A

Mini Project

On

MACHINE LEARNING TECHNIQUES FOR CYBER ATTACKS DETECTION

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

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

K SUPRIYA (217R5A0506)

G SHIVARANI (217R5A0505)

G ROHITH (207R1A0514)

Under the Guidance of

RAHEEM UNNISA

(Assistant Professor)

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

CERTIFICATE

This is to certify that the project entitled **"MACHINE LEARNING TECHNIQUES FOR CYBER ATTACKS DETECTION"** being submitted by **K SUPRIYA(217R5A0506), G SHIVARANI(217R5A0505) & G ROHITH(207R1A0514)** in partial fulfilment of the requirements for the award of the Degree of B.Tech in Computer Science and Engineering to the CMR Technical Campus, is a record of bonofide 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 University or Institute for the award of any degree or diploma.

Raheem Unnisa   **Dr. A. Raji Reddy**

(Assistant Professor)   DIRECTOR

(INTERNAL GUIDE

Dr. K. Srujan Raju   **EXTERNAL EXAMINER**

HOD

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 largely on 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 **Raheem Unnisa,** Assistant Professor for her exemplary guidance, monitoring and constant constant encouragement throughout the project work. The blessing, help and guidance given by her 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.

K SUPRIYA   (217R5A0506)
G SHIVARANI   (217R5A0505)
G ROHITH   (207R1A05

ABSTRACT

The present-day world has become all dependent on cyber space for every aspect of daily living. The use of cyberspace is rising with each passing day. The world is spending more time on the internet than ever before. As a result, the risks of cyber threats and cybercrimes are increasing. Cybercriminals are changing their techniques with time to pass through the wall of protection. Conventional Techniques are not capable of detecting zero-day attacks and sophisticated attacks.

Thus, far heaps of machine learning techniques have been developed to detect cyber crimes and battle against cyberthreats. The objective of this project is to present the evaluation of some of the widely used machine learning techniques used to detect some of the most threatening cyber threats to the cyber space. Three primary machine learning techniques are mainly investigated, including deep belief network, decision tree and support vector machine.

LIST OF FIGURES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1.INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

This project titled " MACHINE LEARNING TECHNIQUES FOR CYBER ATTACK DETECTION" includes developing and implementing a predictive model that analyzes network traffic patterns and system logs to identify and classify potential threats. The project aims to enhance cybersecurity by automating the detection of various types of cyber attacks, such as malware, intrusion attempts, and anomalous behaviour, thereby improving threat mitigation and reducing the risk of data breaches.

1.2 PROJECT PURPOSE

The purpose of this project is to leverage machine learning techniques to bolster cybersecurity by proactively identifying and mitigating cyber threats. By developing predictive models, the project aims to enhance the detection and response capabilities, reducing the risk of data breaches, financial losses, and reputational damage. Ultimately, the project seeks to create a more secure digital environment for organizations and individuals.

1.3 PROJECT FEATURE

The project features for machine learning techniques in cyber attack detection include real-time monitoring of network traffic and system logs for immediate threat identification. It incorporates a diverse set of machine learning algorithms to analyze and classify different types of cyber threats, from malware and phishing to advanced persistent threats. Furthermore, it provides a user-friendly interface for security analysts to investigate detected anomalies and initiate response procedures swiftly.

2.SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

2.1 SYSTEM ANALYSIS

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. 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

Machine learning techniques for cyber attack detection revolves around Creating intelligent, adaptable, and scalable systems that can accurately identify and classify cyber threats while minimizing false positives and offering proactive insights to enhance overall. cybersecurity in an ever-evolving digital landscape.

2.3 EXISTING SYSTEM

The existing system for machine learning techniques in cyber attack detection relies on traditional signature-based detection methods and rule-based approaches. It primarily uses predefined patterns and known threat indicators to identify cyber threats. While this system is effective against well-known and documented attacks.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

Following are the disadvantages of existing System

- Cost of Implementation
- Lack of Content
- Limited Historical Data
- Scalability for Real- Time Detection
- Data Privacy Concerns

2.3 PROPOSED SYSTEM

The proposed system for cyber attack detection through machine learning techniques will leverage state-of-the-art algorithms for real-time threat identification. It will incorporate anomaly detection models, deep learning, and ensemble methods to enhance accuracy and reduce false positives. The system will be designed with scalability in mind offering seamless integration with existing security infrastructure, and it will emphasize continuous learning and adaptability to address evolving cyber threats effectively.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- Continuous Monitoring
- Multi-Layered Defence
- Pattern Recognition
- Reduced Human Error
- Identifying Unknown Threats
- Continuous Improvement

2.4 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,

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

2.4.1 ECONOMIC 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 . In to 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.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that

Is Technical requirements of the system. Any system developed must not have a High demand on the available technical resources. This will lead to high On the available technical resources. This will lead to high demands being   Placed the client. The developed system must have a modest requirements, As only minimal or null changes are required for implementing this system.

