Neurawall

By:

Muhammad Ubaid Ullah
32602
Umar Abdullah
33073
Hamza Gulzar
31621

Supervised by:

Mr. Osama Raza
Mr. Awais Nawaz

Faculty of Computing
Riphah International University, Islamabad
Spring 2025

Submitted To

Faculty of Computing,

Riphah International University, Islamabad

As a Partial Fulfillment of the Requirement for the Award of

the Degree of

Bachelor of Science in Cyber Security

Faculty of Computing
Riphah International University, Islamabad
May 14, 2025

Final Approval

This is to certify that we have read the Report submitted by ***Muhammad Ubaid Ullah (32602), Umar Abdullah (33073),*** *and* ***Hamza Gulzar (31621),*** for the partial fulfillment of the requirements for the degree of the Bachelor of Science in Cyber Security (BS CYS). It is our judgment that this Report is of sufficient standard to warrant its acceptance by Riphah International University, Islamabad for the degree of Bachelor of Science in Cyber Security (BS CYS).

Committee:

1

_____

Mr. Osama Raza
(Supervisor)

2

_____

Mr. Awais Nawaz
(Co-Supervisor)

3

_____

Dr. Jawaid Iqbal
(In-charge Cyber Security)

4

_____

Dr. Musharraf Ahmed
(Head of Department)

Declaration

We hereby declare that this document "**Neurawall**" neither as a whole nor as a part has been copied out from any source. It is further declared that we have done this project with the accompanied Report entirely based on our personal efforts, under the proficient guidance of our teachers especially our supervisors **Mr. Osama Raza** and **Mr. Awais Nawaz**. If any part of the system is proved to be copied out from any source or found to be a reproduction of any project from anywhere else, we shall stand by the consequences.

Muhammad Ubaid Ullah

32602

Umar Abdullah

33073

Hamza Gulzar

31621

Dedication

We dedicate this work to our families, whose unwavering support, love, and encouragement have been our foundation to our parents, for their guidance, patience, and belief in our potential; to our friends, for their understanding and constant motivation; and to our mentors, who inspired us with their knowledge, insight, and dedication.

And this book is dedicated to all those who share the desire for knowledge, innovative spirit, and excellence. Together we look forward to making a meaningful difference and contributing to a better and more secure world.

Acknowledgement

First, we are obliged to Allah Almighty the Merciful, the Beneficent and the source of all knowledge, for granting us the courage and knowledge to complete this project.

We would like to express our heartfelt gratitude to all those who supported us throughout the development of this project and its accompanying report titled **"Neurawall."**

First and foremost, we are deeply thankful to our teachers, especially our supervisor **Mr. Osama Raza** and **Mr. Awais Nawaz**, for their constant guidance, encouragement, and technical support. Their valuable insights and constructive feedback played a critical role in shaping this project into its final form.

We are also grateful to our fellow classmates, who provided motivation, collaboration, and healthy competition during this project. Their encouragement and willingness to exchange ideas greatly enhanced our learning experience.

Abstract

This project is a new approach to improve traditional firewalls by using artificial intelligence to improve network security and give the firewall the capability to detect real-time malicious traffic and integrate with VPN and Wazuh for secure communication for enhanced security and use Wazuh for real-time alert generation and mitigation. Traditional firewalls rely on a rule-based approach to detect and mitigate known attack patterns or signatures which make traditional firewalls reactive and not proactive. This allows the firewall to detect and block malicious traffic in real-time. VPN provides a secure communication channel for enhanced security and the Wazuh allows real-time logging and alerting functionality. This project showcases how AI can be integrated with network security to enhance threat detection and sets a new standard for network security.

Legacy firewalls are often insufficient in the current cybersecurity landscape for detecting and preventing sophisticated and unexpected attack techniques. To fill this void, this project introduces Neurawall, an artificial intelligence-based firewall system that combines automated mitigation, real-time threat detection, and machine learning-based traffic analysis. Through the utilization of a hybrid detection model that integrates Gradient Boosting for supervised classification and One-Class SVM for anomaly detection, the system is able to better detect known and unknown threats. Training data was derived from real-time traffic captured using Dumpcap and processed with CICFlowMeter, as well as from popular benchmark datasets such as CIC IDS 2017 and 2018.

While Neurawall works well against a broad range of network threats, its effectiveness in modern HTTPS or VPN-protected environments is hindered by the current limitation of the model to inspect only plaintext traffic. Also, lack of real, diverse traffic patterns in artificially created data led to a slight drop in real-world accuracy. Despite these limitations, the system's intelligent detection feature, real-time response, and multi-layered security approach demonstrate how machine learning can significantly enhance network perimeter security.

Table of Contents

Table of Figures

List of Tables

Chapter 1: Introduction

Chapter 1: Introduction

1.1 Introduction

Cyber threats are becoming more advanced, and traditional firewalls that work on a rule-based approach cannot keep up with the advancing threats. This dependency on rules makes it difficult for the traditional firewall to detect new threats and can easily bypass the firewall. This advancing threat landscape requires a new and modern approach that can detect these new threats to overcome the issues of a rule-based approach. This project introduces an AI-driven approach to provide improved threat detection and improve the overall network security of a system.

Through a function of network traffic pattern analysis and abnormality detection that implies malicious behavior, the AI-based firewall solution aims to provide a robust, yet flexible solution set that adapts itself to live and future threats. The AI model uses the ability to independently classify safe and dangerous traffic, thus automating the threat detection and response processes. There is a VPN Integration Module, meaning safe data transfer, an AI Traffic Analysis Module to monitor unencrypted network traffic in real-time, and a Wazuh Integration for recording irregularities and making real-time alerts. By combining these modules, they offer a holistic approach to network defense that can protect against various kinds of attacks with zero human intervention.

This project focuses on unencrypted communication to achieve low latency and high processing rates, which are essential for detection as well as prevention in real time. It trains a machine learning model with a log dataset to learn from past events to refine its detections over time. This firewall AI can evolve its content over the emergent patterns of the data trends concerning threat signatures. It improves security while reducing the administrative burden for the cybersecurity staff. This is the opposite to traditional firewalls, which are usually harder to set up and update rules manually.

1.2 Opportunity & Stakeholders

The growing sophistication of cyber threats combined with their frequency poses a serious challenge, especially for businesses and individuals who utilize predominantly rule-based traditional firewalls that may have problems identifying new or fast-changing attack patterns. Traditional systems are less responsive to emerging threats to the extent that their static rules make possible security breaches. This would open the window for developing an AI firewall capable of using machine learning algorithms for the adaptive sensing of threats and adaptive response towards the threat sensed for the improvement of security. An intelligent firewall may complement the gaps found in traditional firewall technology by automatically incorporating real-time traffic monitoring and predictive threat identification.

One of the objectives of an AI-based firewall solution project is to provide a proactive defense solution that may identify hostile network traffic without depending on pre-defined rules. Instead of depending on traditional techniques, the AI-based traffic analysis within the system will adapt to novel or unexpected patterns used by attackers, thus providing better protection. Moreover, integration with Wazuh features to support real-time anomaly monitoring and alerting in addition to supporting VPN connections provides safe data transfer, enhancing the firewall's effectiveness and making it among the most appealing protection solutions for business organizations with rife cyber threats.

Key stakeholders involved in the project

The following are the significant stakeholders in the project:

- **Cybersecurity Analysts:** It helps analysts respond to serious threats faster without having to sort through a lot of benign traffic data by using Wazuh integration, which monitors abnormalities and provides real-time alerts.
- **Businesses and Organizations:** Businesses of all kinds rely on secure networks to store sensitive data. A firewall powered by AI helps keep the important data, hence assuring the integrity of data and business continuance.

- **Compliance Officers:** The intelligent firewall solution can benefit organizations subject to high data security standards by automated detection and significantly lowering the potential of an unnoticed data breach, thus earning compliance standards.

Indeed, our AI-based firewall solution project would resonate with the cybersecurity needs of many stakeholders because it offers a solution that adapts to changing threat profiles and offloads the burden of manual network security oversight.

1.3 Motivations and Challenges

1.3.1 Motivation

The limitation of traditional rule-based firewalls in the highly dynamic cyber world today presents a driving force toward an AI-driven firewall. The traditional firewalls use predefined rules to identify and block the threat, which does not seem too effective in newly developed attack methods. Whereas, on the other hand, an AI-driven firewall solution provides a dynamic, learning-based approach to adapt itself against threats and patterns noticed in the network traffic flow. The ability to automatically recognize and respond to new assault signatures provides a far more flexible and robust defense.

Besides the above, cybersecurity experts have a lot of network traffic data in addition to the requirement of detecting threats in real time and defeating them. This project proposes making network defenses more proactive and efficient by saving labor, optimizing security processes, and speeding up detection using machine learning to recognize malicious activity. Another driving force behind this is the need for an intelligent system that would be scalable to different network sizes and could allow the management of complexity without regular manual updates. This AI-driven firewall solution will fulfill these needs and offer businesses a more autonomous and versatile security solution.

1.3.2 Challenges

Although the advantages of an AI-driven firewall solution are innumerable, implementing it is quite challenging.

- **The kind of traffic the network will be controlling:** The pattern of traffic is different depending on the applications used, the number of users, devices, and the reason behind the network, which could be corporate public or home network. Hence, unpredictability creates the need for a firewall solution to react to various forms of traffic, like online browsing, file sharing, email streaming, video, or VoIP. A one-size-fits-all approach will hardly suffice to distinguish benign from malicious communications.
- **Impact:** A firewall that is insensitive to diverse network behaviors might either over block legitimate traffic, hence causing service interruptions, or under block malicious traffic, hence letting destructive data through. The firewall needs to be dynamic and learn the normal flow of traffic on a given network.
- **Real-Time Processing:** Real-time management of traffic is an important issue for firewall systems. As the traffic moves constantly, the firewall must analyze each packet in real time to allow decisions on whether to accept or reject the packet. Delays of high magnitudes coming from the firewall may even affect the performance of the network or crash major applications dependent on the traffic.
- **Impact:** High latency can lead to communication delay, slow communication, or even complete system failure. Very efficient algorithms that process a large amount of data in minimal time are required for real-time processing without introducing latency.
- **Integration to Wazuh and VPN:** In addition, a firewall needs to integrate amicably with other aspects of security elements, such as Wazuh, for intrusion detection and monitoring, and VPNs for secure encrypted connections. Its integration will certainly provide a guarantee that the firewall operates under an all-inclusive security framework, which impacts positively on the effective monitoring of the devices and threat detection.
- **Wazuh and VPN integration** with the firewall may be technically challenging. Firewalls cannot inspect the contents of encrypted VPN communications, nor can they inspect the contents of any packets. Likewise, making sure that Wazuh warnings and logs are impacted by firewall actions, like blocking the evil IP, increases the complexity of the system.

- **False Positives:** A firewall may mistake legal traffic for malicious traffic. This can cause real business-stoppage problems, which may include the failure of legitimate users to access some services or applications.
- **Impact:** High false-positive rates weaken the firewalls' potential because users or administrators will be annoyed by the rate of legitimate requests that are denied. Furthermore, this also boosts the workload for the security team, who will have to clear and respond to all false alarms.
- **False Negatives:** The firewall fails to detect malicious traffic and allows it to pass through without interruption. This means that attacks will go undetected passing through the firewall, resulting in data leakage or other intrusions.
- **Impact:** Concerning because false negatives compromise the security of the network. False positives are inconvenient because they cause interruptions in services or communication. True negatives are not threats to the system, yet they expose the system to threats that the firewall does not detect.
- **Shifting Baselines with Scaling Networks**: As networks increase in size, so also does the baseline of normal traffic. New devices, applications, and protocols constantly come online; hence, the firewall's evolving perception of what constitutes "normal" net traffic needs to adapt in order not to misclassify legitimate activity as hostile.
- **Impact:** Failure of the firewall to evolve using new baselines means that potentially it might not correctly identify malice in its traffic and thereby elevate false positives or negatives. Some legitimate new patterns of traffic might be flagged down and listed as attacks since the firewall has not updated its understanding of the regular traffic baseline.

These issues mean that to design an efficient and reliable AI-based firewall solution for a real scenario, tests, optimizations, and development must be performed in depth. Since all the above challenges were also overcome by the firewall, it can provide robust highly intelligent adaptive defense systems against Internet attacks.

1.4 Significance of Study

A cybersecurity innovation is an artificial intelligence-driven firewall solution, which will help in healing a few of the most important flaws with traditional network defense systems. Traditional firewalls operate upon pre-defined rule sets. The ability of such systems to resist advanced attacks, which are constantly modified to evade detection, is constrained by their reliance on static rules. An AI-based firewall solution can identify threats based on patterns that aren't categorized by static rules because it utilizes machine learning to identify dangerous patterns in network traffic. In contrast, traditional firewalls rely on patterns that are categorized using static rules.

This Report demonstrates a dynamic and adaptive security solution, one that learns with every input piece of data by tying together machine learning with firewall technology. Since the firewall is inherently capable of reacting to threats, without the need for frequent updates to the rule base, this adaptive feature not only enhances the current state of security but also reduces the burden on administrative security teams in the process.

The second part is Advanced Scalable Architecture, which advances cybersecurity by providing an architecture that can scale up or down depending on the network size and complexities. Enhanced firewall functionality with Wazuh functionality for real-time warnings and anomaly detection, and VPN to process traffic securely make this a robust and versatile solution for modern-day enterprises in addressing their needs. Thus, the Report demonstrates how AI-based solutions may provide much more proactive forms of defense mechanisms and hence, prove to be vital in the battle against new cyber threats.

