A Major Project Mid-Term Report on

Praharak - An Enhanced Threat Detection and Protection

Submitted in Partial Fulfillment of the Requirements for

The Degree of **Bachelor of Engineering in Computer Engineering**

Under Pokhara University

Submitted by:

Aishu Gyawali, 191306

Bishal Poudel, 191318

Date

28 May, 2024

Department of Computer Engineering

NEPAL COLLEGE OF INFORMATION TECHNOLOGY

Balkumari, Lalitpur, Nepal

ABSTRACT

Cyberattacks are becoming increasingly frequent and complicated, posing a substantial danger to individuals and businesses throughout the world. Traditional cybersecurity solutions frequently fall short of the adaptive techniques used by hackers, leaving networks open to attack. In response, this project "Praharak - An Enhanced Threat Detection and Protection" provides a complete security system for detecting and preventing unauthorized access, malicious actions, and other network security risks. Praharak integrates IDS, IPS, and SIEM features to provide a more comprehensive approach to network security. Praharak uses deep learning algorithms to analyze network traffic patterns, detect irregularities that indicate threatening activities, and take required steps in real time. This project intends to create an open-source, user-friendly interface for system configuration, monitoring, and reporting that will be accessible to a wide variety of users. Praharak aims to improve cybersecurity measures and guard against emerging threats by offering a low-cost, AI-powered alternative to standard network security solutions.

Keywords: Network security, Intrusion Detection and prevention, Intrusion Prevention System, Security Information and Event Management, Deep learning

Table of Content

1. Introduction

Praharak - An Enhanced threat detection and protection is a security solution which is designed to detect and prevent unauthorized access, malicious activities, and other security threats within a network with the integration of IPS, IDS and SEIM.

IDS stands for Intrusion Detection System, which is a security system designed to monitor network or system activities for malicious activities or policy violations. IPS stands for Intrusion Prevention System, which is a security tool that monitors network and/or system activities for malicious or unwanted behavior and can react, in real-time, to block or prevent those activities. SIEM stands for Security Information and Event Management, which is a software solution that aggregates and analyzes activity from many different resources across an entire IT infrastructure. SIEM collects security data from network devices, servers, domain controllers, and more, and analyzes that data to identify and respond to security threats.

Integrating SEIM, IDS and IPS capabilities enables a more comprehensive approach to security. IDS can detect suspicious behavior, but IPS can take proactive steps to prevent such risks and SIEM enables centralized logging, analysis, and correlation of security events across several network devices. Praharak uses deep learning models to evaluate network traffic patterns, detect abnormalities indicating hostile behavior, and initiate necessary actions since Deep learning algorithms have shown promising results for anomaly detection and threat categorization.

2. Problem Statement

Hackers target computers connected to the internet every 39 seconds increasing the frequency and complexity of cyberattacks that impact one in three people daily [1]. These assaults, which mostly target network endpoints, present a serious problem for people and enterprises everywhere. Even with advancements in cybersecurity, traditional measures typically fall short against adaptive strategies used by hackers, leaving the network vulnerable to exploit.

3. Project objective

The primary objective of the 'Praharak' project is

- To develop an open-source network security system by using deep learning algorithms, which is capable of detecting and preventing a wide range of security threats in real-time.
- To create a user-friendly interface for a system configuration, monitoring and reporting by integrating IDS, IPS and SEIM functionalities.

4. Scope and Limitation

4.1 Scope

The system's scope includes real-time analysis of network activity, which allows for continuous monitoring to detect and respond to any threats. Furthermore, its use spans other sectors, including companies, banking institutions, workplace and industries providing a powerful security solution.

4.2 Limitation

As we are limited to resources like representative datasets, that might encounter false positives and false negatives which result in false alerts.

5. Literature review

5.1 Snort

Snort, a free and open-source NIDS/IPS, has gained significant recognition for its effectiveness and flexibility.Developed by Martin Roesch in 1998,Snort has evolved as a versatile significant tool in network security for two years.It is rule based detection mechanism.

Snort is based on the packet capture library (libpcap), a system-independent interface for capturing traffic that is widely used in network analyzers [2]. Snort monitors network traffic and compares it against a Snort rule set defined by users in a config file. It applies these rules to packets in network traffic and issues alerts when it detects any anomalous activity.

Snort finds widespread application across diverse sectors of cybersecurity. It is extensively deployed in enterprise networks, where it acts as a vigilant guardian, detecting and thwarting unauthorized access attempts, malware intrusions, and other cyber threats in real-time. Government agencies and defense organizations rely on Snort to safeguard critical infrastructure and national security interests, detecting espionage activities and defending against cyber attacks.Snort serves as a valuable tool for cybersecurity research and development, enabling researchers and developers to analyze network traffic, devise new detection techniques, and evaluate the effectiveness of security solutions.Beyond its core functionality, researchers have sought to expand Snort's capabilities through the integration of plugins and preprocessors. These extensions enable Snort to handle a wider range of network protocols and data formats, thereby increasing its versatility and applicability in diverse network environments [3].

Snort doesn’t offer advanced machine learning capabilities,relying only on old tools like signature whereas ‘Praharak’ is an all-in-one approach that makes it easier to manage your network security and fight off new attacks, unlike older tools that rely on signatures. Praharak uses advanced machine learning to catch tricky attacks that other systems might miss.

5.2 Suricata

Suricata is also an open-source detection engine that can act as an intrusion detection system (IDS) and an intrusion prevention system (IPS). It was developed by the Open Information Security Foundation (OSIF) and is a free tool used by enterprises.multi-threaded. Meaning that the tool can use multiple cores at once, allowing for greater load balancing. This allows us to process more data without dialing back on the number of rules we implement.

