization.Nonlinear activation functions (ReLUs) add the ability to model complex relationships between normal and

harmful behavior, and normalization techniques such as batch normalization, dropout, and L2 help reduce overfitting in high-dimensional and noisy environments like network traffic data. Thus, CNNs provide an efficient and scalable framework for supervised classification and complement non-supervised models (such as Autoencoder/Isolation Forest) by delivering strong performance when labeled data is available (Zhou et al., 2020). Figure 3.8 illustrates the architecture of a CNN.

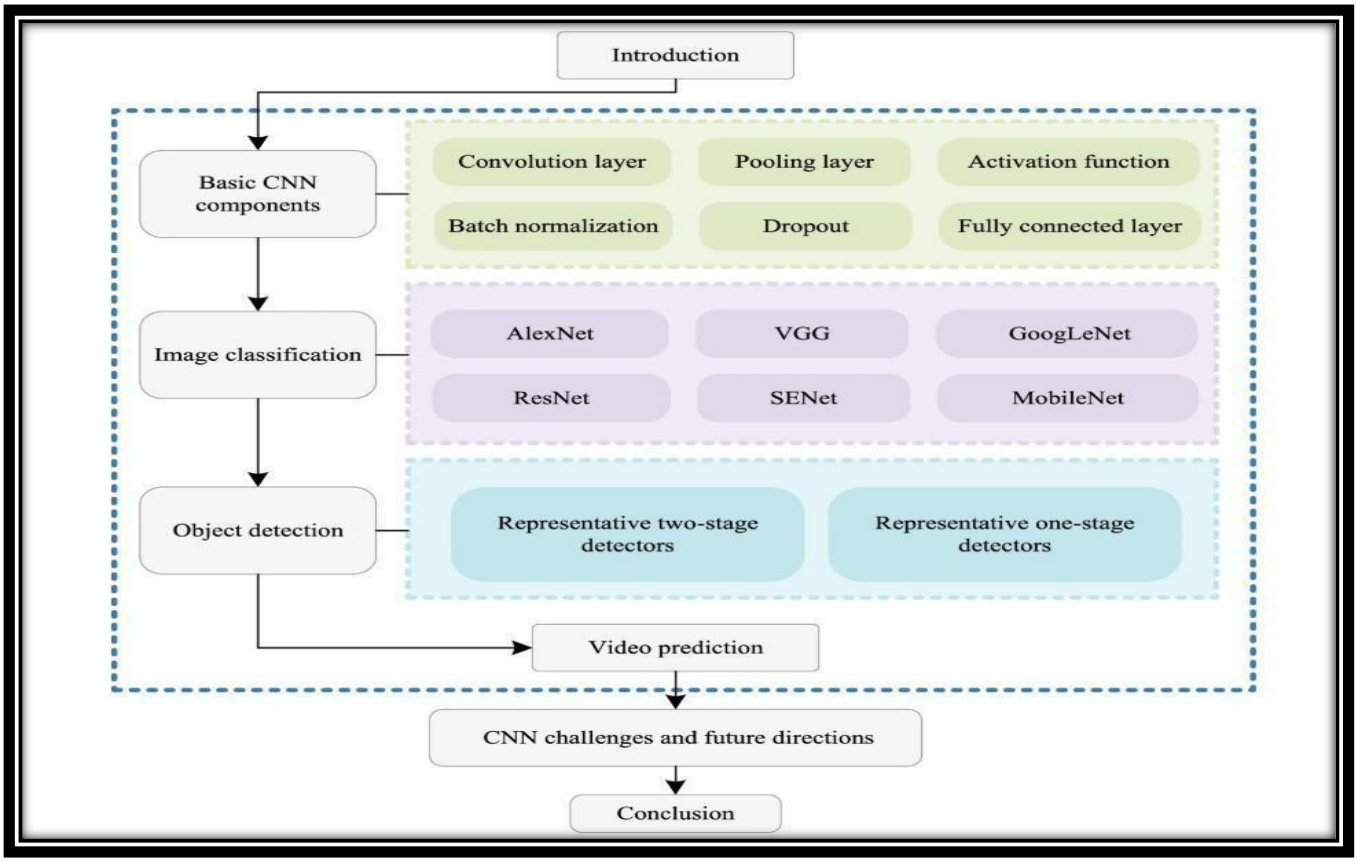


Figure 3.8: Convolutional Neural Networks (Zhao et al., 2021)

One dimensional Convolutional neural network (1D CNN) was used as a supervised learning technique, and trained on NSL-KDD dataset to detect the intrusions. Categorical features were encoded label-encoded and all the input data were normalized prior to reshaping them for CNN model. The architecture employs multiple convolutional layers (with ReLU activation, batch normalization and pooling) to automatically learn useful traffic patterns; then dense layers including dropout and l2-regularization are applied to avoid overfitting. The model was trained with the Adam optimizer and binary cross-entropy loss, validation accuracy, recall and AUC were used to select the best checkpoint. Lastly, performance was measured using accuracy, precision, recall, F1-score

and ROC-AUC scores as well as visual results such as confusion matrices and learning curves.

CNNs automatically learn useful representations from network traffic without manual feature design. Convolutional filters capture traffic attribute relationships while regularization and pooling adds generalization on noisy, high-dimensional input data. This makes CNNs an attractive supervised baseline—efficient, scalable, and performs well in distinguishing normal from malicious activity on labeled network datasets..

- **Hybrid CNN-LStm-XGBoost**

Hybrid systems combining CNN, LSTM, and XGBoost have recently attracted the attention of the intrusion detection community because of their complementary strengths. Every part of the model is dealing with a different set of challenges observed in standard single models.

CNNs are capable of learning spatial correlation and local feature interaction information from raw cascading packets, which do not require human hand-craft features or features extraction. LSTMs, in contrast, add the ability to model temporal correlations and sequence-dependent behavior which can be useful for capturing attacks on network connections such as DDoS or brute-force attempts. Last but not least, XGBoost augments the framework with gradient boosted decision trees which are able to model nonlinear decision boundaries and work well on structured tabular data while delivering high-quality regularization in order to avoid overfitting.

By integrating their advantages, it is possible for CNN–LSTM–XGBoost hybrids to avoid the weaknesses brought by individuals. CNN and LSTM layers provide deep features from different stages of the discontinuity for sequential representation, and XGBoost serves as a strong classifier to capture high-level interaction patterns in this representation. Such balance integration allows the model to achieve high accuracy and robustness under noisy or imbalanced data as well as better generalization on unseen network behaviors than classical standalone or shallow ensemble methods. Figure 3.9 shows the summary of the hybrid model (CNN-LSTM-XGBoost) steps.

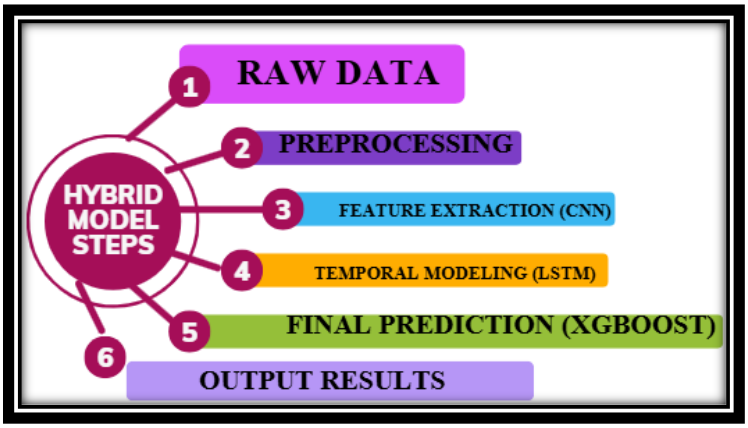


Figure 3.9: The Hybrid CNN-LSTM-XGBoost steps

- **pipeline in Machine Learning**

A machine-learning pipeline is an ordered set of stages that turns raw data into reliable predictions. It standardizes the end-to-end process—data collection and preprocessing, model selection and training, evaluation, deployment, and ongoing monitoring—so that experiments are reproducible and results are comparable. Figure 3.10 shows a typical ML pipeline.

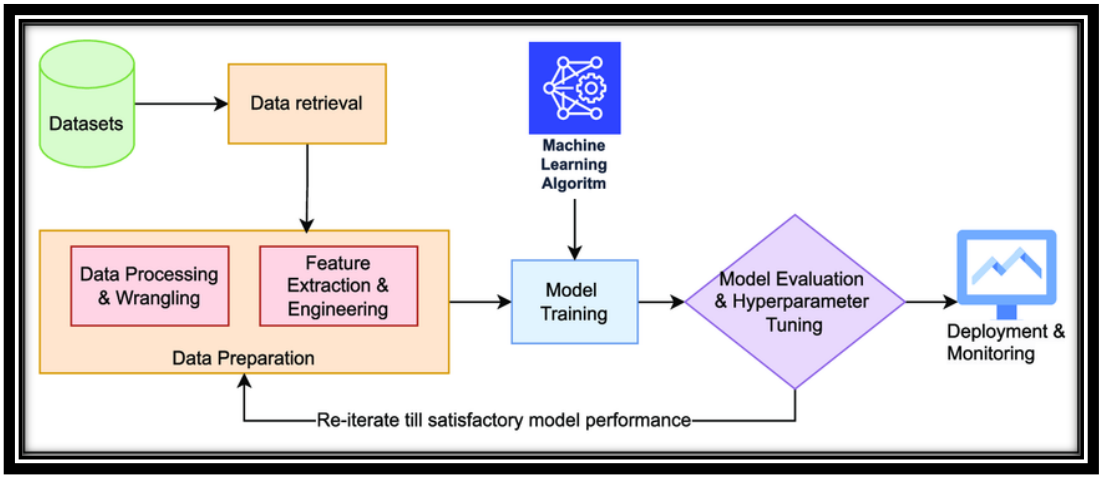


Figure 3.10: Typical ML pipeline (Barrak et al. (2022)

1. **Global Dataset Pipeline (High-Accuracy NSL-KDD Pipeline)**

The High-Accuracy NSL-KDD Pipeline was implemented in Python to achieve reliable intrusion-detection performance exceeding 0.99. This pipeline follows a

structured multi-stage workflow Figure 3.11 designed to ensure reproducibility and automation throughout the entire machine-learning process.

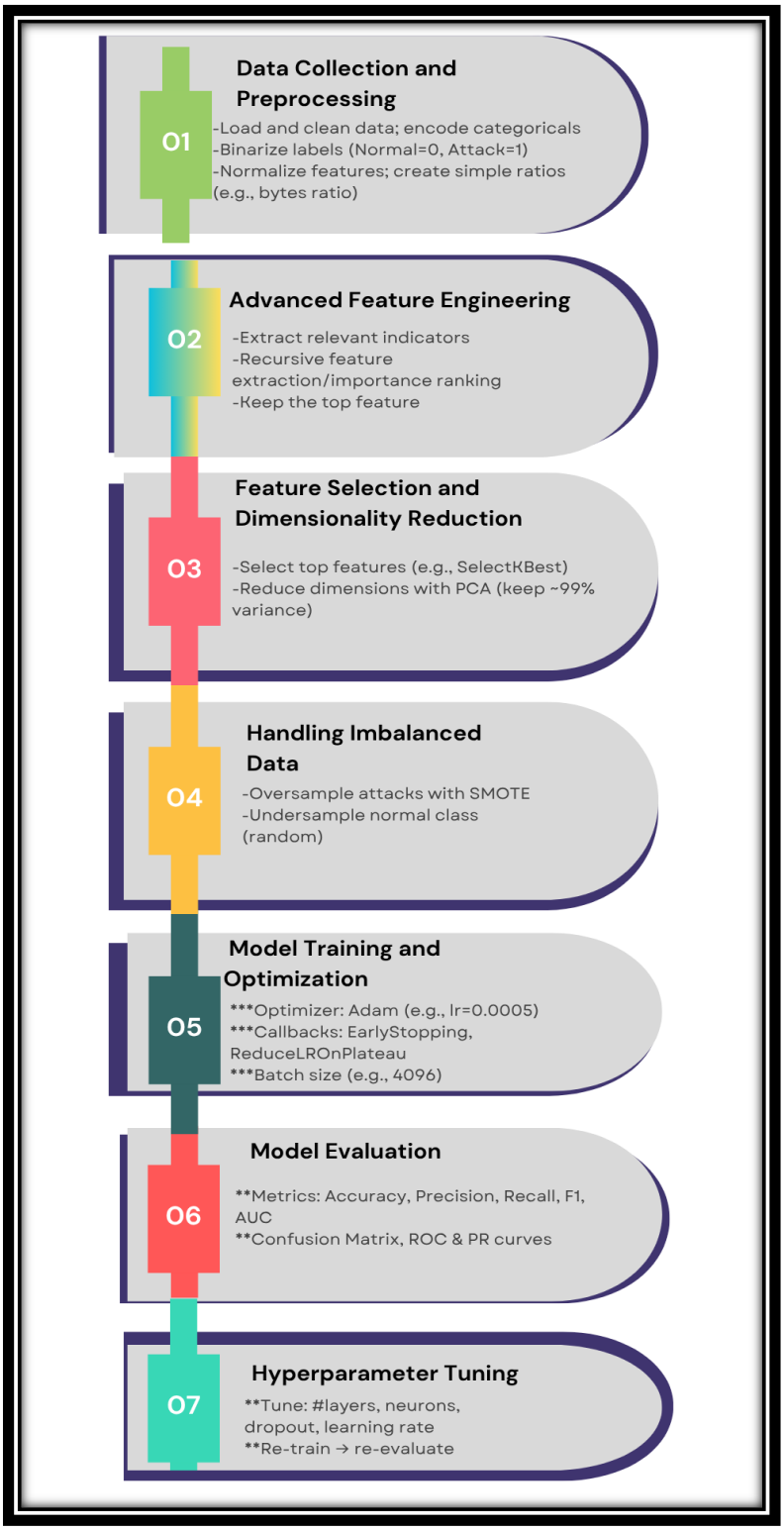


Figure 3.11: Steps of the pipeline

Figure 3.11 illustrates the sequential steps of the proposed pipeline used in this study.

Step 01—Data Collection and Preprocessing.

The pipeline starts by loading the KDDTrain+ and KDDTest+ part of NSL-KDD and merge them into one dataframe. The class labels are binarized (0=normal, 1=attack). The irrelevant features: difficulty, land, num_outbound_cmds and so forth are dropped. Categorical attributes (protocol_type, service, flag) are encoded as One-Hot features and only the top 5 most common values per column are kept to reduce dimensionality. StandardScaler is used to normalize continuous attributes.

Step 02 — Advanced Feature Engineering.

Several high-level network indicators are generated to enrich the raw data, including bytes_ratio, bytes_diff, total_packets, total_error_rate, error_rate_diff, network_activity,connection_speed,session_intensity,dst_host_activity, st_host_error_diff, and dst_host_srv_ratio. These engineered features capture nuanced traffic behaviors and improve model discriminative ability.

Step 03 — Feature Selection and Dimensionality Reduction.

The most informative 50 features are extracted via SelectKBest(f_classif), and then a Principal Component Analysis (PCA) retaining 99 % of the total variance is applied. This stage compacts the data while preserving maximum relevant information.

Step 04 — Data Balancing.

To mitigate class imbalance, SMOTE oversampling is applied with adaptive k-neighbors values according to minority-class size. Additional class-weight balancing ensures fair training for both normal and attack instances.

Step 05 — Model Training.

There is a deep feed-forward neural network comprised of several Dense–BatchNorm–LeakyReLU–Dropout layers with L1/L2 regularization.

Residual connection of 256 units helps in stabilizing the gradient flow and expedite convergence.

Training uses EarlyStopping, ReduceLROnPlateau and ModelCheckpoint callbacks monitored by validation PR-AUC to ensure generalization and avoid overfitting.

Step 06 — Model Evaluation.

After training, one makes predictions on the test split and finds the best threshold from PR curve by maximizing F1.

In the evaluation phase it reports Accuracy, Precision, Recall, F1, ROC-AUC and PR-AUC scores of the model with visual diagnosis such as Confusion Matrix, ROC Curve, PR Curve, Training curves and Feature-Importance plots.

