Prediction of Cyber Attacks Using Machine Learning Techniques

A project report submitted to

Rajiv Gandhi University of knowledge technologies

SRIKAKULAM

In partial fulfilment of the requirements for the

Award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

3rd year B. Tech 2nd semester

Yalla Sandhya (S170185)

Under the Esteemed Guidance of

Mr. Kumar Sekhar Sir

Rajiv Gandhi University of Knowledge Technologies – SKLM

CERTIFICATE

This is to certify that the thesis work titled "Prediction of Cyber-attacks by using the Machine Learning Techniques" was successfully completed by Yalla Sandhya(s170185). In partial fulfilment of the requirements for the Summer internship Project in Computer Science and Engineering of **Rajiv Gandhi University of Knowledge Technologies** under my guidance and output of the work carried out is satisfactory.

Mr. Kumar Sekhar sir

Project Guide

Mr.S.Sathish Kumar sir,Asst Prof(CSE)

Project Coordinator

DECLARATION

I declared that this thesis work titled "Prediction of Cyber-attacks by using the Machine Learning Techniques" is carried out by our team during the year 2021-22 in partial fulfillment of the requirements for the Summer Internship Project in **Computer Science and Engineering.**

I further declare that this dissertation and the matter embodied in this report has not been submitted elsewhere for any Degree. Furthermore, the technical details furnished in various chapters of this thesis are purely relevant to the above project and there is no deviation from the theoretical point of view for design, development and implementation.

Signed by:

YALLA SANDHYA (S170185)

ACKNOWLEDGEMENT

I would like to articulate my profound gratitude and indebtedness to my project guide Mr. Kumar Sekhar sir who has always been a constant motivation and guiding factor throughout the project time. It has been a great pleasure for me to get an opportunity to work under his guidance and complete the thesis work successfully.

We would like to thank Ms.M.Roopa madam(Asst Prof), Head of the department for her cooperation in completing our project.

I wish to extend my sincere thanks to Mr.S.Sathish Kumar sir(Asst Prof), Project Coordinator of Computer Science and Engineering Department, for his constant encouragement throughout the project.

I am also grateful to other members of the department without their support my work would have been carried out so successfully.

I thank one and all who have rendered help to me directly or indirectly in the completion of my thesis work.

Project Associate :

Yalla Sandhya(S170185),

ABSTRACT

With the increase in cyber data attacks, the manual method of investigating cyber-attacks is more prone to errors and is time consuming. With the increase in advanced cyber threat attacks with the same patterns, timely investigation is not possible. There are many systems proposed which analyse and predict threats using various machine learning methods, we applied machine learning algorithms to analyse and predict cyber-attacks.

KEYWORDS:

1) Most targeted Destination IP Address

2) Most Logical Ports attacked

3) Most Frequently/common type of Attack

4) Different time of the day, (odd, hours, day or night)

5) Find the Pattern

INDEX

Chapter-1

1. INTRODUCTION

1.1 Introduction

Cyber-attack, via cyberspace, targets an enterprise's use of cyberspace for the purpose of disrupting, disabling, destroying, or maliciously controlling a computing environment/infrastructure; or destroying the integrity of the data or stealing controlled information. The state of the cyberspace portends uncertainty for the future Internet and its accelerated number of users. New paradigms add more concerns with big data collected through device sensors divulging large amounts of information, which can be used for targeted attacks.

Though a plethora of extant approaches, models and algorithms have provided the basis for cyber-attack predictions, there is the need to consider new models and algorithms, which are based on data representations other than task-specific techniques. However, its non-linear information processing architecture can be adapted towards learning the different data representations of network traffic to classify type of network attack. In this, we are modeling cyber-attack prediction as a classification problem, Networking sectors have to predict the type of Network attack from given dataset using machine learning techniques. The analysis of dataset by supervised machine learning technique(SMLT) to capture several information's like, variable identification, uni-variate analysis, bi-variate and multi-variate analysis, missing value treatments etc. A comparative study between machine learning algorithms had been carried out in order to determine which algorithm is the most accurate in predicting the type Cyber Attacks. We classify four types of attacks are DOS Attack, R2L Attack, U2R Attack, Probe attack. The results show that the effectiveness of the proposed machine learning algorithm technique can be compared with best accuracy with entropy calculation, precision, Recall, F1 Score, Sensitivity, Specificity and Entropy.

The Machine learning approach taken by most organizations to counter cyber attacks is defensive and reactionary. Threats are only removed and analyzed once they are detected; at which point, the harm is done—the network has already been breached and valuable information compromised. Intrusion detection and prevention, such as anti-virus software and firewalls combined with access controls such as tracking the ports, Cyber-attacks are the common tools and measures employed by the majority of organizations.

1.2 Statement of the problem

1.3 OBJECTIVE

To build a model to recognize cyber attacks using the ipaddress and scipy.stats and datetime libraries and the Cyberattacks and TCP-ports datasets. CAP(Cyber Attacks Prediction) aims to recognize the underlying attacks. The primary objective of CAP is to improve man-machine interface. It can also be used to monitor the most targeted destination IP Address, Most Logical Ports attacked, Most Frequently/common type of Attack, Different time of the day(odd, hours, day or night), Find the Pattern.

1.4 GOALS

The goal is to identify correct prediction values so that the classifier can accurately predict unseen testing data. Cyber Attacks Prediction is a system through which various ports are useful for identifying IP address of different attacks.

1.5 SCOPE

- In the feature subsystems with further advancement and research can be used to detect severe mental illness like depression and others
- using the combination of ml approaches as this increase the accuracy and efficiency of the system

cyber prediction plays a crucial role in the area of Artificial intelligence and Internet of things. It offers tremendous scope to human computer interaction, robotics, health care, biometric security and behavioral modeling. Cyber Attacks Prediction recognize attacks from different, text data, IP movements signals. Along with the time of the day(hours, min and sec)control over attacks and power of activation of security can also be examined for analyzed. It identifies various supervised and unsupervised machine-learning techniques for feature extraction and attacks classification.

1.6 APPLICATIONS

There are many applications of detecting the Cyber Attacks Prediction like in the interface with

- *robots,*
- *audio surveillance,*
- *web-based E-learning,*
- *commercial applications,*
- *clinical studies,*
- *entertainment,*
- *banking,*
- *call centers,*
- *cardboard systems,*
- *computer games, etc*

For online transactions, information about the state of attack can provide focus on the enhancement of quality and data protection.

1.7 LIMITATIONS

- Each attack may correspond to the different portions of the cyber crimes. The same attack may show different types of crimes. Therefore it is very difficult to differentiate these portions of Cyber Attacks. Another problem is that Result of attack is depending on the victim and their culture and environment.
- Difficult to build a perfect system. Filtering the attack is a task and it is too much that can even be difficult for humans to accomplish.
- When crimes are more and invites data losses to others.
- Filtering Cyber Attacks is a task and it is too much that can even be difficult for humans to accomplish.
- Error and misinterpretation of IP address when there are using VPN.

Chapter 2

LITERATURE SURVEY

2.1 Collect Information

An IDS generally has to deal with problems such as large network traffic volumes, highly uneven data distribution, the difficulty to realize decision boundaries between normal and abnormal behaviour, and a requirement for continuous adaptation to a constantly changing environment. In general, the challenge is to efficiently capture and classify various behaviours in a computer network. Strategies for classification of network behaviours are typically divided into two categories: misuse detection and anomaly detection. Misuse detection techniques examine both network and system activity for known instances of misuse using signature matching algorithms. This technique is effective at detecting attacks that are already known. However, novel attacks are often missed giving rise to false negatives. Alerts may be generated by the IDS, but reaction to every alert wastes time and resources leading to instability of the system. To overcome this problem, IDS should not start elimination procedure as soon as the first symptom has been detected but rather it should be patient enough to collect alert stand decide based on the correlation of them. Some research statistics with regards to the impact of cyber security to businesses, organizations, and individuals include: In recent years, cybercrime has been responsible for more than $400 billion in funds stolen and costs to mitigate damages caused by crimes. It has been predicted that a shortage of over 1.8 million cybersecurity workers will be experienced by 2022.It's been predicted that organizations globally will spend at least $100 billion annually on cyber security protection. Attackers currently make over $1 billion in annual revenue such as crypto wall attacks from Ransomware attacks and cyber crimes.

