TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PURWANCHAL CAMPUS

A

FINAL YEAR PROJECT REPORT

ON

INTRUSUION DETECTION SYSTEM IN NETWORKS USING RANDOM FOREST

Submitted By:

Abdullah Waqar [26302]
Sandesh Chudal [26334]
Sarthak Parajuli [26338]
Srijan Chaudhary [26342]

PROJECT REPORT SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILMENT OF THE REQUIRMENTS FOR THE DEGREE OF BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

PURWANCHAL CAMPUS

DHARAN

May, 2O22

INTRUSUION DETECTION SYSTEM IN NETWORKS USING RANDOM FOREST

PROJECT REPORT SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILMENT OF THE REQUIRMENTS FOR THE DEGREE OF BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING

By:

Abdullah Waqar [26302]
Sandesh Chudal [26334]
Sarthak Parajuli [26338]
Srijan Chaudhary [26342]

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
PURWANCHAL CAMPUS, INSTITUTE OF ENGINEERING
TRIBHUVAN UNIVERSITY
DHARAN, SUNSARI DISTRICT, NEPAL

May, 2O22

PAGE OF APPROVAL

CERTIFICATE

This is to certify that the project entitled, **"Intrusion Detection System in Networks using Random Forest"** by **Abdullah Waqar, Sandesh Chudal, Sarthak Parajuli** and **Srijan Chaudhary** presented toward the partial fulfilment of the requirement of **Bachelor of Engineering (B.E.) degree in Computer Engineering** has been completed under my supervision. I recommend the same for acceptance by Tribhuvan University, Institute of Engineering.

.....

Er. Tantra Nath Jha

(Project Supervisor)

© COPYRIGHT

ABSTRACT

An intrusion detection system is a system that passively monitors the data exchange in the network with external entities and looks for malicious activities that can be classified as an intrusion or attack. It then notifies the user or sends a notification to some other system which may or may not take action against detected intruder. The objective of our project is to make a system that can monitor the network anonymously, use Random Forest approach to classify the traffic as attack or not and notify the user for detected attacks. The visual structure of our project was made using ReactJS, for the backend we have used Django Framework of python. The Random Forest model was trained using scikit-learn library of python. For the collection of packets, we have used Tshark. For the model training we used the data collected by ourselves. We trained the model with the accuracy of 99.95 per cent. It detected the attacks for most of the time but produced false alerts for some scenario which is common in this type of models. IDS can act as second line of defense. This system can be used in Realtime and improves security of the system. Based on the methodology we adopted and results obtained we can conclude that our system can detect most of the attacks.

Keywords: *Random Forest, Host based intrusion detection, Packet analysis, anomaly detection*

ACKNOWLEDGEMENT

We are extremely thankful to our project supervisor Er. Tantra Nath Jha for his expert guidance and continuous encouragement throughout to see that adequate research has been conducted to complete the project. We would like to thank the Department of Electronics and Computer Engineering for providing us the opportunity to complete our project with all available technical and financial support. Collectively, we would also like to thank our project committee and all helping hands for their time, suggestions, and for graciously agreeing to be on our committee, and always making themselves available.

We would like to offer our heartfelt thanks to the whole department and college personnel for their valuable support throughout our time at this college and on this project. We would like to express deepest appreciation towards Er. Manoj Kumar Guragai, Head of Department of Electronics and Computer Engineering. We would like to use this occasion to express our gratitude to every one of our classmates and colleagues who supported us complete this project successfully. This project would not have advanced to this extent without their assistance, feedback, and encouragement. We would like to thank the authors and publishers of numerous literatures on these topics for their study and hard work, as well as for allowing us to reproduce their material. Lastly, we would like to remember Tribhuvan University and Institute of Engineering – Tribhuvan University for their support throughout our academic years and in this project.

Group Members

Abdullah Waqar [26302]

Sandesh Chudal [26334]

Sarthak Parajuli [26338]

Srijan Chaudhary [26342]

TABLE OF CONTENTS

ABBREVIATIONS

1. ML – Machine Learning
2. IDS – Intrusion Detection System
3. HIDS – Host-based Intrusion Detection System
4. NIDS – Network-based Intrusion Detection System
5. GUI – Graphical User Interface
6. Npm – Node Package Manager
7. Csv – Comma Separated Values
8. HTTP – Hypertext Transfer Protocol
9. IP – Internet Protocol
10. P – Polynomial time solvable algorithm
11. NP – Non-deterministic polynomial time solvable algorithm
12. FE – Front End
13. BE – Back End
14. UML – Unified Modeling Language
15. TCP – Transmission Control Protocol
16. DDoS – Distributed Denial of Service
17. MITM – Man in The Middle
18. WS – Web Socket
19. KNN – K Nearest Neighbors
20. CNN – Convolutional Neural Network
21. RNN – Recurrent Neural Network

LIST OF FIGURES

1. INTRODUCTION

1.1. Background And Basics

An intrusion detection system is a system that passively monitors the data exchange in the network and of the network with externals entities and looks for malicious activities that can be classified as an intrusion or attack. It then notifies the user or sends a notification to some other system which may or may not take action against the detected intruder. Simply put, if the network of devices is a home, an intrusion detection system is a CCTV camera.

Also, a step up from just an intrusion detection system is an anomaly-based intrusion detection system. This type of system creates a profile of normal behavior, and any activity that falls outside of the normal category is marked as anomalous. Anomaly based system is better suited to defend the system against zero-day attacks.

The IDS system should be build keeping in mind all these challenges and they should be overcome in the most efficient manner.

Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges a.s. to a limit as the number of trees in the forest becomes large. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them. Using a random selection of features to split each node yields error rates that compare favorably to Adaboost (Freund and Schapire [1996]), but are more robust with respect to noise. Internal estimates monitor error, strength, and correlation and these are used to show the response to increasing the number of features used in the splitting. Internal estimates are also used to measure variable importance. These ideas are also applicable to regression. [1]

1.2. Problem Statement

Design and implement an Intrusion Detection System using Random Forest which will passively monitor the network traffic and alerts the user for any intrusion detected.

1.3. Purpose of the project

The main function of the IDS system is to detect anomalous behavior that can be caused by the attacks. So, it can be deployed anywhere where we need security measures in place to protect from any intruders.

1.4. Objectives

The objectives of this project are listed below:

1. To monitor network anonymously
2. To use Random Forest Classifier to analyse the monitored packets and identify any intrusion attempts and notify system administrator

2. LITERATURE REVIEW

2.1. Introduction

Literature review is searching similar system and identifying the difference between researcher's projects with existing systems. This helps to get a deep idea of the project. It provides the combination of theoretical, methodological and current knowledge of findings according to subject. There is need to gather the information according to the project. This chapter describes how this app is different from other similar system.

2.2. Existing Researches

With the advancement in technology, cybersecurity is an integral part of our daily online activities. The increased ratio in attacks is also becoming more powerful and difficult to mitigate. Security researchers and hackers are always in a war to win. The mitigation of sophisticated attacks requires high-end hardware or a well-developed software-based system. Existing tools for detection and prevention requires constant maintenance to counter recent attacks. Maintaining mentioned systems also include updating and checking logs on daily basis, which is quite hectic for security experts. Since the internet world is advanced with technology, the attacker's strategies also change which ultimately makes the traditional tools vulnerable. Such liabilities allow intruders to bypass and evade security systems and allowing the intruders to perform malicious activities [2]. Efficiently detecting network intrusions requires the gathering of sensitive information. This means that one has to collect large amounts of network transactions including high details of recent network transactions. Assessments based on meta-heuristic anomaly are important in the intrusion related network transaction data's exploratory analysis. These assessments are needed to make and deliver predictions related to the intrusion possibility based on the available attribute details that are involved in the network transaction. We were able to utilize the NSL-KDD data set, the binary and multiclass problem with a 20% testing dataset. This paper develops a new hybrid model that can be used to estimate the intrusion scope threshold degree based on the network transaction data's optimal features that were made available for training Authors discussed various ML & DL techniques that comes under supervised and unsupervised learning techniques. [3] In the papers [2] [3], authors have proposed

a model in which different data sources such as logs, packets flow, and sessions are collected and presented to ML algorithms. Although, to achieve high accuracy a proper standardized data-set must be available. As per the different reviewed papers for the study, 2017 was the year in which maximum number of publications in journals and conferences were on DL and ANN. This trend is still ongoing with increasing number of publications related to ML or DL techniques. The study shows that the most broadly used methods for NIDS are DL and ANN. IDS keeps track of the state of hardware and software running in the networks for malicious activities that are planned for stealing data. Applying ML models can bring about low false alarm rate and high identification rate. Machine Learning methods can intelligently identify normal and malicious traffic with high accuracy. This research paper highlights different ML approaches utilized to create IDS. Through the broad study and investigation on current literature, the gap for improving and creating efficient IDS can be determined. [4]

2.3. Recent Techniques for Intrusion Detection

The IDS by function can be divided into types; signature-based IDS and the anomaly-based IDS. The signature-based IDS requires the signature patterns available in its signature database to be compared with the packet signature received by the sensor for the intrusion detection. For this type of system, it needs to be up to date with due respect to time, and also its only effective for the known attacks. However, on the other side, the anomaly detection-based Intrusion Detection is effective for the detection of unknown attacks or intrusions.

