DEVELOPMENT OF AN AI MODEL FOR INTRUSION DETECTION

BY

#### AFOLABI BILAAL AYOMIDE

MATRIC NUMBER: 210591009

SUBMITTED TO

THE DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF SCIENCE

LAGOS STATE UNIVERSITY (LASU)

IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF BACHELOR’S DEGREE (B.SC) IN COMPUTER SCIENCE

SUPERVISED BY

PROF. BENJAMIN AREBISALA

# **CERTIFICATION**

This is to certify that this research work was carried out by AFOLABI BILAAL AYOMIDE, matriculation number 210591009, Department of Computer Science, under close guidance and supervision.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NAME OF STUDENT SIGNATURE AND DATE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NAME OF SUPERVISOR SIGNATURE AND DATE

# **DEDICATION**

Dedication is the driving force behind the success of any great project, and it is what makes this work truly meaningful. I dedicate this project to those who have inspired me along the way. To the trailblazers who have come before me and paved the way for success, I honor their contributions. To my friends and family, who have motivated and stood by me throughout this journey, I could not have done it without your unwavering support. To my lecturers and mentors, who have guided and challenged me to be the best I can be, I am truly grateful for your wisdom and encouragement. Lastly, I dedicate this project to the future— to the generations of innovators and creators who will follow. May you build upon this work, push the boundaries of possibility, and make the world a better place through your dedication and hard work.

# **ACKNOWLELDGEMENT**

I would like to express my heartfelt gratitude to the creators of e-learning platforms and software. Without their innovative ideas and tireless efforts, the world of education would not have experienced the phenomenal growth it has today. Their contributions have opened up new avenues for students worldwide, providing access to quality education regardless of location or financial status. I also wish to acknowledge the dedication of lecturers and teachers who have embraced e learning as part of their teaching methodology. Their remarkable flexibility and adaptability in transitioning from traditional classrooms to virtual learning environments have been truly inspiring. Their unwavering commitment to delivering quality education has left an indelible mark on students' lives. To my colleagues, family, and friends, I extend my deepest thanks for your constant encouragement throughout this journey. Your support during challenging moments and your shared joy in my successes have been invaluable. Finally, I give all thanks to Almighty God, the Alpha and Omega, my source of hope, strength, and inspiration. His guidance and presence have made this research a successful accomplishment.

# **ABSTRACT**

The increasing sophistication of cyber threats necessitates advanced security measures to protect critical systems and networks. This project focuses on designing and implementing an AI-driven Intrusion Detection System (IDS) to enhance cybersecurity by identifying malicious activities in real time. The system aims to improve threat detection accuracy by leveraging machine learning algorithms to analyze network traffic and classify anomalies indicative of cyber attacks. By integrating a structured approach to feature selection and classification, the system ensures objective and consistent detection of security breaches, minimizing false positives and improving response efficiency. Key components include dataset preprocessing, feature extraction, model training, and evaluation using established performance metrics. The system utilizes publicly available benchmark datasets and advanced classification techniques to detect a wide range of intrusion attempts effectively. This project highlights the significance of AI-driven security solutions in modern network defense and demonstrates how automated detection mechanisms can contribute to robust cybersecurity frameworks. Ultimately, the goal is to enhance threat detection capabilities, reduce response time, and support proactive security measures in enterprise and critical infrastructure environments.

Table of Contents

[**CERTIFICATION** 2](#_Toc197854176)

[**DEDICATION** 3](#_Toc197854177)

[**ACKNOWLELDGEMENT** 4](#_Toc197854178)

[**ABSTRACT** 5](#_Toc197854179)

[**1.1 BACKGROUND OF THE STUDY** 9](#_Toc197854180)

[**1.2 STATEMENT OF THE PROBLEM** 10](#_Toc197854181)

[**1.3 AIM AND OBJECTIVES** 11](#_Toc197854182)

[**1.4 SIGNIFICANCE OF STUDY** 12](#_Toc197854183)

[**1.5 SCOPE OF STUDY** 13](#_Toc197854184)

[**1.6 GENERAL LIMITATIONS OF AN AI MODEL FOR INTRUSION DETECTION** 13](#_Toc197854185)

[**1.7 DEFINITION OF TERMS** 14](#_Toc197854186)

[**2.1 INTRODUCTION TO INTRUSION DETECTION SYSTEM (IDS)** 16](#_Toc197854187)

[**2.2 ARTIFICIAL INTELLIGENCE IN IDS** 19](#_Toc197854188)

[**2.2.1 MACHINE LEARNING IN IDS** 19](#_Toc197854189)

[**2.2.2 DEEP LEARNING IN IDS** 20](#_Toc197854190)

[**2.2.3 NATURAL LANGUAGE PROCESSING (NLP) IN IDS** 20](#_Toc197854191)

[**2.3 CHALLENGES OF IDS DELOYMENT** 21](#_Toc197854192)

[**2.3.1 CHALLENGES OF INTRUSION DETECTION SYSTEMS (IDS) FOR INDUSTRIAL CONTROL SYSTEMS (ICSS)** 21](#_Toc197854193)

[**2.3.2 CHALLENGE OF IDS ON INTRUSION EVASION DETECTION** 22](#_Toc197854194)

[**2.4 RELATED WORKS** 22](#_Toc197854195)

[**2.5 DATASET FOR IDS DEVELOPMENT** 23](#_Toc197854196)

[**2.5.1 EXISTING DATASETS USED FOR DEVELOPING AND EVALUATING IDS** 24](#_Toc197854197)

[**2.5.1.1 DARPA / KDD CUP99** 24](#_Toc197854198)

[**2.5.1.2 CAIDA** 25](#_Toc197854199)

[**2.5.1.3 NSL-KDD** 25](#_Toc197854200)

[**2.5.1.4 CICIDS 2017** 25](#_Toc197854201)

[**2.5.2 FEATURE SELECTION FOR IDS** 26](#_Toc197854202)

[**2.6 EVALUATION METRICS IN IDS** 27](#_Toc197854203)

[**2.6.1 ACCURACY** 27](#_Toc197854204)

[**2.6.2 PRECISION** 27](#_Toc197854205)

[**2.6.3 RECALL** 28](#_Toc197854206)

[**2.6.4 F1-SCORE** 28](#_Toc197854207)

[**2.6.5 ROC CURVE AND AUC** 28](#_Toc197854208)

[**2.6.6 CROSS-VALIDATION** 29](#_Toc197854209)

[**2.6.7 LEARNING CURVES** 29](#_Toc197854210)

[**2.6.8 MODEL TRANSPARENCY TECHNIQUES** 29](#_Toc197854211)

[**2.7 SECURITY AND PRIVACY CONCERNS IN IDS** 30](#_Toc197854212)

[**2.7.1 CHALLENGE OF IDS ON INTRUSION EVASION DETECTION** 30](#_Toc197854213)

[**3.1 SYSTEM ANALYSIS** 31](#_Toc197854214)

[**3.1.1 SYSTEM DESCRIPTION** 31](#_Toc197854215)

[**3.1.2 HISTORICAL BACKGROUND OF THE CASE STUDY** 32](#_Toc197854216)

[**3.1.3 EVALUATION OF THE CURRENT SYSTEM** 32](#_Toc197854217)

[**3.1.4 BENEFITS OF THE CURRENT SYSTEM** 33](#_Toc197854218)

[**3.1.4.1 EARLY DETECTION OF KNOWN ATTACKS** 33](#_Toc197854219)

[**3.1.4.2 LOW FALSE ALARM RATES FOR KNOWN PATTERNS** 33](#_Toc197854220)

[**3.1.4.3 NETWORK MONITORING AND VISIBILITY** 34](#_Toc197854221)

[**3.1.4.4 POLICY ENFORCEMENT** 34](#_Toc197854222)

[**3.1.4.5 SCALABILITY IN NETWORK-BASED SYSTEM** 34](#_Toc197854223)

[**3.1.4.6 COST EFFECTIVE FOR BASIC OPERATIONS** 34](#_Toc197854224)

[**3.1.5 PROBLEMS OF THE CURRENT SYSTEM** 34](#_Toc197854225)

[**3.1.6 TECHNICAL DETAILS OF SOLUTION TO THE PROBLEM** 35](#_Toc197854226)

[**3.1.6.1 DATASET SELECTION AND PREPROCESSING** 36](#_Toc197854227)

[**3.1.6.2 FEATURE SELECTION** 36](#_Toc197854228)

[**3.1.6.3 MODEL ARCHITECTURE** 37](#_Toc197854229)

[**3.1.6.4 EXPLAINABLE AI (XAI) INTEGRATION** 37](#_Toc197854230)

[**3.1.6.5 EVALUAITON METRICS AND VALIDATION** 37](#_Toc197854231)

[**3.1.6.6 SYTEM DEPLOYMENT CONSIDERATIONS** 37](#_Toc197854232)

[**3.2 SYSTEM DESIGN** 38](#_Toc197854233)

[**3.2.1 OVERVIEW OF THE NEW SYSTEM** 38](#_Toc197854234)

[**3.2.2 OUTPUT DESIGN** 39](#_Toc197854235)

[**3.2.3 INPUT DESIGN** 39](#_Toc197854236)

[**3.2.4 DTABASE DESIGN** 40](#_Toc197854237)

[**3.2.4.1 DATABASE OBJECTIVES:** 41](#_Toc197854238)

[**3.2.4.2 DATABASE TYPE:** 41](#_Toc197854239)

[**3.2.4.3 ENTITY-RELATIONSHIP DIAGRAM (ERD)** 42](#_Toc197854240)

[**3.2.4.4** **SAMPLE DATASET RECORD (NSL-KDD)** 42](#_Toc197854241)

[**3.2.4.5 SECURITY MEASURES** 43](#_Toc197854242)

[**3.2.5 PROCESS DESIGN** 43](#_Toc197854243)

**CHAPTER ONE**

# **1.1 BACKGROUND OF THE STUDY**

The rapid advancement of digital technologies has significantly reshaped modern computing and communication systems. However, this transformation has also led to a surge in cyber threats, including malware attacks, data breaches, and sophisticated intrusion attempts targeting critical infrastructures. Organizations, enterprises, and government institutions face mounting challenges in safeguarding their networks against cyber adversaries who continually develop and refine attack techniques (Thapa & Mailewa, 2020).

An Intrusion Detection System (IDS) is a fundamental security mechanism designed to monitor network traffic and identify suspicious activities that could compromise network security. When a potential threat or malicious activity is detected, the IDS alerts the network administrator, enabling timely intervention. The primary role of an IDS is to detect and report intrusion attempts to the relevant security personnel, thereby preventing unauthorized access and mitigating cyberattacks (Sharafaldin et al., 2018).

