**KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY-KUMASI**

**COLLEGE OF SCIENCE**

**DEPARTMENT OF COMPUTER SCIENCE FINAL YEAR PROJECT**



**PROJECT TOPIC:**

AN INTRUSION DETECTION SYSTEM USING MACHINE LEARNING MODEL.

**SUBMITTED BY:**

ECKHARD WALTER AMEDORME (4192320)

JESSE LANORE (4215420)

**SUPERVISED BY:**

DR. K.O PEASAH

# **DEDICATION**

We would like to dedicate this work first to the LORD for making it possible for us to reach this far in our academic laurels. Again, we want to thank our parents for their immense support and resilience to see us through tertiary education. Finally, to our able and passionate lecturers who took us through the various courses and assignments that make one a competent computer scientist, you have this as a sign of your hard work.

# **DECLARATION BY STUDENT**

We declare, without any reservation, that we personally undertook this project known as “INTRUSION DETECTION SYSTEM USING MACHINE LEARNING MODEL” on KNUST campus, herein submitted under supervision.

Signed: Date:

……………………… ………………………..

ECKHARD WALTER AMEDORME

(4192320)

Signed: Date:

……………………… …………………………

LANORE JESSE

(4215420)

# **DECLARATION BY SUPERVISOR**

I declare that I have personally supervised these students in undertaking the study report herein and I confirm that these students have my permission to present it for assessment.

Signed: Date:

……………………… ……………………….

DR. K.O PEASAH

# **Acknowledgement**

Our deepest gratitude goes to our supervisor, Dr. K.O Peasah, for his guidance, advice, and constructive comments on the Intrusion Detection System development process, requirement specifications, product design, and development. Besides, he also guided us in the software development life cycles and taught us how to build a good system. He also provided us with a lot of feedback during the system development which has increased the overall quality of this project. We would also like to take this opportunity to say thank you to all our friends who gave suggestions and ideas for this project and helped in system testing. Their help is truly appreciated.

# **Abstract**

Intrusion Detection System detects and responds to suspicious activities targeted at computing and communication resources, which have become the mainstream of information assurance in that the number of attacks has dramatically increased. IDS monitors and collects data from a target system that should be protected, processes and correlates the gathered information, and initiates responses when evidence of an intrusion is detected.

This project presents the design of an all-inclusive Intrusion Detection System to upgrade the security infrastructure of any network system. It makes use of both host-based and network-based intrusion detection techniques with the use of machine learning for threat and anomaly detection. In the proposed host-based mechanism, autoencoders will be implemented, which performs anomaly detection using system and application logs to monitor unusual activities. The network-based component makes use of a Random Forest model, trained using the KDD dataset, which detects suspicious network traffic. Proactive features are provisioned in the system for real-time blocking of unauthorized applications, devices, and websites.

Table of Contents

[**DEDICATION** i](#_Toc176102371)

[**DECLARATION BY STUDENT** ii](#_Toc176102372)

[**DECLARATION BY SUPERVISOR** iii](#_Toc176102373)

[**Acknowledgement** iv](#_Toc176102374)

[**Abstract** v](#_Toc176102375)

[**CHAPTER ONE** 1](#_Toc176102376)

[**Introduction** 1](#_Toc176102377)

[1.1 overview 1](#_Toc176102378)

[1.2 Problem Statement 9](#_Toc176102379)

[1.3 Project Aim 9](#_Toc176102380)

[1.4 Project Objectives 9](#_Toc176102381)

[1.5 Justification for The Project 11](#_Toc176102382)

[1.6 Motivation for Undertaking the Project 11](#_Toc176102383)

[1.7 Scope of the Project 12](#_Toc176102384)

[1.8 Project Limitations 13](#_Toc176102385)

[1.9 Project Beneficiaries 13](#_Toc176102386)

[1.10 Academic and Practical Relevance of the Project 14](#_Toc176102387)

[1.11 Project Activity Planning 14](#_Toc176102388)

[1.12 Structure of the Report 15](#_Toc176102389)

[1.13 Project Deliverables 15](#_Toc176102390)

[**CHAPTER TWO** 17](#_Toc176102391)

[**REVIEW OF RELATED SYSTEMS** 17](#_Toc176102392)

[2.1 Related Systems 17](#_Toc176102394)

[2.1.1 Signature-Based Intrusion Detection Systems 18](#_Toc176102395)

[2.1.2 Anomaly-Based Intrusion Detection Systems 18](#_Toc176102396)

[2.1.3 Hybrid Intrusion Detection Systems 19](#_Toc176102397)

[2.1.4 Machine Learning-Based Intrusion Detection Systems 19](#_Toc176102398)

[2.1.5 Host-Based vs Network-Based Intrusion Detection Systems 20](#_Toc176102399)

[2.2 Proposed System 21](#_Toc176102400)

[2.3 Architecture of the Proposed system 22](#_Toc176102401)

[2.4 Components Design and Components Descriptions 22](#_Toc176102402)

[2.5 Development Tools and Environment 25](#_Toc176102403)

[**CHAPTER 3** 26](#_Toc176102404)

[**METHODOLOGY** 26](#_Toc176102405)

[3.0 Overview 26](#_Toc176102406)

[3.1 Requirement Specification 26](#_Toc176102407)

[3.2 Stakeholders of the System 26](#_Toc176102408)

[3.3 Requirement Gathering Process 27](#_Toc176102409)

[3.4 Functional Requirements 28](#_Toc176102410)

[3.5 UML Diagrams 29](#_Toc176102411)

[3.5.1 Use Case diagram 29](#_Toc176102412)

[3.5.2 Activity Diagram 30](#_Toc176102413)

[3.5.3 Sequence Diagram 31](#_Toc176102414)

[3.5.4 Class Diagram 32](#_Toc176102415)

[3.6 Non-Functional Requirements 33](#_Toc176102416)

[3.7 Security Concept 33](#_Toc176102417)

[3.8 Project Methodology 34](#_Toc176102418)

[3.9 Various Software Process 35](#_Toc176102419)

[3.10 Chosen Model and Justification 36](#_Toc176102420)

[**CHAPTER 4** 38](#_Toc176102421)

[**IMPLEMENTATION AND RESULT** 38](#_Toc176102422)

[4.0 Overview 38](#_Toc176102423)

[4.1 Flowchart for the System 39](#_Toc176102428)

[4.2.0 Construction 40](#_Toc176102429)

[4.2.0.1 Frontend coding 40](#_Toc176102430)

[4.2.0.2 Backend Coding 41](#_Toc176102431)

[4.2.1 Desktop App 42](#_Toc176102432)

[4.3 Testing 48](#_Toc176102433)

[4.3.1 Tools Used in Testing 48](#_Toc176102434)

[4.3.2 Reasons for Testing 48](#_Toc176102435)

[4.4 Results 49](#_Toc176102436)

[4.4.1 Results for the System 49](#_Toc176102437)

[**CHAPTER 5** 52](#_Toc176102494)

[**FINDINGS AND CONCLUSION** 52](#_Toc176102495)

[5.0 Overview 52](#_Toc176102496)

[5.1 Findings 52](#_Toc176102497)

[5.2 Conclusion 53](#_Toc176102498)

[5.3 Limitations of the System 53](#_Toc176102499)

[5.4 Lessons Learned 53](#_Toc176102500)

[5.5 Recommendations for Future Work 54](#_Toc176102501)

[5.6 Recommendations for Commercialization 54](#_Toc176102502)

[5.7 References 55](#_Toc176102503)

# **CHAPTER ONE**

# **Introduction**

## overview

Keeping computer networks and systems secure is a matter of huge significance in the current digitalized landscape. Organizations face constant risk from data breaches, service disruptions and other forms of cyberattacks as the threat vectors continue to increase in complexity and frequency. While traditional security measures are necessary, they fall short when it comes to detecting and responding effectively against these increasingly sophisticated threats. This void is filled through Intrusion Detection Systems (IDS), which are considered to be a quintessential addition in the cybersecurity setup.

In a world of computing information is very important and the web is one of the main tools for sharing information. Like Tim Berners-Lee said “*The web is more a social creation than a technical one. I designed the web to be a tool for sharing information. It is for the benefit of everyone.*” As much as information is important it can equally be as dangerous. It can make and unmake an organization. “*Data is a double-edged sword. It can be used to improve our lives, but it can also be used to control and manipulate us. The same data that makes our lives easier can be used to invade our privacy and undermine our freedom*.” Says Shoshana Zuboff. Hence information is very important and as there are thieves trying to steal from banks there are also thieves trying to steal information for personal gain. These thieves are what the mass call hackers. The original concept of a hacker is someone who applies his or her skills for defensive purposes. This has changed over the years due to the bad activities being done with it. The modern concept of hacking began in the 1950s. The first password was a response to the Compatible Time-Sharing System developed in the early 1960. The hacker community has grown since then. Viruses, worms and trojans started appearing in 1988. The modern definition of a hacker is someone with malicious intent who intrudes the privacy of a person or an

organization. A system called Intrusion Detection System was designed by Dorothy Denning and Peter Neuman 1984 to detect such attacks.

An intrusion detection systems (IDS) monitor system logs and network traffic to recognize malicious activities in computer networks. Evaluating and comparing IDSs with respect to their detection accuracies is thereby essential for their selection in specific use-cases. An IDS can continuously analyze data from different sources to recognize the possible threats in real time and inform administrators so that they perform adequate responses. However, an IDS can only be as effective as its capacity to correctly differentiate between normal and abnormal behavior, it has to stay up-to-date on new threats emerging in the market & it must respond rapidly upon detection of intrusion. An Intrusion Detection System can leverage both supervised and unsupervised machine learning model. Machine learning is a subset of artificial intelligence (AI), it involves the development of algorithms that allow computers to learn from and make decisions based on data. Two types of machine learning are supervised and unsupervised learning. Supervised learning is the process of training a model on labeled dataset. Meaning, each training is paired with an output label or response. An advantage of supervised learning is that it can achieve high accuracy when trained on sufficient labeled data. One disadvantage of supervised learning is that there is risk that the model may become too tailored to the training data. Unsupervised learning is the process of training a model on a dataset that contains no labeled data. An advantage unsupervised learning is that it is powerful for discovering hidden patterns or structures in a data. The term machine learning was coined in 1959 by Samuel Arthur, an IBM employee and pioneer in the field of computer gaming and artificial intelligence. When using machine learning models for an intrusion detection system, dataset is needed to train these models. A dataset is the large collection of related data that has been stored in a single file. A dataset that has labels in them is used for supervised learning but a dataset that does not have labels in them is used for unsupervised learning. An example of dataset used in supervised learning is the kdd dataset. The KDD data set is a well-known benchmark in the research of Intrusion Detection techniques. A lot of work is going on for the improvement of intrusion detection strategies while the research on the data used for training and testing the detection model is equally of prime concern because better data quality can improve offline intrusion detection. An example of dataset used in unsupervised learning is the iris dataset Both Host-Based and Network-Based Intrusion Detection System play important roles in protecting an organization’s infrastructure. While HIDS provides detailed insights into host-level activities, NIDS offers a broader perspective by monitoring network traffic. A comprehensive security strategy often involves using both HIDS and NIDS in conjunction, leveraging their respective strengths to provide protection against cyber-attacks.

This project presents a comprehensive Intrusion Detection System that combines both Host-Based Intrusion Detection System (HIDS) and Network-Based Intrusion Detection System (NIDS) components. The dual approach ensures that threats are detected not only within the network but also at the host level, providing a multi-layered defense against cyberattacks. The system also allows the administrator to block applications, usb devices and websites that can serve as a gateway for intrusions.

**Host-Based Intrusion Detection System and how it works**

The Host-based Intrusion Detection System or HIDS for short, is a security system that monitors and analyzes activities involving the internal of a computing system and events occurring within the system to identify signs that are suspect of suspicious activities, probable intrusions, or security policy violations. Contrary to Network-Based Intrusion Detection Systems, which monitor network traffic, HIDS operates on individual hosts or devices.

**How HIDS Works**:

1. **Monitoring System Activity**:

It performs continuous monitoring of events like the integrity of files, activity of processes, system calls, user logins, and network traffic emanating from the host itself. This would monitor critical system file changes, unauthorized configuration changes, or other suspicious user activities to identify whether an intrusion has taken place.

2**. Log Analysis**:

The system collects logs from all system components and analyzes them-application logs, OS logs, security logs-looking for patterns or behavior that seems out of the ordinary.

By doing so, the HIDS will pick up hints in these logs regarding malware or unauthorized access attempts or other forms of malicious activity.

3. **Signature-Based Detection**:

As in anti-virus products, HIDS can also leverage databases of known attack signatures to detect types of intrusions.

If any of those system activities match any known signature, HIDS will raise an alert that points out that there might be an incident regarding security.

4**. Anomaly-Based Detection**:

HIDS also uses anomaly detection techniques, usually establishing a baseline of normal system behavior. Events that don't fit in with this baseline are usually highlighted for potential threats.

For example, if a process starts using an unusually high level of CPU or memory, or if a user account is exhibiting unusual patterns of logins, an alert will be triggered by the HIDS.

**Benefits of HIDS:**

**Granular Monitoring**: HIDS provides a more specific, host-oriented monitoring capability and hence can be considered more efficient than other systems with respect to local threats not visible at the network level.

**Insider Threat Detection**: HIDS is well-suited to detect insider threats like unauthorized access or policy violation by legitimate users.

**Limitations of HIDS**:

**Resource-intensive**: Typically, HIDS are going to be very resource-intensive, especially on hosts that are performing well or have produced large volumes of log data.

**Limited scope:** Moreover, HIDS tends to focus on individual hosts; hence, its scope in detecting attacks across a couple of systems or at a network level is limited.

**False Positives**: Anomaly-based detection systems, if the baseline behavior is not well-defined, may result in false-positives that trigger unnecessary alerts.