It relies on the behavior of the system and compares the system's normal behavior with the deviated behavior if attacked by any threat actor. The set of features are used to identify the network connections such as; frame number, frame time, frame length, source mac address, destination mac address, source IP address, destination IP address, IP protocol, IP length, TCP length, TCP source port, TCP destination port, packet info. The features values are recorded by the model and any deviation in recorded values will be marked as anomalous by the anomaly detection engine. Techniques in anomaly detection can be categorized into three types; Machine Learning, Statistical Techniques and Finite State Machine.

3. METHODOLOGY

Intrusion detection systems (IDS) are security systems that are used to detect security threats to computer systems and computer networks. These systems are configured to detect and respond to security threats automatically there by reducing the risk to monitored computers and networks. Intrusion detection systems use different methodologies such as signature based, anomaly based and a hybrid system that combines some or all of the other systems to detect and respond to security threats.

The basic block diagram of intrusion detection system is as shown:

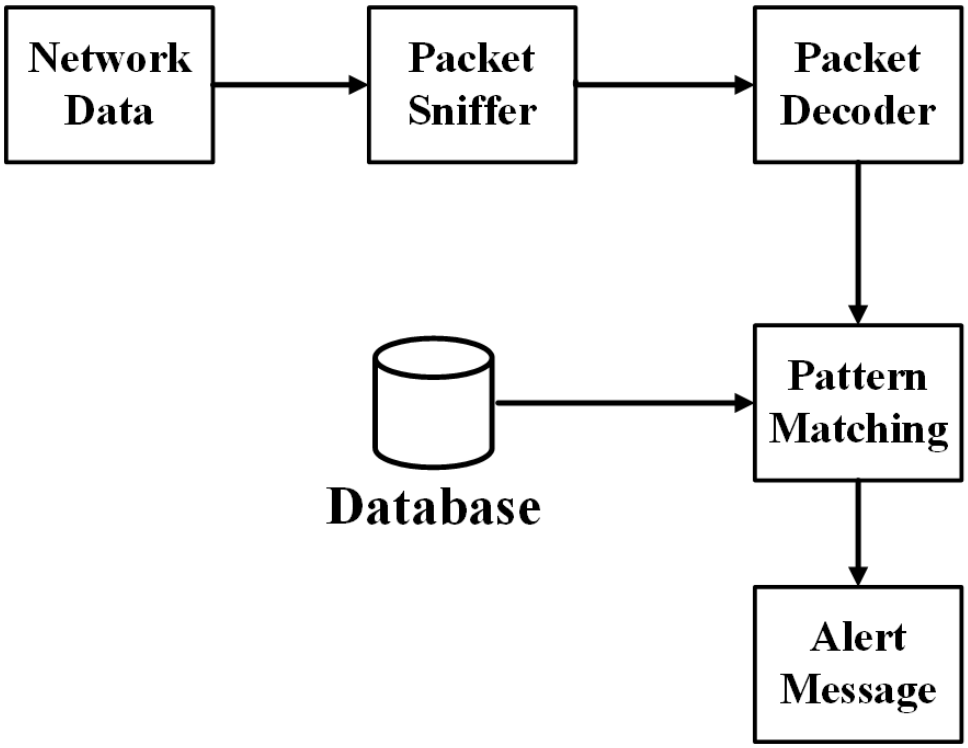


Figure 3.1 Block diagram of Basic Intrusion detection system

For our project development the following aspects are used:

3.1. Frontend

The basic visual structure of our project was made using ReactJS. React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces [5] based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies. React can be used as a base in the development of single-page, mobile, or server-rendered applications with frameworks like Next.js. However, ReactJS is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

3.2. Backend

The backend of our project was done using Django Framework of Python. Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design [6]. Major advantage of using Django is that it is fast, secure, and scalable. PHP web sockets were used to push notifications on dashboard of UI. SQLite [7] .

3.3. Packet Capture

Tshark was used to capture packets. TShark is a network protocol analyzer. It lets you capture packet data from a live network, or read packets from a previously saved capture file, either printing a decoded form of those packets to the standard output or writing the packets to a file. TShark's native capture file format is pcapng format, which is also the format used by Wireshark and various other tools [8].

3.4. Machine Learning

Random Forest Classifier has been used for the building and implementing the model. Basically, a random forest is an average of tree estimators. As with nonparametric regression, simple and interpretable classifiers can be derived by partitioning the range of X.

Let n = A_1, . . ., A_N be a partition of X. Let A_j be the partition element that contains *x*. Then h(x) = 1 if X_i A_j Y_i X_i A_j Y_i and h(x) = 0 otherwise.

We conclude that the corresponding classification risk satisfies R(h)

$$R(h) = O\left(\frac{n1}{d+2}\right) \quad \text{..... (i)}$$

These are bagged trees except that we also choose random subsets of features for each tree. The estimator can be written as

$$\widehat{m}(x) = \frac{1}{M}\sum_j \widehat{\mathrm{m}}j\ (x)\ \ldots\ldots\ldots \text{(ii)}$$

where m_j is a tree estimator based on a subsample (or bootstrap) of size a using p randomly selected feature. The trees are usually required to have some number k of observations in the leaves. There are three tuning parameters: a, p and k. You could also think of M as a tuning parameter but generally we can think of M as tending to. For each tree, we can estimate the prediction error on the un-used data. (The tree is built on a subsample.)

Averaging these prediction errors gives an estimate called the out-of-bag error estimate.

IF Y=F(X) is the function then,

X: frame number, frame time, frame length, source mac address, destination mac address, source IP address, destination IP address, IP protocol, IP length, TCP length,

TCP source port, TCP destination port, packet info as an

input Y: Classification – attack/normal.

1. Notification: This function takes the input which is provided by the (intrusion detection) and notifies the user.

   IF Y=F(X) is the function then,

   X: Output of the function (intrusion detection)

   Y: Intrusion message to the user

2. Output = {display intrusion msg}
   a. display intrusion msg: display error message if any intrusion occurs.
3. Intermediate Results
   a. Successful working of module.
   b. Successful Working of Network.
   c. Successful User authentication.
4. Terminate = {Invalid details, Network failure, Timeout}
   a. Invalid User Authentication.

b. Network failure

c. timeout

5. Success

a. Successful user login.

b. Successful connection establishment of nodes and ids.

c. Successful detection of intrusion.

d. Displaying the results.

e. Appropriate error messages in case of invalid input.

6. Failure

a. Web app Failure.

b. Hardware faults.

c. Network establishment failure.

d. Not displaying required results.

4. SYSTEM ANALYSIS AND FEASIBILITY STUDY

4.1. Feasibility Study

4.1.1 Economic Feasibility

This project is cost-effective since it just requires simple algorithms to construct and is also cost-effective to utilize because it may be used for free. Our project is under budget because it simply requires a personal computer and a Network. However, it requires a computer with high computing power

4.1.2 Technical Feasibility

Our project would be technically feasible as we are proposing to use python. Our project will would work explicitly on a single device attached on a network.