Suricata distinguishes itself by its ability to perform both intrusion detection and prevention functions effectively . Suricata not only merely detects and alerts threats but also actively blocks malicious traffic. Furthermore, Suricata's proficiency in deep packet inspection enhances its efficacy in identifying and mitigating diverse cyber threats, making it well-suited for a wide range of security monitoring initiatives .

Several studies have explored techniques for optimizing Suricata rule sets to improve detection accuracy and performance. This includes research on rule prioritization, rule chaining, and reducing rule redundancy [4].

Suricata, a popular NIDS, relies on static rules and manual management, limiting its ability to adapt to new threats. Praharak, on the other hand, leverages AI for advanced threat detection and automated rule management. Additionally, Praharak is designed for effortless cloud deployment, unlike Suricata. This makes Praharak a superior choice for organizations seeking a future-proof and AI-powered intrusion detection solution.

5.3 Palo Alto

Palo Alto Networks, is a well-known cybersecurity company headquartered in Santa Clara, California,which they focus on providing next-generation firewalls that go beyond traditional firewalls by inspecting traffic based on application, user, and content. .

This enables more precise management and improved security against cyber attacks. Their solution contributes to network security by offering visibility and control over network activity for more than 60,000 business clients globally [5].

They are regarded as a prominent cybersecurity firm, providing a range of solutions to defend organizations from cyberattacks [5].Their main products are sophisticated firewalls that examine application traffic in addition to IP addresses.

Although, Palo Alto Networks' provide a full security suite, they may not be appropriate for many enterprises owing to cost and complexity . Praharak seeks to provide a possibly more cost-effective and configurable alternative, with an emphasis on deep learning-based threat detection and mitigation.

5.4 Splunk

Splunk is the world's leading operational data intelligence platform [6]. This platform is used to search, analyze and visualize the machine-generated data gathered from the websites, applications, sensors, devices etc. which make up IT infrastructure and business. Providing real time processing stands it out from other platforms. Splunk comes in different variants, and which is proprietary software. It is one example of a SEIM solution.

Throughout the course of time, there have been many tries for the development of integrated tools having capability of IDS, IPS and SEIM. One of the important attempts can be considered by Muhammad et al. [7]. They integrated IDS and SEIM for live analysis using machine learning techniques. In this project they have used ELK stack for SEIM functionality, Zeek for IDS and Slip for machine learning analysis. The proposed system by them was tested using Denial of Service (DoS) test with 344.1/sec packet.

ELK is the stack that comprises three popular projects: Elasticsearch, Logstash, and Kibana. It provides the search and analytics engine, data ingestion, visualization and thus SEIM capabilities. It can be considered as an open source alternative for SEIM [8].

Zeek on the other hand works as IDS, It can create different log files based on the flow of traffic such as conn.log, dns.log, http.log etc. Which helps for data feeding to the next stage, Slips. Which is used for real time packet analysis using machine learning algorithm Support Vector Machine and some behavioral techniques.

Beside these, Praharak provides deep learning techniques for the attack detection, provides a more flexible environment and also integrates IPS capabilities. Which itself can be considered an ideal project for security solution.

6. Methodology

Software Development Life Cycle(SDLC) provides a method for building and delivering software projects. There are various stages of SDLC such as communication, planning, design, implementation, and deployment. There are various software development models like waterfall, incremental, spiral, and prototyping, which are based on the framework provided by SDLC. Each model has their usage and criteria. We found an Incremental model also known as Rapid Application Development(RAD) is best suited for our project. This model combines elements of the waterfall model applied in an iterative fashion [9].

In our project there will be two increments, each increment will traverse through the stages of SDLC. Our end product is somehow clear but there are many requirements/features that we have to add or modify over time. In each increment, we will develop a working product and add new features to it.