1.5 Goals and Objectives

The project aims to develop a machine learning-based AI-driven firewall solution that can dynamically classify and remove hazardous traffic. It is thus meant to significantly strengthen network security. The novelty of this firewall is based on getting rid of the prerequisite of static rule sets hence providing a more flexible detecting manner of threats. Some of the goals include:

- **Implementation of AI-Based Data Analysis:** Develop and deploy an AI model capable of showing network traffic in real-time which would reveal trends such as abnormal patterns of behavior indicating harmful intent.
- **Real-Time Threat Response Integration:** Avoid latency associated with threat responses by ensuring the firewall can accept or decline traffic with low latency according to the predictions provided by the AI model.
- **Use a VPN to Process Data Securely:** To maintain privacy and integrity with flows of traffic, add a VPN for secure data handling.
- **Real-Time Alert and Anomaly Logging:** Wazuh can be used to generate alerts and log unusual events that will assist in monitoring them and responding more promptly to potential security breaches
- **Scale and Performance:** Scale the firewall architecture to accommodate various network sizes as well as performance demands while ensuring the ability to continue good operation across the possible operating settings.

1.6 Scope of the Project

This project is intended to design an AI-enabled firewall, which can be programmed to analyze network data. Not relying on traditional rule-based techniques, this project is meant to develop an AI-driven approach that can automatically identify and delete any potential threats through insight from machine learning. The firewall will also feature integration with Wazuh for real-time anomaly monitoring and alerting. A VPN module will also be available for safe traffic management.

This plan will not make use of encrypted traffic inspections. The scope is only bound for the analysis of unencrypted traffic. In addition, the logs dataset from the current generation of firewalls is used for the training of the AI model of the firewall. It is after this approach to ensure getting a high detection accuracy, that is bounded within the dataset. Although the system architecture is scalable, the current implementation is optimized for small- to medium-sized networks.

1.7 Chapter Summary

This chapter, by introducing the project, allowed the possibility of an AI-based firewall solution as a more flexible security solution. We presented the motives behind such a strategy, like the inadequacy of rule-based firewalls and the need for a dynamic, data-driven solution. Identifying key stakeholders in such a project was demonstrated by showing possibly what this project may offer to compliance officers, enterprises, or even network security teams. This chapter also discussed the importance of the study to suggest how AI can be used in improving network security. Finally, we defined the scope of the project and set specific goals and objectives as a basis for laying a framework for technical developments and implementation covered in later chapters.

Chapter 2:
Market Survey

Chapter 2: Market Survey

2.1 Introduction

Due to the increasing sophistication of cyber threats, attackers have been able to bypass traditional firewalls. For this reason, AI-powered systems recently hit the use for cybersecurity purposes. They are designed to improve detection capabilities and lessen dependencies on preset rules. AI-driven firewalls, by using ML for deep analysis of network traffic patterns, can detect emerging threats that would have gone unnoticed by traditional firewalls. This chapter, based on the gap in the research area, contrasts traditional techniques with AI-based ones, and products available in the market, current technology, and products concerned with AI-driven firewalls; it further describes the problem this project intends to solve.

2.2 Technologies & Products Overview

2.2.1 Traditional Firewalls

Traditional firewalls are very effective at countering known attacks because they use static rule-based filtering. They do not have the flexibility to handle new, unidentified attack patterns. According to numerous reports from studies, the maintenance overhead of traditional firewalls is high because they require updating quite frequently to adapt to changing threats. Furthermore, especially in complex network environments, these firewalls give rise to a very high number of false positives that can demoralize security personnel and therefore reduce the efficiency of responses.

2.2.2 Artificial Intelligence in Cyber Security

AI has developed significantly in so many industries. Cyber security is one area where AI has gained tremendous growth. Pattern and anomaly-recognizing models can be developed for AI using machine learning approaches, especially supervised and unsupervised learning. It particularly finds excellent utility in detecting undefined rule-based threats.

Most recent studies depict how precise and versatile the AI-driven systems are in the detection of dangers and prevention when compared to the traditional ones.

2.2.3 AI-Powered Firewall Solution

This firewall solution is an enhancement to conventional firewalls in the sense that they learn automatically from data to identify dangerous patterns. As this solution does not rely on preset rules, they can detect new patterns of attacks which reduces the false positive rate. While many still retain some rule-based elements, many products available today have started making use of AI for enhanced security.

Table 1: *Market Analysis*

Feature	**Traditional Firewalls**	**AI-Driven Firewall Solution**
Detection Method	Rule-based	Pattern recognition using machine learning
Adaptability	Limited (requires manual rule updates)	High
False Positives	High, especially in dynamic environments	Lower, due to continuous learning and adaptation
Maintenance	Low	Reduced need for manual updates

AI-powered firewall solutions offer some advantages over traditional firewalls due to flexibility and the ability to identify unidentified threats, though they may consume more processing power and require higher-quality training data to achieve the best results in busy environments.

2.3 Research Gaps

Although there is huge potential in AI-driven firewall solutions, many research gaps need to be met to make them widely used in the market. To begin with, real-time processing capabilities are indispensable for a firewall's success and are quite compromised in most current solutions. Next, most AI-based security models train on a pretty small amount of data which they compromise when facing a variety of threats; larger datasets are required for more proper training of the models. Moreover, even though AI-based solutions decrease false positives considerably, it is not immune to them entirely; much research is required to discover the best approach to balance detection accuracy with false positives to be reduced.

2.4 Problem Statement

The problem that the current research aims to solve is the traditional firewall's incapability of sensing and responding to threats along with the unknown cyberspace. Traditional firewalls based on rule-set criteria are static, and they don't give effective detection for new attack patterns. This research project was aimed at establishing an AI Firewall that can detect malicious activities without static rule sets by learning from network traffic patterns. The objective is to design a proactive defense solution to modern network security challenges with high adaptability and real-time threat detection, coupled with low false positives.

2.5 Chapter Summary

This chapter discussed the shortcomings of traditional firewalls, opportunities for AI in cyber defense, and the present market scenario of AI-based firewalls. It demonstrated, through comparison, the advantages of AI-based approaches such as flexibility and fewer false positives. There are many questions left unanswered that highlight the need for better real-time processing, more detailed training datasets, and completely autonomous solutions. To cap off the chapter, an issue statement that described the challenges that this project's AI-driven firewall solution would aim to solve laid out the context for a solution that will be described in detail later in the chapters

Chapter 3:
Requirements and System Design

Chapter 3: Requirements and System Design

3.1 Introduction

In this chapter, we shall discuss the system requirements and design considerations of such a system. For instance, we shall report an AI-based firewall system whose learning is found within its architecture to achieve improved network security. Conventional firewalls focus on static rule-based detection methods that are usually too simplistic for detecting evolving threats. Hence, the system proposed here incorporates two AI models, in the sense that it operates on real-time traffic capture, feature extraction, and dynamic IP blocking; this therefore implies a flexible solution. We propose a thorough study of the system architecture, functional and non-functional requirements, and design diagrams. We will discuss threat scenarios and resistance models for the firewall to be more resilient against new and known cyber-attacks.

3.2 System Architecture

The system architecture of the firewall is AI-based, which uses various modules in a cooperative way to catch, analyze, and respond to network traffic:

- **VPN Module:** The module, on the front side, provides a secure communication tunnel, thereby ensuring that all traffic entering and exiting the network is encrypted before reaching the AI-based firewall system. This thus reduces the initial vulnerability that comes about due to external attackers and network traffic.
- **Traffic Collect Module:** This module makes use of ***Dumpcap*** which captures the raw network traffic data in real-time from the network interface. ***Dumpcap*** is configured to run in promiscuous mode, thus enabling it to capture the traffic from all the network devices and thereby providing the full data for analysis.
- **Feature Extraction:** ***CICFlowMeter*** aggregates raw traffic data and extracts relevant features such as packet count, flow length, source and destination ports, and other metadata. These flow-based properties are used as inputs to AI algorithms.
- **AI models:** The architecture consists of two AI models:

    - **One-Class SVM Model:** This model is trained on only benign network traffic from the existing network environment and creates a behavioral baseline against which changes can be identified. The model detects traffic significantly deviating from any known benign behavior as a potential threat.
    - **Supervised Model:** This supervised model works on CIC IDS 2017 and 2018 data sets. It identifies known harmful patterns and attacks like DDoS brute force attacks and port scanning. The supervised model improves threat detection significantly from labeled past data.
- **Wazuh Monitoring and Logging:** It is integrated to log and notify on observed abnormalities; the system monitors activity, giving real-time notifications when suspicious behavior is found, an aspect that brings much-needed context into incident response and auditing.
- **Firewall Module:** It will dynamically blacklist IP addresses identified by AI models in combination with Windows Firewall. Whenever there is malicious traffic observed, it will put that IP address into the block list of the firewall so that subsequently, communication won't happen.

Firewall Agent:

- **Traffic Capture:** Utilizes Dumpcap to capture unencrypted traffic.
- **Feature Extraction:** To extract relevant features (like duration, packet count, byte count, and flow direction), CICFlowMeter is used.
- **AI Detection:** Utilizes a trained machine learning model to detect malicious communications.
- **Reporting:** Informs the central server about malicious IP addresses and detection.

Firewall Server:

All the agents push changes to the central server, which then schedules the response:

- **Command Distribution:** All the agents are commanded based on aggregated intelligence to permit or deny specific IPs.
- **Worldwide threat view:** Aggregates logs and decisions to keep a global threat view.

- **Logging and Monitoring:** Integrate with Wazuh, which collects, analyzes, and feeds the GUI real-time alerts.
- **Platform:** Ubuntu is employed by the server to ensure reliability and compatibility.

GUI (Graphical User Interface): The system uses a friendly GUI to present information in real-time, such as the blocked IP, malicious activity log, and system alarms. In addition, the administrators can simply evaluate the system's current status, including those threats that have been identified and actions performed.

3.3 Functional Requirements

Functional requirements give the general functionality of AI-based firewall systems.

- **Traffic collection:** Round-the-clock gathering of traffic on the network from all devices connected, that deliver a real-time stream of data for analysis.
- **Feature extraction:** Making use of traffic collected, extract with precision relevant network properties to produce reliable input for AI models.
- **Anomaly detection:** The one-class SVM model must create a consistent baseline of benign traffic and must recognize anomalies as potential threats.
- **Threat Identification:** The supervised model should be able to identify labeled data with high accuracy so it identifies known threats, as well as trends towards malicious traffic, and hence, the system must block identified malicious IPs dynamically. In addition, it adds the related entries to the block list of the Windows Firewall without human intervention.
- **Logging and Alerting:** Wazuh should log all suspicious occurrences and pass alert messages in real time to the network administrators so they can take appropriate steps at once.

3.4 Non-Functional Requirements

This system is ensured to gain criteria around performance, usefulness, and adaptability.

- **Accuracy:** AI models must be very accurate to avoid false positives-cases of legal traffic labeled as harmful-and false negatives-malicious traffic that goes undetected.
- **Security:** In order to ensure data integrity and security, any communication between backend services and the web-based GUI needs to be secured with TLS 1.2 or greater.
- **Real-time extraction:** Model analysis, and traffic capture with minimal delay must occur so that fast responses can be made to threats.
- **Scalability:** It should be able to scale up the volume of traffic as well as network size without loss of performance.
- **Reliability:** The system should reliably and consistently identify threats, even in conditions of high traffic.
- **Usability:** The GUI must be easy to use, and have meaningful information such as blocked IP addresses, alarms, and logs, so that it does not require technical expertise for non-technical persons.
- **Adaptability:** AI models are to be continuously updated with new datasets so that they continue to perform well despite the appearance of new dangers.

3.5 Design Diagrams

The design diagrams reveal the interactions and data flow within the system.

The **Use Case Diagram** shows the interactions that the system may have with users on handling their tasks concerning viewing blacklisted IP addresses, being alarmed, and being logged in.