2.2 Study

CAP key features:

- Input Cyber attacks and TCP-Ports dataset
- Feature extraction using SVM and RNN
- Classifier based on SVM
- Training, and testing
- Output finding Pattrens(types of attacks).

2.3 Benefits

Cyber Attacks Prediction benefits to many institutions and aspects of life. It is useful and important for security and healthcare purposes. Also, it is crucial for easy and simple detection of man made attacks, man in the middle attacks at a specific moment without actually asking them.

- Cybersecurity is critical because it **helps to protect organizations and individuals from cyber attacks.**
- Cybersecurity can help to prevent data breaches, identity theft, and other types of cybercrime. Organizations must have strong cybersecurity measures to protect their data and customers.
- Cyber Threat Intelligence identifies potential threats to the organization and details which threats need immediate attention.
- Identifying Attacks through Phone calls (telephone conversation between criminals would help crime investigation) asking for personal information via mobile applications and telling the offers can be analyzed.
- It is Useful for enhancing the IP addresses based on human machine interaction.
- Interactive movie, storytelling & E-tutoring applications would be more practical, if they can adapt themselves to listeners or Students emotional states.

Chapter -3

ANALYSIS

3.1 Existing system

Within the ever-growing and quickly increasing field of cyber security, it is nearly impossible to quantify or justify the explanations why cyber security has such an outsized impact. Permitting malicious threats to run any place, at any time or in any context is a long way from being acceptable, and may cause forceful injury. It particularly applies to the Byzantine web of consumers and using the net and company information that cyber security groups are finding i hard to shield and contain. Cyber security may be a necessary thought for people and families alike, also for businesses, governments, and academic establishments that operate inside the compass of the world network or net. With the facility of Machine Learning, we will advance the cyber security landscape. Today's high-tech infrastructure, that has network and cyber security systems, is gathering tremendous amounts of data and analytics on almost all the key aspects of mission-critical systems. Whereas people still give the key operational oversight and intelligent insights into today's infrastructure. Most intrusion detection systems are focused on the perimeter attack surface threats, starting with your firewall. That offers protection of your network's north south traffic, but what it doesn't take into account is the lateral spread (east-west) that many network threats today take advantage of as they infiltrate your organization's network and remain there unseen. We know this is true because research has shown that only 20% of discovered threats come from north south monitoring. When an IDS detects suspicious activity, the violation is typically reported to a security information and event management (SIEM)system where real threats are ultimately determined amid benign traffic abnormalities or other false alarms. However, the longer it takes to distinguish a threat, the more damage can be done. An IDS is immensely helpful for monitoring the network, but their usefulness all depends on what you do with the information that they give you. Because detection tools don't block or resolve potential issues, they are ineffective at adding a layer of security unless you have the right personnel and policy to administer them and act on any threats. An IDS cannot see into encrypted packets, so intruders can use them to slip into the network. An IDS will not register these intrusions until they are deeper into the network, which leaves your systems vulnerable until the intrusion is discovered. This is a huge concern as encryption is becoming more prevalent to keep our data secure. One significant issue with an IDS is that they regularly alert you to false positives. In many cases false positives are more frequent than actual threats. An IDS can be tuned to reduce the number of false positives; however, your engineers will still have to spend time responding to them. If they don't take care to monitor the false positives, real attacks can slip through or be ignored.

Disadvantages :

➡Validation of dataset is a challenge in order to have accurate attack recognition. The system is very slow as to compare the correlations of the complete dataset.

➡ Variable length attacks files are not understandable. Finding or detecting attack is haystack.

➡It is a challenge to make attacks available. Long pre-processing steps are required for the model to understand the IP signals.

➡Performance and results of the attack system depends on accuracy of the dataset recognition algorithm used and so on. Highly accurate system will be expensive due to use of costly components.

3.3 Proposed System

Machine Learning algorithms can be used to train and detect if there has been a cyber attack. As soon as the attack is detected, an email notification can be sent to the security engineers or users. Any classification algorithm can be used to categorize if it is a DoS/DDoS attack or not. One example of a classification algorithm is Support Vector Machine (SVM) which is a supervised learning method that analyses data and recognizes patterns. Since we cannot control when, where or how an attack may come our way, and absolute prevention against these cannot be guaranteed yet, our best shot for now is early detection which will help mitigate the risk of irreparable damage such incidents can cause. Organizations can use existing solutions or build their own to detect cyber attacks at a very early stage to minimize the impact. Any system that requires minimal human intervention would be ideal.

ADVANTAGES OF PROPOSED SYSTEM:

- Can be implemented in any hardware supporting the python language.
- Very fast in processing the data and easy to use.
- Variable length datasets files are understood by the system.

SYSTEM SPECIFICATIONS:

Hardware:

Processor: AMD A9-9420 RADEON R5, 5 COMPUTER CORES
2C+3G 3.00GHz
Hard disk: 60GB
RAM : 4GB

Software:

Operating system : WINDOWS 10 pro
Programming language: Python

CHAPTER- 4

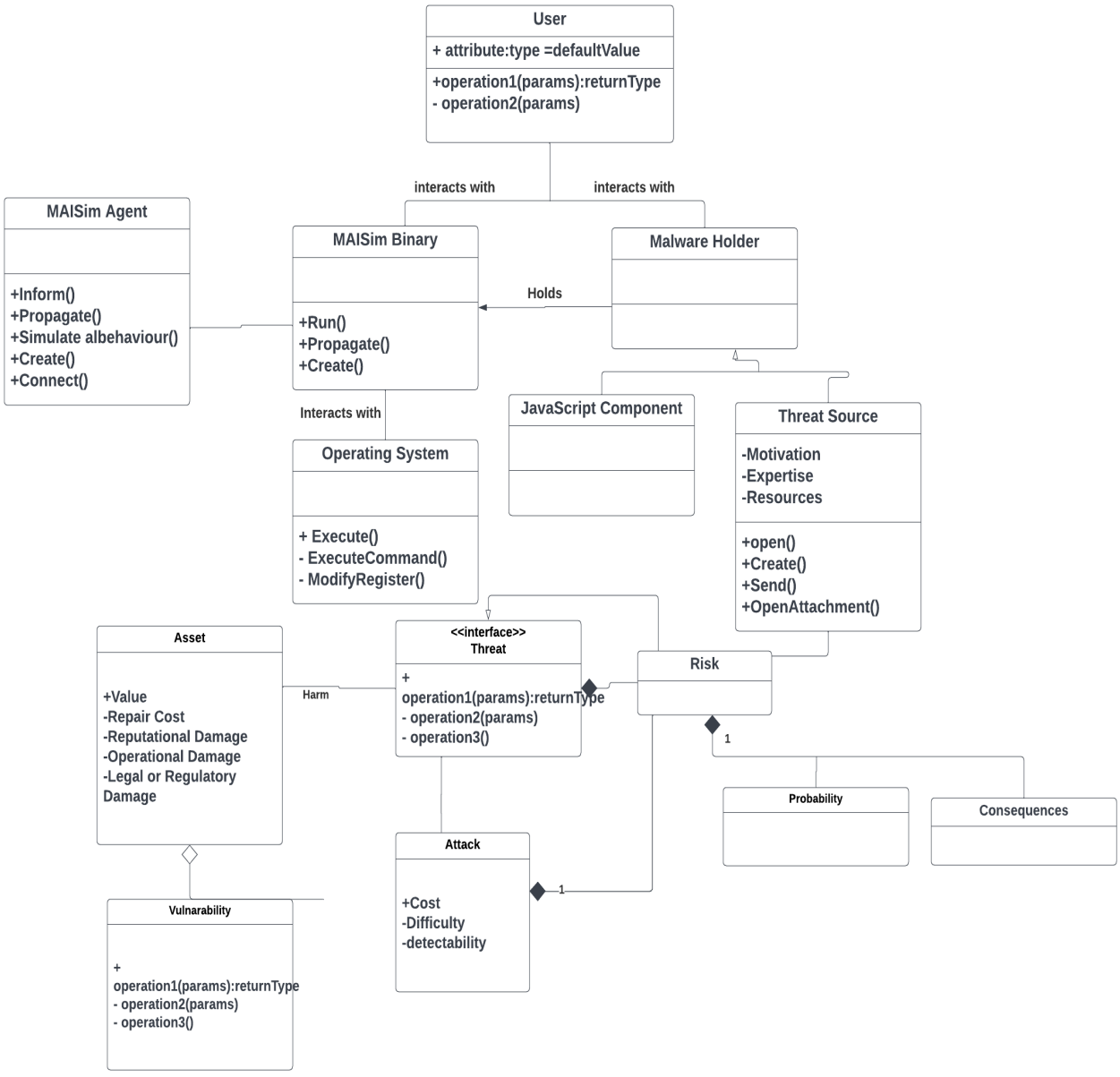
SYSTEM DESIGN

4.1 DESIGN OF THE SYSTEM

Cyber attacks prediction is a complex solution including a dataset of attack samples in a form of short csv records and the tool evaluating database samples by using subjective methods. In order to create the database of attacks samples for learning and training of classifier, it was necessary to extract short recorded files. In the second step, all records in cyber attacks database were evaluated using our designed evaluation tool and results were automatically evaluated how they are credible and reliable and how they represent different states of ipaddress.

