A Hybrid Network Intrusion Detection System Using Machine Learning

By

Mark Njore

138014

A Computer Networks and Cyber Security Project II submitted to the School of Computing and Engineering Sciences in partial fulfilment of the requirements for the award of the Bachelor's Degree in Computer Networks and Cyber Security of Strathmore University

School of Computing and Engineering Sciences

Strathmore University

Nairobi, Kenya

July 2024

Declaration and Approval

I declare that this work has not been previously submitted and approved for the award of a Bachelor's degree by this or any other University. To the best of my knowledge and belief, the proposal contains no material previously published or written by another person except where due reference is made in the proposal itself.

Student Name: Mark Njore

Sign: ________________________ Date: ________________________

Approval

The Computer Networks and Cyber Security Project II proposal of Mark Njore was reviewed and approved (for examination) by:

Supervisor's Name: Dr Victor Rop

Sign: ________________________ Date: ________________________

Abstract

In modern network environments, the need for robust network intrusion detection systems (NIDS) is paramount to safeguard against evolving cyber threats. This project aims to address this problem by developing a Hybrid Network Intrusion Detection System (NIDS) that combines both anomaly-based and signature-based detection methods. The system aims to provide a flexible and adaptive approach to intrusion detection. Key components would include developing both parts of the Hybrid NIDS and integrating the Machine Learning model for robust intrusion detection to efficiently analyse network traffic and identify potential security incidents. In order to ensure the system's effectiveness and responsiveness, the project would use iterative development cycles, or sprints, which include feedback and can adjust in case of changing requirements. Evaluation through testing and validation methods would verify how reliable the system is in identifying both known and unknown threats, thereby improving network security in dynamic circumstances.

Table of Contents

Declaration and Approval ii

Abstract iii

Table of Contents iv

Acknowledgement viii

List of Figures ix

List of Tables x

List of Abbreviations xi

Chapter 1: Introduction 1

1.1 Background 1

1.2 Problem Statement 2

1.3 General Objective 3

1.4 Specific Objectives 3

1.5 Research Questions 3

1.6 Justification 4

1.7 Scope 4

1.8 Limitations and delimitations 4

Chapter 2: Literature Review 6

2.1 Introduction 6

2.2 Understanding Intrusion Detection 6

2.3 Intrusion Detection Challenges 8

2.4 Review of Existing Systems 11

2.4.1 Signature-Based Intrusion Detection Systems 11

2.4.2 Anomaly-Based Intrusion Detection Systems 12

2.4.3 Hybrid Intrusion Detection Systems 14

2.5 Existing gaps 15

2.6 Conceptual Framework 16

Chapter 3: Methodology 18

3.1 Introduction 18

3.2 Research Approach 18

3.3 Methodology 18

3.3.1 Planning and Scope 19

3.3.2 Product Backlog 19

3.3.3 Design 19

3.3.4 Sprint Planning 20

3.3.5 Sprint Backlog 20

3.3.6 Sprint Execution 20

3.3.7 Sprint Review 20

3.3.8 Sprint Retrospective 20

3.3.9 Launch Final Software 20

3.4 Analysis of the System 21

3.5 Deliverables 21

3.5.1 The Proposed Hybrid Network Intrusion Detection System 21

3.5.2 Research Proposal and Documentation 22

3.5.3 Final Presentation and Demo 22

3.6 Tools and Techniques 22

3.6.1 Python 22

3.6.2 Jupyter Notebook/Google Colab 23

3.6.3 Scapy 23

3.6.4 Pandas 23

3.7 Ethical Considerations 23

Chapter 4: System Analysis and Design 25

4.1 Introduction 25
4.2 System Requirements 25
4.2.1 Functional Requirements 25
4.2.2 Non-Functional Requirements 26
4.3 System Architecture Diagram 26
4.4 System Analysis Diagrams 27
4.4.1 Network Diagram 27
4.4.2 Use Case Diagram 28
4.4.3 Sequence Diagram 29
4.4.4 Flowchart 30
4.4.5 State Transition Diagram 32
Chapter 5: Implementation and Testing 33
5.1 Introduction 33
5.2 System Specifications 33
5.2.1 Hardware Specifications 33
5.2.2 Software Specifications 34
5.3 Description of the Dataset 35
5.4 System Testing 35
5.4.1 Testing of Functional Requirements 36
5.4.2 Testing of Non-Functional Requirements 37
5.5 Discussion of Results 38
Chapter 6: Conclusions and Recommendations 40
6.1 Conclusions 40
6.2 Recommendations 40
6.3 Future Works 40
References 42

Appendices 49

Appendix A: Gantt Chart 49

Acknowledgement

Even for a moment, I don’t fool myself into believing that this work would have been possible without the help of others. That being said I hope I will be able to thank everyone who has helped me in the next few lines. First, I would like to thank my supervisor, Dr Rop, for all support provided to me during this period. I have him to thank for guiding me through the process and helping with the corrections made in my research. I would also like to express my gratitude towards my colleagues at the university whose inputs were also crucial for my research. I would like to thank Dr Rop again for providing tips on how to work on the project and how to manage the project timewise as the unit lecturer. Finally, I would like to thank my family for their support during this period.

List of Figures

Figure 2.1 A classification of intrusion detection systems 7

Figure 2.2 Signature-based IDS architecture 12

Figure 2.3 Anomaly-based IDS architecture 13

Figure 2.4 Conceptual Framework 17

Figure 3.1 Scrum methodology overview 19

Figure 4.1 System Architecture Diagram 27

Figure 4.2 Network Diagram 28

Figure 4.3 Use Case Diagram 29

Figure 4.4 Sequence Diagram 30

Figure 4.5 Flowchart 31

Figure 4.6 State Transition Diagram 32

List of Tables

Table 5.1 Table of Hardware specifications 33

Table 5.2 Table of Functional Requirements Testing 36

Table 5.3 Table of Non-Functional Requirements Testing 37

List of Abbreviations

AI – Artificial Intelligence	

AI – Artificial Intelligence

AIDS – Anomaly-based Intrusion Detection System

ANN – Artificial Neural Networks

CIA – Confidentiality, Integrity and Availability

CNN – Convolutional Neural Network

DL – Deep Learning

DoS – Denial of Service

DDoS – Distributed Denial of Service

HIDS – Host Intrusion Detection System

IDS – Intrusion Detection System

IoT – Internet of Things

KNN - K-Nearest Neighbour

LSTM – Long Short-Term Memory

ML – Machine Learning

NIDS – Network Intrusion Detection System

NN – Neural Networks

RNN – Recurrent Neural Networks

SIDS – Signature-based Intrusion Detection System

SVM - Support Vector Machine

Chapter 1: Introduction

1.1 Background

In today's world, with computer networks playing such an important part in sustaining the operations of businesses and organizations, it is very important that we safeguard their security. The increasing number and complexity of cyberattacks brings about problems on this topic of network security. This study's main objective is to investigate intrusion detection systems (IDSs) specifically in computer networks. These systems are made to identify malicious network traffic, which according to Cruz-Cunha & Mateus-Coelho, (2020) includes any suspicious connections or data transmitted inside a network. This category includes a wide range of cyberthreats, such as viruses, ransomware, phishing scams, and distributed denial of service (DDoS) attacks.

The vulnerabilities and risks that a network could experience have increased due to the growth in the number of internet-connected devices and the how much more everyone depends on online services. The effects are extensive and affect people, companies, and institutions in every way. An example of this is identity theft and data loss. An NTSC report from 2019 states that approximately 620 million account details were compromised by hackers and afterwards sold on the dark web (NTSC, 2020). Companies may suffer from negative publicity, legal issues, and financial losses. In fact according to projections, cybercrime is expected to cost the world economy $10.5 trillion every year by 2025 (Sausalito, 2020).

Institutions could experience operational issues and a loss in trust, especially those that deal with sensitive data like healthcare and banking. This is a concern as according to the 2024 IBM Security X-Force Threat Intelligence Index, one of the more prominent cyberattacks in 2023 was infostealer malware, which experienced a 266% increase in activity (Piazza, 2024). There are serious repercussions if this problem of detecting malicious traffic is not resolved. If left unchecked can result in long-lasting security vulnerabilities that give attackers continual access to exploit networks. This might result in widespread data breaches, significant financial losses, and a decline in public confidence in digital systems.

Network security and intrusion detection have been the subject of in-depth research by academics and industry professionals (Dina & Manivannan, 2021). Standard techniques, including signature-based detection, are useful for recognizing recognized threats, but they are frequently ineffective against recently developed or complex attacks (Talukder et al., 2023). Artificial intelligence and machine learning techniques have been applied recently to improve the identification of already-existing systems`(Dina & Manivannan, 2021). Despite these efforts, there are still substantial gaps. Many of the systems currently in use suffer from high false positive rates, scalability problems, or inadequate zero-day attack detection capabilities.

More resilient, flexible, and intelligent systems are required in order to reliably differentiate between malicious and legitimate communications in real time. Significant advancements have been made in the evolution of current systems, which range from simple pattern matching methods to advanced anomaly detection systems. Improving detection accuracy, decreasing false positives, and increasing systems' ability to react to zero day threats are the gaps that need to be addressed when it comes to intrusion detection (Musa et al., 2020; Talukder et al., 2023).

This research aims to develop a hybrid network intrusion detection system that leverages machine learning techniques to improve the detection of malicious traffic. By addressing the current gaps, this study seeks to contribute to a more secure network infrastructure.

1.2 Problem Statement

This research focuses on the essential issue of accurately detecting cyber threats such as DOS attacks, zero-day exploits, polymorphic malware, and stealthy intrusion techniques. This is a critical concern since it increases the danger of data breaches, financial losses, operational disruptions, and reputational harm. The challenge is made worse by the dynamic nature of cyber threats, as attackers are always coming up with new ways to avoid detection.

High false positive rates, detection delays and an inability to effectively respond to newly discovered threats are all problems with current systems (Dina & Manivannan, 2021). While it is true that current systems frequently fall short of these ideal capabilities, in an ideal world, network systems should reliably detect and categorize harmful traffic in real-time, reducing false positives and delay in response. They have trouble identifying new and sophisticated threats, which causes them to fail to recognize threats, react slowly, and rely more on human input for threat analysis. This is a point of concern as according to the 2017 Symantec Internet Security Threat Report, more than three billion zero-day attacks were reported in 2016, and the volume and intensity of the zero-day attacks were substantially greater than the previous (Symantec, 2017).

The gap between expectations and reality has significant consequences for those involved in different sectors. Individuals, businesses, and government institutions are all affected, with each facing distinct challenges and consequences. This gap has led to operational, financial, and other consequences. For example, according to an IBM report, the average global cost of a data breach in 2023 was estimated to be USD 4.45 million, a 15% rise over the previous three years (IBM, 2023). This reduces trust in digital systems, limits innovation, and discourages efforts to create a secure online environment. Closing this gap is essential to safeguarding operations, protecting digital assets, and building a strong cybersecurity infrastructure.

1.3 General Objective

The general objective is to develop a hybrid network intrusion detection system for monitoring of network traffic.

1.4 Specific Objectives

i. To analyse the current techniques used in intrusion detection.

ii. To evaluate the challenges associated with intrusion detection.

iii. To design and develop the proposed hybrid network intrusion detection system for monitoring of network traffic.

iv. To validate the proposed hybrid network intrusion detection system for monitoring of network traffic.

1.5 Research Questions

i. What are the intrusion detection techniques currently in use?

ii. What are the challenges associated with intrusion detection?

iii. How can a hybrid network intrusion detection system be designed and developed?

iv. How can a hybrid network intrusion detection system be validated?

1.6 Justification

The urgent need to find solutions to the shortcomings of the IDS technologies in use today justifies the research project. The study is essential because cyber threats are becoming more and more dangerous for people, companies, and government organizations. The proposed methodology could potentially improve detection accuracy and adaptability to new threats, resulting in reduced financial losses, reputational damage, and regulatory penalties. This research would contribute to a more secure and resilient digital environment by not just advancing cybersecurity knowledge but also translating theoretical advances into practical solutions.