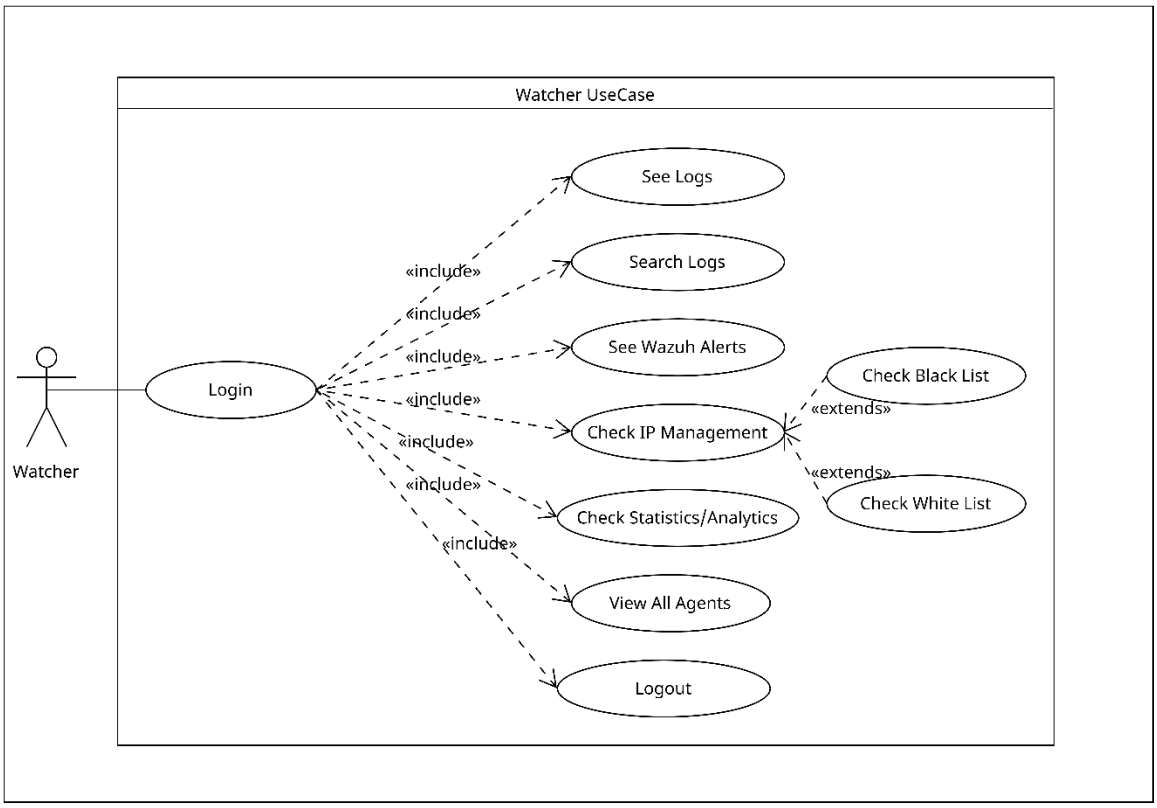

Figure 1: *User (Watcher) Use case Diagram*

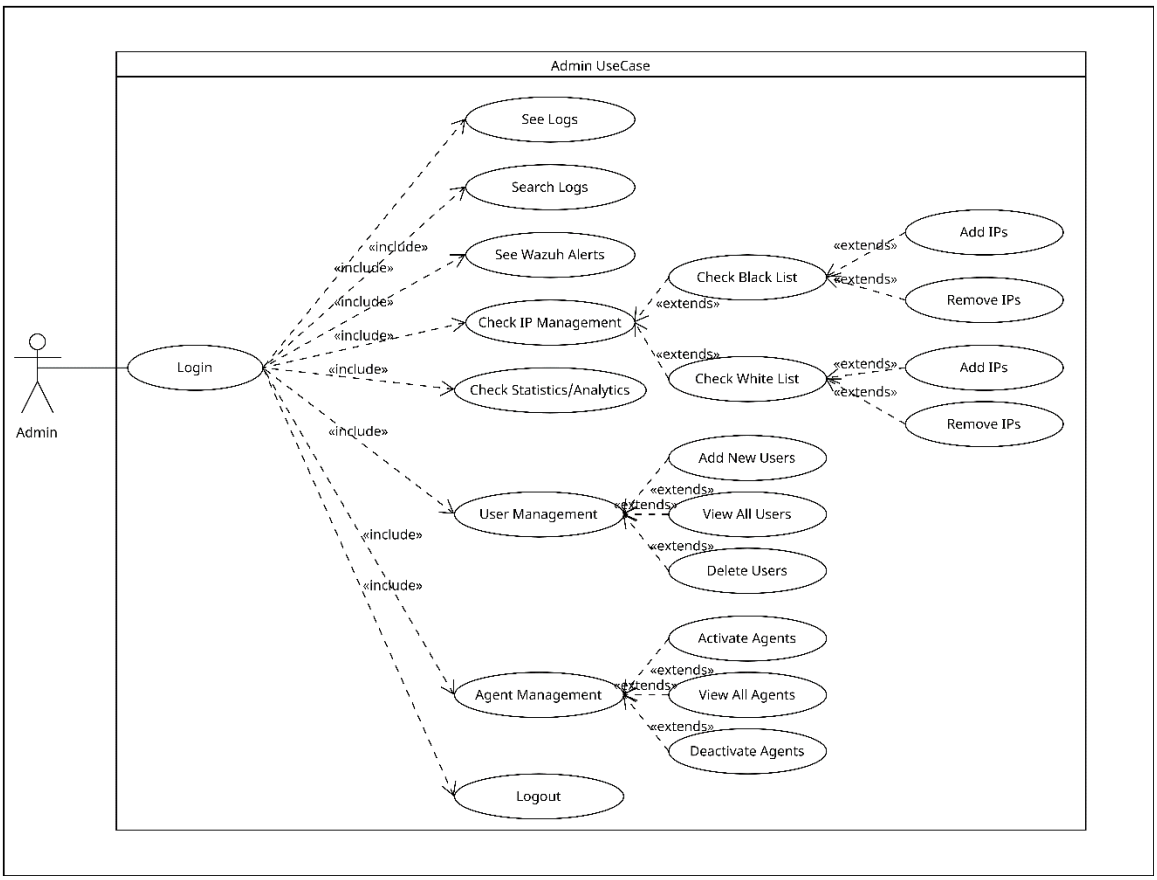

Figure 2: *Admin Use case Diagram*

This is depicted in the **Data Flow Diagram** showing data flow from initial capture of traffic from the network to feature extraction into the AI model processor and then on to firewall blocking.

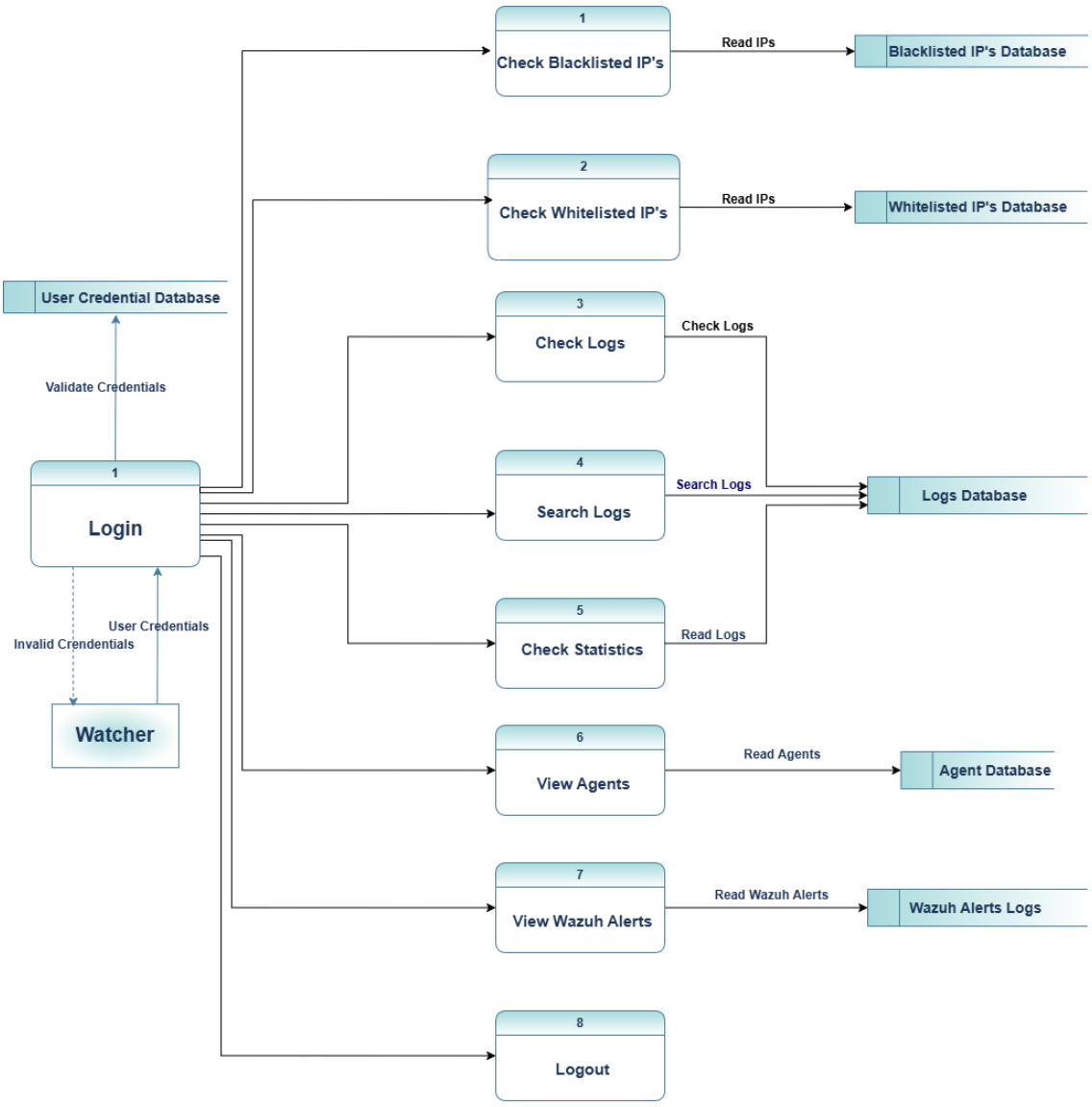

Figure 3: *User (Watcher) DFD Diagram*

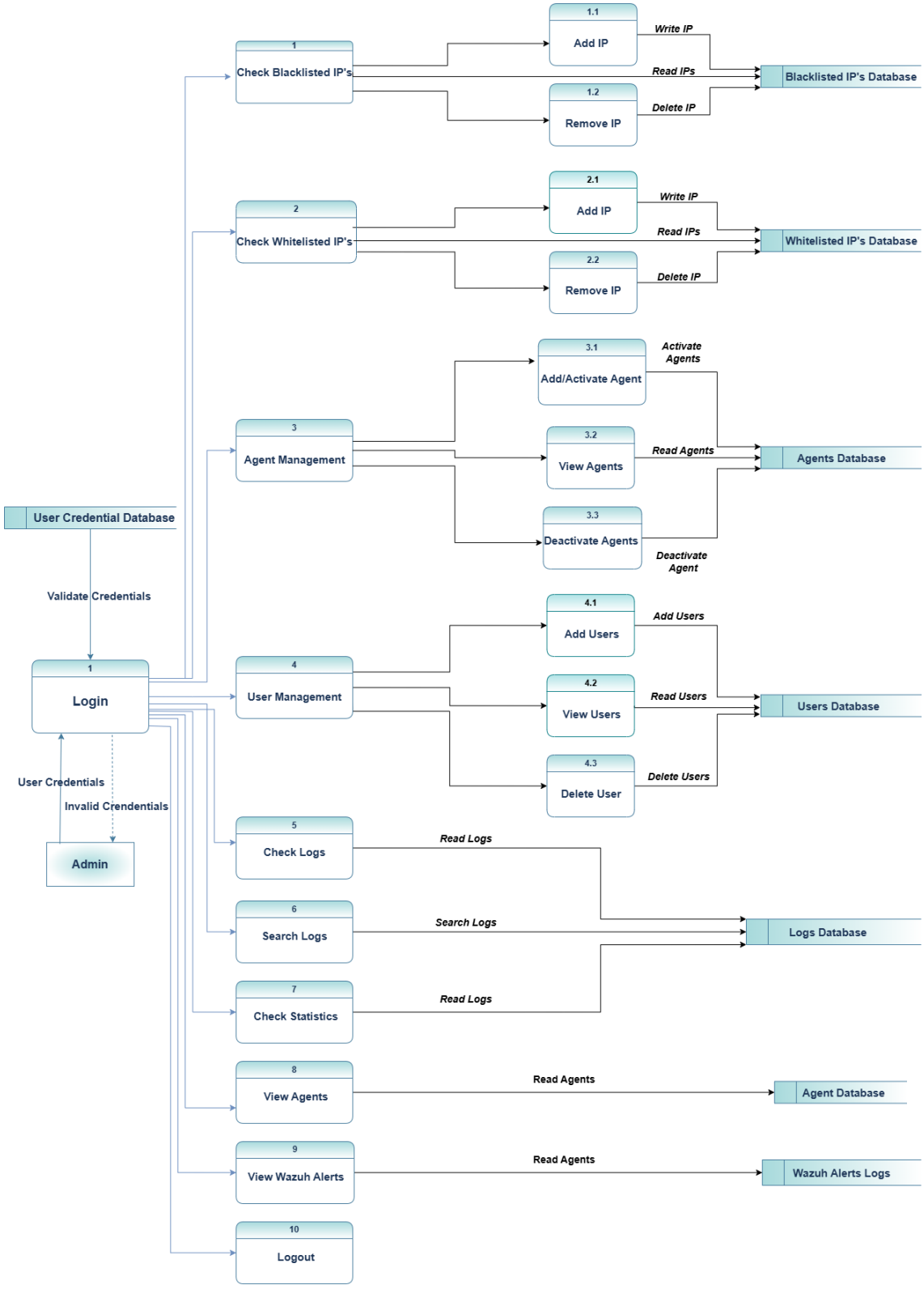

Figure 4: *Admin DFD Diagram*

Sequence Diagram: These diagrams show the step-by-step process of how the process happens from gathering traffic, then analyzing features, making a model prediction, and updating the firewall.