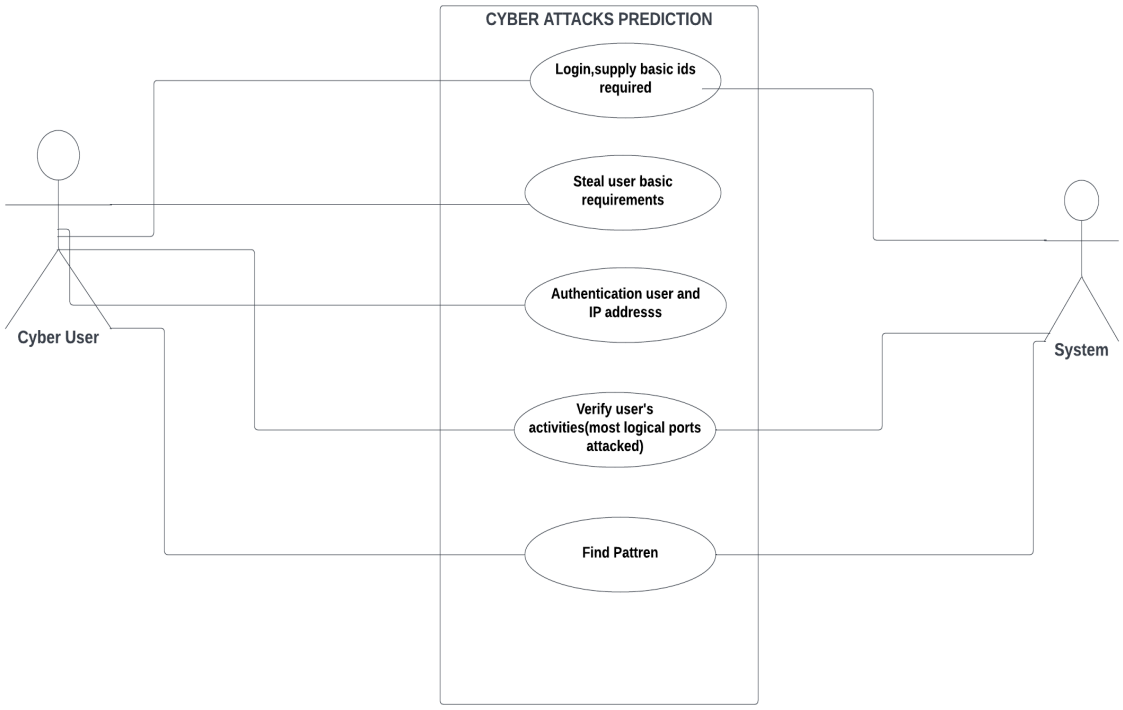
4.1.1 Class diagram:

Class diagram in the Unified Modelling Language (UML), is a kind of static structure diagram hat describes the constitution of a process through showing the system's classes, their attributes, and the relationships between the class. The motive of a class diagram is to depict the classes within a model. In an object-oriented software, classes have attributes (member variables), operations (member capabilities) and relation.



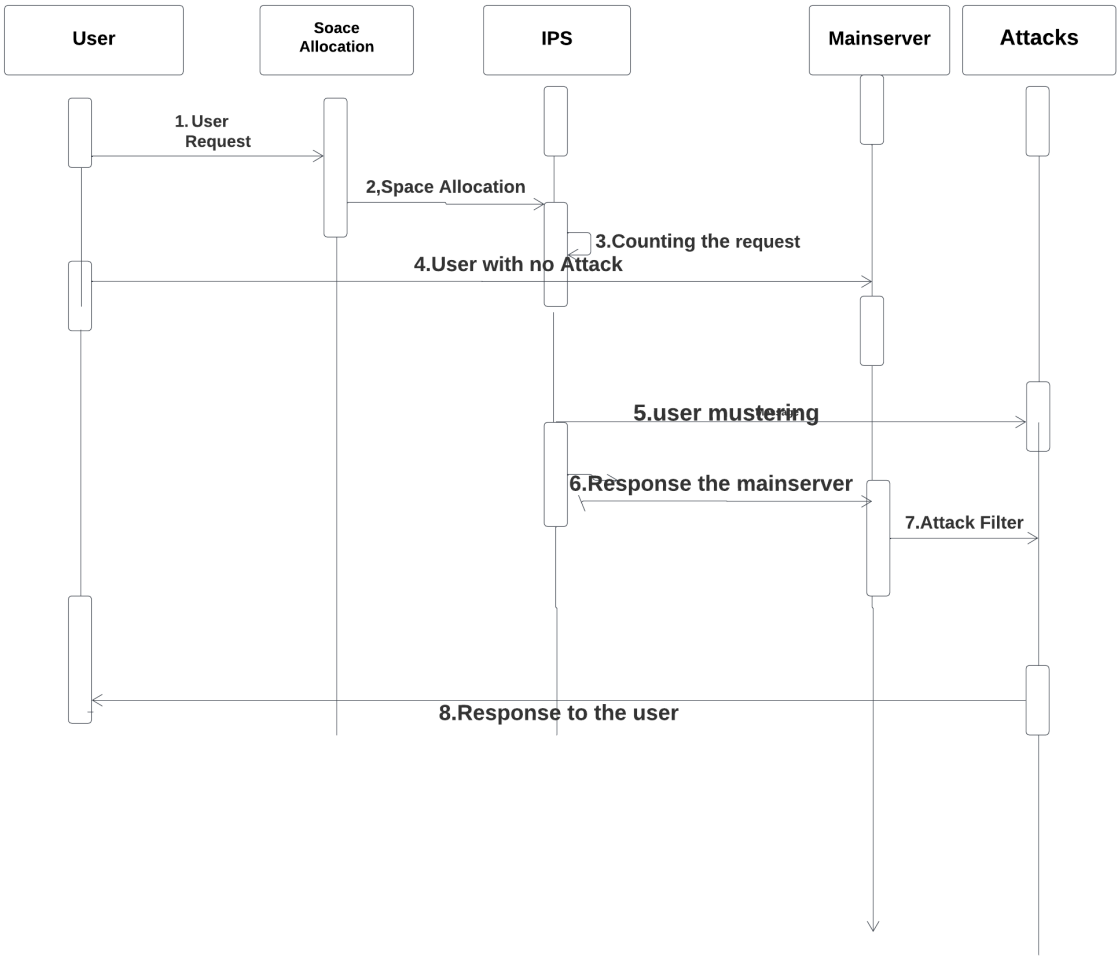
4.1.2 UseCase Diagram:

It is a visually representation what happens when actor interacts with system. A use case diagram captures the functional aspects of a system. The system is shown as a rectangle with name of the system inside, the actor are shown as stick figures, the use case are shown as solid bordered ovals labeled with name of the use case and relationships are lines or arrows between actor and use cases. Symbols used in Use case are as follows-