4.1.3 Social Feasibility

Due to the growing demand of Cyber security, our project with some more enhancement can sustain itself in cybersecurity field. An application must be socially acceptable and should not cause any harm to its users. For a product to grow successfully, it must gain approval from society or users. Our project is socially feasible because it does not hurt or harm anyone.

4.1.4 Time Feasibility

Training Complexity: $O((n^2)$ pntrees)

Prediction Complexity: O(pntrees)

Calling n the number of training sample, p the number of features, n trees the number of trees (for methods based on various trees),
where, n= 21000 (Approx.) p = 17 ntrees= 4
Hence O(pntrees) runs in polynomial time complexity. The algorithm falls in P. Hence, the given problem is NP.

4.2. Requirement Analysis

4.2.1 Functional Requirements

These are the requirements that the end user specifically demands as basic facilities that the system should offer.

4.2.1.1 Connectivity

The devices are to be connected to the network through Wi-Fi. This feature is of high priority because the sensed data needs to be processed further. The data will be processed only if it gets proper way to reach to the server. Only risk of this feature is that it can sometimes not send the data to the server.

Stimulus/Response Sequences

Data sensed is sent to the server.

Functional Requirements

Apache server of the network along with Wi-Fi module.

4.2.1.2 Data Processing

Description and Priority

Once the data gets to the server, software performs processing on it. Here IDS software is used to determine the category of the attack. This feature is of high priority because this is the main objective behind selecting this project. Only risk of this feature is that it can sometimes not categorize the attack perfectly.

Stimulus/Response Sequences

In response, it will determine whether the attack was normal or malicious.

Functional Requirements

IDS software.

4.2.2 Non-Functional Requirements

4.2.2.1 Performance Requirements

The processing of the data should be fast enough so as to give proper result as soon as the attack happens, i.e., the time between the occurrence and detection of the attack should be minimum.

4.2.2.2 Safety Requirements

The backend could crash resulting the failure of whole system. If the system takes up all the processing power, the machine could crash. Regular maintenance of the networking should be done.

4.2.2.3 Security Requirements

User need to signup first and then login to get access to the system, in this case users data should be protected. Systems should be secure against unauthorized access to any of their data, unauthorized use of them or any of their components. Details regarding their data should also be protected.

4.2.3 Deployment Environment

4.2.3.1 Software Requirements

1) Python 3 - Coding language used in our project
2) Django Framework - Used for backend work
3) Node - Used for executing JavaScript code outside of a web browser
4) Npm - Used as package manager for the Node JavaScript platform
5) React.js - Used for frontend work
6) SQLite - Used for database
7) Apache server - Used for storing network data
8) Tshark - Used for network data analysis

4.2.3.2 Hardware Requirements

- Laptop/Pc
- Router/Switch (Active Network)
- Wi-Fi Modules / Ethernet Modules

4.2.3.3 Operating System

We have chosen Kali Linux OS since it has pre-installed drivers and tools required for data capture and any other tools required can be easily installed on it. It provides security and performance.

5. SYSTEM DESIGN AND ARCHITECTURE

5.1. System block diagram

The block diagram of our system is shown below. The information about packets is captured by Tshark and it is stored in CSV file. The python script compares the captured file with the pickle file of model stored on backend and in case of detected attacks the notification is sent on the frontend dashboard using web sockets. The pho script stores the collected packets statistics and displays them in the frontend live monitoring panel.

