

DAYANANDA SAGAR UNIVERSITY

KUDLU GATE, BANGALORE – 560068



**Bachelor of Technology
in
COMPUTER SCIENCE AND ENGINEERING**

Major Project Phase-II Report

CYBER SECURITY ANALYSIS USING MACHINE LEARNING

By

D Lakshminadh (ENG19CS0154)

M Maheswar (ENG19CS0178)

P Akhil Reddy (ENG19CS0216)

P Raj Sandeep (ENG19CS0224)

Team - 27

Under the supervision of

Dr.RANGARAJ

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY,
BANGALORE**

(2022-2023)



DAYANANDA SAGAR UNIVERSITY

School of Engineering
Department of Computer Science & Engineering

Kudlu Gate, Bangalore – 560068
Karnataka, India

CERTIFICATE

This is to certify that the Phase-II project work titled “**CYBER SECURITY ANALYSIS USING MACHINE LEARNING**” is carried out by **D Lakshminadh (ENG19CS0154)**, **M Maheswar (ENG19CS0178)**, **P Akhil Reddy (ENG19CS0216)**, **P Raj Sandeep (ENG19CS0224)** bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year **2022-2023**.

Dr. Rangaraj

Professor
Dept. of CS&E,
School of Engineering
Dayananda Sagar University

Date:

Dr. Girisha G S

Chairman CSE
School of Engineering
Dayananda Sagar University

Date:

**Dr. Udaya Kumar
Reddy K R**

Dean
School of Engineering
Dayananda Sagar
University

Date:

Name of the Examiner

Signature of Examiner

1.

2.

DECLARATION

We **D Lakshminadh (ENG19CS0154), M Maheswar (ENG19CS0178), P Akhil Reddy (ENG19CS0216), P Raj Sandeep (ENG19CS0224)** are students of eighth semester **B. Tech in Computer Science and Engineering**, at School of Engineering, **Dayananda Sagar University**, hereby declare that the Major Project Stage-II titled **“CYBER SECURITY ANALYSIS USING MACHINE LEARNING”** has been carried out by us and submitted in partial fulfillment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2022-2023**.

Student

Signature

Name : D Lakshminadh

USN : ENG19CS0154

Name : M Maheswar

USN: ENG19CS0178

Name : P Akhil Reddy

USN : ENG19CS0216

Name: P Raj Sandeep

USN : ENG19CS0224

Place : Bangalore

Date :

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

*We would like to thank **Dr. Udaya Kumar Reddy K R, Dean, School of Engineering & Technology, Dayananda Sagar University** for his constant encouragement and expert advice.*

*It is a matter of immense pleasure to express our sincere thanks to **Dr. Girisha G S, Department Chairman, Computer Science and Engineering, Dayananda Sagar University**, for providing right academic guidance that made our task possible.*

*We would like to thank our guide **Dr. Rangaraj, Professor, Dept. of Computer Science and Engineering, Dayananda Sagar University**, for sparing his/her valuable time to extend help in every step of our project work, which paved the way for smooth progress and fruitful culmination of the project.*

*We would like to thank our **Project Coordinator Dr. Meenakshi Malhotra and Dr. Pramod Kumar Naik** as well as all the staff members of Computer Science and Engineering for their support.*

We are also grateful to our family and friends who provided us with every requirement throughout the course.

We would like to thank one and all who directly or indirectly helped us in the Project work.

TABLE OF CONTENTS

	Page
LIST OF ABBREVIATIONS	vi
LIST OF FIGURES	vii
LIST OF TABLES	viii
ABSTRACT	ix
CHAPTER 1 INTRODUCTION.....	1
1.1. INTRODUCTION.....	2
1.1. OBJECTIVE.....	2
1.2. SCOPE.....	2
CHAPTER 2 PROBLEM DEFINITION	3
CHAPTER 3 LITERATURE SURVEY.....	5
CHAPTER 4 PROJECT DESCRIPTION.....	9
4.1. SYSTEM DESIGN	10
CHAPTER 5 REQUIREMENTS	12
5.1. FUNCTIONAL REQUIREMENTS	13
5.2. NON-FUNCTIONAL REQUIREMENTS	13
5.3. HARDWARE AND SOFTWARE REQUIREMENTS.....	14
CHAPTER 6 METHODOLOGY.....	15
CHAPTER 7 EXPERIMENTATION.....	17
CHAPTER 8 TESTING AND RESULTS	21
CHAPTER 9 CONCLUSION AND FUTURE WORK	
10.1. CONCLUSION.....	24
10.1. SCOPE FOR FUTUREWORK	24
CHAPTER 10	
REFERENCES... ..	26
APPENDIX A	27

