TRIAGE

A PROJECT REPORT

Submitted by

ASWANTH VISWANATHAN (SNG19CS018)

ASWIN GOPAKUMAR (SNG19CS020)

ATHEESH (SNG19CS021)

C.V ANANDHRAMAN (LSNG19CS078)

to

The APJ Abdul Kalam Technological University

in partial fulfilment of the requirements for the award of the Degree of

Bachelor of Technology

In

Computer Science and Engineering

Department of Computer Science and Engineering

Sree Narayana Gurukulam College of Engineering

Kadayiruppu682311

NOVEMBER 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,
SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING,
KADAYIRUPPU, 682311

(Affiliated to APJ Abdul Kalam Technological University& Approved by A.I.C.T.E)

CERTIFICATE

This is to certify that the project report, **"TRIAGE"** submitted by **ASWANTH VISWANATHAN, ASWIN GOPAKUMAR, ATHEESH, C.V ANANDHRAMAN** to the APJ Abdul Kalam Technological University in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering during the year 2022.

Head of the Department	**Project Coordinator**	**Guided by**
(Dr.) Smitha Suresh	**Mr. Anil CB**	**Mr. Anil CB**
(Prof., CSE Dept)	Asst. Prof., CSE Dept)	(Asst. Prof., CSE Dept)

Submitted for University Evaluation on ……

University Register No……

……

……

……

DECLARATION

We undersigned hereby declare that the project report **"TRIAGE"** submitted for partial fulfilment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Asst. Prof. Anil CB.This submission represents our ideas in our own words and where ideas or words of others havebeen included, we have adequately and accurately cited and referenced the original sources. Wealso declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. We understandthat any violation of the above will be a cause for disciplinary action by the institute and/or theUniversity and can also evoke penal action from the sources which have thus not been properlycited or from whom proper permission has not been obtained. This report has not beenpreviously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Kadayiruppu
Date: 19/06/2023–

Aswanth Viswanathan
Aswin Gopakumar
Atheesh
CV Anandharaman

COURSE OUTCOMES AND PROGRAM OUTCOMES

COURSE OUTCOMES: After the completion of the course the student will be able to

CO1	Identify technically and economically feasible problems (Cognitive KnowledgeLevel: Apply)
CO2	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (CognitiveKnowledge Level: Apply)
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools &advanced programming techniques (Cognitive Knowledge Level: Apply)
CO4	Prepare technical report and deliver presentation (Cognitive Knowledge Level:Apply)
CO5	Apply engineering and management principles to achieve the goal of the project(Cognitive Knowledge Level: Apply)

Program outcomes
Engineering Graduates will be able to:
PO1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution ofcomplex engineering problems.
PO2. Problem analysis: Identify, formulate, review research literature, and analyzecomplex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3. Design/development of solutions: Design solutions for complex engineeringproblems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling tocomplex engineering activities with an understanding of the limitations.
PO6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and theconsequent responsibilities relevant

to the professional engineering practice.
PO7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate theknowledge of, and need for sustainable development.
PO8. Ethics: Apply ethical principles and commit to professional ethics andresponsibilities and norms of the engineering practice.
PO9. Individual and team work: Function effectively as an individual, and as a leader in diverse teams, and in multidisciplinary settings.
PO10. Communication: Communicate effectively on complex engineering activitieswith the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effectivepresentations, and give and receive clear instructions.
PO11. Project management and finance: Demonstrate knowledge and understandingof the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12. Life-long learning: Recognize the need for, and have the preparation andability to engage in independent and life-long learning in the broadest context oftechnological change

PROGRAM SPECIFIC OUTCOMES (PSO's)
PSO1: Shall enhance the employability skills by finding innovative solutions forchallenges and problems in various domains of CS.
PSO2: Shall apply the acquired knowledge to develop software solutions andinnovative mobile applications for various problems.

CO PO PSO MAPPING

	PO1 (Engineering Knowledge)	PO2 (Problem Analysis)	PO3 (Design/Development of solutions)	investigations of complex problems)	PO5(Modern tool usage)	PO6(The Engineer and Society)	PO7(Environment and Sustainability)	PO8(Ethics)	PO9(ndividual and team work)	PO10(Communicatio n)	Management and Finance)	PO12(Life long learning)	PSO1 findding innovative solutions	PSO2 Software evelopement
CO1	2	2	2	2		2	2	2	2	2	2	2	2	2
CO2	2	2	2	2	2	2		2	2	2	2	2	2	2
CO3	2	3	3	3	3	2	2	2	3	3	2	2	3	3
CO4	2	2	2	2	2			2	2	2	2	2		2
CO5	3	3	3	3	3	2	2	2	2		3	3	3	3
AVERAGE	2.2	2.4	2.4	2.4	2.5	2.0	2.0	2.0	2.2	2.25	2.2	2.2	2.5	2.4

PO PSO ATTAINMENT AND JUSTIFICATION

PO	Attained point (0/1/2/3)	Justification
PO1	2	We have used the mathematics of neural networks, activation functions in order to find the best architecture for our purpose. The flowcharts we used to draw the relations, data flows, etc. are the basics of computer science engineering fundamentals.
PO2	3	Reviewed some research literature on advancements in the field of automobile and analyzed the problems related to automobile industry.
PO3	3	Activity diagrams, data flow diagrams, architecture diagrams, use case diagram have been constructed to visualize all the processes and data flow of the system.
PO4	2	Data had been collected from a reputed source. Several deep learning model architectures had been proposed and all of them are trained to get the better that suits the data.
PO5	3	Modern deep learning tools such as colaboratory by Google has been used to train the deep learning model. Google provide free RAM and GPU for training the machine learning models on the cloud. Anaconda which is the latest and most popular software for data science was helpful in visualizing the charts.
PO6	3	Our driver monitoring system uses deep learning to predict whether the driver is distracted/drowsy or not. Deep learning techniques has much potential to learn from the given data than of traditional logical methods.
PO7	0	
PO8	2	We have attached all the materials that we've reviewed. We could understand the disadvantages of the existing system and rectify those in our system.
PO9	3	Weekly discussion about the project via video conferencing, monthly physical meetings to know what all things should be added or removed. Assign task to team members, getting feedback from the members and updating the project.
PO10	3	Weekly presentation to the guide, class members about how far we've reached, what all things are we updating, what all things are we planning to do in the next week are discussed. Preliminary presentation to the guides and project coordinator, schedule of our work has been submitted successfully.
PO11	3	Managing tasks to team members has been efficiently done by the project manager. Overall budget for our project has been successfully and carefully planned in order to efficiently use our funds.
PO12	3	We have learned so many technologies, topics, methods that was not a part of our academic curriculum. So many experimental methods were done by us in-order to find the best one that fits our problem.

PSO1	3	We have used the software and libraries which are currently being used in the industry for design and research purpose. No trail and error method was followed for software architecture design. Proper diagrams such as activity diagram, use case diagram, sequence diagrams were followed.
PSO2	3	We have learned so many technologies, topics, methods that was not a part of our academic curriculum. How to think innovatively was our first concern when starting our project. By referring blog posts, research papers, YouTube tutorial videos we have developed an ability to think innovatively to find solutions to various problems.

Table of Contents

ACKNOWLEDGEMENT

Dedicating this report to the Almighty God whose abundant grace and mercy enabled its successful completion, we would like to express our profound gratitude to all the people who had inspired and motivated us to undertake this project.

We wish to express our sincere thanks to our Head of the Department, **Prof. (Dr.) Smitha Suresh**, for providing us the opportunity to undertake this project. We are deeply indebted to our project guide **Assoc Prof Anil CB** in the Department of Computer Science and Engineering for providing us with valuable advice and guidance during the course of the project.

Finally, we would like to express my gratitude to Sree Narayana Gurukulam College of Engineering for providing me with all the required facilities without which the successful completion of the project work would not have been possible.

ABSTRACT

Triage is a firewall and router software solution that is designed to be installed on a physical computer or virtual machine to provide dedicated firewall and routing capabilities for a network. It includes a user-friendly web interface that allows for easy administration, even for users with limited networking knowledge.

One of the key features of Triage is its automated IDS (Intrusion Detection System), which uses machine learning to detect and respond to potential security threats on the network. This helps to protect against cyber-attacks and other security threats, and can alert administrators to any potential issues.

In addition to its security features, Triage also offers a range of tools for managing and optimizing network performance. This includes support for VPNs (Virtual Private Networks) and VLANs (Virtual Local Area Networks), as well as tools for monitoring network traffic and bandwidth usage.

Overall, Triage is a comprehensive solution for protecting and managing a network, and its web-based interface makes it accessible and easy to use for a wide range of users. Whether you are an experienced network administrator or a beginner, Triage has the features and tools you need to keep your network running smoothly and securely.

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1.1 GENERAL BACKGROUND

Triage is a term that originally referred to the process of sorting and prioritizing patients in a medical setting based on the severity of their injuries or illnesses. The goal of triage is to ensure that the most critically ill or injured patients receive medical attention first, in order to maximize the chances of survival and recovery.

In the context of computer networking, the term "triage" refers to the process of sorting and prioritizing incoming network traffic based on various criteria such as destination, type of traffic, and bandwidth usage. This is typically done using a network device such as a firewall or router, which is responsible for routing incoming traffic to the appropriate destination and enforcing security policies.