The research would improve privacy protection, operational continuity, and consumer confidence in digital systems by filling important gaps in current traffic and threat analysis procedures. In today's cybersecurity landscape, the strategic advantage organizations could achieve by implementing it, along with compliance assurance and reduced cyber risk exposure, illustrates the importance and significance of this research project.

1.7 Scope

The project's scope includes designing, creating, and testing a hybrid network intrusion detection system which utilizes machine learning techniques. Its main objective will be network traffic analysis. The software implementation of the suggested system will be included in the project. Since the primary goal of this project is detection, the mitigation and prevention of subsequent attacks are not included in its scope. The effectiveness of the detection system will be assessed using a number of performance measures, including detection accuracy. However, the scope does not include the deployment of the system in a large-scale network environment or the assessment of its performance under high-volume traffic. Instead, the model will be evaluated in a managed environment.

1.8 Limitations and delimitations

Although the scope is ambitious, several limitations and obstacles are expected. The first is the availability of data. There might not be as many labelled datasets available for testing and training machine learning models. Another is resource limitations as it's probable the computational resources needed to build an extensive system with real-time analytics and machine learning training will not be available. The other limitation is that because cyberthreats and attack techniques are dynamic, the machine learning model may need to be updated and modified often. Time is also another one of the limitations. The project had a deadline since it was a two-semester project. Because of this, the project's scope is limited to the detection of intrusions and not their prevention.

One more limitation is that there is no access to suitable real-life network environments for testing. This access would have allowed the project to test the model in real-world scenarios for evaluation. The alternative choice of conducting the test virtually or in simulation was chosen. A restricted skill set was another drawback. The team working on the project does not have expert programming skills. As a result, the project will use comparatively simpler programming techniques and languages and its complexity will not be as high as that of industry level systems.

Chapter 2: Literature Review

2.1 Introduction

The objective of this literature review is to better understand and investigate the previously described issue of identifying and analysing malicious traffic within computer networks. An overview of related books, articles, and publications that focus on the topic of intrusion detection systems (IDS) will be provided in this chapter. This will critically evaluate previous research, spot trends, look at the current state of this field's research, identify any gaps in the literature, and show the importance of the solution’s contribution to the field through this review. The analysis and critical assessment of previous research will lay a strong foundation for the development as well as implementation of the suggested solution.

2.2 Understanding Intrusion Detection

Intrusion detection, which focuses on detecting harmful activities and unauthorized access to a computer or network, is one of the most crucial components of cybersecurity (Ozkan-Okay et al., 2021). According to NIST (Bace & Mell, 2001), an intrusion is an effort to successfully get past a computer or network's security measures or compromise CIA. Intrusion detection is the process of identifying such threats before they have a chance to cause a significant amount of damage to the network by analysing network traffic for unusual behaviour (Jie Li et al., 2018). The intrusions could be external intrusions (attacks from outside the system) or internal (attacks from inside the system) (Buczak & Guven, 2016).

Jim Anderson first proposed the notion of an IDS in 1980 (Anderson, 1980). Since that time, the primary goal of intrusion detection systems (IDSs) has been to ensure the availability, integrity, and confidentiality of networks and their resources (Bayerl et al., 2017). IDSs are also used for other purposes, such identifying problems with security policy, informing people about current threats, and putting off people from participating in cybercriminal activity. The significance of intrusion detection systems has grown even more important in maintaining strong network security due to the increasing complexity of cyber threats (Ozkan-Okay et al., 2021).

IDSs can be categorized based on their monitoring and detection methods. Host-based IDS (HIDS) monitor individual devices, analysing system calls, logs, and file changes. Network-based IDS (NIDS) examine network traffic for suspicious activities. A NIDS would protect all hosts on the network, unlike the HIDS which would protect only one (Ni, 2023). These can then be further categorized into Signature-based, Anomaly-Based and Hybrid IDSs.

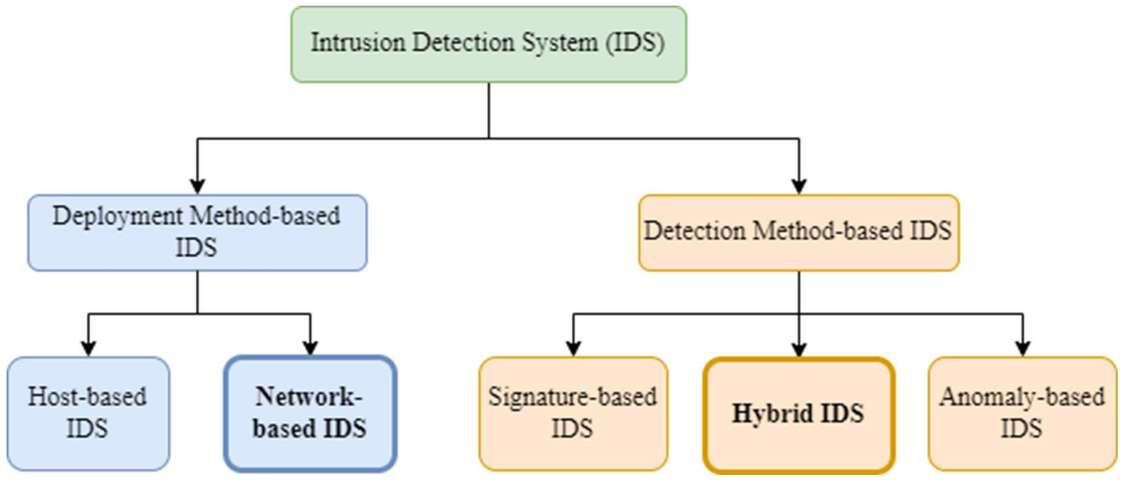

Intrusion Detection System (IDS)
Deployment Method-based IDS
Detection Method-based IDS
Host-based IDS
Network-based IDS
Signature-based IDS
Hybrid IDS
Anomaly-based IDS

Figure 2.1 *A classification of intrusion detection systems*

Network traffic anomalies are monitored, and analysed by a variety of methods and procedures that are part of the processes involved in network intrusion detection. Data collecting, feature extraction, pattern identification, anomaly detection, alert production, and evaluation are the core procedures. Data collection involves using network taps or sensors positioned at key network nodes to record network traffic data. Network traffic is typically recorded using either a flow-based or packet-based format (Ring et al., 2019). The technique of extracting appropriate characteristics from the unprocessed network data in order to analyse the traffic patterns and behaviours is known as feature extraction. The processes of analysis and detection that follow depend on the feature extraction.

Network traffic can then be examined using methods like machine learning (ML) to identify patterns. Using past network traffic data, these algorithms can be trained to differentiate between normal and suspicious network traffic (Chalapathy & Chawla, 2019). Anomaly detection could also be sued which involves detecting variations from baseline network behaviour using techniques such as statistical analysis and clustering. These anomalies may be signs of potential intrusions or hostile activity in the network. When suspicious activity is detected, the IDS generates alerts to notify the appropriate person, such as the network administrator, who can then acts accordingly (Akalanka Mailewa & Suman Thapa, 2020).

There is a noticeable difference between the expectation and the reality in the field of network intrusion detection. Expectations can be high, with the goal being intrusion detection systems (IDS) that can accurately identify malicious activity without any false positives (Aminanto & Kim, 2016). The reality, on the other hand, falls short of this. One example of the difference is the frequency of false alarms, or false positives, which occur when an intrusion detection system (IDS) misidentifies normal behaviour as malicious, which then causes unnecessary alerts and disruptions (Burgio, 2019).

The truth is that cyber threats are evolving and becoming more and more sophisticated, which makes older intrusion detection systems less effective against newer attack techniques that are being developed by attackers (Ozkan-Okay et al., 2021). Moreover, the need for constant upgrades and adjustments conflicts with the expectation of easy integration and compatibility of Intrusion Detection Systems with today's network infrastructures. Significant financial and human resources are also needed for the setup and maintenance of an IDS, and enterprises may find it difficult to keep up with the evolving cyberthreats (Talukder et al., 2023).

2.3 Intrusion Detection Challenges

There are several significant issues that the intrusion detection process has to resolve, which have restricted the effectiveness of the intrusion detection systems. An important challenge is the high false positive rate, which causes harmless behaviour to be wrongly reported as harmful. This frequently happens because many intrusion detection systems (IDS) use strict rule-based procedures that aren't adequate enough to distinguish between malicious and legitimate activity in network environments. High rates of false positives can overwhelm security teams with notifications, resulting in alert fatigue and sometimes obscuring genuine risks.

Quickly adapting to new threats is another difficulty. The challenge occurs because a lot of intrusion detection systems (IDS) are made to rely on predefined signatures to detect known threats. Networks are then left susceptible for a while when new forms of attacks emerge because signatures take time to update and propagate. In fact, keeping the signature databases updated is one of the most distinct challenges associated with an IDS (Folorunso et al., 2016). Systems that are unable to identify new threats are susceptible to sophisticated exploits and zero-day attacks, which may expose sensitive data and vital systems.

Furthermore, there are notable computational resources associated with implementing an effective IDS (Liao et al., 2013). Although there is potential for better detection accuracy with machine learning-based intrusion detection systems, the training and update of models necessitates large computational resources. For firms with little resources and technical expertise, this would affect detection performance, especially in big, dynamic networks (Yang et al., 2022).

Compatibility and integration with current network infrastructures present another challenge. IDSs must function seamlessly in a variety of complex network environments, which frequently calls for customized setups and continuous modifications. Incompatibility could result in network security coverage gaps and performance bottlenecks, which increase the likelihood of undetected breaches.

Scalability is also a significant challenge since intrusion detection systems (IDS) have to process enormous amounts of network traffic in real-time, which may exhaust their resources and degrade performance. Scalability problems make it more difficult for the IDS to manage the increasing amount of network traffic, which can cause degraded performance and higher latency, which then leads to slow down business processes and cause operational disruptions. This is made worse by the intricate setup and management that these systems require, which frequently call for specialized training and a large amount of manual labour to keep the IDS operational and up to date.

Another challenge for intrusion detection systems (IDS) is obtaining reliable training data (Shone et al., 2018). Effective IDS, particularly those using machine learning, require vast amounts of high-quality data to accurately identify and respond to threats. Unreliable training data, would affect the IDS's overall effectiveness, making it less capable of protecting networks and resources from evolving cyber threats (Kok et al., 2019).

Another issue is that hackers can use sophisticated and evasive techniques to get around detection systems by taking advantage of flaws in IDS implementation and algorithms (Ozkan-Okay et al., 2021). Another distinct problem is dealing with insider threats, which come from within the company and might be hard to find using traditional IDS techniques. Since intrusion detection systems frequently need to examine network traffic that can contain sensitive data, privacy concerns are becoming more and more crucial. It can be difficult to strike a balance between privacy safeguards and effective intrusion detection; careful planning and execution are needed to prevent violating users' rights.

These difficulties may give rise to other issues. Businesses may suffer from serious data breaches, monetary losses, and reputational damage (Ahmad et al., 2020). Legal implications and fines could result from a breach in regulatory compliance. Long-term business connections and market position may also be impacted by the constant threat of unknown assaults, which can also undermine trust among partners and consumers.

Various measures have been implemented to address these challenges. To decrease false positives as well as improve the identification of new threats, artificial intelligence and machine learning approaches have been applied (Ahmim et al., 2018). Neural Networks (NN) (sometimes called Artificial Neural Networks or ANN), Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) are related subjects of study; each of these fields, in this sequence, can be thought of as a sub-field of the preceding one (Dina & Manivannan, 2021). Deep learning, for instance, is a promising technique for intrusion detection as it can automatically identify correlation in the data (Tang et al., 2016). And, other techniques such as convolutional neural network (CNN) (Vinayakumar et al., 2017), K-Nearest Neighbor (KNN) (Talukder et al., 2023), (recurrent neural network (RNN) (Yin et al., 2017), long short-term memory (LSTM) (Roy et al., 2017) and support vector machine (SVM) models (Muneer et al., 2024), are popular in intrusion detection. Also, large-scale network traffic may now be handled more effectively because of the development of distributed and cloud-based intrusion detection systems (IDS) that improve scalability and performance (Ma, 2020). The usage of ML is comprehensive in the network security domain (Prasad & Rohokale, 2019).