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
GUI	Graphical User Interface
PHP	Preprocessor Hypertext
MySQL	My Structured Query Language

LIST OF FIGURES

Fig. No.	Description of the figure	Page No.
4.1(a)	System Design	10
4.1(b)	User Interaction	11
7(a)	Login Page	18
7(b)	LabelEncoder	18
7(c)	SVM	19
7(d)	Random Forest	19
7(e)	Random Integration	20
7(f)	Accuracy	20
8(a)	Uploading dataset	22
8(b)	Attacks found in the test data	22
8(c)	Attacks graph	23
8(d)	Comparison graph	23

ABSTRACT

The purpose of our project is to provide an analysis on cyber attacks based on the data of the organization. Now-a-days cyber attacks have become a major issue, a lot of user's and organizations' sensitive information is getting compromised and falling in the hands of unauthorized users . So, we are taking a KDD Cup 99 dataset and using machine learning technology to analyze that dataset using classification algorithms. We are providing an analysis on the KDD Cup 99 dataset and which classification algorithm is efficient and fast based on the accuracy values. We are providing the count of attacks in the test dataset.

CHAPTER 1

INTRODUCTION

CHAPTER 1 INTRODUCTION

1.1.INTRODUCTION

The main purpose of our project is to present the graph based on different algorithms based on the attacks occurring in the KDD dataset. The algorithm we are proposing shows better accuracy than the other existing machine learning algorithms. The KDD dataset contains two types of data i.e., labeled and unlabeled. We are using a labeled KDD dataset because it is made for the sole purpose of using for research. We are performing preprocessing to eliminate the duplicates and convert the raw data into clean data. We are reading the data in rows and columns . After preprocessing the data , we use fit transform to read the input and use transformation methods on the data. We are using tuples and converting the tuples into lists. We are applying mathematical conditions and formulas to find accuracy.. The accuracy is used to plot the graph. We are using the flask module to create interaction between the user and the system in the web browser.

1.2.OBJECTIVE

One of the most recent cyber attacks is NATO data leak involving leakage of sensitive NATO documents that are published and sold on the dark web.

Solution: Similarly here in our project we will be taking a dataset which contains some type of attacks and we analyze them and state which algorithm is the best performer.

1.3SCOPE

The scope of the project is to predict an attack before it happens based on the datasets using machine learning algorithms. We are using the KDD dataset to assess the performance of intrusion detection systems during the last ten years.

CHAPTER 2

PROBLEM DEFINITION

CHAPTER 2 PROBLEM DEFINITION

Lot of user sensitive information such as credit card, ATM pin number etc. got hacked and posted on public websites. There are different types of attacks in networking such as smurf, neptune, buffer overflow and normal attacks. A normal attack is unauthorized access by a third party. This attack aims to destroy or steal confidential information from a computer network. A smurf attack creates an internet traffic jam. In this attack, the attacker sends a request to the server using the host IP address, and collects confidential information by this process. The neptune attack is also known as half opened TCP SYN attack. To launch an attack, the attacker first exploits all flaws in three-way-handshake TCP protocol by sending large amounts of spoofed synchronization packing continuously to the TCP server. The aim of this attack is to reject any new connection from an authorized TCP client. In buffer overflow attack, attacker manipulates the coding error to carry out malicious actions and comprise the affected system. The attacker alters the application's execution path and overwrites elements of its memory, so that attackers can damage existing files or exposed data. So we have chosen this concept to explore new ways to completely analyze cyber security issues. Our system we are proposing will be based on a machine learning technique which will take the input as a dataset KDD Cup 99.