**Network Based IDS and how it works**

Network-Based Intrusion Detection System is a security system designed to monitor and analyze suspicious activities that could potentially lead to intrusions or violations of security policy. The major difference between NIDS and HIDS is that the latter works on individual hosts while NIDS monitors the traffic across an entire network or segments of a network.

1. **Monitoring system activities:**

NIDS monitors all inbound and outbound traffic on the network in real time by inspecting the data packets as they traverse the network. It is usually positioned around key junctions of the network, near gateways, routers, or switches, to tap from and analyze traffic flowing from various sources.

1. **Packet Analysis**:

NIDS performs packet header and payload inspection for suspicious patterns, anomalies, or known attack signatures. It identifies malicious activities of port scanning, DoS, malware propagation, and unauthorized access attempts.

1. **Signature-Based Detection**:

Since the detection database of the NIDS is dependent on the database of attack signatures to identify the particular type of threats, it is essentially similar to antivirus software. When the network traffic matches the known signatures, NIDS notifies any security incident via alerts that it raises.

1. **Anomaly-Based Detection**:

NIDS solutions also deploy the technique of anomaly detection, in which a baseline of normal network behavior is built. Any deviation from that is highlighted for potential threats. Examples include unusual traffic patterns, such as sudden spikes of data transfer or unexpected connections to unfamiliar IP addresses-might indicate that a breach has occurred.

**Benefits of NIDS**:

**Wide Coverage**: NIDS is designed to monitor all network traffic; hence, it proves very efficient in the detection of attacks that are targeted at more than one host or at the entire network.

**Early Detection**: Monitoring network traffic, NIDS is able to identify an attack before it actually reaches the host; hence, this acts as an early warning mechanism for impending dangers.

**Passive**: NIDS is passively listened to; thus, it does not interfere with the normal flow of the network. Therefore, it fits well for monitoring high-traffic environments.

**Limitations of NIDS**

**Encrypted Traffic**: Due to general native support lacking for decryption, encrypted traffic on NIDS is highly problematic to analyze. This may decrease efficiency in detecting threats hidden in secure connections, such as HTTPS and VPNs.

**False Positives**: Due to anomaly-based detection methods, there can be unnecessary alerts when the baseline behavior is not well defined.

**Limited Host-Level Visibility**: Unlike HIDS, NIDS does not have explicit visibility into the internal processes of the hosts, thus limiting its functionality of detecting some kinds of attacks, including insider threats or local exploits.

**Random forest**

Random Forest is thus an adaptable, strong machine learning algorithm used by almost any data scientist, who has applied it to various classification and regression purposes. This comes under the ensemble learning category, where it builds several decision trees and then combines them for more proper accuracy in its prediction and regulating overfitting of data.

**Benefits:**

**Robustness**: Random Forest, being a committee, has less chance of overfitting compared to its base decision trees alone, when large datasets are available for training.

**High accuracy**: It generally provides a very close and accurate prediction by the aggregation of results over several trees.

**Handles Missing Data**: Random Forest has a good way of handling missing values using surrogate splits.

**The limitations include**:

**Complexity**: Much more complex and less interpretable than a single decision tree.

**The computational cost**: usually making a large number of trees can be very expensive in computational cost when big data.

**Autoencoders**

Autoencoder is a type of neural network that performs unsupervised learning from data without labeled output. This is usually used in anomaly detection. There are three types of autoencoder. Encoder, bottleneck and decoder.

* **Encoder:** Compresses the input data into a lower-dimensional representation.
* **Bottleneck:** The compressed, lower-dimensional representation of the data.
* **Decoder:** Reconstruct the original data from compressed representation. The goal is to produce an output close to the original.

**Benefits**:

* **Unsupervised Learning**: Autoencoders do not use labeled data, making them useful for a wide range of tasks where labeled data is scarce.
* **Versatility**: They can be used for various purposes, including compression, anomaly detection, and generative modeling.

**Limitations**:

* **Reconstruction Bias**: Autoencoders may struggle to generalize if the input data has a high degree of variability or complexity.

## Problem Statement

In today's digital world, companies are facing increasing threats from cyber attackers who want to get illegal access to their systems and steal important information. Simply relying on traditional defenses around the perimeter of a network isn't enough anymore because these attackers are using more advanced tactics. As their methods become more diverse and complicated, there's a pressing need for a system called an Intrusion Detection System (IDS) that can adapt quickly to new threats.

Right now, many security measures aren't agile enough to catch small signs of trouble or new attack patterns. Also, with so much data flowing through a company's networks, it's hard to tell what's normal and what's not, making it easy for attackers to go unnoticed. Without a good IDS, companies are at risk of having their data stolen, their services disrupted, or unauthorized people getting into their systems, which could put important information at risk.

## Project Aim

The primary objective of this project is to develop an excellent intrusion detection system (IDS) that strengthens an organization's cybersecurity. We are going to employ the use of both host-based and network-based intrusion detection system (hybrid IDS). This will help analyze anomaly detection and machine learning to accomplish this. Our goal is to ensure that this IDS is capable of detecting data breaches, illegal access, and other negative events in digital systems. Another aim is to avoid potential intrusions by blocking devices, application and website that an be a channel for intrusions. In the end, we want to provide enterprises with a robust defense system that safeguards their critical assets and fortifies their cybersecurity, even when evolving cyberthreats arise.

## Project Objectives

1**. Develop a Comprehensive Threat Model:**

- Identify and analyze potential cyber threats relevant to the organizational context.

- Create a detailed threat model encompassing known attack vectors and emerging risks.

2. **Data Collection and Preprocessing:**

- Acquire diverse datasets reflecting normal network behavior and various types of cyber threats.

- Implement robust preprocessing techniques to cleanse and structure the collected data for analysis.

3. **Implement Machine Learning Algorithms:**

- Design and implement machine learning models for both host and network-based intrusion detection.

- Design autoencoder model for unsupervised learning

- Design random forest model for supervised learning.

- Optimize algorithms to minimize false positives and false negatives while maximizing detection accuracy.

4. **Real-time Monitoring and Alerting**:

- Integrate the IDS with network monitoring tools to enable real-time analysis of network traffic.

- Implement a responsive alerting system to promptly notify administrators of potential security incidents.

5**. Generate Comprehensive Report**:

- Develop mechanisms for the IDS to generate report for both host-based and network-based intrusion detection system.

- Design a database that can store and generate reports on intrusions.

6. **Block Devices, Apps and Websites:**

- Create a mechanism to block dangerous application, usb devices and websites.

7. **User-Friendly Interface:**

- Design an intuitive user interface for monitoring and managing the IDS.

8. **Employ the use of behavior analysis:**

**-** The use of behavior analysis helps monitor users use of a system.

## Justification for The Project

The digital landscape is constantly evolving, with cyber threats becoming more sophisticated and diverse. An IDS is necessary to counter these threats effectively by continuously monitoring and identifying malicious activities. Relying solely on traditional perimeter defenses is no longer sufficient as attackers find ways to bypass these measures. An IDS provides an additional layer of defense by detecting intrusions that may have evaded traditional security measures. Organizations need to be able to respond to cyber threats as they occur, rather than after the damage has been done. An IDS offers real-time detection and response capabilities, allowing for immediate action to mitigate potential risks. With the increasing volume and variety of data being processed by organizations, it's crucial to ensure the confidentiality, integrity, and availability of this information. An IDS helps protect sensitive data and ensures compliance with regulatory requirements. By implementing an IDS, organizations demonstrate a commitment to maintaining a strong cybersecurity posture. It serves as a proactive defense mechanism, reducing the likelihood of successful cyber-attacks and minimizing the impact of potential breaches.

## Motivation for Undertaking the Project

The demand for cybersecurity professionals is at an all-time high, with organizations worldwide seeking expert who can secure their system. By undertaking this project, we gain experience in a critical area of cybersecurity. This project will also allow you to get a deep understanding of network security, threat detection and practical application of machine learning in cybersecurity.

With increasing reliance on digital infrastructure in every sector, cybersecurity threats are becoming more sophisticated and frequent. Organizations of all sizes face the risk of cyberattacks, such as data breaches, ransomware and denial-of-service attacks. Developing an intrusion detection system is crucial for detecting these threats before they cause significant damage. By working on an IDS project, we can contribute to the global effort to protect sensitive information.

## Scope of the Project

The scope of this project covers the application model of an intrusion detection system. The model of the intrusion detection system incorporates machine learning model (random forest), autoencoders and robust defensive mechanism. Our design is to ensure detection of intrusions in live network packets, system logs and application logs.

This project encompasses the development and implementation of an advanced Intrusion Detection System (IDS) with the following key areas of focus:

* Collection of system and application logs from host machines.
* Utilization of autoencoders to identify anomalies in host behavior by comparing real-time data with learned normal patterns.
* Implementation of automated response mechanisms, such as blocking suspicious applications or processes on the host.
* Monitoring and analyzing incoming and outgoing network packets across the network.
* Leveraging a Random Forest algorithm, trained on the KDD dataset, to classify traffic and detect anomalies.
* Integration of automated blocking of devices found to be involved in malicious activities.
* Blocking access to known malicious websites to prevent phishing, malware downloads, and data exfiltration attempts.
* Training autoencoders on normal host system behavior to establish a baseline.
* Real-time comparison of current system behavior against this baseline to detect anomalies.
* Training the Random Forest model using the KDD dataset for accurate classification of network traffic.
* Ongoing model refinement and adaptation to emerging threats based on new data.

## Project Limitations

Despite the use of advanced machine learning models, the IDS might still generate false positives or false negatives, as is common for any anomaly detection solution. High false-positive rate may lead to alert fatigue, causing the security team to disregard many notifications, missing actual threats and false negatives leading to undetected intrusions. The training data for the model is based only on known security incidents, and the IDS is not immediately effective against unknown threats. Thus, it would be necessary to update the model and training data, periodically retraining it with the newest data to provide timely detection of new threats. Machine learning models require resources for real-time analysis, including storage and CPU power. Performance degradation, slow response times, and crashes can result when these systems have low resources. On deployment, configuration, and management of the IDS are some technical expertise in machine learning models and network security. The organization will not find it easy to make full use of the system with a few technical personnel. This naturally decreases effectiveness.

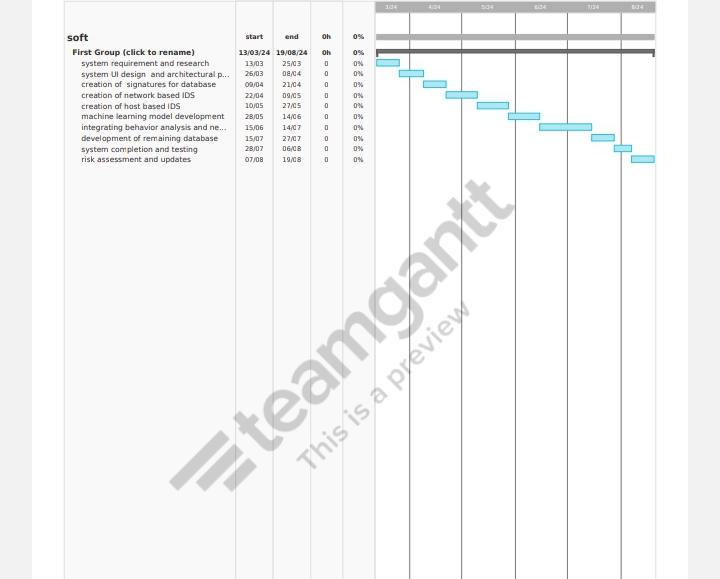
## Project Beneficiaries

Small to Medium Enterprises (SMEs) often have limited resources for cybersecurity. This IDS provides them with an affordable and automated solution to detect and respond to security threats, helping protect their sensitive data and maintain business continuity. Security teams benefit from the automated detection and response capabilities of the IDS, which reduces the manual workload of monitoring and analyzing potential threats. This allows them to focus on more strategic tasks, such as threat hunting and incident response. By protecting the network and host systems, the IDS helps prevent cyber-attacks that could disrupt daily work, such as ransomware that might lock them out of their computers. Employees can work in a more secure environment without worrying about phishing attacks or other cyber threats.

## Academic and Practical Relevance of the Project

As Alan Kay said, "*The best way to predict the future is to invent it.*" This IDS project will be directly practical in enhancing organizational security with a multi-layered defense against internal and external threats. It is the direct resolution to one of the most important needs in today's digital world, where cyber-attacks are becoming increasingly enhanced. This IDS can easily be used for case studies or teaching purposes in an academic environment to provide students with a better view of the usage of machine learning in cybersecurity. This project adds to the literature pool relating to machine learning techniques applied in cybersecurity. In the project, autoencoders and Random Forest algorithms have been integrated into intrusion detection to give a practical example of how these models can be used effectively to detect and prevent against cyber threats. *"Machine learning is a core, transformative way by which we're rethinking how we're doing everything. We're thinking about all the different ways it can be applied.*" — Sundar Pichai, CEO of Alphabet Inc. (Google)

## Project Activity Planning



## Structure of the Report

Five components make up this thesis. The first is an introduction that outlines the context of the research issue, the need for a study, the research question, the purpose, any limitations, and the format of this thesis. Since the entirety of the conduct of this study was based on a systematic review of prior studies, the literature review section (section two) was left to offer concepts and definitions that strengthen the general understanding of the important topics covered in depth in the result and discussion section (section four and five). The methods part follows as the third section. This section provided a comprehensive explanation of the stages involved in conducting a systematic review of the literature. The final section of this thesis, titled "Conclusion," presents conclusions derived from the result and discussion sections along with potential suggestions for future research topics.

## Project Deliverables

1**. System Architecture Documentation**: A detailed document with the overall architecture of the IDS comprising both host-based and network-based systems.

2. **Host-Based Intrusion Detection System-HIDS Module**

• **Description**: The functional HIDS module to monitor host activities and detect anomaly using autoencoders and then respond to the threats.

3. **Network-Based Intrusion Detection System-NIDS Module**

• **Description**: The operational NIDS module will sniff the network packets and send alerts in case of intrusion detection using the Random Forest model and perform mitigative action on the threats identified.