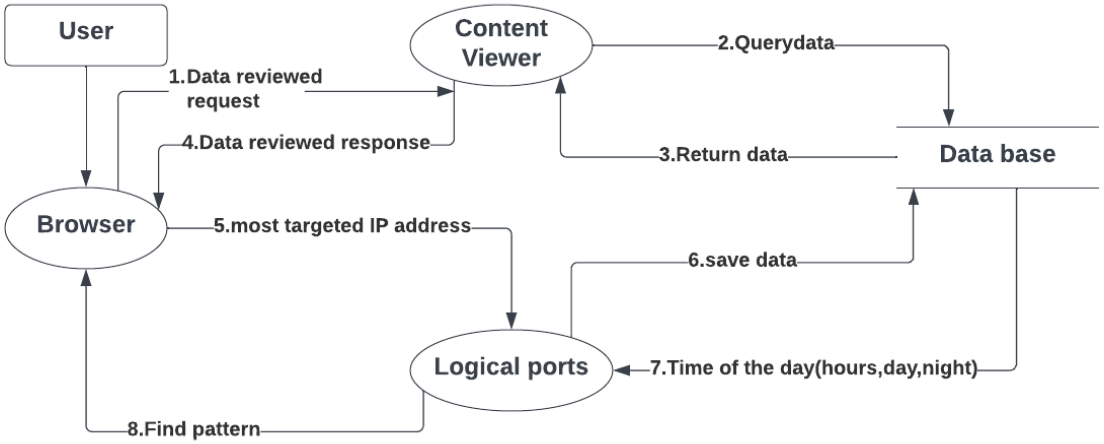
4.1.3 SEQUENCE DIAGRAM

A sequence diagram in Unified Modelling Language (UML) is one variety of interaction diagram that suggests how methods operate with one other and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are quite often referred to as event-hint diagrams, event situations, and timing diagrams. A sequence diagram suggests, as parallel vertical traces (lifelines), special systems or objects that are residing at the same time, and, as horizontal arrows, the messages exchanged between them, within the order the place they occur.

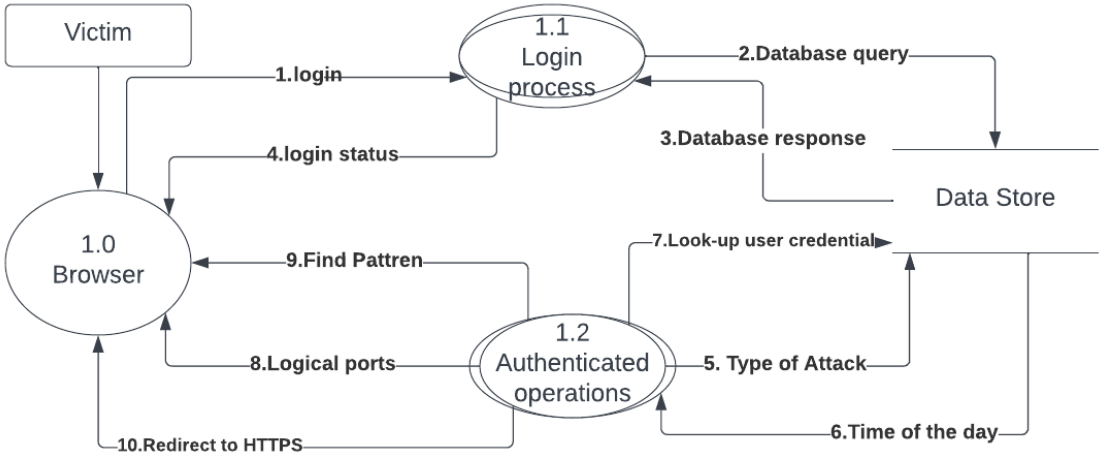


4.1.4 DFD Diagram:

A data flow diagram or bubble chart (DFD) is a graphical representation of the "flow" of data. through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).



DFD- 0 Level



DFD- Level 1

CHAPTER -5

SYSTEM IMPLEMENTATION

1.1 Prediction of Cyber attacks by using the machine learning

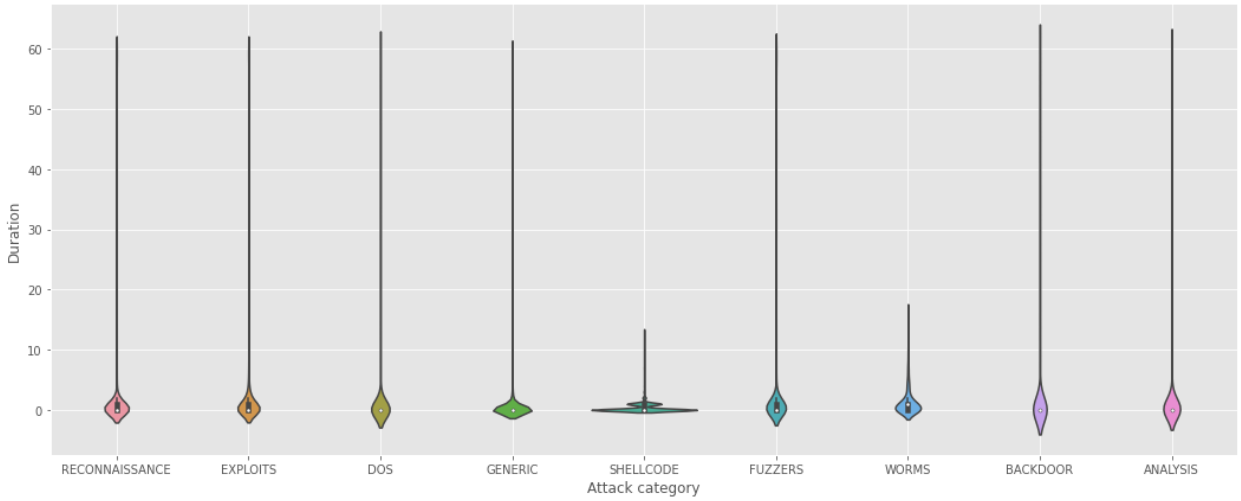
It is done by using Python programming language and run on the environment GOOGLE COLLABS notebook platform, we use the necessary libraries to implement this. To develop this Project we use Python programming language and to implement this project we use the models like SVM, CNN, RNN and and also collecting different types of dataset Cybersecurity attacks.

1.1.1 Output

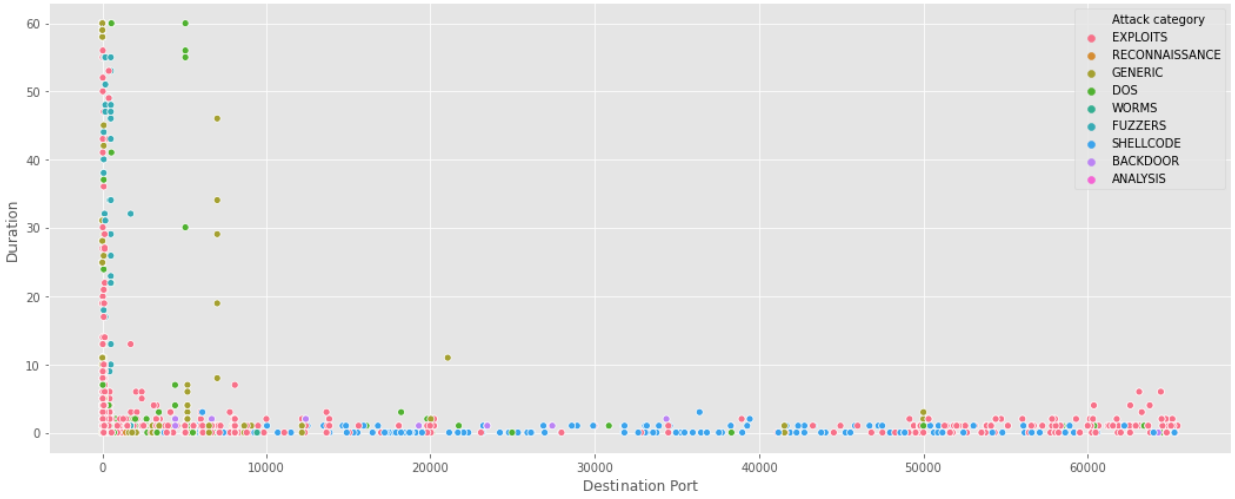
	Attack category	Attack subcategory	Protocol	Source IP	Source Port	Destination IP	Destination Port	Attack Name	Attack Reference	Start time	Last time	Destination Port Service	Duratio
0	RECONNAISSANCE	HTTP	TCP	175.45.176.0	13284	149.171.126.16	80	Domino Web Server Database Access: /doladmin.n...	-	2015-01-22 11:50:14	2015-01-22 11:50:16	HTTP	
1	EXPLOITS	Unix 'r' Service	UDP	175.45.176.3	21223	149.171.126.18	32780	Solaris rwalld Format String Vulnerability (ht...	CVE 2002-0573 (http://cve.mitre.org/cgi-bin/cv...	2015-01-22 11:50:15	2015-01-22 11:50:15	NaN	
2	EXPLOITS	Browser	TCP	175.45.176.2	23357	149.171.126.16	80	Windows Metafile (WMF) SetAbortProc() Code Exe...	CVE 2005-4560 (http://cve.mitre.org/cgi-bin/cv...	2015-01-22 11:50:16	2015-01-22 11:50:16	HTTP	
3	EXPLOITS	Miscellaneous Batch	TCP	175.45.176.2	13792	149.171.126.16	5555	HP Data Protector Backup (https://strikecenter...	CVE 2011-1729 (http://cve.mitre.org/cgi-bin/cv...	2015-01-22 11:50:17	2015-01-22 11:50:17	PERSONAL-AGENT	
4	EXPLOITS	Cisco IOS	TCP	175.45.176.2	26939	149.171.126.10	80	Cisco IOS HTTP Authentication Bypass Level 64 ...	CVE 2001-0537 (http://cve.mitre.org/cgi-bin/cv...	2015-01-22 11:50:18	2015-01-22 11:50:18	HTTP	