2.4.3 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 hi m 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 this is the final user of the system.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

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

- Processor : Intel core I3 or above.
- RAM : 256 MB or above.
- Space on Hard Disk: Minimum 512 MB

2.5.2 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, Windows XP, Windows 8
- Languages   : Python, Django, MySql, WampServer 2.4

3.ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITETURE

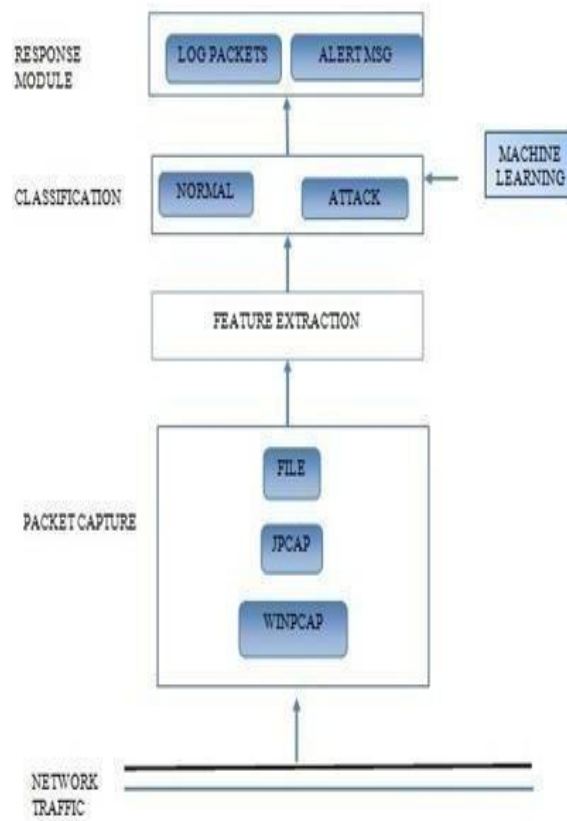The figure shows the architecture of this project

Figure3.1: Architecture of Machine Learning Techniques For Cyber Attack Detection.

3.2 DESCRIPTION

Machine learning techniques for cyber attack detection employ a sophisticated architecture that combines data processing, feature extraction, model training, and real-time monitoring. This architecture begins with the ingestion of vast amounts of network data, including logs, packets, and metadata, which is then preprocessed to clean and normalize the information. Feature extraction algorithms play a pivotal role in converting this raw data into meaningful representations that can be used by machine learning models.

The core of the architecture lies in the machine learning models themselves, which are trained on historical data to learn patterns of normal network behavior and various attack types. These models can include supervised learning algorithms such as Random Forests, Support Vector Machines, or deep learning models like Convolutional Neural Networks or Recurrent Neural Networks. Continuous model training and updating are essential to adapt to the ever-evolving threat landscape.

The core of the architecture lies in the machine learning models themselves, which are trained on historical data to learn patterns of normal network behaviour and various attack types. These models can include supervised learning algorithms such as Random Forests, Support Vector Machines, or deep learning models like Convolutional Neural Networks or Recurrent Neural Networks. Continuous model training and updating are essential to adapt to the ever-evolving threat landscape.

DJANGO

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source. Django's primary goal is to ease the creation of complex, databasedriven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models.
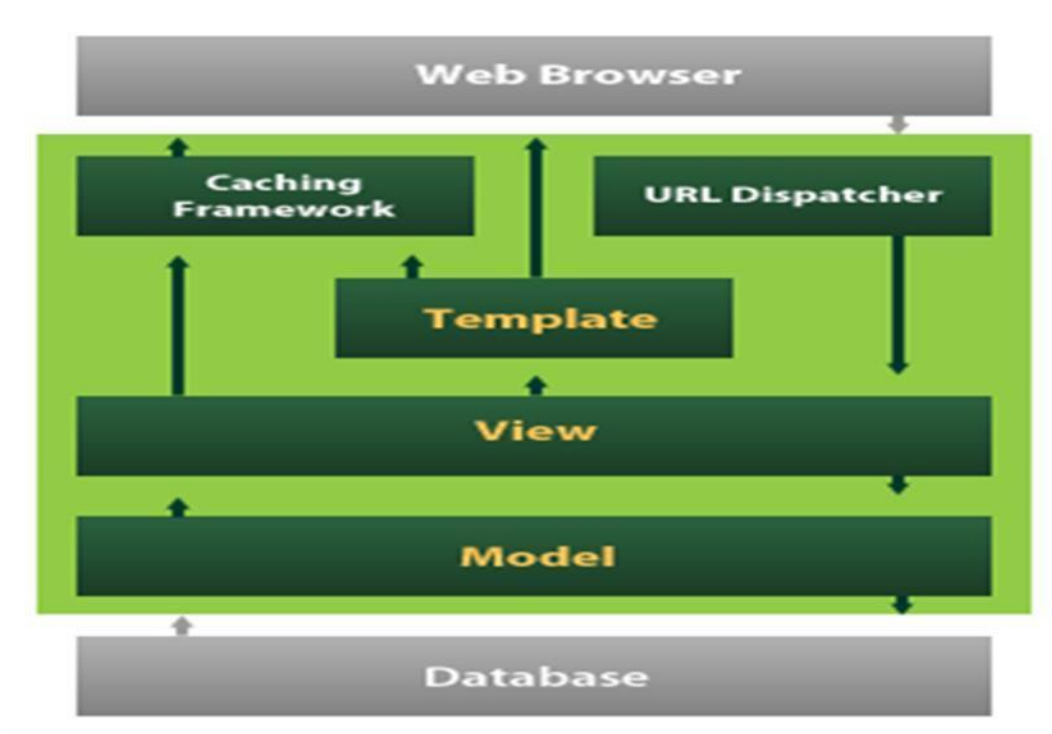
Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.
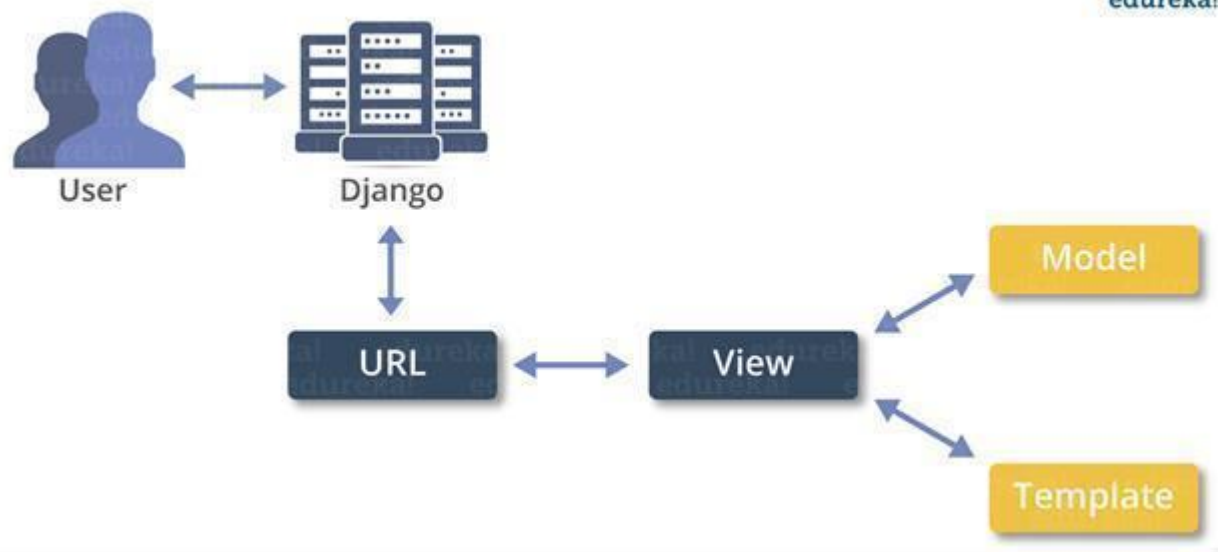
ABOUT PYTHON