Triage firewall and router software is designed to provide dedicated firewall and routing capabilities for a network, and includes tools for managing and optimizing network performance as well as detecting and responding to potential security threats. It is typically installed on a physical computer or virtual machine, and includes a user-friendly web interface for easy administration. Overall, Triage is a comprehensive solution for protecting and managing a network.

1.2 OBJECTIVE

The objective of this project is to build a Network Intrusion Detection System (NIDS) using two powerful machine learning algorithms: XGBoost and Isolation Forest. The NIDS, named "Triage," aims to detect and respond to network intrusions and anomalies in real-time.

XG Boost, an optimized gradient boosting algorithm, is known for its exceptional performance in classification tasks. It can effectively handle complex and high-dimensional data, making it well-suited for identifying patterns and anomalies in network traffic.

Isolation Forest, on the other hand, is an unsupervised learning algorithm designed specifically for anomaly detection. It works by isolating abnormal instances, making it effective in identifying previously unseen and suspicious network behaviors.

By combining the strengths of XGBoost and Isolation Forest, Triage aims to provide a comprehensive and accurate network intrusion detection solution. The project will leverage the UNSW-NB15 dataset, a widely used benchmark dataset for NIDS, to train and evaluate the performance of Triage.

The ultimate goal of building Triage using XGBoost and Isolation Forest is to develop a robust and efficient NIDS that can effectively detect a wide range of network intrusions and anomalies. The project aims to contribute to the field of network security by exploring the capabilities of machine learning algorithms in improving the accuracy and efficiency of intrusion detection systems.

1.3 SCOPE

The scope of Triage firewall and router software is to provide dedicated firewall and routing capabilities for a network. This includes:

1. Protecting the network from external threats: Triage includes an automated IDS (Intrusion Detection System) that uses machine learning to detect and respond to potential security threats on the network. This helps to protect against cyber-attacks and other security threats, and can alert administrators to any potential issues.

2. Managing and optimizing network performance: Triage includes a range of tools for monitoring network traffic and bandwidth usage, as well as support for VPNs (Virtual Private Networks) and VLANs (Virtual Local Area Networks). These

tools can help administrators to optimize network performance and troubleshoot any issues that may arise.

3. Providing a user-friendly interface for easy administration: Triage includes a web-based interface that is designed to be easy to use, even for users with limited networking knowledge. This allows administrators to manage and configure the network easily, and helps to make Triage accessible to a wide range of users.

Overall, the scope of Triage is to provide a comprehensive solution for protecting and managing a network, and to make it easy for a wide range of users to access and utilize these capabilities.

1.4 OVERVIEW

The UNSW-NB15 dataset is a widely used network traffic dataset that was developed for network intrusion detection system (NIDS) research and evaluation. It was created by the Cyber Range Lab at the University of New South Wales (UNSW) in Australia. The dataset is specifically designed to capture a wide range of network traffic patterns, including both normal and malicious activities.

The dataset contains a diverse set of features extracted from network packets, network flows, and host-level logs. It includes a total of 49 features, encompassing both numerical and categorical data. The features cover various aspects of network traffic, such as protocol types, service types, source and destination addresses, packet sizes, and duration of connections.

One of the significant advantages of the UNSW-NB15 dataset is its comprehensive labeling. The network traffic instances in the dataset are labeled as either normal or belonging to one of the several attack categories, including Denial-of-Service (DoS), Probe, Remote to Local (R2L), and User to Root (U2R). This labeling allows for supervised learning approaches and the evaluation of NIDS performance based on accurately classified instances.

The relevance of the UNSW-NB15 dataset to this project is substantial. As the project aims to build a NIDS using XGBoost and Isolation Forest, the dataset provides a suitable benchmark for training, testing, and evaluating the performance of the developed NIDS, Triage. The dataset's diversity in network traffic patterns and attack types ensures that Triage can be trained to detect a wide range of intrusions and

anomalies effectively.

Moreover, the labeling of the dataset allows for the assessment of Triage's performance in terms of accuracy, precision, recall, and other evaluation metrics. The availability of labeled instances representing different attack categories facilitates the identification of specific types of network intrusions and helps assess Triage's ability to differentiate between normal and malicious traffic.

Overall, the UNSW-NB15 dataset serves as a valuable resource for training and evaluating the NIDS Triage. Its relevance lies in providing realistic and comprehensive network traffic data, enabling the project to develop an effective intrusion detection system capable of detecting and responding to various network-based attacks and anomalies

CHAPTER 2 LITERATURE SURVEY

The purpose of a literature review is to, as the name suggests, “review” the literature surrounding a certain topic area. The word “literature” means “sources of information” or “research.” The literature will inform us about the research that has already been conducted onour chosen subject.

In a paper published by Basant Subba [1] we discuss using a contemporary UNSW-NB15 dataset to train a Neural Network. The proposed NIDS framework uses Convex Logistic Regression cost functions along with stochastic gradient descent and simulated annealing to fine tune various hyperparameters of the Neural Network based NIDS classifier. Experimental results on the contemporary UNSW-NB15 dataset show that the proposed NIDS framework achieves high detection rate against a wide range of modern-day network attacks, while maintaining a relatively low false alarm rate. The proposed Neural Network based NIDS framework outperforms other NIDS frameworks based on Decision Tree, SVM and Voting Ensemble Methods. This is achieved by optimizing various hyper parameters of the proposed NIDS frameworks. An advantage of this method is Low false alarm rate and High detection rate

In a work by Dongyang Li and Daisuke Kotani[2] they have proposed a new hybrid oversampling model using GAN to improve attack detection performance in anomaly-based NIDS. It contains three main steps: feature extraction by Information Gain and PCA, data clustering by DBSCAN and data generation by WGAN-DIV. The feature extraction step produces effective features of datasets and reduces the complexity and volumes of them, then the data clustering step removes outliers of malicious samples and splits them into several smallclusters for the preparation of GAN training. And in the data generation, those attack clusters are fed to WGAN-DIV separately for training and corresponding augmented attack data are generated and added to original training sets to alleviate class imbalance problem. From the experiments in three datasets and six NIDS classifiers, this model has achieved the best F1-scorewith XGBoost, and compared with SMOTE which is a conventional oversampling method, thismodel has comparable or even better results. An advantage of this method is This model with XGBoost has achieved best F1-score

In this paper [3] An intrusion detection system (IDS) monitors a system for malicious activities and policy violations. Traditional intrusion detection techniques use either pattern matching or blacklisting. In a blacklisting IDS a firewall or a proxy server maintains a list of

malicious servers and denies access for any server in the list. On the other hand, behavior-based detection technique analyses the attack behavior, and detects attacks with a higher detection rate than the previous two types. However, all three types of IDSs are unable to detect some well-known attacks, like Drive by Download attack (DbD) , C&C traffic, and unseen malicious traffic.

The main contribution of this paper is the development of an intrusion detection mechanism that analyses natural language-based network traffic using NLP techniques and detects anomalous, potentially malicious, traffic using an ensemble-based machine learning scheme. This system provides an overview of the natural language processing and machine learning based scheme to detect network intrusion. In the first phase, natural language content of the file of HTTP requests is converted into bag of words, and the bag of words creates sentences. These vector-spaces are used to train ensemble machine learning framework, and the trained models are used to detect network intrusion as anomalies in the data.

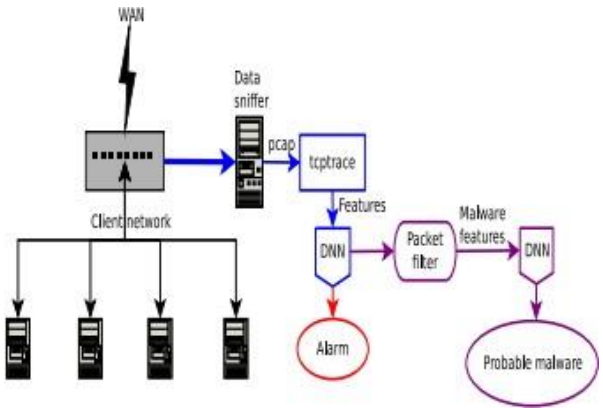


Figure 2.1: Architecture

Analyzing huge network traffic data is the main work of intrusion detection system. A well-organized classification methodology is required to overcome this issue. This issue is taken in proposed approach. Machine learning techniques like Support Vector Machine (SVM) and Naïve Bayes are applied [4]. These techniques are well-known to solve the classification problems. For evaluation of intrusion detection system, NSL– KDD knowledge discovery Dataset is taken. The outcomes show that SVM works better than Naïve Bayes. To perform comparative analysis, effective classification methods like Support Vector Machine and Naive Bayes are taken, their accuracy and misclassification rate get calculated.

SVM is a supervised ML algorithm based on the idea of max-margin separation hyper-planein *n*-dimensional feature space. It is used for the solution of both linear and nonlinear problems.For nonlinear problems, kernel functions are used. The idea is to first map a low dimensional input vector into a high dimensional feature space using the kernel function. Next, an optimal maximum marginal hyper-plane is obtained, which works as a decision boundary using the support vectors.