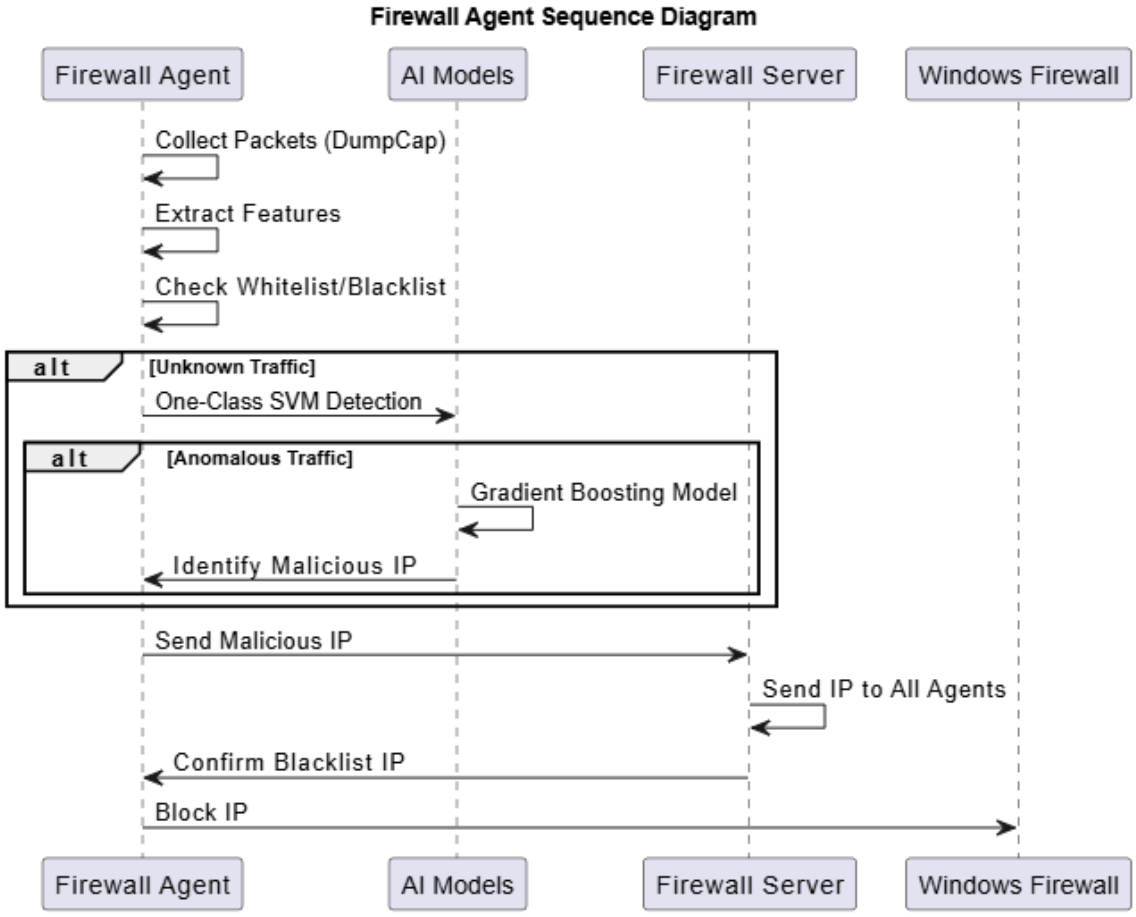

Figure 5: *Firewall Agent Sequence Diagram*

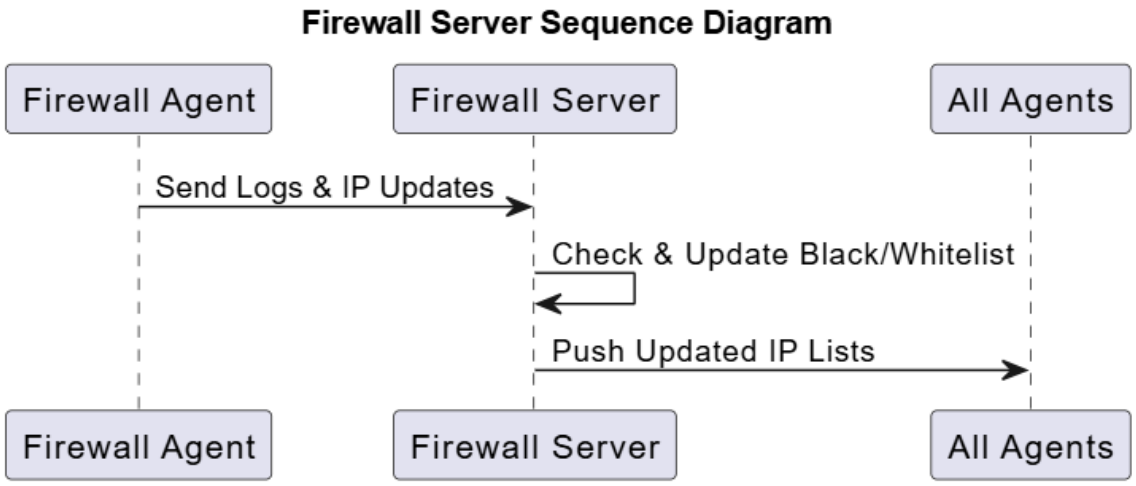

Figure 6: *Firewall Server Sequence Diagram*

Figure 7: *Wazuh Sequence Diagram*

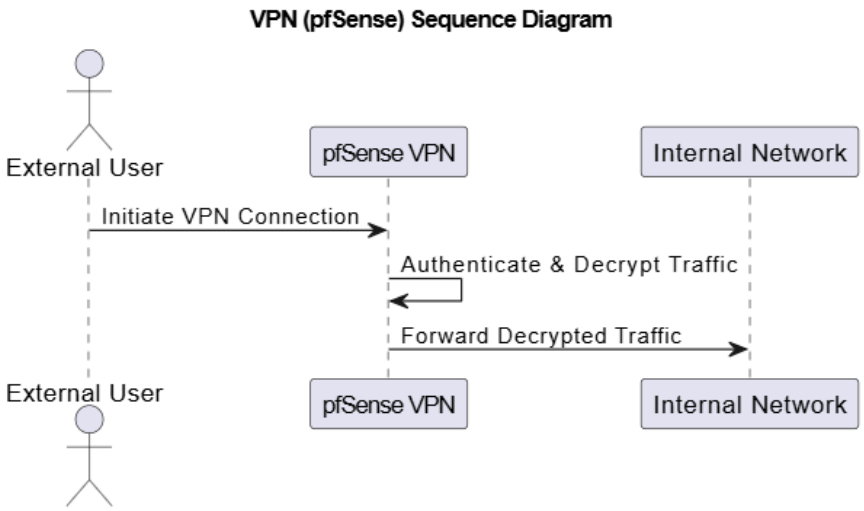

Figure 8: *VPN Sequence Diagram*

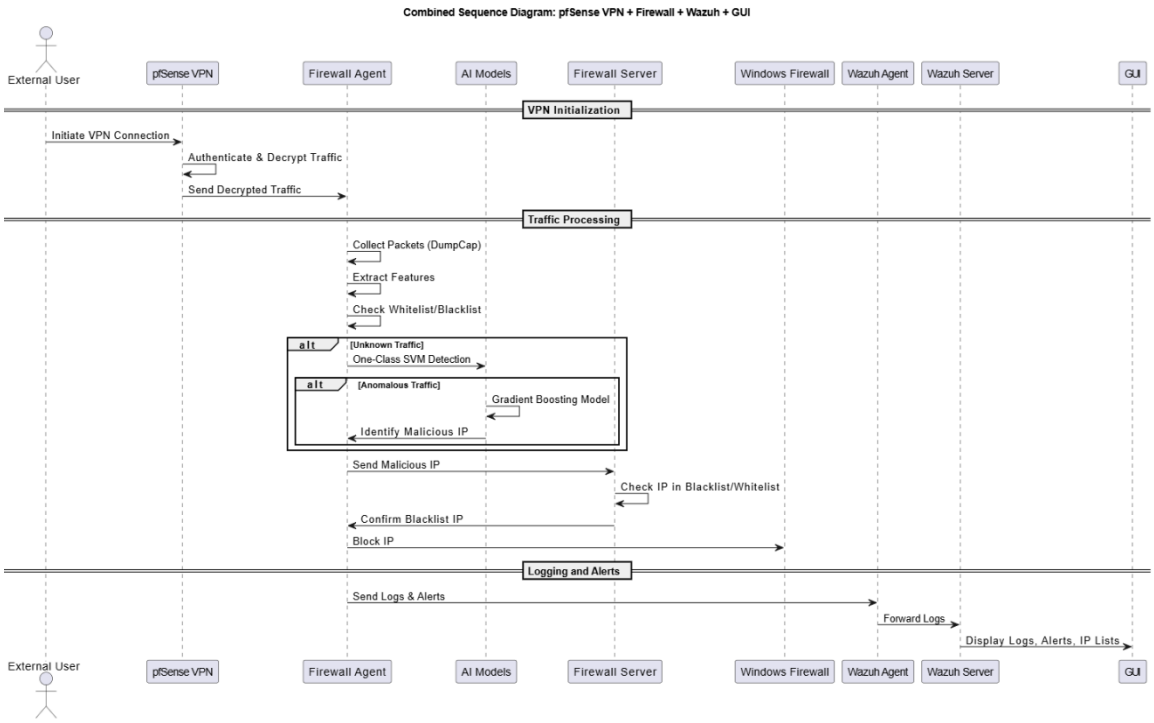

Figure 9: *Full System Sequence Diagram*

Architecture Diagram: Comprehensive View of All Components, with VPN, Traffic Capture, AI Models, Wazuh, Firewall, and the Interconnecting Points among them.

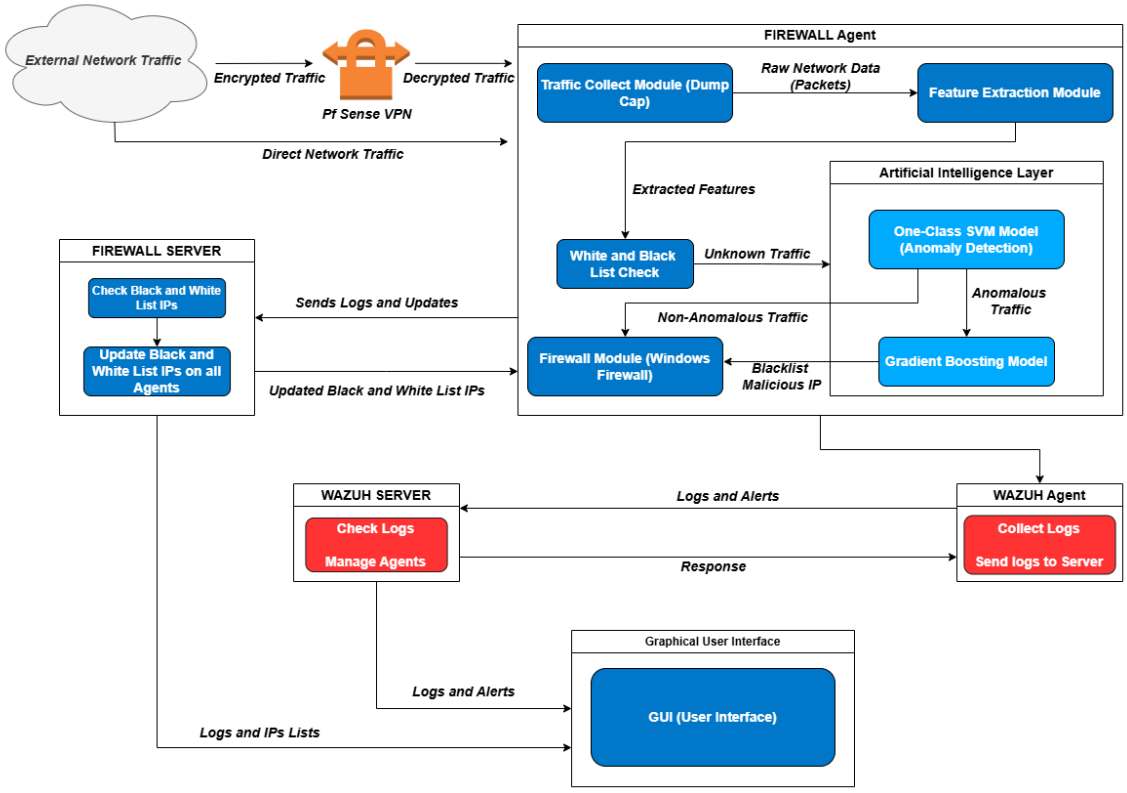

Figure 10 *Architecture Diagram*

Server Agent Diagram: This shows the Agent-Server architecture of the system which includes Wazuh and Firewall.

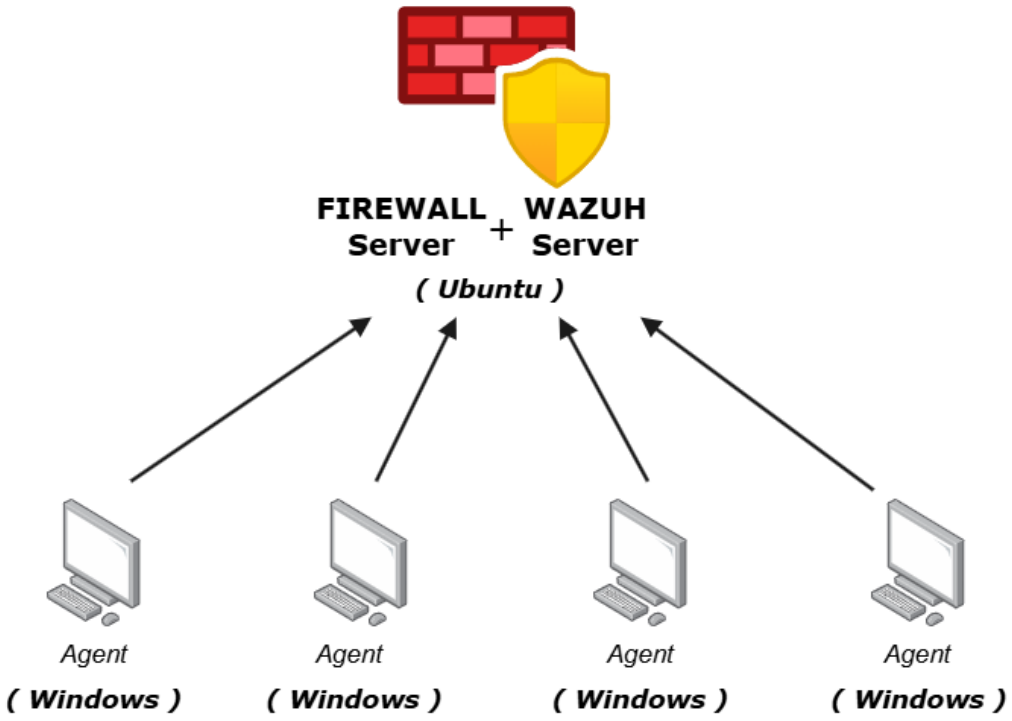

Figure 11 *Server Agent Diagram*

3.6 Hardware and Software Requirements

Server Requirements:

- **Operating System**: Ubuntu 22.04 LTS
- **CPU**: Quad-Core (Intel i5/i7 or equivalent)
- **RAM**: 8–16 GB
- **Storage**: 500 GB SSD
- **Software**: Python 3, Flask, Wazuh, SQLite, CICFlowMeter
- **Network**: Static IP, Open port for agent communication.