Python is a programming language, which means it's a language both people and computers can understand. Python was developed by a Dutch software engineer named Guido van Rossum, who created the language to solve some problems he saw in computer languages of the time.

Python is an interpreted high-level programming language for generalpurpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, and a syntax that allows programmers to express concepts in fewer lines of code, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. C Python, the reference implementation of Python, is an open-source software and has a community-based development model, as do nearly all of its variant implementations. C Python is managed by the non-profit Python Software Foundation.

You Can Use Python for Pretty Much Anything

One significant advantage of learning Python is that it is general-purpose language that can be applied in a large variety of projects. Below are just some of the most common fields where Python has found its use:

- Data science
- Scientific and mathematical computing
- Web development
- Computer graphics
- Basic game development
- Mapping and geography (GIS software)

Python Is Widely Used in Data Science

Python's ecosystem has been growing over the years and it's more and more capable of statistical analysis. It's the best compromise between scale and sophistication (in terms of data processing). Python emphasizes productivity and readability. Python is used by programmers that want to delve into data analysis or apply statistical techniques (and by Devs that turn to data science).

There are plenty of Python scientific packages for data visualization, machine learning, natural language processing, complex data analysis and more. All of these factors make Python a great tool for scientific computing and a solid alternative for commercial packages such as MATLAB.

3.3 USE CASE DIAGRAM

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.
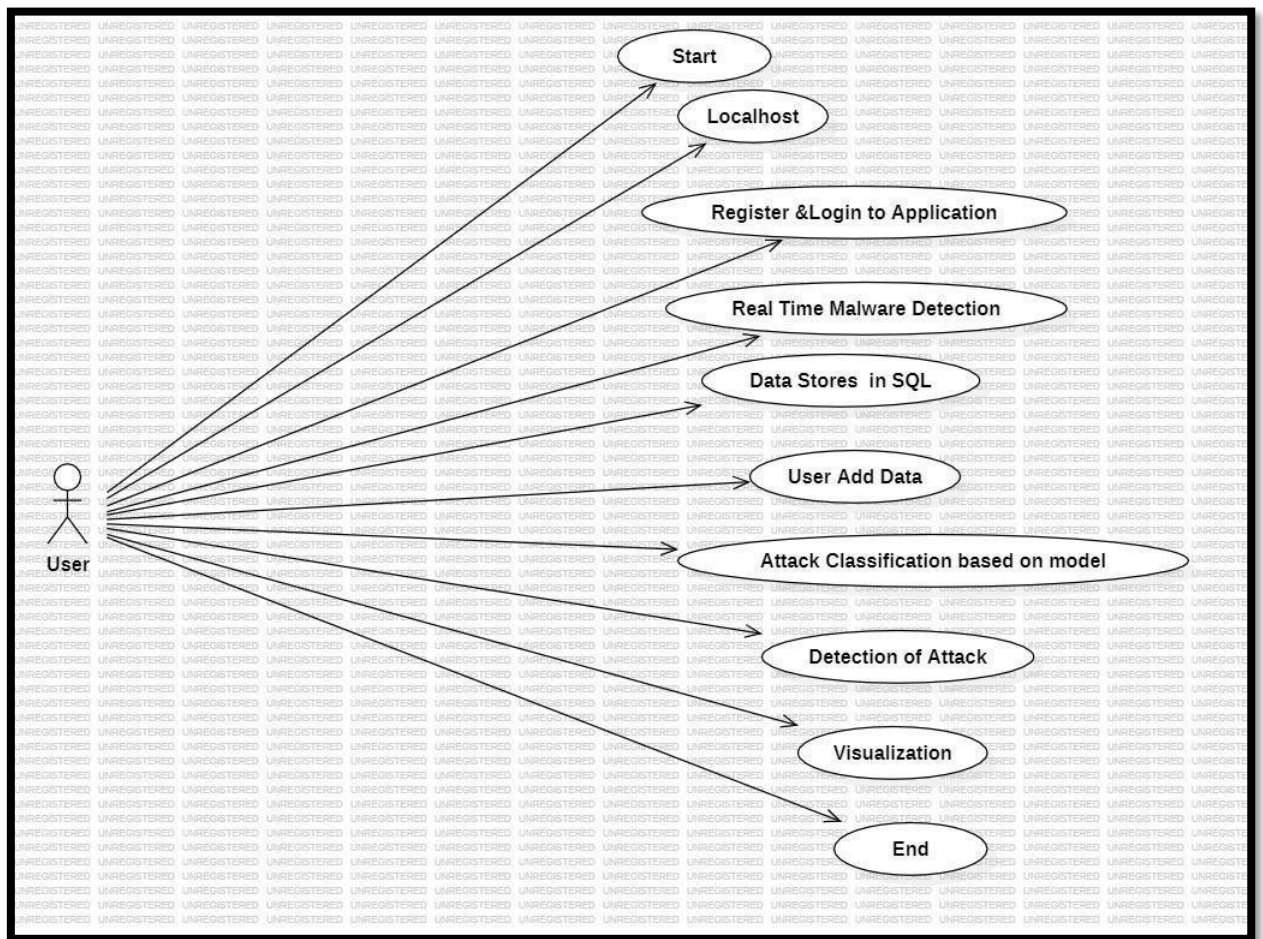