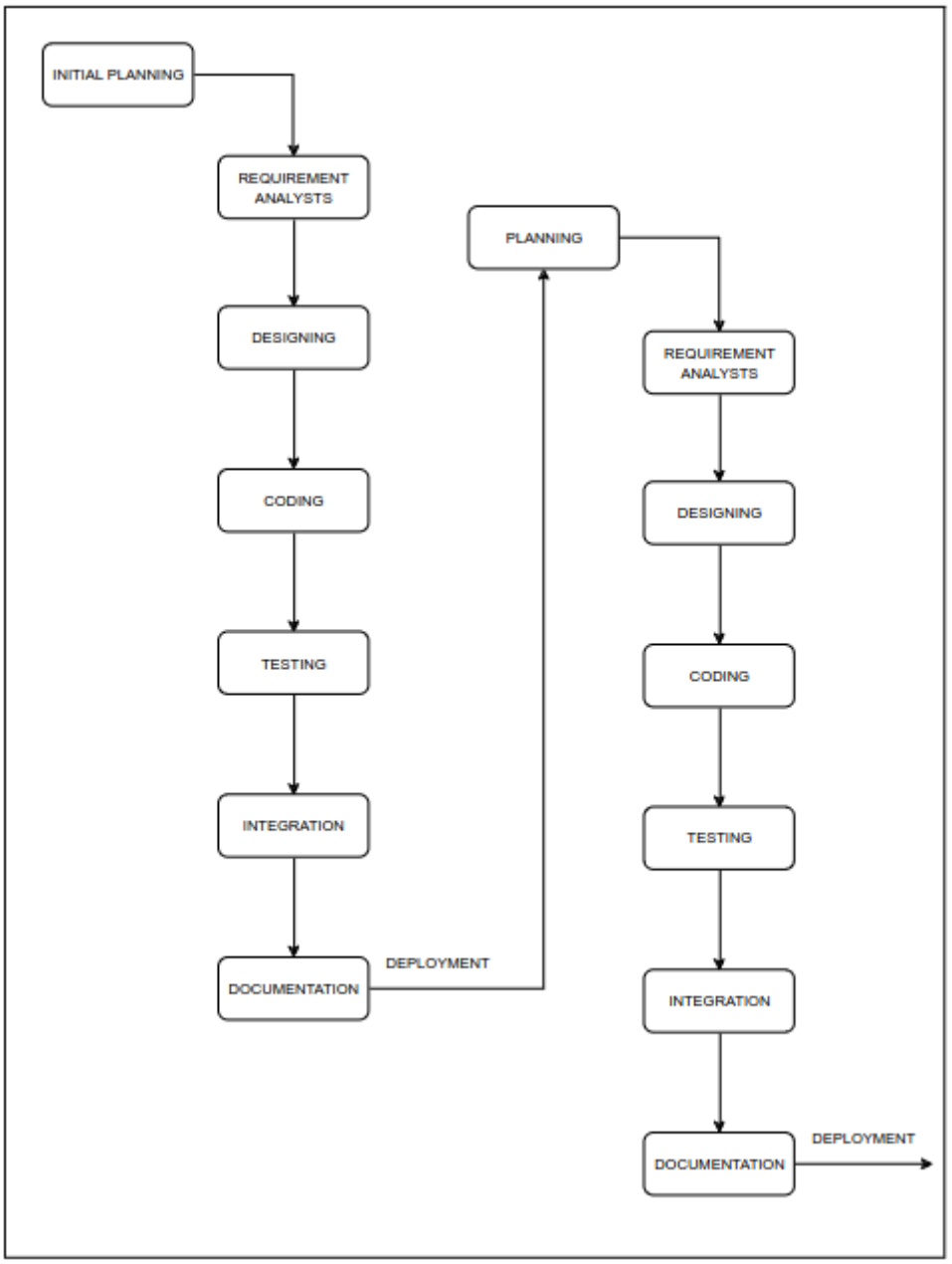


Figure 1. Methodology of Praharak (incremental model)

6.1 Technical Details

6.1.1 Data Collection and Preprocessing

In order to build an effective and reliable model for detecting and preventing network security threats, Utilizing an extensive and representative dataset is essential. For our project Praharak, we used publicly available network traffic dataset that include a mix of normal and malicious traffic.

Dataset Source

The primary dataset used in this project is the modified CICDDoS-2019 dataset, provided by the Canadian Institute for Cybersecurity (CIC).This dataset includes comprehensive records of both normal and DDoS attack traffic, making it ideal for training and validating our deep learning models.

Data Features

The dataset includes the following key features, which are critical for detecting anomalies in network traffic:

1. Source IP: The IP address of the packet's origin.
2. Destination IP: The IP address of the packet's destination.
3. Source Port: The port number at the source.
4. Destination Port: The port number at the destination.
5. Timestamp: The time at which the packet was captured.
6. Protocol: The protocol used for communication (e.g., TCP, UDP).
7. Packet Size: The size of the packet in bytes.
8. Flow Duration: The duration of the communication flow.
9. Total Fwd Packets: Total number of packets sent in the forward direction.
10. Total Bwd Packets: Total number of packets sent in the backward direction.
11. Fwd Packet Length Mean: Mean length of forward packets.
12. Bwd Packet Length Mean: Mean length of backward packets.
13. Flow Bytes/s: The number of bytes per second in the flow.
14. Flow Packets/s: The number of packets per second in the flow.
15. Label: Classification of the traffic (e.g., benign or malicious).

Number of Data Items Distribution

The dataset consists of a substantial number of data items distributed across various classes of network traffic. The distribution is as follows:

- Normal Traffic: 23% of the dataset

- Attack Traffic: 77% of the dataset, including DDoS attacks such as ‘UDP', 'MSSQL', 'Benign', 'Portmap', 'Syn', 'NetBIOS', 'UDPLag' 'LDAP', 'DrDoS_DNS', 'UDP-lag', 'WebDDoS', 'TFTP', 'DrDoS_UDP', 'DrDoS_SNMP', 'DrDoS_NetBIOS', 'DrDoS_LDAP', 'DrDoS_MSSQL' , 'DrDoS_NTP

Data Manipulation and Visualization

To process and analyze the dataset, the following popular Python libraries were used:

- pandas: Used for data manipulation and analysis, providing data structures and operations needed to manipulate numerical tables and time series.
- numpy: Utilized for numerical operations and handling arrays, supporting large, multi-dimensional arrays and matrices.
- matplotlib: Employed for data visualization, enabling the creation of static, interactive, and animated visualizations in Python.

The preprocessing steps involved the following key actions:

1. Data Cleaning: Removing any incomplete or corrupted records to ensure the quality of the dataset.
2. Feature Engineering: Creating new features or modifying existing ones to enhance the predictive power of the model.
3. Normalization: Scaling numerical features to a standard 0-1 range to facilitate faster convergence during model training.
4. Label Encoding: Converting categorical labels into numerical values that can be processed by the deep learning model for attack data we encoded 1 and for normal data we encoded 0.

6.1.2 Deep Learning and Anomaly Detection

To effectively detect anomalies and potential threats within network traffic, Praharak employs a deep learning approach utilizing Long Short-Term Memory (LSTM) networks. LSTM networks are a type of recurrent neural network (RNN) particularly well-suited for sequence prediction problems due to their ability to remember long-term dependencies.

Long Short Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a specific recurrent neural network (RNN) architecture that was designed to model temporal sequences and their long-range dependencies more accurately than conventional RNNs [1].They were introduced by Hochreiter and Schmidhuber (1997) and have since been widely adopted for various applications, including time series prediction and anomaly detection. The key feature of LSTMs is their ability to selectively remember or forget information through a series of gates.

Key Components of LSTM:

1. Cell State: The memory of the network, which carries information across time steps.
2. Forget Gate: Decides what information to discard from the cell state.
3. Input Gate: Determines what new information to add to the cell state.
4. Output Gate: Decides what information to output based on the cell state.

These gates allow LSTMs to maintain and update information over long sequences, making them ideal for analyzing sequential data like network traffic.