IDS solutions employ various tools and techniques to analyze network behavior at both host and network levels. Based on their deployment, IDS can be classified into two primary categories: **Host-Based Intrusion Detection Systems (HIDS),** which monitor individual devices for signs of compromise, and **Network-Based Intrusion Detection Systems (NIDS)**, which analyze network traffic to detect potential intrusions (Thapa & Mailewa, 2020). Despite significant research efforts aimed at improving IDS performance, real-world deployment remains a challenge due to system complexity. Effective IDS implementation requires extensive testing, evaluation, and fine-tuning. Ideally, IDS models should be trained and evaluated using labeled datasets containing a mixture of normal and malicious activities. However, due to privacy concerns and dataset limitations, researchers often rely on publicly available datasets, which may not always reflect real-world attack scenarios .

Artificial Intelligence (AI) and Machine Learning (ML) have emerged as transformative technologies in cybersecurity. Unlike traditional methods, AI and ML leverage advanced algorithms to analyze vast datasets, identify anomalies, and predict potential vulnerabilities in real time. By automating threat detection and response, these technologies enable organizations to transition from reactive to proactive security strategies, thereby enhancing their ability to detect, mitigate, and prevent cyber threats before they escalate into significant security breaches (Mirjat, 2024).

To assess IDS performance, machine learning models are typically evaluated using standard methodologies. In this study, the training and testing phases were executed using a **70-30 dataset split**, ensuring a balanced evaluation of model performance. The effectiveness of the proposed IDS model was compared against benchmarked machine learning models, including AbM1 (Ahmed et al., 2024). The evaluation criteria for IDS performance include key metrics such as **Accuracy, Precision, Recall, Cross-Validation**, and other relevant statistical measure.

The growing complexity of cyber threats necessitates continuous improvements in intrusion detection methodologies. By integrating AI-driven techniques with traditional IDS approaches, cybersecurity professionals can develop more resilient defense mechanisms, ultimately reducing the risks associated with evolving cyber attacks.

# **1.2 STATEMENT OF THE PROBLEM**

The growing complexity of cyber threats presents a major challenge to the security of modern computing systems. Organizations worldwide are continuously targeted by attackers leveraging advanced techniques to bypass traditional security mechanisms, leading to data breaches, financial losses, and operational disruptions. Intrusion Detection Systems (IDS) have emerged as a critical component of network security, designed to monitor traffic, detect anomalies, and identify potential threats in real time. However, despite their importance, existing IDS solutions face several limitations that hinder their effectiveness in real-world applications.

Intrusion Detection Systems (IDS) are critical for identifying cyber threats, but they face significant challenges in practical implementation. A major limitation is the **excessive number of alerts** generated, many of which are **false positives** triggered by normal system activity or misconfigurations rather than actual attacks. For even an average-sized network, an IDS can produce thousands of alerts daily, creating an overwhelming workload for security analysts.

The problem is further complicated by the **dynamic nature of cyber threats**. Attack techniques continuously evolve, making it difficult for IDS technologies to maintain high accuracy over time. Additionally, the **complexity of classifying malicious activity** in modern, multi-factor environments means that false alarms remain an inherent issue in intrusion detection.

As a result, security teams struggle to efficiently analyze alerts, leading to **delayed threat response, alert fatigue, and potential oversight of real attacks.** While research has explored various methods to address these challenges, the fundamental issues of**alert overload and false positives**persist, highlighting the need for more effective solutions(Sokratis K. & Georgios P., 2010). Modern IDS rely on Machine Learning (ML) for threat detection but often lack transparency, making it hard for security experts to understand their decisions. This black-box nature reduces trust and hinders effective response. While some studies have explored explainability, many lack a solid theoretical basis. To address this, SHapley Additive exPlanations (SHAP) has been proposed, offering local and global explanations to improve interpretability. This enhances trust, aids decision-making, and optimizes IDS performance for better cybersecurity (Wang et al., 2020).

# **1.3 AIM AND OBJECTIVES**

The objectives of this study are designed to provide a comprehensive understanding of AI-based Intrusion Detection Systems (IDS), with a focus on their theoretical, methodological, and practical dimensions. Specifically, this study aims to:

* Critically analyze the existing approaches to intrusion detection, highlighting their strengths and limitations in addressing modern cybersecurity threats.
* Examine the role of Artificial Intelligence (AI) and Machine Learning (ML) in enhancing IDS performance.
* Investigate the effectiveness of various AI-driven techniques, such as supervised and unsupervised learning, in detecting network anomalies and preventing cyber intrusions (Mirjat, 2024).
* Compare the proposed AI-based IDS model against traditional intrusion detection methods using performance metrics such as accuracy, precision, recall, and false positive rates.
* Explore the implications of AI-based IDS on cybersecurity strategies, including its impact on proactive threat mitigation, automated response mechanisms, and adaptability to evolving attack vectors.

By achieving these objectives, the study aims to contribute to the development of more intelligent, efficient, and reliable intrusion detection solutions.

# **1.4 SIGNIFICANCE OF STUDY**

The increasing complexity of cyber threats necessitates the development of **more efficient and interpretable Intrusion Detection Systems (IDS)** to enhance cybersecurity defense mechanisms. This study is significant for multiple reasons, as it contributes to both **theoretical advancements and practical applications** in the field of AI-driven IDS.

Traditional IDS rely on machine learning models that, while achieving high accuracy, often lack transparency, creating a “black box” effect that makes it difficult for security analysts to understand their decision-making processes. This opacity hinders trust, slows response times, and reduces the overall effectiveness of cybersecurity measures (Mohale & Obagbuwa, 2025). Addressing this issue, the study explores the integration of Explainable Artificial Intelligence (XAI) in IDS, providing interpretability and transparency that allow cybersecurity professionals to understand, trust, and optimize IDS models effectively (Mohale & Obagbuwa, 2025).

As cyber threats evolve, IDS must dynamically adapt to detect new attack patterns. While machine learning-based IDS improve detection rates, deep learning models often sacrifice interpretability, making it difficult for security analysts to understand their decisions. To address this, the study integrates Explainable AI (XAI) techniques such as SHAP, LIME, and Contrastive Explanations Method to enhance IDS transparency. By applying these methods to the NSL-KDD dataset, the research demonstrates how XAI provides insights into attack detection, aiding security teams in decision-making. The study also highlights the need for hybrid models balancing accuracy and interpretability, standardized evaluation metrics, and real-time applicability, ultimately strengthening cybersecurity defenses(Mane & Rao, 2021).

# **1.5 SCOPE OF STUDY**

This research focuses on enhancing **Intrusion Detection Systems (IDS)** through the integration of **Artificial Intelligence (AI) and Explainable AI (XAI)** techniques. The study specifically investigates how **interpretable machine learning models**, supported by XAI methods such as **SHAP (SHapley Additive exPlanations), LIME (Local Interpretable Model-agnostic Explanations), and Contrastive Explanations**, can improve the transparency and reliability of intrusion detection.

The evaluation is conducted using the **NSL-KDD dataset**, with performance metrics centered on **detection accuracy, false positive rates, and model interpretability**. While the study emphasizes**network-based intrusion detection systems**, it does not extend to **host-based IDS or endpoint security mechanisms.**

By addressing the **opacity of traditional AI-driven IDS models**, this research aims to develop more **understandable and actionable** cybersecurity solutions for real-world deployment in enterprise environments.

# **1.6 GENERAL LIMITATIONS OF AN AI MODEL FOR INTRUSION DETECTION**

Despite the contributions of this research to enhancing Intrusion Detection Systems (IDS) through Explainable AI (XAI), several limitations must be acknowledged:

* **Dataset Constraints** – The study relies on the NSL-KDD dataset, which, while widely used, does not fully represent modern cyber threats. More recent and real-world datasets could provide better validation of the proposed approach.
* **Computational Complexity** – Implementing deep learning-based IDS with explainability techniques such as SHAP and LIME requires high computational power. This may limit real-time deployment in resource-constrained environments.
* **Interpretability vs. Accuracy Trade-off** – While XAI techniques improve transparency, they may reduce the detection accuracy of IDS models, presenting a challenge in balancing interpretability and performance.
* **Scalability Issues** – The effectiveness of explainable IDS in large-scale enterprise networks remains uncertain, as this study focuses on a controlled dataset and environment. Further research is needed for deployment in real-world scenarios.
* **Generalizability** – The findings are specific to the tested dataset and models, meaning the effectiveness of the approach may vary across different network architectures and evolving attack patterns.

# **1.7 DEFINITION OF TERMS**

1. **Intrusion Detection System:** An Intrusion Detection System is a hardware or software product, which dynamically monitors the actions taken in a given system, and decides whether these actions constitute an attack or a legitimate use of the system.
2. **Explainable Artificial Intelligence (XAI):** **Explainable Artificial Intelligence (XAI)** refers to a suite of machine learning techniques designed to enhance the transparency and interpretability of AI models, allowing human users to understand, appropriately trust, and deploy AI systems effectively (Dwivedi et al., 2023).
3. **Machine Learning (ML):** Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without being explicitly programmed (Mahesh, 2020).
4. **Deep Learning (DL):** **Deep Learning** is a subset of machine learning that utilizes computational models with multiple processing layers to learn data representations at various levels of abstraction (LeCun et al., 2015).
5. **SHAP (Shapley Additive Explanations):** An XAI technique that assigns importance values to individual features in a model to explain their contribution to a prediction.
6. **LIME (Local Interpretable Model-agnostic Explanations):** A method that generates local approximations of complex ML models to improve interpretability and explain IDS decisions.
7. **Cyber Threats:** Malicious activities aimed at compromising the security of information systems, including hacking, malware, and denial-of-service attacks.
8. **NSL-KDD Dataset:** **NSL-KDD** is a publicly available dataset derived from the earlier KDD Cup 99 dataset, designed for evaluating Intrusion Detection Systems (IDS).
9. **Black Box Model:** A machine learning model that makes predictions without providing insight into its internal decision-making process, making interpretability difficult.
10. **False Positives in IDS: False positives** in Intrusion Detection Systems (IDS) refer to alerts that incorrectly indicate the presence of an attack when no actual intrusion has occurred (Hachmi et al., 2019).