Figure3.3: Use Case Diagram of Machine Learning Techniques for Cyber Attacks Detection.

3.4 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static View of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.
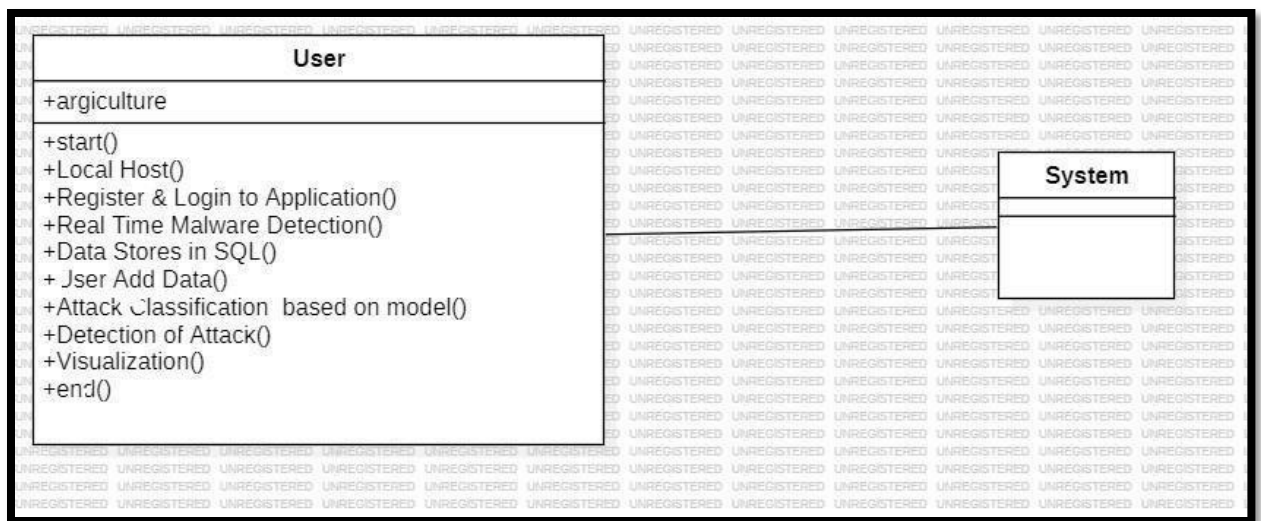

Figure3.4: Class Diagram of Machine Learning Techniques for Cyber Attacks Detection.