Agent Requirements:

- **CPU:** Multi-core processor: Intel i5 or higher, to handle data processing and parallel real-time analysis.
- **RAM:** At least 8GB to handle traffic and model processing

- **Storage:** 256 GB SSD: for logs, datasets, and extracted traffic features
- **Network Adapter**: It is a network card with the network adapter to enable the process of capturing traffic in promiscuous mode and monitoring it in all ways.

Operating System:

- Windows OS to ensure compatibility with Windows Firewall.
- Programming language that supports machine learning frameworks and is integrated with CICFlowMeter.

Tools:

- It includes tools like **Dumpcap** to capture network data in real-time.
- **CICFlowMeter** can extract features from PCAP files.
- Monitoring and alerting through Wazuh.
- **Windows Firewall** dynamically limit IP addresses.
- Flask, Wazuh and SQLite.

3.7 Threat Scenarios (Threat Handling)

3.7.1 Threats Monitored and Identified by Firewall:

All threat scenarios involve a type of attack and model, or approach used to detect and prevent it.

- **Port scanning:** It is detected using a supervised model trained on CIC IDS data that recognizes features typical of a port scan and flags all such IP addresses are blocked by the firewall.
- **DDoS attacks**: It detects DDoS attacks with a combination of Wazuh alerts of traffic spikes and also with the supervised algorithm for suspicious connection patterns. The algorithm then dynamically prohibits affected IP addresses.
- **Brute Force Attacks:** Identified based on an anomalous login pattern and detected by Wazuh logging as well as the supervised model. Those IP addresses, attempting brute force attacks multiple times, are blacklisted.
- **Network Probing:** It is identified by the One-Class SVM model, where IP addresses identified with atypical communication patterns are reported as possible probes.

3.7.2 Threats Monitored and Identified via Wazuh:

In this design, Wazuh serves as a Security Information and Event Management (SIEM) component and makes a substantial contribution to threat identification. It offers file integrity monitoring, real-time log analysis, and suspicious activity identification. Among the noteworthy hazard categories Wazuh highlighted are:

- **Attempts at Unauthorized Access:**
    - Brute-force login assaults
    - recurring instances of unsuccessful authentication
- **Signs of malware:**
    - identification of known harmful signatures or hashes
    - suspicious script execution or process creation
- **Tampering with files and configurations:**
    - Unexpected modifications to files in important system folders
    - Changes made to system policy or configuration files
- **Attempts at Privilege Escalation:**
    - Using sudo/root commands in an unusual way
    - Changes to user/group rights
- **Intrusions into networks:**
    - Attempts to connect to odd services via port scanning
    - Unexpected patterns of inbound and outgoing traffic
- **Violations of Compliance:**
    - Monitoring deviations from the system baselines
    - Setting off notifications for policy breaches.

3.8 Threat Modeling Techniques

To find potential vulnerabilities based on system components and functionality, a light-weight threat identification method was used in this project. The process focused on realistic observation and expected attack surfaces instead of formal modeling methods.
The risk identification process:

- **Risk Analysis of Traffic Flow:** Network traffic in and out of the system was analyzed for exposure to denial of service, scanning, and flooding attacks.
- **Component-level Inspection:** For identifying possible entry and attack points, each prominent module—namely, AI model, GUI, Wazuh logger, and agent-server communication—was inspected.
- **Log and Data Handling:** To prevent leakage or tampering of information, extra care was taken while collecting, transmitting, and storing logs.

Examples of Risks Analyzed:

- Unauthorized accessed logs through unprotected endpoints.
- Attacks through injections on the web-based GUI.
- TLS prevents data eavesdropping between agent-server communication.
- Malicious traffic masquerading as benign patterns to evade detection.

The team was able to focus on impactful vulnerabilities that could be tested and corrected directly during implementation due to this streamlined process.

3.9 Threat Resistance Model

The threat resistance model describes the defenses for the following threat scenario:

Proactive Detection: Using One-Class SVM and a supervised model, there would be early detection of known as well as new threats.

Layered Defense: Each layer-VPN, Wazuh, AI models, and Windows Firewall-will increase the defense and the likelihood of a successful attack being limited.

Real-Time Response: With immediate IP blocking, a firewall is capable of providing a rapid response to the damage originating from known malicious sources.

3.10 Chapter Summary

It includes topics such as architecture, functional and non-functional requirements, hardware and software requirements, threat scenarios, and modeling methods of an AI-driven firewall system. A guarantee in this system is the multi-layer approach; in it, the combination of anomaly detection/threat classification processes with dynamic IP blocking ensures a robust, adaptive response to a wide range of network threats

Chapter 4:
Proposed Solution

Chapter 4: Proposed Solution

4.1 Introduction

It describes a solution for creating a machine learning-enabled AI-driven firewall system to enrich network security. The proposed solution is a multi-layer of feature information, and learning-based threat identification. The system is composed of anomaly detection with the supervised classification that will be used for identifying known patterns as well as previously unknown attack patterns, which will make the system quite potent and multifaceted for a network-based attack-defense system.

4.2 Proposed Model

The proposed model is hybrid, implementing the methods of quantitative and experimental approaches as given below:

Quantitative Approach: The strategy involves training machine learning models on a labeled dataset (CIC IDS 2017 and 2018) to determine the accuracy and threat detection in a controlled environment. Objectively measuring the model's classification capacity of harmful traffic patterns will be reflected using metrics such as accuracy, false positive rate, and response time.

Qualitative Observations: Based on the types of threats and attack patterns identified by the model, qualitative observations will be made to understand how well the model generalizes to new attack vectors and behaviors.

Experimental Design: In the model, two machine learning models have been used:

- **One-class SVM model** trained only on benign traffic from the current network environment. In this model, aberrant traffic patterns are highlighted as deviations from a learned baseline.
- **A Supervised Classification Model** was trained on CIC IDS datasets to detect common threats like DoS, brute force, and port scanning. This model classifies fraudulent traffic using patterns learned from training data.

Anomaly-based and signature-based detection can be used together to target the threat in a wider range while reducing false positives.

4.3 Data Collection

For data collection of this project, both benign and malicious traffic data would be collected to train the AI model along with its evaluation:

- **Network Traffic Log Generation: Dumpcap** creates network traffic logs by gathering real-time network packets from a network interface. This helps in analyzing baseline learning for the model as it treats actual, benign traffic.
- **Benchmark Datasets:** For supervised model training, it utilizes the benchmark datasets **CIC IDS 2017** and **CIC IDS 2018**. These datasets contain a variety of labeled network traffic, such as many sorts of assaults like DDoS, port scanning, **brute force**, and other standard intrusion attempts. In this way, supervised models can identify and classify dangerous patterns correctly.
- **Malicious IP Samples:** The model is tested against malicious IP samples to evaluate its capacity to block known threats.

This combination of real-time and synthetic data makes the model adaptive to actual traffic patterns and resilient against known as well as emerging threats.

4.4 Data Pre-Processing

Data Preprocessing: Getting clean and consistent inputs is important for any machine learning model. The process involves the following:

- **Outlier Removal:** To avoid false alarms, data on any traffic with values highly deviating (big enough to skew the outcome) should be removed from the original dataset.
- **Filtering:** It would ensure that only the relevant features are preserved for analysis. For example, parameters such as packet count, duration, source and destination IP addresses, and protocol type are selected to focus on characteristics of traffic that may likely reveal suspicious activity.

- **Classification and Labelling:** This supervised model has used already attack categories-labeled data from CIC IDS 2017 and 2018. In training, such data aids in learning particular dangerous patterns while on the other hand, benign traffic is classified separately to enhance the classification.
- **Clustering and Prioritization:** It groups the traffic flows according to similarity in the detection of behavioral trends in between the traffic sessions and gives priority to unusual patterns in the traffic to ensure potential risks are addressed immediately.

4.5 Tools and Techniques

This project combines various tools and techniques to achieve the required functionality and accuracy in threat detection.

- **Traffic Capture:** Using ***Dumpcap*** captures real-time network traffic and returns raw packet data for monitoring and feature extraction in real-time.
- **CICFlowMeter:** It makes use of flow-based parameters of raw PCAP files like duration, packet count, and IP address. These properties form the backbone of integrating structured data into AI algorithms.
- **Machine learning models:**
    - **One-Class SVM:** Anomaly-based detection model that trains to depict only benign traffic and marks differences from normal behavior.
    - **Supervised Model:** Classification model where CIC IDS datasets are used for training to identify specific types of attacks.
- **Wazuh** is a real-time logging, alerting, and threat-monitoring tool that provides visibility as well as clarity.
- **Windows Firewall**: Integrated with the ability to dynamically block IP addresses recognized as malicious. This allows for rapidly blocking detected threats.
- **GUI:** Network administrators can monitor logs, blacklisted IPs, and real-time notifications with this web graphical user interface. TLS encryption is applied by the GUI to ensure secure communication between the frontend and backend, protecting data from interception and unauthorized access.

4.6 Evaluation Metrics

The following assessment metrics are used in determining the performance of the AI-powered firewall system.

- **Accuracy:** It gives the percentage of threats identified correctly compared to benign traffic. Therefore, it provides a picture of the reliability of the model.
- **Rate of False Positives:** This deals with the number of times harmless traffic is identified as dangerous. Low rate of false positives- prevents over-blocking an IP.
- **False Negative Rate:** It determines the rate at which malicious traffic goes unnoticed. Low false negatives are always essential to neutralize threats successfully.
- **Detection Speed:** It determines how long the model will take to process traffic data, extract features, make predictions, and block the IP address. To achieve real-time threat response, detection speed is an important factor.
- **Security Robustness:** It pertains to testing how robust the model is in terms of security dealing with many different types of threat scenarios, including DDoS, port scans, and brute force attacks, and its ability to manage threats.

Performance Comparison

- **Performance of Individual Models: Strengths and Weaknesses**

Table 2: *Models Performance Comparison*

Model	**Strengths**	**Weaknesses**
SVM One-Class	Best suited for picking up anomalous boundaries	Issues with high false negatives and high-dimensional/imbalanced data.
Gradient Boosting	Identifies non-linear trends and handles the relevance of features	Lower recall for anomalies and over-weights the main class.

- **SVM Alone:** The missed anomalies were due to reduced recall and mediocre precision.

```
Classification Report (One-Class SVM):
              precision    recall  f1-score   support

           0       0.82      0.80      0.81     19061
           1       0.58      0.55      0.56    288198

    accuracy                           0.56    307259
   macro avg       0.70      0.67      0.69    307259
weighted avg       0.61      0.56      0.58    307259

Confusion Matrix (One-Class SVM):
 [[     15249  3812]
 [     129889 158309]]
```

Figure 12: *Performance Metrics SVM*

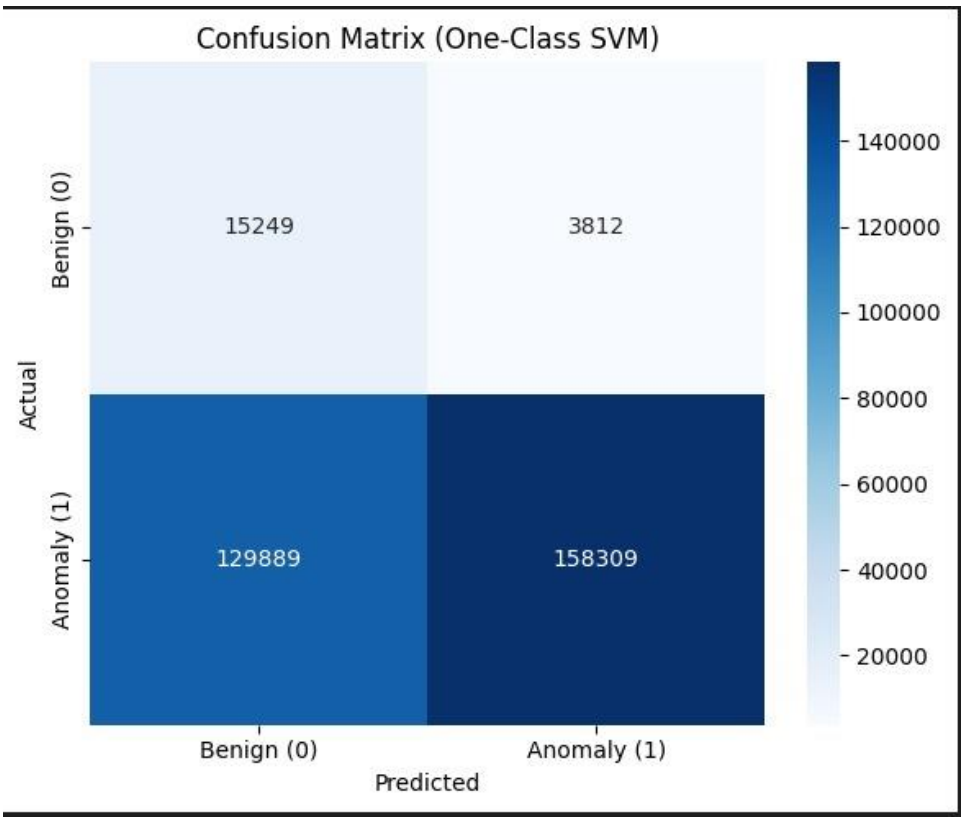

Figure 13: *SVM Confusion Matrix*

- **Gradient Boosting:** When is employed in isolation, accuracy is improved but benign traffic is favored, resulting in false positives.

```
Classification Report (Gradient Boosting):
              precision    recall  f1-score   support

           0       0.70      0.50      0.58     19061
           1       0.97      0.99      0.98    288198

    accuracy                           0.96    307259
   macro avg       0.84      0.74      0.78    307259
weighted avg       0.96      0.96      0.96    307259

Confusion Matrix (Gradient Boosting):
 [[    9531  9520]
 [     4085 284113]]
```