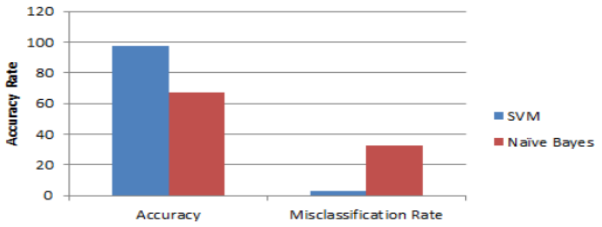


Figure 2.2 : Graph

CHAPTER 3 EXISTING SYSTEM

3.1 INTRODUCTION

Routers are networking devices that are responsible for forwarding data packets between computer networks. They are typically used to connect a local area network (LAN) to a wide area network (WAN) such as the internet, and use routing tables and protocols to determine the best path for the packets to take. Routers are an essential component of modern computer networks, as they allow devices on one network to communicate with devices on another network.

While routers are an important tool for connecting and managing networks, they are not designed to provide comprehensive protection against cyber threats. While some routers do include built-in firewalls and other security features, these are often limited in scope and may not be sufficient to protect against more advanced threats.

One of the main limitations of routers is that they are designed to focus on routing traffic rather than security. This means that they are primarily concerned with ensuring that data packets reach their intended destination, and may not be equipped to detect and respond to potential security threats.

Additionally, routers may not have the processing power or memory capacity to support advanced security features such as deep packet inspection or machine learning-based intrusion detection. As a result, they may not be able to detect or respond to sophisticated threats that may be able to bypass more basic security measures.

In order to properly protect a network, it is important to use a combination of tools and strategies. While routers can play a role in securing a network, they should not be relied upon as the sole means of protection. Instead, it is recommended to use additional security measures such as firewalls, antivirus software, and intrusion detection systems to provide a more comprehensive level of protection.

3.2 METHODOLOGY

A network router is a networking device that is responsible for forwarding data packets between computer networks. It is connected to at least two networks, typically a local area network (LAN) and a wide area network (WAN) such as the internet, and uses routing tables and protocols to determine the best path for the packets to take. Routers are used to connect networks together and allow devices on one network to communicate with devices on another network.

3.2.1 HOW A NETWORK ROUTER WORKS:

1. A device on the network, such as a computer or printer, generates a data packet that needs to be sent to another device on the network or on a different network.
2. The data packet is sent to the router, which is connected to both the source device's network (the LAN) and the destination device's network (the WAN).
3. The router checks the destination address of the data packet and consults its routing table to determine the best path for the packet to take.
4. The router forwards the data packet to the next hop on the network, which could be another router or the final destination device.
5. The data packet is delivered to the destination device, and a response packet is generated and sent back to the source device through the same process.

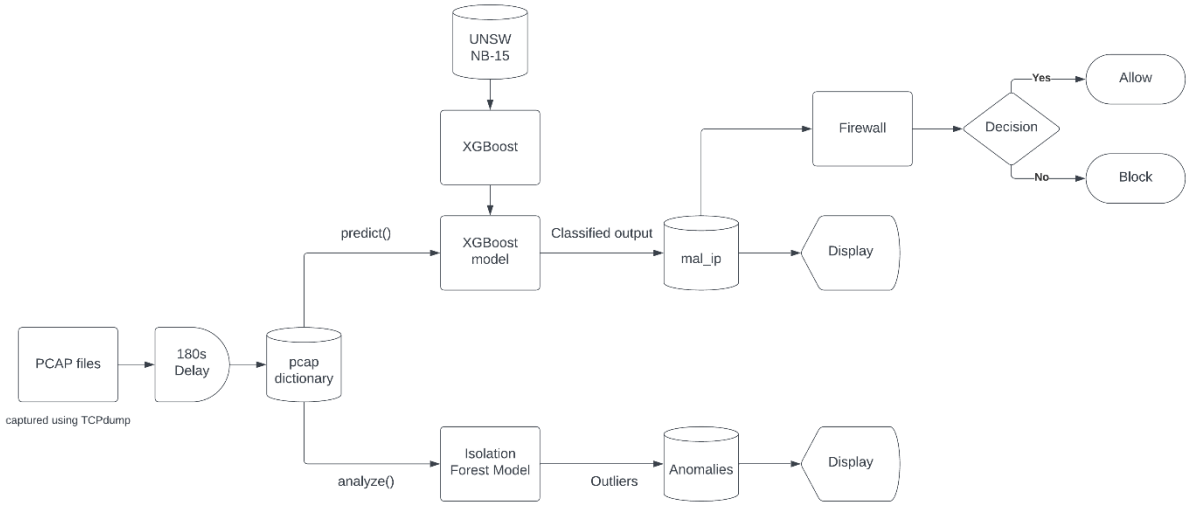
3.2.2 HOW A NETWORK ROUTER PROTECTS THE NETWORK

1. Firewall: Most routers include a built-in firewall that is designed to block unauthorized incoming traffic and allow only authorized outgoing traffic. This helps to protect against external threats such as cyber-attacks and malware.
2. Network address translation (NAT): NAT is a feature that allows a router to translate the IP addresses of devices on the LAN to a single public IP address, which is used to communicate with the WAN. This helps to protect the devices on the LAN from being directly accessible from the internet, which can reduce the risk of cyber-attacks.
3. Virtual private network (VPN): Some routers include support for VPNs, which allow users to establish secure, encrypted connections over the internet. VPNs can help to protect against external threats by encrypting data and making it more difficult for attackers to intercept or tamper with it.

Overall, network routers play a critical role in connecting and protecting modern computer networks. By forwarding data packets and enforcing security policies, routers help to ensure that devices on the network can communicate and exchange data securely and efficiently.

CHAPTER 4 DESIGN

4.1 ARCHITECTURE DIAGRAM



CHAPTER 5 PROPOSED SYSTEM

Triage is a comprehensive firewall and router software solution that is designed to protect and manage networks. It includes a range of features and tools that are designed to make it easy for administrators to configure and manage the network, even for users with limited networking knowledge.

One of the main features of Triage is its user-friendly web interface, which allows administrators to easily access and configure the firewall and router settings. The interface includes a range of tools for managing and optimizing network performance, such as support for VPNs (Virtual Private Networks) and VLANs (Virtual Local Area Networks). It also includes an automated IDS (Intrusion Detection System) with machine learning support, which helps to detect and respond to potential security threats on the network.

Triage is designed to be installed on a physical computer or virtual machine, and can be customized to meet the specific needs of the network. It is compatible with a wide range of operating systems and devices, and can be easily integrated with other security and networking tools.

Overall, Triage is a comprehensive and flexible solution for protecting and managing networks. Its web-based interface and range of features make it easy to use and accessible to a wide range of users, and its support for machine learning-based intrusion detection helps to ensure that the network is protected against the most advanced security threats. As a result, Triage is a highly effective and reliable system for protecting and managing networks.

CHAPTER 6 BACKGROUND

6.1 THE POTENTIAL OF XGBOOST AND ISOLATION FOREST IN NIDS

Machine learning algorithms, such as XGBoost and Isolation Forest, play a crucial role in enhancing the capabilities of Network Intrusion Detection Systems (NIDS). Let's discuss their potential based on the working of the system outlined above:XGBoost: XGBoost is a powerful gradient boosting algorithm known for its excellent performance in classification tasks.In the NIDS system, XGBoost is utilized to classify the captured network packets as either benign or malicious.By leveraging the comprehensive features extracted from the packet data, XGBoost can learn complex patterns and relationships between network traffic attributes.It can handle high-dimensional data effectively, which is crucial for analyzing diverse network traffic patterns.The predictive capability of XGBoost allows for accurate identification of malicious packets, contributing to the detection and prevention of network intrusions.XGBoost's regularization techniques help mitigate overfitting issues, enhancing the model's ability to generalize and adapt to new attack patterns.Isolation Forest:Isolation Forest is an unsupervised learning algorithm designed specifically for anomaly detection, making it well-suited for detecting network intrusions.In the NIDS system, the Isolation Forest algorithm is applied to identify outliers and anomalies in the captured network traffic.By isolating anomalous instances in the dataset, Isolation Forest can effectively detect previously unseen and suspicious network behaviors.It operates based on the principle that anomalies are likely to be fewer in number and more easily separable from normal instances, enabling efficient detection.Isolation Forest can handle large-scale datasets and provides a scalable approach to anomaly detection, which is crucial for real-time network monitoring and intrusion detection.By incorporating Isolation Forest into the NIDS system, potential network intrusions can be detected promptly, enabling timely responses to mitigate security risks.Overall, the combination of XGBoost and Isolation Forest in the NIDS system offers significant potential for effective network intrusion detection. XGBoost's classification capabilities enable accurate identification of malicious packets, while Isolation Forest enhances the system's ability to detect anomalies and identify previously unknown attack patterns. By leveraging the strengths of these machine learning algorithms, the NIDS system can provide robust and proactive defense against network intrusions, ensuring the security and integrity of the network infrastructure.

6.2 ADVANTAGES OF XGBOOST AND ISOLATION FOREST FOR ANOMALY DETECTION IN NETWORK TRAFFIC:

6.2.1XGBoost:

- Accuracy and Performance: XGBoost is known for its high accuracy and performance in classification tasks. It leverages gradient boosting techniques to build an ensemble of weak learners, which collectively form a strong and accurate model. In the NIDS system, XGBoost provides accurate classification of network packets as benign or malicious, helping in effective anomaly detection.