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.
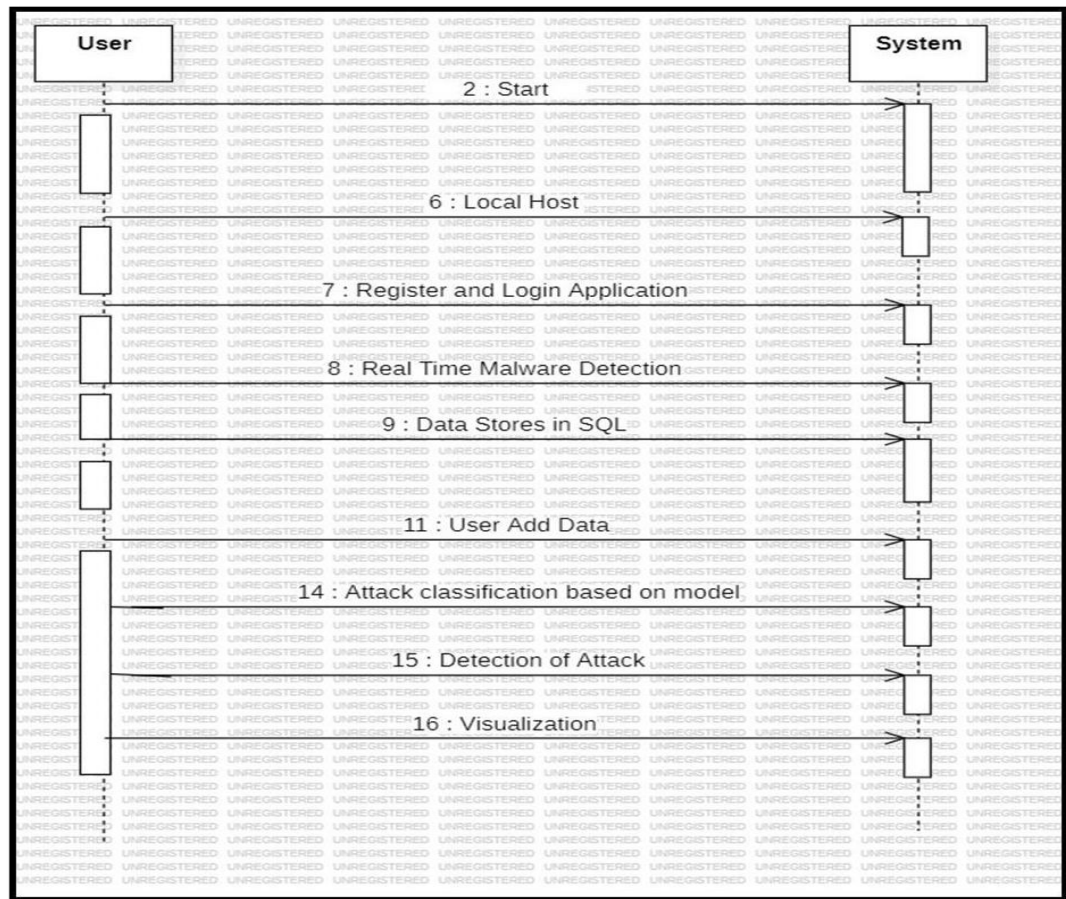

Figure3.5: Sequence Diagram for Machine Learning Techniques For Cyber Attacks Detection

4.IMPLEMENTATION

4.1 SAMPLE CODE

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

```python
import itertools
import seaborn as sns
import pandas_profiling
import statsmodels.formula.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor
from patsy import dmatrices
```
```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is dep
recated. Use the functions in the public API at pandas.testing instead.
  import pandas.util.testing as tm
```

```python
from sklearn import datasets
from sklearn.feature_selection import RFE
import sklearn.metrics as metrics
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2, f_classif, mutual_info_classif
```

```python
train=pd.read_csv('/content/drive/My Drive/kdd/NSL_Dataset/Train.txt',sep=',')
test=pd.read_csv('/content/drive/My Drive/kdd/NSL_Dataset/Test.txt',sep=',')
```

Data preprocessing

```python
In [6]: columns=["duration","protocol_type","service","flag","src_bytes","dst_bytes","land",
        "wrong_fragment","urgent","hot","num_failed_logins","logged_in",
        "num_compromised","root_shell","su_attempted","num_root","num_file_creations",
        "num_shells","num_access_files","num_outbound_cmds","is_host_login",
        "is_guest_login","count","srv_count","serror_rate", "srv_serror_rate",
        "rerror_rate","srv_rerror_rate","same_srv_rate", "diff_srv_rate","srv_diff_host_rate","dst_host_count","dst_host_sr
        "dst_host_diff_srv_rate","dst_host_same_src_port_rate",
        "dst_host_srv_diff_host_rate","dst_host_serror_rate","dst_host_srv_serror_rate",
        "dst_host_rerror_rate","dst_host_srv_rerror_rate","attack", "last_flag"]
```

```python
In [7]: train.columns=columns
        test.columns=columns
```

```python
In [8]: train.head()
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	num_failed_logins	logged_in	num_compromised	root_s
0	0	udp	other	SF	146	0	0	0	0	0	0	0	0	
1	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	
2	0	tcp	http	SF	232	8153	0	0	0	0	0	1	0	
3	0	tcp	http	SF	199	420	0	0	0	0	0	1	0	
4	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	

```python
In [9]: test.head()
```

```
# Protocol type distribution
plt.figure(figsize=(9,8))
sns.countplot(x="protocol_type", data=train)
plt.show()
```
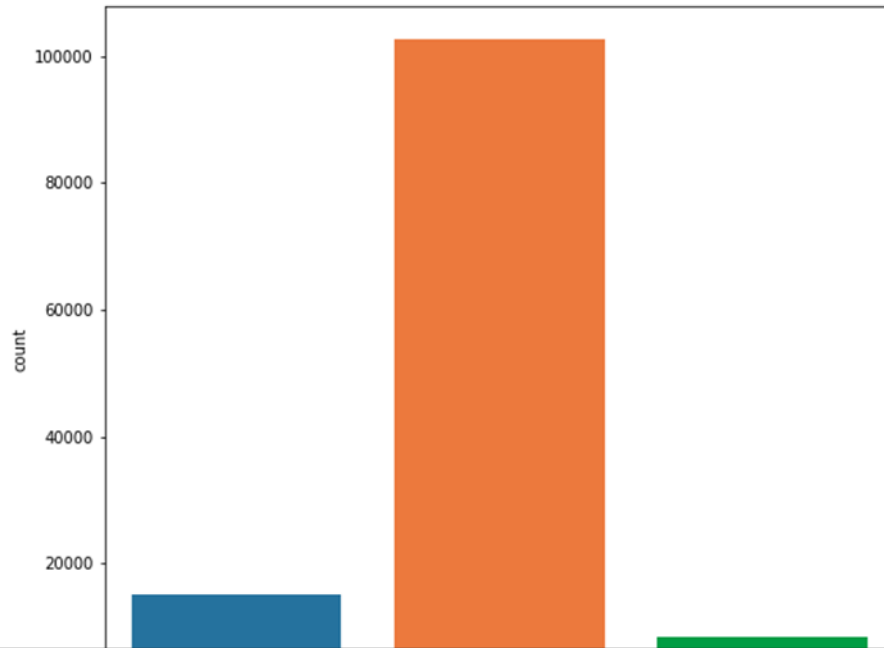