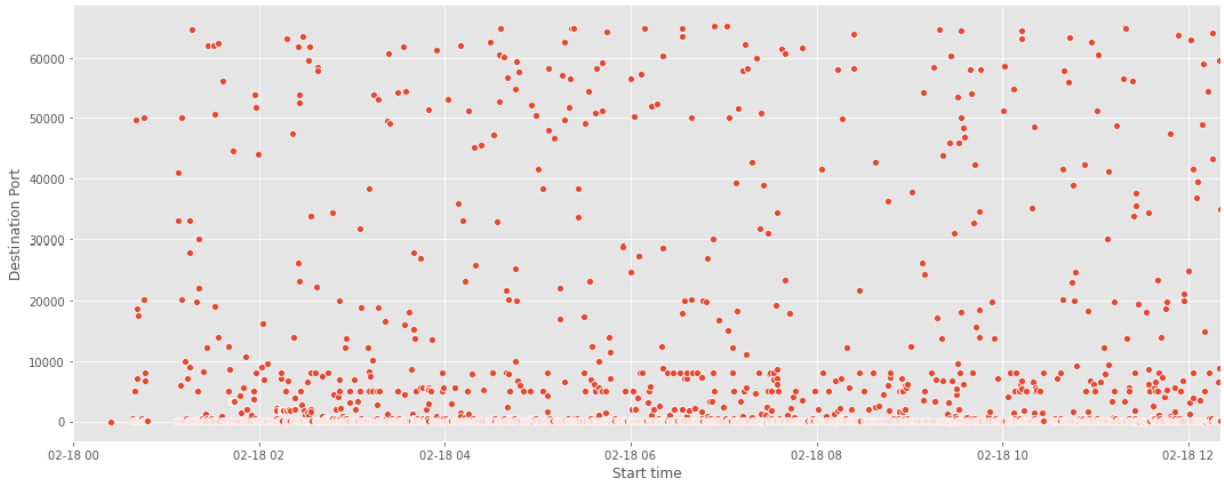
Speech Emotion Recognition

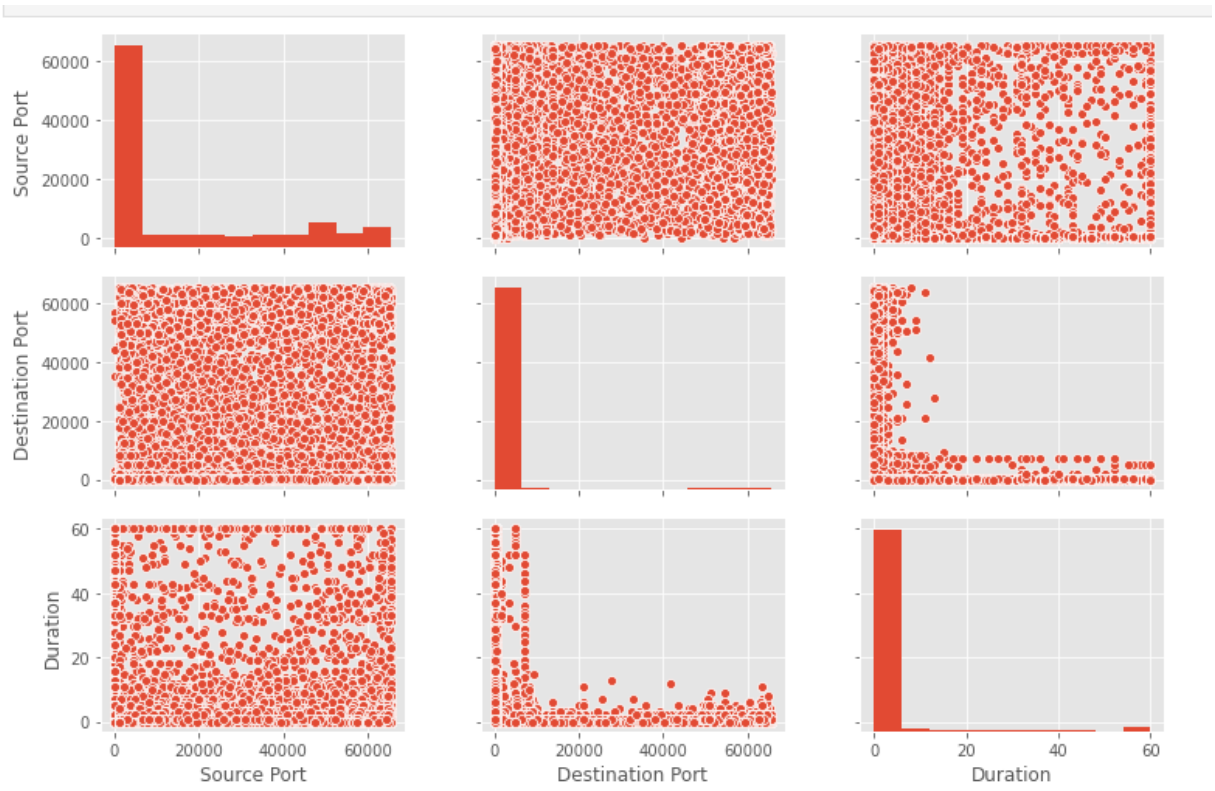
	Attack category	Attack subcategory	Protocol	Source IP	Source Port	Destination IP	Destination Port	Attack Name	Attack Reference	
174347	Generic	IXIA	udp	175.45.176.1	67520	149.171.126.18	53	Microsoft_DNS_Server_ANY_Query_Cache_Weakness_...	CVE 2005-4092 (http://cve.mitre.org/cgi-bin/cv...	14
174348	Exploits	Browser	tcp	175.45.176.3	78573	149.171.126.18	110	Microsoft Internet Explorer 6.0 Png pngfilt.dl...	BPS 2010-0002 (https://strikecenter.bpointsys....	14
174349	Reconnaissance	HTTP	tcp	175.45.176.1	71804	149.171.126.10	80	Domino Web Server Database Access: /internet.n...	NaN	14
174350	DoS	Ethernet	pnni	175.45.176.3	0	149.171.126.19	-753	Cisco IPS Jumbo Frame System Crash (https://st...	CVE 2008-2060 (http://cve.mitre.org/cgi-bin/cv...	14
174351	Fuzzers	OSPF	trunk-1	175.45.176.0	73338	149.171.126.13	0	Fuzzer: OSPF Hello Packet: Long Neighbor Lists...	NaN	14
...	...	...	...	...	...	...	...	...	...	
178026	Generic	IXIA	udp	175.45.176.0	72349	149.171.126.12	53	Microsoft_DNS_Server_ANY_Query_Cache_Weakness_...	CVE 2009-0234 (http://cve.mitre.org/cgi-bin/cv...	14
178027	Exploits	Browser	sep	175.45.176.3	67647	149.171.126.18	0	Persits XUpload ActiveX Method MakeHttpRequest...	CVE 2009-3693 (http://cve.mitre.org/cgi-bin/cv...	14
178028	Exploits	Office Document	tcp	175.45.176.0	78359	149.171.126.13	110	Microsoft Excel SxView Memory Corruption (POP3...	CVE 2009-3128 (http://cve.mitre.org/cgi-bin/cv...	14
178029	Exploits	Browser	tcp	175.45.176.2	68488	149.171.126.19	80	Internet Explorer createTextRange() Code Execu...	CVE 2006-1359 (http://cve.mitre.org/cgi-bin/cv...	14
178030	Reconnaissance	ICMP	unas	175.45.176.3	77929	149.171.126.19	0	IP Options: Loose Source Route (IP Option 3) (...	NaN	14

