A

Mini Project

On

MITIGATING DDOS ATTACK IN IOT NETWORK ENVIRONMENT

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

P.KARTHIKEYA (207R1A05N4)

Under the guidance of

DR.K. SRUJAN RAJU

(Professor of CSE)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

2020-2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR
GROUP OF INSTITUTIONS
EXPLORE TO INVENT

CERTIFICATE

This is to certify that the project entitled **“MITIGATING DDOS ATTACK IN IOT NETWORK ENVIRONMENT”** being submitted by **P.KARTHIKEYA (207R1A05N4)** in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad , is a record of bonafide work carried out by them under our guidance and supervision during the year 2023-2024.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dr. K. Srujan Raju
(Professor of CSE)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on___________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **Dr . K.Srujan Raju,**Professor of CSE for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **G.Vinesh Shanker, Dr. J. Narasimharao, Ms. Shilpa, & Dr. K. Maheswari** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

P.KARTHIKEYA (207R1A05N4)

ABSTRACT

The recent proliferation of Internet of Things (IoT) is paving the way for the emergence of smart cities, where billions of IoT devices are interconnected to provide novel pervasive services and automate our daily life tasks (e.g., smart healthcare, smart home). However, as the number of insecure IoT devices continues to grow at a rapid rate, the impact of Distributed Denial-of-Service (DDoS) attacks is growing rapidly. With the advent of IoT botnets such as Mirai, the view towards IoT has changed from enabler of smart cities into a powerful amplifying tool for cyberattacks. This motivates development of new techniques to provide flexibility and efficiency of decision making on the attack collaboration in a software defined networks (SDN) context.

The new emerging technologies, such as SDN and blockchain, give rise to new opportunities for secure, low-cost, flexible and efficient DDoS attacks collaboration for the IoT environment. In this paper, we propose Co-IoT, a blockchain-based framework for collaborative DDoS attack mitigation; it uses smart contracts (i.e., Ethereum's smart contracts) in order to facilitate the attack collaboration among SDN-based domains and transfer attack information's in a secure, efficient and decentralized manner. Co-IoT's implementation is deployed on the Ethereum official test network Ropsten. The experimental results confirm that Co-IoT achieves flexibility, efficiency, security, cost effectiveness making it a promising scheme to mitigate DDoS attacks in large scale.

IoT networks are uniquely susceptible to DDoS attacks due to their often large-scale, heterogeneous, and geographically dispersed nature. This vulnerability highlights the importance of proactive measures to protect IoT ecosystems from potential threats. In the following paragraphs, we will explore various approaches and techniques for mitigating DDoS attacks in IoT networks, which encompass both preventative and responsive measures to ensure the resilience and security of these interconnected systems.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

TABLE OF CONTENTS

1. INTRODUCTION

1.INTRODUCTION

1.1 PROJECT SCOPE

This project is titled “Mitigating DDoS Attack in IoT Network Environment”. We can say with the increasing of accuracy and effectiveness of the project ,the machine learning algorithms can easily predict many types of DDoS attacks. As a result ,we can decrease false-positive rate through this model. By considering many machine learning algorithms, we can find the attack in one or other way. It gives Robust features such that it can predict different types of attacks.

1.2 PROJECT PURPOSE

The project could lead to the integration of machine learning-based DDoS mitigation techniques into existing security products and services. This integration can enhance the overall cybersecurity. Enhance project's ability to detect and respond to DDoS attacks in real-time. This can involve optimizing algorithms for implementing streaming data processing, and integrating with network monitoring tools.

1.3 PROJECT FEATURES

Exploring anomaly detection techniques to identify unusual traffic patterns that might indicate DDoS attacks. Experiment with data augmentation techniques to increase the diversity and size of your training dataset. This can lead to better generalization and robustness of machine learning models. Conduct extensive testing of the system in a controlled environment to assess its scalability and performance under heavy DDoS attack scenario.

2. SYSTEM ANALYSIS

2.SYSTEM ANALYSIS

2.1 INTRODUCTION

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.2 PROBLEM DEFINITION

A general statement of DDOS attack problem can be formulated in the given IOT devices, identify or verify one or more DDOS attack in the scene or in the network devices using a stored database.

2.3 EXISTING SYSTEM

Blockchain technology can help mitigate DDoS attacks in IoT by providing a decentralized and transparent system for managing network traffic. For example, Blockchain can enable real-time monitoring and analysis of network traffic, which can help identify and respond to potential threats more quickly. Blockchain can also provide a secure and tamper-proof record of network activity, which can be used to investigate and prosecute cybercriminals. One existing Blockchain system for mitigating DDoS attacks in IoT is the Fog Computing-based Blockchain (FCB) system. FCB utilizes a fog computing architecture to distribute the computational load across multiple devices, reducing the burden on individual IoT devices. This helps to mitigate the risk of DDoS attacks by ensuring that no single device is overwhelmed by incoming traffic.

2.3.1 DISADVANTAGES OF EXISTING SYSTEM

- Large scale attacks cannot be handle.
- DDoS mitigation systems can be expensive to implement and maintain.
- DDoS mitigation systems will be complex to implement and requires technical knowledge.
- It can detect and identify some of the DDoS attack but cannot identify critical ones.
- Sometimes, it will fail to detect the attack.

2.4 PROPOSED SYSTEM

For the proposed system of the project , using algorithms such as SVM, Random Forest, XGBoost , AdaBoost , KNN and Naive Bayes to detect DDos.Here ,to train the algorithm we are using 10 different attacks of IOT environment as CIC Dataset.To predict the Accuracy of DDos attack ,test data should be given.Created some different classes to run the algorithms.

2.4.1 ADVANTAGES OF THE PROPOSED SYSTEM

- Machine learning algorithms work by Analyzing network traffic patterns to identify abnormal behavior that may indicate a DDoS attack.
- They are more adaptable to changing attack patterns and can detect attacks that may not fit a pre-defined set of rules.
- They can learn from past attacks to improve their accuracy over time and reduce false positives.
- High Efficiency.
- High Accuracy.