So far, numerous businesses have reported increased detection capabilities and decreased false positive rates as a result of these successful attempts to address the problems. Although smaller businesses may now find it difficult to deploy modern IDS systems since they often require higher fees and are more complex. Even while machine learning-based intrusion detection systems have showed potential, they still need a lot of training data and can struggle with complicated evasion strategies.

2.4 Review of Existing Systems

2.4.1 Signature-Based Intrusion Detection Systems

The most popular kind of intrusion detection systems (IDS) is the signature-based Intrusion Detection System (Shone et al., 2018). In fact, most of the earlier IDS were signature based (Musa et al., 2020). An SIDS compares incoming network traffic to a database of known attack signatures in order to identify known threats (Ni, 2023). These systems provide a strong line of defence against well-documented attacks and are essential for discovering and addressing threats that correspond with pre-established patterns (Talukder et al., 2023). This is why SIDS have been deployed in many fields such as the Internet of Things (IoT) industry (Ioulianou et al., 2018).

In order for the Signature IDS to function, network traffic must be continuously monitored and compared to a predetermined set of rules or signatures. Every signature is a distinct pattern connected to a known assault. An alarm is sent out or a predetermined action is taken by the IDS, such as logging the event or blocking the traffic, when it finds traffic that corresponds to one of these signatures.

A specialized language is used to define the signatures, enabling exact definition of traffic patterns. Frequent updates to the signature database guarantee that the IDS is capable of identifying the most recent threats. Using an extensive and continuously updated knowledge base of attack signatures, this method successfully resolves the problem of identifying known attacks (Buczak & Guven, 2016).

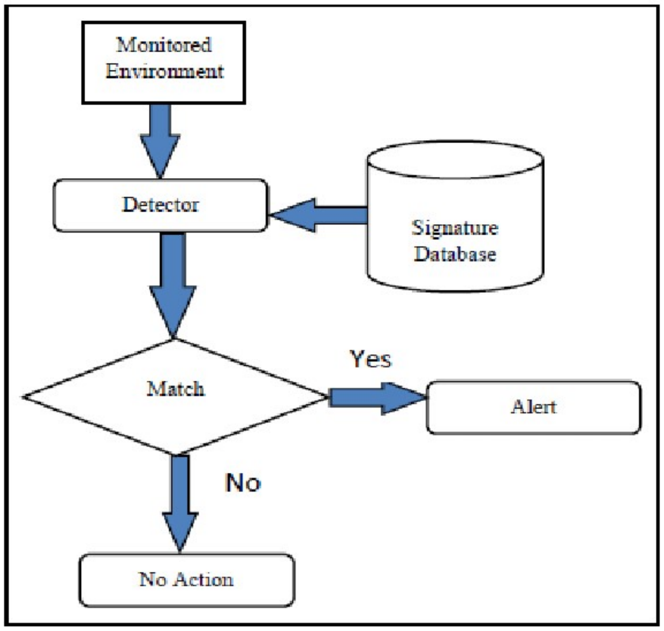


Figure 2.2 *Signature-based IDS architecture*

Although the design of rules and updates for classic signature-based IDS is largely done manually, newer technologies such as ML have been incorporated in various ways to improve its functionality (Aldwairi et al., 2017; Gunduz & Das, 2020; Malek et al., 2020). Using machine learning to automatically create and update signatures based on traffic patterns can save manual labour and increase the system's ability to react to emerging risks (Alsughayyir et al., 2019).

Artificial intelligence (AI) approaches have also been used to prioritize alerts according to the potential severity of the discovered threats and to increase the accuracy of signature matching. These technologies assist reduce false positives and guarantee that the most serious risks are dealt with right away (Aminanto & Kim, 2016).

2.4.2 Anomaly-Based Intrusion Detection Systems

Detecting anomalous patterns or behaviors in network traffic that can point to a possible security risk is the goal of anomaly-based intrusion detection systems (AIDS) (Otoum & Nayak, 2021). Unlike signature-based systems, which rely on known attack patterns, anomaly-based IDS should be able to detect zero-day attacks by recognizing deviations from established normal behaviour (Dina & Manivannan, 2021).

A baseline of typical network behaviour is established during a learning phase by anomaly-based intrusion detection systems. This baseline is built by monitoring factors including protocol usage, bandwidth utilization, and normal user actions using statistical models, machine learning algorithms, or artificial intelligence (AI) techniques. The system then compares the current activity to the baseline to identify anomalies, and it keeps monitoring network traffic in real time (Ma, 2020). This creates the advantage of being able to have a customized normal activity profile for different networks and applications (Guo et al., 2016).

Because AIDS focus on departures from usual behaviour to discover unexpected threats, they are therefore especially useful in situations where the threat landscape is constantly evolving. For example, behaviours such as buffer overflow and DoS would misguide a SIDS but not an AIDS because these behaviours are difficult to represent as signatures. But depending too much on a baseline establishment and learning phase might occasionally result in false positives before the system has adequately learnt the network's normal behaviour.

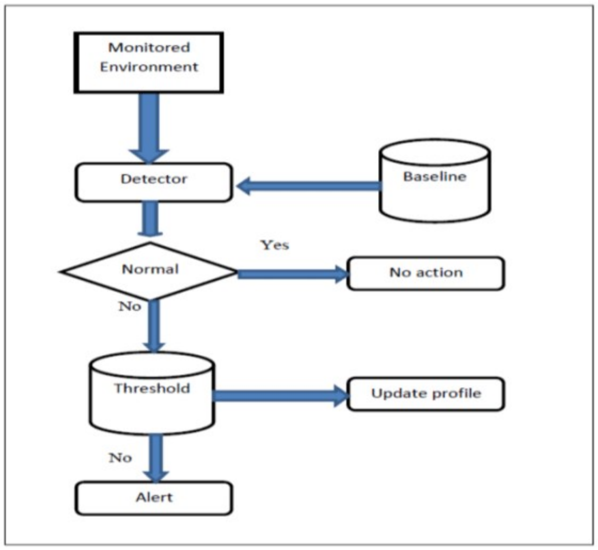


***Figure 2.3** Anomaly-based IDS architecture*

Anomaly-based intrusion detection systems are greatly improved by emerging technologies like artificial intelligence, deep learning, and machine learning (Aldweesh et al., 2020; Aljawarneh et al., 2018). These systems may learn from network data in an adaptable manner thanks to machine learning algorithms, which increases their anomaly detection accuracy

(Aminanto & Kim, 2016). Multiple-layer neural networks used in deep learning techniques can examine complex data patterns and increase detection accuracy by spotting small anomalies that more straightforward models would overlook (Alsughayyir et al., 2019).

AI improves anomaly-based intrusion detection systems by offering sophisticated analytics and judgment. With the help of these technologies, anomaly-based intrusion detection systems (IDS) can remain competitive and offer flexible, adaptable security solutions.

2.4.3 Hybrid Intrusion Detection Systems

The advantages of anomaly- and signature-based detection techniques are combined by hybrid intrusion detection systems (IDS) to offer a more complete security solution. The drawbacks of employing either technique alone is addressed by the hybrid IDS, which successfully detects known and unexpected threats by utilizing the strengths of both approaches.
A hybrid intrusion detection system uses a multipronged detection strategy to tackle the challenge of identifying a wide variety of security threats. This algorithm is especially good at spotting well-known attack signatures and can also spot odd behaviour patterns that can point to new or unidentified threats (Talukder et al., 2023).

It functions by executing both anomaly and signature-based detection engines concurrently. The signature-based component compares incoming traffic to a database of attack signatures using established rules to identify known threats. This part works quite well at identifying known risks with identifiable patterns. In contrast, the anomaly-based component creates a baseline of typical network behaviour and keeps an eye out for any deviations from it in order to identify any potential new threats.

The problem of identifying a broad variety of threats is resolved by the hybrid intrusion detection system's extensive coverage. While the anomaly-based component improves security by identifying new and emerging threats that might not have existing signatures, the signature-based component makes sure that well-known threats are quickly discovered.

Artificial intelligence and machine learning are two emerging technologies that are essential to the operation of hybrid IDS (Harish & Kumar, 2017). By continuously learning from fresh data, machine learning algorithms improve the anomaly detection component and increase the system's capacity to recognize new threats (Fraley & Cannady, 2017). Deep learning algorithms, for example, are very helpful in detecting zero-day threats because they can recognize patterns in data (Ahmad et al., 2020).

The accuracy and effectiveness of hybrid IDS are further enhanced by artificial intelligence approaches. By using AI to rank warnings according to the seriousness of anomalies found, security teams may concentrate on the most serious risks while decreasing the amount of false positives (Aminanto & Kim, 2016). Furthermore, greater insights into abnormalities can be obtained through AI-driven analytics, which speeds up and improves decision-making.

2.5 Existing gaps

Even with the advances in intrusion detection systems (IDS), there are still significant shortcomings and gaps in the current solutions that prevent them from effectively addressing the issue of network security in its complete form.

One is the heavy reliance of signature-based intrusion detection systems on predetermined attack signatures. Although they work well against known threats, they have trouble identifying emerging or new threats, such zero-day exploits, that don't match the signatures that are currently in use (Ni, 2023). Networks are left open to new assaults due to this constraint until new signatures are created and implemented. This reliance also impedes performance as the search speed for one signature reduces as the size of the database increases (Ni, 2023).

Another is that anomaly-based intrusion detection systems frequently experience considerable false positive rates due to a misreading of harmless but unexpected activity as threats (Ahmad et al., 2020). This may result in an excessive number of notifications for security teams, decreasing the detection process's overall effectiveness. Although AIDS have more false positives, SIDS tend to have more false negatives(Dina & Manivannan, 2021).

Complexity is another shortcoming. The goal of hybrid IDS is to bring together the advantages of anomaly and signature-based strategies. Although they provide better detecting capabilities, their deployment, management become more difficult and costly (Kok et al., 2019). It can be difficult to ensure the smooth integration and efficient functioning of both detection techniques, frequently needing a large amount of processing power and experience.

Many IDS are unable to instantly adjust to emerging threats. Although AI and machine learning have been combined to increase flexibility, these technologies are not always completely utilized, which prevents systems from updating their detection models in real time in response to the most recent threat intelligence. To keep these models efficient, they also need to be trained regularly with the new data obtained as a result of monitoring the network traffic, which could also take time (Ahmad et al., 2020).

There are also scalability issues as an IDS needs to expand in tandem with the size and complexity of network environments. Large-scale deployments of current systems frequently cause them to lose accuracy and performance, which results in ineffective threat identification and response.

Another gap is the integration with other security products. Firewalls, antivirus programs, and security information and event management (SIEM) systems are examples of other security products that must be integrated with intrusion detection systems (IDS) for effective cybersecurity. Lack of seamless integration capabilities in many IDS solutions can lead to security coverage gaps and impede coordinated threat response activities.

Even if they are sophisticated, the current IDS solutions have several flaws and inefficiencies that reduce their ability to offer complete network protection. IDS may be strengthened and improved to provide greater defence against known and new threats by filling up these gaps.

2.6 Conceptual Framework

To solve the issues mentioned in the preceding sections, the detection solution combines both anomaly-based intrusion detection systems (AIDS) and signature-based intrusion detection systems (SIDS). The objectives of this hybrid technique are to increase performance and scalability, decrease false positives, and improve detection accuracy.

Network packets are used to gather data about network traffic. These serve as the input for the AIDS and SIDS components. After that, the SIDS processes the incoming data by first matching patterns of known threats against a database of defined signatures. The data is classified as malicious and forwarded to the system administrator for additional action if a signature match is found. The AIDS then performs further analysis on the same input data.

Data pre-processing, which involves cleaning and transforming raw data; feature selection, which selects relevant features; model building, which involves training the machine learning model with historical data to identify patterns of normal and abnormal behaviour; and model evaluation, which assesses the precision and efficiency of the trained model in detecting intrusions, are the steps involved in this process.