Model Architecture

The LSTM model used in Praharak is designed to process sequences of network traffic data and identify anomalies indicative of security threats. Below is a detailed explanation of the model architecture:

1. Input Layer:

- The input to the LSTM model is a sequence of network traffic features. Each input sample consists of multiple time steps, with each time step containing feature values such as packet size, flow duration, and protocol type.

2. LSTM Layers:

- LSTM Layer: This layer takes the input sequence and processes it to capture the temporal dependencies in the data. It outputs a sequence of hidden states.
- Dropout Layer: To prevent overfitting, a dropout layer is added after the LSTM layer, which randomly drops a fraction of the units during training.

3. Dense Layers:

- First Dense Layer: This layer processes the output of the LSTM layer and helps in interpreting the learned features. It uses the ReLU activation function.
- Output Dense Layer: This layer produces a probability score indicating whether the input sequence is normal or anomalous, using a sigmoid activation function.

Model Diagram

Below is a schematic representation of the LSTM model architecture:

Implementation Details

Deep Learning Framework: TensorFlow

- TensorFlow provides the core infrastructure for defining and training the LSTM model.

Libraries: Keras

- Keras, a high-level API running on top of TensorFlow, is used for building and training the deep learning model due to its simplicity and ease of use.

Algorithm: Long Short-Term Memory (LSTM)

- LSTM networks are chosen for their ability to handle sequential data and capture long-term dependencies.

Model Training

The model is trained using the preprocessed network traffic dataset. Key steps in the training process include:

1.Data Splitting: The dataset is split into training and testing sets, typically using an 80(80:20):20 ratio.

2. Scaling: Feature values are scaled to a standard range to ensure faster convergence during training.

3. Model Training: Train the machine learning model using the preprocessed data.

The model's performance is evaluated using metrics such as accuracy, precision, recall, and F1-score to ensure it effectively detects and prevents network security threats.

6.1.3 Real-time Packet Capture and Flow Feature Extraction

Real-time packet capture and flow feature extraction are crucial components of Praharak's network security system. This section outlines the process of capturing packets and extracting relevant flow features, along with the technologies and frameworks used for implementation.

6.1.4 Web Application Development

The frontend interface of Praharak provides users with a comprehensive dashboard for monitoring network security metrics and managing system configurations. The dashboard includes various user interface components that facilitate easy navigation, data visualization, and analysis. The following diagram illustrates the web app architecture:

Components:

1. Django Web Framework: Django serves as the primary frontend framework for developing the web-based interface. It provides a robust set of tools and features for building secure and scalable web applications.
2. HTML, CSS, JavaScript: HTML, CSS, and JavaScript are used to create the structure, style, and interactivity of the web interface, respectively. HTML defines the content and structure of web pages, CSS styles the appearance of elements, and JavaScript adds dynamic behavior and functionality.
3. Views and Templates: Django's views and templates handle the logic for rendering dynamic content and processing user input. Views interact with the backend to retrieve data and render HTML templates, which are then sent to the user's web browser for display.
4. Chart.js: Chart.js is a JavaScript library for creating interactive and customizable charts and graphs. It allows developers to generate various types of charts, including line charts, bar charts, pie charts, and more, to visualize network security metrics such as traffic volume, threat levels, and anomaly detection results.

Dashboard Overview:

The dashboard is divided into multiple sections, each focusing on different aspects of network security monitoring and analysis. The main sections include:

1. Network Traffic Visualization: This section displays interactive charts and graphs that visualize network traffic patterns over time. Users can view data on traffic and traffic volume at real time. The visualization provides insights into overall network activity with respect to time and helps identify anomalies or suspicious behavior.
2. Log Sources Table: In this section, there is a table listing different log sources within the network. Each log source is associated with specific information such as IP address, protocol, and remarks. The remarks column indicates the status or classification of each log source, categorizing them as normal, or malicious based on their behavior.If malicious it will sign danger.
3. Attack vs. Normal Data Barchart: This section provides a comparison between attack data and normal network traffic. Users can visualize the distribution of network traffic, including normal traffic, and malicious. This comparison helps users identify potential security threats and prioritize response actions accordingly.