2.5 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are:

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.5.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited**.** The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.5.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.5.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

2.6 HARDWARE & SOFTWARE REQUIREMENTS

2.6.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements:

- System : Intel I3 or Above.
- Hard Disk : 40 GB.
- Monitor : 15 inch VGA Color.
- Mouse : Logitech Mouse.
- Ram : 4GB or Above.
- Keyboard : Standard Keyboard.
- CPU :1GHZ.

2.6.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements:

- Operating System : Windows 7 or Above.
- Platform : Python Technology
- Tool : Spyder, Python 3.5
- Front End : Anaconda
- Back End : Python Anaconda Script

3. ARCHITECTURE

3.ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction.

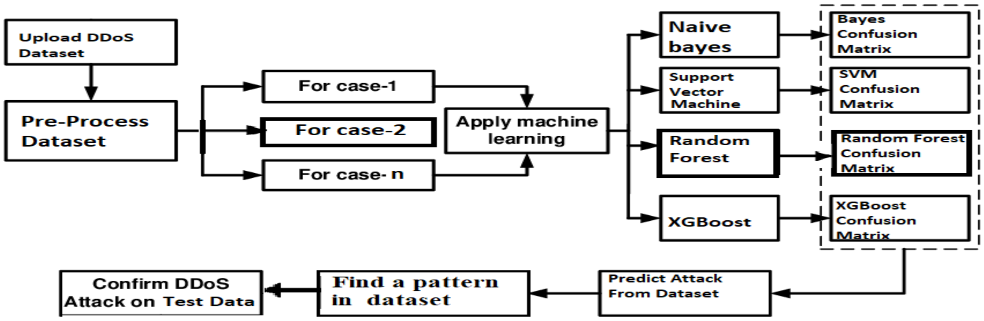


Figure 3.1: Project Architecture for mitigation of DDoS attack in IoT network environment

3.2 DESCRIPTION

The use of machine learning algorithms such as SVM, KNN, Random Forest, and Naive Bayes for mitigating DDoS attacks in IoT is a novel approach that has gained significant attention in recent years. These algorithms are capable of detecting and preventing attacks in real-time, making them more effective than traditional methods. They can also adapt to changing attack patterns, making them more tough against evolving threats. These algorithms have ability to learn from large amounts of data and identify patterns that may not be immediately apparent to human analysts. This allows them to detect and respond to attacks more quickly and accurately, reducing the risk of damage to IoT devices and networks.

3.3 USE CASE DIAGRAM

In the use case diagram, we have basically one actor who is the user in the trained model.

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

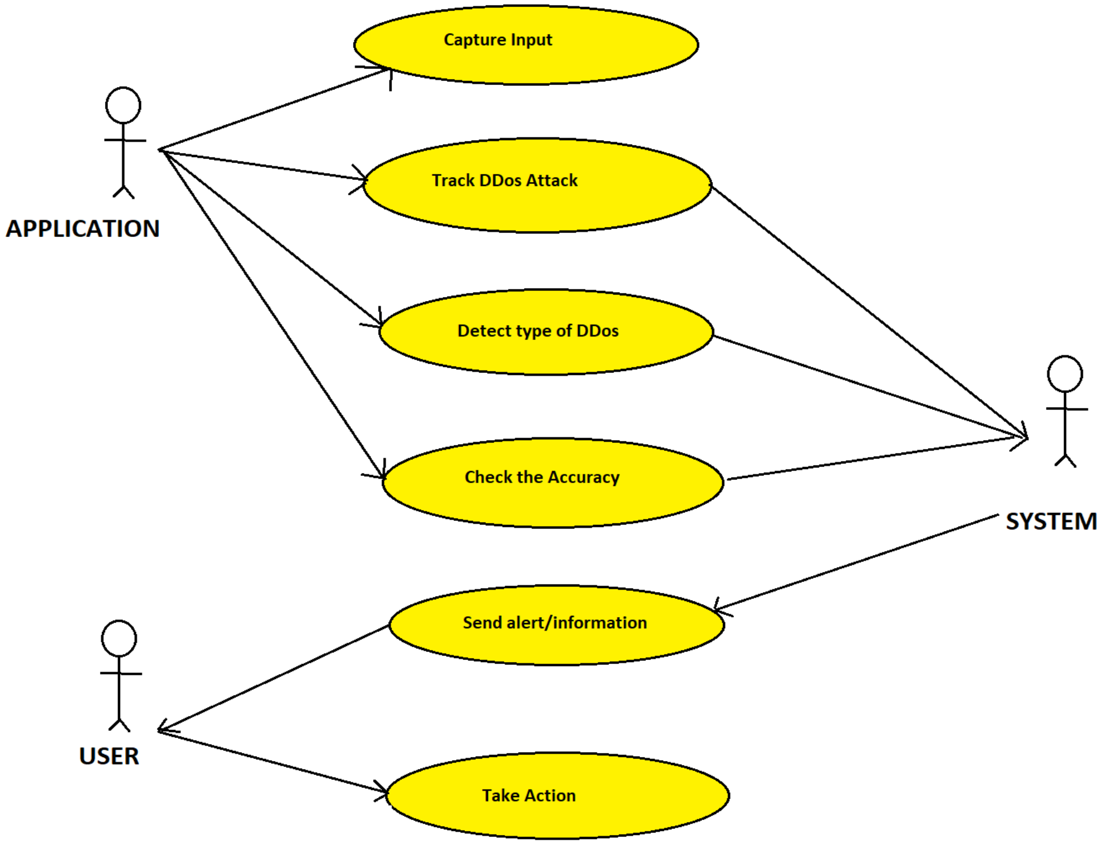


Figure 3.2: Use Case Diagram for mitigation of DDoS attack in IoT network environment

3.4 CLASS DIAGRAM

Class diagram is a type of 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.

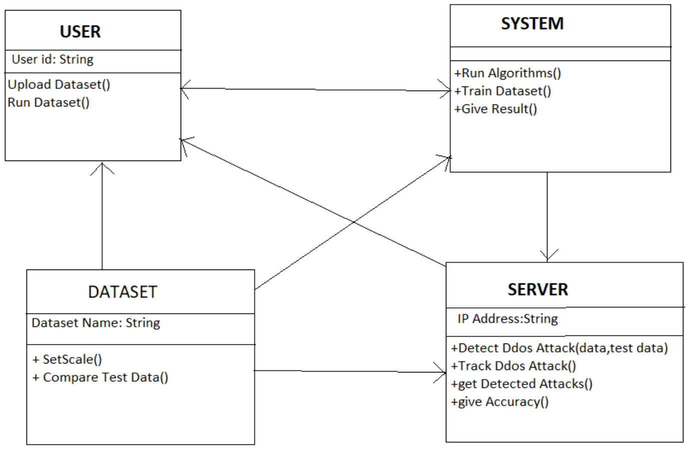


Figure 3.3: Class Diagram for mitigation of DDoS attack in IoT network environment

3.5 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development.