- Handling High-Dimensional Data: Network traffic data often contains numerous features, making it a high-dimensional dataset. XGBoost is well- equipped to handle such data and effectively learn complex patterns and relationships between network traffic attributes. It can capture non- linear interactions and dependencies, enhancing the detection of anomalous patterns in network traffic.

- Regularization Techniques: XGBoost incorporates regularization techniques, such as L1 and L2 regularization, to mitigate overfitting issues. These techniques help control the complexity of the model and prevent it from memorizing noise or irrelevant patterns in the dataset. As a result, XGBoost provides better generalization and robustness, enabling it to adapt to new and unseen attack patterns.

Code Implementation

```
import pandas as pd
import numpy as np
import xgboost as xgb
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import accuracy_score

# Load preprocessed dataset
df = pd.read_csv('UNSW_NB15_training-set.csv')

# Create feature and target arrays
X = df.drop('label', axis=1).values
y = df['label'].values
```

```
# Encode categorical variables
categorical_cols = [col for col in df.columns if df[col].dtype == 'object']
label_encoders = {}
for col in categorical_cols:
    label_encoders[col] = LabelEncoder()
    X[:, df.columns.get_loc(col)] = label_encoders[col].fit_transform(X[:, df.columns.get_loc(col)])

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Define xgboost parameters
params = {
    'max_depth': [3, 5, 7],
    'learning_rate': [0.01, 0.1, 0.3],
    'n_estimators': [100, 200, 300],
    'objective': ['multi:softmax'],
    'num_class': [len(np.unique(y_train))]
}

# Create xgboost classifier
xgb_model = xgb.XGBClassifier()

# Perform grid search to find the best hyperparameters
grid_search = GridSearchCV(xgb_model, params, cv=5)
grid_search.fit(X_train, y_train)

# Get the best model and print the best hyperparameters
best_model = grid_search.best_estimator_
print("Best Hyperparameters:", grid_search.best_params_)
```

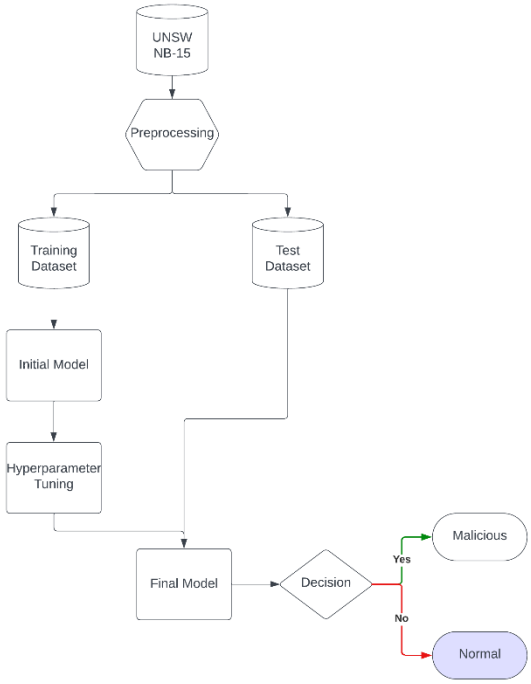
```
# Train xgboost model with the best hyperparameters
best_model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = best_model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Save trained model
best_model.save_model('unsw_nb15_xgb_model2.model')
```

XGBoost Architecture Diagram



6.2.2 Isolation Forest:

- Anomaly Detection: Isolation Forest is specifically designed for anomaly detection, making it a valuable tool for identifying abnormal patterns in network traffic. It works by isolating anomalies into individual partitions, which are easier to separate from the majority of normal instances. This unique approach allows Isolation Forest to detect anomalies efficiently, even in the presence of complex and high- dimensional data.

- Scalability: Isolation Forest exhibits good scalability, making it suitable for real-time network monitoring and intrusion detection. It can efficiently handle large-scale datasets, which is essential in network traffic analysis where the volume of data is often substantial. Isolation Forest's ability to process data in parallel and its low computational complexity contribute to its scalability.

- Independence from Distribution Assumptions: Isolation Forest does not assume any specific distribution of data. It operates based on the principle that anomalies are rare and have different statistical properties compared to normal instances. This characteristic makes Isolation Forest robust to variations in network traffic patterns and enables it to detect both known and unknown anomalies.

- Outlier Interpretability: Isolation Forest provides an anomaly score for each instance, representing the degree of abnormality. This score can be utilized to rank instances based on their level of deviation from the norm. It offers interpretability and facilitates decision-making by prioritizing the identification of highly anomalous instances in network traffic.

- By leveraging XGBoost and Isolation Forest in the NIDS system, the advantages and characteristics of these algorithms contribute to accurate classification, effective anomaly

detection, scalability, robustness, andinterpretability. This enhances the system's ability to detect and respond to network intrusions and anomalous activities, ultimately strengthening the security of the network infrastructure.

Here's an overview of how the Isolation Forest algorithm works:

Random Selection: The algorithm starts by randomly selecting a feature and a random value within the range of that feature.

Splitting Data: The data is split based on the selected feature and value. Instances with feature values below the chosen value are assigned to the left branch, and instances with feature values above the chosen value are assigned to the right branch.

Recursive Splitting: Steps 1 and 2 are repeated recursively for each resulting branch until isolated individual instances or predefined termination conditions are met.

Building the Forest: Multiple random trees are constructed using steps 1-3, each tree being built independently.

Anomaly Scoring: To determine the anomaly score of a data point, the algorithm counts the average number of splits required to isolate the point across all trees. Anomalies are expected to require fewer splits on average to be isolated compared to normal data points.

Anomaly Detection: The anomaly score is compared to a predefined threshold. If the score is below the threshold, the data point is considered normal. Conversely, if the score exceeds the threshold, the data point is classified as an anomaly.

By leveraging the isolation properties of random trees, the Isolation Forest algorithm can effectively identify anomalies or outliers in the data. Anomalies tend to have shorter average path lengths in the tree structures compared to normal instances, making them easier to isolate.

It's important to note that the algorithm does not require any prior knowledge of normal or anomalous patterns in the data. It is able to detect anomalies solely based on their deviation from the normal behavior exhibited by the majority of the data points.

CHAPTER 7 METHODOLOGY

7.1 SEQUENTIAL SEARCH

Data Collection and Preparation:

Gather the UNSW-NB15 dataset, which contains network traffic data for training and evaluation.
jj
Perform necessary preprocessing steps, such as data cleaning, normalization, and handling ategorical features.

Split the dataset into training and testing sets.

XGBoost Model Training:

Implement XGBoost algorithm to build a classification model for detecting network intrusions.
Set appropriate hyperparameters for the XGBoost model, such as learning rate, number of trees, and regularization parameters.
Train the XGBoost model using the training set.
Evaluate the model's performance on the testing set, considering metrics like accuracy, precision, recall, and F1-score.
Fine-tune the model by adjusting hyperparameters based on the evaluation results to optimize performance.

Integration of XGBoost Model into Triage:

Develop a dashboard interface using a framework like Python Flask and frontend technologies like HTML, CSS, and JavaScript.
Implement the "Analyse Network" button in the dashboard, which triggers the packet capture process using TCP dump.
Capture network packets for a duration of 3 minutes.
Pass the captured packets to the XGBoost model for classification and identification of malicious packets.
Store the malicious packets in the MAL_IP table.

Anomaly Detection using Isolation Forest:

Implement the Isolation Forest algorithm, a popular unsupervised anomaly detection algorithm.
Apply the Isolation Forest algorithm to the captured packets to identify outliers and detect anomalies.
Store the detected outliers and anomalous packets in the ANOMALIES table.

Displaying Results in index.html:

Redirect the user to the index.html page after the packet analysis process.

Design the index.html page to display a table listing the common entities present in the

MAL_IP and ANOMALIES tables, representing the detected malicious packets.

Firewall Functionality:

Develop the firewall page, which displays a table of malicious packets with options to block and unblock specific packets.

Implement the block button functionality, which triggers the firewall function to block the MAC address associated with the selected packet.

Implement the unblock button functionality to remove the block for a specific packet.

Provide additional text boxes in the firewall page for manual blocking of IP addresses and websites.

Database Integration:

Utilize a database management system like SQLite 3 to store and manage the tables, MAL_IP and ANOMALIES.

Establish a connection to the SQLite database within the backend

By following this step-by-step approach, you can successfully build Triage, a Network Intrusion Detection System using XGBoost for classification and Isolation Forest for anomaly detection. The system provides a user-friendly interface for analyzing network traffic, identifying and storing malicious packets, and offering firewall functionality to block or unblock specific packets or manually block IP addresses and websites.

7.2 PREPROCESSING FOR UNSW-NB15 DATASET

Data Cleaning:

Identify and handle missing values: Analyze the dataset for missing values in features and decide on a strategy to handle them, such as imputation or removal of instances or features with missing values.

Remove duplicate instances: Check for duplicate instances in the dataset and remove them to ensure data integrity.

Data Transformation:

Feature Scaling: Normalize the numerical features to a common scale using techniques like min-max scaling or standardization. This ensures that features with different scales do not bias the model.