**CHAPTER TWO**

**LITERATURE REVIEW**

# **2.1 INTRODUCTION TO INTRUSION DETECTION SYSTEM (IDS)**

  Intrusion is a security event, or a combination of multiple security events, that constitutes a security incident in which an intruder gains, or attempts to gain, access to a system or system resource without having authorization to do so (Shirey, R. RFC 2828, 2000)

 Intrusion detection, is the attempt to monitor and possibly prevent attempts to intrude into or otherwise compromise your system and network resources. Intrusions are caused by attackers accessing the systems from the Internet, authorized users of the systems who attempt to gain additional privileges for which they are not authorized, and authorized users who misuse the privileges given them. Intrusion Detection Systems (IDSs) are software or hardware products that automate this monitoring and analysis process (Bace, R., & Mell, P., 2001)

An Intrusion Detection System is a hardware or software product, which dynamically monitors the actions taken in a given system, and decides whether these actions constitute an attack or a legitimate use of the system. IDS collects data from current activities in a system, analyzes the data and presents it to the administrator for further action/analysis. An Intrusion Detection System aims at identifying intrusions that are caused by malicious users who attempt to gain privileges which are not authorized to them (outside intrusion) and also by authorized users who try to misuse the privileges assigned to them (inside intrusion) (Mohapatra, H, 2017)

**CATEGORIES OF INTRUSION DETECTION SYSTEM**

Intrusion detection system is classified into three categories: signature based detection systems, anomaly based detection systems, and specification based detection systems.

1. **Signature based detection systems**

Signature based detection system (also called misuse based), This type of detection is very effective against known attacks, and it depends on the receiving of regular updates of patterns and will be unable to detect unknown previous threats or new releases.

1. **Anomaly based detection system**

This type of detection depends on the classification of the network to the normal and anomalous, as this classification is based on rules or heuristics rather than patterns or signatures and the implementation of this system we first need to know the normal behavior of the network. Anomaly based detection system unlike the misuse based detection system because it can detect previous unknown threats, But the false positive to rise more probably.

1. **Specification based detection system**

This type of detection systems is responsible for monitoring the processes and matching the actual data with the program and in case of any Abnormal behavior will be issued an alert and must be maintained and updated whenever a change was made on the surveillance programs in order to be able to detect the previous attacks the unknown and the number of false positives what can be less than the anomaly detection system approach (Ashoor & Gore, 2011)

**CLASSIFICATION OF INTRUSION DETECTION** **SYSTEM**

Intrusion detection system are classified into three types:

* Host based IDS
* Network based IDS
* Hybrid based IDS or Mixed IDS

1. **Host based IDS (HIDS)**

Host based IDS is single computer specific intrusion detection system which monitors the security of that system or computer from internal and external attacks. The internal attacks refer to the situation where it detects which program access which resource and is there any security break (Pharate, Bhat et al. 2015).

This type is placed on one device such as server or workstation, where the data is analyzed locally to the machine and are collecting this data from different sources. HIDS can use both anomaly and misuse detection system.

1. **Network based IDS (NIDS)**

NIDS are deployed on strategic point in network infrastructure. The NIDS can capture and analyze data to detect known attacks by comparing patterns or signatures of the database or detection of illegal activities by scanning traffic for anomalous activity. NIDS are also referred as “packet-sniffers”, Because it captures the packets passing through the of communication mediums.

1. **Hybrid based IDS or Mixed IDS**

MIDS Combines two types or more of IDS to achieve the advantages of IDS and complete an accurate detection such as Double Guard that uses host ids and network IDS. However, MIDS takes a long time in analyzing data. (Othman et al., 2018)The management and alerting from both network and host based intrusion detection devices, and provide the logical complement to NID and HID - central intrusion detection management.

An intrusion detection system is a part of the defensive operations that complements the defences such as firewalls, UTM etc. The intrusion detection system basically detects attack signs and then alerts. According to the detection methodology, intrusion detection systems are typically categorized as misuse detection and anomaly detection systems. The deployment perspective, they can be classified in network based or host-based IDS. In current intrusion detection systems where information is collected from both network and host resources. In terms of performance, an intrusion detection system becomes more accurate as it detects more attacks and raises fewer false positive alarms.

# **2.2 ARTIFICIAL INTELLIGENCE IN IDS**

Artificial Intelligence is concerned with improving algorithms by employing problem solving techniques used by human beings Humans excel at tasks such as learning or gaining the ability to perform tasks from examples and training an expert system handles problems using a computer model of expert human reasoning. However most expert systems must undergo continuous maintenance to perform well. Other systems can acquire knowledge from a set of training instances. These training instances can be questions and correct answer pairs or problems and the steps of a solution. Rule Based Induction derives rules which explain the training instances more clearly than a mathematical or statistical analysis of data. Classifier systems attempt to learn how to classify future examples from a set of training data. An example of a system that can be used as a classier is a Neural Network which uses a model of biological systems to perform classification.

Neural networks are characterized by highly connected networks which are trained on a set of data in the hopes that the network will correctly classify future examples. Another example of a classier is a Decision Tree. Decision trees are constructed by finding ways to separate the data into two or more groups. We then separate each of these groups in turn until we have small groups of examples left. Decision tree algorithms are designed to find the best questions to ask so that most or all of the examples in each group belong to one class. The goal of Feature Selection is to reduce the amount of information required to make good predictions and to improve the error rate of classifiers. This is accomplished by searching subsets of features or information sources and testing the ability of those features to classify the training instances. The search process itself is the subject of continuing research in the AI community. Humans are also able to generalize or abstract from large amounts of information by a process called discovery or clustering. Data clustering techniques are used to group data together according to some criteria. Clustering is used to discover hidden patterns in data that humans might miss (Frank, J, 2000)

## **2.2.1 MACHINE LEARNING IN IDS**

There are two main types of machine learning: supervised and unsupervised learning. Supervised learning relies on useful information in labeled data. Classification is the most common task in supervised learning (and is also used most frequently in IDS); however, labeling data manually is expensive and time consuming. Consequently, the lack of sufficient labeled data forms the main bottleneck to supervised learning. In contrast, unsupervised learning extracts valuable feature information from unlabeled data, making it much easier to obtain training data. However, the detection performance of unsupervised learning methods is usually inferior to those of supervised learning methods.

The traditional machine learning models (shallow models) for IDS primarily include the artificial neural network (ANN), support vector machine (SVM), K-nearest neighbor (KNN), naïve Bayes, logistic regression (LR), decision tree, clustering, and combined and hybrid methods. Some of these methods have been studied for several decades, and their methodology is mature. They focus not only on the detection effect but also on practical problems, e.g., detection efficiency and data management (Liu, Hongyu & Lang, Bo, 2019)

## **2.2.2 DEEP LEARNING IN IDS**

The capacity of deep learning approaches to automatically learn and extract complex patterns from vast datasets has propelled them to the forefront of machine learning. One popular application of deep learning architectures in intrusion detection is the use of Convolutional Neural Networks (CNNs), while Recurrent Neural Networks (RNNs) are also used. CNNs excel at handling image-based attacks or multidimensional arrays of network traffic data. In contrast, RNNs excel at processing sequences and are hence well-suited to time-series network data. Improved precision in identifying complicated intrusion patterns is a direct result of their ability to capture temporal and geographical relationships in data (Ahmed, U. Sarwar, A., & et al. & Nazir, M., 2025)

## **2.2.3 NATURAL LANGUAGE PROCESSING (NLP) IN IDS**

Natural Language Processing (NLP), a set of tools designed to extract insights from unstructured text, stands out in this context. NLP's core capabilities are diverse, including named entity recognition and sentiment analysis. Topic modeling, document classification, and topic modeling are also included. The application of NLP in cybersecurity holds promise for facilitating anomaly detection, identifying novel threats, pinpointing misconfigurations, or unauthorized access. The use of NLP in cybersecurity is not without its challenges. Despite the benefits that NLP can provide, there are several obstacles to overcome. The nuanced meanings and complex semantics of cybersecurity texts are a major obstacle, making it impossible to use standard NLP tools. It is essential to have a deep understanding of domain-specific complexities in order to achieve accuracy for tasks like entity extraction, threat modelling, and intent detection. This problem is compounded by the lack of cybersecurity corpora that are essential to train robust NLP models tailored for the intricacies within the cybersecurity domain. NLP persists despite these obstacles, due to the wealth and depth of insights contained within unstructured data (Arjunan, T., 2024)

# **2.3 CHALLENGES OF IDS DELOYMENT**

Although there has been a lot of research on IDSs, many essential matters remain. IDSs have to be more accurate, with the capability to detect a varied ranging of intrusions with fewer false alarms and other challenges.

## **2.3.1 CHALLENGES OF INTRUSION DETECTION SYSTEMS (IDS) FOR INDUSTRIAL CONTROL SYSTEMS (ICSS)**

Industrial Control Systems (ICSs) typically consist of two main components: Supervisory Control and Data Acquisition (SCADA) hardware, which gathers information from sensors and controls mechanical machinery, and software that allows human operators to manage these machines.

Cyberattacks on ICSs pose significant challenges for Intrusion Detection Systems (IDS) due to the unique architectures and operational characteristics of these systems. Attackers are increasingly targeting ICSs, as demonstrated by notable incidents such as the Stuxnet attack, widely regarded as the first instance of cyber warfare. Unlike typical cyberattacks, Stuxnet's primary focus was likely the Iranian nuclear program (Nourian & Madnick, 2015). Threats to ICSs can originate from state-sponsored actors, competitors, insider threats with malicious intent, or hacktivists.

The impact of compromised ICSs can be catastrophic, affecting public health and safety, national security, and economic stability. Past incidents have shown that compromised ICSs can result in widespread power outages, hazardous chemical releases, and even explosions. To ensure reliable, safe, and flexible operations, it is vital to implement secure ICSs (Khraisat et al., 2019).

Developing IDS solutions tailored to ICSs requires addressing their unique architectures, real-time operational demands, and dynamic environments to safeguard critical facilities from cyberattacks effectively. Some of the previous attacks in ICS system are given below:

* In 2008, Conficker malware infected ICS systems, such as an aeroplane’s internal systems. Conficker disables many security features and automatic backup settings, erases stored data and opens associations to get commands from a remote PC (Pretorius & van Niekerk, [2016](https://link.springer.com/article/10.1186/s42400-019-0038-7#ref-CR86))
* In 2017, WannaCry ransomware spread globally and seriously effected the National Health System, UK and prevented emergency clinic specialists from using health systems (Mohurle & Patil, [2017](https://link.springer.com/article/10.1186/s42400-019-0038-7#ref-CR81)).
* . On May 12, the “WannaCry” ransomware began affecting dozens of NHS facilities. Eventually, more than 60 NHS trusts were hit. Many facilities could not access the records, which led to delays of non-urgent surgeries and cancelled patient appointments. Some hospitals had to divert ambulances to other facilities.(R. 2017)

Legacy systems, such as those using outdated Microsoft operating systems, are at greater risk of ransomware and zero-day malware because they no longer receive security updates. Similarly, Industrial Control Systems (ICSs) often rely on legacy applications that can’t be easily updated, making them vulnerable to cyber threats. Unfortunately, most current intrusion detection methods focus only on software-level protection, which isn’t enough to handle today’s sophisticated attacks. A better approach is to combine hardware-based and software-based Intrusion Detection Systems (IDS), using the strengths of both Host-based (HIDS) and Network-based (NIDS) systems to detect complex and zero-day threats across all layers.