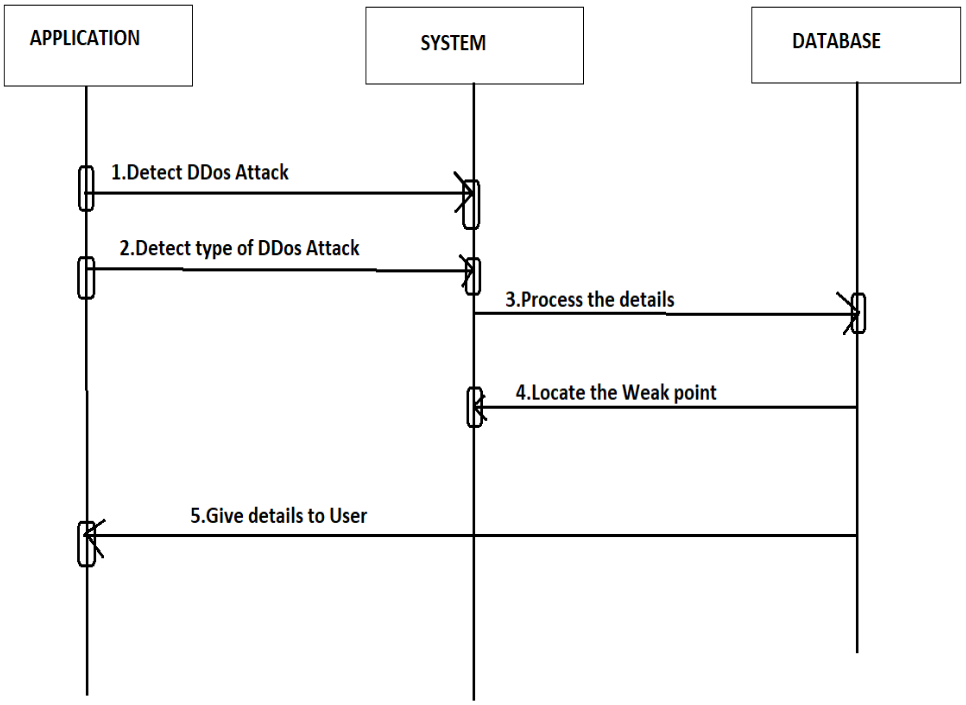


Figure 3.4:Sequence Diagram for mitigation of DDoS attack in IoT network environment

3.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. They can also include elements showing the flow of data between activities through one or more data stores.

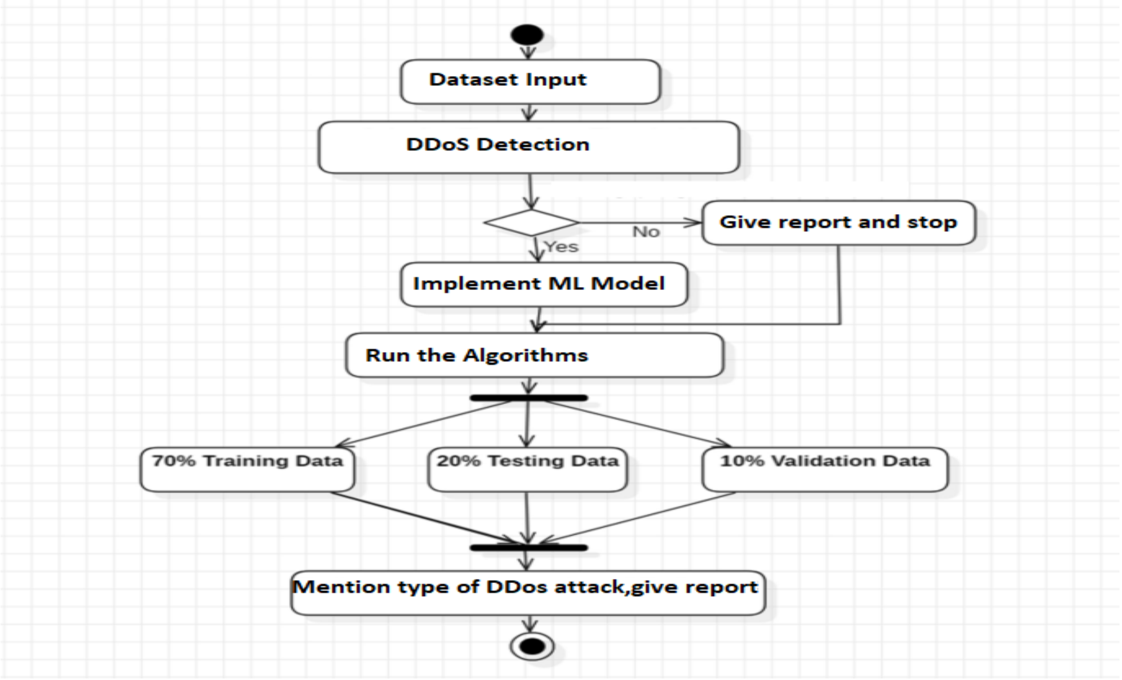


Figure 3.5: Activity Diagram for mitigation of DDoS attack in IoT network environment