CHAPTER 3

LITERATURE REVIEW

CHAPTER 3 LITERATURE REVIEW

[1] A detailed analysis of the KDD CUP 99 dataset (2019), Mahbod Tavallaee, Ali A.Ghorbani, Wei Lu. anomaly detection has attracted the attention of many researchers to overcome the weakness of signature-based IDS in detecting novel attacks. At that time, KDD CUP 99 was the most widely used dataset for the evaluation of these systems. After analyzing KDD train and test sets, they found that about 78% and 75% of the records are duplicated in the train and test set respectively. This large amount of redundant records in the train set will cause learning algorithms to be biased towards the more frequent records, and thus prevent it from learning infrequent records which are usually harmful to networks. The test data have around 2 million connection records. KDD training dataset consists of approximately 4,900,000 single connection vectors each of which contains 41 features and they are labeled as normal or an attack. The simulated attacks are Denial of Service attack (DoS), User to Root Attack (U2R), Remote to Local Attack (R2L) and Probing attack. The analysis showed that there are two important issues in the data set which highly affects the performance of evaluated systems and results in a very poor evaluation of anomaly detection approaches. To solve these issues, the researchers have proposed a new data set which consists of selected records of the complete KDD dataset.

[2] Support Vector Machine, Batta Mahesh, 2019. The most widely used state-of-the-art machine learning technique is Support Vector Machine (SMV). In machine learning support machine learning comes under supervised machine learning which will analyze data for classification and regression analysis. In addition to performing linear classification, SMVs can efficiently perform a non-linear classification kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. It basically draws margins between the classes. The margins are drawn in such a way that the distance between the margin and classes is maximum and hence, minimizing the classification error. If the given is in single dimension then by using it can change it to 2D or 3D accordingly. Here the line is drawn between two points or classes based on their characteristics these are represented in a plain as points and their coordinates as their features if it is 2D then a hyper line is drawn and if it is 3D then a hyperplane is drawn and the distance between

hyperplane and nearest point is called margin , If the margin is maximum then SVM can perform with maximum accuracy. As it comes under supervised learning first it should be trained and based on the trained objects it will predict.

[3] Random Forest Algorithm, Leo Breiman Statistics Department University of California Berkeley 2020. Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. It takes less training time as compared to other algorithms. It predicts output with high accuracy, even for the large dataset it runs efficiently. It can also maintain accuracy when a large proportion of data is missing. Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase. The Working process can be explained in the following steps first step is to select a random k data points from the training set second step is to build the decision trees associated with select data points known as subsets third step is to choose the number of n for decision trees that you want to build fourth step is to repeat first and second step and last step is to find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes. There are mainly four sectors where Random forest mostly used are Banking, Medicine, Land use, Marketing.

[4] Big data preprocessing: methods and prospects, Salvador García*, Sergio Ramírez-Gallego, Julián Luengo, José Manuel Benítez and Francisco Herrera . The massive growth in the scale of data has been observed in recent years being a key factor of the Big Data scenario. Big Data can be defined as high volume, velocity and variety of data that require a new high-performance processing. Addressing big data is a challenging and time-demanding task that requires a large computational infrastructure to ensure successful data processing and analysis. The presence of data preprocessing methods for data mining in big data is reviewed in this paper. Vast amounts of raw data is surrounding us in our world, data that cannot be directly treated by humans or manual applications. Technologies such as the World Wide Web, engineering and science applications and networks, business services and many more generate data in exponential growth thanks to the development of powerful storage and connection tools. Nowadays, the current volume of data managed by our systems have surpassed the processing capacity of traditional systems, and this applies to data mining as well. The arising of new technologies and services (like Cloud computing) as well as the reduction in hardware price are leading to an ever-growing rate of information on the Internet.

CHAPTER 4

PROJECT DESCRIPTION

CHAPTER 4 PROJECT DESCRIPTION

Now-a-days cyber security has become a major issue in any field, due to it a lot of sensitive information is getting compromised and are used for unauthorized causes. So we have chosen this concept to explore new ways to deal with cyber security attacks. The purpose of our project is to detect an attack before it happens based on the dataset trained using machine learning algorithms. After providing a test data, the system performs preprocessing to eliminate the duplicates and convert the raw data into clean data. We are reading the data in rows and columns. After preprocessing the data, we use fit transform to read the input and use transformation methods on the data. We are using tuples and converting the tuples into lists. We are applying mathematical conditions and formulas to find the accuracy. We are using the KDD dataset because it has a large amount of redundant records which helps in knowing the different attacks.