4. **Blocking Mechanism**

• **Description**: Integrated system which allows automatic blocking of suspected applications, devices, and websites detected from the anomalies and threats detected.

5. **Centralized Management Dashboard**

• **Description**: User-friendly interface to monitor the IDS, alarms, and manage the overall system.

6. **Training and Detection Models**

• **Description**: IDS machine learning models applied to detect anomalies and intrusion • **Contents**:

o Trained autoencoder model for the HIDS

o Trained Random Forest model for the NIDS

o Scripts that can be used to retrain these models with new data

o Walkthrough of the models, how they have been trained, and how they will be updated

7. **Final Project Report**

• **Description:** Comprehensive report that sums up the entire project, including its objectives, methodologies, outcomes, and future recommendations.

# **CHAPTER TWO**

# **REVIEW OF RELATED SYSTEMS**



## Related Systems

Intrusion Detection Systems have evolved over the years, working out the continuously increasing problem and sophistication in cyber threats. The traditional IDS has faced serious challenges concerning detection accuracy and performance of the system. Among these, a number of works indeed have identified one major issue: the large number of false positives that paralyze the security teams with what is so-called alert fatigue, while the configuration of such systems to correctly identify malicious activity without degrading performance remains elusive.

For this reason, architectural design is a great concern in ascertaining the efficiency and reliability of an IDS. García-Teodoro et al. (2009) add that the nature of software and hardware architecture determines whether the system will be able to execute threat detection and data processing with huge volumes in real time. A well-designed architecture is essential for ensuring security and performance of an IDS.

Among various intrusion detection-related research papers and studies, two major approaches for intrusion detection have been derived:

**Signature-Based Detection:** This approach uses predefined patterns of known malicious behavior, called signatures, to identify threats. Signature-based IDSs, such as Snort, are very effective in detecting known types of attack but clearly cannot detect new or modified threats (Roesch, 1999).

**Anomaly-Based Detection:** It establishes a baseline of normal traffic and events. Everything that falls out of this baseline will be labeled as a potential threat. However, though anomaly-based systems can identify previously unseen threats, they tend to generate a large number of false positives (Laskov et al., 2005).

To overcome these weaknesses, several contemporary IDS solutions have integrated hybrid models that combine signature-based detection and anomaly-based detection. These balances the strengths and shortcomings, thus offering superior security coverage accordingly.

### Signature-Based Intrusion Detection Systems

Signature-based IDS for example, Snort and Suricata has been designed to maintain and update the database of attack signatures. They match the network traffic and host activities against such a database to identify and flag any potential danger.

**Advantages of signature-based intrusion detection**:

High accuracy for detection of known threats, with minimum false positives

Real-time alerting and rapid response time

**Disadvantages**:

It is confined to revealing only known threats and therefore is found ineffective against new ones or change in any of the previous.

Maintenance for updating the signature database for this kind is very high.

### Anomaly-Based Intrusion Detection Systems

IDS based on anomalies, such as Zeek, a security software previously known as Bro, detect potential perils by establishing a baseline of normal network or system activity and trigger alerts when something happens out of that normal baseline. These kinds of systems are pretty effective in identifying new types of threats which may not fall under any signature.

**Advantages:**

Ability to detect new and unknown threats.

Adaptive learning capabilities that improve detection as time progresses.

**Challenges:**

Large number of false positives caused by marking benign deviations as attackers is high.

They are difficult to configure and tune for accurate baselines.

### Hybrid Intrusion Detection Systems

Hybrid IDS solutions, such as OSSEC, combine the best of signature-based and anomaly-based methods into a more complete approach to detecting threats. In the end, such systems aim to find both known and unknown threats by the blending of multiple detection methods.

**Strengths:**

Improve coverage of threats, reducing the chances of failing to capture any attack. Lower false-positive rates compared to systems whose underpinnings are completely based on anomalies.

**Disadvantages:**

The increasing system complexity will allow higher resource consumption and complicate maintenance.

The ability to produce multiple entries for the same threat is another challenge related to threat analysis.

### Machine Learning-Based Intrusion Detection Systems

In the recent years, there have been several IDS developed which based on machine learning algorithms, like the Random Forests, Support Vector Machines, and the Neural Networks for threats detection. These systems process large amounts of data by searching for patterns which can help them recognize the threat from potential malicious activities.

**Advantages:**

It also scales for large and complex data inputs.

Ability to generalize and ability to adapt to new threats by retraining the models.

**Challenges:**

High dependence on the quality and quantity of the training data may not guarantee the detection of a new threat.

High accuracy models are more complex, usually masking the reasoning that led to the alarming decision.

### Host-Based vs Network-Based Intrusion Detection Systems

HIDS solutions monitor host-based events for malicious activity, unauthorized file access, or system configuration changes. NIDS solutions, on the other hand, monitor network traffic for threat detection processes across the entire network.

**Advantages of HIDS**

Since HIDS is host-based, it clearly gives in-depth auditing and monitoring of individual hosts, thus the possibility to detect insider threats. This may take into account file integrity and system configurations.

**Cons of HIDS**

The system is very narrow because it will be confined to individual systems only. In most cases, it highly consumes systems resources and degrades system performance.

**Strengths of NIDS:**

General coverage that allows it to look at network traffic to detect outside threats.

This makes it scalable and allows for broad coverage across large, distributed networks.

**Challenges of NIDS:**

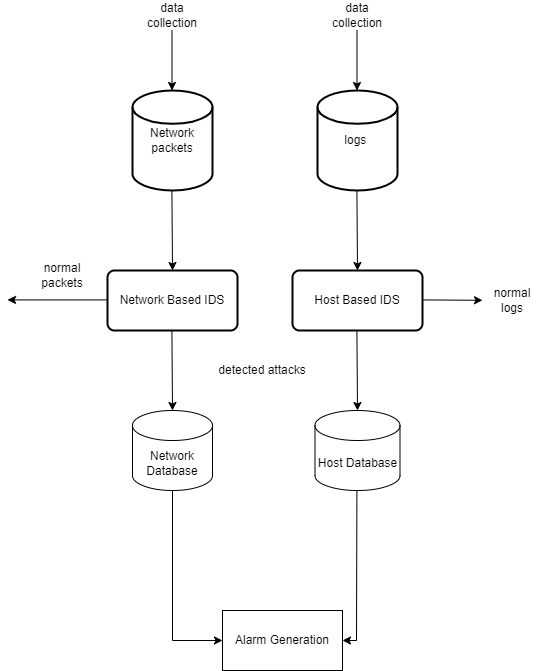
It's basically not efficient in terms of analyzing encrypted traffic.

If a professional attacker is at play, it can use evasion techniques to avoid detection.

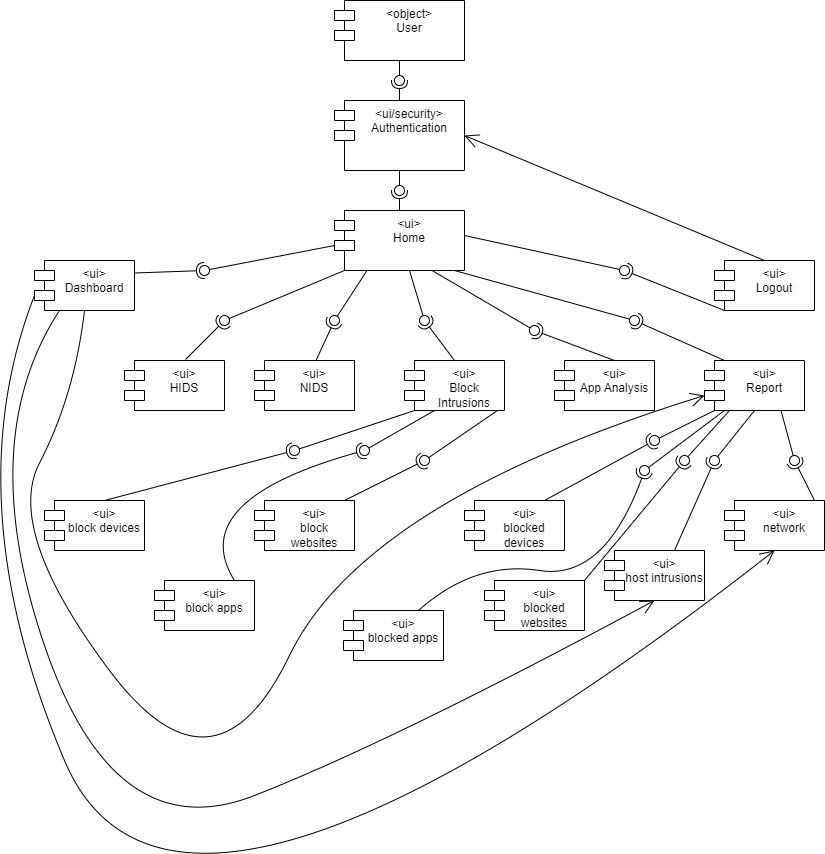
## Proposed System

Our project introduces a hybrid IDS as a solution, making it possible to deploy both host-based and network-based detection. These models use extension and provisions of machine learning models for more accurate means of detection with very minimal false-positives that cause violations on the user's privacy. The proposed system also provides defensive mechanism to prevent against potential intrusions.

## Architecture of the Proposed system

****

## Components Design and Components Descriptions

****

• **User Object**: The user object are people who interact with the system. Also known as administrators.

• **Authentication**: The authentication is a UI security that makes sure that users who access the platform are who they claim to be.

• **Home**: The home displays the various buttons and navigations for routing the application from one interface to another.

**• Dashboard**: The app's dashboard page gives you an overview of the intrusions detected in the network and blocked mechanism.

• **HIDS**: This side is responsible for detecting intrusions in a host-based system.

• **NIDS**: This page is responsible for detecting intrusions in a network-based system.

• **Block Intrusions**: This page is responsible for blocking applications, devices and websites that can be a gateway for intrusions.

• **App analysis**: This page is responsible for monitoring running applications.

* **Report:** This is page displays detected intrusions and blocked devices from the database
* **Logout:** This logs the user out of the system and takes you to the login page.
* **Block Devices:** This page displays a list of connected USB devices and provides the option to block devices.
* **Block Apps:** This page displays a list of applications and provides the option to block potential apps that could be a threat
* **Block Websites:** This page allows administrator to block harmful websites.
* **Network Intrusions:** This page displays detected intrusions in a network.
* **Host Intrusions:** This page displays detected intrusion in a host system.
* **Blocked Devices:** This tab displays blocked devices.
* **Blocked Apps:** This tab displays blocked apps.
* **Blocked Website:** This tab displays blocked websites.

## **Development Tools and Environment**

* **Python:** The primary programming language used for the development of the IDS. Python is chosen for its simplicity, extensive libraries, and strong support for data analysis and machine learning.
* **Scikit-learn:** A robust machine learning library in Python used for implementing the Random Forest model for network-based intrusion detection. Scikit-learn provides efficient tools for data mining and data analysis.
* **TensorFlow/Keras:** These libraries are used for building and training the autoencoder model for host-based anomaly detection. TensorFlow and Keras offer high-level APIs for developing deep learning models.
* **Pandas and NumPy:** These libraries are essential for data manipulation, preprocessing, and handling large datasets efficiently
* **Wireshark:** A network protocol analyzer used for capturing and analyzing network traffic, which helps in generating datasets for training and testing the IDS.
* **Visual Studio Code (VS Code):** An alternative lightweight IDE, used for both Python development and working with other languages or scripts. It is highly extensible with a wide range of plugins.
* **PyQt:** This library is used for ui development of the system.
* **QtSql:** This helps in the development of the database system.
* **Winreg:** This library is essential in blocking applications.

# **CHAPTER 3**

# **METHODOLOGY**

## 3.0 Overview

This section describes the methodology that was used in the development of the proposed IDS. For this project, we have adopted an Agile development methodology for the development of this project. It enables us to adopt Agile methodology because it is flexible and accommodating toward changes in requirements, while the hybrid nature allows structured planning and execution.

The project tools used are: Python, generally used for machine learning and data processing; TensorFlow is also used, especially in implementing the autoencoders in the host-based IDS; and Scikit-learn is used in the Random Forest model for the network-based IDS.

## 3.1 Requirement Specification

The Software Requirement Specification (SRS) is a formal document that depicts the functional, performance, and security requirements of the software. It presents the basis with which the various stakeholders can refer for the system to meet the requirements in terms of needs and expectations of the stakeholders. SRS can include things like intrusion detection, alerting functional requirements, and non-functional requirements such as scalability and performance of a system.

## 3.2 Stakeholders of the System

The following are stakeholders who would be involved in the development and deployment of an IDS, including:  
• **System Administrators**: Operate the deployment and maintenance of the IDS to ensure that it is reliable and performs well.  
• **Security Analysts**: Utilize the IDS in monitoring, detecting, and responding to security threats within the network.  
**• IT Management**: Supervise the integration of the IDS with existing IT infrastructure and make sure that it aligns with the organizational security policies.

• **Compliance Officers:** Ensure the IDS is adhering to the various industry and regulatory standards. This decreases the chance of several legal issues and compliance issues.

**• Developers**: They are to design the IDS by specifying what shall be coded and tested in order for it to meet given specifications and be resistant against security threats.  
• **Project Manager**: Heads the development process to ensure timelines are met and that the project stays within scope and budget.

## 3.3 Requirement Gathering Process

The process of gathering the requirements of the IDS was iterative, hence it included steps that could accommodate all kinds of needs. The most important steps are described as follows:

• **Stakeholder Identification**: Security analysts, IT managers, and compliance officers were involved in stakeholder identification since it had to do with various needs and concerns about intrusion detection.

• **Objective Definition:** This defined concrete objectives of the IDS, such as a reduction in false positives, improvement in accuracy of detection, and scalability of the system.

• **Requirement Elicitation:** Workshops, interviews, and questionnaires were conducted to develop elicited requirements from stakeholders. This included what technological and business needs the IDS must meet.