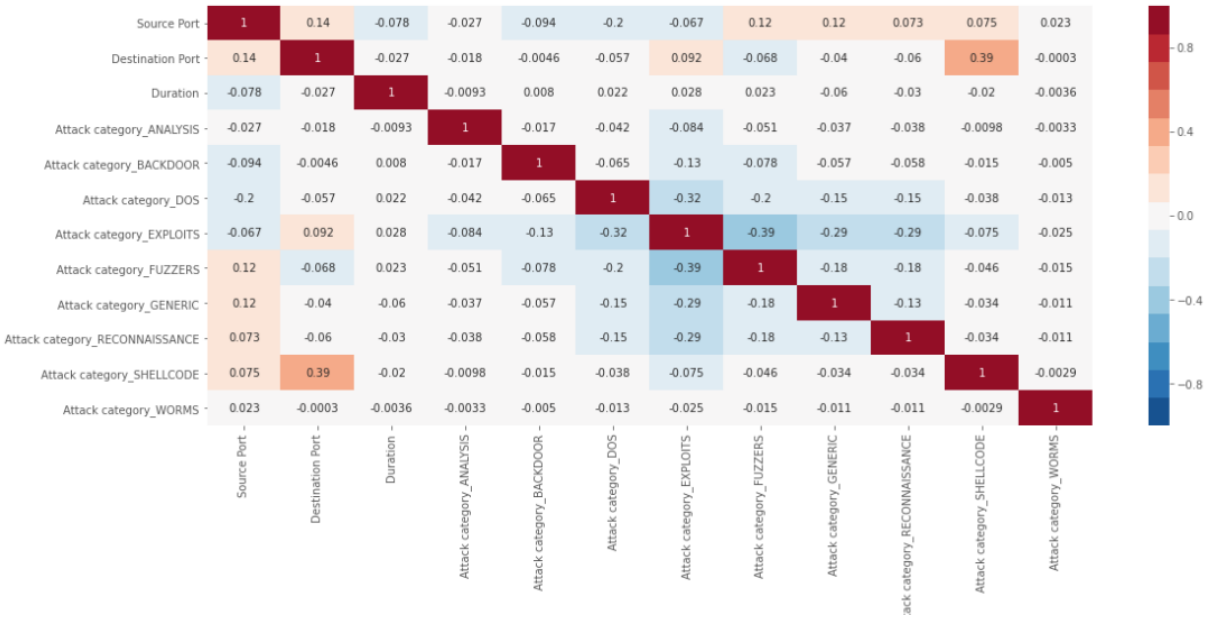


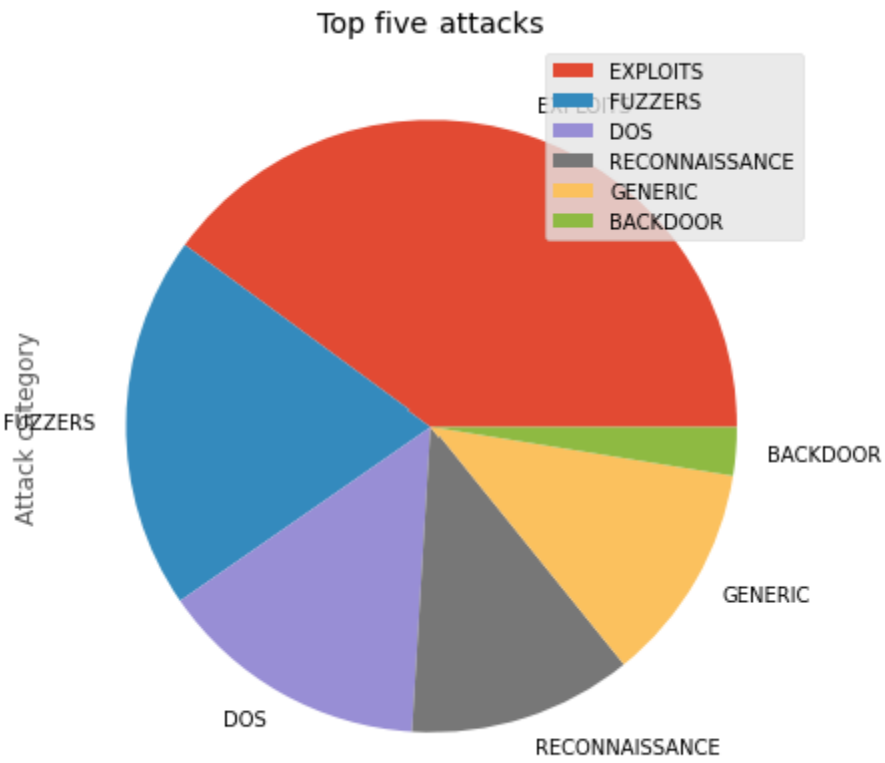
Speech Emotion Recognition



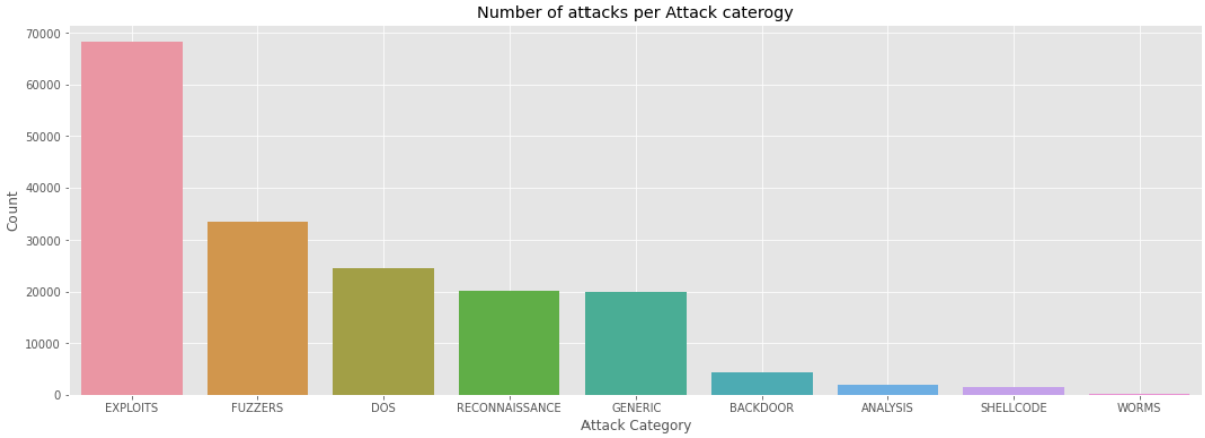



Source Port
Destination Port
Duration
0
20000
40000
60000
0
20
40
60


Source Port
Destination Port
Attack category_ANALYSIS
Attack category_BACKDOOR
Attack category_DOS
Attack category_EXPLOITS
Attack category_FUZZERS
Attack category_GENERIC
Attack category_RECONNAISSANCE
Attack category_SHELLCODE
Attack category_WORMS
0.8
0.4
0.0
−0.4
−0.8