## **2.3.2 CHALLENGE OF IDS ON INTRUSION EVASION DETECTION**

Detecting attacks masked by evasion techniques is a challenge for both SIDS and AIDS. The ability of evasion techniques would be determined by the ability of IDS to bring back the original signature of the attacks or create new signatures to cover the modification of the attacks. Robustness of IDS to various evasion techniques still needs further investigation. For example, SIDS in regular expressions can detect the deviations from simple mutation such as manipulating space characters, but they are still useless against a number of encryption techniques

# **2.4 RELATED WORKS**

* (Sharma, V., Shah, D., Sharma, S., & Gautam, S., 2024)discussed the implementation of an AI-based IDS utilizing machine learning algorithms to analyze network traffic patterns and identify anomalous activities. Publicly available datasets containing labeled normal and malicious traffic to train and evaluate the system's effectiveness were utilized. However, the authors acknowledge the challenges associated with achieving high detection rates while minimizing false positives. They emphasize the importance of continuous learning and adaptation to effectively address the evolving nature of cyber threats.
* (Ahmed et al., 2024)introduced a novel Hybrid Adaptive Ensemble for Intrusion Detection (HAEnID), a sophisticated approach to improving the accuracy of intrusion detection systems. The HAEnID combines the strengths of various machine learning models, including decision trees, support vector machines, and neural networks, within a single ensemble. A key feature of this model is its emphasis on explainable AI, aiming to provide insights into the reasoning behind the system's decisions. The study evaluates the HAEnID's performance using the NSL-KDD dataset and acknowledges the trade-off between enhanced detection accuracy and increased computational complexity.
* (Rajapaksha, S., Kalutarage, H., Al-Kadri, M. O., & et al., 2023)The article reviews AI-based intrusion detection systems for in-vehicle networks, categorizing detection techniques and attack types. It explores machine learning and deep learning methods, such as anomaly and misuse detection models, while addressing the limited availability of comprehensive in-vehicle network traffic datasets. Key challenges include achieving real-time processing and adapting models to diverse vehicle architectures.

# **2.5 DATASET FOR IDS DEVELOPMENT**

Intrusion detection datasets are crucial for validating any IDS approach, as they enable the assessment of a method's effectiveness in identifying intrusive behavior. Due to privacy concerns, network packet analysis datasets used in commercial products are often inaccessible. However, several publicly available datasets, such as DARPA, KDD, NSL-KDD, and ADFA-LD, are commonly used as benchmarks in the field (Khraisat et al., 2019)

## **2.5.1 EXISTING DATASETS USED FOR DEVELOPING AND EVALUATING IDS**

This section discusses some of the existing datasets used for developing and evaluating IDS, highlighting their features and limitations.

### **2.5.1.1 DARPA / KDD CUP99**

The Defense Advanced Research Projects Agency (DARPA) made the first significant effort to create an IDS dataset in 1998, resulting in the Knowledge Discovery and Data Mining (KDD98) dataset. That same year, DARPA initiated a program at MIT Lincoln Labs to develop a realistic and comprehensive benchmarking environment for IDS (MIT Lincoln Laboratory, 1999). While this dataset was a pivotal contribution to IDS research, it has been widely criticized for its limitations in reflecting real-world conditions (Creech & Hu, 2014b).

The dataset was generated using multiple computers connected to the Internet to simulate a small U.S. Air Force base with restricted personnel. Network packets and host log files were collected using an experimental testbed designed by Lincoln Labs. Over two months, they captured approximately four gigabytes of TCP packet dumps for a Local Area Network (LAN), simulating a typical U.S. Air Force LAN environment. This environment included a mix of normal activities and several simulated intrusions.

The dataset contained around 4.9 million records, with the two-week test data comprising approximately 2 million connection records. Each record included 41 features and was labeled as either normal or abnormal. The extracted data represented TCP sessions with well-defined start and end times, capturing interactions between source and target IP addresses. These sessions included a variety of simulated attacks in a military network setting.

The 1998 DARPA dataset served as the foundation for creating the KDD Cup 99 dataset, which was later used in the Third International Knowledge Discovery and Data Mining Tools Competition (KDD, 1999).

These datasets are considered outdated as they lack records of recent malware attacks. For instance, attackers' behaviors vary across different network topologies, operating systems, software, and crime toolkits. Despite these limitations, the KDD99 dataset continues to be widely used as a benchmark in the IDS research community and remains a common choice among researchers (Alazab et al., 2014; Duque & Omar, 2015; Ji et al., 2016).

### **2.5.1.2 CAIDA**

This dataset, collected in 2007, captures network traffic traces from Distributed Denial-of-Service (DDoS) attacks . DDoS attacks aim to disrupt the normal traffic of a targeted computer or network by flooding it with excessive network packets, thereby preventing legitimate traffic from reaching its destination. However, the CAIDA dataset has some notable limitations. It lacks diversity in the types of attacks and does not include features from the entire network, making it challenging to differentiate between normal and abnormal traffic flows.

### **2.5.1.3 NSL-KDD**

The NSL-KDD dataset is a public dataset developed from the earlier KDD Cup 99 dataset (Tavallaee et al., 2009). A statistical analysis of the KDD Cup 99 dataset revealed significant issues, such as a high volume of duplicate packets, which negatively impact intrusion detection accuracy and lead to misleading evaluations of IDS (Tavallaee et al., 2009).

One of the primary problems with the KDD dataset is that approximately 78% of the training packets and 75% of the testing packets are duplicates. This redundancy biases machine learning models toward normal instances, hindering their ability to learn irregular instances that are often more damaging to systems. To address these issues, (Tavallaee et al., 2009) developed the NSL-KDD dataset in 2009 by removing duplicate records.

### **2.5.1.4 CICIDS 2017**

The NSL-KDD dataset includes 125,973 records in the training set and 22,544 records in the test set. Its manageable size allows researchers to use the entire dataset without random sampling, enabling consistent and comparable results across studies. The dataset consists of 22 types of intrusion attacks and 41 features. Among these, 21 features describe the connection itself, while 19 features characterize the nature of connections within the same host (Tavallaee et al., 2009).

The CICIDS2017 dataset includes both benign behaviors and detailed information on new malware attacks, such as Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet, and DDoS (Sharafaldin et al., 2018). The dataset is labeled with timestamps, source and destination IPs, source and destination ports, protocols, and attack types.

To collect this dataset, a complete network topology was set up, comprising modems, firewalls, switches, routers, and nodes running various operating systems, including Microsoft Windows (Windows 10, 8, 7, and XP), Apple macOS, iOS, and Linux. The dataset captures 80 network flow features from the recorded network traffic.

## **2.5.2 FEATURE SELECTION FOR IDS**

Feature selection is a crucial step in improving Intrusion Detection Systems (IDS), as it helps reduce computational complexity, eliminate redundant data, enhance detection rates, and simplify the data while minimizing false alarms. Some techniques have been developed to create lightweight IDS solutions using feature selection.

With the increase in the usage of the Internet, a large amount of information is exchanged between different communicating devices. The data should be communicated securely between the communicating devices and therefore, network security is one of the dominant research areas for the current network scenario (Thakkar 2021).

Feature selection methods are generally divided into two categories: wrapper and filter methods. Wrapper methods assess subsets of variables to identify potential interactions between them. However, these methods have two main drawbacks: they can lead to overfitting when there’s insufficient data, and they can be time-consuming when dealing with a large number of variables.

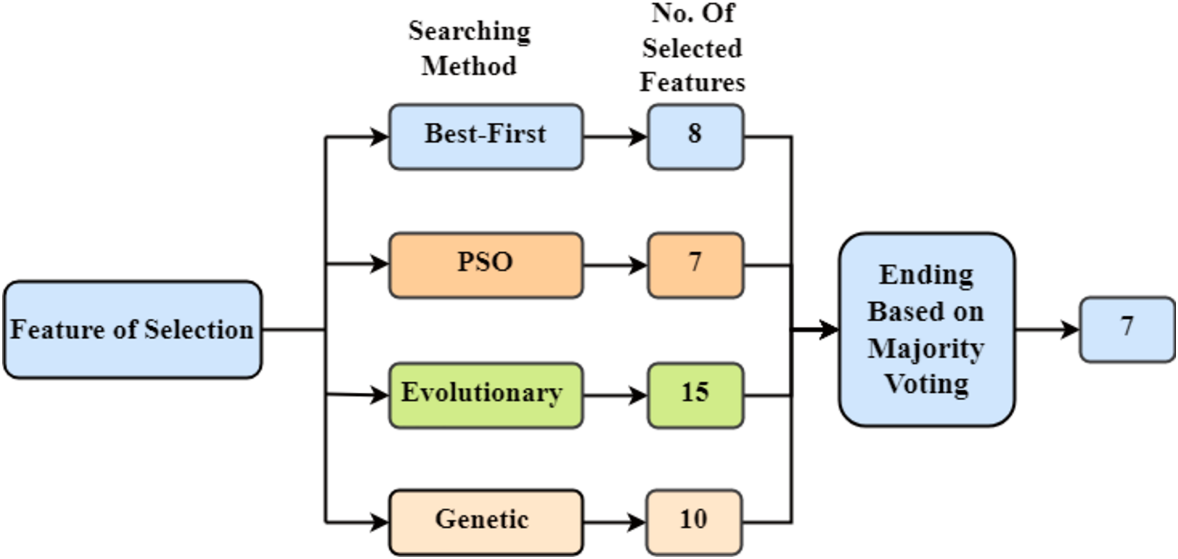
On the other hand, filter methods are typically used as a pre-processing step, independent of machine learning techniques. These methods select features based on statistical tests, ranking them by their correlation with the target variable. The overall methodology for feature selection is presented in figure 2.1 below.

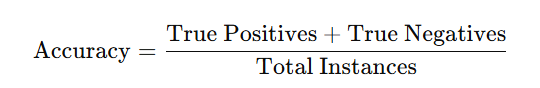
Figure 2.1: Feature Selection Methods.

# **2.6 EVALUATION METRICS IN IDS**

This study uses several assessment metrics to evaluate the proposed intrusion detection model: accuracy, precision, Recall, and F1-score. These mathematical metrics measure different aspects of model performance and thus are widely used for evaluating the model (Ahmed et al., 2024).

## **2.6.1 ACCURACY**

Accuracy measures the proportion of correctly classified instances among all instances, as shown in Equation 1.