Figure 14: *Performance Metrics Gradient Boosting*

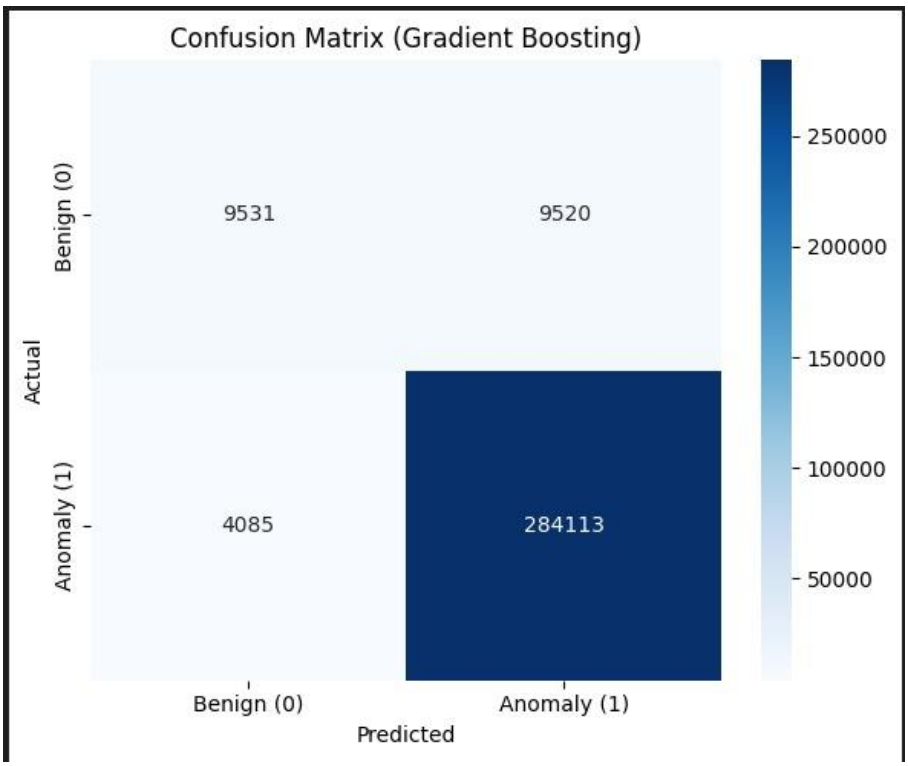

Figure 15: *Gradient Boosting Confusion Matrix*

Advantages of the Hybrid Model

The hybrid model integrates the enhanced classification strengths of Gradient Boosting with the detection of One-Class SVM outliers:

- Enhanced anomaly detection
- Less false negatives
- Balance trade-off between recall and precision

```
Classification Report (Combined One-Class SVM and Gradient Boosting):
              precision    recall  f1-score   support

           0       0.79      0.80      0.80     19061
           1       0.62      0.88      0.73    288198

    accuracy                           0.87    307259
   macro avg       0.71      0.84      0.77    307259
weighted avg       0.65      0.87      0.75    307259

Confusion Matrix (Combined One-Class SVM and Gradient Boosting):
 [[    15249  3812]
 [     34584 253614]]
```

Figure 16: *Combined Model Evaluation Metrics*

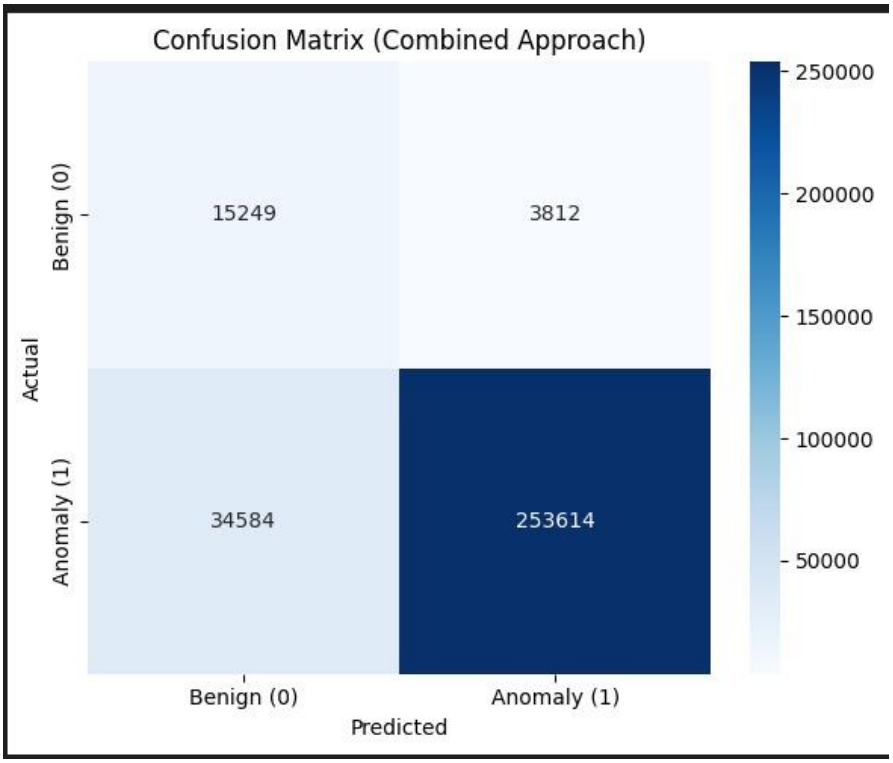

Figure 17: *Combined Model Confusion Matrix*

4.7 Chapter Summary

This chapter presented the proposed AI-enabled firewall solution, its model structure, data gathering, preprocessing tools, and procedures, and an assessment metric. The anomaly-based and supervised learning models of the proposed solution will identify harmful traffic patterns across a wide range and will have a layered defense approach with the help of VPN, Wazuh, and Windows Firewall. Data gathered from real-time network traffic and benchmark datasets can be flexible and resilient to real-world challenges. In addition, there are many different techniques and metrics of pre-processing as well as evaluation metrics that are applied to ensure effectiveness, accuracy, and reliability in threat identification.

Chapter 5:
Implementation and Testing

Chapter 5: Implementation and Testing

5.1 Introduction

Several critical security components were analyzed to ensure the proposed AI-based firewall provides a secure framework:

- **Confidentiality:** TLS encryption for encrypting communications between parts, especially the web-based GUI, to ensure confidentiality.
- It prevents man-in-the-middle attacks and eavesdropping of logs or alert information.
- **Integrity:** To detect cases of tampering, SHA-256 checksums were applied to hash data transmitted from agents to the server and logs.
- **Availability:** To prevent the firewall from crashing and denying service, the system was put through high-traffic and simulated DDoS tests.
- **Authentication and Authorization:** To prevent unwanted access to logs and control options, role-based access control, or RBAC, was implemented on the GUI.
- **Timely Threat Response:** The system demonstrated the ability to quickly block bad IPs as soon as they were identified, keeping the time users spent vulnerable to attacks as short as possible.

5.2 System Setup

5.2.1 Environment Configuration

- **Operating Systems**:
    - Server: Ubuntu 22.04 LTS
    - Agents: Windows 10/11
- **Frameworks & Languages**:
    - Backend: Python (Flask), Bash scripting
    - Frontend: HTML, CSS, JavaScript (React)
- **Machine Learning Libraries**:
    - Scikit-learn, Pandas, NumPy, Joblib
- **Other Tools**:

- CICFlowMeter, Dumpcap, Wazuh, Windows Firewall, OpenSSL for TLS setup

5.2.2 Key Functions and Algorithms:

Feature Extraction: CICFlowMeter is used by agents to extract features based on flow from unprocessed packet data.

Pseudocode:

Traffic Analysis and Anomaly Detection.txt

```
Function analyze_traffic(csv_file):
    Load whitelist IPs
    Load scaler, SVM model, Gradient Boosting model, and selected features
    Read CSV file into DataFrame
    Replace infinite values with NaN
    Fill NaN values in selected features with median
    Label traffic from whitelisted IPs as "Benign"
    Filter data with label "No Label" for anomaly detection
    If non-whitelisted data is not empty:
        Scale features
        Predict anomalies using SVM
        For anomalies detected by SVM:
            Predict probabilities using Gradient Boosting
            Label as "Malicious" if probability exceeds threshold
    Identify malicious IPs from labeled data
    Update ip_malicious_history with counts
    For each IP in history:
        If max count > flow_threshold_single or total count > flow_threshold_total:
            If IP not in existing CSV and not in whitelist:
                Append IP to CSV_FILE_PATH
    Append all labeled data to CAPTURED_FILE_PATH
    Delete csv_file

```

Figure 18: *Pseudocode Traffic Analysis and Detection*

Traffic Capture and Processing.txt

```
Function capture_traffic(INTERFACE_NUMBER):
    Initialize file_index to 1
    Loop indefinitely:
        Define pcap_file path using file_index
        Execute dumpcap command to capture traffic for DUMP_DURATION seconds
        Add pcap_file to QUEUE
        Increment file_index

Function process_pcap_to_csv():
    Loop indefinitely:
        Get pcap_file from QUEUE
        Define csv_file path based on pcap_file
        Execute Gradle command to process pcap_file into csv_file
        Call standardize_csv(csv_file)
        Call analyze_traffic(csv_file)
        Delete pcap_file

```

Figure 19: *Pseudocode Traffic Capture and Processing*

firewall rule.txt

```
Function get_ips_from_csv(file_path):
    If file does not exist or is empty:
        Return empty set
    Else:
        Read CSV and return set of IPs

Function get_ips_from_log(file_path):
    If file does not exist:
        Return empty set
    Else:
        Read log file and return set of IPs

Function block_ip(ip):
    Execute netsh command to add inbound and outbound firewall rules for the IP

Function unblock_ip(ip):
    Execute netsh command to delete inbound and outbound firewall rules for the IP

Function get_ips_from_whitelist(file_path):
    If file does not exist or is empty:
        Return empty set
    Else:
        Read CSV and return set of IPs

Function add_ip_to_whitelist(ip):
    If IP already in whitelist:
        Return False
    Else:
        Append IP to whitelist file

Function remove_ip_from_whitelist(ip):
    Remove IP from whitelist CSV

Function update_rules():
    Load IPs from CSV, whitelist, and log
    Remove whitelisted IPs from CSV IPs
    For IPs in CSV but not in log:
        Block IP and append to log
    For IPs in log but not in CSV:
        Unblock IP
    Overwrite log file with updated IPs

```

Figure 20: *Pseudocode Firewall Rule Implementation*

Main Execution Flow.txt

```
If script is run directly:
    Print "Starting NeuraWall"
    Start capture_traffic thread
    Start process_pcap_to_csv thread
    Call update_rules() for initial synchronization
    Initialize CSVChangeHandler
    Initialize Observer to monitor LOGS_DIR
    Start Observer
    Loop indefinitely:
        Sleep for 1 second
    On KeyboardInterrupt:
        Print "Stopping NeuraWall"
        Stop Observer
        Join Observer thread
```

Figure 21: *Pseudocode Main Execution Flow*

Anomaly Detection: Anomalies from normal traffic behavior are identified using One-Class SVM.

Signature-Based Detection: Through patterns learned on CIC-IDS datasets in a supervised machine learning model, traffic is classified.

Threat Action: A positive detection dynamic IP blocking with Windows Firewall, and the firewall notifies the central server.

GUI Display: Through a safe online interface, an instant dashboard displays blocked IPs, traffic statistics, and alarms.

5.3 System Integration

5.3.1 Components Integration

Firewall Agent:

- captures traffic in real time.

- executes the trained AI model after feature extraction.
- sends alarms and detection logs to the central server.

Central Server:

- receive notifications from all agents
- sends agents to block malicious shared IPs.
- hosts the centralized monitoring web-based GUI.

5.3.2 Communication Protocols / APIs

- HTTPS is an agent-to-server protocol (TLS-encrypted Flask API).
- Server-to-Agent Commands: Flask with authentication tokens via REST API
- GUI Access: TLS-encrypted browser-based communication over HTTPS

5.3.3 Databases and Log Management

Log Storage:

- Agent-side logs via SQLite
- Ubuntu's Wazuh ELK Stack centrally for log aggregation in real time

Log Sync:

- Periodically push logs using an API
- TLS encryption ensures secure transfer.

5.4 Test Cases

Firewall (AI Detection & IP Blocking)

The tests conducted to validate the core functionality of the AI-powered firewall are enumerated here. Tests for detecting malicious traffic, blocking IPs with Windows Firewall, and ensuring whitelisted IPs aren't blocked unintentionally are also encompassed.