6.2 System Architecture

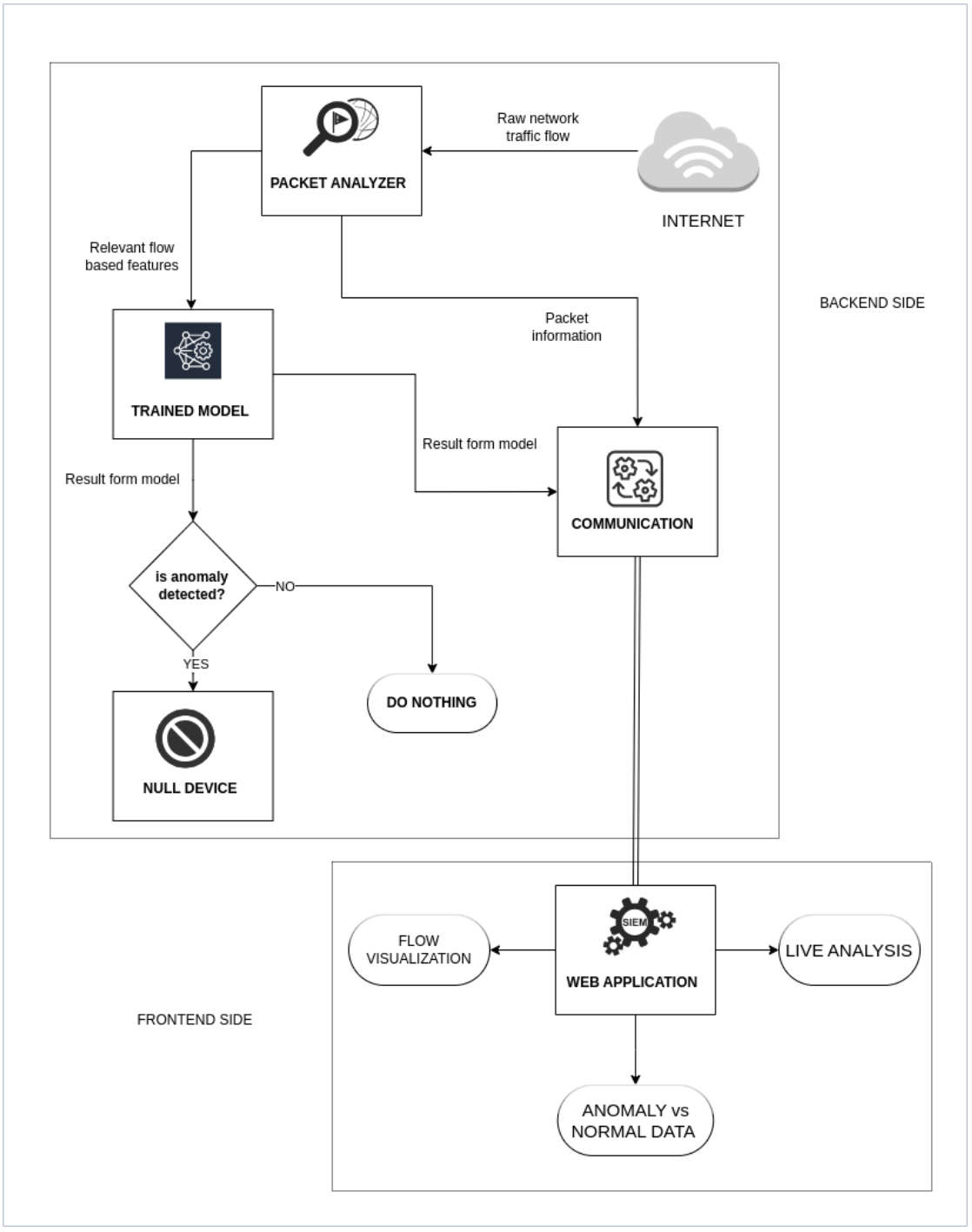


Figure 2. Generalized System Architecture

6.3 Use case diagram

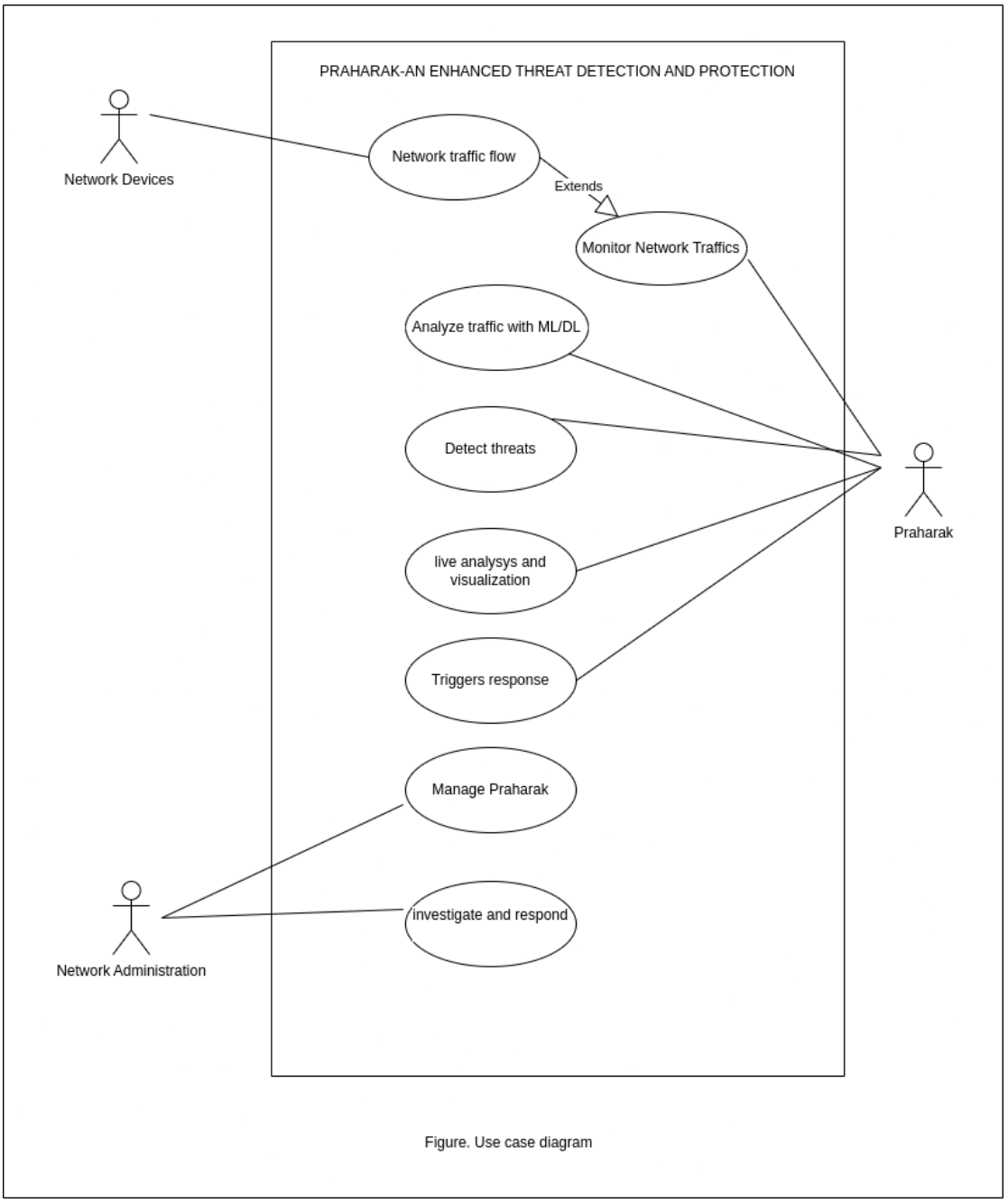
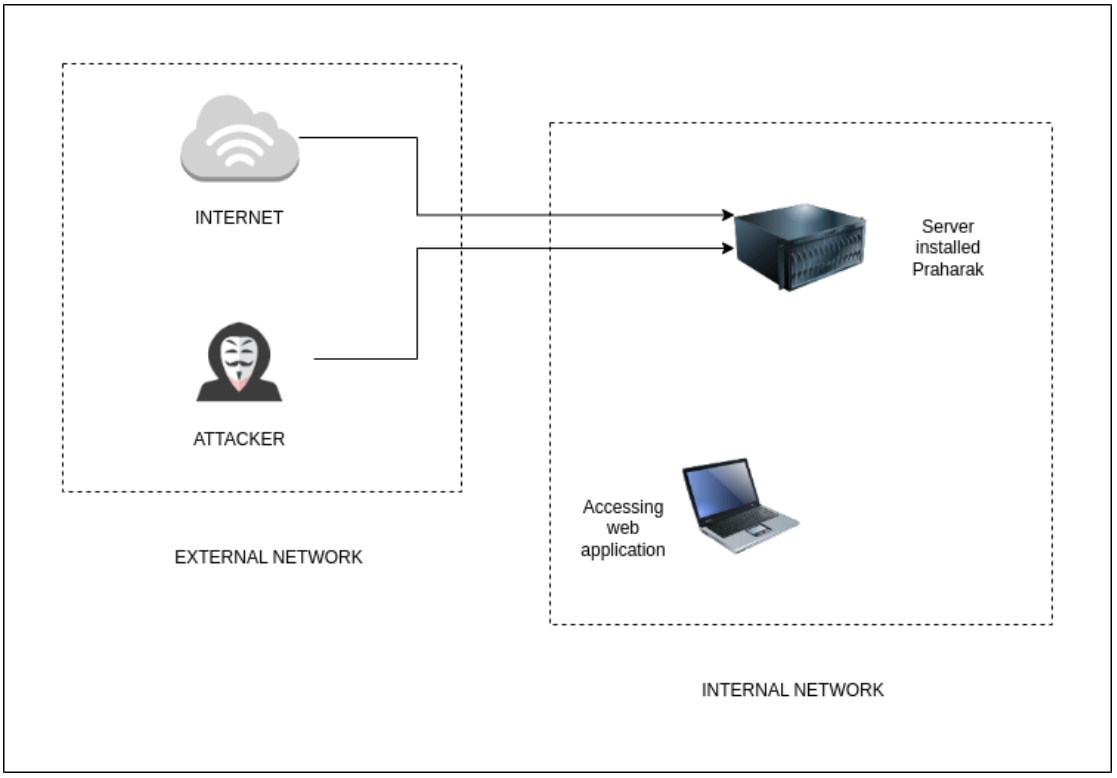