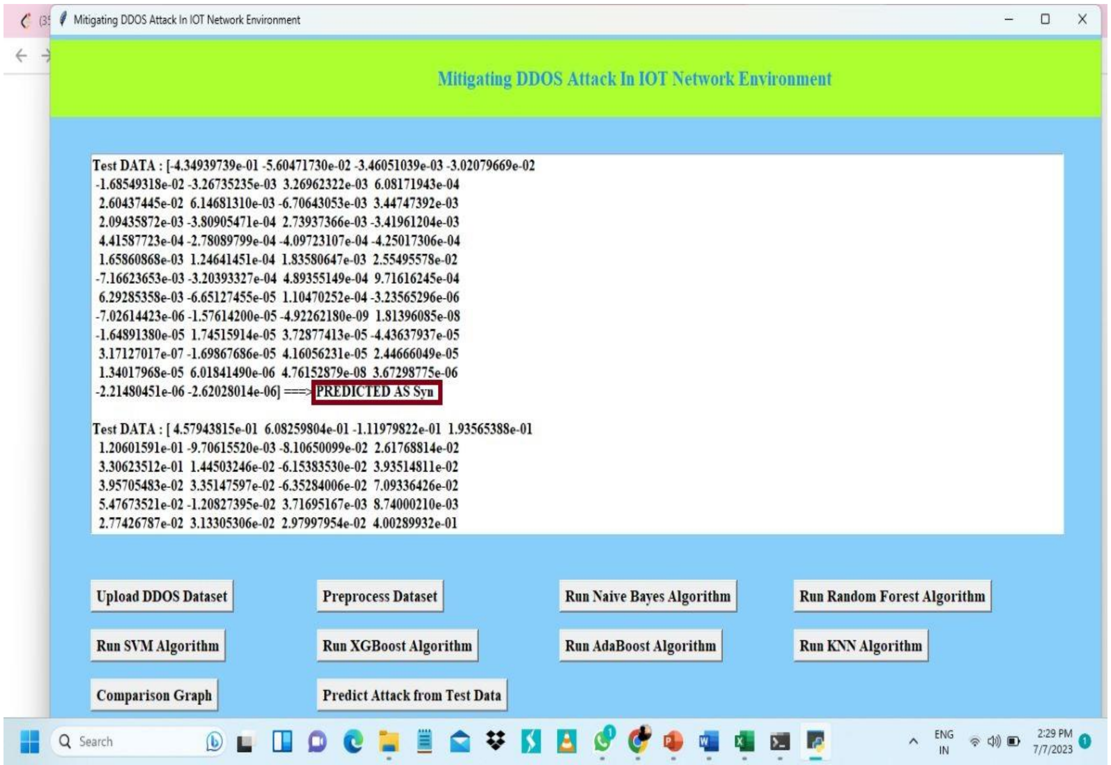
4. IMPLEMENTATION

4.1 SAMPLE CODE

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import os
import xgboost as xgb
global filename
global X,Y
global dataset
global main
global text
global accuracy, precision, recall, fscore
global X_train, X_test, y_train, y_test
global classifier
global label_encoder, labels, columns, types, pca
main = tkinter.Tk()
main.title("Mitigating DDOS Attack In IOT Network Environment")
#designing main screen
main.geometry("1300x1200")

def runRandomForest():
global classifier
if os.path.exists('model/rf.txt'):
with open('model/rf.txt', 'rb') as file:
rf = pickle.load(file)
file.close()
else:
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
with open('model/rf.txt', 'wb') as file:
pickle.dump(rf, file)
file.close()
predict = rf.predict(X_test)
classifier = rf
calculateMetrics("Random Forest", predict, y_test)

def runKNN():
if os.path.exists('model/knn.txt'):
with open('model/knn.txt', 'rb') as file:
knn_cls = pickle.load(file)
file.close()
```

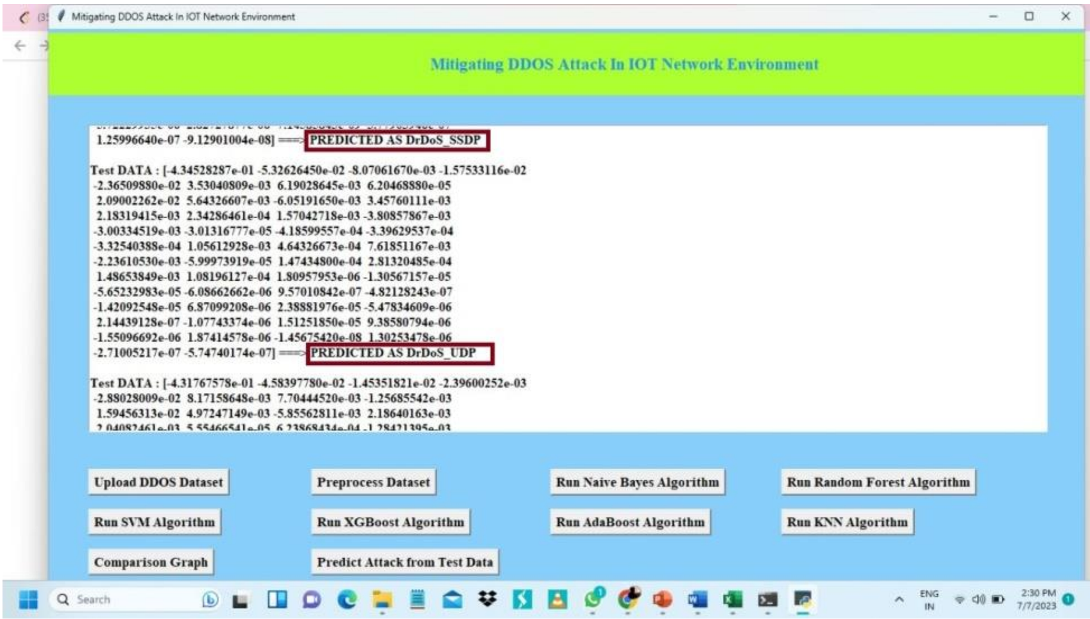
```
else:

  knn_cls = KNeighborsClassifier(n_neighbors = 2)
  knn_cls.fit(X_train, y_train)
  with open('model/knn.txt', 'wb') as file:
  pickle.dump(knn_cls, file)
  file.close()
  predict = knn_cls.predict(X_test)