4.1 System Design

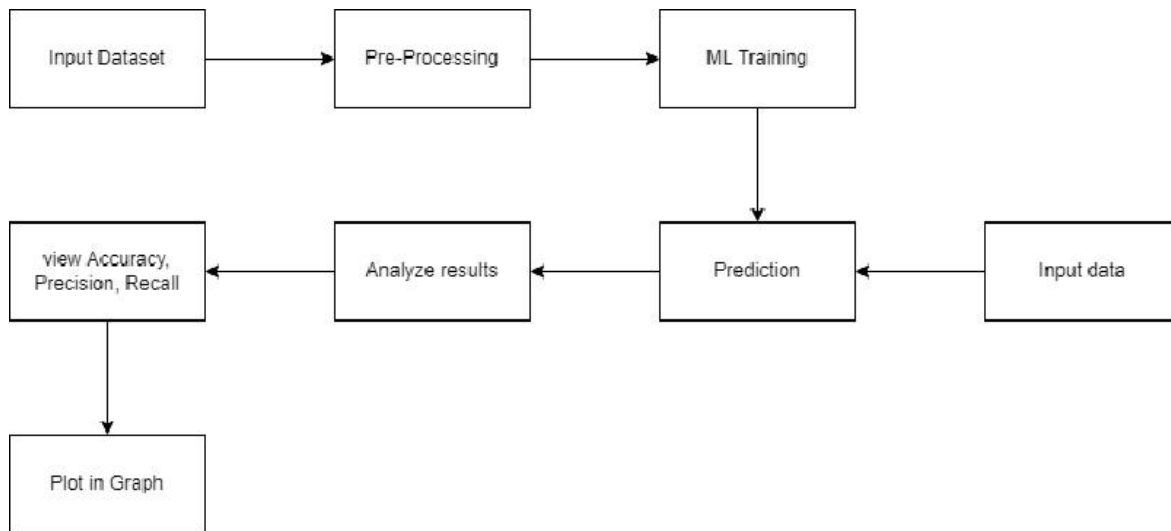


Fig. 4.1(a) System Design

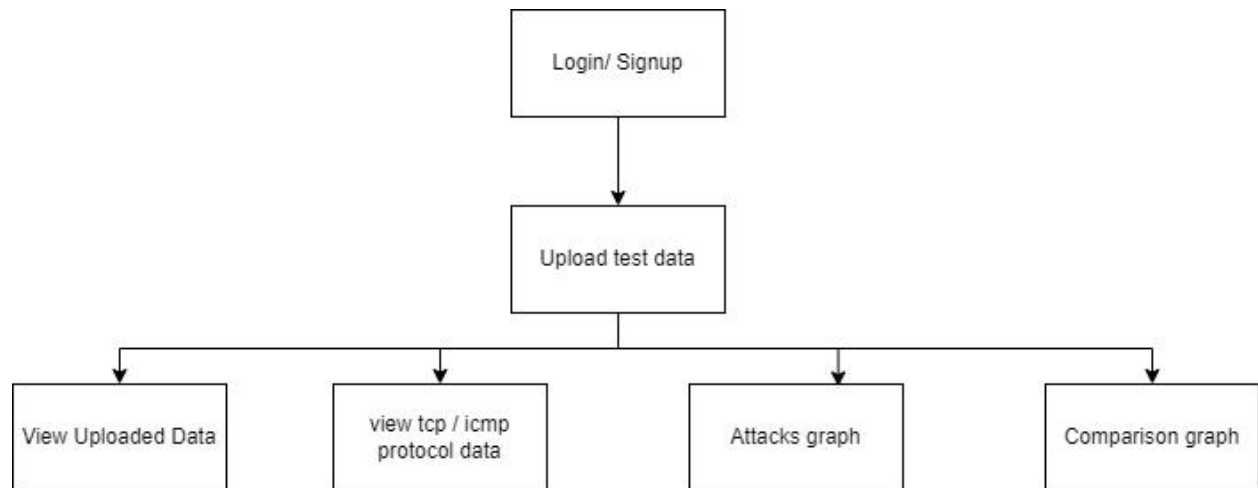


Fig.4.1(b) User Interaction

CHAPTER 5

REQUIREMENTS

CHAPTER 5 REQUIREMENTS

5.1 FUNCTIONAL REQUIREMENTS

We are going to provide an analysis on the cyber issues through our system. The result is shown in a designed framework. Our system which we are proposing will be based on machine learning techniques which will take the input as a dataset such as KDD Cup 99 or some other dataset which is helpful in finding the cyber security issues. The system recommends a dataset to start testing, we provide test data by copying a certain length of data from the KDD dataset. We create a new dataset for testing, once we upload the test data, the system runs the data by applying preprocessing and eliminating null values. The system contains features that differ with tcp or icmp data. The accuracy values are used to plot graphs of different classification algorithms. The attacks of test data are shown in a graph , the graph represents the count of the attacks in the dataset.