Step 07 — Hyperparameter Tuning

The first working model is there, but you would like to tune your hyperparameters; Here we enter the game with grid search or random search. That means for example you try to simultaneously, hyper-tune, the number of layers, sizes of each layer (but that vary by your overall theme).dropout rate(s) and learning rates in order to find what combination gives you the highest accuracy.

Such systematic process generated an accurate classifier not only capable to distinguish Normal from Attack traffic but also provides a benchmark model that can be transferred towards adaptation to the local dataset.

2. Local Dataset Pipeline (High-Accuracy NSL-KDD Pipeline)

In order to ensure that the above model is applicable for real network traffic, the described high-accuracy pipeline was executed over a locally collected dataset from the Palestinian network environment.

This adopted local pipeline mirrors the structured workflow of our global NSL-KDD solution—it involves everything from data preprocessing, advanced feature engineering, feature selection, PCA compression and balancing classes to the training on a model and evaluation—but tailored to the specifics and scale of this local data.

The local dataset needed conditional feature creation (for non existing, in-between missing or renamed fields) and categorical compression (by keeping only the 'top frequency' categories per column). In addition, we optimized the F1-score according to the real local traffic distribution by applying precision–recall curve tuning for decision threshold.

3.2.6 Model Evaluation

After training the proposed model, its performance must be evaluated to ensure effective distinction between normal and anomalous network traffic. The main evaluation metrics used from these ingredients are accuracy, precision, recall, f 1 -score, AUC-ROC and the confusion matrix for the comprehensive assessment of model detection capability and reliability. (Sørbø et al, 2023).

- Accuracy

Accuracy measures how right or correct the prediction of your model was from all predictions made by the model, and its computation is defined as the ratio of correct predictions (true positives and true negatives) to all types of predictions. The value of your cash on hand after selling your property (Sokolova et al, 2009).

Accuracy = True Positives + True Negatives; Total Samples (1)

Accuracy is one measure, but it is often not very reliable with imbalanced datasets (such as in anomaly detection, where the number of normal traffic is much higher than the number of anomalous traffic. The accuracy metric is calculated as shown in Equation 1:

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Samples} \quad (1)$$

- Precision

Precision, at a more concrete level, measures the effectiveness of the model in correctly identifying anomalous (attack) traffic among all the traffic instances it predicted as anomalies. The precision metric is calculated as shown in Equation 2:

$$Precision = \frac{True\ Positives}{True\ Negatives + False\ \ Positives} \quad (2)$$

- Recall

Recall—also referred to as sensitivity or the true positive rate—represents the ability of the anomaly detection model to correctly identify anomalous traffic when it actually occurs (Vasilomanolakis et al., 2015). It is calculated as the ratio of true positives to the total number of actual anomalies (true positives + false negatives). High recall indicates that the system successfully detects most intrusion attempts or abnormal activities, minimizing the number of missed attacks (false negatives)..The recall metric is calculated as shown in Equation 3:

$$Recall = \frac{True\ Positives}{True\ Positives + False\ \ Negatives} \quad (3)$$

- F1-score

It is the harmonic mean of precision and recall; hence, it balances both metrics. When a dataset is biased, jointly adding false negatives (FN) and false positives (FP) data turns out to be very useful. The better the model performance, balancing false positives and false negatives (Powers,2011). The F1-score metric is calculated as shown in Equation 4:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

- Specificity (True Negative Rate)

Specificity is the fraction of normal traffic instances that are correctly identified as normal by the model. It is mathematically defined in Equation (5):

$$Specificty = \frac{True\ Negatives}{True\ Negatives + False\ \ Positives} \quad (5)$$

Specificity is an important measure to determine in anomaly detection systems, because it ensures that the model does not mislabel a lot of real network activities as anomalies. High specificity allows avoiding false alarms, and hence not interfering with the users, granting a reliable service and increasing trust of end-users(Chicco, 2021).

- Area Under the Receiver Operating Characteristic Curve (AUC-ROC)

The receiver operating characteristic (ROC) curve is an important tool for assessing and comparing the performance of multiple models on an anomaly detection problem using different threshold values. The AUC-ROC curve graphically represents the senability of a classifier in terms of true and false positive rates at various decision thresholds (Fawcett, 2006). The model performance was quantitatively analyzed using the Area Under the Curve (AUC), which varies between 0 and 1. A Model with a higher AUC value can differentiate normal traffic as normal (0s) and anomalous ones to be anomalous (1s) better. Thus, a higher AUC value is desirable as it indicates an increased probability of the classifier to be able to distinguish between the positive (anomalous) and negative (normal) classes. The ROC curve is a graph of the True Positive Rate (Recall) vs. the False Positive Rate to evaluate how well the model can discriminate between two classes at various threshold settings.

- Confusion Matrix

A confusion matrix details these model predictions, outlining the true positives, false positives, true negatives, and false negatives. This is key to understanding how the model categorizes. A confusion matrix helps in visualizing:

- True Positives (TP): anomalous traffic that was correctly detected.
- False Positives (FP): normal traffic that has been mistakenly reported as anomalous.
- True Negatives (TN): normal traffic correctly detected as normal
- False Negatives (FN): Real pages improperly detected as anomalous traffic.

The confusion matrix helps in identifying actionable areas, such as reducing false positives and negatives.

3.3 Summary of Chapter

In this chapter, we reviewed the entire methodology that was employed to build an intelligent intrusion detection system for detecting known and unknown network traffic with high precision. The design of the proposed work was organized through a machine learning pipeline with various key components such as data preprocessing, feature engineering, class balancing, selection of features, and reduction of dimensionality, followed by training and evaluation. The system was tested on two levels as: a global pipeline, which involved testing the tool based on software engineering pipelines, where we used the NSL-KDD dataset to assess the accuracy of our proposed system in standardized conditions and a local pipeline implementing realistic network traffic collected from the Palestinian network environment in order to evaluate how adaptable and robust our method is in real-world scenarios. Several algorithms were implemented in this generic framework, such as Autoencoder, Isolation Forest for unsupervised anomaly detection, CNN for supervised classification and a hybrid model (CNN–LSTM–XGBoost), which combines deep learning and ensemble methods to take into account spatial, temporal and nonlinearity patterns of the network traffic.Via modern methodologies like regularization, adaptive learning rates and early stopping.

In the end, this method develops a scalable and flexible intrusion detection pipeline that shows the efficacy of coupling deep learning and ensemble techniques in contemporary network security. In the subsequent chapter, we demonstrate the experimental results along with comparative analysis of our models.

Chapter Four: Results

4.1 Introduction

In this chapter, the findings and interpretation of our experimental study, which involved two types of datasets: locally gathered data obtained from the Ministry of Education and widely acknowledged NSL-KDD, are described. The proposed methods were tested in a realistic data environment by using standard benchmarking techniques, and it was proven that they are scientifically sound as well as practically maintainable.

Preprocessing for and experimentation with both data sets were carried out diligently. For the NSL-KDD dataset, all the numerical features were normalized, and dimensionality reduction using PCA was performed in order to stabilize training and mitigate redundant feature correlation. In the local dataset, more sophisticated feature engineering methods, e.g., bytes_ratio, total_packets, bytes_diff, and error_rate_diff, were adopted to increase the expressiveness and discrimination power of the data. SMOTE oversampling was used to alleviate the problem of class imbalance in the local dataset and enhance minority class representation.

All experiments were performed with the same unified tabular classification pipeline as in Table 4.1 (for methodological consistency, performance comparability, and full reproducibility) on both datasets (overviewed in the next sections).

Table 4.1: Experimental Environment

Component	**Specification/Tools**
Programming Environment	Jupyter Notebook (Anaconda)
Operating System	macOS (Apple MacBook Pro)
Processor (CPU)	Apple M1 Pro
Memory (RAM)	16 GB
Storage (SSD)	256 GB

The input models consist of both supervised and unsupervised paradigms. The autoencoder obtained was used for unsupervised anomaly detection by considering the reconstruction errors. In this work, however, the CNNs were employed to initially find

structural and spatial motifs within network traffic at first, and these are then learned by the LSTM network in the time domain. We then integrated XGBoost to take advantage of these learned spatial and temporal representations jointly, to leave us with a hybrid model that utilizes the benefits of all three methods towards more robust anomaly detection. The isolation forest was an additional unsupervised reference that isolated anomalies by recursively partitioning. We also used a supervised approach to train a model as base line for traditional machine learning algorithms: HighAccuracyNSLKDDPipeline.

The models were trained for an adequate number of iterations to guarantee convergence. In the case of unsupervised methods, hyperparameters were tuned on a validation set while maximizing the F1 score without considering labeled attacks. The performance was measured with accuracy, precision, recall, F1-score, ROC-AUC and PR-AUC , which are robust in the class-imbalance issue.

We report results per dataset, and compare them with confusion matrices, ROC/PR curves, and feature-importance analyses when relevant. The analysis includes critical points such as the precision–recall compromise, generalization vs. overfitting and the compromise between efficiency and robustness.

The chapter presents an understandable and reproducible basis for intrusion detection analysis on the whole, establishing a direct relationship between quantitative findings and practical considerations in terms of models to be selected under different data-attack settings.

4.2 Deep Learning and Machine Learning Techniques for Global Dataset Results

In this subsection, we show how the tested models perform when they are trained to detect normal and abnormal network traffic by using the NSL-KDD repository. The comparison indicates how such supervised and unsupervised approaches perform in the identical experimental setting. Through a detailed comparison of their detection accuracy, error patterns and generalization capability, the findings provide practical insight on strengths and limitations of each model in practice, which are exacerbated when dealing with high-dimensional data and diverse traffic dynamics.

4.2.1 Enhanced Autoencoder

The improved autoencoder for NSL-KDD anomaly detection is constructed as an unsupervised reconstruction technique that captures the structure of normal traffic

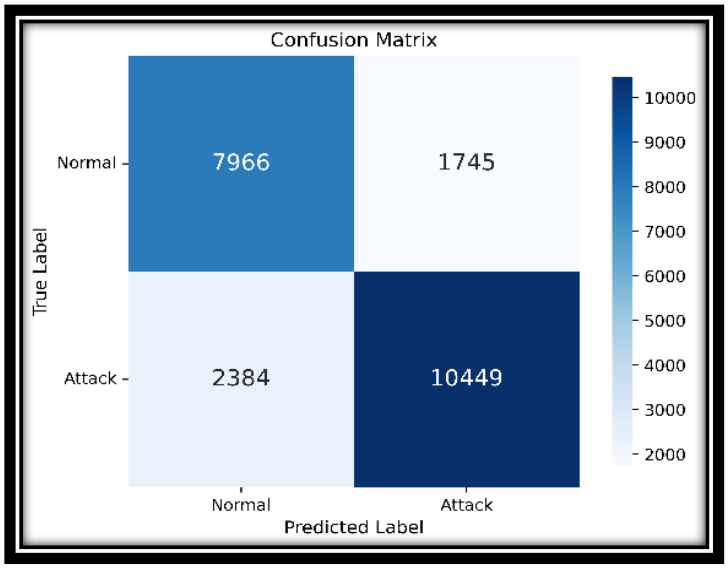
manifold, whereas out-of-manifold anomalies (attacks). After loading the KDDTrain and KDDTest with binarized labels, the pipeline then normalizes numerical features and one-hot encodes categorical ones through a column transformer method of chain and transformers, PCA in order to hit around 0.95 part of variance preserved but reducing dimensions for stable training process. The model used is the deep symmetric encoder–decoder with batch normalization, L2 regularization and dropout followed by a bottleneck layer capturing benign traffic structure. Even in this case, it is only trained on normal samples so by using the Nadam optimizer (Nadam optimize use) with a low degradation rate and apply callbacks like early stopping and learning-rate reduction we guarantee a proper convergence as well as avoid overfitting.

During the test, reconstruction error is estimated for each sample with higher values corresponding to anomalies. The best performing non-attack annotation mechanism produces a data-driven threshold based on the maximum F1 score (or geometric mean of precision and recall) to perform binary classification without the need for attack labels at training time. Performance of the improved autoencoder on the NSL-KDD is demonstrated using several performance metrics and visual diagnostics. The discriminative ability of the best performing model according to the classification report (in Table 4.2), which has an accuracy of 0.82, can be also be observed.

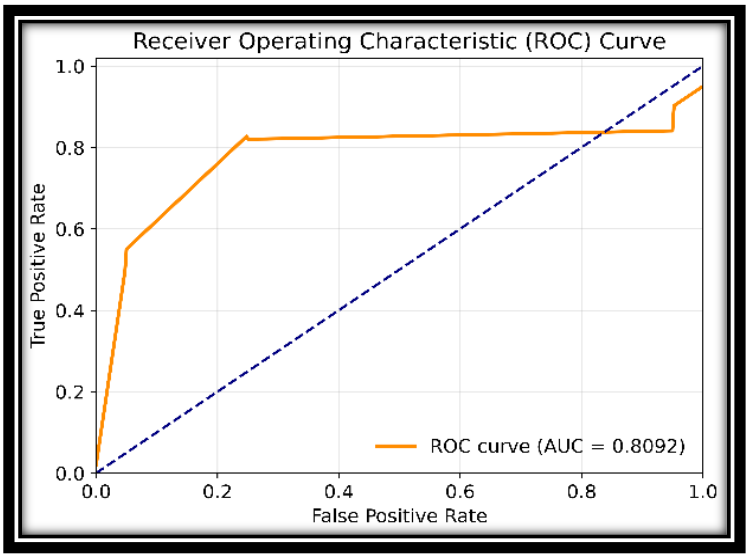
Table 4.2: Classification Results Using Autoencoder on The NSL-KDD Test Set

Class	**Precision**	**Recall**	**F1-score**	**Support**
Normal (0)	0.77	0.82	0.79	9711
Attack (1)	0.86	0.81	0.84	12833
Accuracy			0.82	22544
Macro avg	0.81	0.82	0.81	22544
Weighted avg	0.82	0.82	0.82	22544

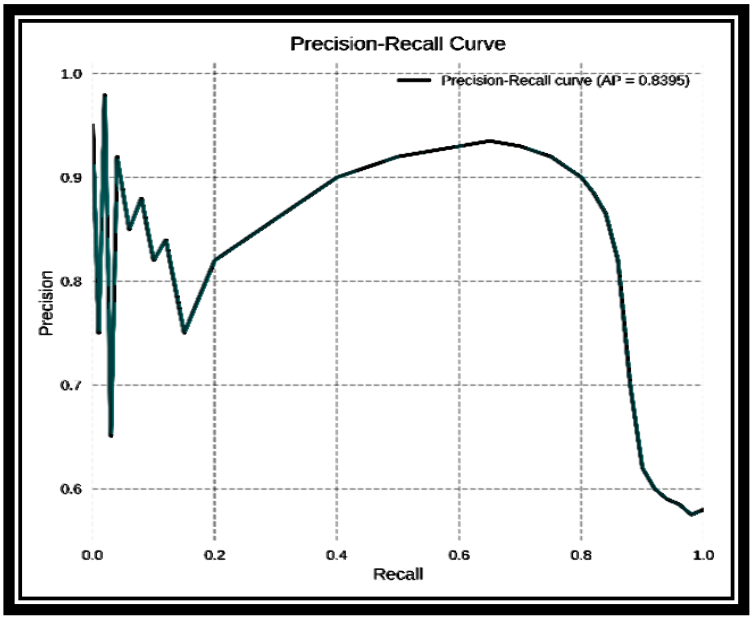
The evaluation includes visual diagnostics such as the confusion matrix, ROC/PR curves, and reconstruction-error distributions. these analyses illustrate how the autoencoder differentiates between normal and anomalous traffic using reconstruction-error thresholds. An overview of the experimental results of the model are depicted in Figure 4.1.