• **Documentation:** While gathering requirements, they were documented in some structured format that was easy to review and update. The documentation included use cases, user stories, and systems diagrams for clarity on what the system is supposed to do.

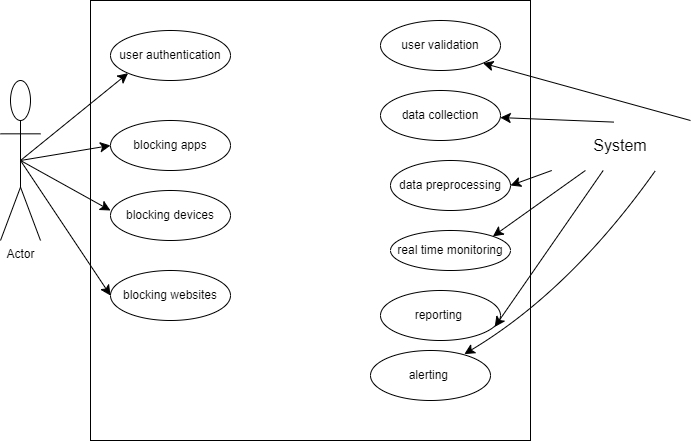
•**Validation:** Regular review sessions were conducted with stakeholders to validate the documented requirements. Any ambiguities or misunderstandings were spotted, and requirements were fine-tuned according to needs.

•**Prioritization:** The prioritization of requirements was made concerning their importance for the core functionality of the system and the overall goal of the project. This ensures that most crucial features will be developed and tested first.

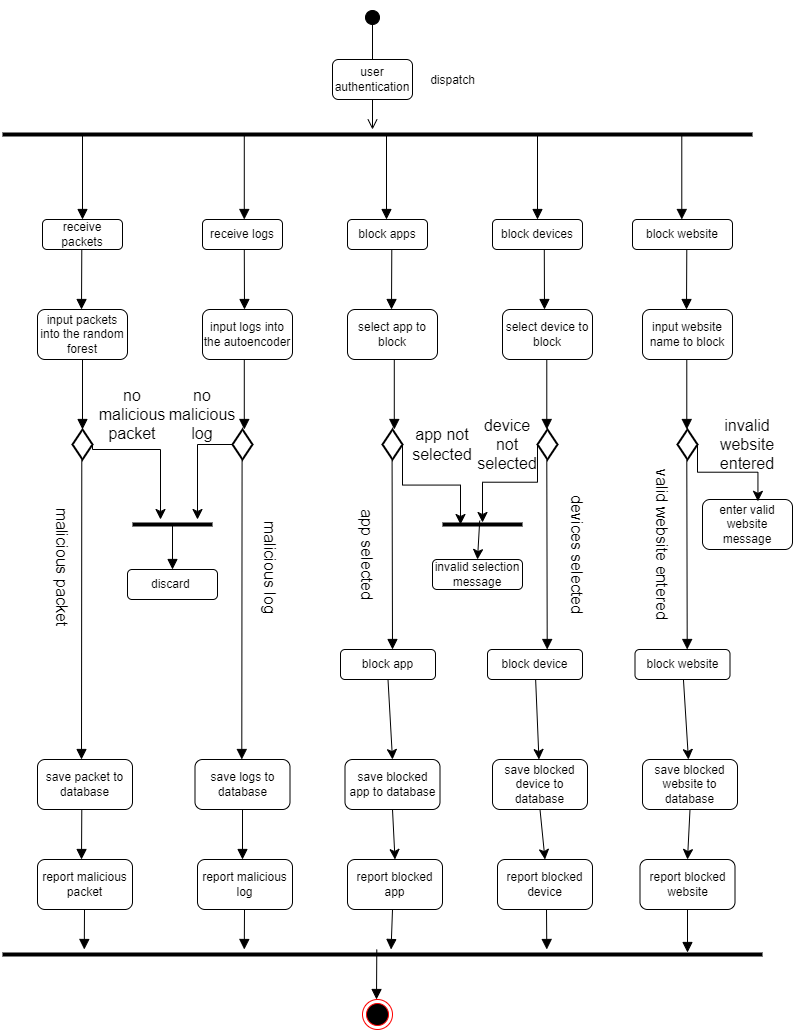
## 3.4 Functional Requirements

* The system will monitor and analyze system logs, application logs and network packets to detect suspicious activities.
* The system will use autoencoder and random forest model to identify anomalies.
* The system will generate alert when an anomaly is detected.
* The system will provide reports on detected threats.
* The system will block application, devices and websites that can be a gateway for intrusions.
* The system will process data in real-time to ensure timely detection and response to threats.

## 3.5 UML Diagrams

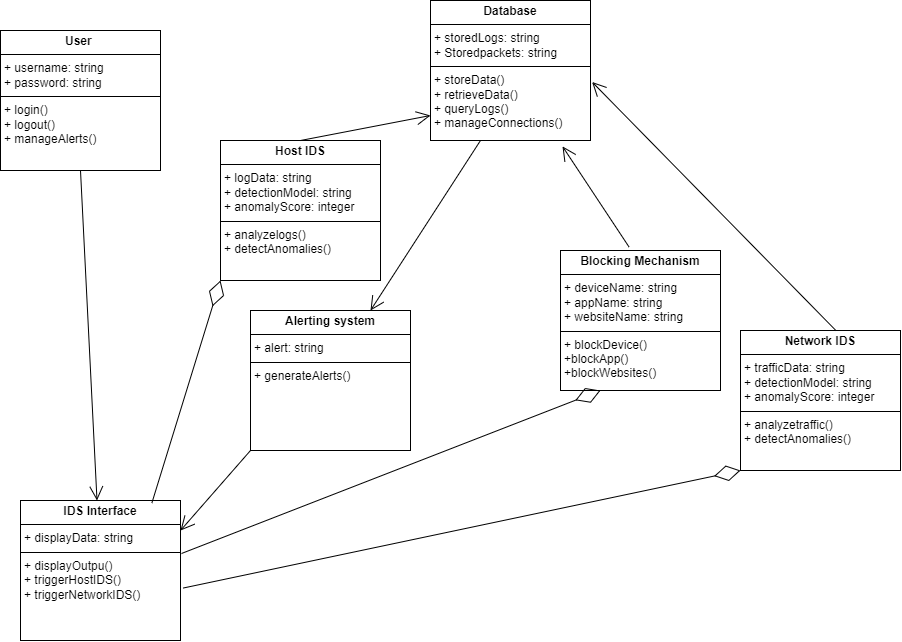
3.5.1 Use Case diagram****

### 3.5.2 Activity Diagram

****

3.5.3 Sequence Diagram****

### 3.5.4 Class Diagram

****

## 3.6 Non-Functional Requirements

* The system's user interface must be intuitive and easy to navigate, with a consistent design across all modules.
* The dashboard must provide customizable views, allowing users to focus on relevant data and alerts.
* The system must detect and generate alerts for threats within 5 seconds of their occurrence.
* The user interface must load and display data within 3 seconds, even under high traffic conditions.

## 3.7 Security Concept

This system has integrated four important concepts of security with a view to offering protection that is not only robust but also reliable. These are:

**• Confidentiality**:

Confidentiality: The system ensures sensitive information is only accessible by those users and entities authorized to have access. In the IDS, this ensures that logs, alerts, and system configuration are tightly restricted for access, view, and modification of critical data to only authenticated and authorized personnel. Encryption is extended to both data in rest and transit against unauthorized access.

**• Authentication**:

Authentication helps in proving the identity of the user who wants to access the IDS. All users must be authenticated by the system before they can have access. In this way, only valid and legitimate users operate the system and access information, thus preventing unauthorized persons from manipulating sensitive information and system controls.

•**Data Integrity**:

Data integrity means consistency and accuracy of data over its whole life cycle. IDS protects logs, alerts, and all configuration files against corruption or tampering. Integrity checks like hashing are applied to the critical data, and all discrepancies detected are flagged for review to make sure that the integrity of information used by decision-makers is valid and unchanged.

## **3.8 Project Methodology**

**Overview:**  
The Agile methodology embraces iterative development whereby the requirements and the solution evolve through collaboration among cross-functional teams. This is quite fitting for the IDS project, as this will enable the continuous improvement and adaptation against new security threats and evolving user needs.

**Advantages:**

• Flexibility: Agile allows changes to be made even in later stages of the project, which is essential in such a dynamic field of cybersecurity.

• Making Sure Customer Collaboration: Continuous feedback is acquired from the stakeholders, with the final product serving the needs and expectations.

• Incremental Delivery: The IDS will be created in little bits of code that are manageable. It advocates for the earlier detection and smoothening out of issues.

The development of the IDS was divided into sprints, each with a specific concentration on one or more of the components that this system will comprise: a host-based detection system, a network-based detection system, and integration of the app, device, and website blocking mechanisms. After each sprint, a review was done to garner feedback, make necessary adjustments, and plan the next phase.

**Waterfall Methodology**:

This traditional methodology follows a linear, sequential approach judged to be too rigid for the dynamic and evolutionary process of IDS development.

**Scrum Methodology**:

Scrum is a subset of Agile, which was considered in the project due to its emphasis on small cross-functional teams and regular feedback loops. However, the broader flexibility of Agile, without the strict roles and ceremonies of Scrum, was better suited for this project.

**Lean Methodology**:

Lean is about maximizing value by minimizing waste; this has been considered, but Agile gave more insight because of the tools it would provide to handle complex and iterative development cycles.

**Rapid Application Development RAD Methodology**:

The idea of RAD that attracted was its principle of rapid prototyping and immediate feedback. Yet, Agile was broader in its way of managing the most complex and ever-evolving variation in requirements.

## **3.9 Various Software Process**

A software process can be defined as a set of organized activities comprising a pathway to developing a software product. These activities are mainly performed by software engineers and require many steps to follow in order for the software to be just like it was required. All software processes contain all or most of these basic steps:

**• Software Specifications**:  
A clear demarcation of the capabilities and constraints of the software has to be done. The process generally consists of elaborating on the system to be developed, in addition to its functional and non-functional requirements. This means that for IDS, a description of the detection methodology, system architecture, and response mechanisms will have to be determined.  
  
 **• Software Development:**  
This is the production phase, where the software is developed to meet the required specifications. System design, coding, documentation, testing, and fixing of bugs take place in this phase. In the IDS project, it involved developing the autoencoder-based host-based detection system and the random forest-based network-based detection system, followed by integration with the blocking capabilities.

**• Software Validation:**Moreover, it makes sure that the software meets the customer's expectations. It involves testing the IDS to ensure that the requirements have been satisfied and the IDS can detect intrusions and respond to them accordingly as expected. Testing phases include unit testing, integration testing, and system testing. Attention will be paid to real situations of testing.  
  
**• Software Evolution:**This is where the software should constantly change and adjust according to the dynamic needs of customers and newly arising threats. After an initial development, improvements of IDS will be updated and refined based on feedback, new threat intelligence, and technological advances. This means that over time, the system will retain its effectiveness and relevance.  
( Software Processes - Javatpoint, n.d.) (Software Processes in Software Engineering - GeeksforGeeks, 2019)

## 3.10 Chosen Model and Justification

The Agile methodology was used for the development of the Intrusion Detection System. Agile is an iterative, flexible methodology that lays emphasis on teamwork, customer feedback, and quick adaptation to changes. It also goes well with the nature of the IDS project, as it needs to be adaptable with evolving security threats and ever-changing requirements.

**Agile Approach**:

**Specification:** Requirements are gathered and prioritized within short iterative cycles, called sprints. The result is continuous refinement and adaptation as new threats emerge, and feedback from stakeholders is received.

**Development**: The system will be developed in incremental steps, with each sprint delivering a functional component, say, host-based detection or network-based detection systems. In this way, each iteration of the system will improve upon the previous version, so that early detection of problems is possible, along with continuous improvement.

**Validation**: Testing and validation are included in every sprint so that every increment not only meets the set standards but also integrates well into the system as a whole. Continuous validation ensures security and reliability during the life of the system.

This agile methodology was chosen for this project because it allows for dynamic security requirements, it is inclusive of the stakeholders, and it allows for quick responses towards emerging threats.

# **CHAPTER 4**

# **IMPLEMENTATION AND RESULT**

## **4.0 Overview**

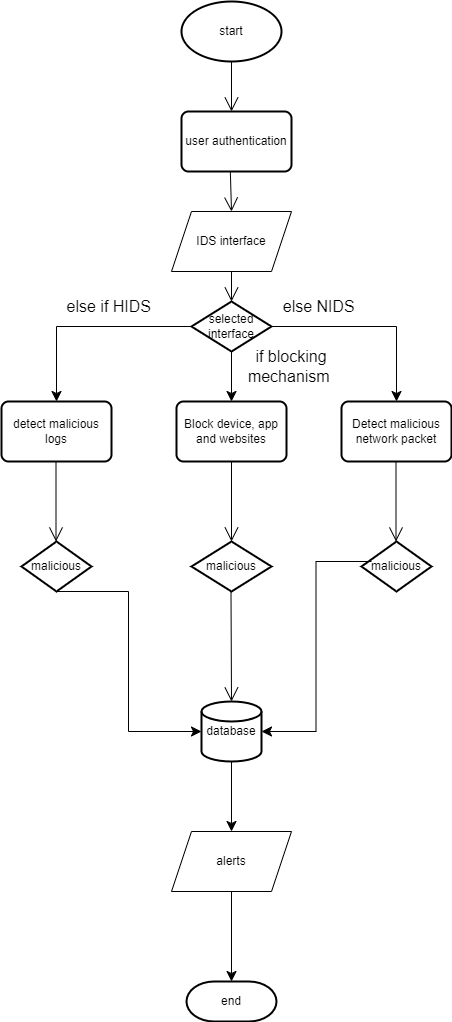
The system implementation and documentation is a very crucial stage in putting the IDS into practice. This is where the actual embodiment of the design takes place, whereby the conceived framework now gets transformed into an operational system. The planning for such actually began during the detailed design stage, where smooth transitioning from theory to practice needed to be ensured.