5.2 NON-FUNCTIONAL REQUIREMENTS

Portability: The application WAR file can be copied and can be run on any versions operating systems.

Reliability: The application is very reliable enough to manage the connections of end users.

Security

The application has several types of security enhanced on to the application which provides much more security.

5.3 SOFTWARE AND HARDWARE REQUIREMENTS

Hardware Requirements

RAM - minimum of 256 mb

Hard disk - 20 GB

Keyboard, Floppy drive

Software Requirements

Operating System : Windows 95/98/XP

Front end : Python

Libraries : Pandas, matplotlib, sklearn, flask

Platform: XAMPP

CHAPTER 6

METHODOLOGY

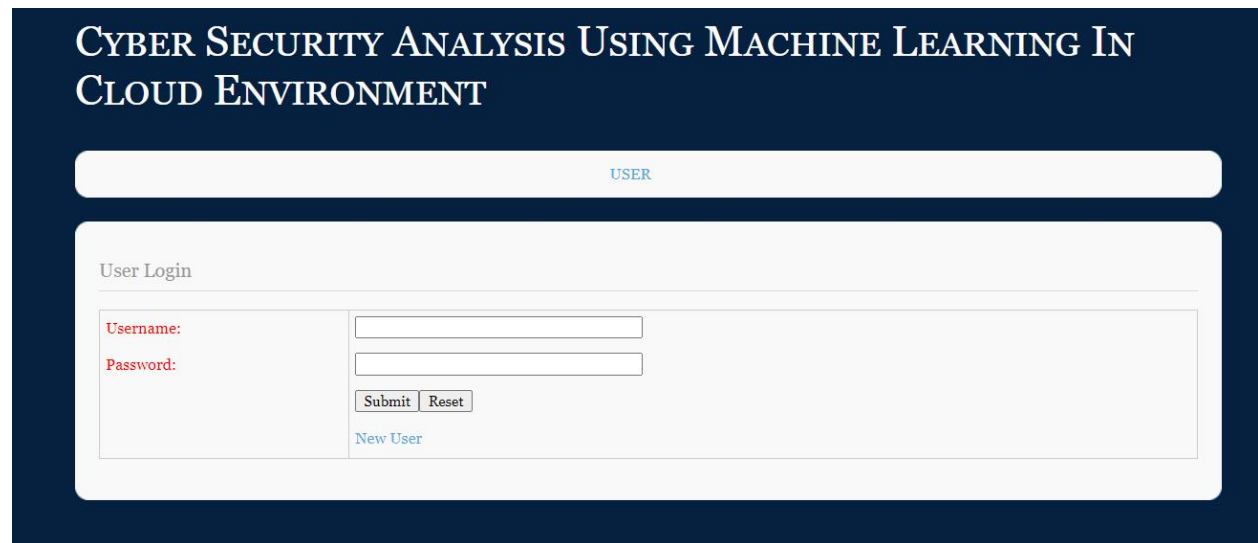
CHAPTER 6 METHODOLOGY

In our project, we train our model using KDD dataset and we dropped all the null, empty values from the dataset and then removed all duplicates using drop_duplicates function. After all these operations, we get a analyzed data. From analyzed data, we take a test data and provide it to system, We find accuracy for Support Vector Machine and random forest algorithms. We create a random integration algorithm based on random forest. Once we find the accuracy of all algorithms we plot graphs of attacks and comparison graphs of algorithms. These graphs are displayed in the web framework using HTML, CSS. We use the XAMPP platform for storing the dataset and analysis set. In future , we plan to add a few more algorithms to our system.