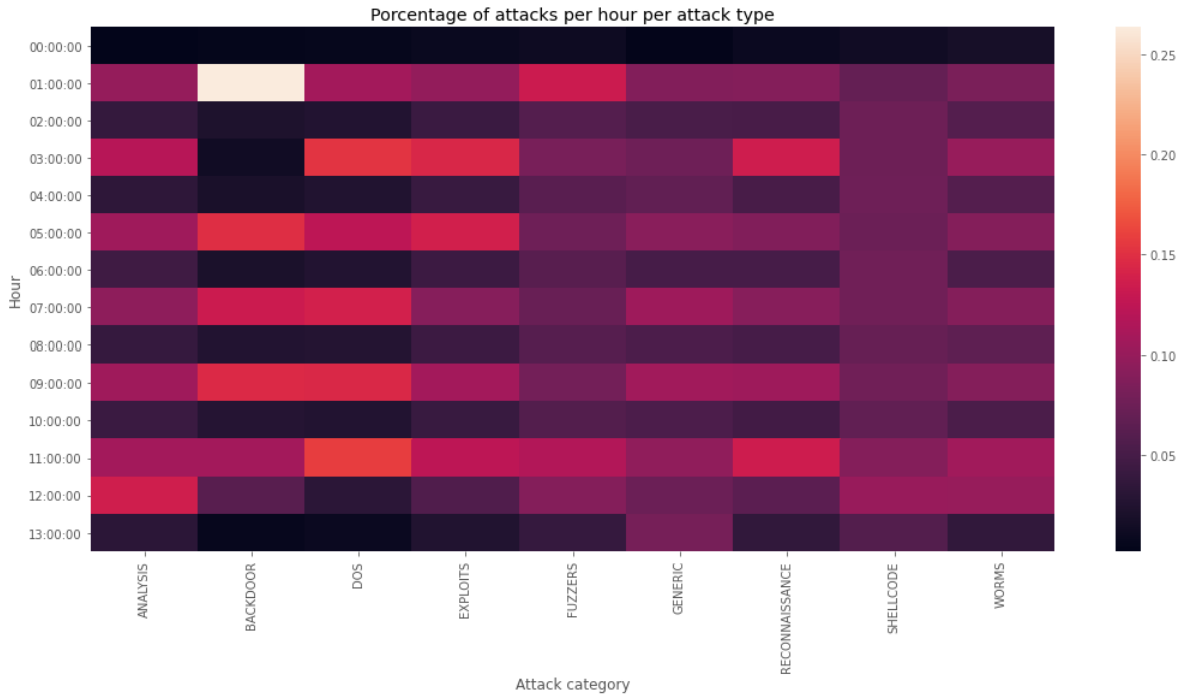

Top five attacks
EXPLOITS
FUZZERS
DOS
RECONNAISSANCE
GENERIC
BACKDOOR
Attack category
FUZZERS
BACKDOOR
GENERIC
DOS
RECONNAISSANCE

Speech Emotion Recognition

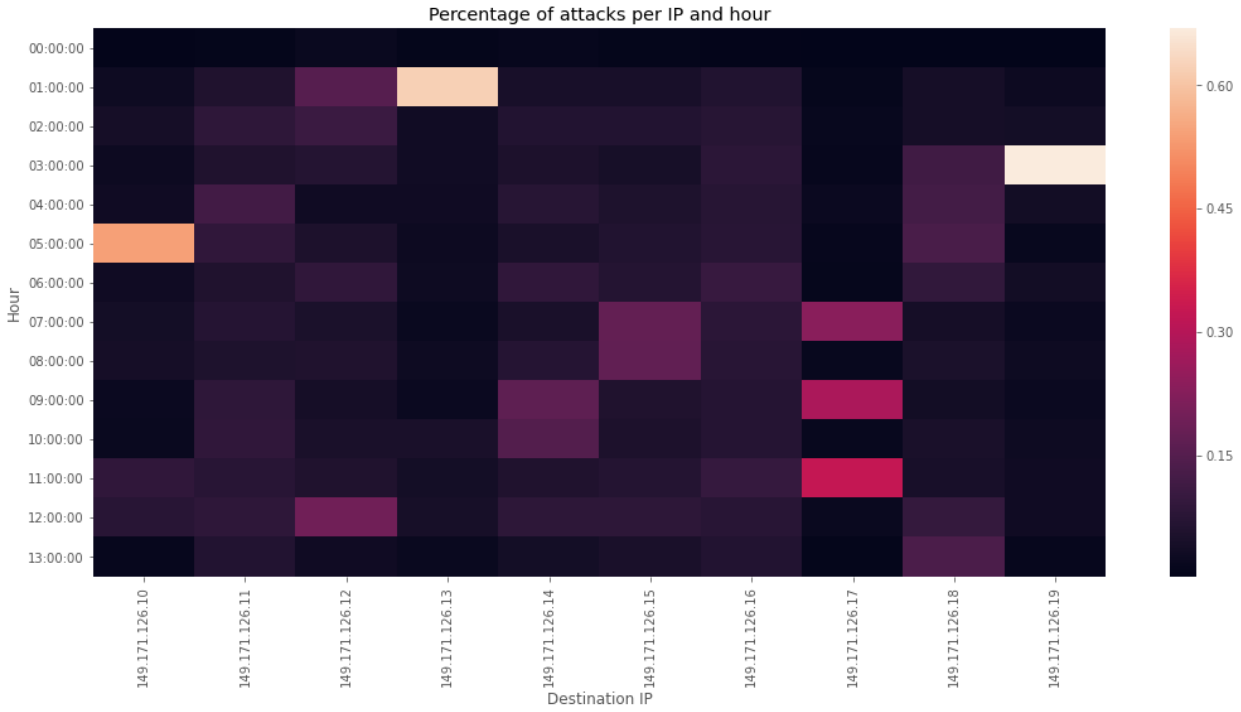


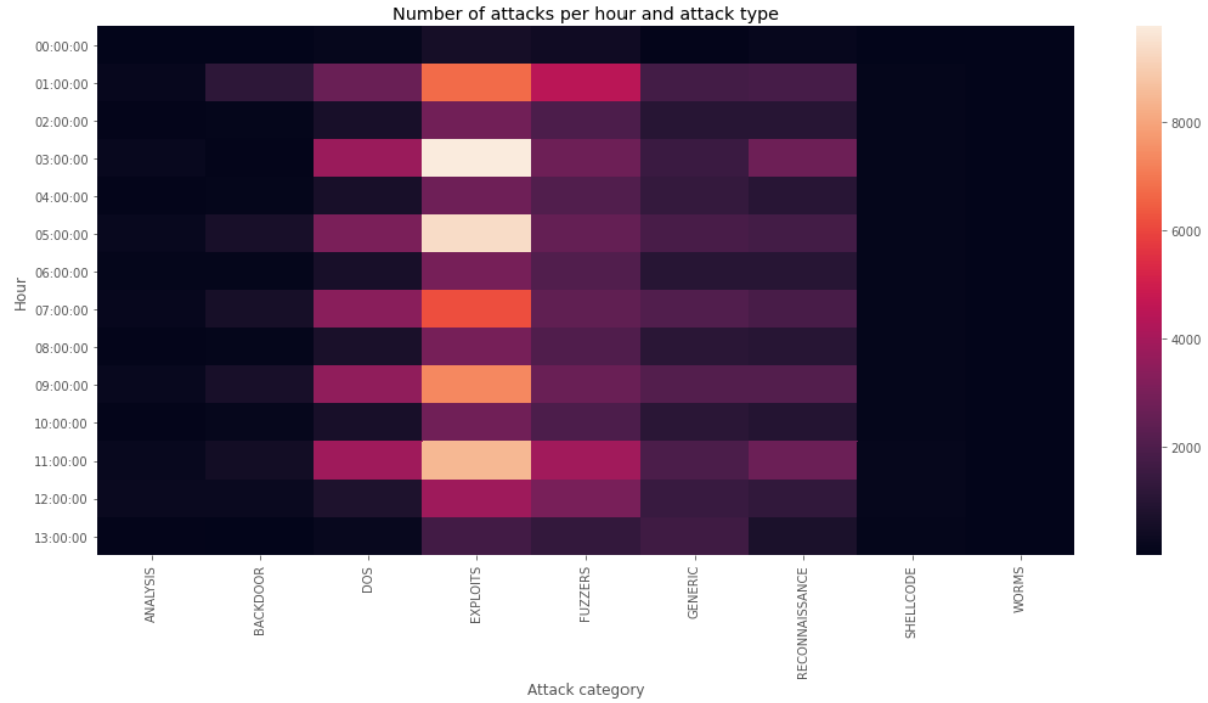
Destination Port	0	10	21	22	23	25	31	42	53	67	...	65455	65460	65470	65479	65486	65497	65502	65511	65517	65535
Attack category																					
ANALYSIS	1442	0	0	0	0	6	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
BACKDOOR	4000	0	7	0	0	0	7	0	0	0	...	0	0	0	0	0	0	0	0	0	0
DOS	20825	4	75	0	13	425	0	0	154	33	...	0	0	0	0	0	0	0	0	0	0
EXPLOITS	40143	0	2198	14	135	4412	0	21	209	98	...	2	2	2	2	2	4	2	2	2	0
FUZZERS	13355	0	758	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
GENERIC	2612	0	26	6	0	427	0	0	13438	54	...	0	0	0	0	0	0	0	0	0	6
RECONNAISSANCE	8324	0	0	0	7	7	0	0	41	0	...	0	0	0	0	0	0	0	0	0	0
SHELLCODE	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
WORMS	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