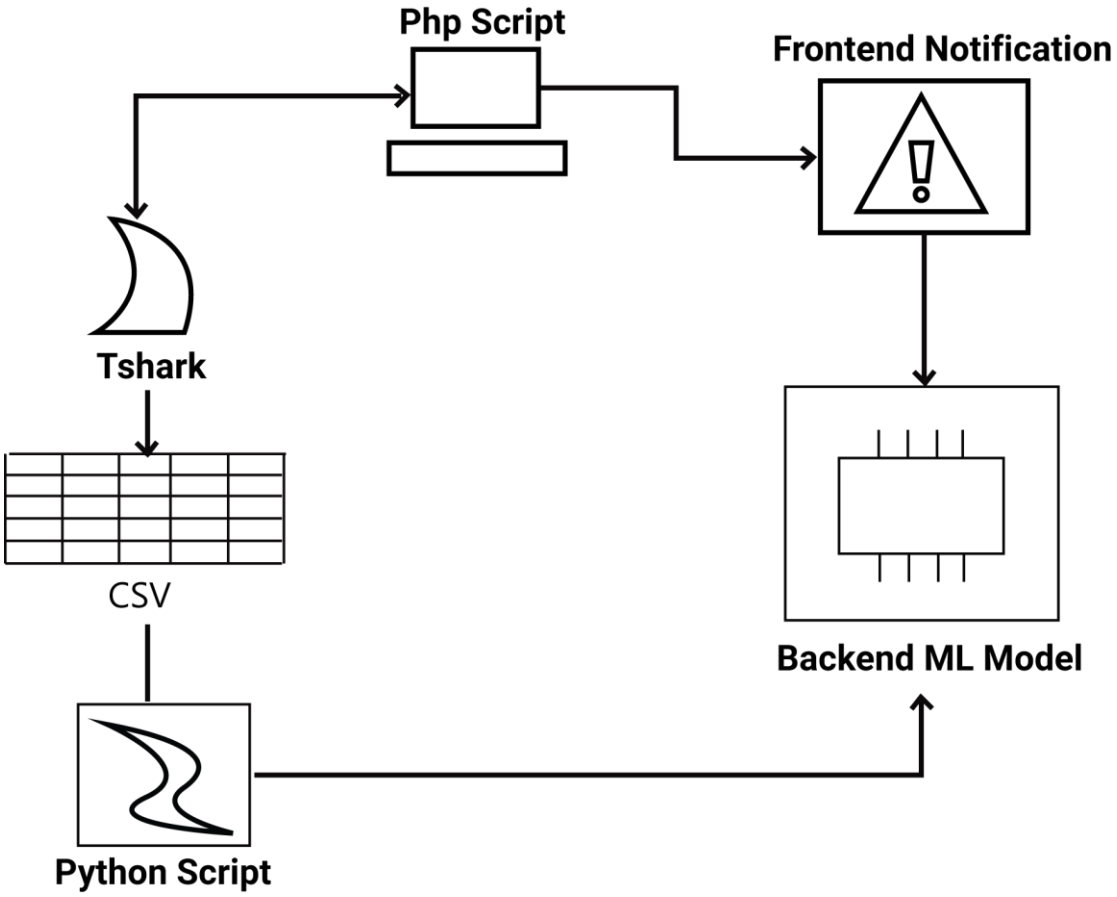


Figure 5.1 System block diagram

5.2. Use Case Diagram

The diagram below shows the use case diagram of our project

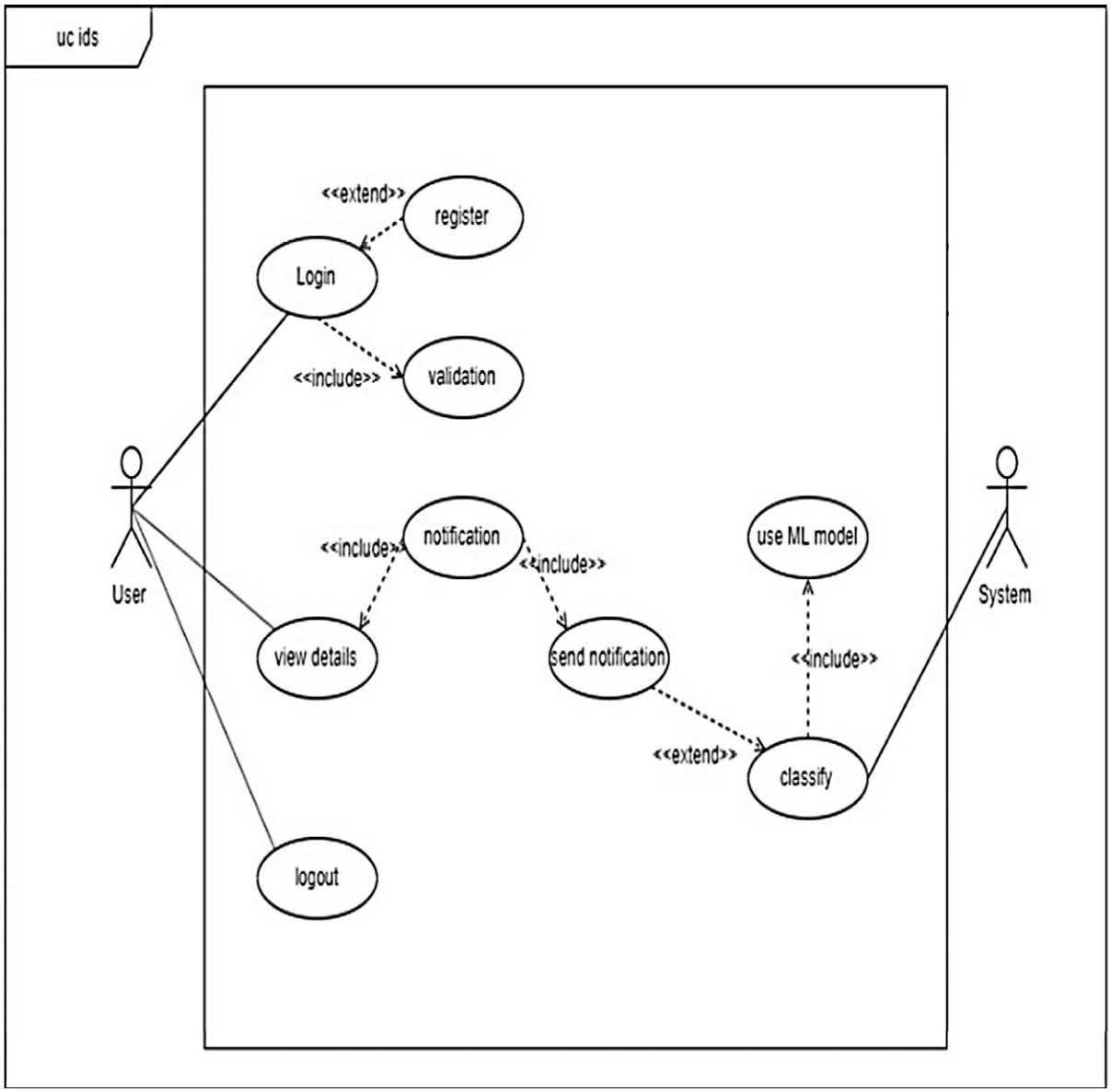


Figure 5.2 Use case diagram

5.3. Activity Diagram

The activity diagram shows the stepwise flow of activities and actions. Here it explains the flow of the system, how it starts with login/registration. Then its user can perform their regular tasks while the IDS keeps working in the background and notifies the user when an attack is detected.

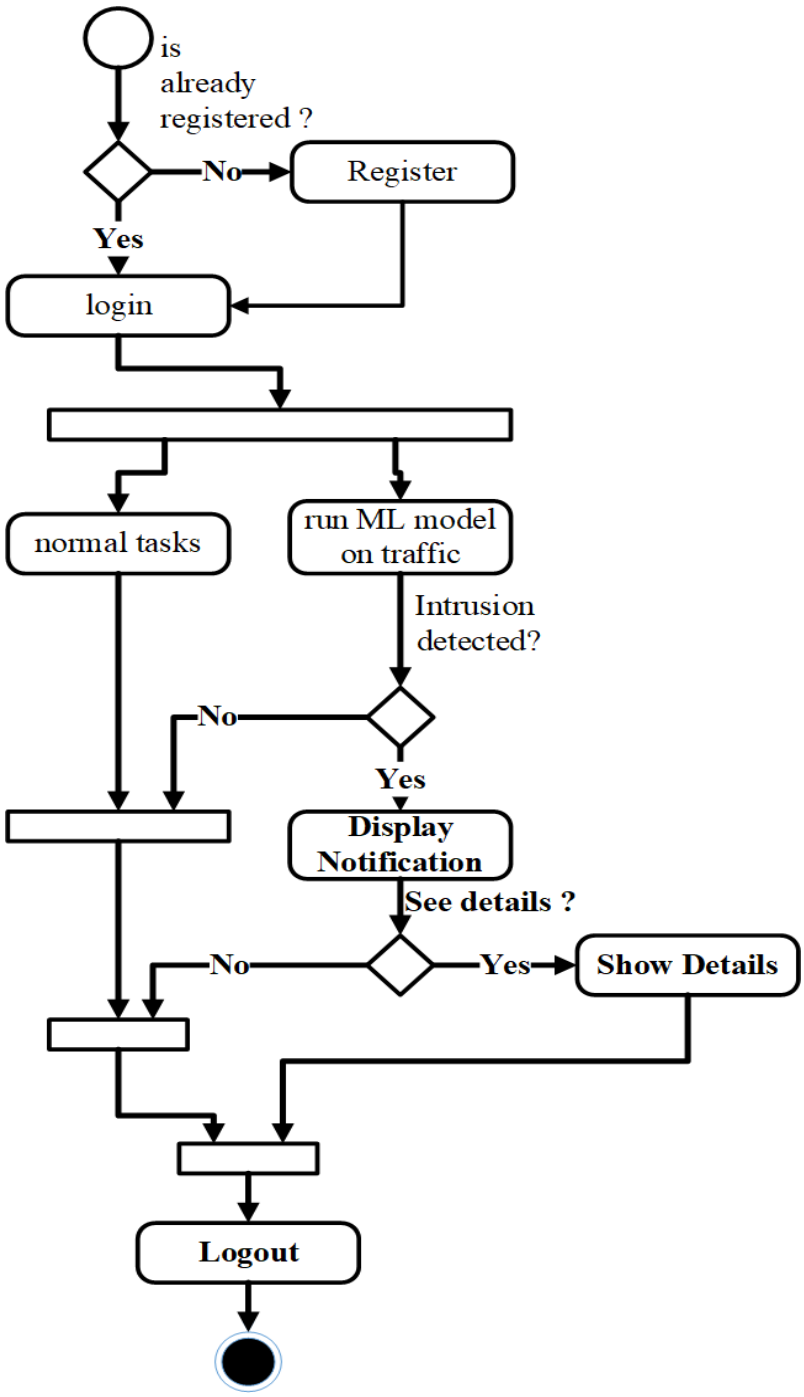


Figure 5.3 Activity Diagram

5.4. Class Diagram

A class diagram is a static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. Here it shows the structure of objects such as user, notification, model, etc.