Handling Categorical Features: Convert categorical features into numerical representations to make them compatible with machine learning algorithms. Techniques like one-hot encoding or label encoding can be used.

Handling Imbalanced Data:

Address Class Imbalance: Check for class imbalance in the dataset, where the number of instances belonging to different attack categories may be imbalanced. Apply techniques such as oversampling (e.g., SMOTE) or undersampling to balance the dataset. This helps prevent bias toward the majority class during model training.

Feature Selection:

Correlation Analysis: Calculate the correlation between features and the target variable (e.g., attack or normal). Remove features with low correlation, as they may not contribute significantly to the classification task.

Information Gain or Mutual Information: Assess the information gain or mutual information of features with respect to the target variable. Select features with high information gain or mutual information as they provide more relevant information for classification.

Recursive Feature Elimination (RFE): Use RFE to iteratively select the most important features based on a model's performance. Start with all features, train the model, and eliminate the least important feature in each iteration until the desired number of features is reached.

Data Partitioning:

Split the dataset into training and testing sets. The training set is used to train the XGBoost model, while the testing set is used to evaluate the model's performance.

By applying these preprocessing techniques, the UNSW-NB15 dataset can be prepared for training the XGBoost model and improving the performance of the network intrusion detection system.

7.3 FEATURE SELECTION PROCESS:

In the described working, the feature selection process refers to the selection of relevant features from the captured network packets for classification and anomaly detection. The rationale behind selecting specific features is to identify the most informative attributes that contribute significantly to distinguishing between benign and malicious packets and detecting anomalies.

The feature selection process typically involves analyzing the characteristics of the dataset and considering the requirements of the classification and anomaly detection tasks. While the specific

features chosen may vary, here are some common factors and rationale for selecting features in network intrusion detection:

Protocol-related Features:

Network protocols can provide valuable information for distinguishing between normal and malicious network traffic. Features such as protocol type (e.g., TCP, UDP), flags, packet size, and port numbers are often considered as they can indicate the nature of the communication.

Statistical Features:

Calculating statistical measures of network traffic attributes can be useful in identifying patterns or anomalies. Features like mean, standard deviation, maximum, minimum, and entropy of packet attributes (e.g., packet length, time duration, number of packets) are commonly selected.

Payload Features:

Analyzing the payload or content of network packets can reveal specific patterns or signatures associated with malicious activities. Features such as keyword presence, regular expressions, or statistical properties of payload data can be considered.

Connection-based Features:

Features derived from the characteristics of network connections can be informative for intrusion detection. These features include the number of connections, duration of connections, number of packets per connection, and rate of packet transmission.

Traffic-based Features:

Analyzing overall network traffic patterns and behaviors can provide insights into detecting anomalies. Features like the number of incoming and outgoing connections, traffic volume, and the ratio of incoming to outgoing traffic can be relevant.

Time-based Features:

Considering the temporal aspects of network traffic can help in capturing time-related patterns and detecting attacks that exhibit specific temporal characteristics. Features such as time of day, day of the week, or inter-arrival times between packets can be selected.

The selection of specific features depends on the nature of the network traffic, the types of attacks being targeted, and the performance requirements of the intrusion detection system. It is often an iterative process where different combinations of features are evaluated, and domain knowledge is used to guide the selection based on their relevance and effectiveness in capturing malicious activities and anomalies.

7.4 TRAINING TRIAGE MODELS: XGBOOST AND ISOLATION FOREST

In the described working, the XGBoost model and the Isolation Forest model are used for classification and anomaly detection, respectively, within the Triage system. Here is an explanation of the training procedure for both models:

XGBoost Training Procedure:
Input: The training dataset consists of labeled network packets, where each packet is represented by a set of selected features.

Data Preparation: The dataset is preprocessed using techniques such as data cleaning, feature selection, and feature scaling as mentioned earlier.

Train-Test Split:

The preprocessed dataset is divided into training and testing subsets, typically with a ratio of 70-80% for training and 20-30% for testing.

XGBoost Configuration:

The XGBoost algorithm is configured with hyperparameters such as learning rate, maximum tree depth, and number of boosting rounds. These parameters can be tuned using techniques like cross-validation or grid search to find the optimal values.
Model Training: The XGBoost model is trained on the training subset of the dataset using gradient boosting. The model iteratively builds an ensemble of decision trees, minimizing the loss function and optimizing the classification performance.

Model Evaluation: The trained XGBoost model is evaluated using the testing subset of the dataset. Performance metrics such as accuracy, precision, recall, and F1-score are computed to assess the model's effectiveness in classifying network packets as benign or malicious.

Model Deployment: Once the XGBoost model has been trained and evaluated, it can be deployed in the Triage system to classify incoming packets during the prediction phase.

Isolation Forest Training Procedure:

Input:

The training dataset comprises network packets, including both benign and malicious packets.

Data Preparation:

Similar to the XGBoost training procedure, the dataset is preprocessed using techniques like data cleaning, feature selection, and feature scaling.

Isolation Forest Configuration:

The Isolation Forest algorithm is configured with hyperparameters such as the number of trees and subsampling size. These parameters can be optimized using techniques like cross-validation or grid search.

Model Training:

The Isolation Forest model is trained on the preprocessed training dataset. The algorithm randomly

selects a feature and splits the data along a random threshold until the packets are isolated or until the maximum tree depth is reached.

Model Evaluation:

The trained Isolation Forest model can be evaluated using various metrics such as anomaly score, precision, and recall. Anomaly scores are computed to determine the likelihood of a packet being an outlier or anomaly.

Model Deployment:

Once the Isolation Forest model is trained and evaluated, it can be deployed in the Triage system to identify and store anomalous packets in the ANOMALIES table, as mentioned in the working description.

Both the XGBoost and Isolation Forest models are trained offline using historical network packet data. Once trained, they can be used online in real-time for classifying packets and detecting anomalies within the Triage system.

7.5 METRICES FOR TRIAGE PERFORMANCE ASSESSMENT

Accuracy:

The proportion of correctly classified instances (both benign and malicious) out of the total instances. It provides an overall measure of the model's performance.

Precision:

The proportion of correctly classified malicious instances out of all instances classified as malicious. It indicates the model's ability to correctly identify true positives and avoid false positives.

Recall (Sensitivity):

The proportion of correctly classified malicious instances out of all actual malicious instances. It measures the model's ability to detect true positives and avoid false negatives.

F1-Score:

The harmonic mean of precision and recall. It provides a balanced measure of the model's performance by considering both precision and recall.

Specificity:

The proportion of correctly classified benign instances out of all actual benign instances. It measures the model's ability to detect true negatives and avoid false positives.

False Positive Rate:

The proportion of falsely classified benign instances out of all actual benign instances. It represents the rate of false alarms raised by the model.

Area Under the ROC Curve (AUC-ROC):

A graphical representation of the model's performance across different classification thresholds. It provides an aggregate measure of the model's ability to distinguish between classes, with a higher AUC indicating better performance.

Confusion Matrix:

A table that summarizes the model's performance by showing the count of true positives, true negatives, false positives, and false negatives. It provides detailed information about the model's

classification results.

During the evaluation of Triage, these metrics can be calculated by comparing the predicted classifications with the ground truth labels for the testing set. The choice of evaluation metrics depends on the specific requirements and priorities of the network intrusion detection system.
For example, precision may be more important if avoiding false positives is a priority, while recall may be crucial for minimizing false negatives.

By analyzing these model evaluation metrics, you can assess Triage's performance in classifying network packets, detecting malicious instances accurately, and avoiding false alarms, thus ensuring effective network security.

CHAPTER 8 DATASET DESCRIPTION

8.1 UNSW-NB15 DATASET

The UNSW-NB15 dataset is a widely used dataset in the field of network intrusion detection. It was created by the University of New South Wales (UNSW) in Australia and is designed to simulate realistic network traffic scenarios. The dataset consists of network traffic captures collected from a real-world network environment, making it suitable for training and evaluating Network Intrusion Detection Systems (NIDS).

Detailed Description of the UNSW-NB15 Dataset:

Structure:

The UNSW-NB15 dataset is organized into multiple CSV (Comma-Separated Values) files, each representing a specific type of network traffic data. The dataset includes information about network flows, protocols, source and destination IP addresses, port numbers, timestamps, packet sizes, and other relevant attributes.

Features:

The dataset contains a wide range of features that capture various aspects of network traffic behavior. These features include transport layer protocol indicators (e.g., TCP, UDP), service indicators (e.g., HTTP, FTP), network flow statistics (e.g., total bytes, packets, duration), and connection-related features (e.g., SYN/FIN flags, connection state). Additionally, it provides information about network attacks and their corresponding labels.

Labels:

The UNSW-NB15 dataset includes labeled instances for different types of network traffic, including both normal and malicious activities. Each instance is assigned a label indicating its class, such as normal, denial-of-service (DoS), probe, remote-to-local (R2L), or user-to-root (U2R) attack. These labels serve as ground truth for training and evaluating NIDS models.

Relevance to NIDS:

The UNSW-NB15 dataset is particularly relevant for building and testing NIDS systems. It offers a realistic representation of network traffic and includes various attack scenarios, making it suitable for training models to detect and classify network intrusions accurately. The dataset's diversity and complexity allow for the evaluation of NIDS algorithms under different traffic conditions and attack types.