```
Model Building

train_X=train_new[cols]
train_y=train_new['attack_class']
test_X=test_new[cols]
test_y=test_new['attack_class']
```

ML Deploy

```
Logistic Regression

# Building Models
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression(random_state=0,solver='lbfgs',multi_class='multinomial')
logreg.fit( train_X, train_y)
logreg.predict(train_X)    #by default, it use cut-off as 0.5

list( zip( cols, logreg.coef_[0] ) )


logreg.intercept_


logreg.score(train_X,train_y)
```

```
Decision Trees


train_X.shape

param_grid = {'max_depth': np.arange(2, 12),
              'max_features': np.arange(10,15)}

train_y.shape

from sklearn.model_selection import GridSearchCV
from sklearn.tree import DecisionTreeClassifier, export_graphviz, export
tree = GridSearchCV(DecisionTreeClassifier(), param_grid, cv = 10,verbose=1,n_jobs=-1)
tree.fit( train_X, train_y )


tree.best_score_

tree.best_estimator_
tree.best_params_

train_pred = tree.predict(train_X)

print(metrics.classification_report(train_y, train_pred))

test_pred = tree.predict(test_X)
```

Random Forest

```python
from sklearn.ensemble import RandomForestClassifier
pargrid_rf = {'n_estimators': [50,60,70,80,90,100],
                'max_features': [2,3,4,5,6,7]}
```

```python
from sklearn.model_selection import GridSearchCV
gscv_rf = GridSearchCV(estimator=RandomForestClassifier(),
                          param_grid=pargrid_rf,
                          cv=10,
                          verbose=True, n_jobs=-1)

gscv_results = gscv_rf.fit(train_X, train_y)
```

```python
gscv_results.best_params_
```

```python
gscv_rf.best_score_
```

```python
radm_clf = RandomForestClassifier(oob_score=True,n_estimators=80, max_features=5, n_jobs=-1)
radm_clf.fit( train_X, train_y )
```

```python
radm_test_pred = pd.DataFrame( { 'actual':  test_y,
                                 'predicted': radm_clf.predict( test_X ) } )
```

Support Vector Machine (SVM)

```python
from sklearn.svm import LinearSVC
svm_clf = LinearSVC(random_state=0, tol=1e-5)
svm_clf.fit(train_X,train_y)
```

```python
print(svm_clf.coef_)
print(svm_clf.intercept_)
print(svm_clf.predict(train_X))
```

```python
from sklearn.svm import SVC
from sklearn.pipeline import make_pipeline

model = SVC(kernel='rbf', class_weight='balanced',gamma='scale')
```

```python
model.fit(train_X,train_y)
```
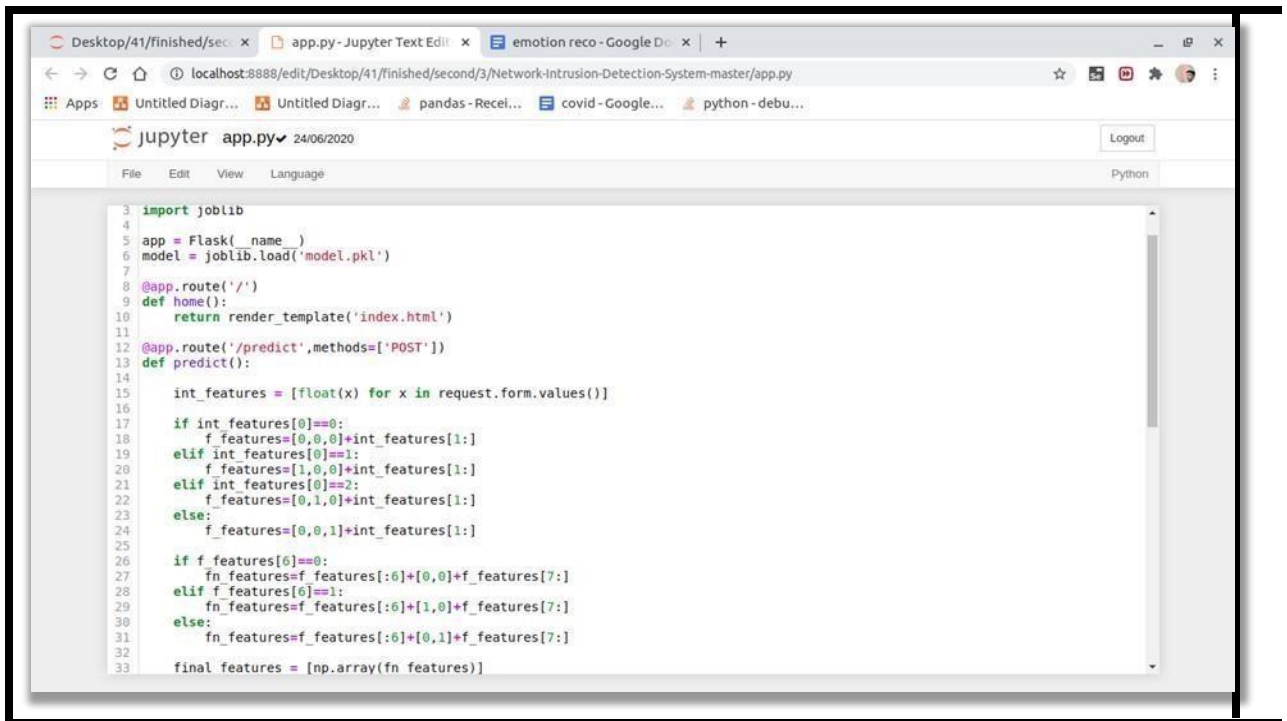
```python
from sklearn.model_selection import GridSearchCV
param_grid = {'C': [1, 10],
              'gamma': [0.0001, 0.001]}
grid = GridSearchCV(model, param_grid)