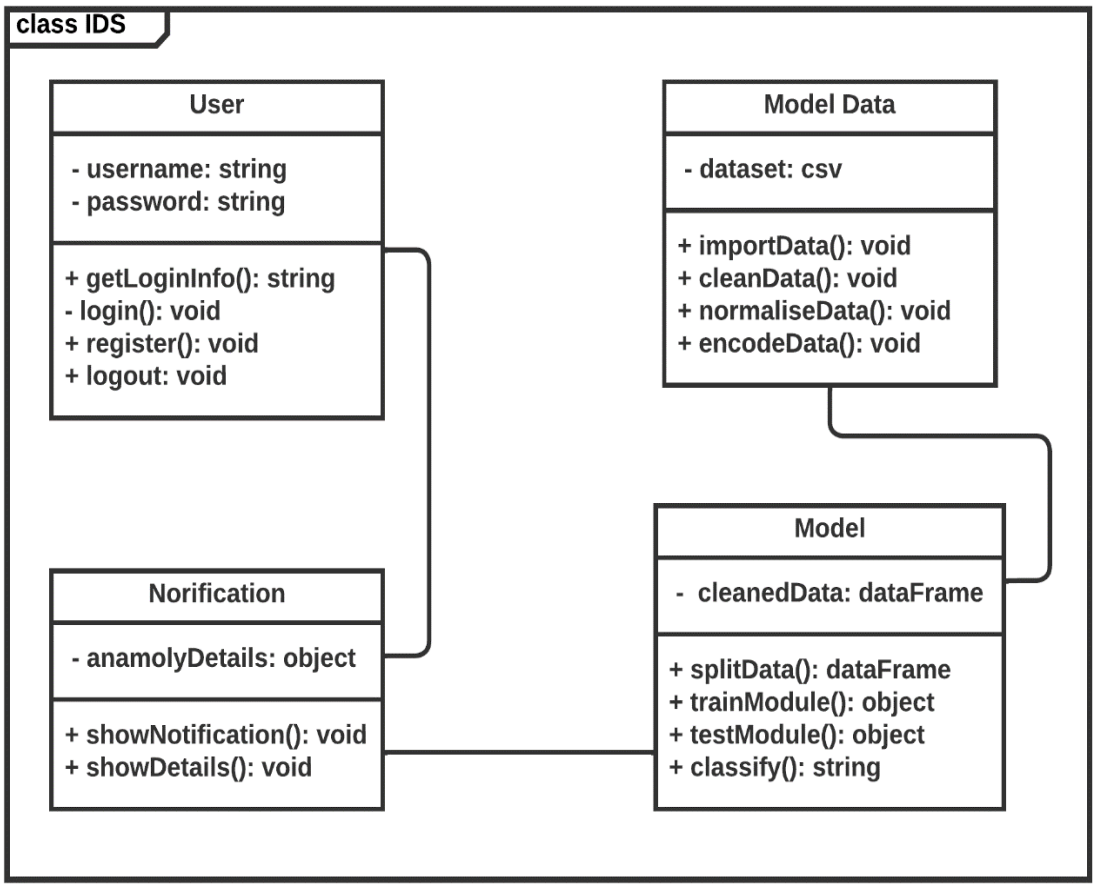


Figure 5.4 Class Diagram

5.5. State Machine Diagram

UML state machine, also known as UML state chart, is a significantly enhanced realization of the mathematical concept of a finite automaton in computer science applications as expressed in the Unified Modeling Language (UML) notation.

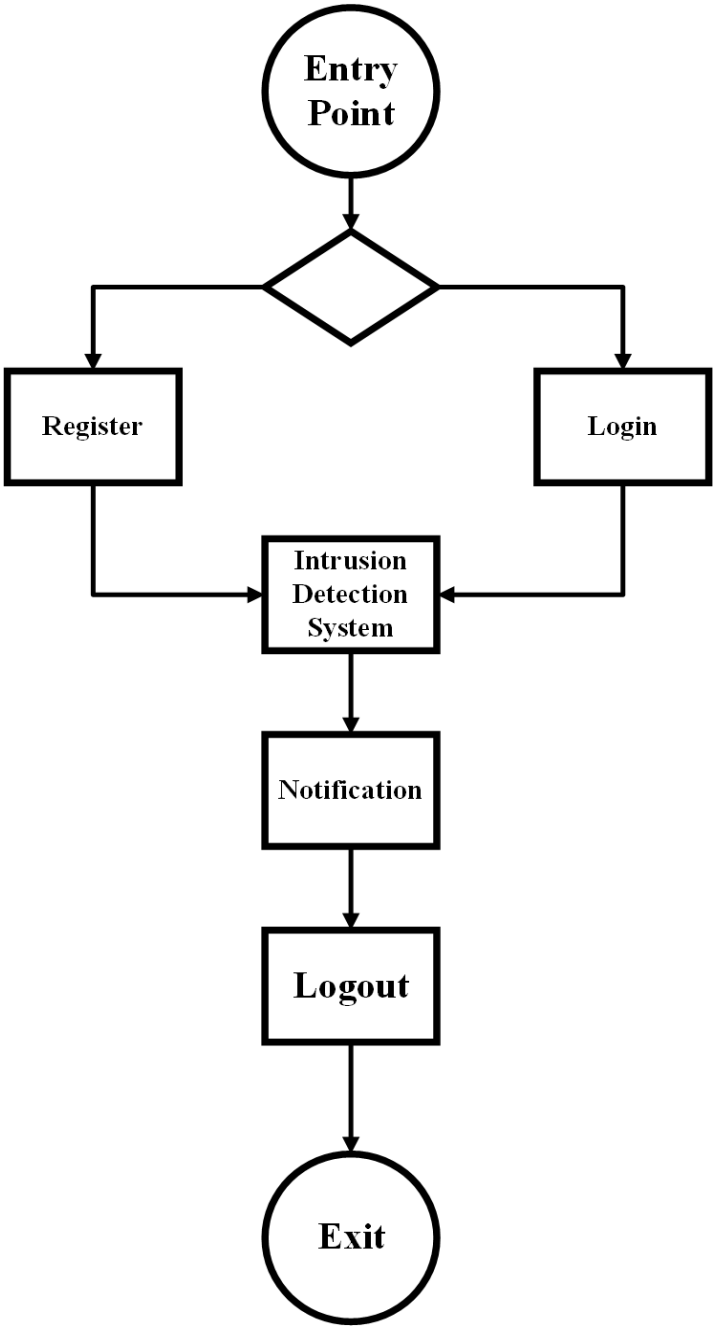


Figure 5.5 **State machine diagram**

5.6. Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. The following diagram shows the sequence of steps that follows one after the other in the system. It shows when the connection is established and when the model starts giving the intrusion alerts.

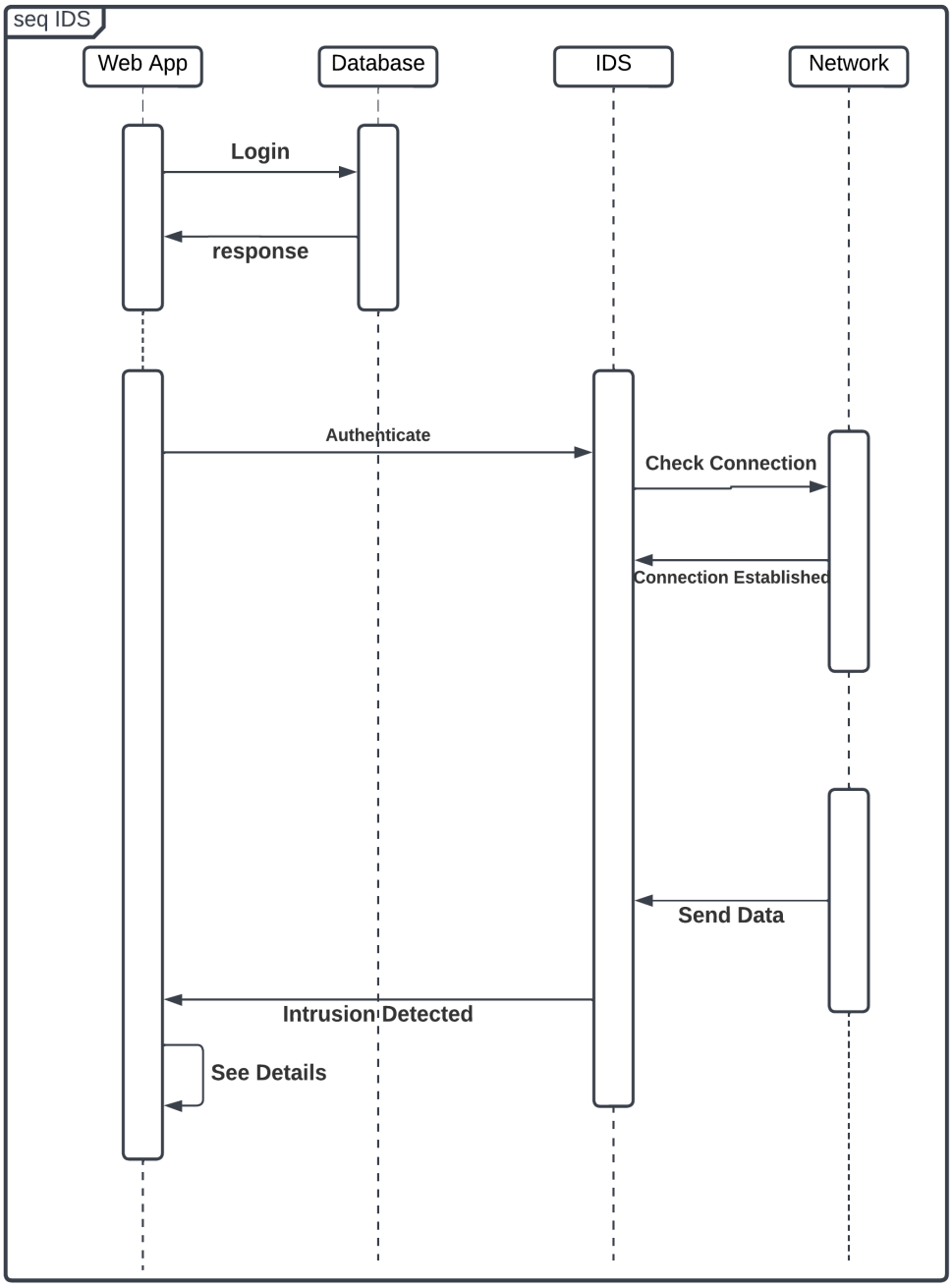


Figure 5.6 Sequence Diagram

6. RESULT AND ANALYSIS

6.1. Introduction

This chapter covers the role of various subsystems/modules/classes along with implementation details listing of the code for the major functionalities. The network traffic is monitored by the TShark script and dumped in the file.csv. The Django Python backend reads this data in real time, feeds it to the ML model based on Random Forest and classifies it either as normal or attack. This data is sent to the front end and displayed in the form of a notification.