Figure 3. Use Case diagram

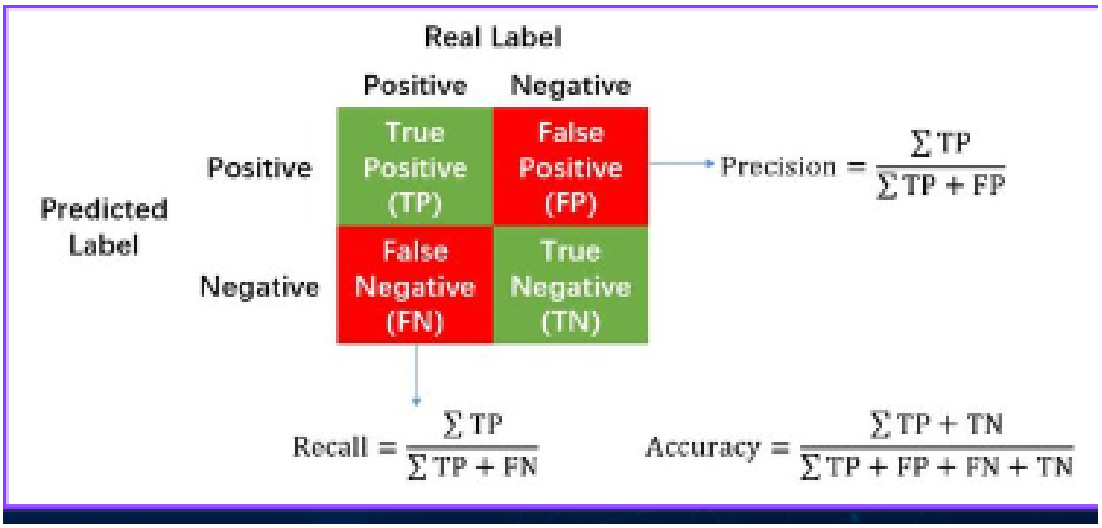
6.4 Deployment Diagram



7. Performance Analysis and Validation

7.1 Performance Analysis

7.1.1 Confusion Matrix



The performance of a classification model is frequently assessed using a table called a confusion matrix. We may assess the overall accuracy of the model by looking at the comprehensive breakdown of the number of occurrences that the model identified correctly or erroneously.