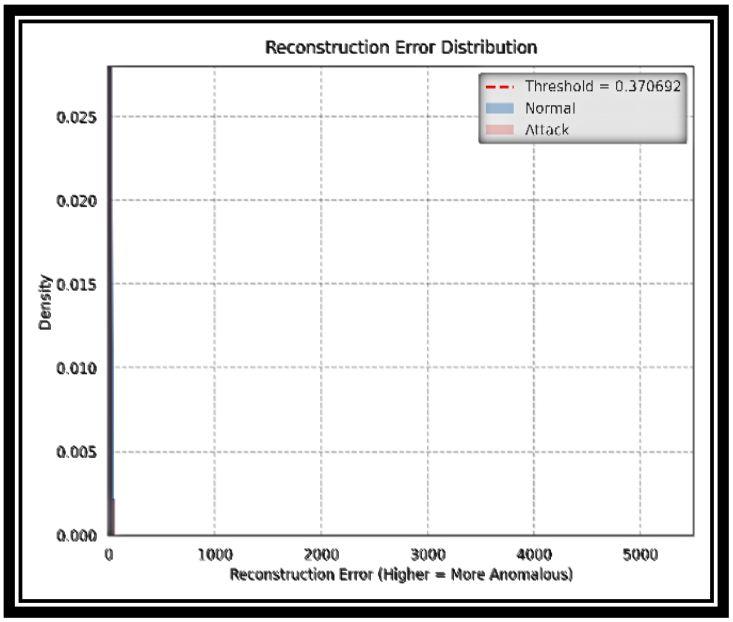
(A)



(B)



(C)



(D)

Figure 4.1: Comprehensive Performance Visualization of the Model

The confusion matrix is displayed in Figure 4.1(A) It can be seen that most attack samples were correctly classified, while a considerable number of normal samples were misclassified as attack and vice versa. The AUC for the ROC curve is 0.8092 in Figure 4.1(B), which also shows acceptable performance to classify normal and attack traffic. Precision-Recall (PR) curve in Figure 4.1(C) exemplifies the difficulty to keep precision at a higher level when recall improves, i.e., the trade-off between sensitivity and specificity. Finally, Figure 4.1 (D) shows the distribution of reconstruction error, which demonstrates that there is an overlapping area between normal and abnormal samples according to reconstruction errors, which indicates the inherent trade-off between FP and FN.

In summary, these results demonstrate the value and limitations of autoencoder—namely that it was able to dynamically provide detection of anomalies without labeled attack data—a valuable property for dynamic or new contexts—but its accuracy is similarly reduced compared to supervised methods.

4.2.2 CNN-Based Intrusion Detection System

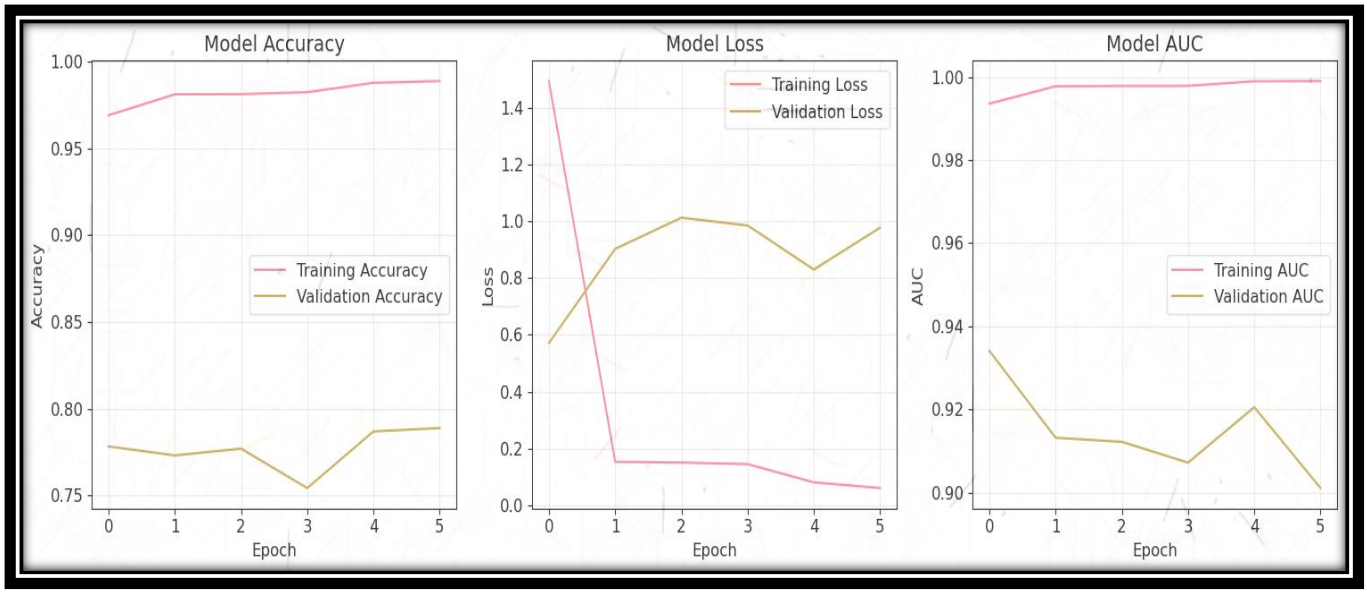
CNNs based intrusion detection system is implemented with one dimensional CNN and the NSL-KDD dataset to categorize network flow as normal or attack. Preprocessing entails encoding discrete variables, standardizing continuous features, and converting them to CNN format. The architecture includes a combination of convolutional and pooling layers for pattern extraction (with dropout for regularization), followed by dense layers for binary classification. We use Adam and binary cross-entropy loss for optimization, and prevent overfitting using early stopping, learning-rate scheduling, and checkpoints. After training the model and testing it against the dataset, we're going to summarize results within a classification report, which will show metrics such as precision, recall, F1 score and accuracy. The corresponding report is presented here in Table 4.3.

Table 4.3: performance metrics of the CNN model on the NSL-KDD Test Set

Class	**Precision**	**Recall**	**F1-score**	**Support**
Normal (0)	0.67	0.97	0.79	9711

Attack (1)	0.97	0.63	0.76	12833
Accuracy			0.78	22544
Macro avg	0.82	0.80	0.78	22544
Weighted avg	0.84	0.78	0.78	22544

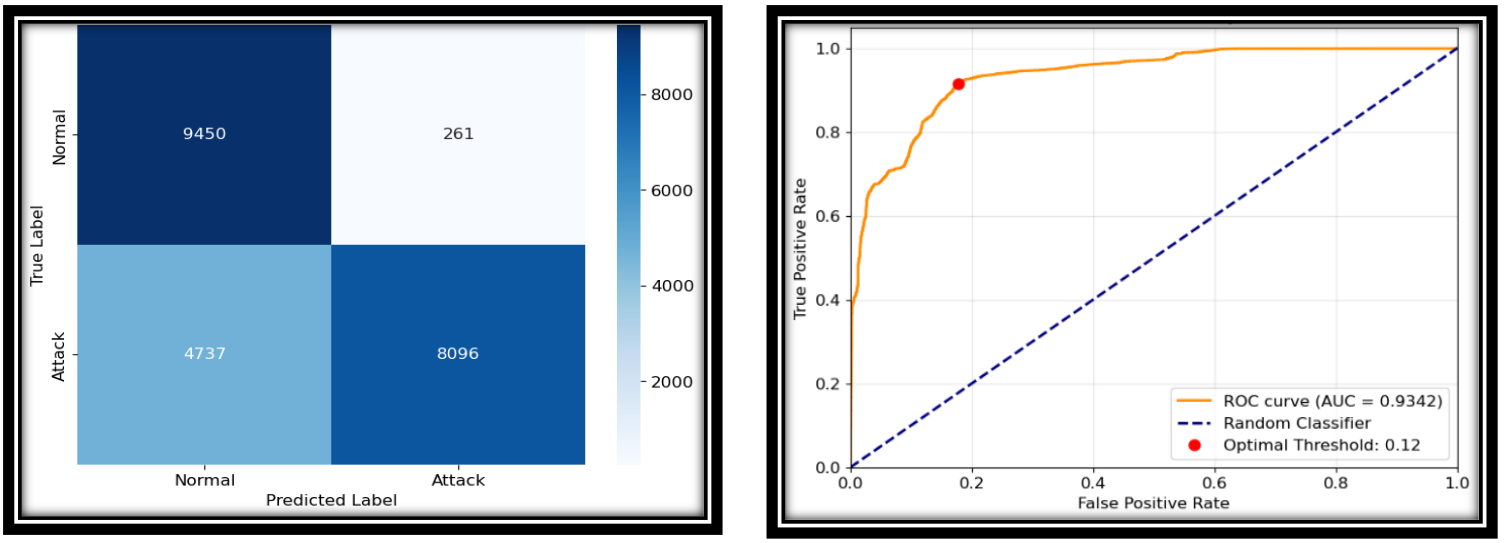
The training history clearly indicates that the CNN model overfits rapidly. As shown in Figure 4.2.



(A) (B) (C)

Figure 4.2: Training and validation accuracy, loss, and AUC curves of the CNN model

Figure 4.2 (A), the training accuracy rises sharply and stabilizes near 0.99 after only a few epochs, while the validation accuracy plateaus around 0.78, showing limited generalization. Similarly, Figure 4.2 (B) shows that the training loss decreases rapidly and remains very low, whereas the validation loss remains high and fluctuates, confirming poor validation performance. Lastly, as we see that training AUC gets close to 1.0 and validation AUC fluctuates around between 0.90 - 0.92 (Figure 4.2 C). More generally, these curves all show clear cases of overfitting: the model has learned the training data rather than a useful pattern.



(A) (B)

Figure 4.3: Confusion Matrix and ROC Curve of the CNN-Based Intrusion Detection System

The confusion matrix and roc curve presented in Figure4.3 shows a closer look at the classification of CNN-based IDS. Figure 4.3 (A) The effective ability of the model in normal traffic detection, which correctly classified 9,450 instances as normal and misclassified only 261 of them into attacks or false positives. On the attack side, the system detects correctly 8,096 attack samples and incorrectly classifies them as normal type (false negatives) there are 4,737.

This asymmetry reveals a trade-off, where while the normal traffic is detected with high confidence, up to 50% of the attacks are not detected. As a result, the model has strong precision on normal samples but poor recall in attack detection, thus leading to potential security risks (false negatives) in real-world applications where false alarms are tolerable.

The ROC curve in Fig. 4.3(B) gives a broader perspective, showing high AUC of 0.9342 for the discriminative ability between normal class and attack classening (see Section 5). The optimal cutpoint, determined as 0.12, defined the balance between sensitivity and specificity. This threshold provides a convenient interface for deploying to different settings: in more conservative environments, with a higher attack detection accuracy requirement, one could afford to have high recall rates and instead balance the overall quality of detection against all other involved trade-offs.

The CNN-based IDS showed good discriminative power as indicated by the high AUC value. It successfully classified normal traffic and showed great classification accuracy overall. Unfortunately, the significant number of false negatives reveals a major limitation: species of attacks are often discarded. This limitation highlights the necessity of alternative (or supplementary) mechanisms that would improve detection, especially with respect to undetected attacks.

4.2.3 Isolation Forest for Anomaly Detection on NSL-KDD

The Isolation Forest is an unsupervised learning method that separates the data space, out of which normal samples get isolated after a small number of splits and they will have larger anomaly scores..

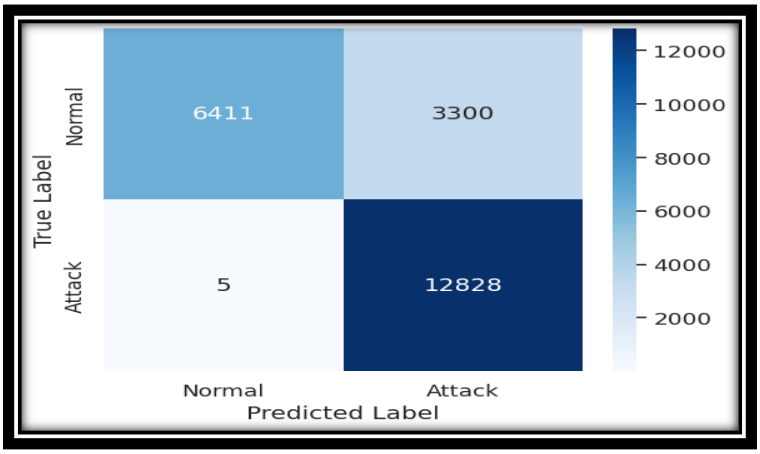
In the present study, the Isolation Forest was trained only on normal samples from the training split, allowing it to learn the baseline distribution of benign network traffic. After training, anomaly scores were computed for all test samples using the negative of the model's score samples output, so that higher scores correspond to more anomalous connections. A decision threshold was then fixed at the 90th percentile of the anomaly-score distribution on the test set: samples above this threshold were labeled as attacks, while those below were treated as normal. Based on these predicted labels, a classification report summarizing precision, recall, F1-score, and overall accuracy was generated; its results are reported in Table 4.4. In addition, a confusion matrix, a precision–recall curve, and anomaly-score histograms for normal versus attack samples were plotted to further assess the detector's behavior.

Table 4.4: performance metrics of the Isolation Forest model on the NSL-KDD Test Set

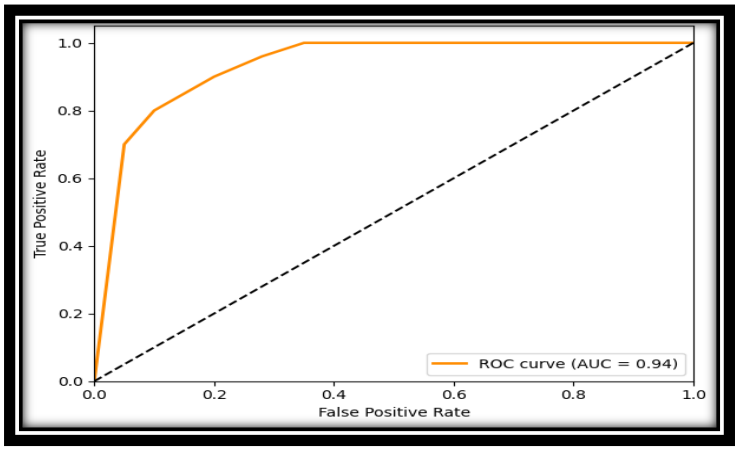
Class	**Precision**	**Recall**	**F1-score**	**Support**
Normal (0)	100	0.66	0.80	9711
Attack (1)	0.80	100	0.89	12833
Accuracy			0.85	22544

Macro avg	0.90	0.83	0.84	22544
Weighted avg	0.88	0.85	0.85	22544

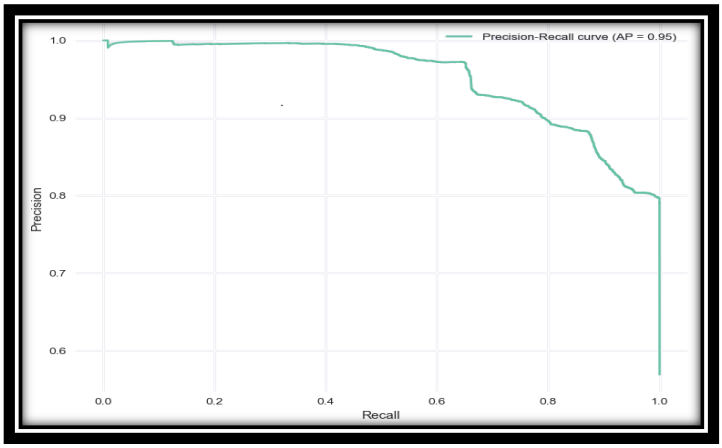
The classification report in Table 4.4 summarizes the quantitative performance of the Isolation Forest model on the NSL-KDD test set. The model achieved an overall accuracy of 0.85, with a macro-average F1-score of 0.84, reflecting a good ability to distinguish between normal and attack samples. While the model reached nearly perfect recall for attack instances (Recall = 1.0), the low precision for normal traffic (Precision = 0.66) shows that there are more FP cases classified in sample combinations. To further visualize these results, Figure 4.4 displays the confusion matrix, ROC and precision-recall curves, as well as the histogram of anomaly scores.