  def runAdaBoost():
  if os.path.exists('model/adb.txt'):
  with open('model/adb.txt', 'rb') as file:
  adb_cls = pickle.load(file)
  file.close()
  else:
  adb_cls = AdaBoostClassifier()
  adb_cls.fit(X_train, y_train)
  with open('model/adb.txt', 'wb') as file:
  pickle.dump(adb_cls, file)
  file.close()
  predict = adb_cls.predict(X_test)
  calculateMetrics("AdaBoost", predict, y_test)
  calculateMetrics("KNN", predict, y_test)
```

5. SCREENSHOTS


Mitigating DDOS Attack In IOT Network Environment
Mitigating DDOS Attack In IOT Network Environment
Upload DDOS Dataset
Preprocess Dataset
Run Naive Bayes Algorithm
Run Random Forest Algorithm
Run SVM Algorithm
Run XGBoost Algorithm
Run AdaBoost Algorithm
Run KNN Algorithm
Comparison Graph
Predict Attack from Test Data
Search
ENG
IN
2:29 PM
7/7/2023

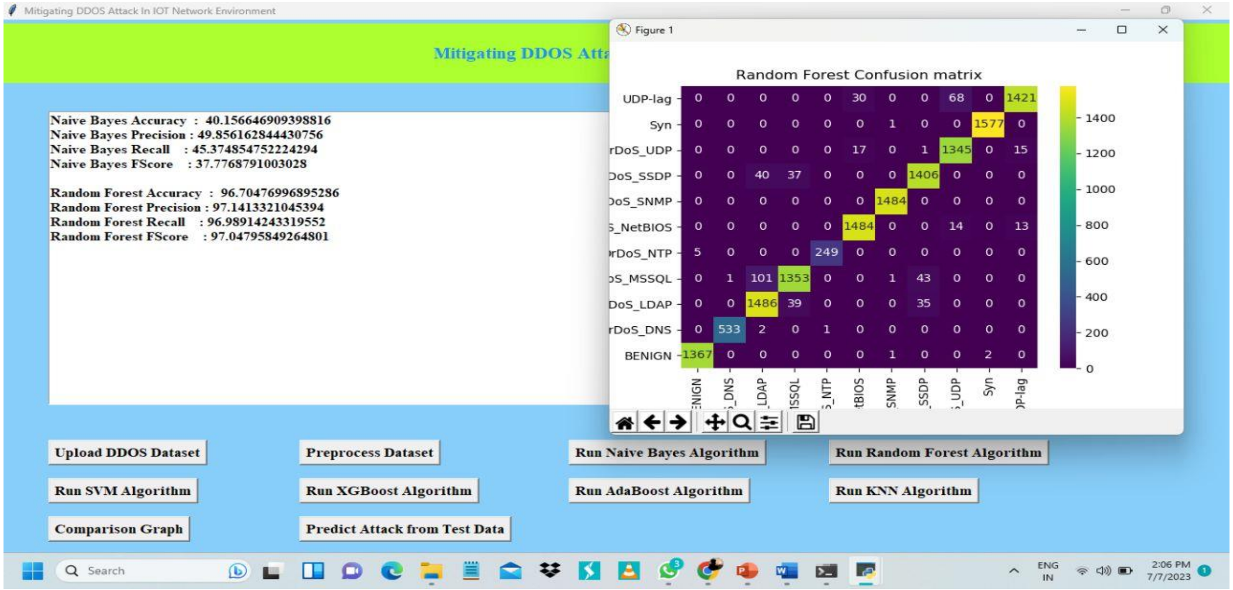
```
Test DATA : [-4.34939739e-01 -5.60471730e-02 -3.46051039e-03 -3.02079669e-02
 -1.68549318e-02 -3.26735235e-03  3.26962322e-03  6.08171943e-04
  2.60437445e-02  6.14681310e-03 -6.70643053e-03  3.44747392e-03
  2.09435872e-03 -3.80905471e-04  2.73937366e-03 -3.41961204e-03
  4.41587723e-04 -2.78089799e-04 -4.09723107e-04 -4.25017306e-04
  1.65860868e-03  1.24641451e-04  1.83580647e-03  2.55495578e-02
 -7.16623653e-03 -3.20393327e-04  4.89355149e-04  9.71616245e-04
  6.29285358e-03 -6.65127455e-05  1.10470252e-04 -3.23565296e-06
 -7.02614423e-06 -1.57614200e-05 -4.92262180e-09  1.81396085e-08
 -1.64891380e-05  1.74515914e-05  3.72877413e-05 -4.43637937e-05
  3.17127017e-07 -1.69867686e-05  4.16056231e-05  2.44666049e-05
  1.34017968e-05  6.01841490e-06  4.76152879e-08  3.67298775e-06
 -2.21480451e-06 -2.62028014e-06] ===> PREDICTED AS Syn

Test DATA : [ 4.57943815e-01  6.08259804e-01 -1.11979822e-01  1.93565388e-01
  1.20601591e-01 -9.70615520e-03 -8.10650099e-02  2.61768814e-02
  3.30623512e-01  1.44503246e-02 -6.15383530e-02  3.93514811e-02
  3.95705483e-02  3.35147597e-02 -6.35284006e-02  7.09336426e-02
  5.47673521e-02 -1.20827395e-02  3.71695167e-03  8.74000210e-03
  2.77426787e-02  3.13305306e-02  2.97997954e-02  4.00289932e-01
```

Screenshot 5.1: Predicting Syn DDoS attack


Mitigating DDOS Attack In IOT Network Environment
Mitigating DDOS Attack In IOT Network Environment
Upload DDOS Dataset
Preprocess Dataset
Run Naive Bayes Algorithm
Run Random Forest Algorithm
Run SVM Algorithm
Run XGBoost Algorithm
Run AdaBoost Algorithm
Run KNN Algorithm
Comparison Graph
Predict Attack from Test Data
Search
ENG
IN
2:30 PM
7/7/2023

```
  1.25996640e-07 -9.12901004e-08] ===> PREDICTED AS DrDoS_SSDP

Test DATA : [-4.34528287e-01 -5.32626450e-02 -8.07061670e-03 -1.57533116e-02
 -2.36509880e-02  3.53040809e-03  6.19028645e-03  6.20468880e-05
  2.09002262e-02  5.64326607e-03 -6.05191650e-03  3.45760111e-03
  2.18319415e-03  2.34286461e-04  1.57042718e-03 -3.80857867e-03
 -3.00334519e-03 -3.01316777e-05 -4.18599557e-04 -3.39629537e-04
 -3.32540388e-04  1.05612928e-03  4.64326673e-04  7.61851167e-03
 -2.23610530e-03 -5.99973919e-05  1.47434800e-04  2.81320485e-04
  1.48653849e-03  1.08196127e-04  1.80957953e-06 -1.30567157e-05
 -5.65232983e-05 -6.08662662e-06  9.57010842e-07 -4.82128243e-07
 -1.42092548e-05  6.87099208e-06  2.38881976e-05 -5.47834609e-06
  2.14439128e-07 -1.07743374e-06  1.51251850e-05  9.38580794e-06
 -1.55096692e-06  1.87414578e-06 -1.45675420e-08  1.30253478e-06
 -2.71005217e-07 -5.74740174e-07] ===> PREDICTED AS DrDoS_UDP