1. True Positives (TP): This is the number of positive cases that the model accurately classifies.

2. False Positives (FP): This is the quantity of negative examples that the model mistakenly labeled as positive.

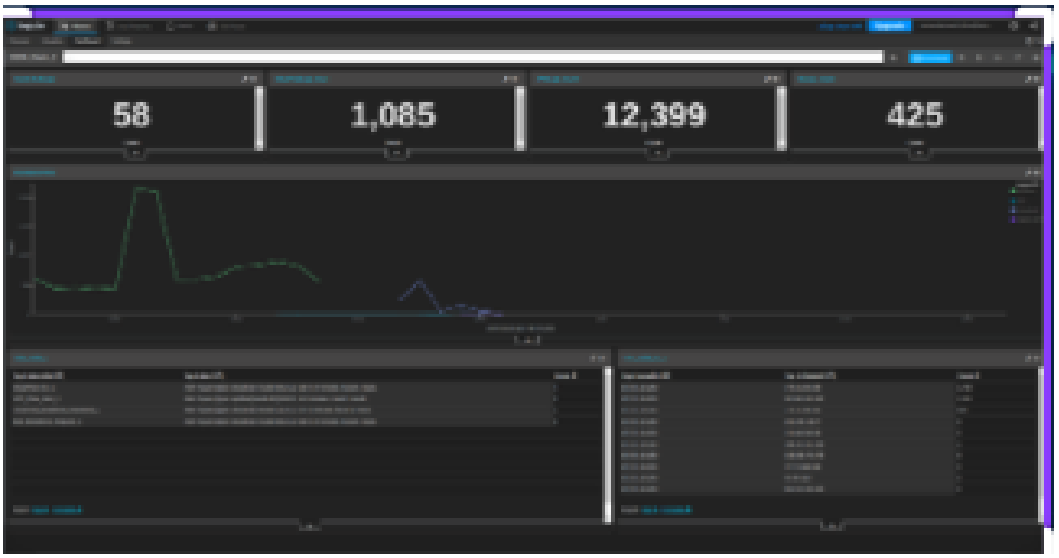
3. False Negatives (FN): This is the quantity of positive examples that the model misclassified as negative.

4. True Negatives (TN): This is the quantity of negative examples that the model accurately categorized .

Method:

- Dataset: Use a combination of publicly available network traffic datasets (Modified CICDDoS-2019) and synthetic datasets that include a mix of normal and malicious traffic.
- Evaluation: Perform cross-validation by splitting the dataset into training and testing sets(80(80:20):20). Use metrics to evaluate detection performance.

7.1.2 Processing Speed



Assess the real-time capabilities of Praharak in detecting and responding to threats.

Method:

- **Simulations:** Simulating various network attack scenarios to assess Praharak's detection and prevention capabilities.

- **Real-world Traffic Analysis:** Analyzing real network traffic containing both malicious and benign activity to evaluate Praharak's performance in a practical setting.

7.2 Validation

7.2.1 Testing

Praharak's functionalities can be validated through:

- **Unit Testing:** Testing individual components of IDS and SEIM the system to ensure they function as intended.
- **Integration Testing:** Verifying how different components interact and work together.
- **Penetration Testing:** Simulating real-world attacks at real time to assess Praharak's ability to defend against them.

7.2.2 Evaluation

The validation process should determine if Praharak meets its design goals, such as:

- **Accuracy:** Praharak effectively detects and prevents a broad range of threats with minimal false positives and negatives.
- **Real-time Performance:** Praharak can analyze and respond to threats quickly enough to prevent damage.
- **Efficiency:** Praharak utilizes system resources efficiently without compromising performance.

8. Workdone So far

- Trained LSTM deep-learning model for DDoS attack detection
    - Collected datasets (modified CIC-DDoS 2019)
    - Preprocessed and extracted relevant features.
    - Trained, validated and tested the model.

- Designed user interface to visualize the packet flow and result.
    - Used chart.js library for visualization.
    - Web socket is configured for data flow.

- Captured real-time packet and extracted relevant features.
    - Raw packet was captured using socket.
    - Network information was extracted from the raw packets.
    - Forward and backward packets were identified and created a session for the individual communication.
    - At the end of session flow statistics was calculated.

- Extracted data features were fed to the LSTM model and detected whether the data were anomalous or normal in real time .

9. Task Remaining

- Integration of web-application and backend software.
- To block the anomalies packets.
- Deployment of model architecture.
- Performance testing.
- Enhancing the solution

10. Result and Discussion

The performance of Praharak was evaluated using several key metrics to determine its effectiveness in detecting and preventing network security threats. The results obtained from testing on various datasets and in different scenarios are discussed below.