Table 3: *Firewall Test Cases*

Test Case ID	Description	Input	Expected Output	Status Criteria	Passed
FW-01	Test malicious IP detection	Simulated malicious traffic (from	IP detected and flagged as malicious	Pass if IP logged & blocked	YES

		dataset or tool)			
FW-02	Test benign traffic	Simulated normal web browsing	Traffic allowed	Pass if traffic flows normally	YES
FW-03	Test IP blocking via Windows Firewall	IP manually marked as malicious	IP blocked in Windows Firewall	Pass if block rule is created	YES
FW-04	Test whitelist enforcement	Whitelisted IP sends suspicious traffic	Traffic allowed	Pass if not blocked	YES

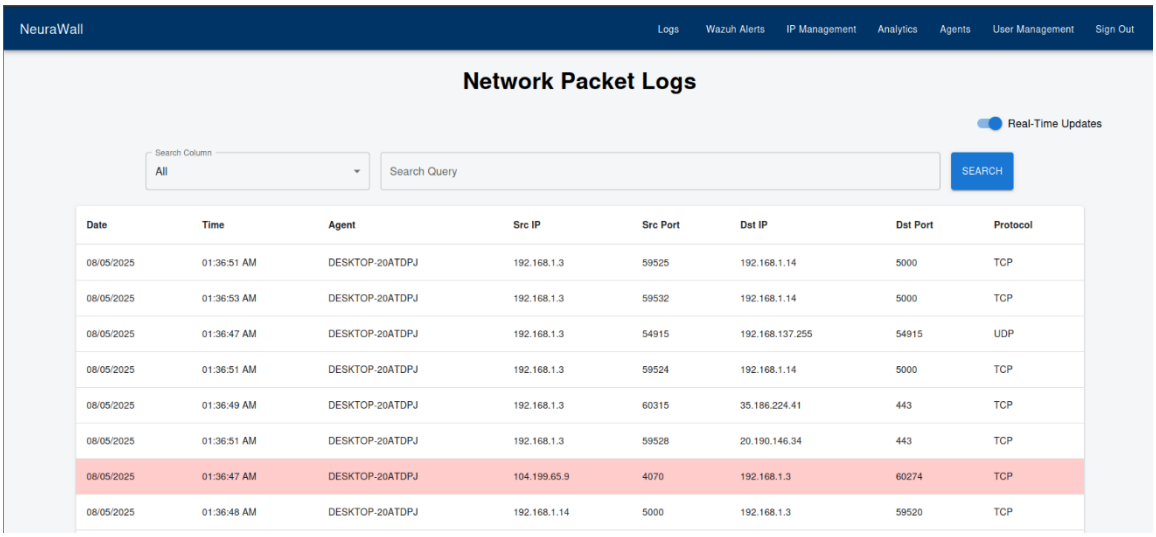

Figure 22: *Logs Interface*

Agent (Real-time Feature Extraction + Detection)

This section tests the firewall agent's ability to connect with the server, perform classification based on the learned model, extract features using CICFlowMeter, and monitor live network traffic.

Table 4: *Agent Test Cases*

Test Case ID	Description	Input	Expected Output	Status Criteria	Passed

AG-01	Test real-time packet capture	Simulated PCAP stream	Features extracted in CSV	Pass if valid CICFlowMeter output	YES
AG-02	Test model inference on live traffic	Real-time flows	AI model classifies flows	Pass if classification happens	YES
AG-03	Test failed connection to server	Server down	Detection runs but not pushed	Pass if local log shows warning	YES
AG-04	Test push of malicious IP to server	Malicious IP detected	IP sent to server API	Pass if visible in server log	YES

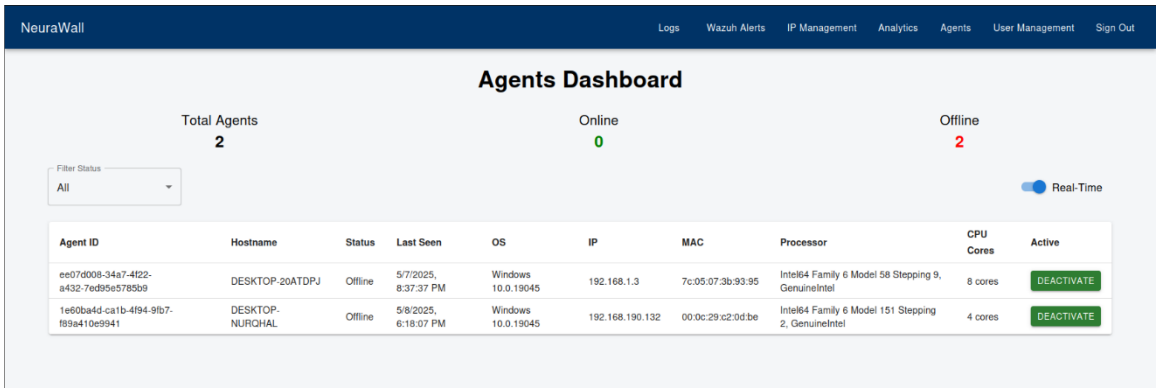

Figure 23: *Agent Interface*

Server (Centralized Logging, Command Dispatch)

This table evaluates the server's ability to store and process offending IPs, accept data from the agents, and return block or whitelist instructions to the agents.

Table 5: *Server Test Cases*

Test Case ID	Description	Input	Expected Output	Status Criteria	Passed
SV-01	Test API reception of IPs	IP pushed by agent	Server logs it	Pass if stored/acknowledged	YES
SV-02	Test broadcast of block command	New IP in blacklist	Agents receive command	Pass if agents block it	YES

SV-03	Test communication failure	Agent unreachable	Server retries or logs error	Pass if error shown in server log	YES
SV-04	Test command to whitelist IP	Admin whitelists IP	Command sent to agent	Pass if rule is removed from firewall	YES

Wazuh

The Wazuh testing ensures that system logs generated by the firewall and agent are correctly digested, indexed, and alarms are triggered on threat detection due to Wazuh testing. It also ensures that error handling and log processing are trustworthy.

Table 6: *Wazuh Test Cases*

Test Case ID	Description	Input	Expected Output	Status Criteria	Passed
WZ-01	Test log ingestion	Firewall agent generates logs	Logs appear in Wazuh dashboard	Pass if log visible	YES
WZ-02	Test alert on detection	Agent blocks malicious IP	Wazuh raises alert	Pass if alert triggered	YES
WZ-03	Test log forwarding	Log generated in server	Forwarded to Wazuh indexer	Pass if stored in Elasticsearch	YES
WZ-04	Test failure to ingest log	Invalid log format	Wazuh ignores or raises parsing error	Pass if error reported	YES

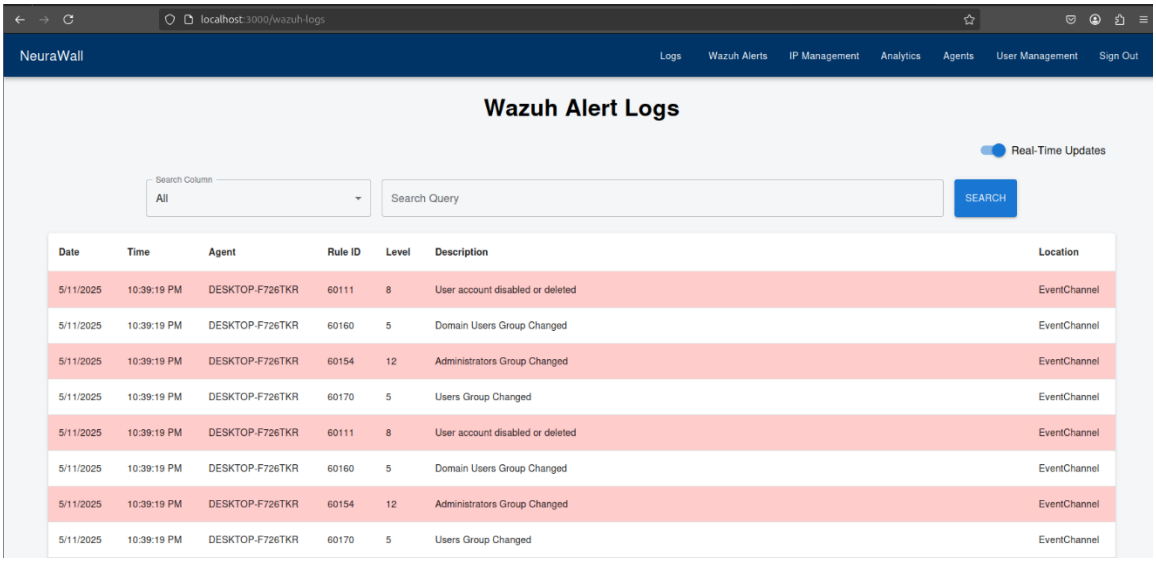

Figure 24: *Wazuh Alerts Interface*

VPN Module (if integrated for agent-server communication)

By checking the VPN tunnel setup between agents and the server, these tests ensure encrypted communication. They also check that the system handles failures in tunnels properly and that detection remains operational while using a VPN.

Table 7: *VPN Test Cases*

Test Case ID	Description	Input	Expected Output	Status Criteria	Passed
VPN-01	Test VPN connection establishment	Agent connects to server via VPN	Encrypted connection established	Pass if tunnel up	YES
VPN-02	Test detection behind VPN	Traffic from VPN user	Firewall still detects & classifies	Pass if normal detection flow	YES
VPN-03	Test VPN reconnection	Restart VPN service	Tunnel restored	Pass if data resumes flow	YES

System Integration Tests

Full system integration testing is addressed here. It ensures all components work as a unit, from the agent's threat identification to the server's command execution, Wazuh's log collection, and the dashboard's updates.

Table 8: *Integration Test Cases*

Test Case ID	Description	Input	Expected Output	Status Criteria	Passed
INT-01	Full cycle: Detection → Block → Alert	Malicious traffic from test IP	Detected by agent → Blocked → Alert in Wazuh	Pass if all stages work	YES
INT-02	Real-time detection + dashboard logging	Continuous traffic simulation	Logs populate on GUI + Wazuh	Pass if frontend shows updates	YES
INT-03	System under heavy traffic	Simulated DDoS or high traffic burst	System still processes & detects	Pass if no crash/loss	YES
INT-04	Fail-safe for whitelist IPs	Suspicious packet from whitelisted IP	Not blocked or flagged	Pass if whitelist honored	YES

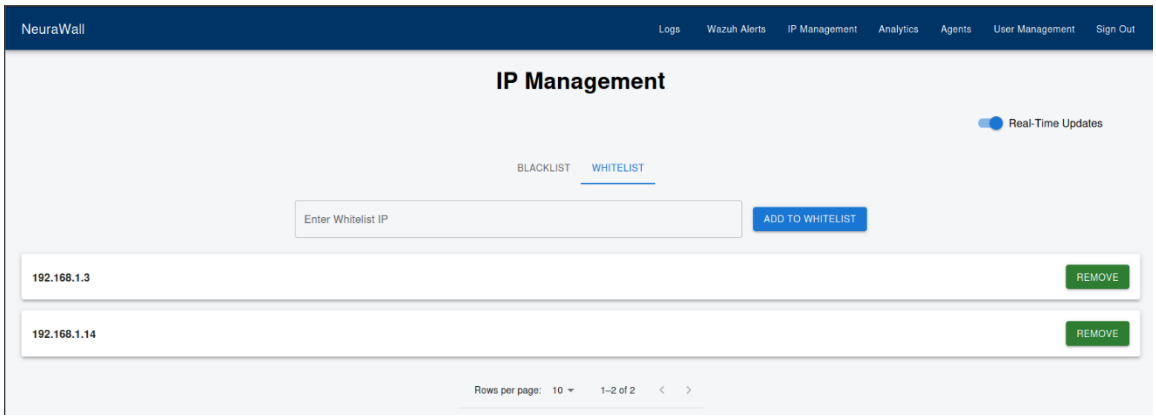

Figure 25: *Whitelist Management Interface*

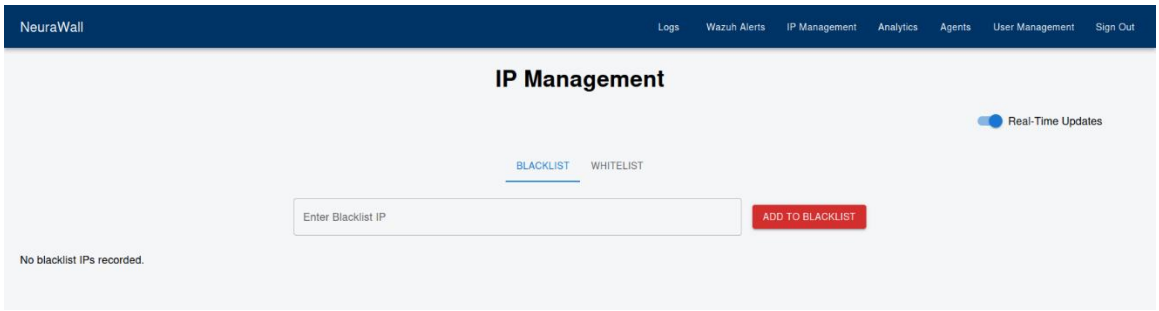

Figure 26: *Blacklist Management Interface*

5.5 Best Practices / Coding Standards

5.5.1 Code Validation

To ensure the correctness, readability, and reliability of frontend and backend parts, we employed a manual code verification method because of the size and timeline of the project. Among the main methods were:

- **Manual Testing of Functional Modules:** To ensure anticipated outcomes, all crucial functions, such as log gathering, malicious IP blocking, and traffic classification based on AI, were thoroughly tested against various input scenarios. It was particularly important to validate log forwarding behavior and model predictions.
- **Step-by-Step Input Validation:** To ensure data flow, model answers, and API interactions between the agent, server, and frontend dashboard, we carefully monitor code behavior during development using debugging tools and print/log statements.
- **Code Review:** To identify grammatical errors, logical errors, or poor design methods, team members routinely reviewed each other's work. Overall quality was improved, and maintainability was ensured by this informal peer review.
- **Frontend Browser Testing:** To verify that charts, logs, and controls rendered correctly and responded to user input as expected, the web based React GUI was tested manually in several browsers.