6.2. Discussion

The dataset created had a lower number of records labeled as attack as compared to the records labeled as normal as can be seen in fig 1. This issue was resolved by the oversampling done, the results of which can be seen in the fig 2. The Random Forest ML model has been proven to be more accurate than other models in comparison – Naïve Bayes, KNN and Decision Tree with high accuracy (~99 per cent). Although the accuracy seems very high, it has in fact been overfitted on the dataset. This means that even if it shows very high accuracy on training set, it performs poorly on new data. This has given rise to a higher rate of false alarms in the real time working of the IDS. Although this can be an issue, the advantage of this is that even if it has false alarms, it does catch all of the intrusions that occur, thereby giving high true positive rates.

6.3. Screenshots

6.3.1 Attack Screenshots

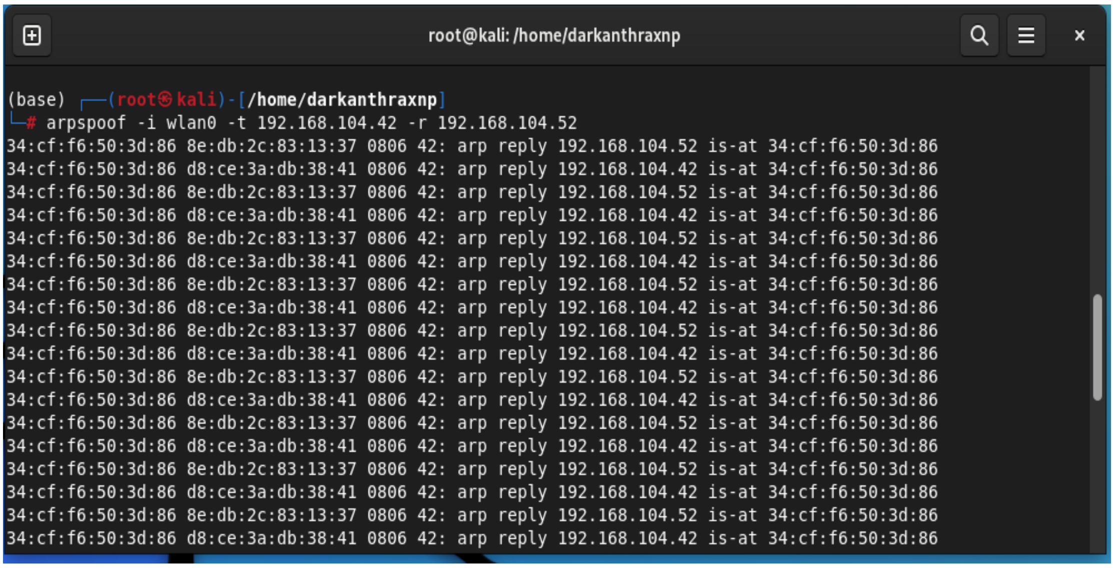


Figure 6.1 MITM Attack (Arpspoof)

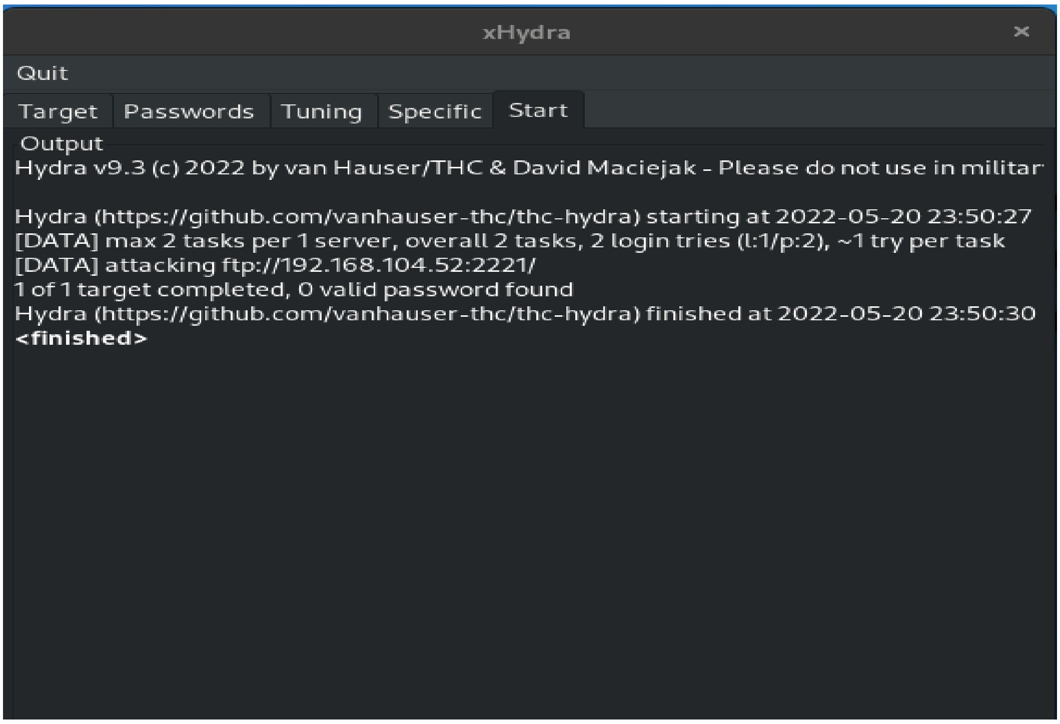
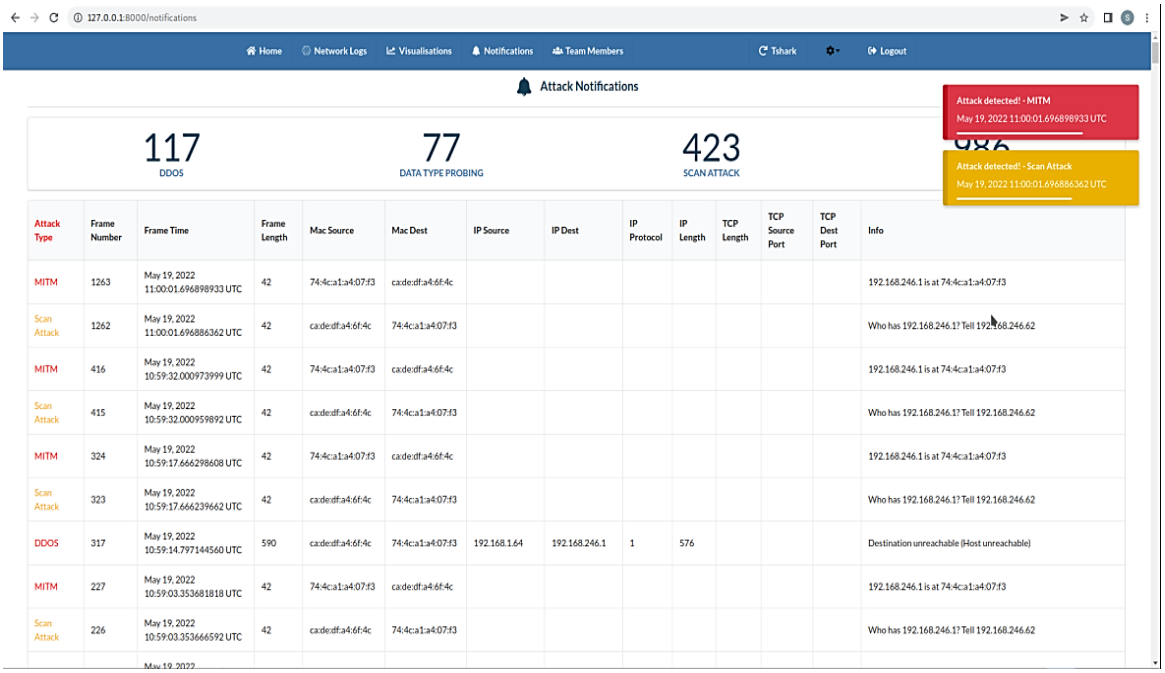
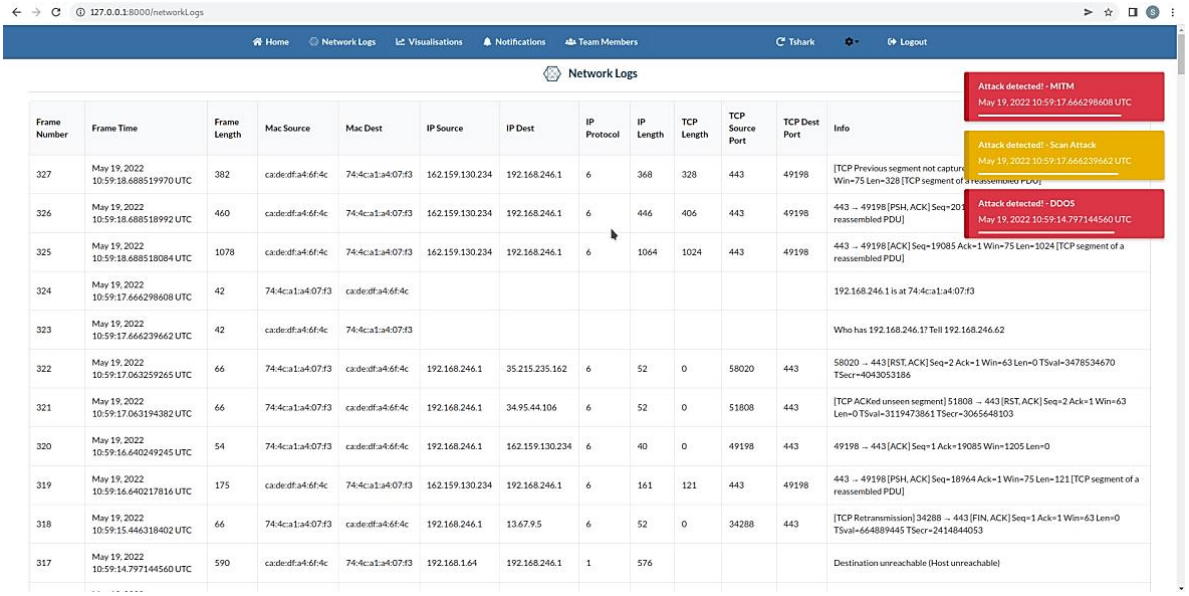