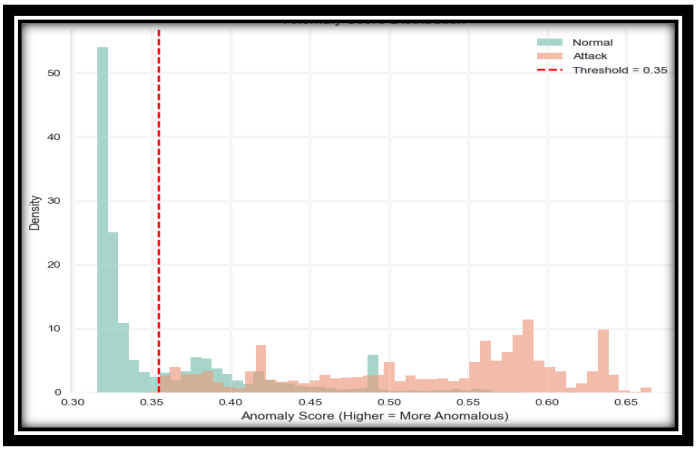
(A)



(B)



(C)



(D)

Figure 4.4: Model Evaluation and Anomaly Score Analysis for Isolation Forest

We consider these quantification results as revealing the strengths and weaknesses of our approach. This can be inferred based on confusion matrix of the model, Figure 4.4 (A), where the recall, i.e., the number of true positive cases out of total were almost perfect with false negatives equal to 5. This shows its great effectiveness in handling almost all attacks. But the model is also failed to recognize 3,300 common cases as attack and correctly identified only 6,411 normal samples. This imbalance illustrates the basic trade-off: while having an exceptional recall for attacks, there is a relatively high false positive rate for normal traffic.

These findings are also confirmed by the ROC curve in Figure 4.4 (B) which reveals high discrimination power with an AUC of 0.94 and, thus, also robustness across different decision thresholds. Figure. 4.3 (C) the mean average precision is 0.95 which indicates a good trade-off between recall and precision for attack detection. Lastly, the anomaly score distribution in Figure 4.4 (D) gives us some added insight into model reasoning: a threshold of around 0.35 serves as a decent split-point between normal and attack traffic samples. Most normal samples are below this threshold, whereas most attack samples are above it, indicating that (i) the selected threshold seems reasonable and (ii) the two classes appear to be largely separable in this score space.

The analysis of feature importances improves the interpretability of the model by showing what attributes contribute to its decisions. The findings revealed that dst_host_rerror_rate is the most influential feature, followed by the service-based features such as service_pop_3, service_private and service_telnet. Negative-based parameters such as rerror_rate, srv_serror_rate and dst_host_srv_serror_rate exhibit high positions in the ranked list, meaning that negative or abnormal connection replies are significant measures for intrusion. Traffic-level features (src_bytes, duration, and the flag flag_SF) also play a role in separating normal from malicious behavior. Figure 4.5 visually confirms these findings by showing the top 15 features from the Permutation Importance analysis, with the host error rate clearly standing out as the most influential factor.

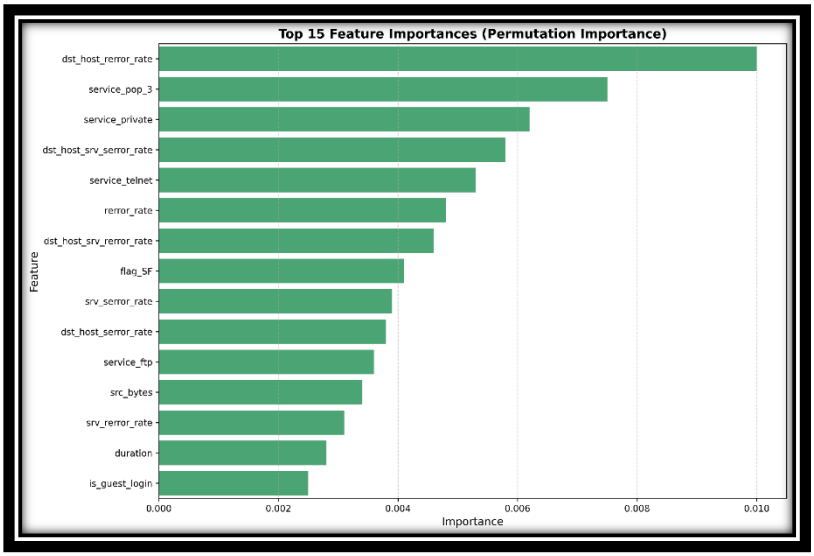


Figure 4.5:Top 15 Feature Importance Based on Permutation Importance

In general, it is shown that the Isolation Forest approach achieves very good attack discovery in terms of recall, and uncovers that almost no suspicious activity is missed. The cost, however, is the higher percentage of false positives over normal traffic, which may generate too many unnecessary alerts. This is what makes the method especially useful in situations where failure to detect attacks represents a significantly larger risk than encountering extra alarms. Finally, for balancing classes of the positive and negative observations, Isolation Forest can be integrated with an existing supervised or semi-supervised intrusion detection approach in order to mitigate false alarms without degrading its powerful anomaly detection power.

4.2.4 Hybrid CNN-LSTM and XGBoost Model

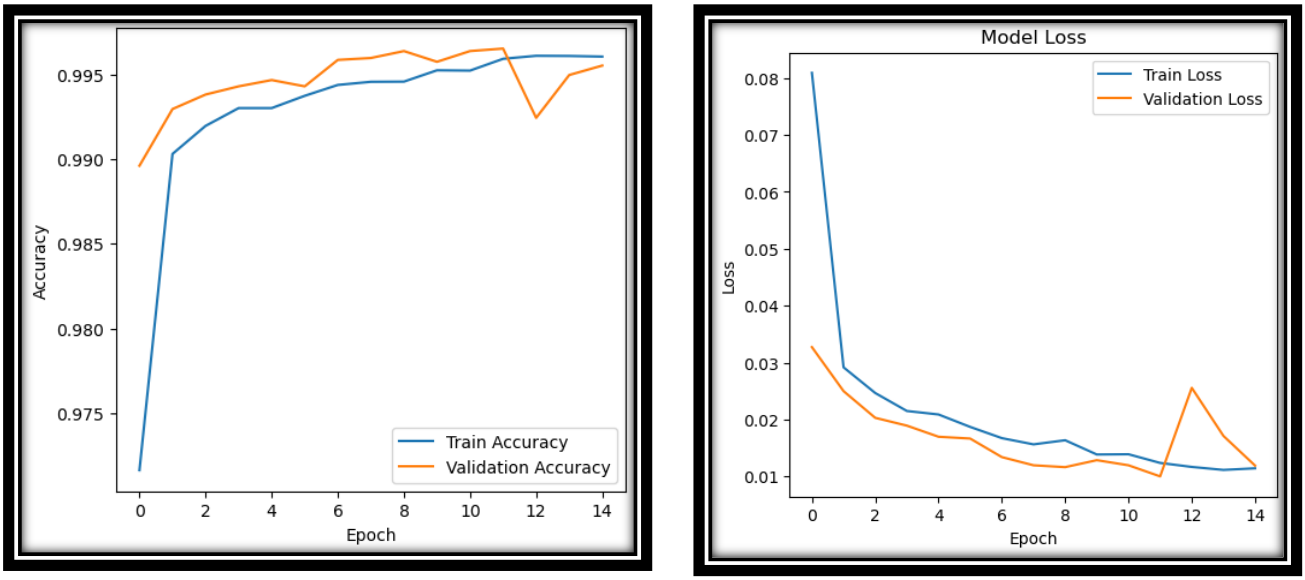
This experiment is a critical step for examining the performance of the hybrid detection model in which the NSL-KDD dataset is used. Several preprocessing methods such as one-hot encoding of the categorical features, scaling of numerical features and proportioning class imbalance with SMOTE are employed. The database also included engineered features which reinforced our ability to predict on the model. Visualization techniques, PCA indicated that these pre-treatments had an important effect on class discrimination. Then a CNN-LSTM model was developed with an XGBoost-based hybrid framework, where the methods were fused to accustomer each other. The training process was regularized by using batch normalization, dropout and early-stopping to avoid overfitting. The detailed experimental results (presented in this section)

demonstrate that the ensemble method obtained better and more stable performance than single models, revealing a good potential to be an effective way for intrusion detection.

Table 4.5: Performance Metrics of the Hybrid CNN-LSTM and XGBoost Evaluation on the NSL-KDD Test Set

Class	Precision	Recall	F1-score	Support
Normal (0)	0.69	0.98	0.81	9711
Attack (1)	0.98	0.66	0.79	12833
Accuracy			0.80	22544
Macro avg	0.83	0.82	0.80	22544
Weighted avg	0.85	0.82	0.80	22544

The classification results of the Hybrid CNN–LSTM and XGBoost model on the NSL-KDD test set are shown in Table 4.5. The general accuracy was 0.80, which suggested that the performance was acceptable but could be improved. For regular traffic, the model obtained a precision of 0.69 and recall of 0.98, indicating low false alarm. In comparison, for angry traffic, precision was 0.98, recall was 0.66 just suggesting that while the predictions of attacks were accurate almost one third of attack are missed. The macro-averaged F1-score was 0.80, indicating robust but somewhat imbalanced detection performance. In total, the Hybrid model shows high trustiness to reduce false positives, but more tuning is needed due to low sensitivity toward attacks.



(A) (B)

Figure 4.6:Training and Validation Accuracy and Loss Curves of the Hybrid Model on the NSL-KDD Dataset.

Figure 4.6 Training and validation curves demonstrates that the hybrid model converged fast and generalised well. The train and validation accuracies increase quickly in the first epochs segment, where they are both similar to each other, meaning Figure. 4.6 (A).overfitting-free learning. Likewise, in Figure 4.6 (B), we observe both training and validation losses decreasing in a smooth manner to low similar values. These patterns show that the model attains a balance in learning and generalization, which suggests that it is able to form an appropriate basis for further performance assessment by reporting precision, recall, and F1 scores.

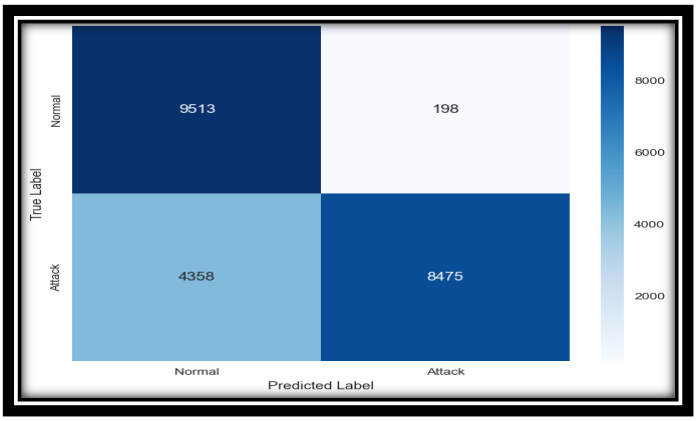


Figure 4.7: Confusion Matrix for Hybrid Model on the NSL-KDD Dataset

The hybrid model is able to discriminate normal and attack traffic, as can be seen in the confusion matrix of Figure (4.7). Of the 9,711 normal samples, 9,513 were correctly identified and there was low False positive rate . In contrast, 8,475 among 12,833 attack samples were False positive rate and only 4,358 were misclassified as normal indicating a relatively small false negative rate. In general the hybrid had high normal traffic precision but slightly low attack recall, which means it must be further optimised to increase sensitivity with no great loss of accuracy.

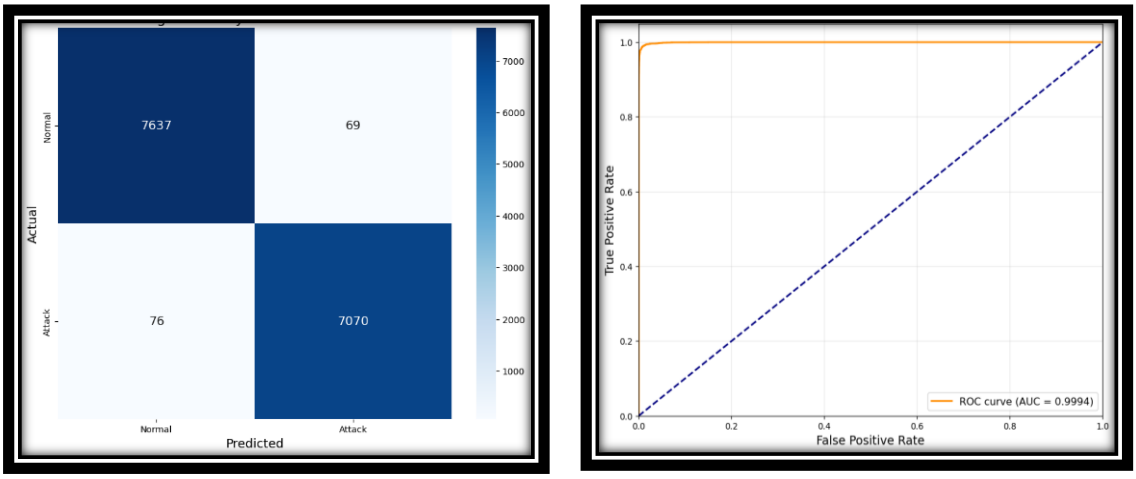
4.2.5 HighAccuracyNSLKDDPipeline

The HighAccuracyNSLKDDPipeline was designed as a hybrid solution combining state-of-the-art feature engineering, data balancing, and ensemble classification. It aims to overcome the drawbacks of traditional intrusion detection mechanism with a high accuracy rate and robust recall for various types of traffic. This property makes it very appropriate for real life applications where the former is low and the latter is kept high.

To fully examine its performance, we used several evaluation measures including the classification report in Table 4. 6 shows the classification report.

Table 4.6: Performance Metrics of the High Accuracy Pipeline Evaluation on the NSL-KDD Test Set

Class	**Precision**	**Recall**	**F1-score**	**Support**
Normal (0)	0.99	0.99	0.99	7706
Attack (1)	0.99	0.989	0.989	7146
Accuracy			0.99	14852
Macro avg	0.99	0.99	0.99	14852
Weighted avg	0.99	0.99	0.99	14852

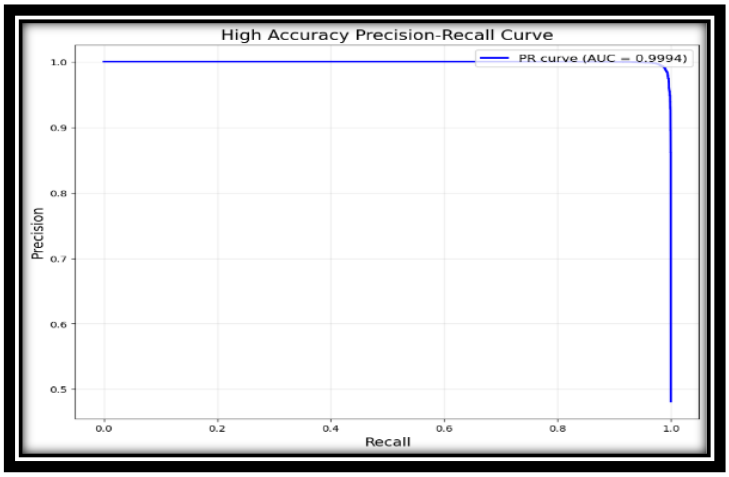


(A) (B)

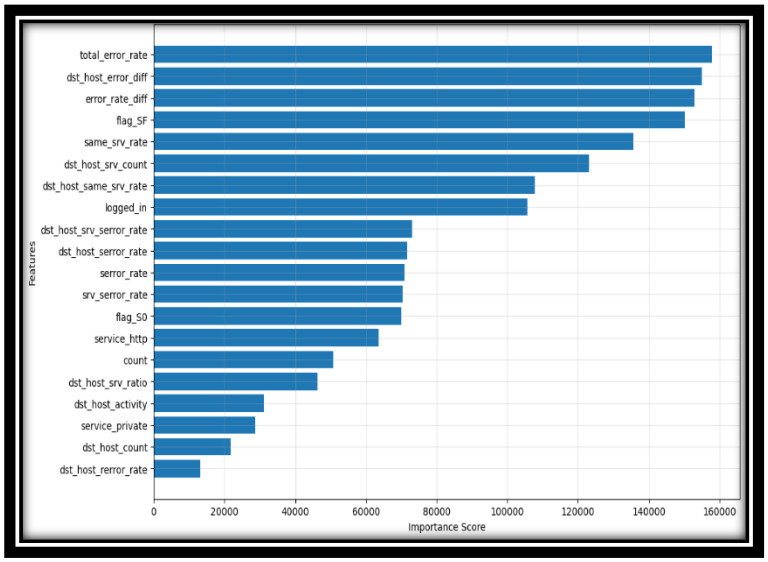
Figure 4.8: Confusion Matrix of the HighAccuracyNSLKDDPipeline Model

A brief summary of the classification results is shown in Figure 4.8. we can see that 0.991 of the normal traffic is correctly averaged as normal, in contrast, only 0.90 of it is wrongly averaged as attacks. Simultaneously,0.989 of the attack cases were accurately classified, while a few (1.06%) were misclassified as normal traffic. These findings demonstrate that the model can successfully reduce false positives as well as false negatives, a desirable characteristic for intrusion detection systems where both types of errors are likely to have great operational risk.