In this phase of the project, the IDS components will be installed, configured, and tested to ensure that the system performs as required. The focus of this activity is to deploy both a host-based and a network-based detection system to ensure they are integrated well and that their effectiveness in real situations has been assured. This is a very crucial stage in the process, given that the success of this entire project is pegged on ensuring the system has been put in place accurately and with effectiveness.

Due to the intrinsic underlying complexity of intrusion detection, great attention had to be paid to the minute details in setting up the implementation environment-right from hardware and software to optimizing detection algorithms and effective blocking of malicious applications, devices, and websites. The main goal of this presentation was to provide a robust, reliable, secure IDS that met all the specified requirements for the protection of a wide range of threats.

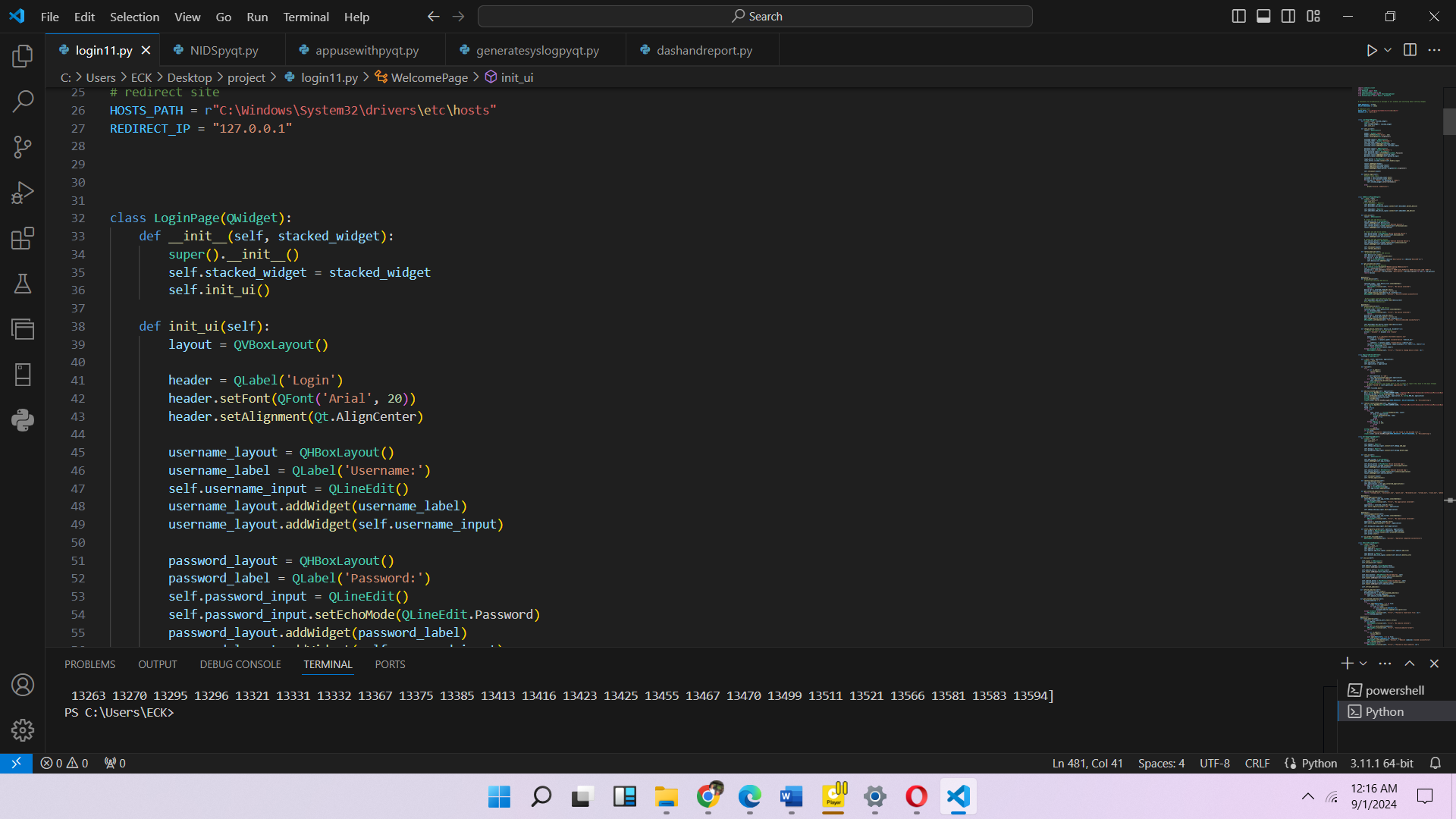
This is the most crucial phase of this project because the effectiveness of IDS in a live environment depends entirely on how robustly and accurately the implementation of the said system has been done. The system was fine-tuned for desired output using best practices and extensive testing.