While accuracy provides an overall measure of model correctness, it may be misleading in the presence of class imbalance. For example, a dataset where 95% of samples are benign can result in high accuracy even if all attacks are misclassified. Therefore, accuracy must be complemented by other metrics for a balanced evaluation (Ahmed et al., 2024)

## **2.6.2 PRECISION**

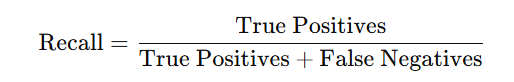
Precision, detailed in Equation 2, measures the proportion of correctly predicted positive instances out of all predicted positives:



In the context of intrusion detection, high precision reduces false positives, which helps minimize unnecessary alerts that may waste resources or disrupt services (Ahmed et al., 2024).

## **2.6.3 RECALL**

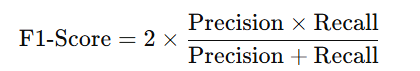
Recall, represented in Equation 3, evaluates the model’s ability to identify all positive instances (true positives):



High recall is essential in intrusion detection to minimize missed attacks (false negatives), which can lead to severe consequences (Ahmed et al., 2024)

## **2.6.4 F1-SCORE**

The F1-score as represented in equation 4, a harmonic mean of precision and recall, is particularly useful for handling imbalanced datasets.



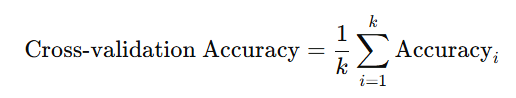
This metric balances the trade-off between precision and recall, making it suitable for scenarios where both false positives and false negatives carry significant costs, such as intrusion detection (Ahmed et al., 2024)

## **2.6.5 ROC CURVE AND AUC**

The Receiver Operating Characteristic (ROC) curve plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings, while the Area Under the Curve (AUC) quantifies the classifier’s performance . A higher AUC indicates better model performance (Ahmed et al., 2024)

## **2.6.6 CROSS-VALIDATION**

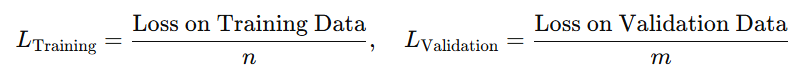
To assess model robustness, a k-fold cross-validation method with k=5k=5 was utilized, as illustrated in equation 5:



This technique ensures consistency and generalizability by splitting the dataset into kk parts, training on k−1k-1 parts, and testing on the remaining part.

## **2.6.7 LEARNING CURVES**

Learning curves were used to visualize model performance against the size of the training dataset. These curves help identify issues like overfitting or underfitting and guide decisions on the optimal training data size as shown in equation 6:



where LL is the loss function, nn is the number of training samples, and mm is the number of validation samples.

## **2.6.8 MODEL TRANSPARENCY TECHNIQUES**

Techniques such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations) were applied to improve trust and comprehension in decision-making. These methods provide insights into the contribution of individual features to the model’s predictions, enhancing interpretability and reliability.

This formatted section is ready for direct inclusion in your project and aligns with academic writing standards. Let me know if you need any adjustments!

# **2.7 SECURITY AND PRIVACY CONCERNS IN IDS**

Over the past two decades, numerous intrusion detection tools have been developed and implemented. Early systems primarily focused on anomaly detection, but modern tools are predominantly misuse detection-based, often combined with hybrid approaches. Notable early systems include IDES, Haystack, and MIDAS, while more recent tools include Emerald and Bro (Lundin, Emilie & Jonsson, Erland, 2000)

## **2.7.1 CHALLENGE OF IDS ON INTRUSION EVASION DETECTION**

Detecting attacks masked by evasion techniques is a challenge for both SIDS and AIDS. The ability of evasion techniques would be determined by the ability of IDS to bring back the original signature of the attacks or create new signatures to cover the modification of the attacks. Robustness of IDS to various evasion techniques still needs further investigation. For example, SIDS in regular expressions can detect the deviations from simple mutation such as manipulating space characters, but they are still useless against a number of encryption techniques (Khraisat, Ansam et al., 2019).

**CHAPTER 3**

**METHODOLOGY**

# **3.1 SYSTEM ANALYSIS**

This section presents a comprehensive analysis of the existing system and its limitations, as well as the motivation behind the proposed AI-based Intrusion Detection System (IDS). It examines the background, benefits, setbacks, and technical requirements needed to address the identified security challenges. This analysis forms the foundation upon which the proposed solution is designed and developed.

## **3.1.1 SYSTEM DESCRIPTION**

The system under study is an Intrusion Detection System (IDS), a cybersecurity mechanism designed to monitor, detect, and report unauthorized, anomalous or suspicious activities within a network environment. Traditional IDS typically rely on either signature-based or anomaly-based detection methods. Signature-based systems match incoming traffic against a database of known attack patterns, while anomaly-based systems identify deviations from established normal behavior. Although these approaches have been fundamental in defending computer networks, they often suffer from high false positive rates, delayed detection, and poor adaptability to novel or evolving threats.

In the current digital landscape, where cyberattacks have become more dynamic and complex, relying solely on traditional IDS approaches does not seem sufficient anymore. Many systems struggle to accurately differentiate between legitimate and malicious activities, especially in large-scale environments with high volumes of data. This often overwhelms security analysts with redundant or irrelevant alerts, leading to alert fatigue and slower incident response times.

The proposed system addresses these limitations by leveraging **Artificial Intelligence (AI)** and **Explainable AI (XAI)** techniques to enhance the detection accuracy and interpretability of IDS. The system is designed to process input network data, perform intelligent feature selection, train a machine learning model, and generate alerts with transparent explanations. Through the integration of models such as **SHAP** and **LIME**, this system not only makes intrusion detection more efficient but also provides insights into why specific decisions were made, fostering trust and aiding faster human response.

Ultimately, the AI-driven IDS aims to function as a real-time, intelligent security layer that continuously evolves by learning from network behavior, reducing false positives, and ensuring that even previously unseen attacks can be identified and explained clearly.

## **3.1.2 HISTORICAL BACKGROUND OF THE CASE STUDY**

The increasing frequency and sophistication of cyber threats have made the development of security mechanisms capable of detecting and mitigating unauthorized access in digital environments necessary. The concept of intrusion detection has evolved over time from simple rule-based monitoring systems to advanced solutions incorporating artificial intelligence (AI) and machine learning (ML). Traditional IDS were primarily signature-based and relied on known patterns of attacks, which limited their ability to detect novel threats or adapt to new environments.

The use of AI in intrusion detection began gaining prominence in the early 2000s, with the introduction of intelligent systems capable of learning from data and making predictions about malicious activity. The integration of AI methods such as decision trees, support vector machines, and later deep learning architectures allowed IDS to become more dynamic and accurate.

This project builds upon previous research efforts by applying a modern AI approach to enhance the effectiveness of IDS. It considers the limitations of previous systems particularly high false positives and lack of adaptability and proposes a more robust and interpretable model that addresses these challenges. By adopting a historical view of how IDS solutions have advanced over time, this case study contextualizes the need for innovative methodologies in tackling current cybersecurity issues.

## **3.1.3 EVALUATION OF THE CURRENT SYSTEM**

The existing Intrusion Detection Systems (IDS) in most organizations rely heavily on traditional methods such as signature-based and anomaly-based detection. While these systems have served as the first line of defense in identifying known attack patterns, their effectiveness is increasingly limited in today’s dynamic cyber threat landscape.

Signature-based IDS, although highly accurate in detecting previously known threats, fail to identify novel or zero-day attacks. On the other hand, anomaly-based systems, while capable of detecting unknown threats, tend to generate a high number of false positives, overwhelming security analysts and reducing trust in alert quality (Khraisat et al., 2019). This creates alert fatigue and delays in responding to actual threats.

Furthermore, many current systems lack adaptability and real-time learning capabilities. Once deployed, they do not improve automatically over time or learn from new data unless manually updated. This static nature reduces their relevance in fast-evolving network environments.

Another major limitation is the lack of transparency. Many modern ML-based IDS operate as “black boxes,” offering little to no explanation behind the decisions they make. This lack of interpretability hinders adoption and makes it difficult for cybersecurity experts to validate or trust the model’s output (Wang et al., 2020).

Overall, while the current IDS implementations provide a foundational level of protection, they are insufficient to meet the demands of a modern threat environment. This project aims to address these issues by developing an AI-powered, explainable IDS model that improves accuracy, reduces false alarms, and enhances decision transparency.

## **3.1.4 BENEFITS OF THE CURRENT SYSTEM**

Despite the limitations observed in traditional Intrusion Detection Systems (IDS), they still offer several key benefits that have supported organizations in maintaining a baseline level of cybersecurity.

### **3.1.4.1 EARLY DETECTION OF KNOWN ATTACKS**

Signature-based IDSs are highly effective at detecting previously encountered and well-documented threats. When kept up-to-date with threat signatures, they can quickly alert administrators to the presence of malicious activity.

### **3.1.4.2 LOW FALSE ALARM RATES FOR KNOWN PATTERNS**

For attack patterns that are known and well-defined, signature-based IDSs produce very few false positives, allowing security personnel to act with confidence when alerts are raised.

### **3.1.4.3 NETWORK MONITORING AND VISIBILITY**

IDSs provide a centralized view of the system or network traffic, helping administrators monitor activity in real time and detect unusual behaviors. This visibility is crucial for situational awareness in cybersecurity operations.

### **3.1.4.4 POLICY ENFORCEMENT**

IDS tools help organizations enforce security policies by monitoring for violations such as unauthorized access, use of restricted ports or protocols, and suspicious login attempts.

### **3.1.4.5 SCALABILITY IN NETWORK-BASED SYSTEM**

Network-Based Intrusion Detection Systems (NIDS) can be scaled across large enterprise networks to monitor traffic between multiple hosts, providing protection beyond individual systems.

### **3.1.4.6 COST EFFECTIVE FOR BASIC OPERATIONS**

Many traditional IDS solutions are open-source or relatively low-cost and easy to integrate with existing security infrastructures, making them accessible to smaller organizations.

While these benefits make the current systems useful, their effectiveness is increasingly challenged by advanced persistent threats (APTs), zero-day vulnerabilities, and the sheer scale of modern network environments. This underscores the need for AI-driven and explainable IDS solutions capable of adapting to emerging threat patterns.

## **3.1.5 PROBLEMS OF THE CURRENT SYSTEM**

Despite the progress in the development of Intrusion Detection Systems (IDS), several limitations still hinder their effectiveness in modern network environments. One of the major challenges is the high rate of false positives and false negatives generated by traditional IDS. These inaccuracies not only burden security analysts with irrelevant alerts but also increase the likelihood of overlooking actual attacks (Khraisat et al., 2019).