Figure 6.2 DDoS attack (xHydra)



Attack Type	Frame Number	Frame Time	Frame Length	Mac Source	Mac Dest	IP Source	IP Dest	IP Protocol	IP Length	TCP Length	TCP Source Port	TCP Dest Port	Info
MITM	1263	May 19, 2022 11:00:01.696898933 UTC	42	74:4c:a1:a4:07:f3	ca:de:df:a4:6f:4c								192.168.246.1 is at 74:4c:a1:a4:07:f3
Scan Attack	1262	May 19, 2022 11:00:01.696886362 UTC	42	ca:de:df:a4:6f:4c	74:4c:a1:a4:07:f3								Who has 192.168.246.1? Tell 192.168.246.62
MITM	416	May 19, 2022 10:59:32.000973999 UTC	42	74:4c:a1:a4:07:f3	ca:de:df:a4:6f:4c								192.168.246.1 is at 74:4c:a1:a4:07:f3
Scan Attack	415	May 19, 2022 10:59:32.000959892 UTC	42	ca:de:df:a4:6f:4c	74:4c:a1:a4:07:f3								Who has 192.168.246.1? Tell 192.168.246.62
MITM	324	May 19, 2022 10:59:17.666298608 UTC	42	74:4c:a1:a4:07:f3	ca:de:df:a4:6f:4c								192.168.246.1 is at 74:4c:a1:a4:07:f3
Scan Attack	323	May 19, 2022 10:59:17.666239662 UTC	42	ca:de:df:a4:6f:4c	74:4c:a1:a4:07:f3								Who has 192.168.246.1? Tell 192.168.246.62
DDOS	317	May 19, 2022 10:59:14.797144560 UTC	590	ca:de:df:a4:6f:4c	74:4c:a1:a4:07:f3	192.168.1.64	192.168.246.1	1	576				Destination unreachable (Host unreachable)
MITM	227	May 19, 2022 10:59:03.353681818 UTC	42	74:4c:a1:a4:07:f3	ca:de:df:a4:6f:4c								192.168.246.1 is at 74:4c:a1:a4:07:f3
Scan Attack	226	May 19, 2022 10:59:03.353666592 UTC	42	ca:de:df:a4:6f:4c	74:4c:a1:a4:07:f3								Who has 192.168.246.1? Tell 192.168.246.62

Figure 6.3 Notifications Screenshot



Frame Number	Frame Time	Frame Length	Mac Source	Mac Dest	IP Source	IP Dest	IP Protocol	IP Length	TCP Length	TCP Source Port	TCP Dest Port	Info
327	May 19, 2022 10:59:18.688519970 UTC	382	ca:de:df:a4:6f:4c	74:4c:a1:a4:07:f3	162.159.130.234	192.168.246.1	6	368	328	443	49198	[TCP Previous segment not captur Win=75 Len=328 [TCP segment of a reassembled PDU]
326	May 19, 2022 10:59:18.688518992 UTC	460	ca:de:df:a4:6f:4c	74:4c:a1:a4:07:f3	162.159.130.234	192.168.246.1	6	446	406	443	49198	443 → 49198 [PSH, ACK] Seq=20 reassembled PDU]
325	May 19, 2022 10:59:18.688518084 UTC	1078	ca:de:df:a4:6f:4c	74:4c:a1:a4:07:f3	162.159.130.234	192.168.246.1	6	1064	1024	443	49198	443 → 49198 [ACK] Seq=19085 Ack=1 Win=75 Len=1024 [TCP segment of a reassembled PDU]
324	May 19, 2022 10:59:17.666298608 UTC	42	74:4c:a1:a4:07:f3	ca:de:df:a4:6f:4c								192.168.246.1 is at 74:4c:a1:a4:07:f3
323	May 19, 2022 10:59:17.666239662 UTC	42	ca:de:df:a4:6f:4c	74:4c:a1:a4:07:f3								Who has 192.168.246.1? Tell 192.168.246.62
322	May 19, 2022 10:59:17.063259265 UTC	66	74:4c:a1:a4:07:f3	ca:de:df:a4:6f:4c	192.168.246.1	35.215.235.162	6	52	0	58020	443	58020 → 443 [RST, ACK] Seq=2 Ack=1 Win=63 Len=0 TSval=3478534670 TSecr=4043053186
321	May 19, 2022 10:59:17.063194382 UTC	66	74:4c:a1:a4:07:f3	ca:de:df:a4:6f:4c	192.168.246.1	34.95.44.106	6	52	0	51808	443	[TCP ACKed unseen segment] 51808 → 443 [RST, ACK] Seq=2 Ack=1 Win=63 Len=0 TSval=3119473861 TSecr=3065648103
320	May 19, 2022 10:59:16.640249245 UTC	54	74:4c:a1:a4:07:f3	ca:de:df:a4:6f:4c	192.168.246.1	162.159.130.234	6	40	0	49198	443	49198 → 443 [ACK] Seq=1 Ack=19085 Win=1205 Len=0
319	May 19, 2022 10:59:16.640217816 UTC	175	ca:de:df:a4:6f:4c	74:4c:a1:a4:07:f3	162.159.130.234	192.168.246.1	6	161	121	443	49198	443 → 49198 [PSH, ACK] Seq=18964 Ack=1 Win=75 Len=121 [TCP segment of a reassembled PDU]
318	May 19, 2022 10:59:15.446318402 UTC	66	74:4c:a1:a4:07:f3	ca:de:df:a4:6f:4c	192.168.246.1	13.67.9.5	6	52	0	34288	443	[TCP Retransmission] 34288 → 443 [FIN, ACK] Seq=1 Ack=1 Win=63 Len=0 TSval=664889445 TSecr=2414844053
317	May 19, 2022 10:59:14.797144560 UTC	590	ca:de:df:a4:6f:4c	74:4c:a1:a4:07:f3	192.168.1.64	192.168.246.1	1	576				Destination unreachable (Host unreachable)