CHAPTER 7

EXPERIMENTATION

CHAPTER 7 EXPERIMENTATION



CYBER SECURITY ANALYSIS USING MACHINE LEARNING IN CLOUD ENVIRONMENT

USER

User Login

Username:

Password:

[New User](#)

Fig.7(a).Login Page

```
from sklearn.preprocessing import LabelEncoder

labelencoder_Y = LabelEncoder()

dataset.iloc[:, 1] = labelencoder_Y.fit_transform(dataset.iloc[:, 1].values)

print("Encoding : {}".format(labelencoder_Y.fit_transform(dataset.iloc[:, 1].values)))

dataset.iloc[:, 2] = labelencoder_Y.fit_transform(dataset.iloc[:, 2].values)

print("Encoding : {}".format(labelencoder_Y.fit_transform(dataset.iloc[:, 2].values)))

dataset.iloc[:, 3] = labelencoder_Y.fit_transform(dataset.iloc[:, 3].values)

print("Encoding : {}".format(labelencoder_Y.fit_transform(dataset.iloc[:, 3].values)))
```

Fig.7(b).LabelEncoder

```

#Import svm model
from sklearn import svm

#Create a svm Classifier
clf = svm.SVC(kernel='linear') # Linear Kernel

#Train the model using the training sets
clf.fit(X_train, y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)

```

Fig. 7(c) SVM

```

#Import scikit-learn metrics module for accuracy calculation
from sklearn.ensemble import RandomForestClassifier

#Create a Random Forest Classifier
classifier = RandomForestClassifier(n_estimators = 100)

#Train the model using the training sets
classifier.fit(X_train, y_train)

#Predict the response for test dataset
y_pred_1 = classifier.predict(X_test)

```

Fig.7(d) Random Forest

```

#Create Fusion Classifier
integration = objRandomIntegration.Random_Integration()

#Train the model using the training sets
integration.fit(X_train, y_train)

#Predict the response for test dataset
y_pred_2 = integration.predict(X_test)

# Model Accuracy, how often is the classifier correct?
confusion_matrix = pd.crosstab(y_test, y_pred_2)

result3 = err_metric(confusion_matrix)

print("Random Integration Accuracy:", result3)

```

Fig.7(e) Random Integration

```

def err_metric(CM):
    TN = CM.iloc[0, 0]
    FN = CM.iloc[1, 0]
    TP = CM.iloc[1, 1]
    FP = CM.iloc[0, 1]
    accuracy_model = (TP + TN) / (TP + TN + FP + FN)
    return accuracy_model

```

Fig.7(f) Accuracy

CHAPTER 8

TESTING AND RESULTS

CHAPTER 8 TESTING AND RESULTS

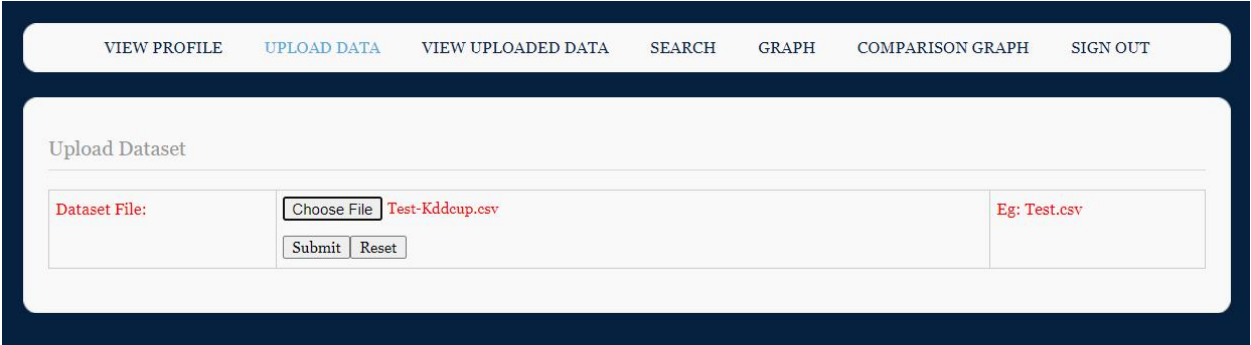


Fig.8(a)Uploading dataset

```
[[{'c': 2, 'Attack': 'buffer_overflow.'}, {'c': 2, 'Attack': 'loadmodule.'}, {'c': 2, 'Attack': 'neptune.'}, {'c': 5, 'Attack': 'normal.'}, {'c': 2, 'Attack': 'perl.'}, {'c': 1, 'Attack': 'smurf.'}]
2
2
2
5
2
1
```