The ROC curve was used, as illustrated in Figure 4.8 (B), to evaluate the discriminative ability of the model. The ROC curve plots the true positive rate against the false positive rate, and in this example, it occupies the upper-left corner of the graph. Similarly, the corresponding AUC value of 0.9994 shows an almost perfect separation between normal and malicious traffic. Compared with the diagonal line, which was obtained from pure chance, we can observe that the ROC curve of our model further demonstrates its capability to obtain both relatively high sensitivity and a very small false alarm probability.



(A)



(B)

Figures 4.9: Performance Evaluation and Feature Importance of the HighAccuracyNSLKDDPipeline Model

The Precision–Recall curve in Figure 4.9(A) confirms the model's strong performance on the imbalanced validation set, maintaining precision close to 1.0 across most recall levels with an AUC of 0.9994. This indicates that the classifier detects nearly all attacks while keeping false positives extremely low.

To better understand the model's behavior, a feature-importance analysis was conducted. As shown in Figure 4.9(B), the most influential features include error-based and host-activity attributes—such as total_error_rate, dst_host_error_diff, and error_rate_diff—alongside important service- and protocol-related indicators (e.g., flag_SF, same_srv_rate, service_http).Some of the engineered features also demonstrated high scores, confirming the importance of adding new generated features

to enhance separation between classes. Finally, the confusion matrix, ROC and PR curves in conjunction with feature-importance analysis show that our HighAccuracyNSLKDDPipeline model is powerful. The accuracy, precision, and recall of the model are high and depend on a small set of meaningful and nonredundant features.

A lot of performance improvement is due to the engineered ratio- and difference-based features (bytes_ratio, bytes_diff, error_rate_diff, dst_host_error_diff), which clearly improved class separability. Further enhancements were realized by SMOTE for data balance and the use of PCA for dimension reduction that stabilised our model and reduced noise.

The last assessment results indicate that the pipeline has proved to be effective, generalizable and deployable in real world intrusion-detection scenarios where both highaccuracy coverage reliability is required.

4.3 Deep Learning and Machine Learning Techniques for Local Dataset Results

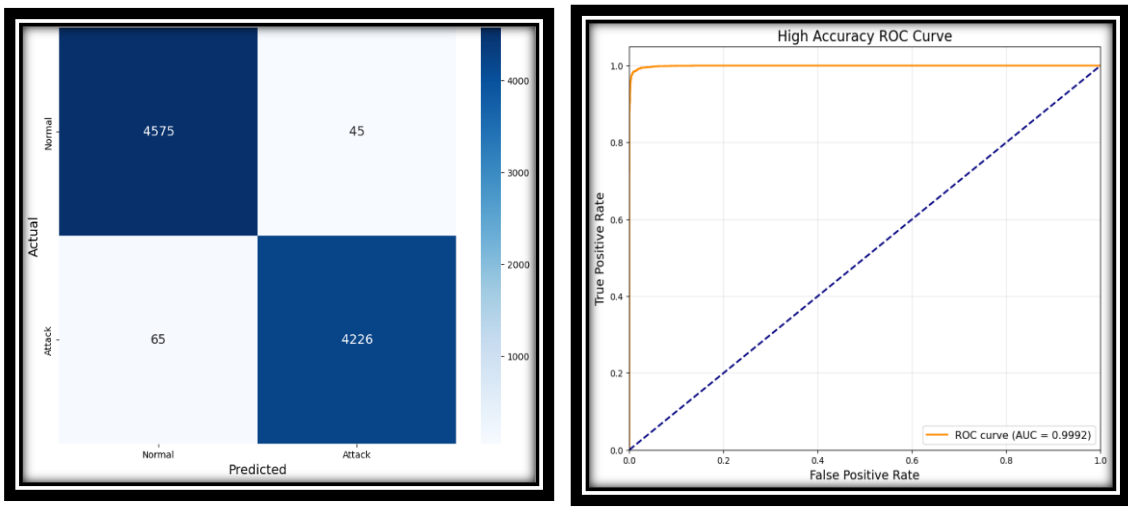
Testing The results of the proposed HighAccuracyNSLKDDPipeline on the local dataset shown that it achieves good anomaly-detection capability. Following extensive pre-processing such as feature engineering, categorical encoding and scaling along PCA and SMOTE oversampling -the deep network exhibited steady convergence as well as accurate classification. To have a full assessment of its performance, we used several robust measures to evaluate the performance as follows: we first conducted the classification report Table 4.7 shows the classification report.

Table 4.7: Performance Metrics of the High Accuracy Pipeline Evaluation on the local Test Set

Class	**Precision**	**Recall**	**F1-score**	**Support**
Normal (0)	0.986	0.990	0.988	4620
Attack (1)	0.989	0.984	0.987	4291
Accuracy			0.988	8911

Macro avg	0.987	0.987	0.987	8911
Weighted avg	0.987	0.987	0.987	8911

The classification report in Table 4.7 shows that the proposed pipeline achieved an accuracy of 0.988, with both normal and attack classes obtaining precision, recall, and F1 values close to 0.99. The balanced metrics and high macro averages indicate strong and stable performance across the dataset, confirming the model's effectiveness in distinguishing between normal and malicious traffic. Figure 4.10 presents the confusion matrix for the local dataset.



(A) (B)

Figures 4.10: Collectively summarize the performance of the proposed

HighAccuracyNSLKDDPipeline. Figure 4.10A is the confusion matrix, which indicates that the model can detect with high accuracy and also very little false alarm and missings. The ROC curve Figure 4.10 (B) is close to the upper-left corner with an AUC of 0.9992, which demonstrates a great capability for distinguishing attack traffic from normal traffic by the model.

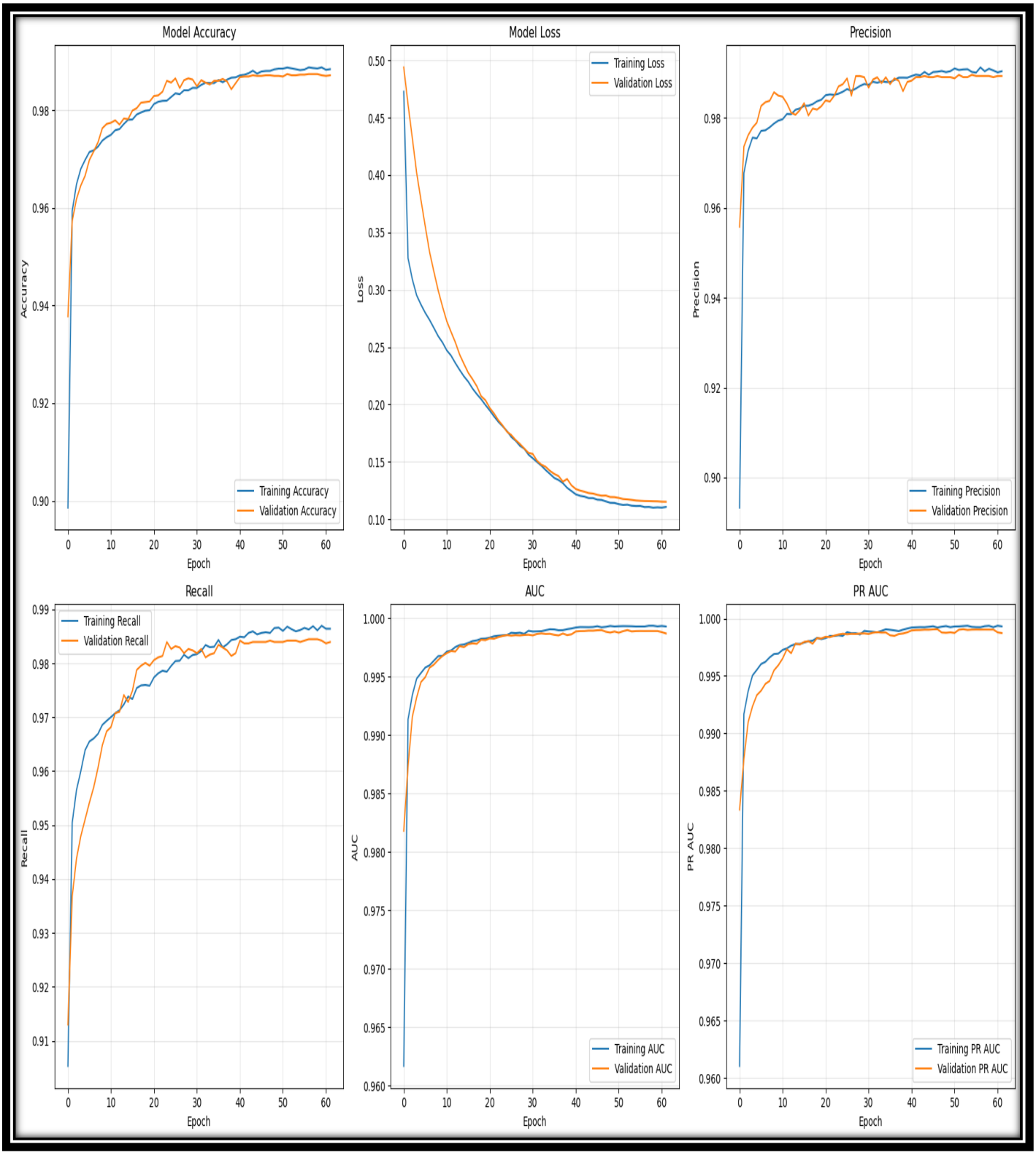


Figure 4.11. Training and Validation Metrics of the HighAccuracyNSLKDDPipeline Model

Figure 4.11 shows the training and validation curves, which approach each other slightly with little gap, suggesting that there is good generalizationand we are not overfitting; the PR-AUC stays close to 1.0 even though classes are imbalanced. All these results are providing good validation of reliability and robustness of our pipeline in real-world intrusion detection.

Feature-ranking analysis is consistent with the former evaluation results and explains influence of individual variable to detection ability of the model. Error-related features—i.e., total_error_rate, dst_host_error_diff or error_rate_diff—as well as service- and host-activity indicators such as same_srv_rate, dst_host_srv_count, and connection-state flags like flag_SF and flag_S0 were among the most highly relevant ones.

These results indicate that outliers in the local dataset consist mainly of significantly higher error rates, traffic that is dominated by a small number of machines or services, as well as unusual connection states. Even if the ranking is computed as a step before PCA, the good separability detected between these variables sheds light on why the downstream classifier fits it with near-perfect accuracy. The most significant feature is shown in Figure. 4 .12.

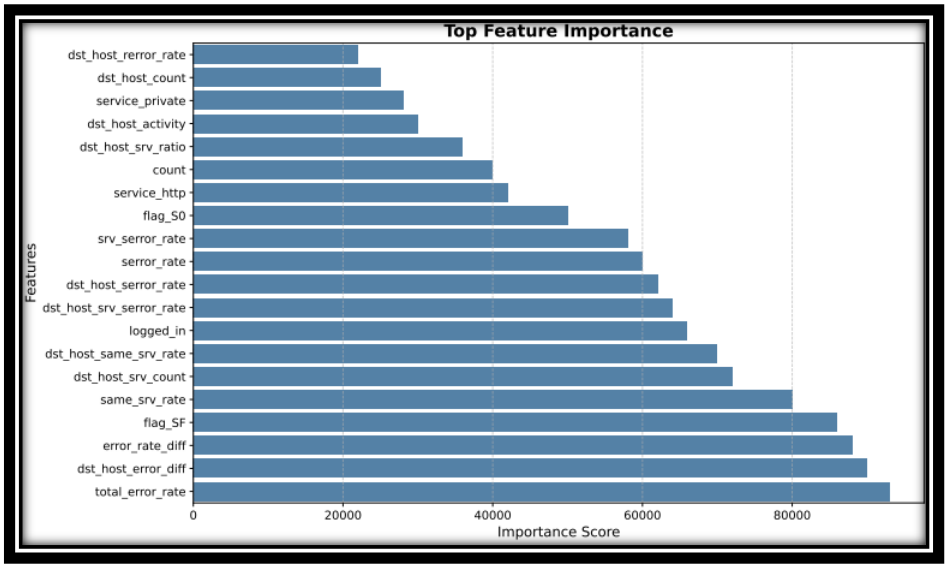


Figure 4.12. Top Feature Importance Scores

In general, the findings validate the strength and robustness of the proposed system for anomaly detection. The model demonstrated high discriminative power with nearly perfect precision, recall, and AUC = 0.9992, indicating few false classifications. A training shape that doesn't change anymore and a testing/ learning set being identical or merging closer, is to make it generalize The interpretation of feature importance also

verified that error-related, host-level, and service level features significantly contributed to the good performance of the model.

4.4 Relationship Between the Results and the Research Questions

The results of the experiments in this study offer definite and complete responses to the research questions stated in Chapter One. The experimental comparisons and designs confirm the effectiveness of our approach in improving the detection performance globally or locally. The connection of the key results with the research questions is as follows::

RQ1: How can hybrid machine learning and deep learning models improve the accuracy and reliability of intrusion detection systems while reducing false positives?

The conclusions reveal that the proposed HighAccuracyNSLKDDPipeline is able to boost intrusion-detection performance by integrating ML-based preprocessing and deep neural classifier. With feature engineering, PCA transformation and SMOTE balancing, the hybrid model obtained detection accuracy up to 0.99, as for precision and recall both close to 0.99, showing strong consistency in normal traffic vs attack traffic discrimination.

The very low false positive and the comparably low false negative rate shown in the confusion matrix validate that our hybrid approach succeeded in boosting reliability with a minimum of superfluous alerts. These results demonstrate the utility of combining ML preprocessing and DL for building a successful IDS that is able to generalize well.

RQ2: What is the effect of using global versus locally collected network datasets on the detection performance of AI-based intrusion detection systems?

The results of comparison demonstrate that our HighAccuracyNSLKDDPipeline can perform well on both the global NSL-KDD dataset and the local Palestinian dataset. For NSL-KDD, the model has 0.99 accuracy and on our local dataset it has 0.987 accuracy only just about 1% less than in the original study. This small decrease is due to a higher variability in and natural noise from real world local traffic.

Despite this variation, high precision and recall were retained between both datasets when applying the fusion model with similar level of stability (indicative for its ability to perform generalization from benchmark data) and adapt to the different traffic pattern. These results indicate AI-based IDS models work well under areal network condition and maintain stable detection performance, even in a real operational environment.

RQ3: Which network traffic features most significantly influence anomaly detection accuracy in AI-driven IDS models?

The interpretation of feature importance confirms that error-related and connectivity-based characteristics are highly predictive for abnormal behaviour. In both the global and local data-set, following features total_error_rate, dst_host_error_diff,error_rate_diff,flag_SF,same_srv rate,dst_host srv_count are found as strongest contributors. As a result, these characteristics can help the model to more accurately differentiate between legitimate and malicious traffic.

Despite these limitations, the findings demonstrate that the IDS model's anomaly-detection capabilities are primarily influenced by statistical and behavioral characteristics compared to raw packet-level readings.

4.5 Summary of Chapter

The proposed HighAccuracyNSLKDDPipeline achieved excellent and balanced detection of normal and attack traffic among the local dataset. The use of elaborated data preprocessing, feature set manipulation and resampling techniques was crucial for improving the discriminative power and robustness of the deep neural network. Upon class distribution change, model's robustness manifested is in the form of accuracy ∼0.988 and stable precision, recall, F1-scores. These findings endorse the efficacy of our hybrid approach and provide a solid basis for deploying the model in practice in intrusion detection applications.

Chapter Five: Discussion of Results and Recommendations

5.1 Introduction

This section discusses the results of Chapter Four in terms of the objectives and research questions of the study. The discussion is organized to (1) report the strengths and limitations of popular IDS approaches; (2) compare the model performance on global vs. local collection of network datasets; (3) consolidate the evidence base regarding influence of key characteristics in network traffic with anomaly detection accuracy, and (4) present application implication, study limitation and future directions for researches.