grid.fit(train_X,train_y)
```
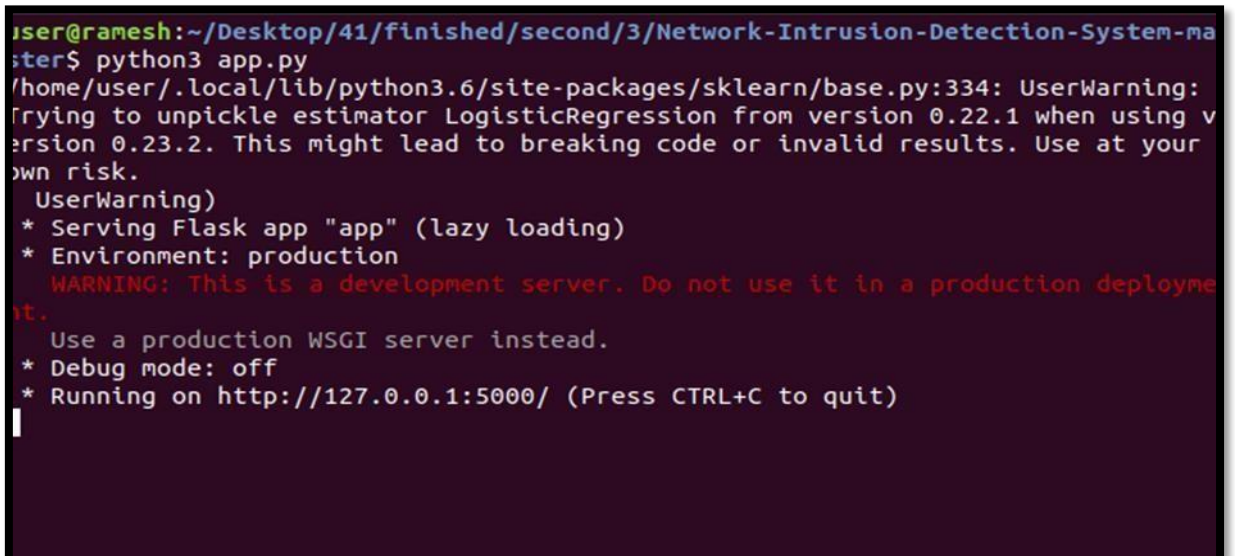
```python
print(grid.best_params_)
```
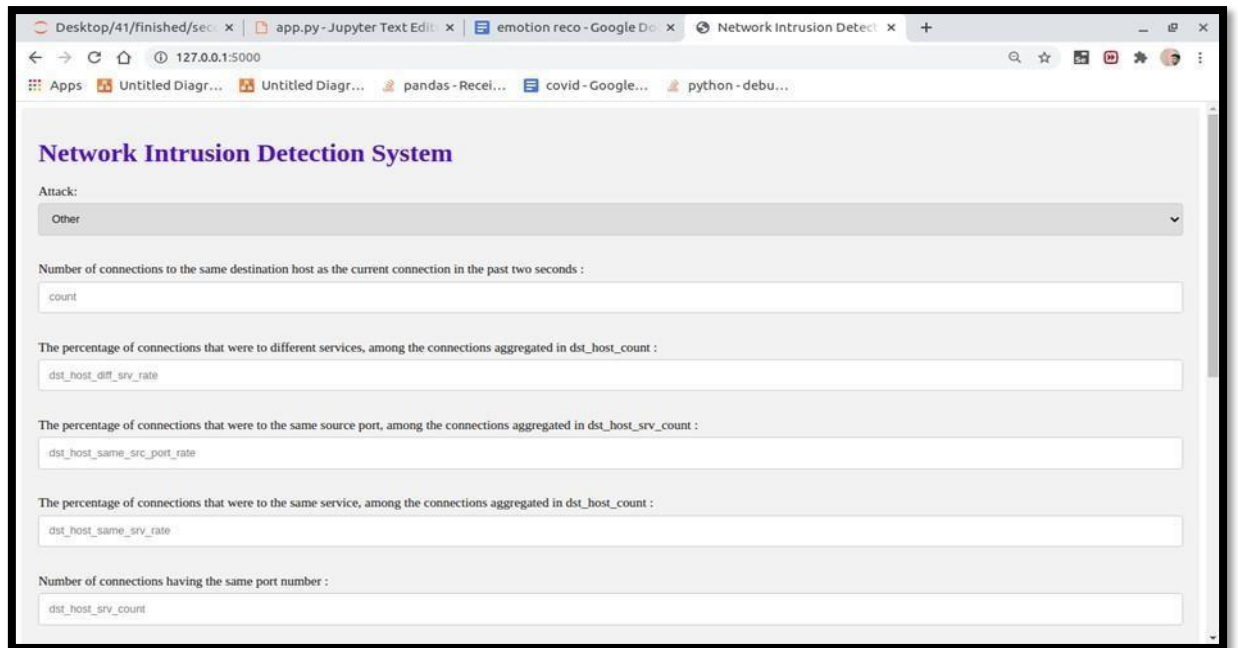
5.RESULTS

SCREEN SHOTS

Screenshot 5.1:Application of Network detection System.

Screenshot 5.2: Localhost in cmd python app.py

Screenshot 5.3: Network Intrusion Detection System.

Screenshot 5.4: Input values
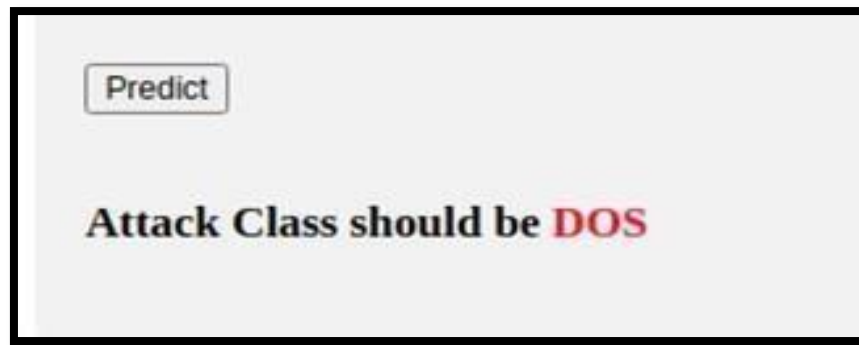
Screenshot 5.5: Attack has been predicted.

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 land 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 result

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 Input : identified classes of invalid input must 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.

6.3 TEST CASES

Experiments include an arrangement of steps, conditions and

Source of info that can be utilized while performing testing undertakings. The principal expectation of this action is to guarantee whether a product passes or bombs as far as usefulness and different perspectives. The way toward creating experiments can likewise help discover issues in the prerequisites or plan of an application. Experiment goes about as the beginning stage for the test execution, and in the wake of applying an arrangement of information esteems, the application has a conclusive result and leaves the framework at some end point or otherwise called execution post condition.

TEST CASES:

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	Pass	Invalid login details will not allowed here
		success he get his home page		
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

In conclusion, machine learning techniques have prevent to be invaluable tools in the realm of cyber attack detection. Their ability to analyze vast and complex datasets, identify patterns, and adapt to evolving threats make them a crucial component in safeguarding digital environments. From anomaly detection to deep learning models, these methods continue to enhance the cybersecurity landscape by offering proactive, efficient, and scalable solutions to mitigate the ever-present and evolving threat of cyber attacks. As technology advances and threats become increasingly sophisticated, the continued development and deployment of machine learning in cybersecurity are paramount to maintaining the security and integrity of digital ecosystems.

7.2 FUTURE SCOPE