Moreover, many existing IDS rely on static signature-based approaches that are ineffective against zero-day and polymorphic attacks. These systems are only capable of detecting known threats with predefined signatures, leaving networks vulnerable to evolving attack patterns. Additionally, the increasing complexity and volume of network traffic make it difficult for conventional IDS to maintain high detection accuracy while operating efficiently in real-time.

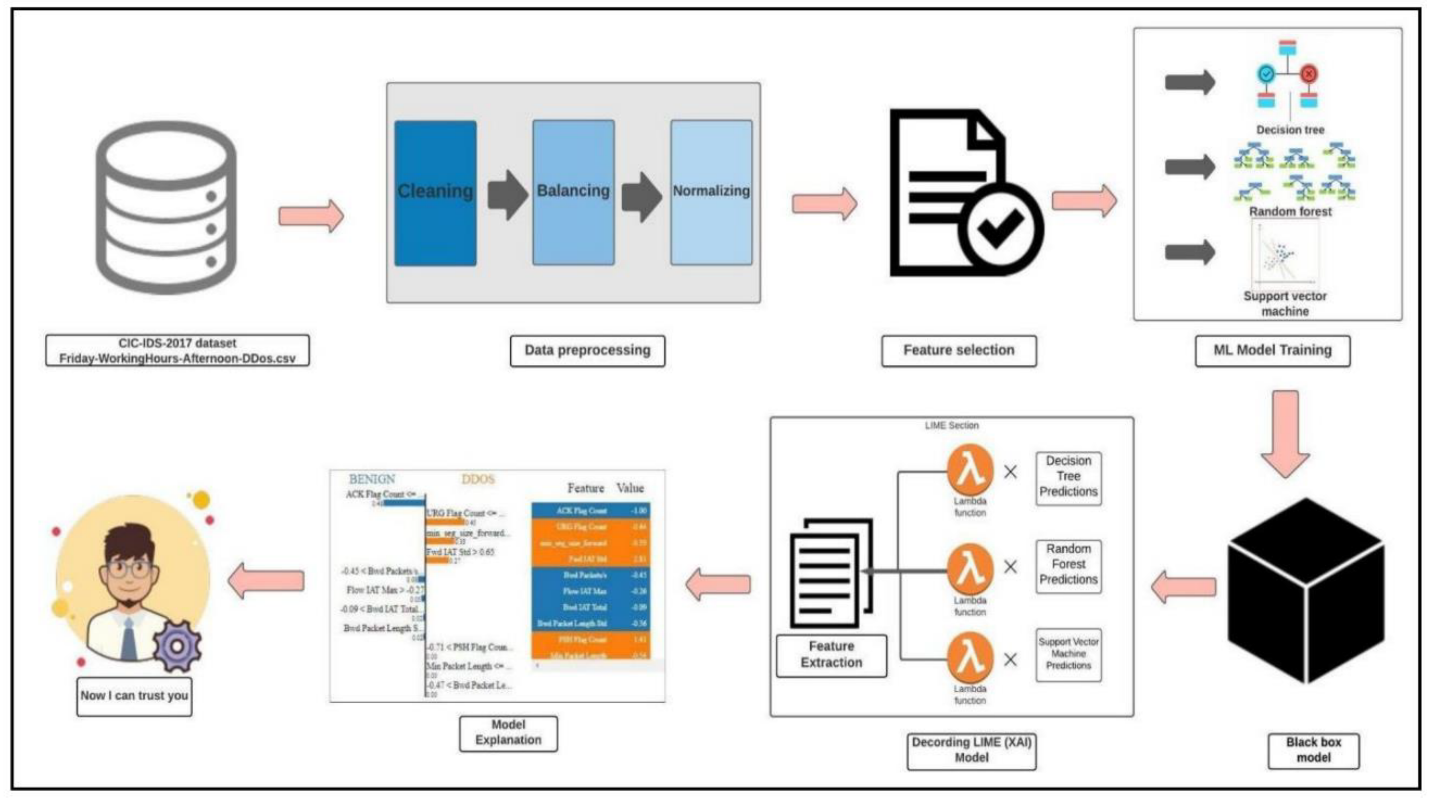
Another significant drawback is the lack of interpretability in modern machine learning-based IDS. These models often function as "black boxes", offering little to no explanation for their predictions, which reduces trust and hinders timely decision-making by cybersecurity professionals (Wang et al., 2020). This limitation has led to growing interest in integrating Explainable AI (XAI) methods to enhance transparency and usability.

The absence of standard evaluation metrics and up-to-date, real-world datasets also contributes to the limited performance and generalizability of many IDS models (Tavallaee et al., 2009). Overall, these issues underline the need for more intelligent, adaptive, and explainable solutions to intrusion detection that can effectively meet the demands of today’s cybersecurity landscape.

## **3.1.6 TECHNICAL DETAILS OF SOLUTION TO THE PROBLEM**

The proposed solution to the challenges identified in existing Intrusion Detection Systems (IDS) involves the development of an AI-based IDS framework that integrates both Machine Learning (ML) and Explainable AI (XAI) techniques to enhance detection accuracy and transparency.

This system will utilize supervised learning algorithms for intrusion classification and apply XAI tools such as SHAP or LIME to provide insight into model decisions. By combining these technologies, the framework addresses the limitations of traditional IDS, especially in detecting new attack patterns and improving interpretability. Figure 3.1 illustrates the overall flow of the AI-based intrusion detection system, highlighting the preprocessing of input data, feature selection, model training, and intrusion classification.



**Figure 3.1:** Data Flow Diagram for Machine Learning-Based IDS.

This solution is structured into several technical components:

### **3.1.6.1 DATASET SELECTION AND PREPROCESSING**

The **NSL-KDD** dataset is chosen due to its balanced distribution and improvements over the KDD’99 dataset, such as the removal of redundant records which had skewed previous machine learning models (Tavallaee et al., 2009). The dataset includes labeled examples of both normal and attack traffic, which allows for training and evaluating the model’s performance. Preprocessing involves feature scaling, encoding categorical variables, and removing irrelevant features to reduce noise and computation time.

### **3.1.6.2 FEATURE SELECTION**

Feature selection is used to identify and retain the most relevant attributes that contribute to intrusion detection. Techniques like Recursive Feature Elimination (RFE), Information Gain, or Chi-square tests may be employed to improve model efficiency and reduce overfitting.

### **3.1.6.3 MODEL ARCHITECTURE**

The core model will be based on supervised learning algorithms such as Decision Trees, Random Forests, or a Deep Neural Network (DNN). These algorithms have demonstrated good performance in classifying attack types and normal traffic. To improve interpretability, models such as Decision Trees or Gradient Boosting Machines (e.g., XGBoost) can be used in combination with SHAP (SHapley Additive exPlanations) values.

### **3.1.6.4 EXPLAINABLE AI (XAI) INTEGRATION**

The black-box nature of many AI systems limits their real-world application in security. This solution integrates XAI tools like SHAP and LIME (Local Interpretable Model-agnostic Explanations) to provide local and global explanations for predictions. This transparency improves analyst trust and allows security teams to understand why a traffic pattern was flagged as malicious (Wang et al., 2020).

### **3.1.6.5 EVALUAITON METRICS AND VALIDATION**

To assess the performance of the system, standard metrics such as accuracy, precision, recall, F1-score, and ROC-AUC will be used. Additionally, **5-fold cross-validation** will be applied to test model generalizability and avoid overfitting. Performance benchmarks will be compared against existing IDS solutions.

### **3.1.6.6 SYTEM DEPLOYMENT CONSIDERATIONS**

The final solution is intended to be deployable on a local network or cloud-based architecture, depending on the use case. Integration with existing security infrastructure like SIEMs (Security Information and Event Management systems) can be facilitated via API interfaces or log stream ingestion.

By addressing both performance and explainability, this solution aims to offer a practical and robust AI-based IDS that is suitable for real-world implementation and can evolve with emerging cyber threats.

# **3.2 SYSTEM DESIGN**

## **3.2.1 OVERVIEW OF THE NEW SYSTEM**

The proposed solution is an AI-driven Intrusion Detection System (IDS) that aims to improve cyber threat detection within network environments. It achieves this by combining established machine learning methods with Explainable Artificial Intelligence (XAI) models. This integration seeks to address the limitations of conventional IDS solutions, specifically high false positive rates, a lack of transparency, and limited adaptability to new threats.

The architecture is structured in modular components including a **data collection module, preprocessing and feature selection unit, AI-based detection engine, XAI interpretability layer,** and **administrative dashboard** for user interaction and alert management. Each module is designed to ensure seamless integration and scalability within modern network infrastructures.

 **Data Collection Module**: Captures real-time network traffic data using packet sniffers and log analysis tools.

 **Preprocessing & Feature Selection**: Cleans, normalizes, and reduces irrelevant data using selected feature engineering techniques, improving model efficiency and detection accuracy.

 **Detection Engine**: Utilizes supervised learning models such as Decision Trees, Random Forests, and deep learning algorithms like CNN or RNN for anomaly and signature-based detection.

 **XAI Layer**: Incorporates SHAP and LIME techniques to explain the model's decisions to security analysts, increasing trust and facilitating quicker threat mitigation.

 **User Interface**: Provides a simple dashboard for monitoring alerts, viewing reports, and visualizing model decisions.

This system aims to not only detect both known and unknown threats efficiently but also to provide interpretable feedback that supports human decision-making, thus improving incident response and overall network security.

## **3.2.2 OUTPUT DESIGN**

The output design of an Intrusion Detection System (IDS) plays a crucial role in ensuring that the results of the system’s detection processes are presented in a clear, actionable, and timely manner to relevant stakeholders, such as network administrators or security analysts.

The proposed AI-based IDS is designed to provide **real-time alerts** and **explanations** for detected intrusions. To improve usability and decision-making, the system incorporates the following output components:

1. **Alert Notification Panel:** This interface generates and displays immediate alerts when a suspicious activity is detected. Alerts include:

* Timestamp of the detected event.
* Source and destination IP addresses.
* Type of threat detected (e.g., DoS, Probe, R2L, U2R).
* Threat severity level (Low, Medium, High).

1. **Visual Dashboards**: The system includes a web-based graphical dashboard to display summarized threat statistics such as:

* Number of attacks detected per category.
* Visualization of traffic anomalies.
* Detection trends over time (daily/weekly charts).

1. **Explainability Reports**: Since the model integrates Explainable AI (XAI) techniques, such as SHAP and LIME, the system provides explanations for each prediction, showing:

* The features that influenced the decision.
* The degree to which each feature contributed to the result.

1. **Exportable Logs**: All detections are logged in structured formats (e.g., CSV, JSON) that can be exported for further analysis or auditing.