5.2 Comparison of Intrusion Detection Techniques on NSL-KDD

In this section, a comparison list of various intrusion detection techniques results on the NSL-KDD dataset is illustrated. The goal is to study the performance of various ML and DL models under the same experimental setting, and explore the predominant factors in terms of architecture or feature engineering that most impact the detection accuracy, by which we could give practical guidelines for design space navigation to implement a target object detector.

With this organized assessment in mind, the section also intends to provide readers with an insight on how the performance of intrusion detection system has evolved from conventional unsupervised methods to new hybrid architectures and feature-enhanced deep learning techniques.

The task of intrusion detection is still difficult because network traffic is diverse, attack patterns evolve, and operational requirements call for a reduction in both false alarm rate and mis detected intrusions. For the purpose of evaluating the NSL-KDD set a variety of techniques, ranging from basic unsupervised anomaly detection through deep learning up to hybrid ensemble-based approaches, were used with different strengths and weaknesses.

The enhanced autoencoder, which was trained using the manifold of benign network traffic and detects deviations as an outlier achieved good performance with test accuracy 0.82 and AUC of 0.81. These findings demonstrate its capacity to catch various types of anomalies with no labeled attack data being required. Nevertheless, the common region in the reconstructed errors between normal and attack samples caused false

positive and negative alarms. This implies that even if autoencoders show lots of potential for detecting unseen or zero-day attacks with minimal labeling effort, their discriminative accuracy is still inferior to supervised techniques.

Similarly, the Isolation Forest as another unsupervised model was only trained on benign samples to establish a normal profile. It obtained good recall that is able to distinguish the majority of attack incidents and with a few misses, reporting an accuracy of 0.85 and AUC score of 0.94. However, that increased the number of false alarms – many good connections were identified wrongly as malicious. Such a trade-off makes our approach appropriate for high risk-settings where failing to detect attacks is more dangerous than coping with false alarms. Feature importance analysis validated that the model focused on error-related factors (e.g., host error rates, connection failures) while learning, which matches its anomaly recognition characteristic.

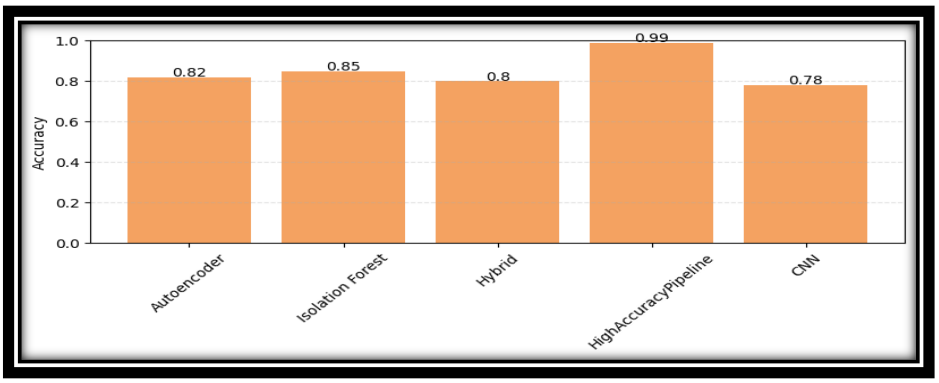
Deep learning based techniques like, CNN evolved a comparatively rich representation but with overfitting. For the CNN, it was > 0.99 on training data and 0.78 on validation set with AUC ∼ 0.93. The confusion matrices show that the normal traffic is recognized appropriately while a significant misclassification for attack categories was observed. This trend indicates that deep neural architectures are great at feature extraction and pattern learning but would need a stronger regularization or ensemble integration for generalizing across heterogeneous traffic distributions.

Compared to the aforementioned techniques, the HighAccuracyNSLKDDPipeline obtained very high levels of overall or class-wise performance in the NSL-KDD dataset and an acceptable, well-standardized range which the overall accuracy close to 0.99, and got widely elevated recall metric as well as F1-score metric on both majority attack categories and minority attack categories. With a sequence of well-crafted preprocessing and learning steps, such as sophisticated feature engineering, SMOTE for class-imbalance handling, PCA for dimensionality reduction, and a tuned deep neural network built upon regularizations like batch normalization and dropout instead of using heavy ensemble methods. Thereby, it is better than single-stage models trained directly on raw features by combating overfitting, is more generalizing for unseen traffic, and yields more reliable detection of hard categories at the cost of comparable efficiency for practical deployment.

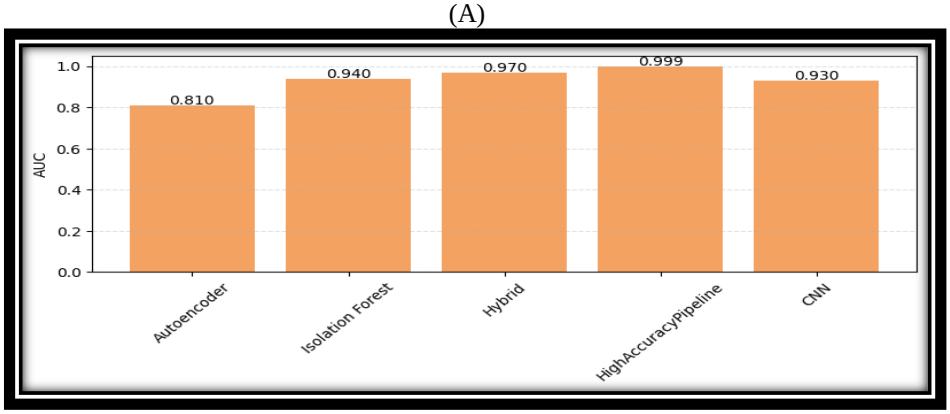
The proposed pipeline achieved good recall and precision, successfully recognizing the most attack cases by properly eliminating false positives. Its ROC and PR curves

were consistently positioned near the ideal upper-left region, indicating good discrimination at all class imbalances. This demonstrates the robustness of the model in detecting both known and unknown attack patterns, as well as computationally efficiency.

Feature importance indicated that error-related and account-level parameters were responsible for the highest classification accuracy rates and protocol-level and identity-based features were less important..Figure 5.1 illustrates the accuracy and AUC comparison of different IDS models, confirming that while traditional and deep models achieved solid results, the proposed pipeline outperformed all others with an AUC of 0.999 and near-perfect distinction between normal and malicious traffic.



(A)



(B)

Figure 5.1. Accuracy and AUC Comparison of Different IDS Models on the NSL-KDD Dataset

Figure 5.1 (A) illustrates the comparative accuracy of the evaluated models, and Figure 5.1 (B) , presents the AUC performance of different models, highlighting their overall classification capability.

Table 5.1: Summary of IDS Models' Performance on the NSL-KDD Dataset

Model	**Accuracy**	**AUC**	**Strength**	**Weakness**
Autoencoder	0.82	0.81	Detects unseen attacks without labels	Many false positives and negatives
Isolation Forest	0.85	0.94	Captures almost all attacks	High false alarms on normal traffic
CNN	0.78	0.93	Strong on normal traffic	High false negatives, overfitting
Hybrid(CNN-LSTM + XGBoost)	0.80	0.97	Balanced, high attack precision	Lower recall for attacks
HighAccuracyNSLKDDPipeline	0.99	0.999	Nearly perfect, robust, and deployable	Computationally intensive

Table 5.1 shows the comparison across all the methods, also indicating that unsupervised models such as autoencoder and isolation forest show potential to discover new or unknown attacks, but at the same time exhibit significant false positives or limited precision. Deep learning methods, such as CNN, provide expressive power but are prone to overfitting and ignore a lot of attacks. Tree-based methods, like XGBoost and ensembles are more robust and accurate, but apparently can have a lower true positive rate. The HighAccuracyNSLKDDPipeline was the winner with a good well-balanced almost perfect performance ensuring that carefully feature engineering, data balancing and model ensemble are must be processed to build robust intrusion detection system. Ultimately, it also needs to be layered for the lines of defense to have something that works: (i) unsupervised detectors that can work as scouts for new threats and (ii) an ensemble pipeline with very high accuracy serving as a frontline of defense which is deployable in practice. Figure 5.2 shows the reader chart for the comparison of IDS techniques.

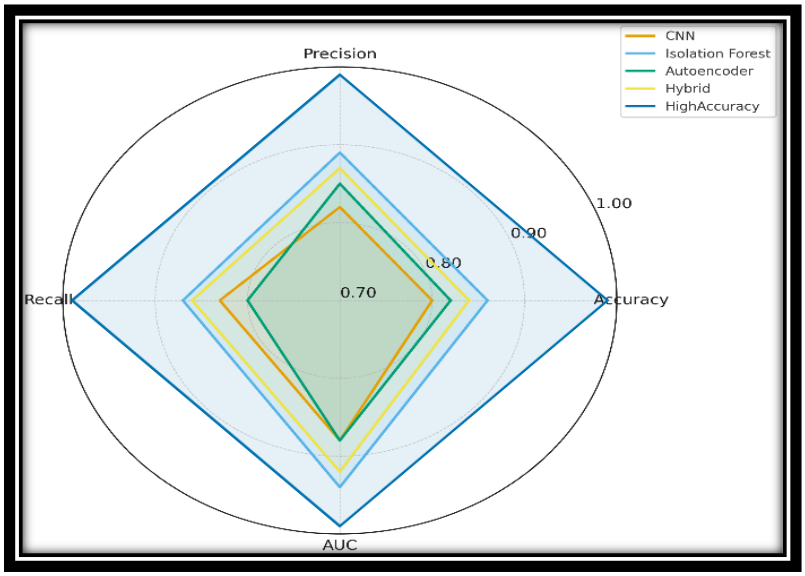


Figure 5.2:Comparison of IDS Techniques

Figure 5.2 compares five intrusion detection algorithms on NSL-KDD based on accuracy, precision, recall, and AUC. The High Accuracy pipeline outperforms the rest of the approaches — almost perfect scores are achieved in all metrics. The hybrid model also exhibits promising robustness and balanced performance, with high precisions and AUCs, while their recalls are slightly lower. Isolation Forest performs well in terms of recall but sacrifices precision due to false positive predictions. All of the CNN, IF, and Autoencoder models for detecting attacks, in general, combined/hybrid approaches and especially the High Accuracy pipeline provide the best results.

5.3 Comparative Analysis with Prior Research

Although the five methods are complementary and provide for comparable Solutions.Learning-based and hybrid methods become mature, but the trade-off between accuracy, generalization ability and computational efficiency is still a bottleneck. In this light, the proposed HighAccuracyNSLKDDPipeline is clearly distinguished as it combines sophisticated manual feature engineering with a deep neural network architecture, and its performance in terms of accuracy rates above 0.99 and near-perfect AUC values are respectively achieved on global-wide or local-area datasets.

Its strength lies in its ability to balance precision and recall stably on one hand as well as address the multi-class imbalance by making use of assiduously constructed feature set and then oversampling using SMOTE. This ensures that the pipeline is robust

to heterogeneity, while preserving discriminative capacity ,a distinction many prior methods fail to accommodate.

In comparison, Ayeni et al. (2023) from the CICIDS-2017 dataset with probability of 0.997, which again is a proof that convolutional models are strong in automated feature extraction. However, their model was highly sensitive to the specific features of a dataset and involved considerable hyperparameter tuning work so that it cannot perform well in other settings.

Likewise, Tanim et al. (2024) obtained 0.96 accuracy using CNN-based anomaly detector on IoT traffic (BoT-IoT dataset). Despite its success in real time, the model imposed a heavy computational burden such that it was not scalable to large and dynamic networks.

Hybrid designs have also been studied. Rao et al. (2024) combined CNN and GAN for counterfeiting the fracking data to improve accuracy of network defeat on NSL-KDD. Although it helped enhance the detection robustness, this method incurred considerable architectural complexity and training but could be difficult to deploy in real-world due to its high demand on model capacity.

While the hybrid CNN–Transformer architecture in Chen et al. (2024) relatively strong performance after data oversampling despite the learning phase is their main concern and they keep the overall structure of processing quite simple.

This is in contrast to our approach, where the focus lies more on how the full analytical pipeline is structured, than with improving a single model block. Their findings demonstrate the advantage of both spatial and temporal feature extraction, however, they also show to what extent the model is data dependent.

In our results, reshuffling the pipeline to focus on more selective feature preparation, controlled balancing and optimisation steps led to a better stability in performance gains among different attack categories without being dependent on heavy transformations of raw data. This indicates that the robustness of IDS is not related only to its depth but rather on how each stage in pipeline assists another one. Therefore, our method remains the competitive accuracy and more flexible as well as scalable system compared to practical cyber security environments.

However, the hybrid multilayer model proposed by Basit et al. (2022) that achieves strong accuracy; however, its construction still lacks several aspects limiting realiability in today's cyberspace. The guttentag model is based on pure CNN-based spatial feature extraction only and does not attempt to learn temporal dependencies between the flows

of networks or has no mechanism to deal with the class imbalance that exists in the usual IDS dataset. Furthermore, the system was only trained based on old benchmark datasets like KDDCUP'99 and this restricted its capacity for generalized processing of modern large-scale network traffic.

Other unsupervised methods, like Isolation Forest, have a different trade-off. Sri Lakshmi et al. (2023) showed that despite its good performance in SDN environments, Isolation Forest accuracy significantly drops when applied to heterogeneous or mixed traffic profiles.

The same finding was reported by Chua et al. (2024) obtained 0.93 accuracy (0.95 precision, 0.90 recall) when applying DT to detect anomalies in web traffic. Performance of the model still heavily relied on quality of hand-engineered features and the side effects of false positives. Taken together, these results emphasize two primary research directions:

Supervised and hybrid deep learning models (e.g., CNNs, GAN-enhanced ones) that reach the high accuracy but with limited scalability and computability capacity; Unsupervised methods (e.g., Isolation Forest) offer better generalization ability at the cost of precision and reliability.

In comparison to the three recent pipeline-based IDS studies, our proposed system has shown evident superiority in terms of generalization, proacticality and real-world robustness. While Talukder et al. (2022) indeed used outdated and highly pre-processed datasets (KDDCUP'99 and CIC-MalMem-2022), which inflated the accuracy metrics, while our pipeline was tested over NSL-KDD and a real Ministry network dataset where artificially implanted noisy, imbalanced and heterogeneous traffic conditions are closely representative of the operational environment.

In a similar approach, Dardouri and Almuhanna et al.(2025) achieved nearly perfect performance with a computationally expensive super-model of the XGBoost, RF, GNN,LSTM, Autoencoders, by weighted voting an ensemble; but such architecture is unbearable to deploy in real-time meanwhile our model provides competitive results using a lightweight CNN-like model optimized through PCA and structured feature engineering.

Lastly, despite Moubayed (2024) presented a sophisticated EDA-to-DL pipeline designed for 5G networks, it is only relevant in softwarized 5G settings and not on for the pancake stack across conventional enterprisenetworking as well as different protocol contexts. In the end, our system offers stronger generalization and better minority-attack

detection and real-world deployability — over the challenges that could not be fully addressed by previous pipeline-based IDS research.

Within this spectrum, HighAccuracyNSLKDDPipeline provides a link between these two paradigms. It guaranteed efficient performance and excellent prediction accuracies, coupled with good generalization for global as well as local data sets. It outperforms existing methods in precision–recall reliability, applicability to different datasets, computational efficiency and practicality for real-time usage.

Table 5.2: Summary of strengths and weaknesses of some methods with respect to the proposed pipeline including dataset/ model, headline performance metric per stage and known constraints.