The future scope for machine learning techniques in cyber attack detection is promising and expansive. As cyber threats continue to evolve in complexity and scale, machine learning algorithms will play a pivotal role in enhancing cybersecurity. Advancements in deep learning, natural language processing, and reinforcement learning will enable more accurate and adaptive threat detection. Moreover, the integration of AIdriven tools with real-time data analysis, threat intelligence, and proactive risk mitigation strategies will be critical in staying ahead of cyber adversaries. Collaborative efforts between industry, academia, and government will drive the development of innovative machine learning solutions, ensuring that organizations can effectively safeguard their digital assets and infrastructure in an increasingly interconnected and data-driven world.

8.BIBILOGRAPHY

8. BIBILOGRAPHY

8.1 REFRENCES

[1] K. Graves, Ceh: Official certified ethical hacker review guide: Exam 312‑ 50. John Wiley & Sons, 2007.

[2] R.Christopher, "Port scanning techniques and the defence against them," SANS Institute, 2001.

[3] Baykara, R. Das and I. Kara Bilgi in 1st International Symposium on Digital Forensics and Security (ISDFS13), 2013, pp. 231–239.

[4] S. Staniford, J. A. Hoagland, and J. M. Mc Alen "Practical automated detection of stealthy port scans," Journal of Computer Security, vol. 10, no. 1-2, pp. 105–136, 2002.

[5] S. Robertson, E. V. Siegel, M. Miller, and S. J. Stolfo, "Surveillance detection in high bandwidth environments," in DARPA Information Survivability.

[6] K. Ibrahimi and M. Odane, "Management of intrusion detection systems based-kdd99: Analysis with lda and pca.

[7] N. Moustafa and J. Slay, "The significant features of the unsw-nb15 and the kdd99 data sets for network intrusion detection systems," in Building Analysis.

[8] L. Sun, T. Anthony, H. Z. Xia, J. Chen, X. Huang, and Y. Zhang, "Detection and classification of malicious patterns in network traffic using benford's law," in Asia-Pacific Signal.

[9] S. M. Almansob and S. S. Lomte, "Addressing challenges for intrusion detection system using naive bayes and pca algorithm," in Convergence.

[10] M. M. A. Rabbani, "Combined analysis of support vector machine and principle Component analysis for ids," in IEEE International Conference.

8.2 GITHUB LINK

https://github.com/supriya0777//Machine Learning Techniques For Cyber Attacks Detection