Data Size:

The dataset consists of approximately 2.5 million instances, providing a substantial amount of data for training and evaluating NIDS models. The dataset size enables the development of robust and accurate intrusion detection models capable of handling large- scale network traffic.

The UNSW-NB15 dataset provides a valuable resource for researchers and practitioners working on network intrusion detection. Its realistic network traffic captures, diverse attack scenarios, and

extensive set of features make it well-suited for training and evaluating NIDS models, including the NIDS system named "Triage" developed in this project.

8.2 STRUCTURES, FEATURES AND LABELS OF UNSW-NB15

The structure, features, and labels of the UNSW-NB15 dataset can be described as follows:

Structure:

The UNSW-NB15 dataset is organized into several CSV files, each containing specific information about network traffic instances.

These files include:

"UNSW_NB15_training-set.csv": Training dataset with labeled instances.
"UNSW_NB15_testing-set.csv": Testing dataset with labeled instances.

Additional files may contain related data, such as attack categories and descriptions

Features:

The dataset provides a wide range of features that capture various aspects of network traffic behavior. These features include:

Basic Features:

IP addresses, port numbers, timestamps, protocol types (e.g., TCP, UDP), packet sizes, and source/destination attributes.

Statistical Features:

These features represent statistical measures computed over network flows, such as the total number of packets and bytes, duration of the flow, average packet size, etc.

Content-Related Features:

These features include indicators of specific network protocols or services, such as HTTP, FTP, DNS, etc.

Connection Features:

Features related to TCP connections, such as TCP flags (SYN, FIN, etc.), connection state, etc.

Payload Features:

Some instances may include payload information, such as payload size, keywords, or patterns.

Traffic Features:

Features related to network traffic, such as the number of inbound/outbound connections, connection types, etc.

Labels:

The dataset provides labels that indicate the class or category of each network traffic instance. The labels are assigned based on the type of network intrusion or activity present in the traffic. The available labels include:

Normal:

Represents normal or benign network traffic.

Denial-of-Service (DoS):

Indicates network attacks aimed at overwhelming the target system or network resources.

Probe:

Indicates network scanning or reconnaissance activities to gather information about target systems.

Remote-to-Local (R2L):

Indicates attacks where an unauthorized remote user attempts to gain access to the target system.

User-to-Root (U2R):

Represents attacks where a local user attempts to gain root-level privileges on the target system.

The combination of features and labels in the UNSW-NB15 dataset allows for training and evaluating Network Intrusion Detection Systems (NIDS) to identify and classify various types of network intrusions and anomalies. These labeled instances serve as ground truth for building and evaluating NIDS models, such as the "Triage" NIDS developed in this project.

8.3 DATASET SUITABILITY FOR TRIAGE TRAINING AND EVALUATION

The suitability of the UNSW-NB15 dataset for training and evaluating Triage, the NIDS (Network Intrusion Detection System), can be assessed based on the following considerations:

Realistic Network Traffic:

The dataset is derived from real-world network traffic captured in a controlled environment. It includes a wide range of network activities, including normal traffic and various types of network attacks. This realism helps in training Triage to handle diverse and representative network traffic scenarios.

Labeled Instances:

The dataset provides labeled instances that classify network traffic into different attack categories and normal traffic. These labels serve as ground truth for training and evaluating the NIDS model. With the availability of labeled data, Triage can learn to distinguish between different types of network intrusions and accurately identify malicious activities.

Feature Richness:

The dataset offers a comprehensive set of features that capture various aspects of network traffic behavior. These features include basic attributes, statistical measures, connection details, content-related indicators, and more. The richness of features allows Triage to learn and leverage relevant patterns and characteristics associated with different types of network intrusions.

Scalability:

The UNSW-NB15 dataset is relatively large, containing a significant number of instances. This scalability enables Triage to handle a substantial volume of network traffic data and learn from a diverse range of examples. It also facilitates evaluating the NIDS model's performance in terms of accuracy, efficiency, and scalability.

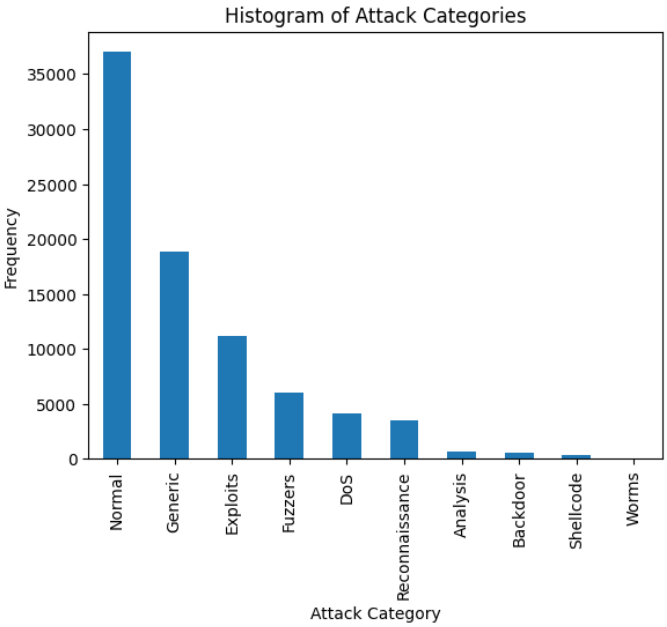
Evaluation Metrics:

The dataset includes a separate testing set, which allows for evaluating the performance of Triage in a controlled manner. By using well-defined evaluation metrics such as precision, recall, F1-score, and accuracy, the effectiveness of Triage can be assessed objectively based on its ability to correctly detect and classify network intrusions.

Overall, the UNSW-NB15 dataset provides a valuable resource for training and evaluating Triage,

the NIDS. It offers realistic network traffic, labeled instances, rich features, scalability, and proper evaluation mechanisms, enabling the development and assessment of an effective intrusion detection system.

8.3.1 Histogram of types of attack (UNSW NB 15 dataset)



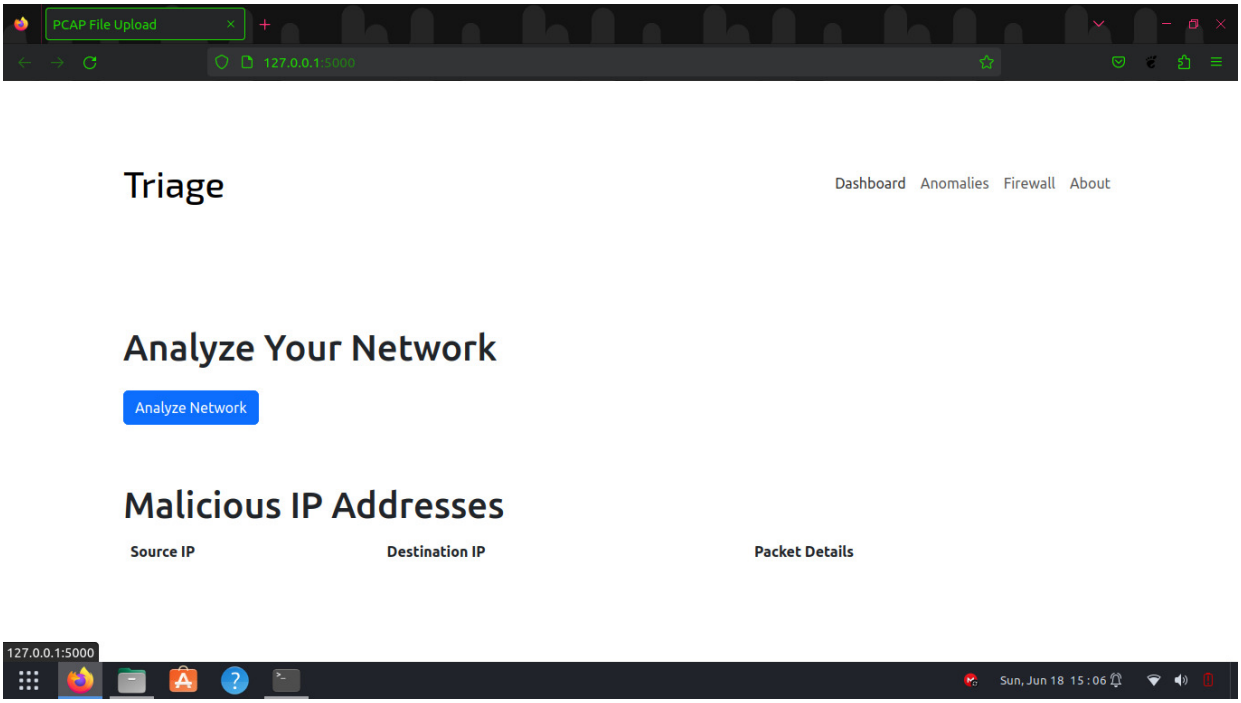
.

CHAPTER 9 TESTING

The dashboard serves as the main interface for interacting with the system. It may include various buttons, menus, and visualizations to monitor and control the network analysis process.