Fig.8(b)Attacks found in the test data

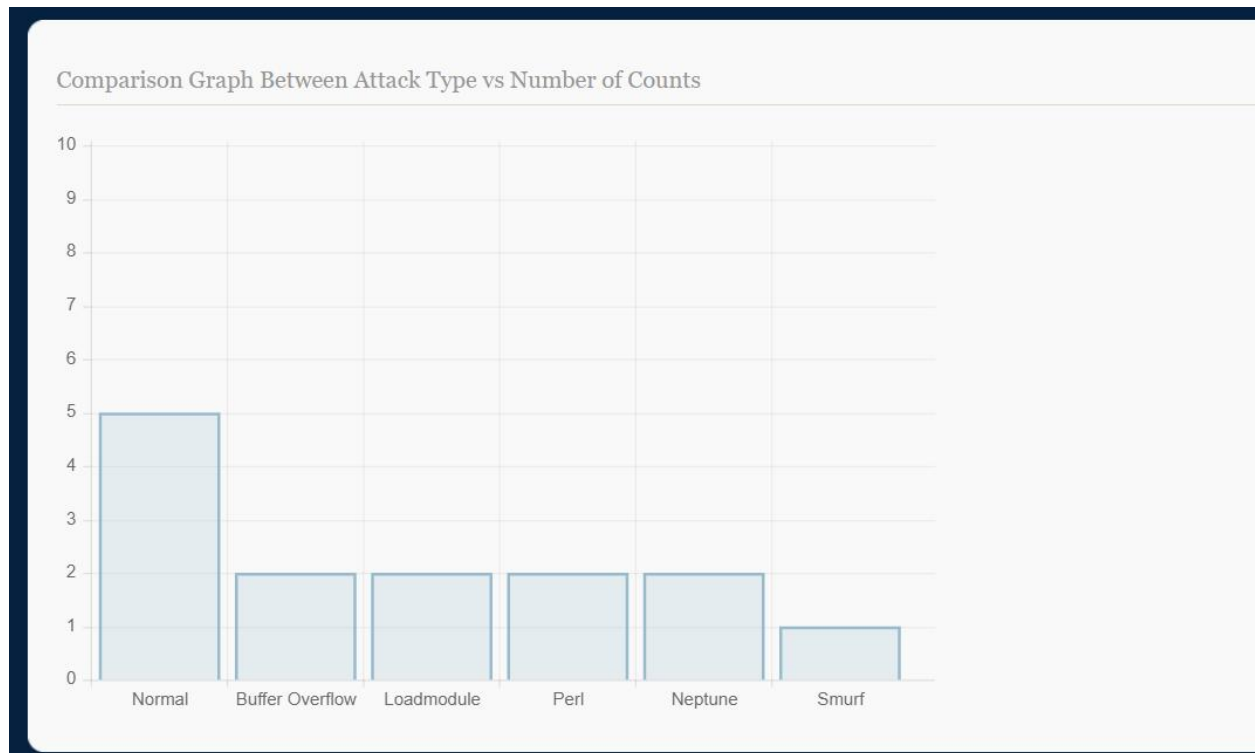


Fig.8(c).Attacks Graph

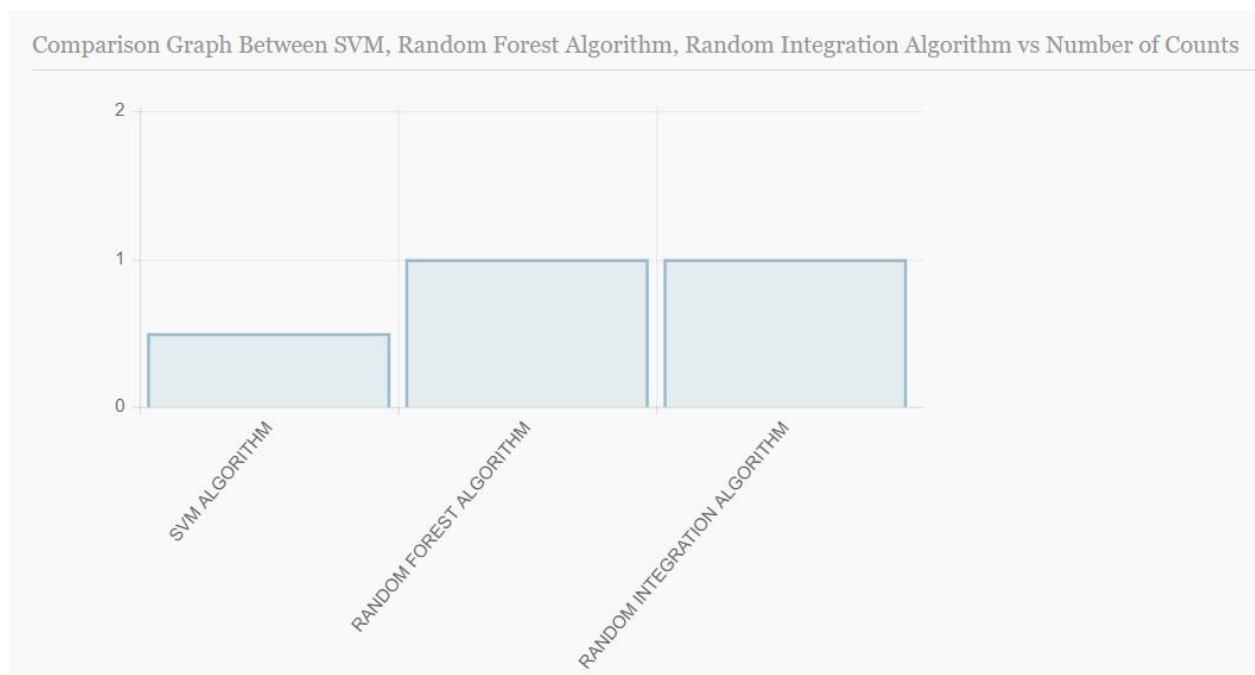


Fig.8(d)Comparison graph

CHAPTER 9

CONCLUSION

CHAPTER 9 CONCLUSION

CHAPTER 9.1 CONCLUSION

In our project, we analyzed the performance of different machine learning algorithms on the KDD Cup 99 data set for cyber security analysis. We used accuracy to plot comparison graph of different classification algorithms. We provide graph on the count of attacks on the dataset provided for testing.

CHAPTER 9.2 FUTURE WORK

In future , we plan to add few more algorithms and can be able to compare from a huge count of algorithms. The major extension of our project would be helping in analyzing few more number of datasets and different attacks.

REFERENCES

- [1] Mahbod Tavallaei, Ali A.Ghorbani, Wei Lu (2019). “A detailed analysis of the KDD CUP 99 dataset”, July 2019.(paper) from IEEE.org
- [2] Salvador García, Sergio Ramírez-Gallego, Julián Luengo, José Manuel Benítez, Francisco Herrera (2016). “Big data preprocessing: methods and prospects” , September 2016.(paper) from IEEE.org
- [3] Batta Mahesh.”Machine Learning algorithms”, January 2019.(paper) from IEEE.org
- [4] Leo Breiman. “Random Forest”, August 2020.(paper) from IEEE.org

APPENDIX

KDD Dataset:

Label	Field Name	Format
1	duration	number (default)
2	protocol_type	string (default)