The method makes use of two main storage elements. The SIDS component uses the rules database to store known attack signatures, while the data repository holds logs and historical data for model evaluation, training, and further analysis. This repository ensures a complete dataset for enhancing model accuracy by including traffic data from both previous harmful activity and normal traffic.

The system administrator is notified if either the SIDS or AIDS determine that traffic is suspicious. Attackers, legitimate users, and the system administrators who control and maintain the IDS are all actors interacting with the system. The system administrator would be responsible for managing of the IDS, handle alarms, update the rules database, and monitor system performance as a whole. Users generate normal network traffic, which serves as input by the IDS, whereas attackers are malicious entities attempting to penetrate the network and whose actions are targeted by it.

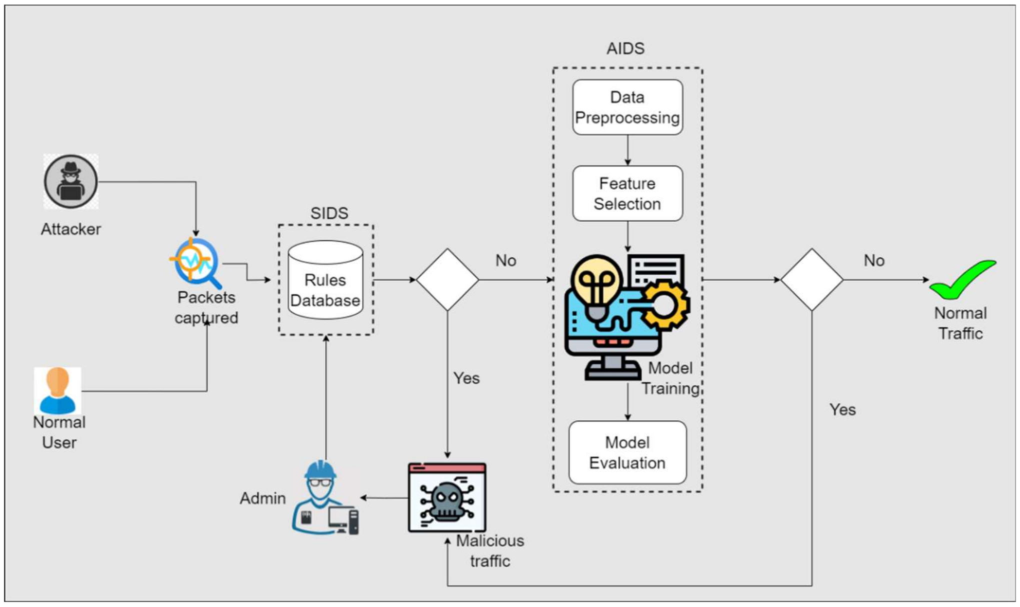


Figure 2.4 *Conceptual Framework*

Chapter 3: Methodology

3.1 Introduction

The term "system development methodology" describes a framework for organizing, arranging, and directing the creation of information systems. The techniques and tactics that will be applied in the creation and testing of the suggested hybrid NIDS are covered in further detail in this chapter.

3.2 Research Approach

Object-Oriented Analysis and Design (OOAD) will be the research approach used to develop the hybrid NIDS. This is because OOAD provides an organized approach to developing software systems by facilitating the identification of important entities, their relationships, and the functioning of the system. The approach consists of several stages, such as requirements gathering, system analysis, design, and implementation.

3.3 Methodology

The methodology known as Scrum has been selected for this project. The flexible and iterative nature of Scrum, a subset of Agile, makes it suitable for this project. Scrum's flexibility is one of the primary factors for its selection. Highly organized methodologies cause delays because they are unable to handle adaptations or unexpected situations. Scrum approaches, on the other hand, view change as a normal component of the development process. Sprints are the iterative cycles used by Scrum that allow for fast reaction to changing requirements, priority adjustments, and scope modifications.

Scrum's emphasis on iterative development makes it possible to complete project phases on a regular basis, which is another reason. When working on the hybrid NIDS, this iterative technique would enable faster feedback loops, which would result in the early discovery of complications and deviations. This reduces the need to repeat work by enabling adjustments to be integrated early. Scrum's iterative structure ensures a more productive development strategy.

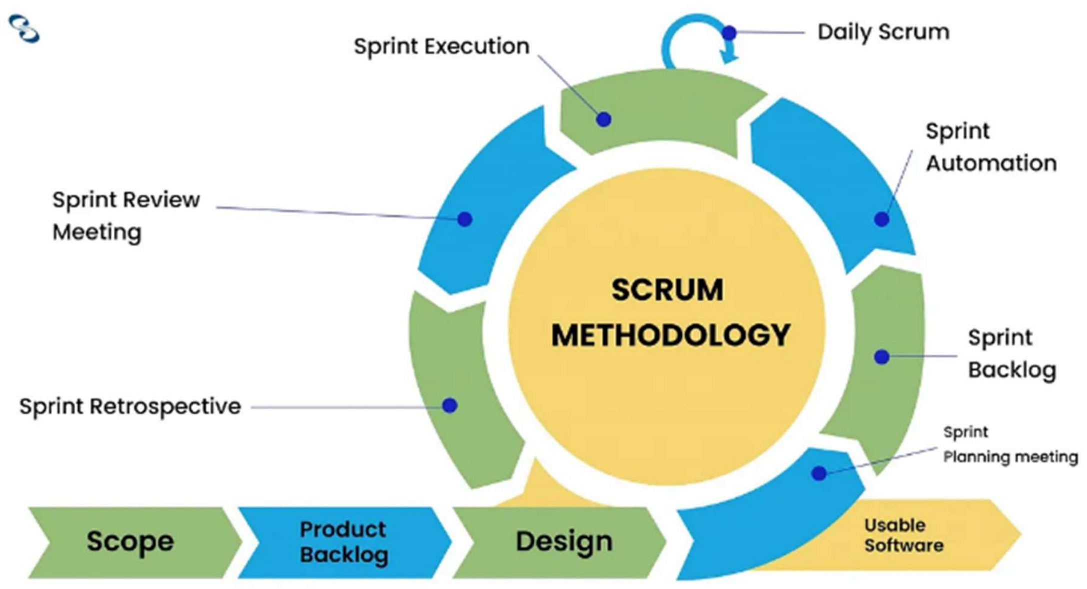


Figure 3.1 *Scrum methodology overview*

3.3.1 Planning and Scope

The planning phase would establish boundaries, goals, deliverables, limitations, and a defined vision, objectives, and scope for this project. In this stage, appropriate data must be gathered, processes must be mapped, and both the functional and non-functional requirements of the system must be determined, along with their feasibility.

3.3.2 Product Backlog

A prioritized list of features, improvements, and other system-related activities known as the product backlog. It will be updated on a regular basis in response to feedback and changes in requirements, guaranteeing that the most important and useful work is finished first.

3.3.3 Design

The design phase would focus on drawing out the hybrid IDS architecture in great detail. This entails deciding on suitable technologies, creating system parts, and making sure the design complies with the specifications and project parameters.

3.3.4 Sprint Planning

As mentioned before, the development of the hybrid IDS would be broken down into sprints. Selecting what to work on in the upcoming sprint from the previously mentioned product backlog is known as sprint planning. To ensure that the workload is reasonable within the sprint timeframe, this involves setting sprint goals, breaking down tasks, and evaluating the effort required.

3.3.5 Sprint Backlog

Tasks that must be finished in the next sprint make up the sprint backlog, which is a smaller portion of the product backlog. It offers a precise and targeted strategy for the deliverables that will be produced at the end of the sprint.

3.3.6 Sprint Execution

The goal of the sprint execution would be now to finish the tasks listed in the sprint backlog. Every day, problems would be addressed and progress tracked to keep things moving forward.

3.3.7 Sprint Review

The work being done is reviewed and given comments during the sprint. This guarantees that the system lives up to expectations and allows modifications based on input, keeping the project's alignment with the goal.

3.3.8 Sprint Retrospective

The goal of the sprint retrospective would be to to reflect on the previous sprint and identify what worked, what didn't, and how to make improvements for the next one. Over time, this process of continual improvement would contribute to improved performance and project outcomes.

3.3.9 Launch Final Software

After all sprints, reviews, and retrospectives are successfully completed, the final software launch takes place. It involves setting up the fully operational IDS and making sure it satisfies all requirements.

3.4 Analysis of the System

Recognizing the functional and non-functional requirements, identifying the IDS's components, and outlining how they interact are all part of the analysis process. The first step in requirements gathering is to list all of the requirements, both functional and non-functional.

In order simulate the structure and behavior of the IDS, a number of diagrams will be created using Object-Oriented Analysis and Design (OOAD). Data flow analysis will involve looking at the sources (network traffic), processing stages (data pre-processing, feature extraction), and sinks (alert logs, reports) of data as it flows through the system. This will guarantee that large amounts of network traffic are handled effectively. The system will be divided into smaller components by component identification, such as modules for reporting, alert management, feature selection, data pre-processing, and the detection engine.

Possible dangers related to creating and implementing the IDS will be evaluated through risk and feasibility analysis. This will cover project risks like resource limitations and schedule delays, as well as technical risks like algorithm accuracy and system performance.

Ultimately, validation and verification will guarantee that the system satisfies its specifications and operates as planned. While validation ensures the system satisfies expectations and accomplishes its intended goal, verification will verify that the system successfully performs certain functionalities. Through compliance with the Scrum framework and an organized analytical methodology utilizing OOAD, the system will be built with adaptability in mind, successfully tackling the problems associated with intrusion detection in dynamic network settings.

3.5 Deliverables

These will be the products of the project. These are:

3.5.1 The Proposed Hybrid Network Intrusion Detection System

The primary deliverable of this project will be the fully developed hybrid network intrusion detection system (IDS). This system will combine the strengths of both anomaly-based and signature-based detection methods to provide protection against known and unknown threats. The hybrid NIDS will be designed to monitor network traffic, identify suspicious activities, and generate alerts for potential security incidents.

3.5.2 Research Proposal and Documentation

The research proposal will outline the objectives, methodology, and expected outcomes of the project, providing a detailed plan for developing the hybrid NIDS. It will address the research questions, theoretical framework, and rationale for choosing specific methodologies and technologies. The proposal will also include a literature review to identify gaps in existing research and how the project aims to fill those gaps. Comprehensive documentation will be maintained throughout the project, capturing all aspects of the system's development. This will include detailed design documents, technical specifications, implementation details, and code documentation to ensure the system can be understood and maintained by future developers. Testing plans and results, including unit, integration, and system tests, will also be documented to demonstrate the system's reliability and effectiveness.

3.5.3 Final Presentation and Demo

The final deliverable will be a presentation and demonstration of the hybrid NIDS. This presentation will summarize the project, highlighting the objectives, methodology, challenges encountered, and the solutions implemented. It will include a live demonstration of the system in action, showcasing its capabilities in detecting and responding to network threats. The demo will illustrate key functionalities, such as real-time traffic monitoring, threat detection, and alert generation.

3.6 Tools and Techniques

3.6.1 Python

Python will be the programming language used to write the code in the project. It is ideal for building this project due to its extensive library support and ease of development. Python's readable syntax and rapid prototyping capabilities would further accelerate development, making it a powerful tool for constructing a hybrid NIDS.

3.6.2 Jupyter Notebook/Google Colab

Jupyter Notebook and Google Colab will be utilized as the primary tools for creating and sharing computational documents. Jupyter Notebook offers a streamlined, which is ideal for developing and presenting projects. Google Colab, a cloud-based alternative, provides similar functionality with added benefits such as access to powerful GPUs and seamless integration with Google Drive. Both platforms will be used to train the machine learning model for the system, ensuring flexibility and efficiency in the development process.

3.6.3 Scapy

Scapy is a powerful Python library used for packet manipulation and network analysis. It will be used for sniffing network packets and performing packet-level operations. It allows the system to capture, dissect, and analyze packets to detect suspicious activities.