## **4.1 Flowchart for the System**

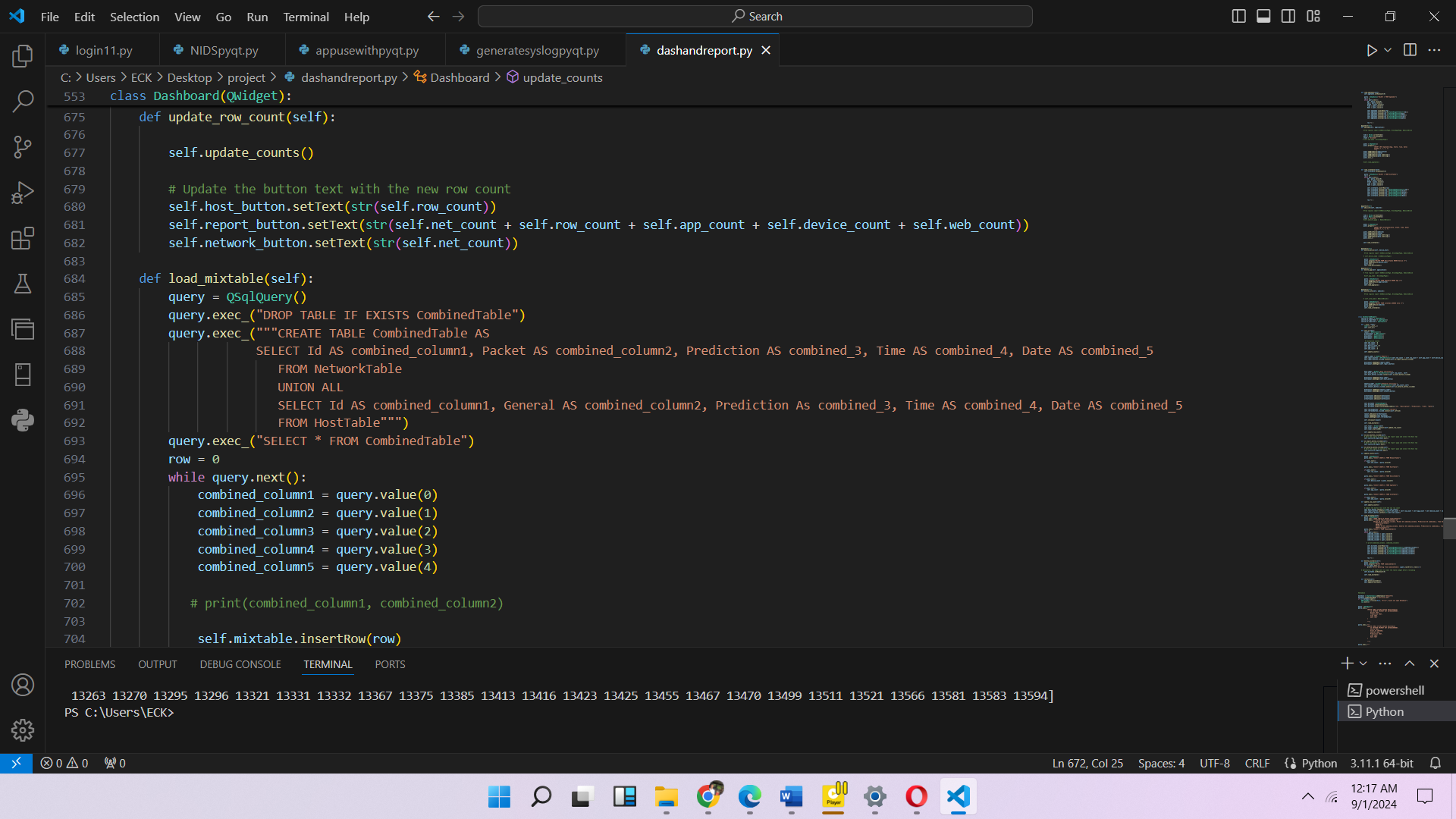
****

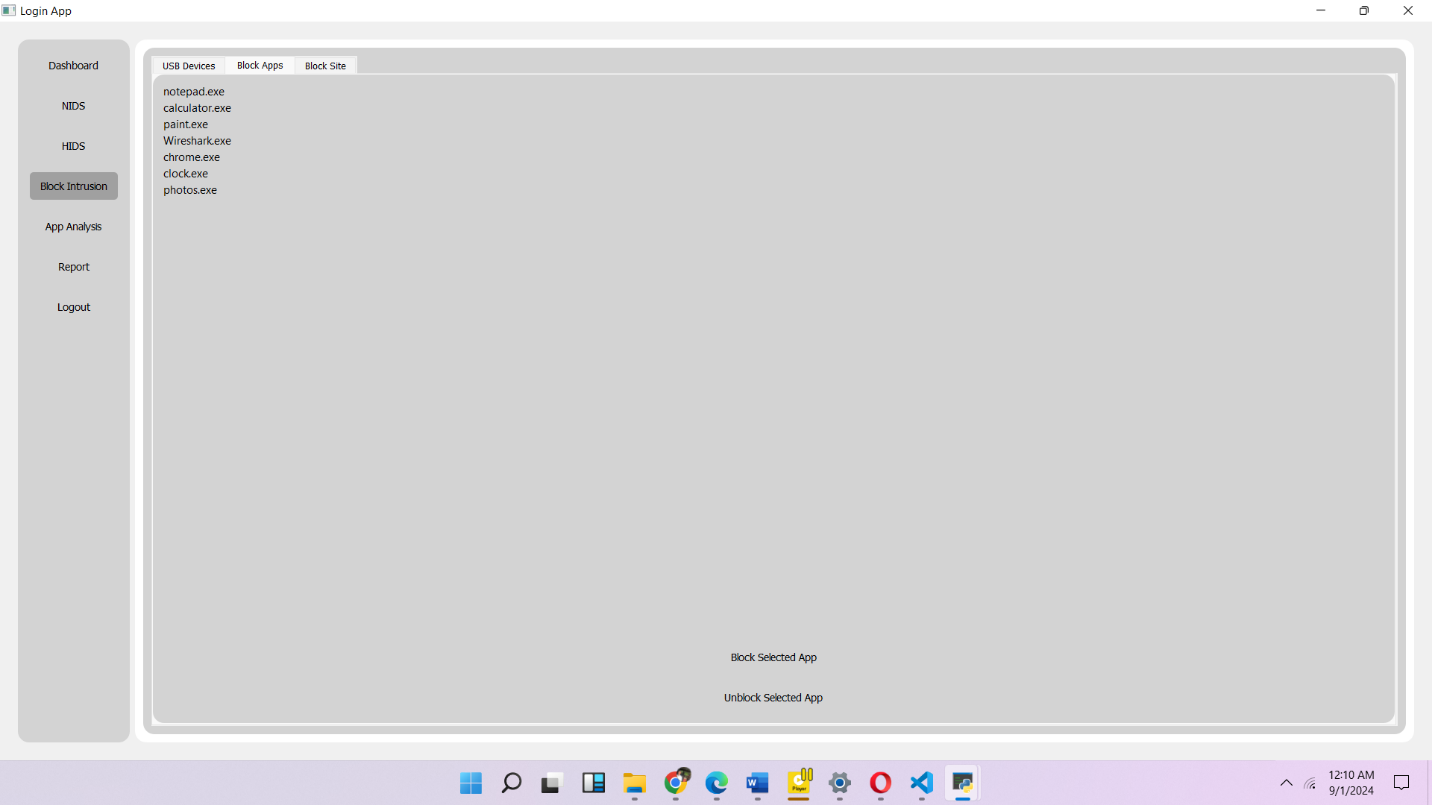
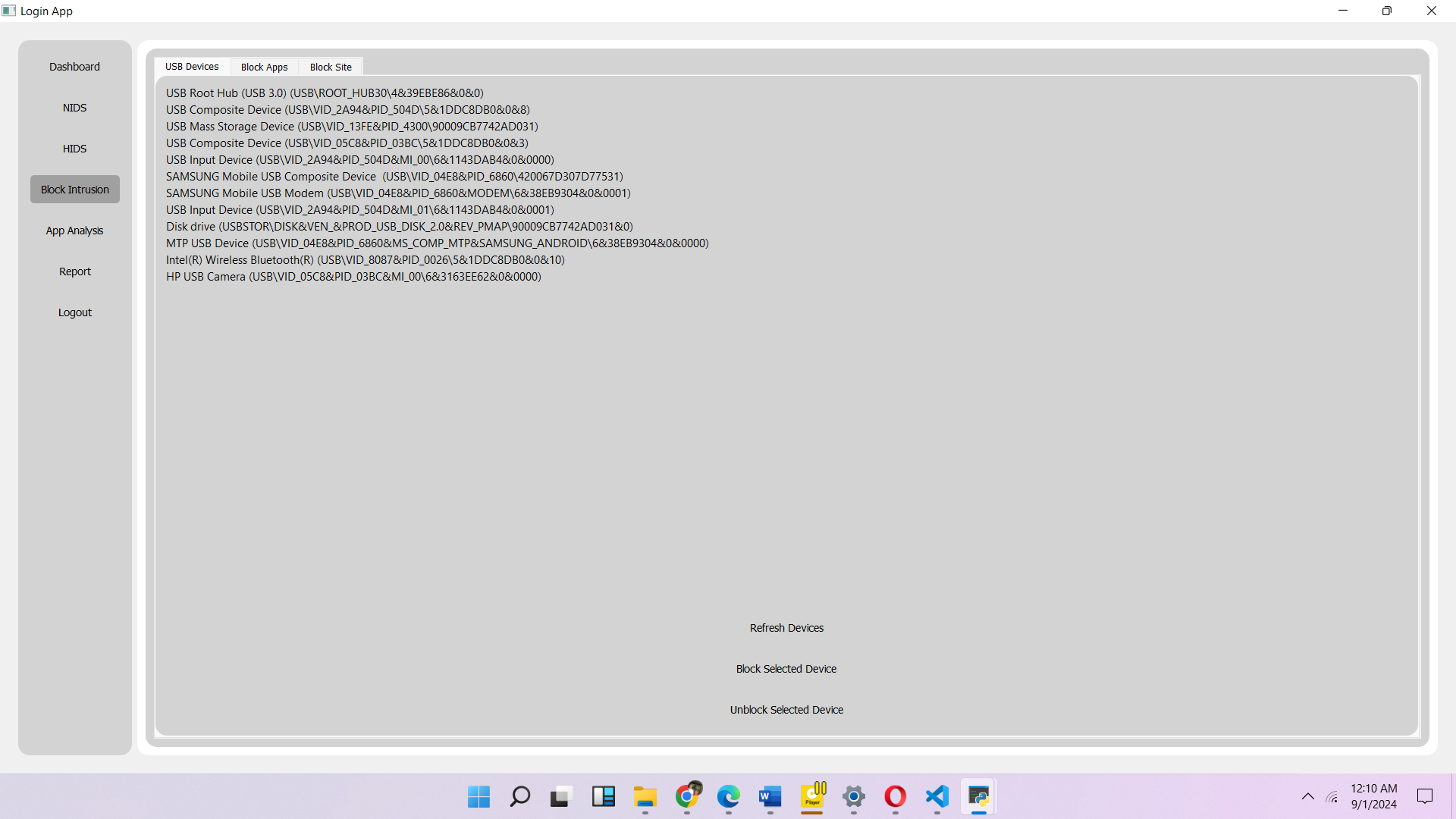
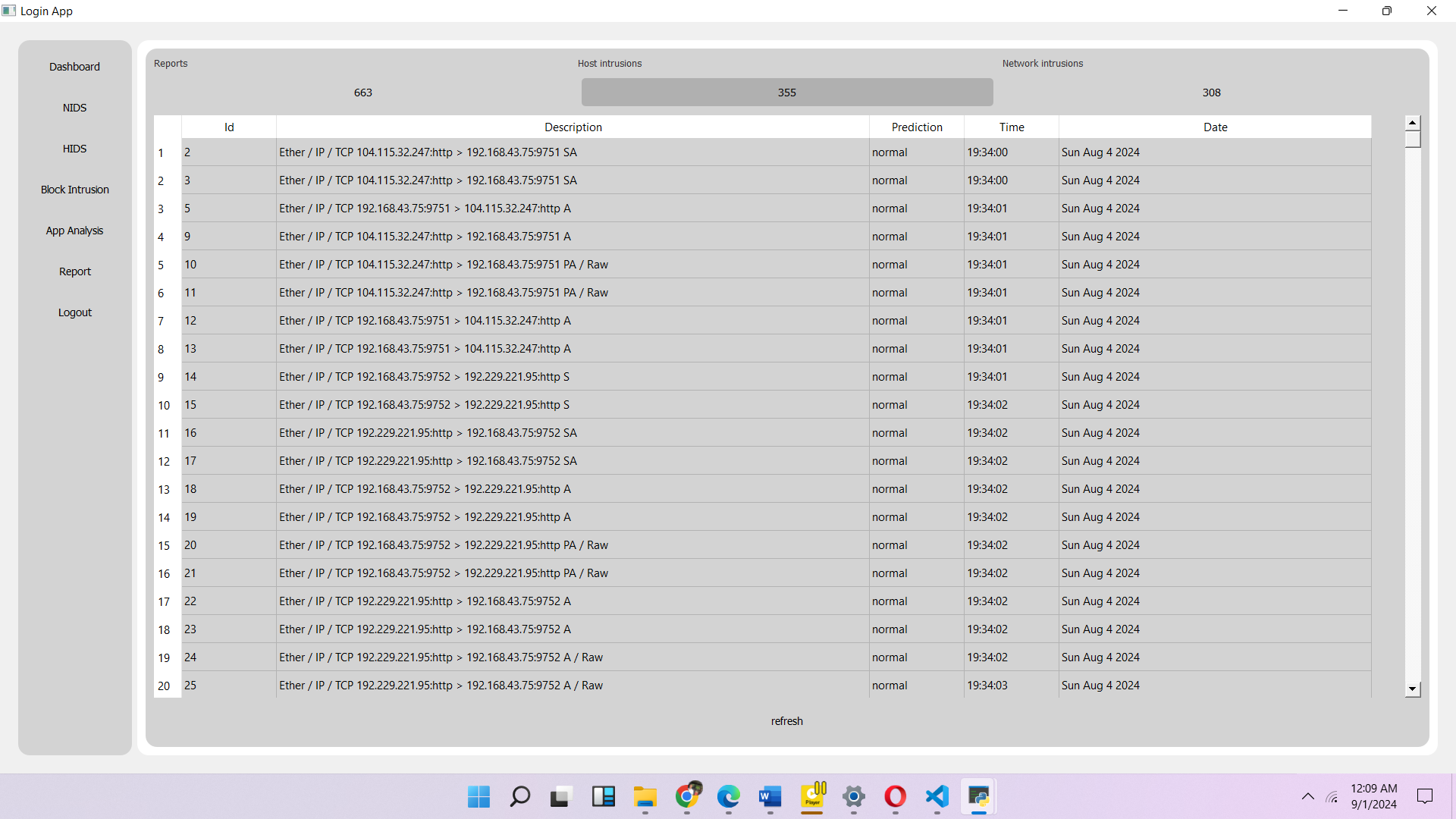
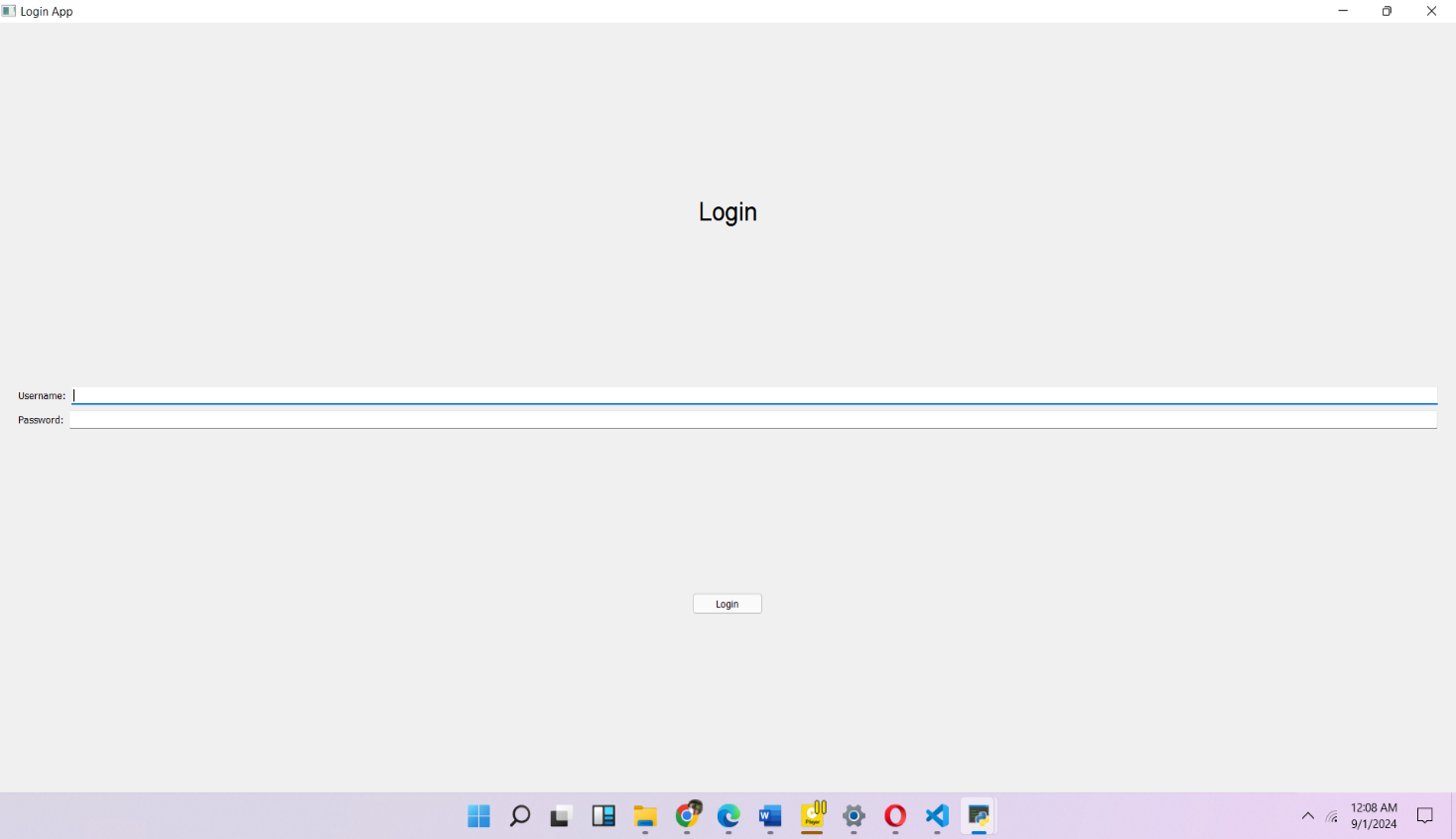
### **4.2.0 Construction**