Test DATA : [-4.31767578e-01 -4.58397780e-02 -1.45351821e-02 -2.39600252e-03
 -2.88028009e-02  8.17158648e-03  7.70444520e-03 -1.25685542e-03
  1.59456313e-02  4.97247149e-03 -5.85562811e-03  2.18640163e-03
  2.04082461e-03  5.55466541e-05  6.23868434e-04 -1.28421395e-03
```

Screenshot 5.2: Predicting SSDP and UDP DDoS attack

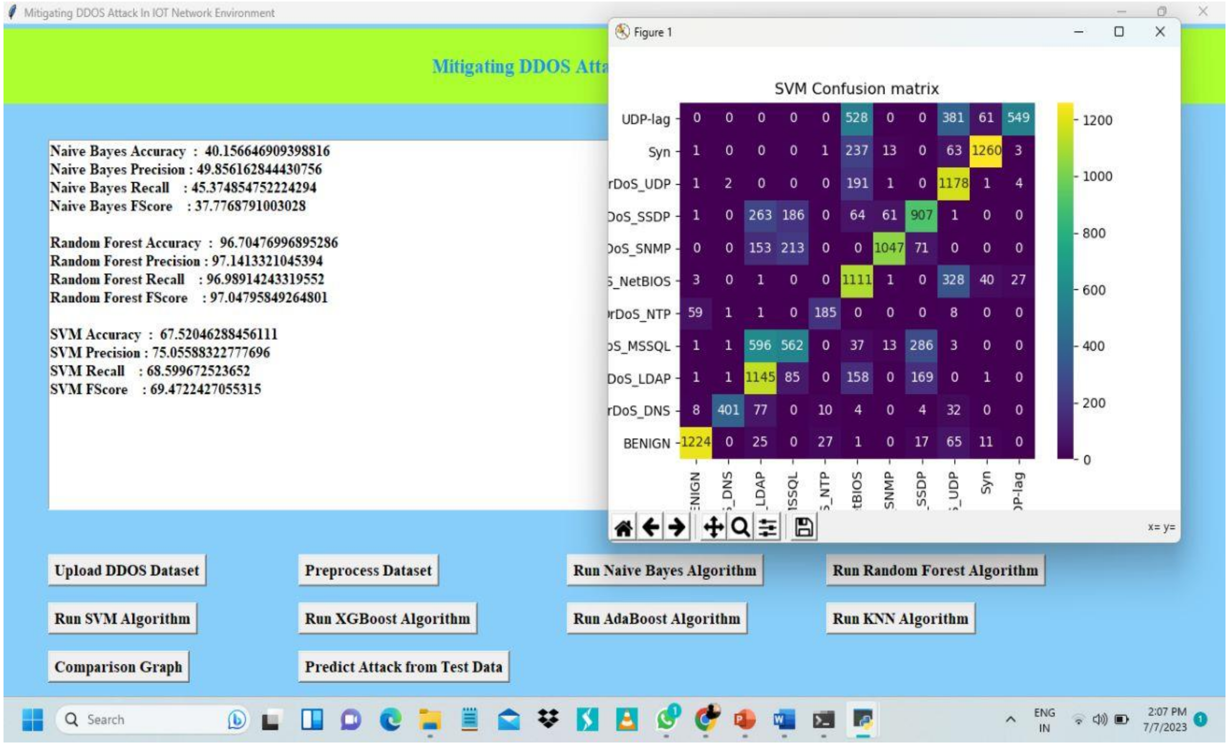


```
Naive Bayes Accuracy  :  40.156646909398816
Naive Bayes Precision : 49.856162844430756
Naive Bayes Recall    : 45.374854752224294
Naive Bayes FScore    : 37.7768791003028