3	service	string (default)
4	Flag	string (default)
5	src_bytes	number (default)
6	dst_bytes	any (default)
7	land	number (default)
8	wrong_fragment	number (default)
9	urgent	number (default)
10	hot	number (default)
11	num_failed_logins	number (default)
12	logged_in	number (default)
13	lnum_compromised	number (default)
14	lroot_shell	number (default)
15	lsu_attempted	number (default)
16	lnum_root	number (default)
17	lnum_file_creations	number (default)
18	lnum_shells	number (default)
19	lnum_access_files	number (default)
20	lnum_outbound_cmds	number (default)
21	is_host_login	number (default)
22	is_guest_login	number (default)
23	count	number (default)
24	srv_count	number (default)
25	error_rate	number (default)
26	srv_error_rate	number (default)
27	rerror_rate	number (default)
28	srv_rerror_rate	number (default)
29	same_srv_rate	number (default)
30	diff_srv_rate	number (default)
31	srv_diff_host_rate	number (default)
32	dst_host_count	number (default)
33	dst_host_srv_count	number (default)
34	dst_host_same_srv_rate	number (default)
35	dst_host_diff_srv_rate	number (default)
36	dst_host_same_src_port_rate	number (default)
37	dst_host_srv_diff_host_rate	number (default)
38	dst_host_error_rate	number (default)

39	dst_host_srv_serror_rate	number (default)
40	dst_host_rerror_rate	number (default)
41	dst_host_srv_rerror_rate	number (default)
42	label	String(default)