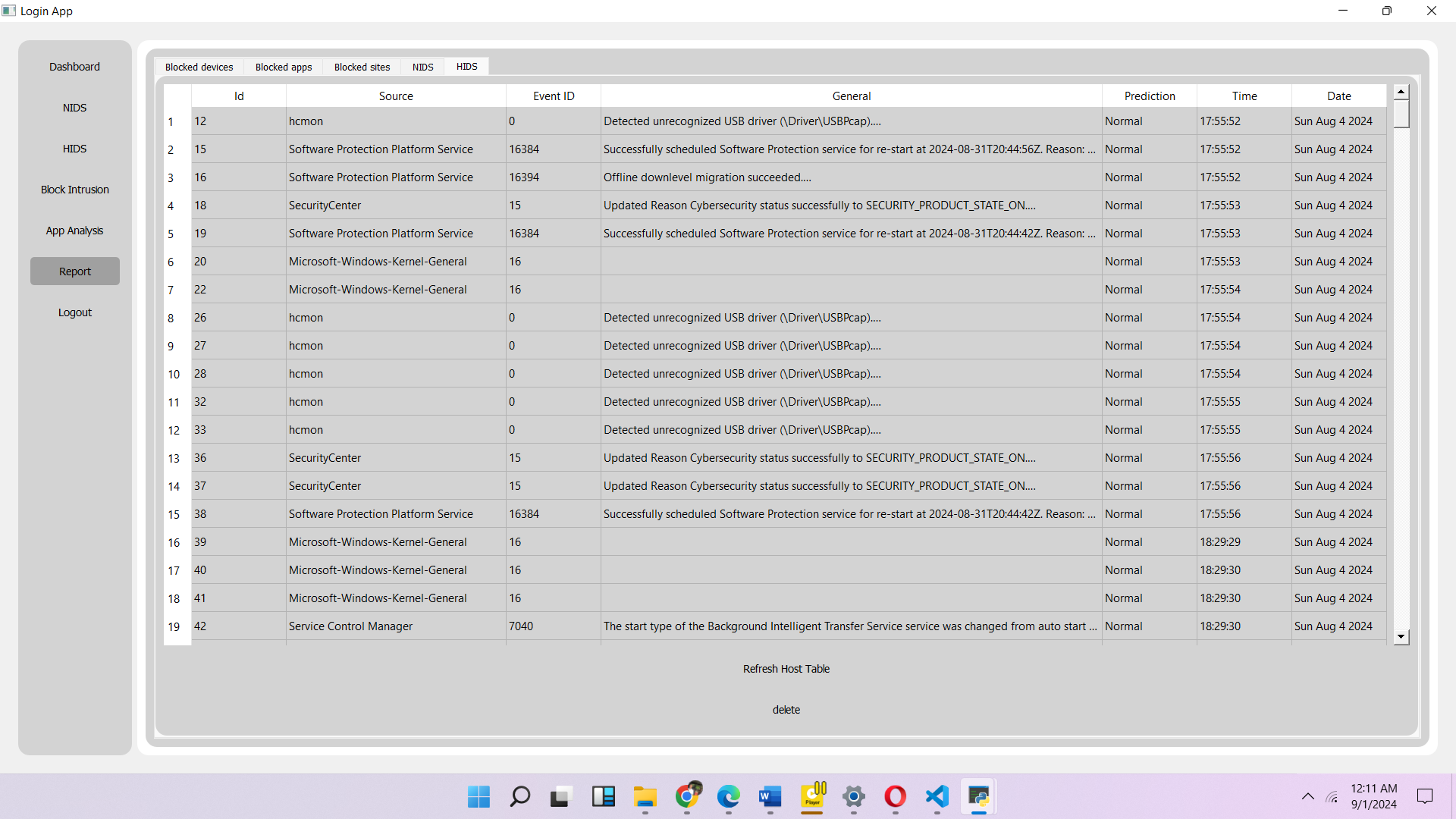
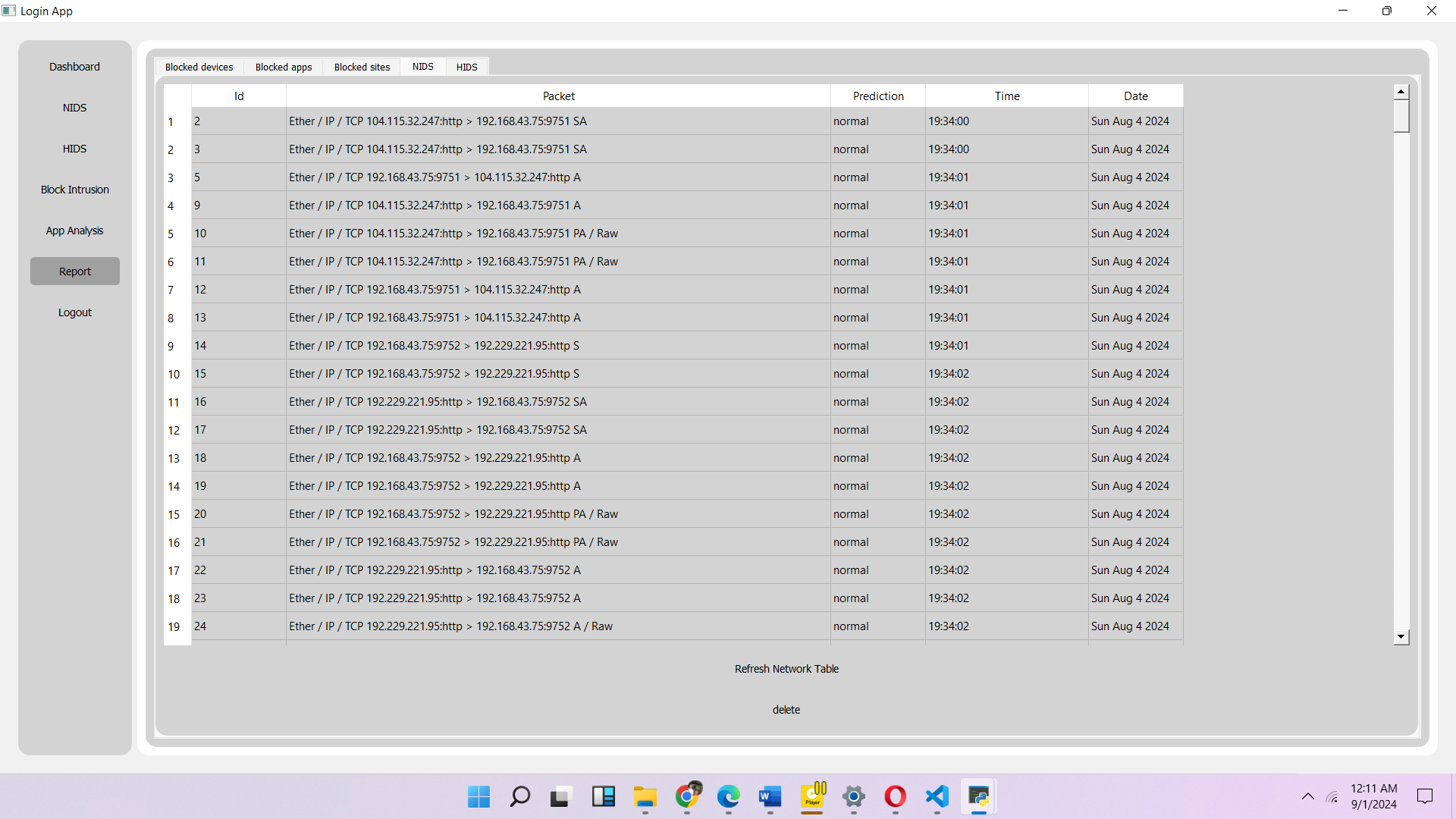
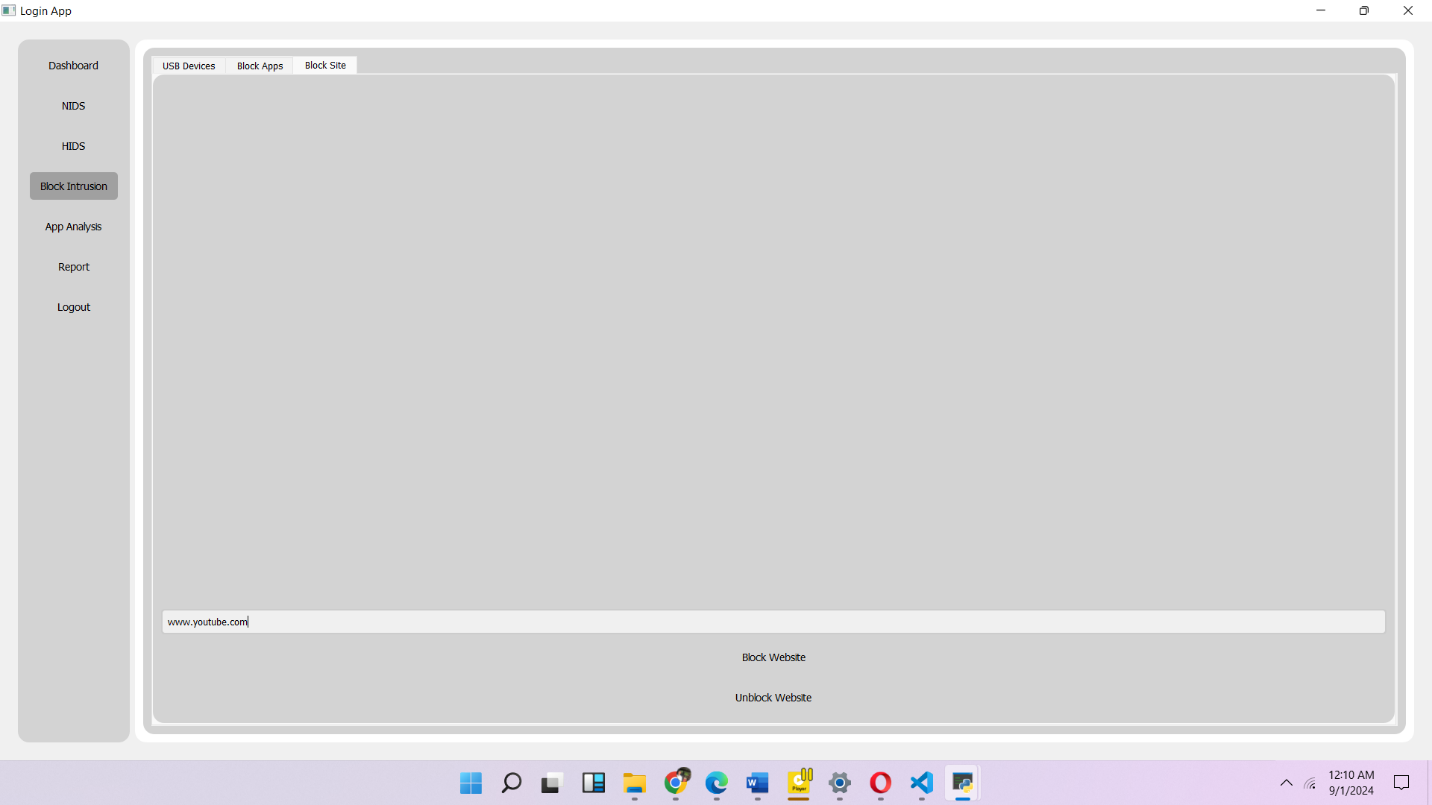
4.2.0.1 Frontend coding

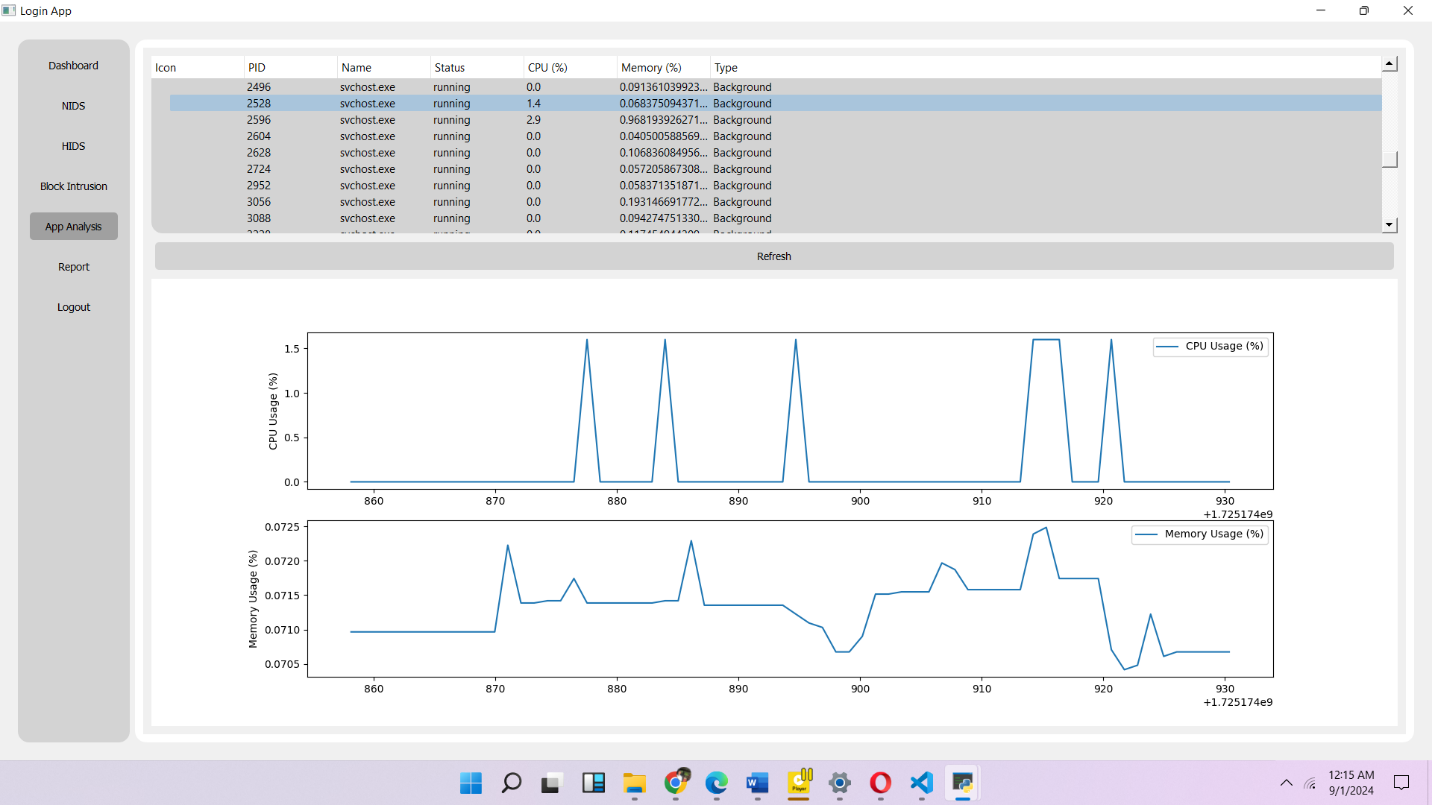


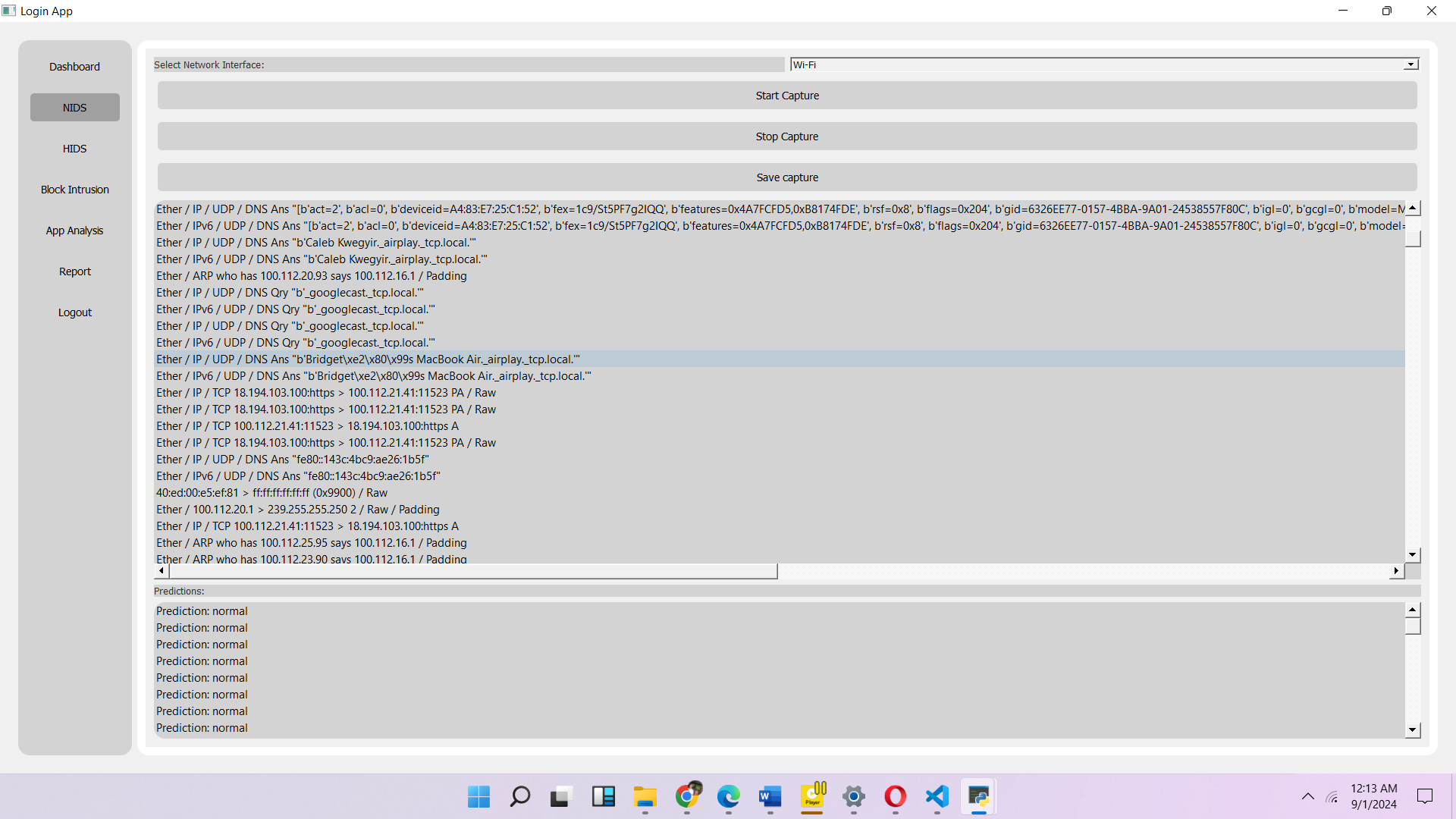
4.2.0.2 Backend Coding



4.2.1 Desktop App







## 4.3 Testing

Testing is a crucial phase in the development of the Intrusion Detection System (IDS), ensuring that both the host-based and network-based components function effectively and reliably. The testing process for the IDS can be broadly categorized into two main areas:

* **Host-Based Testing**: This involves verifying the accuracy and performance of the autoencoder models used to detect anomalies in system and application logs.
* **Network-Based Testing**: This involves evaluating the random forest model's ability to identify and classify network-based threats using the KDD dataset.