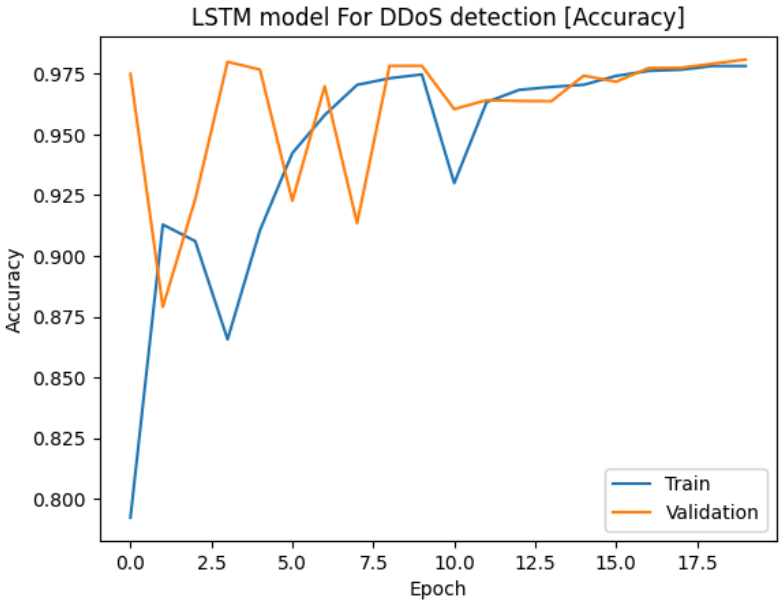
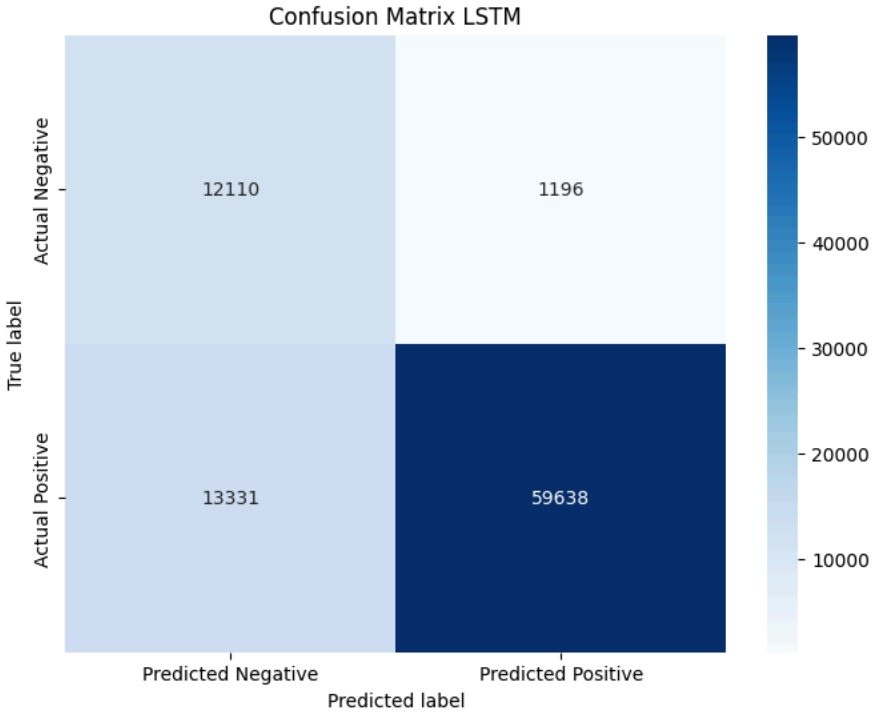
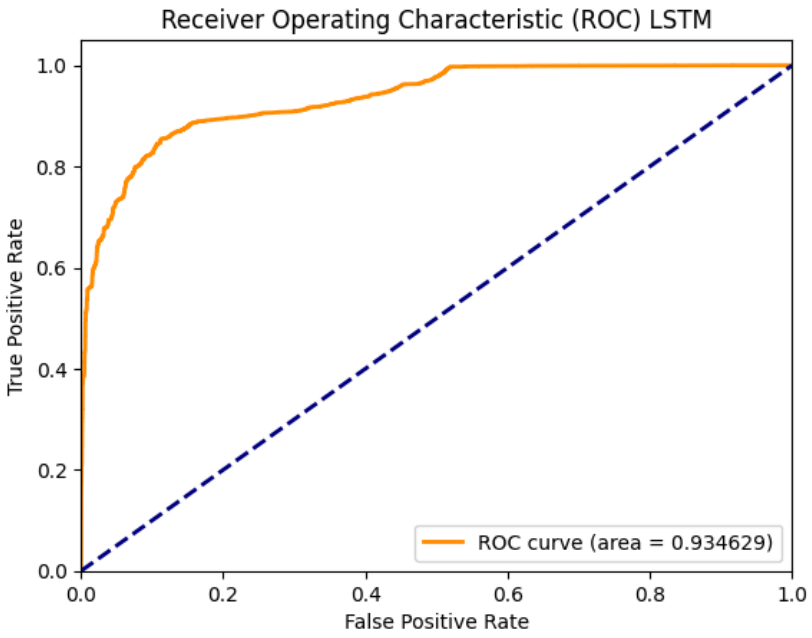


Fig 10.1 - LSTM learning curve





Performance Metrics

The trained LSTM model for Praharak achieved the following performance metrics:

- Accuracy: 83.16%
- Precision: 98.03%
- Recall: 81.73%
- F1 Score: 89.14%

Discussion:

10. Time schedule

Figure 4. Gantt chart for increment 1

Figure 5. Gantt chart for increment 2

References

[1] University of North Georgia. "Cyber Operations." [Online]. Available: https://ung.edu/cyber-operations/index.php [Accessed: April 19, 2024].

[2] "How Snort Works," CrowdStrike, Available: https://www.crowdstrike.com/cybersecurity-101/threat-intelligence/snort-rules/ ,[Accessed :14 Apr, 2024].

[3] Teng, L., et al. (2017). Enhancing Snort with a Protocol Parsing Plugin for SCADA Protocol Detection. Journal of Network and Computer Applications, 89, 40-49.

[4] A. Dulac, X. Levillain, and I. E. Magedanz, "Optimizing Snort and Suricata Rule Sets for Fast and Accurate Intrusion Detection," in 2014 International Conference on Computing, Networking and Security (CNS), pp. 152-157, 2014.

[5] Palo Alto Network [online] .Available:https://www.paloaltonetworks.com/ [Accessed:April 21,2024].

[6] Subramanian, Karun, and Karun Subramanian. "Introducing the Splunk Platform." *Practical Splunk Search Processing Language: A Guide for Mastering SPL Commands for Maximum Efficiency and Outcome* (2020): 1-38.

[7] Muhammad, Adabi & Sukarno, Parman & Wardana, Aulia. (2023). Integrated Security Information and Event Management (SIEM) with Intrusion Detection System (IDS) for Live Analysis based on Machine Learning. Procedia Computer Science. 217. 1406-1415. 10.1016/j.procs.2022.12.339.

[8] Vazão, Ana, et al. "SIEM open source solutions: a comparative study." *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, 2019.

[9] Salve SM, Samreen SN, Khatri-Valmik N. A Comparative Study on Software

Development Life Cycle Models. International Research Journal of Engineering and

Technology (IRJET). 2018 Feb;5(2):696-700.

[10] H. Sak, A. Senior, & F. Beaufays " Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling" [2014]