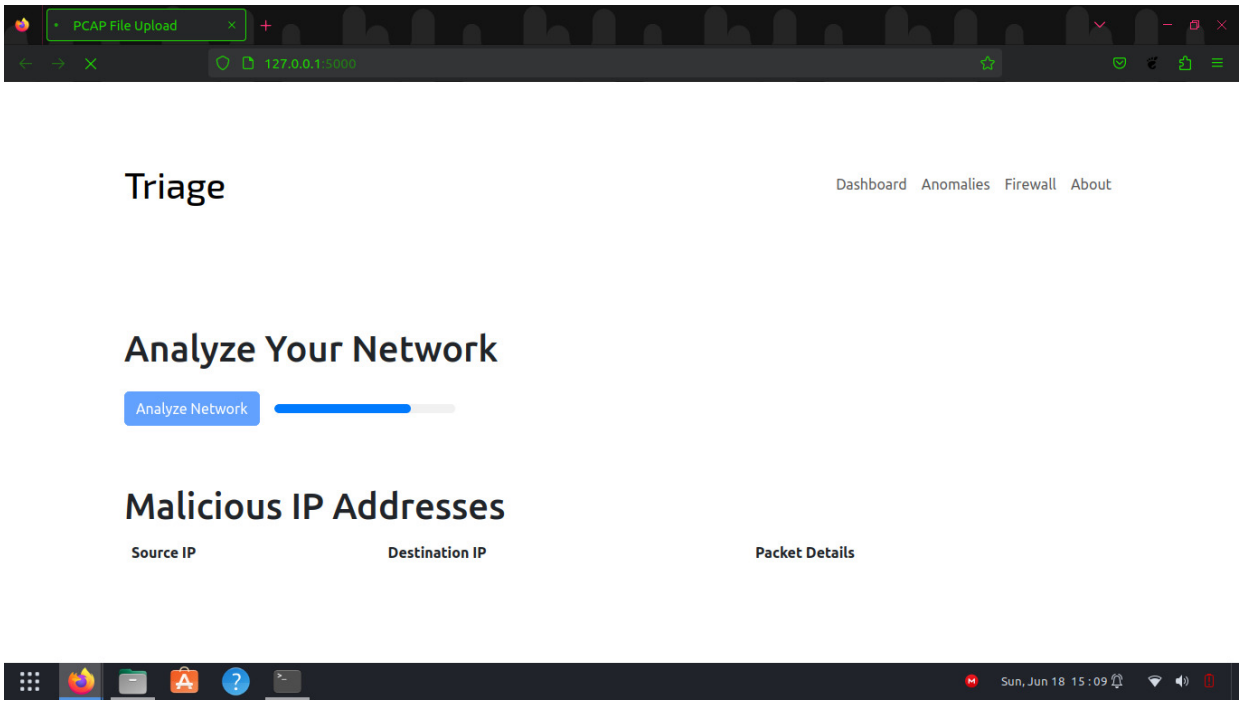
TCP dump is a widely used command-line packet capturing tool. When the "Analyze network" button is clicked, the system initiates the execution of TCP dump for a duration of 3 minutes. During this time, TCP dump captures all the packets flowing through the network, including their source and destination addresses, payload, and other relevant information.

Once the 3-minute packet capture is complete, the system calls the "predict" function. This function utilizes XGBoost, an optimized machine learning algorithm, to classify the captured packets as either malicious or non-malicious. XGBoost is trained on a dataset of known malicious and benign packets, enabling it to make predictions based on various packet features



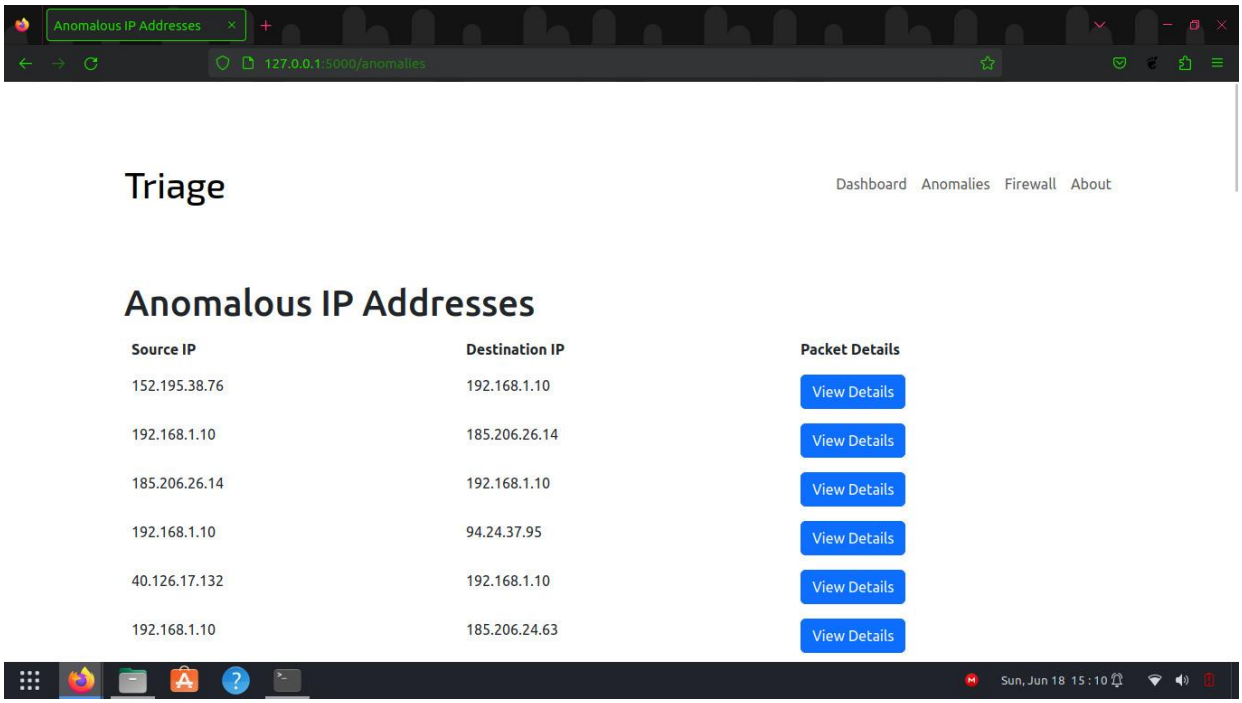
After the predict function analyzes the captured packets, any packets classified as malicious are stored in a table called MAL_IP. This table contains information about the malicious packets, such as their source IP address, destination IP address, timestamps, or other relevant attributes.

Following the predict function, the system calls the "analyse" function. This function applies the Isolation Forest algorithm, a popular unsupervised machine learning algorithm, to detect outliers among the captured packets. The Isolation Forest algorithm identifies anomalies by constructing isolation trees and measuring the average path length required to isolate a packet from the rest of the dataset.



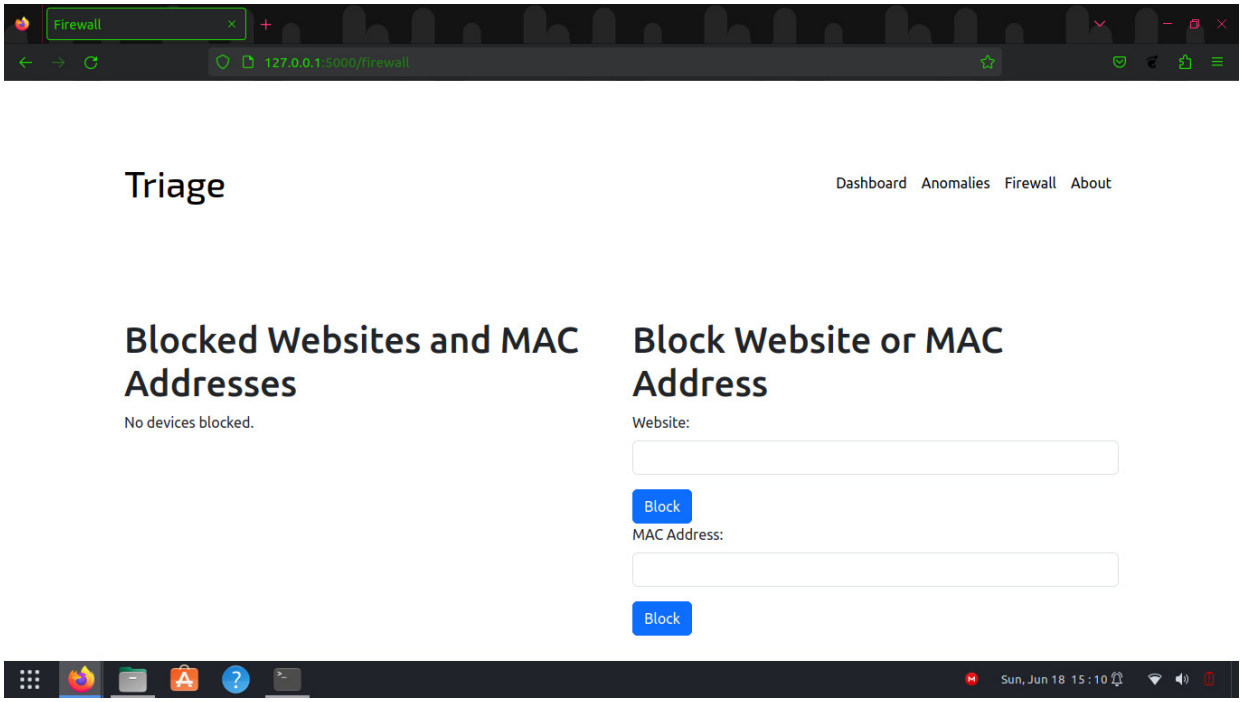
Any packets identified as outliers by the analyse function are stored in a table called ANOMALIES. This table contains information about the anomalous packets, including their attributes and the anomalies' scores or other measures.