The following sections detail the testing strategies, tools used, and the results of the testing process for the IDS.

### 4.3.1 Tools Used in Testing

The tools and methods used for testing the IDS components include:

* **Wireshark**: Used for capturing and analyzing network traffic to ensure that the network-based IDS can accurately detect and respond to malicious activity.
* **Splunk**: Used for analyzing system and application logs, providing a platform to test the effectiveness of the host-based IDS in real-time.
* **Scikit-learn**: Used for implementing and testing the random forest model. Scikit-learn provides tools for cross-validation, feature importance analysis, and model performance evaluation.

### 4.3.2 Reasons for Testing

Testing the IDS was conducted for several critical reasons:

* **Ensuring Accuracy**: To confirm that the IDS accurately identifies threats without generating excessive false positives or false negatives.
* **Performance Validation**: To ensure that the IDS operates efficiently, providing real-time detection without causing significant system overhead or latency.
* **Reliability and Robustness**: To verify that the IDS remains stable and effective under various conditions, including high network traffic and diverse system loads.

## 4.4 Results

Extensive testing of the overall performance and effectiveness of the system was conducted after the implementation of the IDS. Testing of the system helped a lot in identifying and resolving issues; this was important in ensuring that the IDS met both user and system requirements.

Testing had two major components:

* **Unit Testing**: Each module was first tested separately to ensure correctness, such as the autoencoder trained for the host-based IDS and the random forest model developed for the network-based IDS. In such a way, it is warranted that each module can perform its task correctly, either anomaly or intrusion detection.
* **Integration Testing**: The components have been integrated based on the successful unit testing of components so as to test the overall system. This phase ensured that the host-based and network-based IDS components communicated effectively and provided an integrated and comprehensive detection system.

### **Results for the System**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO | Test Objective | Test Step | Expected Result | Results |
| 1. | To ensure the system lunches successfully | Run the code | Display of windows application | Passed |
| 2. | To ensure user authentication works correctly | Enter login credentials and click login button | Display of home screen or dashboard | passed |
| 3. | To ensure navigation buttons works correctly. | Toggle between each of the navigation tabs to switch from one screen to the other | There should be smooth transitions between screens | passed |
| 4. | To ensure Dashboard displays correct results | Check if table and number of reports displays correctly | There should be a table and three buttons displayed | Passed |
|  | To ensure blocking Mechanism works | Try blocking apps, devices and websites in the block intrusion page | Blocked apps, devices and websites cannot be opened | Passed |
|  | To ensure reporting page displays report correctly | Check if the report page displays all the right info | Should display information from the database. | Passed |
|  | To ensure Autoencoder is able to analyze logs | Train log dataset in autoencoder. | Must successfully train dataset | passed |
|  | To ensure that random forest is able to analyze packets | Train kdd dataset in random forest model | Must successfully train dataset | Passed |
|  | To ensure that the system is able to capture live packets | Click start capture in the NIDS page of the app | Should generate live packets | Passed |
|  | To ensure that the system is able to capture live logs | Click start capture in the HIDS page of the app | Should generate live logs | Passed |
|  | To ensure the system is able to predict live logs and packets | Check prediction side of the of either the NIDS or HIDS page for | Should display prediction as normal or malicious | passed |
|  | To ensure that database stores and retrieve data | Try blocking mechanism and check if the blocked item is displayed at the report side. | Should display blocked app, device or website at the report side. | Passed |

# **CHAPTER 5**

# **FINDINGS AND CONCLUSION**

## 5.0 Overview

The development and testing of the Intrusion Detection System (IDS) have been thoroughly documented to ensure flexibility, accessibility, and security. This chapter summarizes the key accomplishments and insights gained throughout the project.

* The IDS successfully integrates both host-based and network-based intrusion detection.
* Machine learning models, such as autoencoders and random forests, have been effectively implemented to detect anomalies in system logs and network traffic.
* The IDS includes functionalities for real-time detection, alerting, and blocking of malicious activities.
* A user-friendly interface allows administrators to manage and monitor the system efficiently.

## 5.1 Findings

Upon completion of the project study and implementation, the following findings were observed:

* **Awareness and Adoption**: While IDS technology is becoming more prevalent, some organizations are still unaware of its benefits (SmallBizEpp, n.d.).
* **Integration Challenges**: Some organizations face difficulties integrating IDS with their existing security infrastructure, particularly when dealing with legacy systems (Sophos, n.d.).
* **Scalability**: The IDS is scalable, but its effectiveness is highly dependent on the quality of data it receives. Ensuring consistent and high-quality data input is crucial for accurate detection (Sophos, n.d.).

## 5.2 Conclusion

Intrusion Detection Systems developed within the framework of this project have been one of the efficient ways not only to monitor the activity within computer systems but also to enhance cybersecurity. Since the system detects host-based and network-based threats, it proves to be a core device in the security structure of any network. This implementation has already shown that the right selection of machine learning models and system monitoring techniques can effectively achieve threat detection.

## 5.3 Limitations of the System

Despite the successful implementation, the IDS has some limitations:

* **Data Quality**: The accuracy of the IDS heavily relies on the quality of the data it analyzes. Poor or incomplete data can lead to false positives.
* **Integration Complexity**: Integrating the IDS with legacy systems or other security tools may require additional customization and expertise.
* **Resource Intensive**: Running the IDS, particularly the machine learning models, can be resource-intensive, which may pose a challenge for organizations with limited computational resources.
* **Real-time Performance**: While the system is designed for real-time detection, high network traffic can cause delays in processing, affecting its responsiveness.

## 5.4 Lessons Learned

* **Importance of Data Quality**: High-quality data is essential for the effective functioning of the IDS. Continuous monitoring and refinement of data inputs are necessary to maintain system accuracy.
* **Iterative Development**: Adopting an iterative approach to development and testing allows for continuous improvement and adaptation to new threats.
* **Testing and Validation**: Extensive testing is crucial to uncover hidden bugs and ensure that the IDS performs as expected.

## 5.5 Recommendations for Future Work

To further enhance the IDS, the following recommendations are made:

* **Enhanced Integration Capabilities**: Future versions of the IDS should focus on seamless integration with a broader range of security tools and platforms, including cloud-based services.
* **Advanced Machine Learning Models**: Incorporating more sophisticated machine learning algorithms, such as deep learning techniques, could improve detection accuracy.
* **User Training and Awareness**: Providing training and resources for users to better understand and utilize the IDS can improve its effectiveness and adoption.
* **Global Deployment**: Consider expanding the system’s applicability to different regions, taking into account the practices of the regional.

## 5.6 Recommendations for Commercialization

For the IDS to succeed in the commercial market, the following strategies should be considered:

* **Targeted Marketing**: Focus on organizations with high-security needs, such as financial institutions, healthcare providers, and government agencies, as primary customers.
* **Scalability and Flexibility**: Ensure that the IDS can be customized and scaled to meet the specific needs of different industries and organizational sizes.
* **Support and Training Services**: Since organizations will need to install and learn to use the IDS effectively, support packages should also be given that contain setup and customization and ongoing training.
* **Partnerships and Alliances**: Collaborate with other cybersecurity vendors to offer the IDS as part of a larger, integrated security solution.

## **5.7 References**

1*. Y. -j. Ou, Y. Lin, Y. Zhang and Y. -j. Ou, "The Design and Implementation of Host-Based Intrusion Detection System," 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, Jian, China, 2010, pp. 595-598, doi: 10.1109/IITSI.2010.127.*

2. *M. Landauer, F. Skopik, M. Frank, W. Hotwagner, M. Wurzenberger and A. Rauber, "Maintainable Log Datasets for Evaluation of Intrusion Detection Systems," in IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 4, pp. 3466-3482, 1 July-Aug. 2023, doi: 10.1109/TDSC.2022.3201582.*

3*. Kim, D.S., Lee, S.M., Park, J.S. (2006). Building Lightweight Intrusion Detection System Based on Random Forest. In: Wang, J., Yi, Z., Zurada, J.M., Lu, BL., Yin, H. (eds) Advances in Neural Networks - ISNN 2006. ISNN 2006. Lecture Notes in Computer Science, vol 3973. Springer, Berlin, Heidelberg.* [*https://doi.org/10.1007/11760191\_33*](https://doi.org/10.1007/11760191_33)

4. *Wang, C., Sun, Y., Wang, W., Liu, H., & Wang, B. (2023). Hybrid intrusion detection system based on combination of random forest and autoencoder. Symmetry, 15(3), 568.*

5. *Zhao, Z., & Liu, H. (2007, June). Spectral feature selection for supervised and unsupervised learning. In Proceedings of the 24th international conference on Machine learning (pp. 1151-1157).*

6. *Berry, Michael W., Azlinah Mohamed, and Bee Wah Yap, eds. Supervised and unsupervised learning for data science. Springer Nature, 2019.*

7*. Singh, Amrit Pal, and Manik Deep Singh. "Analysis of host-based and network-based intrusion detection system." International Journal of Computer Network and Information Security 6.8 (2014): 41-47.*

8*. Hu, Jiankun. "Host-based anomaly intrusion detection." Handbook of information and communication security. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. 235-255.*

9. *Aggarwal, Preeti, and Sudhir Kumar Sharma. "Analysis of KDD dataset attributes-class wise for intrusion detection." Procedia Computer Science 57 (2015): 842-851.*

10. *SmallBizEpp. (n.d.). Intrusion Detection Systems (IDS): Importance and Benefits. Retrieved from* [*https://smallbizepp.com/intrusion-detection-systems*](https://smallbizepp.com/intrusion-detection-systems)

11*. Sophos. (n.d.). What is Intrusion Detection System (IDS)? Retrieved from* [*https://www.sophos.com/en-us/what-is-intrusion-detection-system*](https://www.sophos.com/en-us/what-is-intrusion-detection-system)

12. *SpecterOps. "The Defender’s Guide to the Windows Registry." SpecterOps, 24 Mar. 2021,* [*https://specterops.io/post/the-defenders-guide-to-the-windows-registry*](https://specterops.io/post/the-defenders-guide-to-the-windows-registry)*.*

13. *Microsoft. (n.d.). PnPUtil command-line options. Microsoft Learn. Retrieved [date], from* [*https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/pnputil*](https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/pnputil)

14. *Freedom Matters. (n.d.). How to block websites that are distracting, harmful, and even dangerous. Freedom. Retrieved August 29, 2024, from https://freedom.to/blog/block-websites*

15*. Hochreiter, S., & Schmidhuber, J. (1997). "Long Short-Term Memory." Neural Computation, 9(8), 1735-1780.*

16. *Kim, K., & Kim, B. (2019). "A Robust Log-based Anomaly Detection Method for Generalized Machine Learning." Proceedings of the 2019 IEEE International Conference on Big Data and Smart Computing (BigComp), 205-208.*

17. *Zheng, A., Jin, B., & Wu, J. (2017). "Anomaly Detection in Logs Using Neural Network-based Approaches." Proceedings of the 26th International Conference on World Wide Web Companion (WWW '17 Companion), 983-987.*

18*. Le, T. D., & Zincir-Heywood, A. N. (2006)****.*** *"Evaluating machine learning based intrusion detection algorithms using ROC curves." Proceedings of the International Conference on Intelligence and Security Informatics, 186-198.*

19. *Zargar, S. T., Joshi, J., & Tipper, D. (2013). "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks." IEEE Communications Surveys & Tutorials, 15(4), 2046-2069.*

20. *Patil, P. D., & Thorat, S. A. (2016). "Blocking phishing websites using machine learning." Proceedings of the 2016 International Conference on Computing, Analytics and Security Trends (CAST), 462-467.*

21*. Agarwal, A., & Sharma, A. (2017). "A comprehensive survey of random forest algorithms in big data environment." Proceedings of the 7th International Conference on Cloud Computing, Data Science & Engineering, 106-112.*

22. *Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2014)****.*** *"Network anomaly detection: Methods, systems, and tools." IEEE Communications Surveys & Tutorials, 16(1), 303-336.*

23. *Breiman, L. (2001)****.*** *"Random forests." Machine Learning, 45(1), 5-32.*

24.*Kim, H., Song, J., Kim, S., & Kim, H. (2018)****.*** *"System log-based intrusion detection using deep learning." Security and Communication Networks, 2018, 1-12.*