Figure 6.4 Network Log Screenshot with notification

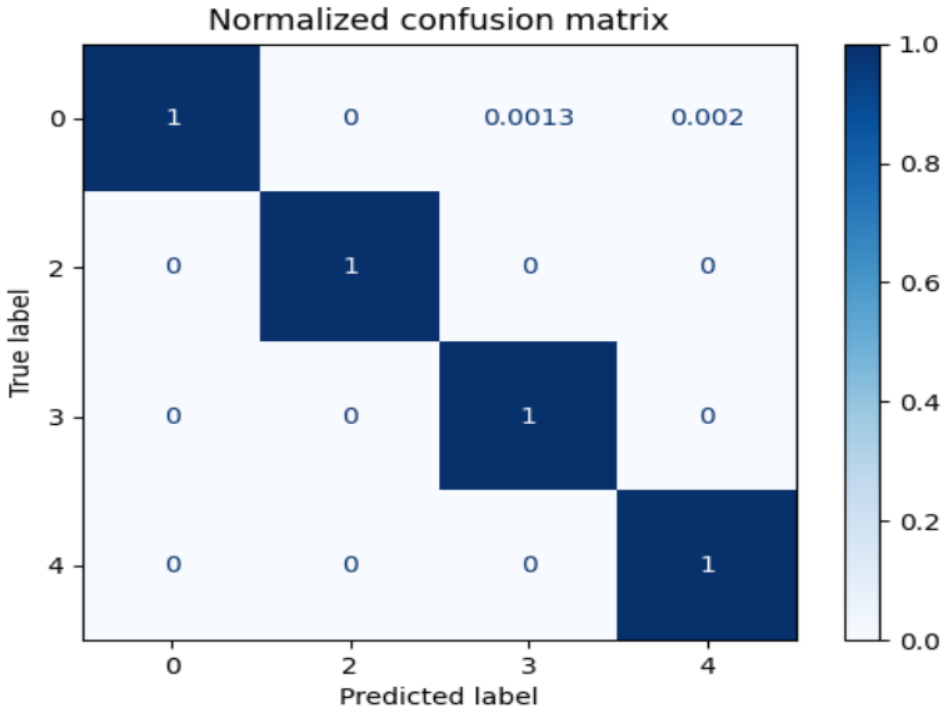


Figure 6.5 Confusion Matrix

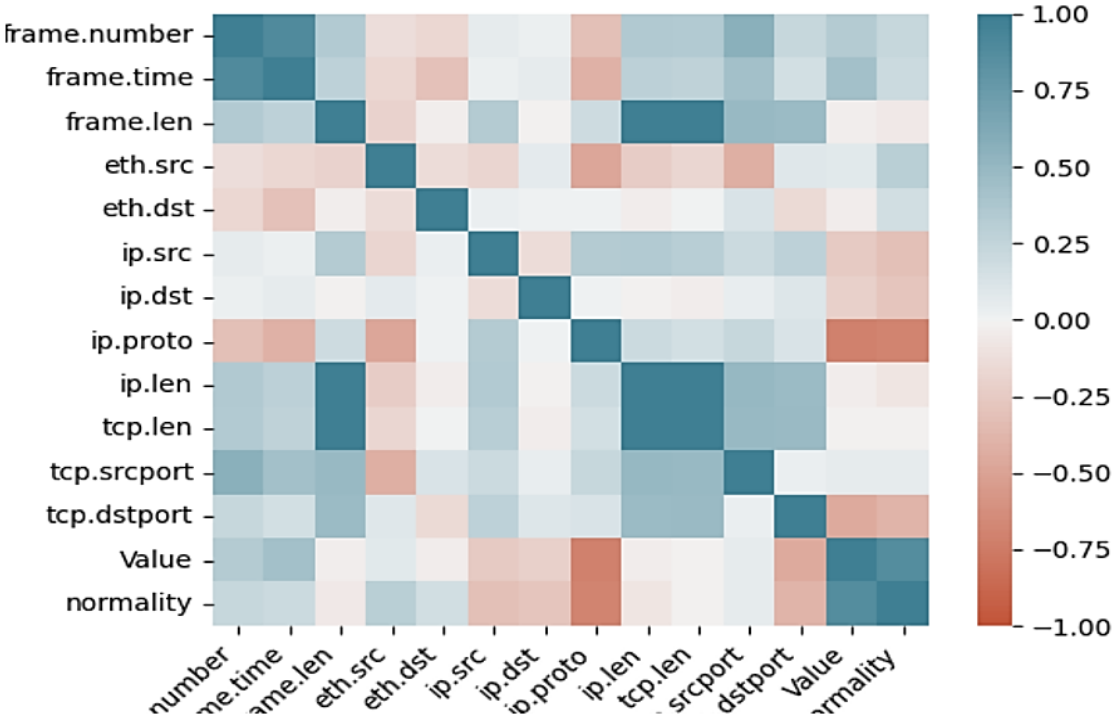


Figure 6.6 Correlation Matrix

7. CONCLUSION

Intrusion detection systems act as a second line of defense after the firewall and are beneficial for the security of the networks. This system has been developed so that can be used in real time networks to improve their security. Using this system, the user can fetch their network data, monitor network traffic, get notified when an intrusion is detected in the network. This has been achieved using the ML model Random Forest. The reduced latency of the system which makes it real time is achieved by web sockets. Since this system is generalized, it can be used for different types of networks to detect intrusions in their network. The model shows a high level of accuracy 99.95 per cent, it correctly classifies most of the attacks in real time, however it suffers from a high rate of false alarms, which is common with these types of models. More work can be done in the future to reduce the false alarm rates and make the model more robust.

8. RECOMMENDATIONS

The main goal of an intrusion detection system (IDS) is to detect abnormal traffic in real time. It operates in near real-time due to the high computational requirements. Using distributed computing, it can reduce reaction time. It should nearly always be correct. Because the dataset for model training was gathered on the same machine, it is less accurate. It is better to use standard datasets like CSE-CIC-IDS2018 [9]. The system simply detects intrusions, not prevents them, which may be addressed by expanding the system to include preventative strategies in the event of an attack being detected. The current system is limited to a single host machine, but it may be expanded to become a distributed application in which different modules of the system operating on different systems communicate. For improved dependability, the false alarm rate must be minimized.

REFERENCES

[1] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, 2001.

[2] A. Chawla, P. Jacob and S. Fallon , Host based Intrusion Detection System with Combined CNN/RNN Model, 2018 .

[3] S. Aljawarneh, M. Aldwairi and M. B. Yasin, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of COMPUTATIONAL SCIENCE,* p. 22, 2017.

[4] P. R. Maidamwar, M. M. Bartere and P. P. Lokulwar, "A Survey on Machine Learning Approaches for Developing Intrusion Detection System," in *Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2021*, 2021.

[5] "ReactJS," [Online]. Available: https://reactjs.org/. [Accessed 20 5 2022].

[6] "Django," Django Software Foundation, [Online]. Available: https://djangoproject.com. [Accessed 20 May 2022].

[7] "SQLite," SQLite Consortium, [Online]. Available: https://www.sqlite.org/index.html. [Accessed 20 May 2022].

[8] "tshark(1) Manual Page," Sysdig, Inc., [Online]. Available: https://www.wireshark.org/docs/man-pages/tshark.html. [Accessed 20 May 2022].

[9] University of New Brunswick, "A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018)," University of New Brunswick, 2018. [Online]. Available: https://registry.opendata.aws/cse-cic-ids2018. [Accessed 22 May 2022].