Random Forest Accuracy  :  96.70476996895286
Random Forest Precision : 97.1413321045394
Random Forest Recall    : 96.98914243319552
Random Forest FScore    : 97.04795849264801
```

Random Forest Confusion matrix

	ENIGN	S_DNS	LDAP	ISSQL	S_NTP	tBIOS	SNMP	SSDP	S_UDP	Syn	P-lag
UDP-lag	0	0	0	0	0	30	0	0	68	0	1421
Syn	0	0	0	0	0	0	1	0	0	1577	0
rDoS_UDP	0	0	0	0	0	17	0	1	1345	0	15
DoS_SSDP	0	0	40	37	0	0	0	1406	0	0	0
DoS_SNMP	0	0	0	0	0	0	1484	0	0	0	0
S_NetBIOS	0	0	0	0	0	1484	0	0	14	0	13
rDoS_NTP	5	0	0	0	249	0	0	0	0	0	0
oS_MSSQL	0	1	101	1353	0	0	1	43	0	0	0
DoS_LDAP	0	0	1486	39	0	0	0	35	0	0	0
rDoS_DNS	0	533	2	0	1	0	0	0	0	0	0
BENIGN	1367	0	0	0	0	0	1	0	0	2	0

Screenshot 5.3: Random forest confusion matrix for given dataset

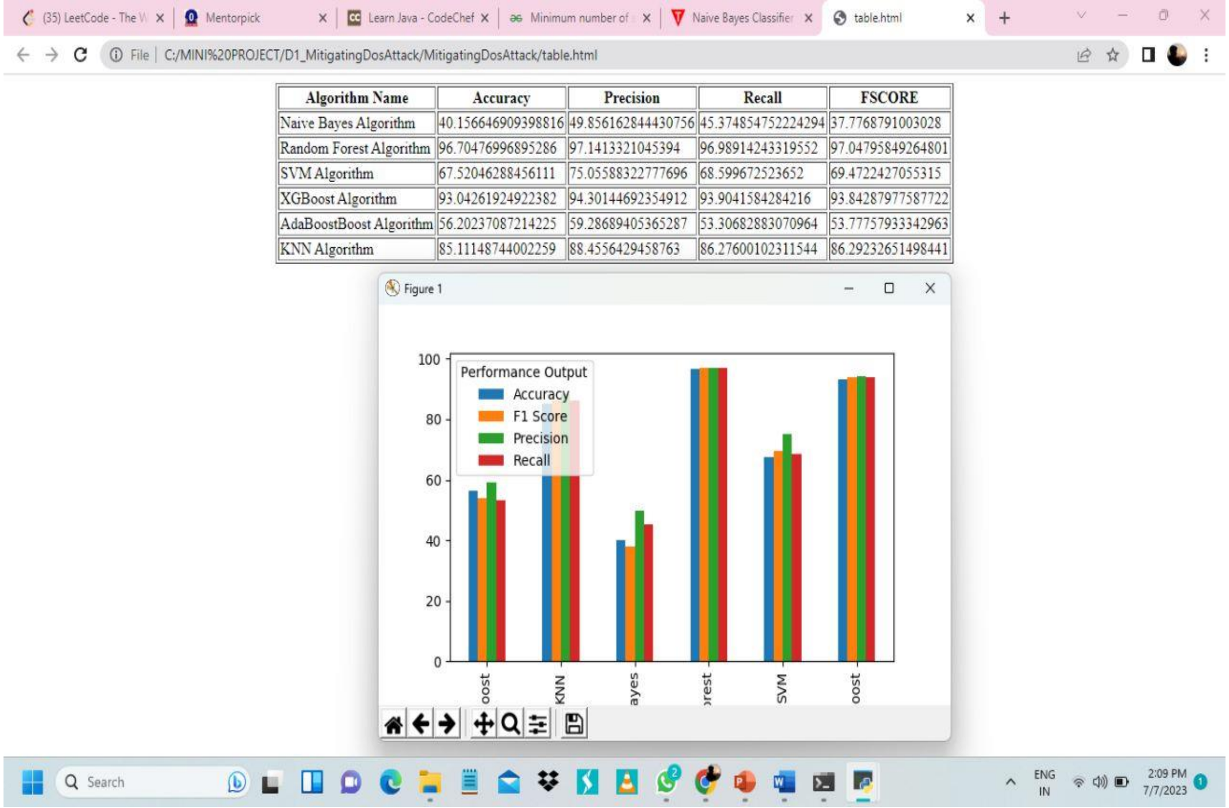


```
Naive Bayes Accuracy  :  40.156646909398816
Naive Bayes Precision : 49.856162844430756
Naive Bayes Recall    : 45.374854752224294
Naive Bayes FScore    : 37.7768791003028

Random Forest Accuracy  :  96.70476996895286
Random Forest Precision : 97.1413321045394
Random Forest Recall    : 96.98914243319552
Random Forest FScore    : 97.04795849264801