Once the analysis is complete, the system returns to an HTML page called index.html. This page is rendered in the web browser and displays a table that combines the information from the MAL_IP and ANOMALIES tables. It presents a list of common entities found in both tables, representing the packets that are considered malicious.

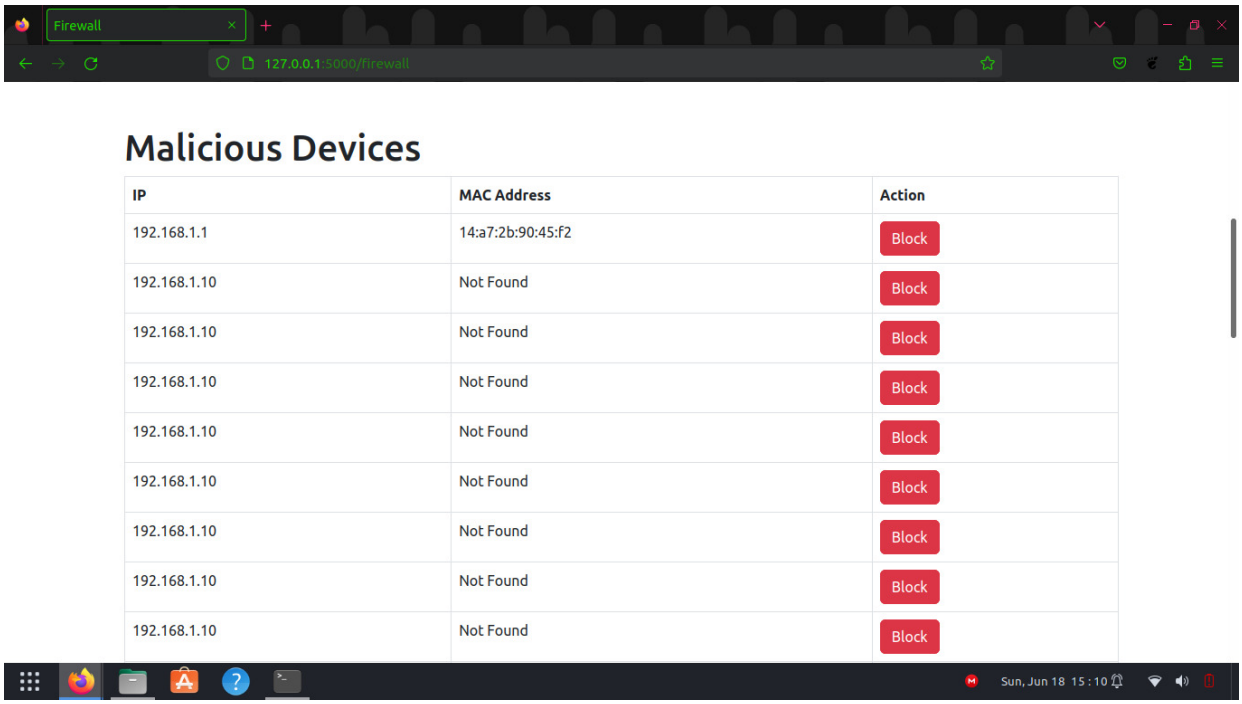


The system includes a separate page called FIREWALL, which focuses on network security and packet filtering. It provides options for blocking and unblocking specific packets based on their characteristics.

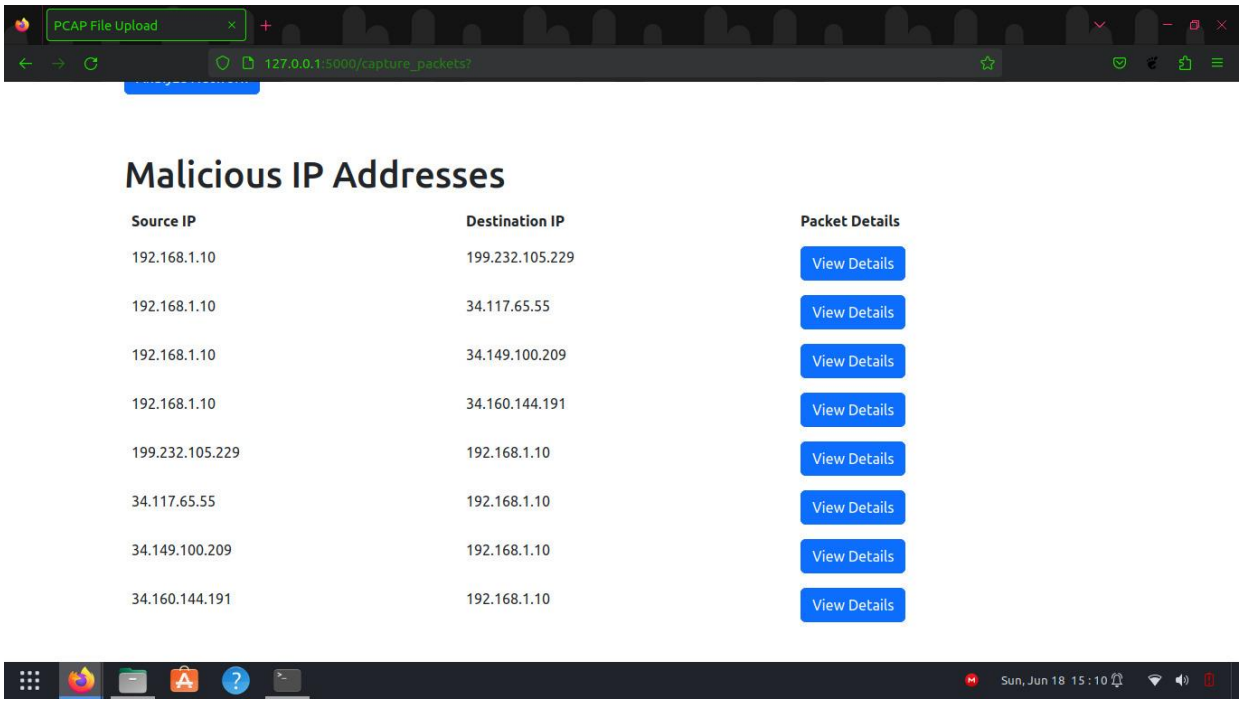
Within the FIREWALL page, when the "block" button is clicked for a particular packet, the system triggers a process. This process extracts the MAC address of the packet and passes it to the firewall function for further handling. The firewall function then utilizes IP tables, a powerful firewall management tool in Linux, to implement the blocking mechanism, effectively preventing further communication from or to the blocked packet.



The system utilizes IP tables for its firewall functionality. IP tables is a flexible and feature-rich firewall configuration tool available in many Linux distributions. It allows the manual blocking and unblocking of IP addresses and websites through text boxes provided in the firewall function. This enables administrators to define specific rules and filters to control network traffic based on various criteria.



In summary, the system captures network packets using TCP dump, classifies them as malicious or non-malicious using XGBoost, identifies outliers using the Isolation Forest algorithm, stores malicious packets in the MAL_IP table, stores outliers in the ANOMALIES table, presents the results in index.html, and incorporates a firewall functionality using IP



CHAPTER 10 CONCLUSION

In conclusion, Triage is a comprehensive firewall and router software solution that is designedto protect and manage networks. Its user-friendly web interface and range of features make it easy to use and accessible to a wide range of users, and its support for machine learning-basedintrusion detection helps to ensure that the network is protected against the most advanced security threats. As a result, Triage is a highly effective and reliable system for protecting and managing networks. Its flexible design and compatibility with a wide range of operating systems and devices make it a suitable solution for a wide range of environments andapplications. Overall, Triage is an innovative and valuable tool for protecting and managing networks, and is well-suited for a wide range of users.

CHAPTER 11 REFERENCES

[1] Basant Subba on "A Neural Network based NIDS framework for intrusion detection in contemporary network traffic"

National Institute of Technology Hamirpur, Himachal Pradesh, India 177005 Email: basantsubba@nith.ac.in J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] "Improving Attack Detection Performance in NIDS Using GAN" Dongyang Li Kyoto University Kyoto, Japan lidongyang@net.ist.i.kyoto-u.ac.jp Daisuke Kotani Kyoto University Kyoto, Japan kotani@media.kyoto-u.ac.jp Yasuo Okabe Kyoto University Kyoto, Japan okabe@media.kyoto-u.ac.jp.

[3] "NNIDS: Neural Network based Intrusion Detection System" Hassan, Hadi Al-Maksousy.

[4] "MACHINE LEARNING BASED INTRUSION DETECTION SYSTEM" byAnish Halimaa A and Dr. K.Sundarakantham.

[5] "A Novel Network Intrusion Detection System Based on CNN" by Lin Chin