3.6.4 Pandas

Pandas is a powerful data manipulation and analysis library. In the project, it will be used to read the .csv files containing the traffic data files for training the ML model.

3.7 Ethical Considerations

When developing and implementing the hybrid Network Intrusion Detection System (NIDS), ethical issues are of high priority. Since network traffic is monitored by NIDSs and may contain sensitive personal and corporate data, protecting user privacy is of utmost importance. To ensure that monitoring does not violate users' right to privacy, steps must be taken to anonymize and preserve data.

Since the NIDS would handle potentially sensitive information regarding network activities and threats found, data security is another crucial component. The information gathered and processed by the system need to be accessible only to authorized persons. Accountability and transparency are also crucial to the NIDS development process. To give clear reasoning for the detection mechanisms of the system, it is essential to record its activities, decision-making procedures, and algorithms. By being transparent, users are better able to understand how the system works and the rationale behind certain actions.

While designing and implementing the machine learning algorithms utilized in the hybrid NIDS, bias and fairness are important factors to take into account. In order to prevent biased detection conclusions that can unfairly target particular users or activities, algorithms must be trained on representative datasets. Since false alarms can have serious repercussions, addressing false positives and false negatives in an ethical manner is essential. False positives may result in pointless investigations and privacy violations. It is important to implement strategies aimed at minimizing these mistakes.

Lastly, it is important to take into account the possibility of the IDS being misused. It is important to make sure the system is only utilized for intrusion detection, as intended, and not for illegal eavesdropping or any other unlawful activities. The development and operation of NIDS may be done ethically by taking these ethical issues into account, guaranteeing the preservation of user rights, data security, openness, and fairness.

Chapter 4: System Analysis and Design

4.1 Introduction

This section outlines the requirements and architecture of the Hybrid Intrusion Detection System (IDS) that combines Signature-based (SIDS) and Anomaly-based (AIDS) detection methods. It includes the functional requirements which specify the system's core capabilities, and non-functional requirements as well. The diagrams in this section provide detailed insights into the system's deployment, interactions and process flow. Together, these diagrams assist in describing the design of the Hybrid IDS.

4.2 System Requirements

This section is divided into two parts, functional and non-functional requirements.

4.2.1 Functional Requirements

i. Real-time Analysis: The system should capture and analyze real-time network traffic to detect potential intrusions or abnormal activity.

ii. Rule-based Detection: The system should identify known threats using predefined attack rules and match traffic against these rules.

iii. Anomaly Detection: The system should analyze traffic patterns to detect deviations from normal behavior, potentially identifying unknown threats.

iv. Notification Generation: The system should generate and log notifications when suspicious or malicious traffic is detected, providing detailed information about the event.

v. Rule Management: The system should allow the creation, modification, and deletion of detection rules, enabling administrators to adapt to new threats.

vi. Logging: The system should log detected incidents that summarize network activity, threats, and performance.

4.2.2 Non-Functional Requirements

i. Reliability: The system should be highly reliable and capable of continuous operation without crashing or losing data

ii. Scalability: The system should be scalable, allowing it to handle increased traffic or larger network environments without degradation in performance

iii. Maintainability: The system codebase should be well-documented and modular, making it easier for developers to fix bugs, implement updates, or add new features.

iv. Resource Efficiency: The system should optimize resource consumption, minimizing CPU, memory, and disk usage to reduce the impact on network performance.

v. Accuracy: The system should be accurate, having few false negatives and false positives, ensuring accurate detection of malicious activity and preventing unnecessary disruptions.

4.3 System Architecture Diagram

This diagram illustrates the architecture of the Hybrid Intrusion Detection System (IDS) with integrated Signature-based Intrusion Detection System (SIDS) and Anomaly-based Intrusion Detection System (AIDS) components. The following are the components:

1. External Traffic: The process begins with incoming network traffic from external sources.

2. Packet Collector: This component captures and collects the incoming network packets for analysis.

3. Packet Parser: The collected packets are parsed to extract relevant information for further processing.

4. Rule Engine: Part of the SIDS, this component analyzes the parsed packets against predefined rules stored in the Rule Database to detect known attack patterns.

5. ML Model: This is part of the AIDS component. It uses machine learning techniques to detect anomalies or unusual patterns in the network traffic that may indicate previously unknown threats.

6. Alerting System: If either the Rule Engine or ML Model detects a potential threat, this component generates alerts to the Admin.

7. Log Files: The system maintains log files to record events, alerts, and system activities for future reference and analysis.

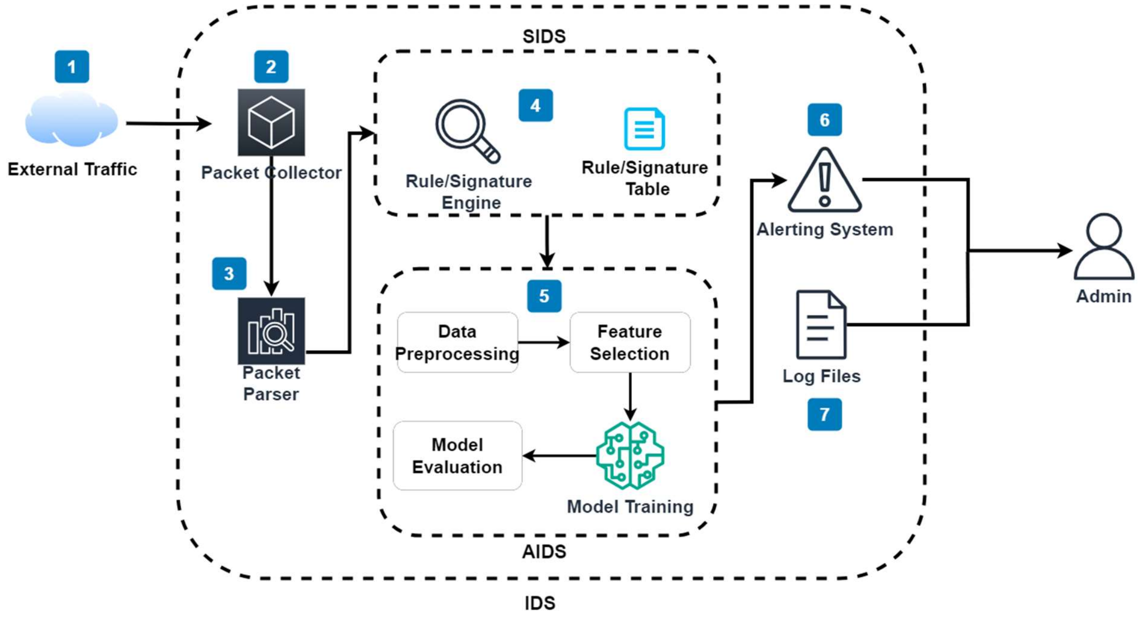


Figure 4.1 *System Architecture Diagram*

4.4 System Analysis Diagrams

4.4.1 Network Diagram

This is a simple illustration of how the IDS will be deployed in a network. It will be placed between the network firewall and the network switch. Placing it here instead of the edge of the network or between the firewall and router would improve its performance.

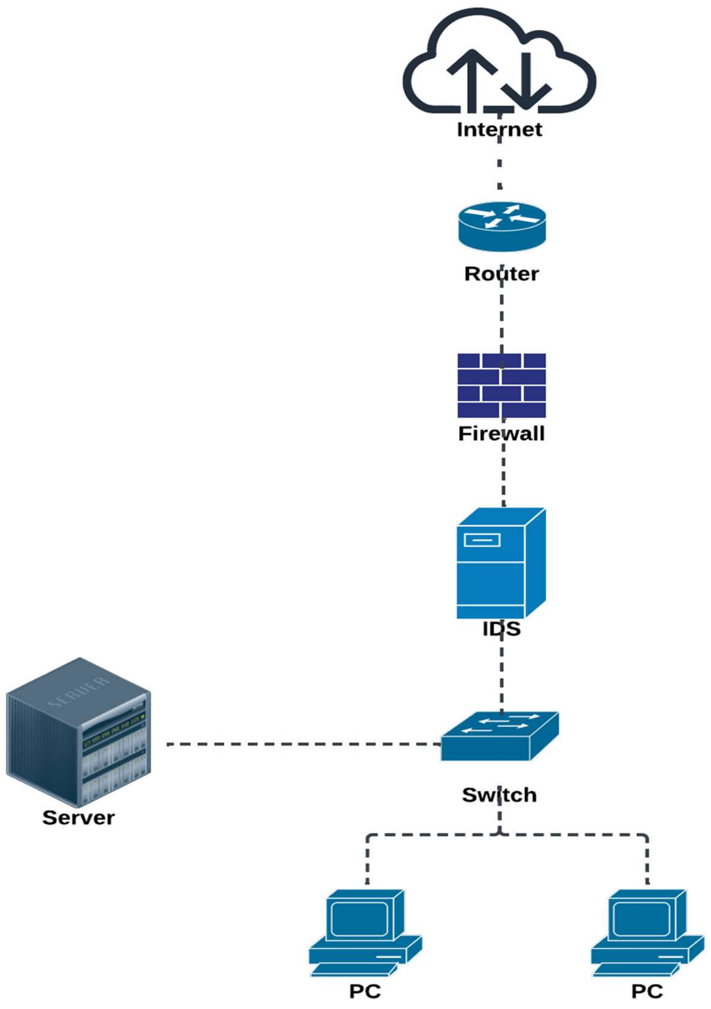


Figure 4.2 *Network Diagram*

4.4.2 Use Case Diagram

This diagram illustrates the actors and the use cases of the system. It shows how the system monitors network traffic generated by both normal users and potential attackers, capturing and analyzing packets to detect intrusions. The diagram shows a logical flow from traffic monitoring through packet capture, analysis, intrusion detection, notification generation, and logging of suspicious activities. The Network Administrator is connected to all these processes, indicating their overarching role in managing the system. The diagram effectively demonstrates how the NIDS operates, from initial traffic observation to final alert logging, while highlighting the relationships between different system components and external actors.

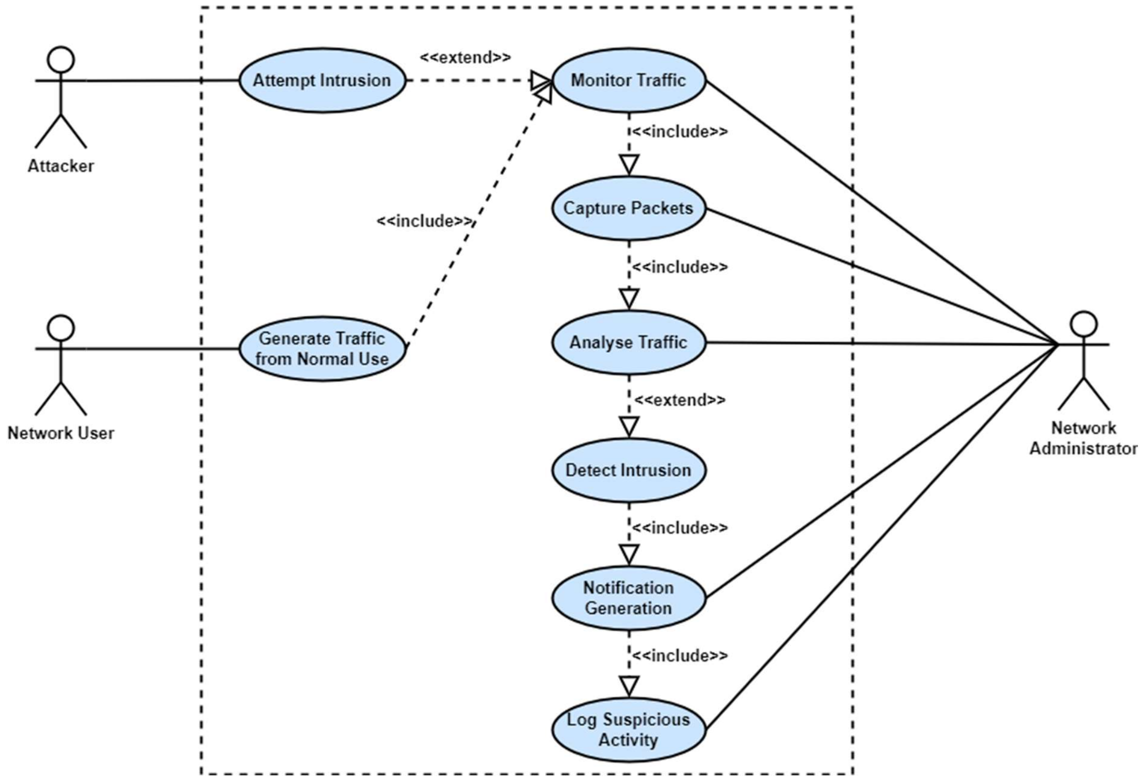


Figure 4.3 *Use Case Diagram*

4.4.3 Sequence Diagram

This sequence diagram illustrates the flow of network traffic through the system. The process begins when an Attacker attempts an intrusion, which is intercepted by the Hybrid IDS. The system then simultaneously forwards the traffic to both its Signature-based and Anomaly-based components for analysis. The Signature-based IDS compares the traffic against known signatures and rules, while the Anomaly-based IDS analyses it for unusual patterns. Both components return their results back to the Hybrid IDS. If either component detects suspicious activity, the Hybrid IDS generates an alert and sends it to the Network Administrator. The system also creates logs of the activity, which are also sent to the Administrator for analysis. Based on this analysis, a rule re-evaluation process may be initiated.