SVM Accuracy  :  67.52046288456111
SVM Precision : 75.05588322777696
SVM Recall    : 68.599672523652
SVM FScore    : 69.4722427055315
```

SVM Confusion matrix

	ENIGN	S_DNS	LDAP	ISSQL	S_NTP	tBIOS	SNMP	SSDP	S_UDP	Syn	P-lag
UDP-lag	0	0	0	0	0	528	0	0	381	61	549
Syn	1	0	0	0	1	237	13	0	63	1260	3
rDoS_UDP	1	2	0	0	0	191	1	0	1178	1	4
DoS_SSDP	1	0	263	186	0	64	61	907	1	0	0
DoS_SNMP	0	0	153	213	0	0	1047	71	0	0	0
S_NetBIOS	3	0	1	0	0	1111	1	0	328	40	27
rDoS_NTP	59	1	1	0	185	0	0	0	8	0	0
oS_MSSQL	1	1	596	562	0	37	13	286	3	0	0
DoS_LDAP	1	1	1145	85	0	158	0	169	0	1	0
rDoS_DNS	8	401	77	0	10	4	0	4	32	0	0
BENIGN	1224	0	25	0	27	1	0	17	65	11	0

Screenshot 5.4: SVM confusion matrix for given dataset



Algorithm Name	Accuracy	Precision	Recall	FSCORE
Naive Bayes Algorithm	40.156646909398816	49.856162844430756	45.374854752224294	37.7768791003028
Random Forest Algorithm	96.70476996895286	97.1413321045394	96.98914243319552	97.04795849264801
SVM Algorithm	67.52046288456111	75.05588322777696	68.599672523652	69.4722427055315
XGBoost Algorithm	93.04261924922382	94.30144692354912	93.9041584284216	93.84287977587722
AdaBoostBoost Algorithm	56.20237087214225	59.28689405365287	53.30682883070964	53.77757933342963
KNN Algorithm	85.11148744002259	88.4556429458763	86.27600102311544	86.29232651498441

Screenshot 5.5: Comparison graph for predicting accuracy

6. TESTING

6.TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid : identified classes of invalid input must Input be rejected.

Functions : identified functions must be exercised.

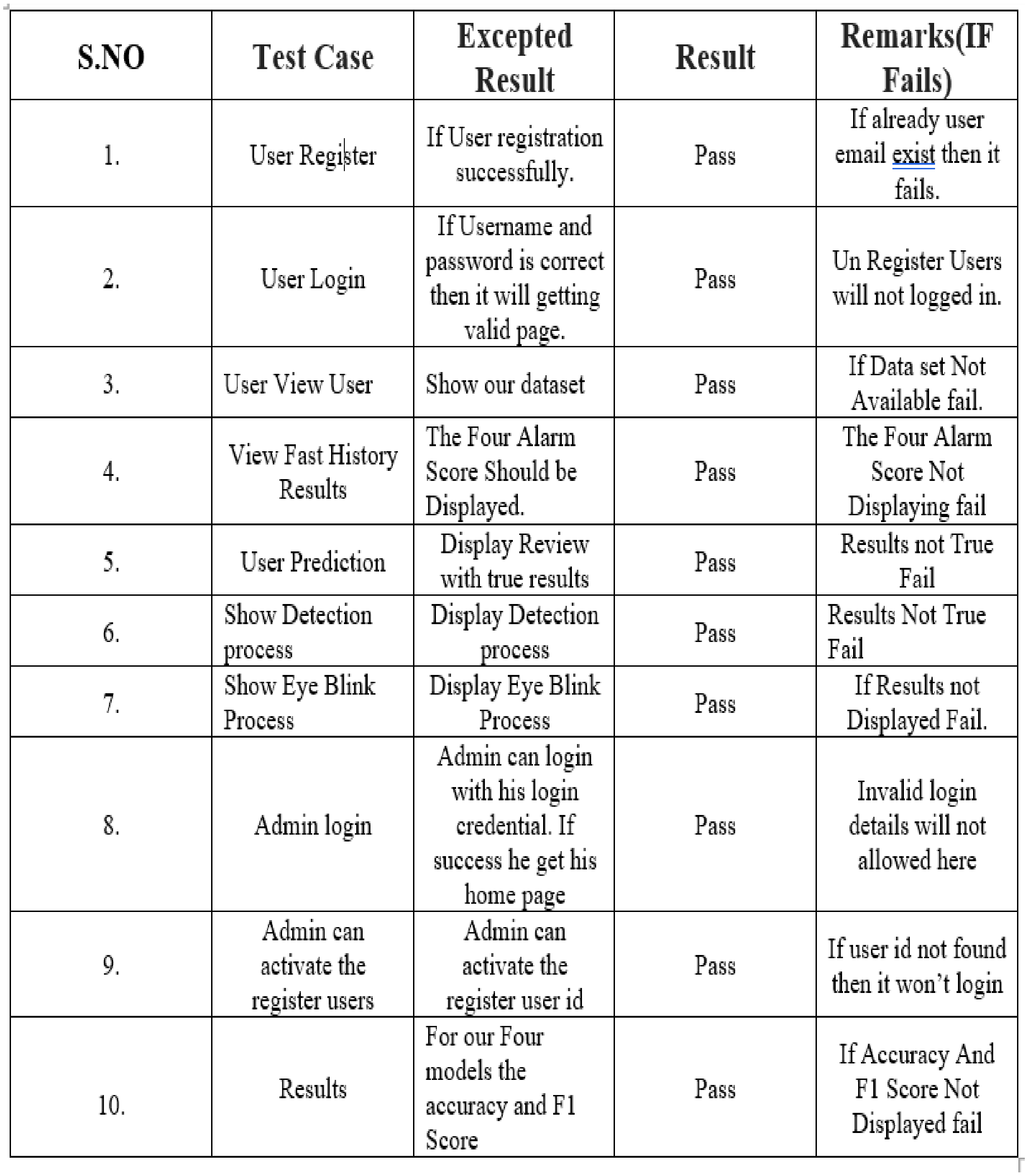
Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.3 TEST CASES

6.3.1 CLASSIFICATION



S.NO	Test Case	Excepted Result	Result	Remarks(IF Fails)
1.	User Register	If User registration successfully.	Pass	If already user email exist then it fails.
2.	User Login	If Username and password is correct then it will getting valid page.	Pass	Un Register Users will not logged in.
3.	User View User	Show our dataset	Pass	If Data set Not Available fail.
4.	View Fast History Results	The Four Alarm Score Should be Displayed.	Pass	The Four Alarm Score Not Displaying fail
5.	User Prediction	Display Review with true results	Pass	Results not True Fail
6.	Show Detection process	Display Detection process	Pass	Results Not True Fail
7.	Show Eye Blink Process	Display Eye Blink Process	Pass	If Results not Displayed Fail.
8.	Admin login	Admin can login with his login credential. If success he get his home page	Pass	Invalid login details will not allowed here
9.	Admin can activate the register users	Admin can activate the register user id	Pass	If user id not found then it won't login
10.	Results	For our Four models the accuracy and F1 Score	Pass	If Accuracy And F1 Score Not Displayed fail

7.CONCLUSION

7.CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

Machine learning algorithms like SVM, KNN, Random Forest, and Naive Bayes provide significant advantages for mitigating DDoS attacks in IoT. These algorithms are capable of detecting and preventing attacks in real-time, making them more effective than traditional methods. Additionally, they can adapt to changing attack patterns, making them more resilient against evolving threats. By implementing these algorithms, organizations can significantly improve their security posture and protect their IoT devices from malicious attacks. As the number of connected devices continues to grow, it is essential to consider these advanced security measures to ensure the safety and privacy of sensitive data.

7.2 FUTURE SCOPE

Integration with Existing Security Solutions:

The project could lead to the integration of machine learning-based DDoS mitigation techniques into existing security products and services. This integration can enhance the overall cybersecurity.

Real-time Detection:

Enhance project's ability to detect and respond to DDoS attacks in real-time. This can involve optimizing algorithms for implementing streaming data processing, and integrating with network monitoring tools.

Anomaly Detection:

Exploring anomaly detection techniques to identify unusual traffic patterns that might indicate DDoS attacks.

Data Augmentation:

Experiment with data augmentation techniques to increase the diversity and size of your training dataset. This can lead to better generalization and robustness of machine learning models.

Large-scale Testing:

Conduct extensive testing of the system in a controlled environment to assess its scalability and performance under heavy DDoS attack scenarios.

8.BIBLIOGRAPHY

8.BIBLIOGRAPHY

8.1 REFERENCES

[1] I. Ahmed, A. Mahmood, S. Hu, and H. Hu "Anomaly-Based DDoS Detection and Mitigation in IoT Networks: A Review", IEEE Access, 2021.

[2] A. Amatullah, S. Gupta, and M. H. Bhuyan "A Survey of DDoS Attacks and Defense Mechanisms in IoT", IEEE Internet of Things Journal, 2020.

[3] S. Kaushik, M. S. Obaidat, and N. Gupta "Detecting DDoS Attacks in IoT Networks: A Machine Learning-Based Approach", Future Generation Computer Systems, 2019.

[4] S. Mahbub, M. H. Bhuiyan, and S. Roy "Machine Learning for IoT Security: Improving Network Intrusion Detection",IEEE Internet of Things Journal, 2017.

8.2 GITHUB LINK

https://github.com/venkatesh944/mitigating_ddos_attack