This design ensures that users not only receive alerts but also gain **insight into the cause** of detections, thereby reducing false alarms and improving trust in the system’s predictions.

## **3.2.3 INPUT DESIGN**

Input design is the process of determining how users will interact with the system by entering data, commands, or instructions. For an AI-based Intrusion Detection System (IDS), the quality and structure of input data are critical because they directly influence the accuracy of threat detection and model performance.

The system is designed to accept data from multiple sources including:

* **Network traffic logs** (e.g., captured packet data)
* User activity records
* System event logs
* Labeled dataset such as NSL-KDD or CICIDS2017.

These inputs are preprocessed through several steps such as feature selection, normalization, and noise filtering to ensure consistency and accuracy. Input interfaces are expected to support:

* Uploading of datasets in .csv or .json formats.
* Real-time data ingestion through APIs or streaming tools (in a full deployment).
* Admin-controlled input validation checks to prevent corrupt or irrelevant data from affecting system output.

The design also incorporates a basic user interface (UI) where an admin or analyst can upload datasets or configure input parameters for the model (e.g., detection thresholds or feature selection options).

Proper input validation is implemented to:

* Ensure compatibility with model requirements.
* Prevent injection of malformed or malicious data.
* Guide users with error prompts if incorrect formats are detected.

Ultimately, this input design ensures that data fed into the system is clean, structured, and ready for effective intrusion analysis using AI-driven models.

## **3.2.4 DTABASE DESIGN**

A well-structured database is central to the efficient performance of the proposed AI-based Intrusion Detection System (IDS). The database will serve as the backbone for storing and retrieving essential data such as system logs, user credentials, network activity records, classification results, and metadata associated with detected intrusions.

### **3.2.4.1 DATABASE OBJECTIVES:**

* To ensure the secure and efficient storage of both raw and processed network traffic data.
* To maintain audit logs for traceability and system accountability.
* To support fast data retrieval for real-time anomaly detection and historical analysis.

### **3.2.4.2 DATABASE TYPE:**

A relational databse (e.g., MySQL or PostgreSQL) is selected for structured storage due to its:

* Simplicity in managing tabular data like logs and user records.
* Strong support for indexing and querying.
* Compatibility with the system’s back-end for smooth integration.

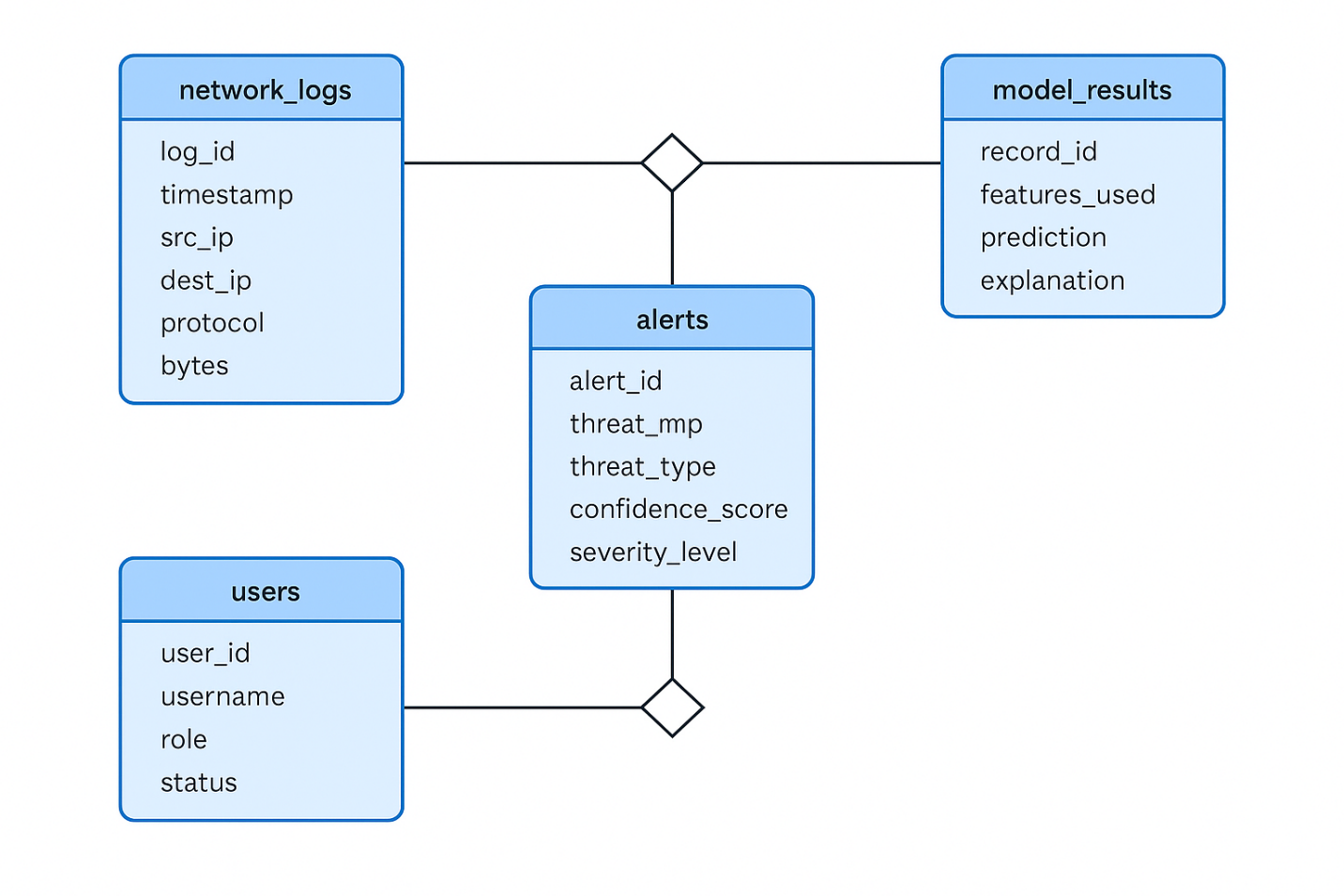
However, for large-scale implementations and real-time analytics, a hybrid architecture may integrate NoSQL solutions like MongoDB or Elasticsearch to handle unstructured logs or time-series data.

Key Tables and Descriptions

|  |  |  |
| --- | --- | --- |
| **Table Name** | **Description** | **Key Fields** |
| users | Stores registered system users | Username , user\_id, role, status |
| network\_logs | Holds raw or parsed network traffic data | log\_id, timestamp, src\_ip, dest\_ip, protocol, bytes |
| alerts | Contains alerts generated by the IDS model | alert\_id, timestamp, threat\_type, confidence\_score, severity\_level |
| audit\_trail | Stores login attempts, settings changes, etc. | event\_id, user\_id, action, timestamp |
| model\_results | Keeps record of the prediction outcomes from ML models | record\_id, features\_used, prediction, explanation |

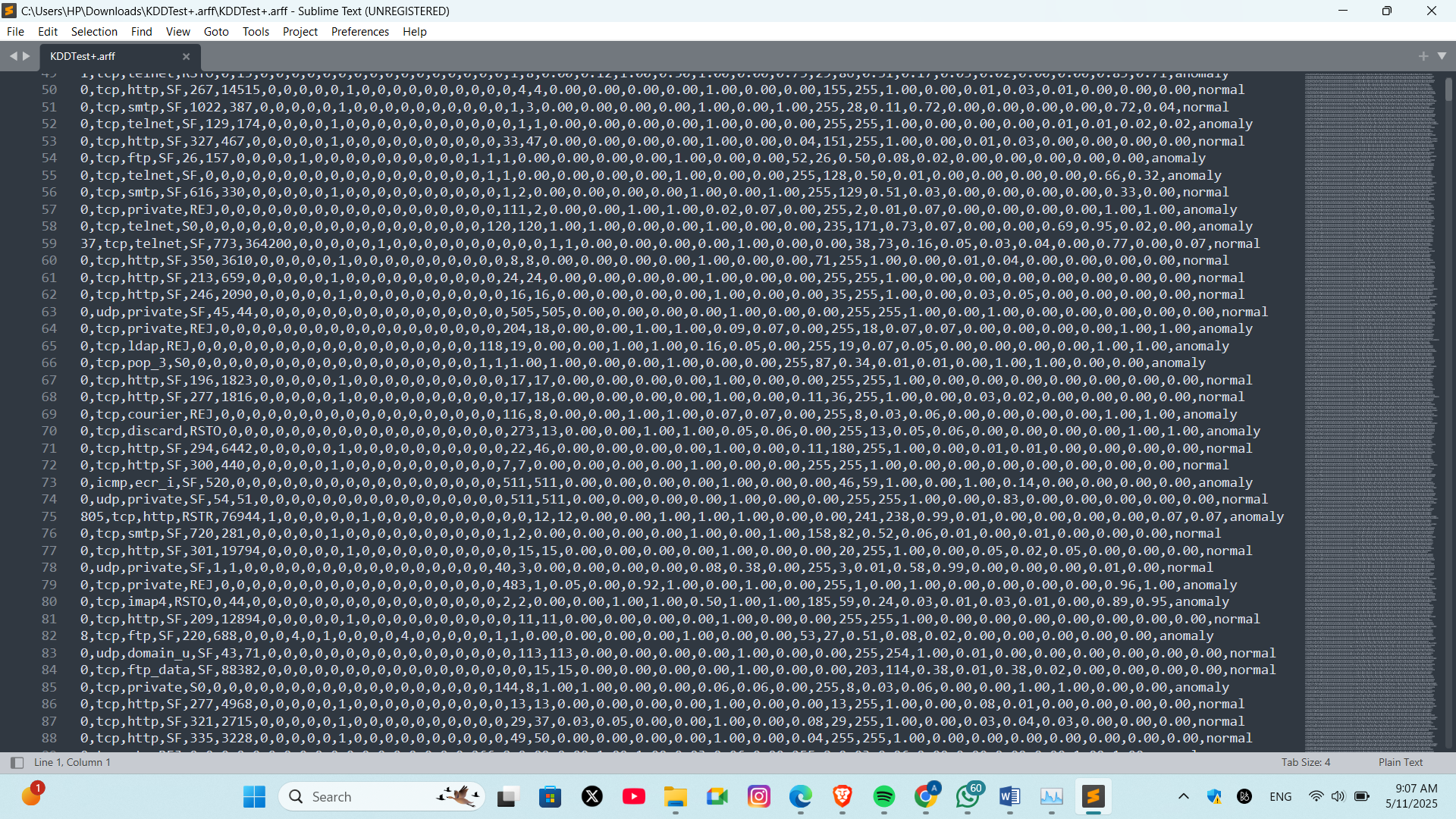
### **3.2.4.3 ENTITY-RELATIONSHIP DIAGRAM (ERD)**

The ERD illustrates the key entities in the AI-based Intrusion Detection System, including Users, Network\_Logs, and Alerts. It shows how users interact with detected alerts and how alerts are linked to specific network log entries. This design ensures organized data flow and supports effective system monitoring. A visual representation of the ERD is shown in Figure 3.2.