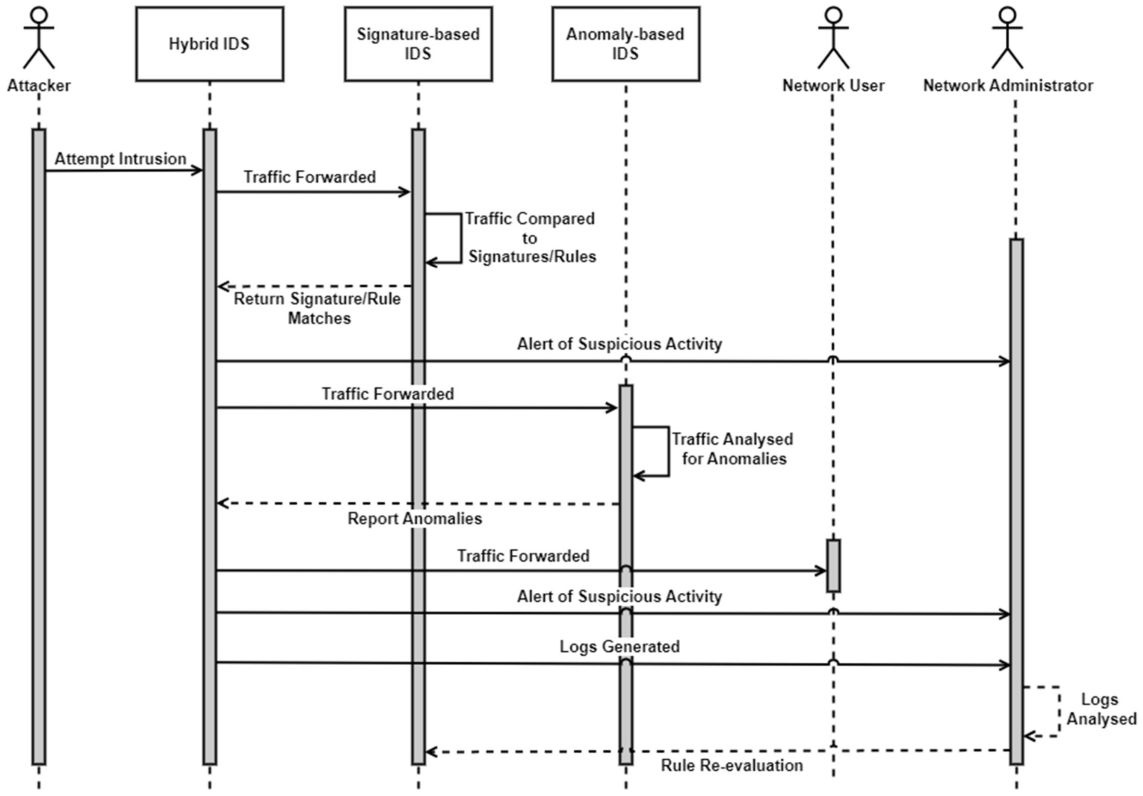


Figure 4.4 *Sequence Diagram*

4.4.4 Flowchart

This flowchart outlines the process of the hybrid intrusion detection system which begins when network traffic enters the system. The system captures all incoming traffic, processes it, and initially checks for known attack patterns using a Signature-based Intrusion Detection System (SIDS). If a match is found with known signatures, the system generates an alert to notify the network administrator. If no match is detected, the system proceeds to the next step, where an Anomaly-based Intrusion Detection System (AIDS) analyzes the traffic for irregular behavior.

If an anomaly is detected, the NIDS generates an alert to notify network administrators, and logs are created to document the suspicious activity for further analysis. If no anomaly is detected, the system loops back to continue monitoring. The system operates continuously, ensuring that network traffic is always being monitored and analyzed for any potential threats.

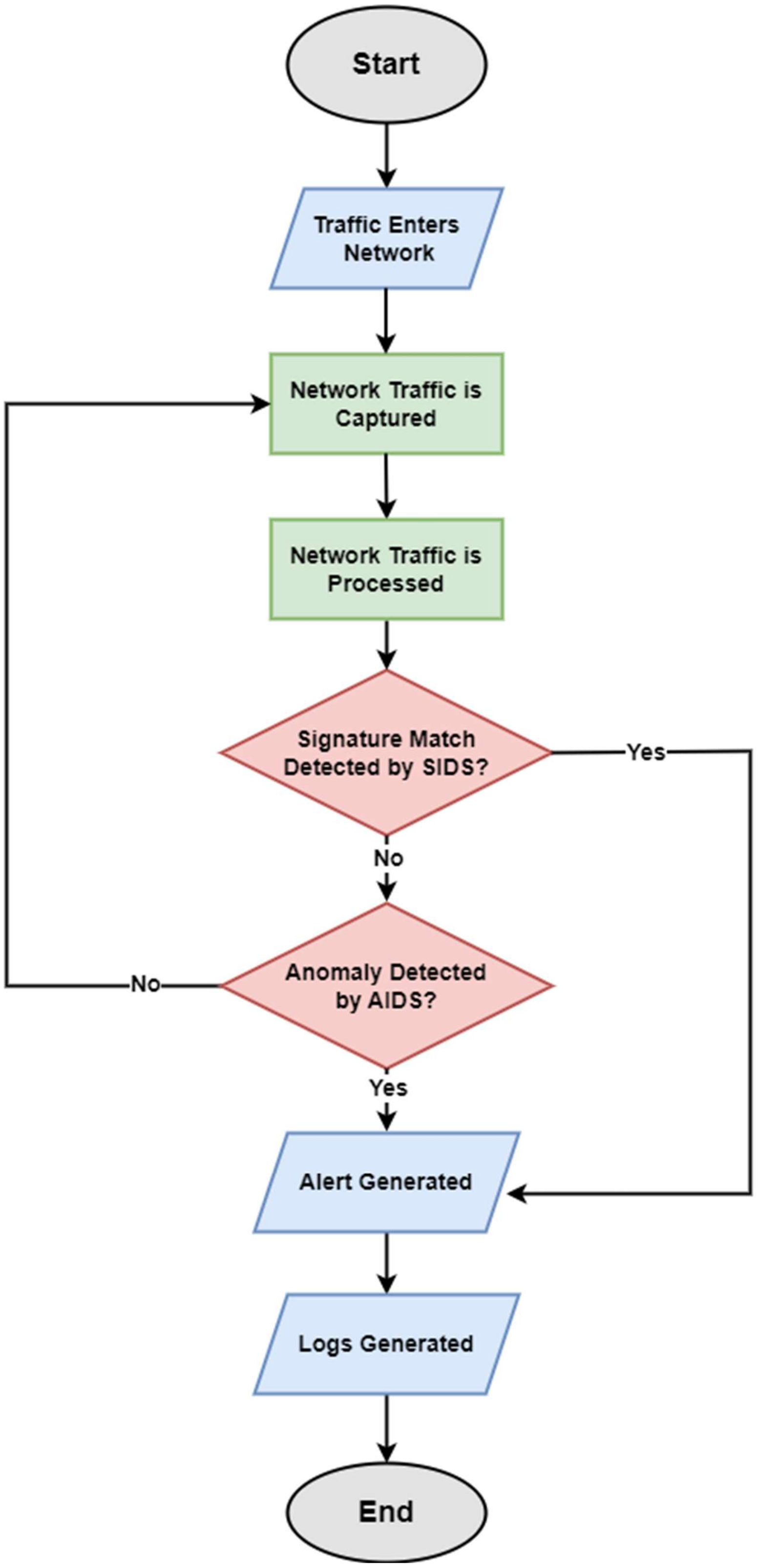

Start
Traffic Enters Network
Network Traffic is Captured
Network Traffic is Processed
Signature Match Detected by SIDS?
Yes
No
Anomaly Detected by AIDS?
No
Yes
Alert Generated
Logs Generated
End

Figure 4.5 *Flowchart*

4.4.5 State Transition Diagram

This state transition diagram illustrates the lifecycle of the traffic processing in the hybrid intrusion detection system. Initially, the system remains in an idle state when there is no network traffic. As soon as traffic enters the network, the system shifts into monitoring mode, where it continuously observes the flow of data for any irregularities. If no suspicious activity is detected, the system continues monitoring the incoming traffic, maintaining an ongoing surveillance of the network.

When suspicious activity is identified, the system transitions from the traffic analysis state to the alert generation state, where it generates an alert to notify of the intrusion. Simultaneously, the NIDS logs the details of the suspicious activity, ensuring a comprehensive record is kept.

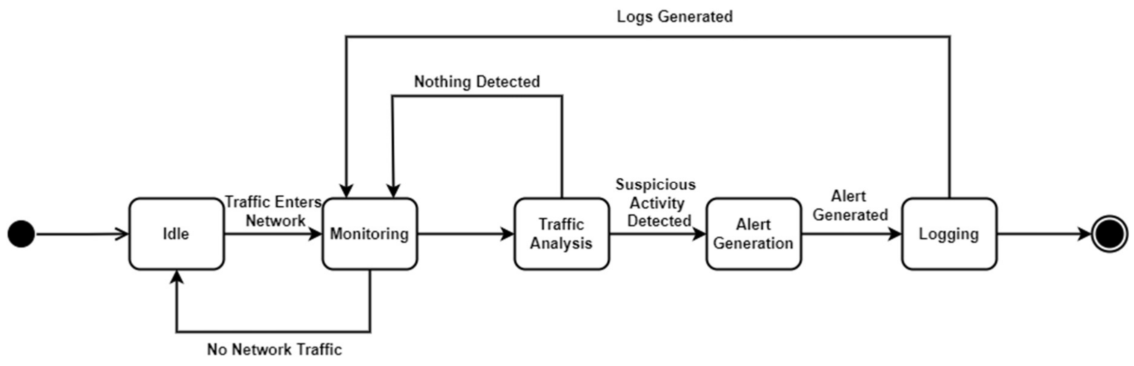


Figure 4.6 *State Transition Diagram*

Chapter 5: Implementation and Testing

5.1 Introduction

This chapter outlines the implementation and testing of the Intrusion Detection System (IDS), covering the hardware and software specifications, the testing environment with VirtualBox and Kali Linux, and tools for simulating attacks. The CIC-IDS2017 dataset which was used for training and testing the model is also discussed, along with the testing of both functional and non-functional requirements.