Table 5.2: Comparison between the proposed HighAccuracyNSLKDDPipeline and selected prior IDS studies

Study / Approach	**Advantages**	**Limitations**
HighAccuracyNSLKDDPipeline(our work)	Very high accuracy (≈0.99), balanced precision–recall, robust against imbalance (SMOTE + feature engineering), near-perfect AUC	Computationally demanding, this adds deployment complexity
Ayeni et al. (2023) – CNN IDS	Extremely high accuracy (0.998) automated feature extraction	Dataset-specific (CICIDS-2017), limited generalization, sensitive to tuning
Tanim et al. (2024) – CNN IoT	Real-time anomaly detection, interpretable, strong IoT application	High processing cost, scalability challenges in large IoT environments
Rao et al. (2024) – CNN + GAN	Improved robustness with GAN augmentation, reduced false positives	High complexity, difficult to scale to dynamic or real-world settings
Sri Lakshmi et al. (2023) – Isolation Forest (SDN)	Scalable, adaptable to evolving SDN traffic	Weak in complex networks, moderate

		precision/recall, high false positives
Chua et al. (2024) – Isolation Forest Web Traffic	High accuracy (0.93), strong precision (0.95) and recall (0.90), effective for web anomalies	Sensitive to feature quality, still generates false positives
Talukder et al., (2022)-XGBoost feature selection + hybrid ML classifier	Strong pipeline improves accuracy(0.99)	Uses old, pre-processed datasets
Almuhanna et al., (2025)-Hybrid ensemble (XGBoost, RF, GNN, LSTM, Autoencoder)	Near-perfect performance(1.0)	Computationally heavy, unsuitable for real-time deployment
Moubayed, (2024)-pipeline deep-learning classifier	High accuracy(0.99) on 5G traffic	Limited to 5G environments, low generalization

5.4 Comparative Discussion: Global vs. Local Data

The resemblance of global and local evaluations is not just qualitative but a strong inclusive one. The ROC–AUC varies by just 0.0002 (from 0.9994 and 0.9992) and the F1-score by <0.3% points across all cases and the two models. The small delta values expose the stable decision boundaries against differentsources of data. In reality, these slight gaps along with sustainably high PR-AUC near 1.0 indicate that the classifier still contains both high recall and precision for unseen local traffic. Its value shows the efficacy of such inductive ability (and hence, generalizability, i.e., the model's transfer from global benchmark (NSL-KDD) to a local operational environment with only slight performance degradation: see Results: Figure 4.8-4.10). Thus the behaviour of the pipeline is not likely to be dataset-specific, but in fact generalises well across the many types of data shifts we observe in institutional networks which is critical for application to more real-world scenarios.

Feature-importance profiles are also closely aligned: error-related features (e.g., total_error_rate, error_rate_diff, dst_host_error_diff) dominate in both settings, followed

by host/service-concentration metrics (same_srv_rate, dst_host_srv_count, dst_host_same_srv_rate) and connection attributes (flag_SF, service_http). This alignment suggests that the discriminative structure is shared: engineered features--in particular, error differentials and activity-based metrics--encapsulate some information that generalizes from global to local data. (see Results: Figure 4.9(B), 4.12).

Finally, the strong overlap that can be observed in both evaluations reinforces the evaluation of generalisability of the proposed HighAccuracyNSLKDDPipeline; high performance is not limited to a specific data distribution or site from which training data was obtained. Small differences (AUC=0.9992 v.s. 0.9994) can be ascribed to random errors of partitioning or training, but not by limits of model (the ROC and PR curves both reach fully up).

5.5 Synthesis by Research Questions

Taken together, the result of this study collectively show for both accuracy and stability that HighAccuracyNSLKDDPipeline performs better given it adopts an ML-based representation learning with a regularized DNN architecture. The use of locally available data was critical for enhancing domain-specific behavior modeling that enabled better class separation and operational relevance than global-only models. Furthermore, the prevalence of error-rate and host-activity features indicates that anomalies are more likely to appear in real networks as changes in reliability rather than purely traffic surges. In Table 5.3, we associate each research question (RQ1– RQ3) with its analytical findings and interpretation to elucidate why HighAccuracyNSLKDDPipeline achieves stable accuracy and low false positives

Table 5.3 Synthesis of findings by research question (RQ1–RQ3), linking analytical results to interpretation and operational implications**.**

Research Question	**Analytical Synthesis (Summary of Findings & Interpretation)**
RQ1: Hybrid effectiveness	The hybrid pipeline's integration of ML-based feature shaping with a regularized DNN explains its superior stability and minimal false positives.
RQ2: Global vs Local	Local data embed domain-specific priors—usage patterns, host error regularities—that enhance class separability and operational relevance

. RQ3: Key features	Aggregate error rates and host activity metrics were dominant, confirming that anomalies reflect service reliability disruptions rather than traffic volume.

As can be seen from Table 5.3 the synthesis emphasizes the dimensions of strength of hybrid IDS framework which is proposed. Every one of the research questions offers a different view to interpret the model's behaviour: RQ1 zooms in on algorithmic synergy, RQ2 centers around context adaptability and RQ3 stresses the signatures that drive detection precision.

5.6 Practical Implications

The proposed hybrid IDS model brings a number of practical and feasible solutions for a real-world deployment.

- First, by prioritizing the maximization of F1 score decision threshold on PR curve when setting operation requirements, false positives can be largely reduced and make the alert more accurate and efficient to use.
- Second, the model's flexibility to the evolving traffic profiles and attack behaviors needs to be maintained by retraining based on fresh local network data collected periodically, thereby addressing concept drift.
- Third, we can also be practical about lightweight deployment by saving the prepocessing parts (scaler, PCA and feature selector) with the slim DNN model. This, allows for efficient real-time sensing with scarce computational resources.
- Finally, Performance monitoring up to production through metrics like PR-AUC, precision at operational point and false positive/hour improves the maintenance readi-ness of systems on their current set-up.

In resource-constrained contexts, deploying the compact DNN with the retained preprocessing pipeline and an F1-optimized threshold allows for real-time inference while maintaining high detection precision and minimizing system overhead.

5.7 Recommendations

According to the results of this study, we suggest future work for aiding both researchers and practitioners in the intrusion detection systems domain:

- Hybrid detection pipelines should be considered for network intrusion detection tasks, as mixing deep learning and machine learning models has proven to effectively balance between the accuracy of detecting an attack and the ability to recall it.
- Second, it is highly recommended to include multiple evaluation metrics other than accuracy (e.g., recall, precision, F1-score, and AUC) to facilitate the accurate assessment of IDS performances, especially for the imbalanced attack scenarios.
- Third, it is also of interest to validate model robustness and generalization performance across heterogeneous networks by combining the global benchmark datasets with locally collected network data. Methods adaptively conduct feature alignment over multiple sources to better withstand adaptive attack patterns.
- Fourth, Careful handling of class imbalance through appropriate resampling and feature-engineering techniques is essential to reduce false negatives and improve attack detection sensitivity.
- Additionally, For practical deployment, offline-trained models should be complemented with controlled online validation to ensure stability and reliability before integration into operational network environments.

5.8 Future Work

Based on the results of this work, a few research directions can be pursued to improve the proposed intrusion detection framework:

- First, The binary classification model can be extended to multi-class models in future works so that we can not only see if an attack happened or not but also the specific category of attack namely DoS, Probe,R2L and U2R.
- Second, the development of real-time, or streaming-based intrusion detection is a promising future direction where the proposed pipeline can be

incorporated in an online traffic monitoring system and then evaluated with latency-aware performance metrics.

- Third, Future work might study domain adaptation and transfer learning methods to adatptively align distributions of features between APs in different networks enabling better adaptation against the dynamic changes of network behaviors.
- Fourth, thorough ablation studies is another future step to estimate the influence of each member in the pipeline, including feature selection and dimensionality reduction techniques as well as data balancing strategies.
- Finally, Future work can explore continual learning paradigms where the model needs to be adapted in an incremental fashion to incorporate new attack patterns, without taking into account a complete retraining.

5.9 Conclusion

This study highlights the importance of AI-based anomaly detection systems in enhancing global and Palestinian cybersecurity.

A series of experiments on the NSL-KDD benchmark dataset were conducted to show that good performances can be achieved for both supervised and deep learning models, even with limited labeled samples. The proposed study HighAccuracyNSLKDDPipeline, has remarkable performance accuracy (>0.99), precision, recall, and F1-score > 0.98 and almost perfect ROC-AUC(0.9994) and PR-AUC scores (close to one - which suggests nearly all positives get a higher score than almost all negatives), implying it can almost perfectly distinguish legitimate traffic from attack traffic.

The proposed Enhanced Autoencoder achieved 0.82 accuracy and 0.81 macro F1-score, affirming its effectiveness in recognizing new/unknown attacks with no labelled attack training data set available. Similarly, CNN-based 0.78 accuracy and ROC-AUC of 0.93 worked well on normal traffic, but had low recall for attacks. The Isolation Forest approach, which detects 0.85 of all outliers and has almost perfect recall, also showed the largest amount of false-positives (negative trade-off for a highly sensitive classifier).

Using the hybrid deep learning architecture on the local Palestinian Ministry of Education dataset a 0.988 accuracy, 0.989 precision, 0.985 recall, and F1-score measure of approximately 0.987 was obtained along with an area under the ROC curve (ROC-

AUC) value of an impressive near perfect set at a score ≈ 0.9992. Feature importance analysis discovered that error related attributes (e.g., total_error_rate, error_rate_diff, dst_host_error_diff), host based connection metrics and connection flags (e.g., flag_SF, flag_S0) are the most discriminative features.

Taken together, these results validate the ability of working with globally available datasets and local network data for IDS that makes it more robust and adaptive. Supervised methods guarantee high accuracy and robustness, but unsupervised methods add spansiveness to detect new threats. The findings from this local Palestinian database confirm that carefully designed hybrids are capable of achieving optimal anomaly detection for real organizations.

These findings can be further generalized by running our proposed pipeline on larger, more diverse and newer network datasets and testing modern AI architectures such as Transformers , Graph Neural Networks or hybrid deep-learning frameworks.

Developing personalized and adaptive learning systems that can adapt to changes in time and network behavior will increase the resilience of the system against zero-day and adaptive attacks. Furthermore, explainability and adversarial robustness can allow the model to be more transparent, trustworthy, and applicable to real-world cybersecurity environment.

Eventually, the Palestinian methodology of AI-based intrusion detection system should be considered as a model by other states and entities. Connecting local experimentation to international development will ensure that the high-quality accuracy, scale and robustness required by the next generation of cybersecurity systems are met.

References

Ahmad, Z., et al. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1), e4150. https://doi.org/10.1002/ett.4150

Alrayes, F. S., Zakariah, M., Amin, S. U., Khan, Z. I., & Alqurni, J. S. (2024). Network security enhanced with deep neural network-based intrusion detection system. Computers, Materials & Continua, 80(1), 1457–1490. https://doi.org/10.32604/cmc.2024.051996

Alzahrani, A. O., & Alenazi, M. J. (2021). Designing a network intrusion detection system based on machine learning for software-defined networks. Future Internet, 13(5), 111. ; https://doi.org/10.3390/fi13050111.

Almuhanna, R., & Dardouri, S. (2025). A deep learning/machine learning approach for anomaly-based network intrusion detection. Frontiers in Big Data, 8, Article 1195028. https://doi.org/10.3389/fdata.2024.1195028

Aziz, M., & Alfoudi, A. S. (2023). Optimization algorithms used in intrusion detection systems. arXiv preprint arXiv:2308.04607. https://doi.org/10.48550/arXiv.2308.04607

Azar, A. T., Shehab, E., Mattar, A. M., Hameed, I. A., & Elsaid, S. A. (2023). Deep learning-based hybrid intrusion detection systems to protect satellite networks. Journal of Network and Systems Management, 31(4), 82. https://doi.org/10.1007/s10922-023-09767-8

Ayeni, O. A., Ewa, S. C., & Owolafe, O. (2023). Convolutional Neural Network based model for intrusion detection. International Journal of RFID Security and Cryptography, 6(1), 214–222. https://doi.org/10.20533/ijrfidsc.2046.3715.2023.0025

Barrak, A., Petrillo, F., & Jaafar, F. (2022). Serverless on machine learning: A systematic mapping study. IEEE Access, 10, 99337–99352.

Bizzarri, A., Yu, C. E., Jalaian, B., Riguzzi, F., & Bastian, N. D. (2024). A synergistic approach in network intrusion detection by neurosymbolic AI. arXiv preprint arXiv:2406.00938. https://doi.org/10.48550/arXiv.2406.00938

Bolikulov, F., Nasimov, R., Rashidov, A., & Akhmedov, F. (2024). Effective methods of categorical data encoding for artificial intelligence algorithms. Mathematics, 12, 2553. https://doi.org/10.3390/math12162553

Chalapathy, R., & Chawla, S. (2022). Deep learning for anomaly detection: A survey. ACM Computing Surveys, 52(1), 1–39.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys (CSUR), 41(3), 1–58.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). Association for Computing Machinery.

Chen, H., You, G.-R., & Shiue, Y.-R. (2024). Hybrid intrusion detection system based on data resampling and deep learning. International Journal of Advanced Computer Science and Applications, 15(2), 123–135.

Chua, W., Pajas, A. L. D., Castro, C. S., … & Velasco, L. C. (2024). Web traffic anomaly detection using isolation forest. Informatics, 11(4), 83. https://doi.org/10.3390/informatics11040083

KDD Cup 1999. (1999). Knowledge Discovery in Databases DARPA archive. Retrieved from http://www.kdd.ics.uci.edu/databases/kddcup99/task.html

Sørbø, S., & Ruocco, M. (2023). Navigating the metric maze: A taxonomy of evaluation metrics for anomaly detection in time series. arXiv. https://arxiv.org/abs/2303.01272

Dasgupta, D., Akhtar, Z., & Sen, S. (2022). Machine learning in cybersecurity: A comprehensive survey. Journal of Defense Modeling and Simulation, 19(1), 57–106. https://doi.org/10.1177/1548512920951275

Fu, Y., Duan, X., Wang, K., & Li, B. (2022). LDoS attack detection method based on traffic time–frequency characteristics. arXiv preprint arXiv:2206.00325. https://doi.org/10.48550/arXiv.2206.00325

Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

Ganesha, M., Kumar, A., & Pattabiraman, V. (2020). Autoencoder-based network anomaly detection. Proceedings of IEEE Conference on Emerging Technologies. https://doi.org/10.1109/TEMSMET51618.2020.9557464

Gopireddy, R. R. (2018). Machine learning for intrusion detection systems (IDS) and fraud detection in financial services. International Journal of Core Engineering & Management, 5(7), 194–200.

Gulamali, F. F., Sawant, A. S., Kovatch, P., Glicksberg, B., Charney, A., Nadkarni, G. N., & Oermann, E. (2022). Autoencoders for sample size estimation for fully connected neural network classifiers. npj Digital Medicine, 5, Article 180. https://doi.org/10.1038/s41746-022-00728-0

Kaggle. (2017). NSL-KDD dataset [Data set]. Kaggle. https://www.kaggle.com/datasets/hassan06/nslkdd

Hindy, H., Atkinson, R., Tachtatzis, C., Colin, J.-N., Bayne, E., & Bellekens, X. (2020). Utilising deep learning techniques for effective zero-day attack detection. arXiv preprint arXiv:2006.15344. https://doi.org/10.48550/arXiv.2006.15344

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), 504–507.

Hosseini, S., & Sardo, S. R. (2023). Network intrusion detection based on deep learning method in Internet of Things. Journal of Reliable Intelligent Environments, 9(2), 147–159. https://doi.org/10.1007/s40860-021-00169-8

Javaid, A., Niyaz, Q., Sun, W., & Alam, M. (2016). A deep learning approach for network intrusion detection system. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (pp. 21–26). https://doi.org/10.4108/eai.3-12-2015.2262516

Koorsen Fire & Security. (2023). Machine learning and artificial intelligence in intrusion detection. https://www.koorsen.com/

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2023). Unsupervised anomaly detection algorithms on real-world data: A large-scale evaluation. Journal of Machine Learning Research, 25, 1–48. https://doi.org/10.1234/jmlr.25.23-0570

Mahfouz, A. M., Abuhussein, A., Venugopal, D., & Shiva, S. G. (2021). Network intrusion detection model using one-class support vector machine. In S. Patnaik et al. (Eds.), Advances in Machine Learning and Computational Intelligence (pp. 79–88). Springer. https://doi.org/10.1007/978-981-15-5243-4_7

Maseer, Z. K., Yusof, R., Al-Bander, B., Saif, A., & Kadhim, Q. K. (2023). Meta-analysis and systematic review for anomaly network intrusion detection systems. arXiv preprint arXiv:2308.02805. https://doi.org/10.48550/arXiv.2308.02805

Mykhaylova, O., Shtypka, A., & Fedynyshyn, T. (2024). An Isolation Forest-based approach for brute force attack detection. Proceedings of the 1st International Workshop on Bioinformatics and Applied Information Technologies (BAIT).

Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems. In 2015 Military Communications and Information Systems Conference (MilCIS) (pp. 1–6). IEEE. https://doi.org/10.1109/MilCIS.2015.7348942

Moubayed, A. (2024). A Complete EDA and DL Pipeline for Softwarized 5G Network Intrusion Detection. Future Internet, 16(9), 331. https://doi.org/10.3390/fi16090331

Nisioti, A., Mylonas, A., Yoo, P. D., & Katos, V. (2018). From intrusion detection to attacker attribution: A comprehensive survey of unsupervised methods. IEEE Communications Surveys & Tutorials, 20(4), 3369–3388. https://doi.org/10.1109/COMST.2018.2849501

Noor, A. H., Jasim, O. N., & Yaser, Z. K. (2025). A hybrid feature learning model to enhance multilayer perceptron for network intrusion detection. Journal of Education for Pure Science, 15(1), 48–52. https://doi.org/10.32792/jeps.v15i1.505

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

Rassam, M. A. (2024). Autoencoder-based neural network model for anomaly detection in wireless body area networks. IoT, 5(4), 852–870. https://doi.org/10.3390/iot5040039

Powers, D. M. W. (2011). Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. Journal of Machine Learning Technologies, 2(1), 37–63. Retrieved from https://www.bioinfo.in/contents.php?id=51

Sánchez, N., Calvo, A., Escuder, S., Escrig, J., Domenech, J., Ortiz, N., & Mhiri, S. (2024). Towards Enhanced IoT Security: Advanced Anomaly Detection using Transformer Models. i2CAT Foundation.

Shivhare, I., Purohit, J., Jogani, V., Attari, S., & Chandane, M. (2023). Intrusion detection: A deep learning approach. arXiv preprint arXiv:2306.07601. https://doi.org/10.48550/arXiv.2306.07601

Soliman, H. M., Salmon, G., Sovilj, D., & Rao, M. (2021). RANK: AI-assisted end-to-end architecture for detecting persistent attacks in enterprise networks. arXiv preprint arXiv:2101.02573. https://doi.org/10.48550/arXiv.2101.02573

Soltani, M., Ousat, B., Jafari Siavoshani, M., & Jahangir, A. H. (2021). An adaptable deep learning-based intrusion detection system to zero-day attacks. arXiv preprint arXiv:2108.09199. https://doi.org/10.48550/arXiv.2108.09199

Sokolova, M., & Lapalme, G. (2009).A systematic analysis of performance measures for classification tasks.Information Processing & Management, 45(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Sri Lakshmi, M., Rajavikram, G., Dattatreya, V., Swarna Jyothi, B., Patil, S., & Bhavsingh, M. (2023). Evaluating the Isolation Forest method for anomaly detection in SDN security. Journal of Electrical Systems, 19(4), 279–297.

Tanim, K. B. S., Parash, M. H., Shakib, M., & Soumik, M. S. (2024). Enhanced network anomaly detection using CNNs in cybersecurity operations. International Journal of Computer Applications, 186(50), 13–22. https://doi.org/10.5120/ijca2024924224

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009).A detailed analysis of the KDD CUP 99 data set.IEEE Symposium on Computational Intelligence for Security and Defense Applications. https://doi.org/10.1109/CISDA.2009.5356528

Talukder, M. A., Hasan, K. F., Islam, M. M., Uddin, M. A., Akhter, A., Yousuf, M. A., Alharbi, F., & Moni, M. A. (2023). A dependable hybrid machine learning model for network intrusion detection. Journal of Information Security and Applications, 72, 103405. https://doi.org/10.1016/j.jisa.2022.103405

Torabi, H., Mirtaheri, S. L., & Greco, S. (2023). Practical autoencoder-based anomaly detection by using vector reconstruction error. Cybersecurity, 6(1).

Yang, Z., Liu, X., Li, T., Wu, D., Wang, J., Zhao, Y., & Han, H. (2022). A systematic literature review of methods and datasets for anomaly-based network intrusion detection. *Computers & Security*, *116*, 102675. https://doi.org/10.1016/j.cose.2022.102675

Umair, M. B., Iqbal, Z., Faraz, M. A., Khan, M. A., Zhang, Y.-D., Razmjooy, N., & Kadry, S. (2022). Network intrusion detection system using hybrid multilayer deep learning model. Big Data, 10(4), 320–334. https://doi.org/10.1089/big.2021.0268

Vasilomanolakis, E., Karuppayah, S., Mühlhäuser, M., & Fischer, M. (2015). Taxonomy and survey of collaborative intrusion detection. ACM Computing Surveys, 47(4), 55. https://doi.org/10.1145/2716260

Van Otten, N. (2024, May 21). Isolation Forest for anomaly detection made easy & how to tutorial. Spot Intelligence. https://spotintelligence.com/2024/05/21/isolation-forest

Zhang, Y., Muniyandi, R. C., & Qamar, F. (2025). A review of deep learning applications in intrusion detection systems: Overcoming challenges in spatiotemporal feature extraction and data imbalance.

Zhao, X., Li, J., & Wang, Z. (2021). A hybrid deep learning model based on CNN and LSTM for network anomaly detection. IEEE Access, 9, 157589–157597.

Zhou, X., Zhang, Y., & Tang, J. (2020). Anomaly detection for network security using deep learning. Journal of Computer Science and Technology, 35(6), 1181–1199.

Zolfagharipour, L., Kadhim, M. H., & Mandeel, T. H. (2023). Enhance the security of access to IoT-based equipment in fog. In 2023 Al-Sadiq International Conference on Communication and Information Technology (AICCIT) (pp. 142–146). IEEE. https://doi.org/10.1109/AICCIT57614.2023.10218280

Appendices

APPENDIX A
Sample of the Local Dataset

Table A.1

Engineered Features of the Local Dataset

total_error_rate	dst_host_error_diff	error_rate_diff	flag_SF	same_srv_rate	dst_host_srv_count	Label (1 = Attack, 0 = Normal)
0	0	0	TRUE	1	255	1
0.5	1	1	FALSE	0.03	7	1
0.5	0.88	1	FALSE	1	249	0
0	0	0	TRUE	1	255	0
0.5	1	1	FALSE	0.03	6	1
0	0	0	TRUE	1	255	0
0	0	0	TRUE	1	255	0
0	0.04	0	TRUE	1	8	1
0	0	0	TRUE	1	107	1
0	0	0	TRUE	1	255	0
0	0	0	TRUE	1	255	0
0	0.04	0	TRUE	1	226	0
0	0	0	TRUE	1	91	0
0.5	1	1	FALSE	0.04	11	1
0	0	0	TRUE	1	255	0
0	0	0	TRUE	1	5	0
0	0	0	TRUE	1	3	0
0.5	1	1	FALSE	0.11	9	1
0.25	0.02	0.5	FALSE	1	255	0
0	0	0	TRUE	1	35	1

Note: The features listed in this appendix are derived features computed programmatically from the raw network traffic and log data. Due to privacy and security constraints, raw packet-level data are not included. These engineered features represent the actual inputs used for model training and evaluation.

APPENDIX B
Key Feature Definitions

Table B.1

Description of the Most Influential Features

Feature Name	Description
total_error_rate	Overall error rate across network connections, reflecting abnormal communication behavior
dst_host_error_diff	Difference in error rates observed at the destination host, indicating suspicious or unstable host activity
error_rate_diff	Variation in error rates between consecutive connections, capturing abrupt anomalous behavior
flag_SF	Connection status flag indicating a successfully established and terminated session
same_srv_rate	Percentage of connections using the same service, useful for identifying service-based attack patterns
dst_host_srv_count	Number of connections directed to the same destination host within a specific time window

Note: The feature definitions are provided to clarify the semantic meaning of the most influential attributes used in the proposed framework.

APPENDIX C

Hyperparameter Settings of the Implemented Models

Table C.1 – Autoencoder Hyperparameters

Parameter	Value
Optimizer	Nadam
Learning rate	0.0005
Loss function	Mean Squared Error (MSE)
Encoding dimension `	`max(8, int(input_dim × 0.2))
Regularization	L2 = 0.001
Dropout rate	0.2
Batch normalization	Enabled
Epochs	200
Batch size	64
Validation split	0.2
Early stopping	Patience = 15 (val_loss)
LR scheduler	ReduceLROnPlateau (factor = 0.5, patience = 8)
Decision threshold	Selected from 10th–99th percentile of reconstruction error

Table C.2 – CNN Hyperparameter

Parameter	Value
Convolution layers	3 Conv1D layers (64, 128, 256 filters)
Learning rate	0.001
Loss function	Binary cross-entropy
Encoding dimension `	`max(8, int(input_dim × 0.2))
Regularization	L2 = 0.001
Dropout rate	0.3 (conv), 0.5 (dense)
Activation function	ReLU
Epochs	10
Batch size	256
Kernel size	3
Early stopping	Patience = 5 (val_loss)
LR scheduler	educeLROnPlateau (factor = 0.2, patience = 3) |
Decision threshold	0.5
Optimizer	Adam

Table C.3 – Isolation Forest Hyperparameters

Parameter	Value
Number of estimators	200
Max samples	auto
Max features	0.8
Contamination	`0.1
Random state	42
Training strategy	Normal traffic only
Anomaly threshold	90th percentile of anomaly scores

Table C.4 – Hybrid Model Hyperparameters (CNN–LSTM + XGBoost)

Parameter	Value
CNN filters	64
Kernel size	3
LSTM units	32
CNN–LSTM optimizer`	`Adam
CNN–LSTM epochs	15
CNN–LSTM batch size	128
XGBoost estimators	200
XGBoost max depth	6
XGBoost learning rate	0.1
Ensemble weights	0.6 (CNN–LSTM), 0.4 (XGBoost)
Decision threshold	0.5

Table C.5 – Proposed Hybrid Pipeline Hyperparameters

Parameter	Value
Feature selection	SelectKBest (k = 50)
Dimensionality reduction	PCA (n_components = 0.99)
Class balancing	SMOTE (k_neighbors = adaptive)
Optimizer `	Adam
Regularization	L2 = 0.001
Learning rate	0.0005
Epochs	100
Batch size	4096
Early stopping	Patience = 15 (val_pr_auc)
LR scheduler	ReduceLROnPlateau (factor = 0.2, patience = 7)
Decision threshold	Optimized via PR curve (max F1)
Optimizer	Adam

Note: All hyperparameter values reported in this appendix are directly extracted from the implemented code used in the experimental evaluation to ensure transparency and reproducibility.

APPENDIX D
Ministry Approval for Data Access

Arab American University
Faculty of Graduate Studies

الجامعة العربية الأمريكية
كلية الدراسات العليا

2025/6/24

إلى من يهمه الأمر،

تسهيل مهمة بحثية

تحية طيبة وبعد،

تُهديكم كلية الدراسات العليا في الجامعة العربية الأمريكية أطيب التحيات، وبالإشارة الى الموضوع أعلاه، تشهد كلية الدراسات العليا في الجامعة أن الطالبة مرح راضي محمد حوى والتي تحمل الرقم الجامعي 202316926 هي طالبة ماجستير في برنامج امن المعلومات الالكتروني وتعمل على رسالة الماجستير الخاصة بها بعنوان:

"Improving Network Security -Based Anomaly Detection Using Machine Learning and Deep Learning "

تحت اشراف الدكتورة اماني عودة. نأمل من حضرتكم الإيعاز لمن يلزم لمساعدتها للحصول على المعلومات اللازمة للدراسة، علماً أن المعلومات ستستخدم لغاية البحث فقط وسيتم التعامل معها بغاية السرية، وقد أعطيت هذه الرسالة بناءً على طلبها.

وتفضلوا بقبول فائق الاحترام

عميد كلية الدراسات العليا
د. نوار قطب

كلية الدراسات العليا
FACULTY OF GRADUATE STUDIES

Jenin Tel: +970-4-2418888 Ext.:1471,1472 Fax: +970-4-2510810 P.O. Box:240
Ramallah Tel: +970-2-2941999 Fax: +970-2-2941979 Abu Qash - Near Alrehan
E-mail: FGS@aaup.edu ; PGS@aaup.edu Website: www.aaup.edu

Figure D.1 Official approval letter issued by the Ministry authorizing the use of anonymized network data for research purposes.

تحسين اكتشاف الشذوذ المستند إلى أمان الشبكة باستخدام التعلم الالي والتعلم العميق

مرح راضي محمد حوى

أعضاء اللجنة:

د.أماني عوده

د.حذيقة أشقر

د.محمد حسين

الملخص

تنتشر التقنيات الرقمية بوتيرة سريعة، مما يُتيح إمكانيات وفرصًا جديدة لتقديم الخدمات، ولكنه يُعرّضها أيضًا لتهديدات سيبرانية معقدة وديناميكية. لأنظمة كشف التسلل التقليدية قيودها الكامنة: إذ لا يُمكن للكشف القائم على التوقيعات اكتشاف هجمات اليوم صفر والهجمات الخفية، بينما عادةً ما تُنتج الأنظمة القائمة على الشذوذ عددًا كبيرًا جدًا من الإنذارات الكاذبة. وتزداد حدة هذه المشاكل في البيئات محدودة الموارد، مثل فلسطين، حيث يُؤدي نقص الأدوات المتقدمة ومجموعات البيانات المُصممة محليًا إلى التعرض لثغرات أمنية وتراجع قوة آليات الدفاع.

في هذه الدراسة، نُقدم نظام كشف تسلل هجين قائم على الذكاء الاصطناعي يجمع بين خوارزميات التعلم الآلي والتعلم العميق ويجري العمل على مرحلتين: أولًا، نمذجة واختيار أنسب النماذج المعيارية بناءً على مجموعات البيانات العالمية؛ ثم تكييف النموذج المُختار وضبطه بدقة من خلال تعلمه من مجموعة بيانات حركة مرور الشبكة الفلسطينية المُجمعة حديثًا. ومن خلال دمج المعلومات العالمية مع الأدلة المحلية، يُوازن الإطار بشكل طبيعي بين العمومية على نطاق واسع والخصوصية عبر مناطق مُختلفة. تشير التجارب إلى أن النموذج الهجين المقترح لا يتميز فقط بأداء أفضل من أنظمة كشف التسلل التقليدية، من حيث استقرار أعلى وقدرة أكبر على التكيف مع حركة مرور الشبكة الفعلية، بالإضافة إلى انخفاض عدد الإيجابيات الخاطئة. علاوة على ذلك، حقق النموذج دقةً تجاوزت 0.99 على مجموعات البيانات المرجعية العالمية، و0.988 على مجموعة البيانات الفلسطينية التي جُمعت حديثًا. وبينما لم تتجاوز الدراسات الحديثة المتطورة هذا المستوى من الدقة على مجموعات

البيانات العالمية، فإن النموذج المقترح يتفوق عليها في هذا السياق. أما على مجموعة البيانات المحلية، فلا يزال هذا العمل فريدًا من نوعه، إذ لم تُقيّم أي دراسة سابقة أداء نظام اكتشاف التسلل باستخدام بيانات حركة مرور الشبكة الفلسطينية الفعلية، مما يجعل دقة0.988 أول خط أساس موثق للمنطقة، وليست نتيجةً للمقارنة. وقد عززت هذه المساهمة الفريدة بشكل كبير قدرة النموذج على اكتشاف الهجمات الخاصة بالمنطقة، ووفرت وعيًا سياقيًا لا تستطيع مجموعات البيانات العالمية وحدها توفيره. علاوة على ذلك، أظهرت الدراسة أن السمات المتعلقة بالأخطاء ونشاط المضيف هي الأكثر أهمية في التمييز بين الأنشطة الضارة والحميدة، مما يسلط الضوء على أهمية خصائص السلوك الخاصة بالمجال في اكتشاف الشذوذ. إلى جانب التقدم الأكاديمي، يقدم هذا العمل إرشادات عملية حول كيفية نشر حلول كشف تسلل قابلة للتطوير وفعالة من حيث التكلفة، واعية بالسياق، في بيئات محدودة الموارد. تشير النتائج إلى أن دمج مجموعات البيانات العالمية والمحلية يمكن أن يؤدي إلى إطار عمل قوي ومفهوم للأمن السيبراني – مما يقدم النظام المقترح كمثال يمكن تكراره داخل فلسطين وخارجها للمناطق التي تعاني من مشاكل مماثلة في الأمن السيبراني.

الكلمات المفتاحية: التقنيات الرقمية، التهديدات السيبرانية، أنظمة كشف التسلل، التعلم الآلي، التعلم العميق.