**Figure 3.2:** ERD showing the relationship among Users, Alerts, and Network Logs.

### **3.2.4.4 SAMPLE DATASET RECORD (NSL-KDD)**

To support model training and evaluation, this project utilizes the NSL-KDD dataset, a widely recognized benchmark dataset for intrusion detection research. The dataset provides labeled network traffic data, including normal and malicious activity, with 41 features describing different characteristics of each network connection. **Figure 3.2** is a sample of the dataset:



**Figure 3.2:** Sample Record from the NSL-KDD Dataset

### **3.2.4.5 SECURITY MEASURES**

* Data encryption for sensitive fields (e.g., user credentials).
* Role-based access control (RBAC) to restrict unauthorized data access.
* Regular backups and logging mechanisms.

## **3.2.5 PROCESS DESIGN**

The process design outlines the logical flow of activities involved in the functioning of the proposed AI-based Intrusion Detection System (IDS). This stage is essential for defining how the system transitions from one operation to another, ensuring that every component interacts correctly and efficiently.

The proposed IDS process begins with **data acquisition**, where traffic data is collected from network logs. The data then undergoes **preprocessing**, including cleaning, normalization, and feature selection. After preprocessing, the data is sent to the **Machine Learning module** for training and classification, which includes supervised or unsupervised algorithms for detecting anomalies or known attack signatures.

To enhance interpretability, the classified outputs are passed to the **Explainable AI (XAI)** module using tools such as SHAP or LIME to provide feature-level explanations of the detection. The final decisions are then reported via the **alerting interface** for the security team to review and take action.

Below is a simplified description of the process flow:

**1. Traffic Collection → 2. Data Preprocessing →**

**3. Feature Selection → 4. Intrusion Detection (ML Model) →**

**5. XAI Interpretation → 6. Alert Generation → 7. Action/Logging.**

This design ensures the system can:

* Process real-time and batch data.
* Provide transparency into its decision-making.
* Support timely and efficient responses to network threats.

**CHAPTER 4**

**IMPLEMENTATION**

This chapter outlines the implementation of the AI-Based Intrusion Detection System (IDS) developed using Python and Flask. It details the tools and technologies used, the system architecture, choice of programming language and platform, and the procedures followed to realize the software project.

# 4.1 OPERATING SYSTEM

An operating system (OS) is a vital system software that manages hardware resources and provides a platform for software development. It acts as an intermediary between users and the hardware, ensuring efficient execution of processes.

This project was developed on a personal computer running **Microsoft Windows 11 Pro**, a widely used operating system that provides a robust environment for web application development. It was particularly suitable due to its compatibility with development tools like **Visual Studio Code**, **Python**, and other frameworks used in this project.

## 4.1.1 CHOICE OF PROGRAMMING LANGUAGE

The programming languages and tools used in the development of the software are categorized into **frontend** and **backend** technologies:

* HTML (HyperText Markup Language): Used to structure content on web pages.
* Tailwind CSS (Cascading Style Sheets): Utilized for designing responsive and modern UI components via utility-first classes.
* **Python:** Served as the backend language, powering the logic of the application through the Flask web framework.
* **Jinja2:** Used as a templating engine to render HTML dynamically.
* **JavaScript (minimal):** Reserved for future enhancements and frontend interactivity.

The use of **Python** was driven by its strong ecosystem in data science and machine learning, making it ideal for implementing an AI-based detection system.

## 4.1.2 CHOICE OF DATABASE MANAGEMENT

The current version of this project does not utilize a traditional database management system like MySQL or MongoDB. Instead, it uses **local file-based storage** (CSV and ARFF datasets) for data ingestion and testing purposes. All user-uploaded files are saved into a designated **uploads folder**, and the system ph rocesses these files in-memory using **pandas.**

This lightweight approach was selected for simplicity, faster prototyping, and minimal setup overhead. However, the design is scalable and can be integrated with a full DBMS in future versions for logging, user management, and real-time alerting.

# 4.2 SYSTEM REQUIREMENT

System requirements are necessary specifications that ensure the software runs efficiently and without interruption. These are categorized as hardware and software requirements.

## 4.2.1 HARDWARE REQUIREMENTS

The minimum hardware specifications for running the IDS system include:

* A processor with a speed of 1GHz or higher
* 8GB RAM (minimum of 4GB)
* At least 10GB available disk space
* A modern web browser (Chrome, Firefox, or Edge).

## 4.2.2 SOFTWARE REQUIREMENTS

The software requirements includes:

* Operating system Windows 10/11 or any OS with Python support
* Python 3.10+
* Flask framework
* Visual Studio Code or any compatible IDE
* Internet Connectivity (for CDN-based Tailwind CSS and package installation)

# 4.3 SYSTEM TESTING

System testing is performed to verify the accuracy, performance, and stability of the application. Two main testing approaches were used:

* **Static Testing**: Code reviews and walkthroughs were done to check for logical errors, syntax issues, and code quality.
* **Dynamic Testing**: Actual execution of the system was performed to test file upload, preview, model detection, chart visualization, and summary statistics.

**Test Cases Included:**

* Uploading valid and invalid CSV files
* Handling ARFF file format errors
* Model prediction correctness
* Display of visualizations and summary statistics
* Downloading and reusing files

# 4.4 SYSTEM DOCUMENTATION

System documentation includes descriptions and usage guidelines for users and developers. It covers:

* **Setup Instructions**: Details on creating the virtual environment, installing dependencies, and running the app.
* **Folder Structure**: Organized into app, data, models, and templates for easy navigation.
* **Routing**: Flask routes defined for home, upload, preview, visualize, detect, summary, and download.
* **Screenshots**: The following section contains visual documentation of various pages in the system.

**Reference Lists:**

Ahmed, A., Asim, M., Ullah, I., & Ateya, A. A. (2024). An optimized ensemble model with advanced feature selection for network intrusion detection. *PeerJ Computer Science*, *10*, e2472.

Ahmed, U. Sarwar, A., & et al., & Nazir, M. (2025). *Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering.*

Arjunan, T. (2024). *Detecting anomalies and intrusions in unstructured cybersecurity data using natural language processing*.

Ashoor, A. S., & Gore, S. (2011). *Importance of intrusion detection system (IDS)*.

Bace, R., & Mell, P. (2001). *Intrusion detection systems*.

Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., Qian, B., Wen, Z., Shah, T., Morgan, G., & Ranjan, R. (2023). Explainable AI (XAI): Core Ideas, Techniques, and Solutions. *ACM Computing Surveys*, *55*(9), 1–33. https://doi.org/10.1145/3561048

Frank, J. (2000). *Artificial intelligence and intrusion detection*.

Hachmi, F., Boujenfa, K., & Limam, M. (2019). Enhancing the Accuracy of Intrusion Detection Systems by Reducing the Rates of False Positives and False Negatives Through Multi-objective Optimization. *Journal of Network and Systems Management*, *27*(1), 93–120. https://doi.org/10.1007/s10922-018-9459-y

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity*, *2*(1), 20. https://doi.org/10.1186/s42400-019-0038-7

Khraisat, Ansam, Gondal, Iqbal, Vamplew, Peter, & Kamruzzaman, Joarder. (2019). *Survey of intrusion detection systems: Techniques, datasets and challenges*. *2*(1), 1–22.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Liu, Hongyu & Lang, Bo. (2019). *Machine learning and deep learning methods for intrusion detection systems: A survey*. *9*(20), 4396.

Lundin, Emilie & Jonsson, Erland. (2000). *Anomaly-based intrusion detection: Privacy concerns and other problems*. *34*(4), 623–640.

Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, *9*(1), 381–386.

Mane, S., & Rao, D. (2021). *Explaining Network Intrusion Detection System Using Explainable AI Framework* (No. arXiv:2103.07110). arXiv. https://doi.org/10.48550/arXiv.2103.07110

Mirjat, N. A. (2024). AI and Machine Learning: Transforming the Landscape of Cybersecurity. *Bulletin of Engineering Science and Technology*, *1*(03), 40–59.

Mohale, V. Z., & Obagbuwa, I. C. (2025). A systematic review on the integration of explainable artificial intelligence in intrusion detection systems to enhancing transparency and interpretability in cybersecurity. *Frontiers in Artificial Intelligence*, *8*, 1526221. https://doi.org/10.3389/frai.2025.1526221

Mohapatra, H. (2017). *Introduction to IDS*.

Nourian, A., & Madnick, S. (2015). A systems theoretic approach to the security threats in cyber physical systems applied to stuxnet. *IEEE Transactions on Dependable and Secure Computing*, *15*(1), 2–13.

Othman, S. M., Ba-Alwi, F. M., Alsohybe, N. T., & Al-Hashida, A. Y. (2018). Intrusion detection model using machine learning algorithm on Big Data environment. *Journal of Big Data*, *5*(1), 34. https://doi.org/10.1186/s40537-018-0145-4

Rajapaksha, S., Kalutarage, H., Al-Kadri, M. O., & et al. (2023). *AI-based intrusion detection systems for in-vehicle networks: A survey. ACM Computing Surveys*.

Sharafaldin, I., Gharib, A., Lashkari, A. H., & Ghorbani, A. A. (2018). Towards a reliable intrusion detection benchmark dataset. *Software Networking*, *2018*(1), 177–200.

Sharma, V., Shah, D., Sharma, S., & Gautam, S. (2024). *Artificial intelligence-based intrusion detection system*.

Shirey, R. RFC 2828. (2000). *: Internet security glossary. The Internet Society.*

Sokratis K., K., & Georgios P., S. (2010). Reducing false positives in intrusion detection systems. *Computers & Security*, *29*(1), 35–44. https://doi.org/10.1016/j.cose.2009.07.008

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 1–6. https://doi.org/10.1109/CISDA.2009.5356528

Thapa, S., & Mailewa, A. (2020). The role of intrusion detection/prevention systems in modern computer networks: A review. *Conference: Midwest Instruction and Computing Symposium (MICS)*, *53*, 1–14. https://www.micsymposium.org/mics\_2020\_Proceedings/MICS2020\_paper\_1.pdf

Wang, M., Zheng, K., Yang, Y., & Wang, X. (2020). An explainable machine learning framework for intrusion detection systems. *IEEE Access*, *8*, 73127–73141.