5.2 System Specifications

5.2.1 Hardware Specifications

The project was implemented on a laptop with the following specifications:

Table 5.1 *Table of Hardware specifications*

Manufacturer	Lenovo
Model	81WD
Memory (RAM)	8 GB
Storage	475 GB
Processor	Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz
Graphics adapter	Intel UHD Graphics G1 (Ice Lake 32 EU)
Networking	802.11 a/b/g/n/ac

5.2.2 Software Specifications

The software used in the project included a range of tools and programming languages. The software specifications are as follows:

1. Windows 11 Pro (64 bit): This is the OS running on the development machine. Windows 10 or later could work with the development tools used.

2. Python 3.8: This is the programming language used for developing the system. It was chosen for its stability and compatibility with the machine learning libraries used.

3. Visual Studio Code (VSCode): This is the IDE was used for the main development of the intrusion detection system code. To be specific, VSCode version 1.95 was used.

4. Google Colab and Jupyter Notebook: These notebook environments were used specifically for developing and training the XGBoost model. These platforms were chosen for their ease of use..

5. Microsoft Excel: Used to view and analyze the CSV dataset files. This tool was used for initial data inspection and validation.

6. Notepad: This was used for editing and maintaining IDS rules in plain text format. Windows built-in Notepad was sufficient for this purpose as the rules were maintained in simple text files.

7. VirtualBox: VirtualBox is an open-source virtualization software developed by Oracle. In this project, VirtualBox, specifically version 7.0.10 is used to host the virtual environment used for testing the intrusion detection system (IDS).

8. Kali Linux: In this project, Kali Linux is deployed as a virtual machine (VM) within VirtualBox to simulate cyber-attacks on the IDS. This environment is used to generate malicious traffic, test detection rules, and analyze the system's response to attack scenarios.

9. hping: This is a command-line packet crafting tool used to generate and manipulate network packets for testing and simulation purposes. In this scenario, it is used in the

Kali VM to create custom network packets that mimic malicious traffic, such as SYN floods or malformed packets, to test the IDS's ability to detect anomalies and attacks.

10. arpspoof: This is tool used to perform ARP (Address Resolution Protocol) spoofing attacks. It redirects network traffic by poisoning the ARP cache of target devices. It is used in this project to generate realistic malicious network activity to test the IDS's effectiveness in detecting such threats.

5.3 Description of the Dataset

The CIC-IDS2017 dataset, developed by the Canadian Institute for Cybersecurity at the University of New Brunswick (https://www.unb.ca/cic/datasets/ids-2017.html), is a comprehensive network traffic dataset around 51.1 GB in size, collected over five days. The data is distributed across the days as follows: Monday (11.0 GB), Tuesday (11.0 GB), Wednesday (13.0 GB), Thursday (7.8 GB), and Friday (8.3 GB).

The dataset is available in two primary formats: raw network captures (PCAP files), and processed CSV files that include over 80 extracted network flow features. Monday's data serves as a baseline, containing only benign traffic, while Tuesday through Friday contain a mix of normal activity and various attacks including Brute Force attacks, DoS/DDoS, Heartbleed, Web Attacks (SQL injection, XSS), Infiltration attempts, Botnet activity, and Port Scanning.

What sets this dataset apart is its realistic representation of network traffic, generated using a B-Profile system to simulate 25 users' behavior across common protocols (HTTP, HTTPS, FTP, SSH, and email). The network architecture included diverse operating systems (Windows, Ubuntu, and MacOS) and complete network infrastructure components (modems, firewalls, switches, and routers). Each activity in the dataset is precisely labeled and timestamped, making it particularly valuable for training and evaluating machine learning-based intrusion detection systems.

5.4 System Testing

5.4.1 Testing of Functional Requirements

Table 5.2 *Table of Functional Requirements Testing*

Test Case	**Description**	**Test Data**	**Result**	**Verdict**
1	The system should capture and analyze real-time network traffic to detect potential intrusions or abnormal activity	Live traffic	The system could capture and analyze real-time network traffic (Refer to Appendix B)	Pass
2	The system should identify known threats using predefined attack rules and match traffic against these rules	Simulated traffic	The system could identify known threats using predefined attack rules	Pass
3	The system should analyze traffic patterns to detect deviations from normal behavior, potentially identifying unknown threats.	Live and simulated traffic	The system could detect deviations from normal behaviour after analysing traffic patterns	Pass
4	The system should generate and log notifications when suspicious or malicious traffic is detected, providing detailed information about the event.	Simulated traffic	Logs and alerts could be generated, providing details about the event	Pass

5	The system should allow the creation, modification, and deletion of detection rules, enabling administrators to adapt to new threats	Live and simulated traffic	The system allowed the creation, modification, and deletion of detection rules	Pass
6	The system should log detected incidents that summarize network activity, threats, and performance	Live and simulated traffic	The system logs detected incidents	Pass

5.4.2 Testing of Non-Functional Requirements

Table 5.3 *Table of Non-Functional Requirements Testing*

Test Case	Description	Test Data	Result	Verdict
1	The system should be highly reliable and capable of continuous operation without crashing or losing data	Live traffic over a long period of time	The system is capable of continuous operation without crashing or losing data	Pass
2	The system should be scalable, allowing it to handle increased traffic or larger network environments without degradation in performance	Large amounts of simulated traffic	The system is scalable	Pass

3	The system codebase should be well-documented and modular, making it easier for developers to fix bugs, implement updates, or add new features.	Live traffic	The system codebase is modular, making it easier for developers to potentially fix bugs, implement updates, or add new features.	Pass
4	The system should optimize resource consumption, minimizing CPU, memory, and disk usage to reduce the impact on network performance.	Live and simulated traffic	The system has optimized resource consumption, minimizing CPU, memory, and disk usage.	Pass
5	The system should be accurate, having few false negatives and false positives.	Live and simulated traffic	The system is accurate, having few false negatives and false positives.	Pass

5.5 Discussion of Results

The implementation and testing of the Intrusion Detection System (IDS) produced promising results, demonstrating how it can detect threats. The system successfully fulfilled all functional requirements during testing, as seen in Table 5.2. It was able to capture and analyze real-time network traffic, accurately identifying known threats through predefined attack rules. Furthermore, it demonstrated the ability to detect deviations from normal traffic behavior, indicating its potential effectiveness in identifying both known and unknown threats.

Notifications and logs were generated for each suspicious or malicious event, providing detailed insights into the nature of the threat, which is critical for network administrators to take timely action. The system also met non-functional requirements, as shown in Table 5.3.

The accuracy of the system was also notable, with a low incidence of false positives and false negatives, reflecting the effectiveness of the integrated XGBoost model.

The testing results show positive overall performance of the IDS. The system successfully demonstrated its core capabilities on limited hardware specifications (8GB RAM, Intel i5-1035G1). However, there's room for improvement in several areas. To enhance the system's capabilities, several improvements could be considered, running the IDS on a PC with more RAM to handle increased traffic volume is one, implementing GPU acceleration for faster machine learning model inference is another, and using a stronger testing framework would be another.

Chapter 6: Conclusions and Recommendations

6.1 Conclusions

The project successfully developed a hybrid Intrusion Detection System (IDS) capable of detecting a range of cyber-attacks by analyzing network traffic. The system met all functional requirements, including detecting both known and unknown threats, generating alerts, and efficiently logging events. It also satisfied non-functional requirements like reliability, scalability, and resource optimization. However, the system's performance could be improved with more powerful hardware and a more balanced dataset to reduce false positives/negatives.

6.2 Recommendations

To improve the IDS developed in this project, the following recommendations can be considered:

i. Hardware upgrades: The system could be implemented and tested on more powerful hardware (e.g., more RAM and GPU acceleration).

ii. More data: Incorporate more real-world data, especially with newer attack types, to improve the system's robustness.

iii. Expert involvement: Engage security experts and network administrators in testing for better rule refinement and system optimization.

iv. Testing: Implement a more robust testing framework, including dynamic network environments.

6.3 Future Works

Several future directions can be explored to build on the success of this project:

i. Integration with SIEM: The IDS could potentially be integrated with Security Information and Event Management (SIEM) systems for centralized security management.

ii. Use of adaptive ML models: Models could be trained to update the detection rules based on new data to handle emerging threats instead of it being done manually.

iii. Cross-platform security: Extend the IDS to monitor other environments, such as cloud and IoT networks.

iv. Usability enhancements: Improve the user interface for better visualization and understanding of system performance.

References

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2020). 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

Ahmim, A., Derdour, M., & Ferrag, M. A. (2018). An intrusion detection system based on combining probability predictions of a tree of classifiers. *International Journal of Communication Systems*, *31*(9), e3547. https://doi.org/10.1002/dac.3547

Akalanka Mailewa & Suman Thapa. (2020). *The role of intrusion detection/prevention systems in modern computer networks: A review. 53.*

Aldwairi, M., Abu-Dalo, A. M., & Jarrah, M. (2017). Pattern matching of signature-based IDS using Myers algorithm under MapReduce framework. *EURASIP Journal on Information Security*, *2017*(1), 9. https://doi.org/10.1186/s13635-017-0062-7

Aldweesh, A., Derhab, A., & Emam, A. Z. (2020). Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowledge-Based Systems*, *189*, 105124. https://doi.org/10.1016/j.knosys.2019.105124

Aljawarneh, S., Aldwairi, M., & Yassein, M. B. (2018). Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *Journal of Computational Science*, *25*, 152–160. https://doi.org/10.1016/j.jocs.2017.03.006

Alsughayyir, B., Qamar, A. M., & Khan, R. (2019). Developing a Network Attack Detection System Using Deep Learning. *2019 International Conference on Computer and Information Sciences (ICCIS)*, 1–5. https://doi.org/10.1109/ICCISci.2019.8716389

Aminanto, M. E., & Kim, K. (2016). *Deep Learning in Intrusion Detection System: An Overview*. 2016 International Research Conference on Engineering and Technology (2016 IRCET).

Anderson, J. P. (1980). Computer Security Threat Monitoring and Surveillance. *James P Anderson Co*. https://csrc.nist.gov/files/pubs/conference/1998/10/08/proceedings-of-the-21st-nissc-1998/final/docs/early-cs-papers/ande80.pdf

Bace, R., & Mell, P. (2001). Intrusion Detection Systems. *NIST*. https://www.nist.gov/publications/intrusion-detection-systems

Bayerl, P. S., Karlović, R., Akhgar, B., & Markarian, G. (Eds.). (2017). *Community Policing - A European Perspective: Strategies, Best Practices and Guidelines* (1st ed. 2017 edition). Springer.

Buczak, A. L., & Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys & Tutorials*, *18*(2), 1153–1176. https://doi.org/10.1109/COMST.2015.2494502

Burgio, D. A. (2019). *Reduction of False Positives in Intrusion Detection Based on Extreme Learning Machine with Situation Awareness* [Nova Southeastern University]. https://nsuworks.nova.edu/gscis_etd/1093

Cruz-Cunha, M. M., & Mateus-Coelho, N. (2020). Handbook of Research on Cyber Crime and Information Privacy (2 Volumes). In *Https://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-7998-5728-0*. IGI Global. https://www.igi-global.com/book/handbook-research-cyber-crime-information/www.igi-global.com/book/handbook-research-cyber-crime-information/251347

Dina, A. S., & Manivannan, D. (2021). Intrusion detection based on Machine Learning techniques in computer networks. *Internet of Things*, *16*, 100462. https://doi.org/10.1016/j.iot.2021.100462

Folorunso, O., Ayo, F., & Babalola, Y. (2016). Ca-NIDS: A network intrusion detection system using combinatorial algorithm approach. *Journal of Information Privacy and Security*, *12*, 181–196. https://doi.org/10.1080/15536548.2016.1257680

Fraley, J. B., & Cannady, J. (2017). The promise of machine learning in cybersecurity. *SoutheastCon 2017*, 1–6. https://doi.org/10.1109/SECON.2017.7925283

Gunduz, M. Z., & Das, R. (2020). Cyber-security on smart grid: Threats and potential solutions. *Computer Networks*, *169*, 107094. https://doi.org/10.1016/j.comnet.2019.107094

Guo, C., Ping, Y., Liu, N., & Luo, S.-S. (2016). A two-level hybrid approach for intrusion detection. *Neurocomputing*, *214*, 391–400. https://doi.org/10.1016/j.neucom.2016.06.021

Harish, B. S., & Kumar, S. V. A. (2017). *Anomaly based Intrusion Detection using Modified Fuzzy Clustering*. https://doi.org/10.9781/ijimai.2017.05.002

IBM. (2023). *Cost of a Data Breach Report 2023* (18).