- **Version Control with Git:** Git was used for simple version control. To provide traceability for changes done during development, important commits were reviewed and noted.

5.5.2 Development Practices

- **Secure coding:** proper input validation, and sanitization.
- **Modular Code:** Maintainability was enhanced by separating ML models, firewall rules, and GUI into different modules.
- **Best Practices for Encryption:** TLS with HTTPS; secure storage of certificates.
- **Documentation:** consisting of README files, deployment guidance, and comments in-line.

5.6 Chapter Summary

The installation and testing of the AI-based firewall were discussed in detail in this chapter. To ensure safe communication, we began by ensuring that important security aspects, like the use of TLS encryption for the web-based GUI, worked properly.

The system installation described how to set up the environment, how to use supervised classifiers and One-Class SVM, and how to employ CICFlowMeter for real-time feature extraction.

A critical part of the deployment was the server-agent mechanism, which supported both centralized threat management and decentralized traffic analysis, improving the scalability and robustness of the system. Wazuh was utilized for real-time monitoring, APIs were employed to integrate components, and a centralized database was employed to record actions for GUI visualization.

Test cases proved effective real-time response, low false positives, and good detection precision. Overall, the testing process verified the AI firewall's real-world performance and its effectiveness in real-time threat detection and system management.

Chapter 6:
Conclusion and Future Work

Chapter 6: Conclusion and Future Work

6.1 Introduction

This chapter concludes with the design and deployment of our machine learning-based firewall system and provides a critical analysis of its achievements, limitations, and areas where it can be made better. Focusing on unencrypted network traffic, the system was designed to identify and respond automatically to malicious traffic by combining machine learning with a traditional firewall.

6.2 Achievements and Improvements

The key achievements of the project include:

- **Effective AI Integration into Firewall Architecture:** The system could dynamically block malicious IPs and employ machine learning models to analyze data in real-time.
- **Secure online graphical user interface with TLS communication:** Logs and system metrics can be accessed securely due to the creation of a centralized, secure monitoring interface.
- **CICFlowMeter**: Using **CICFlowMeter** efficiently enabled reliable flow-level feature extraction, which increased the detection performance of anomaly and signature-based models.
- **Server-Agent model**: Better scalability and real-time coordination across multiple network nodes are characteristics of the **server-agent model**.
- **Wazuh Integration:** Employing an ELK-based stack, it provided real-time log monitoring, threat visualization, and alerting.
- **Low False Positive Rate:** The supervised detection system worked efficiently by reacting to threats while ensuring a high rate of precision with minimal false positives.

6.3 Critical Review

Limited Dataset Scope: Since the AI model was trained on popular datasets, it may not be capable of detecting completely new traffic patterns or zero-day attacks.

Artificial Dataset Bias: The training data was created artificially under closely controlled conditions. While this made it simpler to structure neatly labeled instances, it also made it more difficult for the model to generalize to diverse noisy real-world traffic, which partly undermined its accuracy in real-world testing.

Concentration on Unencrypted Traffic: The utility of the system in most modern networks where HTTPS and VPNs are common is compromised since it was not designed to inspect encrypted traffic owing to the scope of the project.

Model Adaptability: Even though it is effective, the system currently does not provide adaptive learning or live retraining, which could have a bearing on its performance over time as threats evolve.

Resource Consumption: Significant system resources are required to execute real-time packet capture, feature extraction, and prediction simultaneously; this might not be suitable for all deployment contexts.

6.4 Future Recommendations

To enhance the firewall's performance and adaptability, the following recommendations are proposed:

- **Add Online Learning:** To help the AI model learn new types of attacks without human intervention, add live retraining features.
- **Encrypted Traffic Analysis:** To extend detection features to encrypted traffic, add SSL/TLS inspection methods or deep packet inspection (DPI).
- **Integration of Threat Intelligence:** Integrate the system with external threat feeds to improve detection and improve the process of blocking suspicious IP addresses.
- **Dockerization and Load Balancing:** To support scalable deployment on cloud or hybrid environments, containerize all system components.
- **Email/Mobile Alerts:** Provide administrators with multiple channels for alerts, which will improve response times.
- **Self-Adaptation:** It updates itself regularly with new samples of traffic and responds to developing risks.

- **Model Optimization:** To enhance detection accuracy and reduce processing time, experiment with more up-to-date machine learning architectures (e.g., ensemble models and deep learning).
- **Deep Logging Analytics:** Enhance Wazuh integration by correlating events with anomalies in behavior over time and across users.

6.5 Chapter Summary

This chapter laid out the key project milestones and gave concluding thought on the planning, implementation, and evaluation of our AI-based firewall system. Through the combination of supervised learning, anomaly detection, and traditional firewall techniques, a hybrid firewall model was implemented, yielding an intelligent, scalable, and practical solution for real-world network defense scenarios.

Among the significant feats was the establishment of a safe web-based GUI with support for TLS, which provided administrators with the power to remotely observe traffic, manage blacklisted IPs, and observe logs in real time. The utilization of the server-agent idea enhanced scalability and robustness by providing spread-out agents the ability to make decisions autonomously on traffic and centrally manage responses through the server. The architecture reduces overhead on one processing unit and ensures faster decision-making.

Second, the project was able to demonstrate the importance of adding open-source tools such as Wazuh for log management and security event monitoring and the importance of CICFlowMeter in flow-level traffic analysis. These features are combined to produce an efficient, multi-layered system that could protect against a variety of known and unknown threats.

But limitations were also acknowledged. Apart from the growing level of encrypted network traffic, the system was only designed to work with unencrypted traffic. Also, if the behavior of the attacker changes, the learned machine learning models are at risk of being outdated because they are static. These challenges point to the need for future

research focusing on encrypted traffic analysis and adaptive learning approaches to enhance system adaptability and robustness.

The recommendations given for the future, like containerized deployment, Wazuh deeper analytics, and threat intelligence system integration, give a clear direction for improving the system to a production-ready solution. The ability of the system to deal with modern cybersecurity challenges can be enhanced further by the incorporation of real-time retraining and enhancing detection with newer machine learning models.

In conclusion, our project successfully demonstrated the practicability of an intelligent and efficient firewall solution enhanced by AI. It bridges the gap between dynamic, learning-based defense systems and traditional firewall rule-based methods. Despite the potential for improvement, the paradigm that this deployment provides presents a solid foundation for future improvements and innovations in intelligent network security systems.

References

- Hasan, M., & Malik, T. (n.d.). AI-enhanced VPN security framework: Integrating open-source threat intelligence and machine learning to secure digital networks. School of Informatics and Cybersecurity, Technological University Dublin.
- Chakraborty, P., Rahman, M. Z., & Rahman, S. (2019). Building new generation firewall including artificial intelligence. International Journal of Computer Applications, 178(49).
- Zhang, Y. (2023). Research on the application of artificial intelligence technology in the field of network security. Journal of Artificial Intelligence Practice, 6(6). Clausius Scientific Press.
- Efeoğlu, E., & Tuna, G. (n.d.). Classification of firewall log files with different algorithms and performance analysis of these algorithms. Unpublished manuscript, Kutahya Dumlupinar University & Trakya University.
- Armoogum, S., & Mohamudally, N. (2021). A comprehensive review of intrusion detection and prevention systems against single flood attacks in SIP-based systems. International Journal of Computer Network and Information Security (IJCNIS), 13(6), 13–25.
- Ahmadi, S. (2023). Next generation AI-based firewalls: A comparative study. International Journal of Computer (IJC), 49(1), 245–262.
- Zhou, Y., & Liang, Y. (n.d.). Application of artificial intelligence technology in network security. Unpublished manuscript, Southwest University & Chongqing Yucai Secondary School.

Appendix

Evaluation Metrices

One-class SVM model trained only on benign traffic from the current network environment. In this model, aberrant traffic patterns are highlighted as deviations from a learned baseline.

```
Classification Report (One-Class SVM):
              precision    recall  f1-score   support

           0       0.82      0.80      0.81     19061
           1       0.58      0.55      0.56    288198

    accuracy                           0.56    307259
   macro avg       0.70      0.67      0.69    307259
weighted avg       0.61      0.56      0.58    307259

Confusion Matrix (One-Class SVM):
 [[     15249  3812]
 [     129889 158309]]
```

Figure 27: *One-Class SVM Model Performance Metrics*

A Supervised Classification Model was trained on CIC IDS datasets to detect common threats like DoS, brute force, and port scanning. This model classifies fraudulent traffic using patterns learned from training data.

```
Classification Report (Gradient Boosting):
              precision    recall  f1-score   support

           0       0.70      0.50      0.58     19061
           1       0.97      0.99      0.98    288198

    accuracy                           0.96    307259
   macro avg       0.84      0.74      0.78    307259
weighted avg       0.96      0.96      0.96    307259

Confusion Matrix (Gradient Boosting):
 [[    9531  9520]
 [     4085 284113]]
```

Figure 28: *Gradient Boosting Supervised Model Performance Metrics*

```
Classification Report (Combined One-Class SVM and Gradient Boosting):
              precision    recall  f1-score   support

           0       0.79      0.80      0.80     19061
           1       0.62      0.88      0.73    288198

    accuracy                           0.87    307259
   macro avg       0.71      0.84      0.77    307259
weighted avg       0.65      0.87      0.75    307259

Confusion Matrix (Combined One-Class SVM and Gradient Boosting):
 [[    15249  3812]
 [     34584 253614]]
```

Figure 29: *Comparative Performance Metrics of Combined Models*

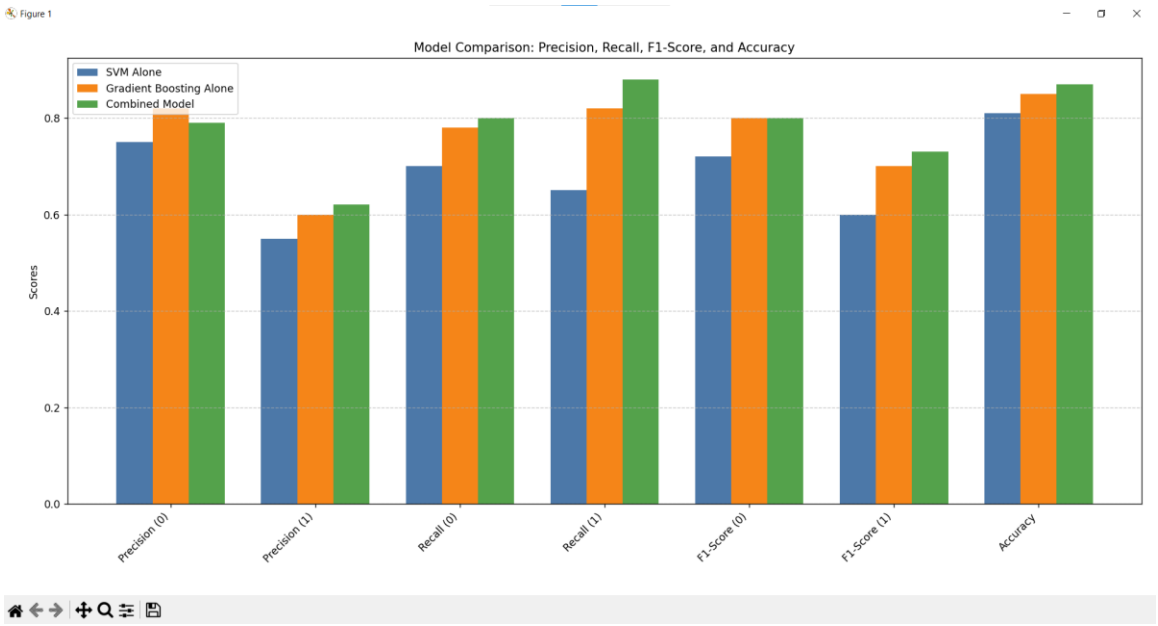

Figure 30: *Model Comparison*

GitHub

https://github.com/33073-umar/NeuraWall

GitHub is a web-based software collaboration platform that facilitates collaborative software development by using the Git version control system. It allows programmers to maintain a record of structure and save changes to their code bases. Git is available to developers of any skill level due to GitHub's intuitive interface.

Regularly Using and Managing GitHub While the Project

GitHub was utilized and managed as follows throughout the project:

- **Repository Creation:** Our project now has a GitHub repository of its own that serves as the central point for all the project files and documentation.
- **Regular Commits:** The team was encouraged to publish regular, descriptive commit messages for changes on a regular basis. This process ensured that changes could be easily traced, and the history of the project was well documented.
- **Pull Requests:** The team made pull requests before adding changes to the main branch. To enhance collaboration and quality of code, one of the team members reviewed the work, provided feedback, and discussed improvements.
- **Documentation Changes:** Documentation was updated regularly within the repository as the project changed. This was done for the installation process, usage, and the development notes, so the current information is always available to the team and potential future contributors.
- **Updates and Regular Meetings:** Regular meetings were conducted among the team to review progress, pull requests and solve any problems that might come.