Ioulianou, P., Vassilakis, V., & Moscholios, I. (2018, July 13). *A Signature-based Intrusion Detection System for the Internet of Things*.

Jie Li, Yanpeng Qu, Fei Chao, Hubert P. H. Shum, Edmond S. L. Ho, & Longzhi Yang. (2018). *Machine Learning Algorithms for Network Intrusion Detection. In: Sikos, L. (eds) AI in Cybersecurity* (Vol. 151). Springer, Cham. https://link.springer.com/book/10.1007/978-3-319-98842-9

Kok, S., Abdullah, A., Jhanjhi, N., & Supramaniam, M. (2019). A review of intrusion detection system using machine learning approach. *International Journal of Engineering Research and Technology*, *12*, 8–15.

Liao, H.-J., Richard Lin, C.-H., Lin, Y.-C., & Tung, K.-Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, *36*(1), 16–24. https://doi.org/10.1016/j.jnca.2012.09.004

Ma, W. (2020). Analysis of anomaly detection method for Internet of things based on deep learning. *Transactions on Emerging Telecommunications Technologies*, *31*(12), e3893. https://doi.org/10.1002/ett.3893

Malek, Z. S., Trivedi, B., & Shah, A. (2020). User behavior Pattern -Signature based Intrusion Detection. *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, 549–552. https://doi.org/10.1109/WorldS450073.2020.9210368

Muneer, S., Farooq, U., Athar, A., Ahsan Raza, M., Ghazal, T. M., & Sakib, S. (2024). A Critical Review of Artificial Intelligence Based Approaches in Intrusion Detection: A Comprehensive Analysis. *Journal of Engineering*, *2024*(1), 3909173. https://doi.org/10.1155/2024/3909173

Musa, U. S., Chhabra, M., Ali, A., & Kaur, M. (2020). Intrusion Detection System using Machine Learning Techniques: A Review. *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, 149–155. https://doi.org/10.1109/ICOSEC49089.2020.9215333

Ni, M. (2023). A review on machine learning methods for intrusion detection system. *Applied and Computational Engineering*, *27*, 57–64. https://doi.org/10.54254/2755-2721/27/20230148

NTSC. (2020). *Cyber Security Report 2020*.

Otoum, Y., & Nayak, A. (2021). AS-IDS: Anomaly and Signature Based IDS for the Internet of Things. *Journal of Network and Systems Management*, *29*(3), 23. https://doi.org/10.1007/s10922-021-09589-6

Ozkan-Okay, M., Samet, R., Aslan, Ö., & Gupta, D. (2021). A Comprehensive Systematic Literature Review on Intrusion Detection Systems. *IEEE Access*, *9*, 157727–157760. https://doi.org/10.1109/ACCESS.2021.3129336

Piazza, A. (2024, February 28). Enterprise security is facing an identity crisis: Findings from the latest X-Force Threat Intelligence Index. *IBM Blog*. https://www.ibm.com/blog/announcement/enterprise-security-identity-crisis-x-force-threat-intelligence-index

Prasad, R., & Rohokale, V. (2019). Artificial Intelligence and Machine Learning in Cyber Security. In R. Prasad & V. Rohokale (Eds.), *Cyber Security: The Lifeline of Information and Communication Technology* (pp. 231–247). Springer International Publishing. https://doi.org/10.1007/978-3-030-31703-4_16

Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho, A. (2019). A survey of network-based intrusion detection data sets. *Computers & Security*, *86*, 147–167. https://doi.org/10.1016/j.cose.2019.06.005

Roy, S. S., Mallik, A., Gulati, R., Obaidat, M. S., & Krishna, P. V. (2017). A Deep Learning Based Artificial Neural Network Approach for Intrusion Detection. In D. Giri, R. N. Mohapatra, H. Begehr, & M. S. Obaidat (Eds.), *Mathematics and Computing* (pp. 44–53). Springer. https://doi.org/10.1007/978-981-10-4642-1_5

Sausalito, C. (2020, 13). Cybercrime To Cost The World $10.5 Trillion Annually By 2025. *Cybercrime To Cost The World $10.5 Trillion Annually By 2025*. https://cybersecurityventures.com/cybercrime-damage-costs-10-trillion-by-2025/

Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A Deep Learning Approach to Network Intrusion Detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *2*(1), 41–50. https://doi.org/10.1109/TETCI.2017.2772792

Symantec. (2017, April). *Internet Security Threat Report, Volume 22*. https://docs.broadcom.com/docs/istr-22-2017-en

Talukder, Md. A., Hasan, K. F., Islam, Md. M., Uddin, Md. 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

Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A. R., & Ghogho, M. (2016). Deep learning approach for Network Intrusion Detection in Software Defined Networking. *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 258–263. https://doi.org/10.1109/WINCOM.2016.7777224

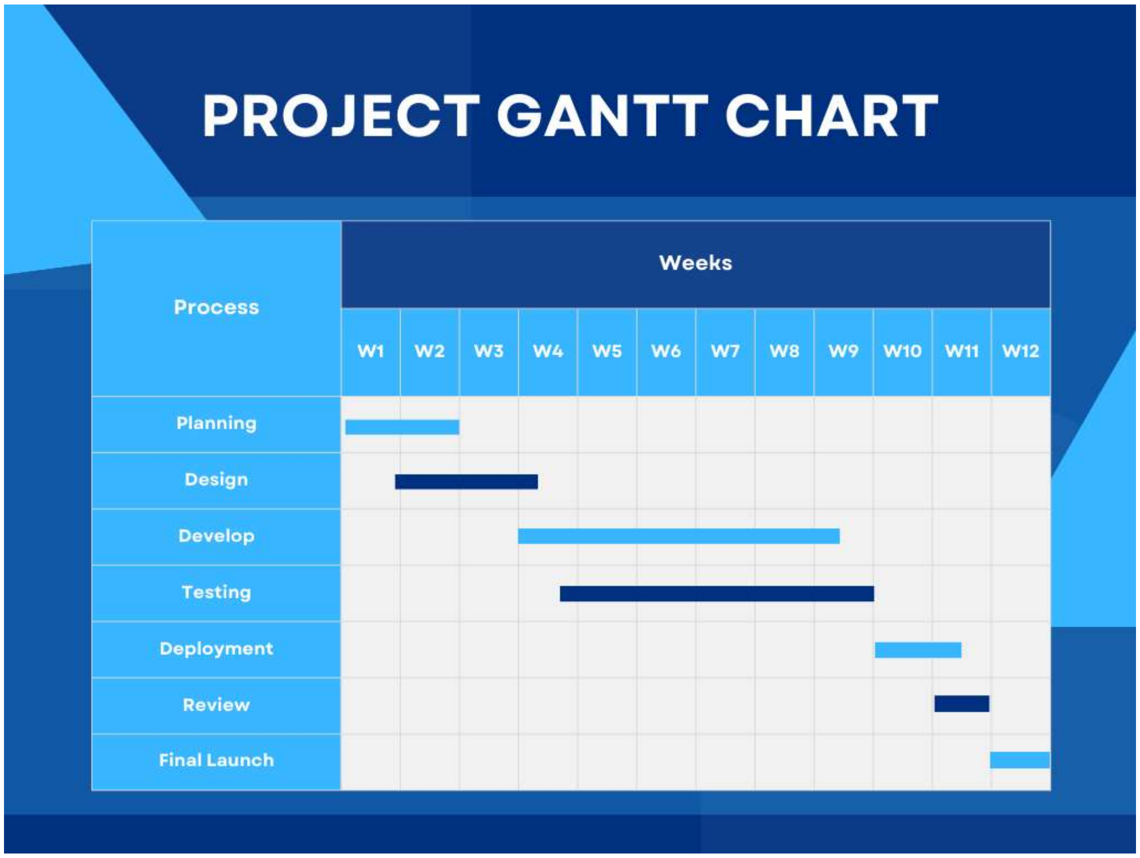
Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017). Applying convolutional neural network for network intrusion detection. *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 1222–1228. https://doi.org/10.1109/ICACCI.2017.8126009

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

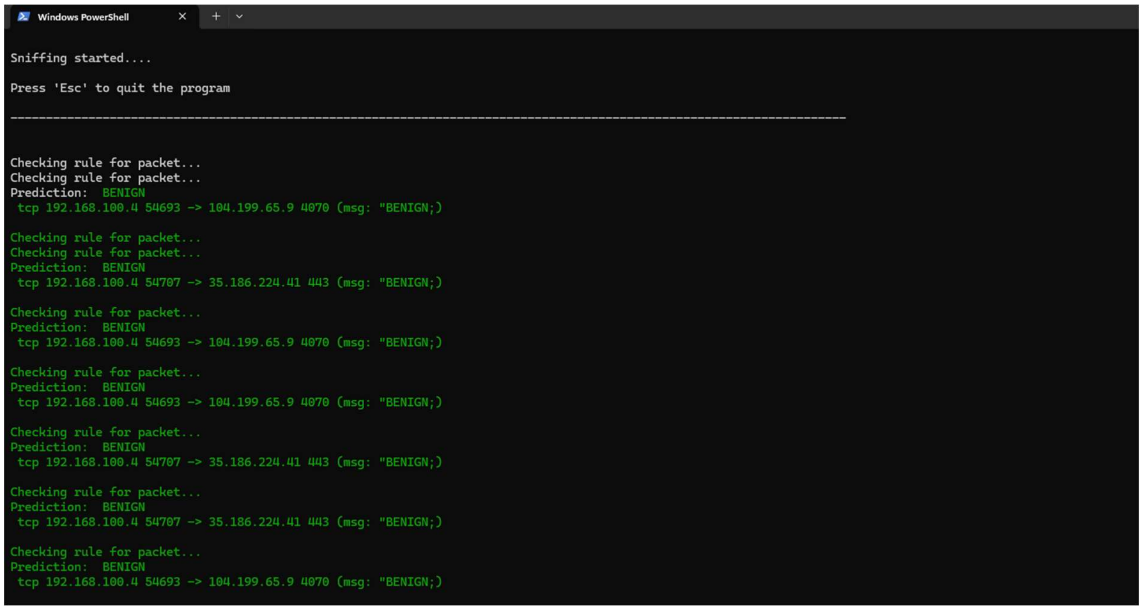
Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access*, *5*, 21954–21961. https://doi.org/10.1109/ACCESS.2017.2762418

Appendices

Appendix A: Gantt Chart



Appendix B


Windows PowerShell

```
Sniffing started....

Press 'Esc' to quit the program

------------------------------------------------------------------------------------------------------------------------


Checking rule for packet...
Checking rule for packet...
Prediction:  BENIGN
 tcp 192.168.100.4 54693 -> 104.199.65.9 4070 (msg: "BENIGN;)

Checking rule for packet...
Checking rule for packet...
Prediction:  BENIGN
 tcp 192.168.100.4 54707 -> 35.186.224.41 443 (msg: "BENIGN;)

Checking rule for packet...
Prediction:  BENIGN
 tcp 192.168.100.4 54693 -> 104.199.65.9 4070 (msg: "BENIGN;)

Checking rule for packet...
Prediction:  BENIGN
 tcp 192.168.100.4 54693 -> 104.199.65.9 4070 (msg: "BENIGN;)

Checking rule for packet...
Prediction:  BENIGN
 tcp 192.168.100.4 54707 -> 35.186.224.41 443 (msg: "BENIGN;)

Checking rule for packet...
Prediction:  BENIGN
 tcp 192.168.100.4 54707 -> 35.186.224.41 443 (msg: "BENIGN;)

Checking rule for packet...
Prediction:  BENIGN
 tcp 192.168.100.4 54693 -> 104.199.65.9 4070 (msg: "BENIGN;)
```