9 rows × 3182 columns



Speech Emotion Recognition





CHAPTER 6

SOURCE CODE

Problem statement : WHAT CAN WE DERIVE

1) Most targeted Destination IP Address
2) Most Logical Ports attacked
3) Most Frequently/common type of Attack
4) Different time of the day, (odd , hours, day or night)
5) Find the Pattern

Import Modules:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import ipaddress
import numpy as np
from scipy import stats
from scipy.stats import chi2_contingency
from datetime import datetime, timedelta
import math
import missingno as msno
plt.style.use('ggplot')
import warnings
warnings.filterwarnings('ignore')


df = pd.read_csv('Cybersecurity_attacks.csv')
df.shape


df.columns


df.head(4)


df[['Start time','Last time']] = df['Time'].str.split('-',expand=True)
df.head()


df['.'].unique()
```

```
df = df.drop(['.', 'Time'],axis=1)# Drop columns and make a copy in memory of the object
df.head()


df.shape


figure, (ax1, ax2) = plt.subplots(1, 2, figsize=(16,5))
msno.matrix(df, ax=ax1, sparkline=False, color=(0.1, 0.25, 0.35))
msno.bar(df, ax=ax2, color=(0.25, 0.7, 0.25))
plt.show()


df.isnull().sum()


df["Attack subcategory"] = df["Attack subcategory"].fillna("Not Registered")


df.isnull().sum()
df[pd.isnull(df).any(axis=1)].shape
df[df.duplicated()].shape


print('Dimensions before dropping duplicated rows: ' + str(df.shape))
df = df.drop(df[df.duplicated()].index)
print('Dimensions after dropping duplicated rows: ' + str(df.shape))


df[df.duplicated()]


invalid_SP = (df['Source Port'] < 0) | (df['Source Port'] > 65535)
invalid_DP = (df['Destination Port'] < 0) | (df['Destination Port'] > 65535)
df[invalid_SP | invalid_DP]


df = df[~(invalid_SP | invalid_DP)].reset_index(drop=True)


print('Total number of different protocols:', len(df['Protocol'].unique()))
print('Total number of different Attack categories:', len(df['Attack category'].unique()))
df['Protocol'].unique()[:15]


df['Attack category'].unique()


df['Protocol'] = df['Protocol'].str.upper().str.strip()
df['Attack category'] = df['Attack category'].str.upper().str.strip()
df['Attack category'] = df['Attack category'].str.strip().replace('BACKDOORS','BACKDOOR')
```

```
print('Total number of different protocols:', len(df['Protocol'].unique()))
print('Total number of different Attack categories:', len(df['Attack category'].unique()))


df[pd.isnull(df['Attack Reference'])].shape
print(df[pd.isnull(df['Attack Reference'])]['Attack category'].value_counts())
print(df['Attack category'].value_counts())


((df[pd.isnull(df['Attack Reference'])]['Attack
category'].value_counts()/df['Attackcategory'].value_counts())*100).dropna().sort_values(ascend
ing=False)


tcp_ports = pd.read_csv('TCP-ports.csv')
tcp_ports['Service'] = tcp_ports['Service'].str.upper()
tcp_ports.head()

print('Dimensions before merging dataframes: ' ,(df.shape))

newdf = pd.merge(df, tcp_ports[['Port','Service']], left_on='Destination Port', right_on='Port',
how='left')
newdf = newdf.rename(columns={'Service':'Destination Port Service'})

print('Dimensions after merging dataframes: ' + str(newdf.shape))


newdf = newdf.drop(columns=['Port'])
newdf.head()

newdf['Attack category'].unique()

newdf['Attack category'].value_counts()

newdf['Attack category'].value_counts()*100/newdf['Attack category'].value_counts().sum()

plt.figure(figsize=(18,6))
sns.barplot(x=newdf['Attack category'].value_counts().index,y=newdf['Attack
category'].value_counts())
plt.xlabel('Attack Category')
plt.ylabel('Count')
plt.title('Number of attacks per Attack caterogy')
plt.grid(True)


pd.DataFrame(newdf['Attack category'].value_counts())[:]
```

```
a=pd.DataFrame(newdf['Attack category'].value_counts())[:6]


a.plot(kind='pie', subplots=True, figsize=(7, 7))
plt.title('Top five attacks')
plt.legend(loc='left')
plt.show()
```

NOW TO ANALYSE Attacks WITH DATE AND TIME

```
newdf['Start time']

newdf['Start time'] = pd.to_datetime(newdf['Start time'], unit='s')
newdf['Last time'] = pd.to_datetime(newdf['Last time'], unit='s')
newdf['Duration'] = ((newdf['Last time'] - newdf['Start time']).dt.seconds).astype(int)

newdf[:5]

newdf.describe()

statistic, pvalue = stats.ttest_ind( newdf['Source Port'], newdf['Destination Port'],
equal_var=False)
print('p-value in T-test: ' + str(pvalue))

newdf.corr(method='pearson')

newdf.corr(method='spearman')

df_dummies = pd.get_dummies(newdf, columns=['Attack category'])

plt.figure(figsize=(18,7))
sns.heatmap(df_dummies.corr(method='pearson'),
        annot=True, vmin=-1.0, vmax=1.0, cmap=sns.color_palette("RdBu_r", 15))
plt.show()

plt.figure(figsize=(18,7))
sns.heatmap(df_dummies.corr(method='spearman'),
        annot=True, vmin=-1.0, vmax=1.0, cmap=sns.color_palette("RdBu_r", 15))
plt.show()

g = sns.pairplot(newdf)
g.fig.set_size_inches(11,7)
plt.show()
```

```
newdf['Destination IP'].value_counts()[:5]

plt.figure(figsize=(18,7))
sns.scatterplot(x=newdf[newdf['Destination IP']=='149.171.126.17']['Start time'],
y=newdf[newdf['Destination IP']=='149.171.126.17']['Destination Port'])
plt.xlim(left=newdf['Start time'].min()-timedelta(days=1),right=newdf['Start
time'].max()+timedelta(days=1))
plt.grid(True)
plt.show()


plt.figure(figsize=(18,7))
sns.scatterplot(x=newdf[newdf['Destination IP']=='149.171.126.17']['Start time'],
y=newdf[newdf['Destination IP']=='149.171.126.17']['Destination Port'])
plt.xlim(left=newdf['Start time'].min(),right=datetime.strptime('15-01-23', '%y-%m-%d'))
plt.grid(True)
plt.show()


plt.figure(figsize=(18,7))
sns.scatterplot(x=newdf[newdf['Destination IP']=='149.171.126.17']['Start time'],
y=newdf[newdf['Destination IP']=='149.171.126.17']['Destination Port'])
plt.xlim(left=datetime.strptime('15-02-18', '%y-%m-%d'),right=newdf['Start time'].max())
plt.grid(True)
plt.show()


plt.figure(figsize=(18,7))
sns.scatterplot(x='Start time', y='Destination Port', hue='Attack category',

data=newdf[(newdf['DestinationIP']=='149.171.126.17')&(newdf['DestinationPort']<=150)],
            s=65)
plt.xlim(left=datetime.strptime('15-02-18 00:00:00', '%y-%m-%d %H:%M:%S'),
        right=datetime.strptime('15-02-18 13:00:00', '%y-%m-%d %H:%M:%S'))
plt.grid(True)
plt.show()
```

Duration vs Destination Ports

```
plt.figure(figsize=(18,7))
sns.scatterplot(x='Destination Port', y='Duration', hue='Attack category',
data=newdf[newdf['Destination IP']=='149.171.126.17'])
plt.grid(True)
plt.show()
```

```
plt.figure(figsize=(18,7))
sns.violinplot(x='Attack category', y='Duration', data=newdf)
plt.grid(True)
plt.show()

def heatmap_graph(df, xlabel, ylabel, title):
    plt.figure(figsize=(18,8))
    ax = sns.heatmap(df)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.title(title)
    plt.xticks(rotation=90)
    plt.yticks(rotation=0)
    plt.show()

newdf["Start time"][1].hour



df_pivot = newdf.copy()
df_pivot['hour'] = df_pivot.apply(lambda row: '0'*(2-len(str(row['Start
time'].hour)))+str(row['Start time'].hour)+':00:00', axis=1)

df_pivot[:5]


df_p1 = pd.pivot_table(df_pivot,values='Attack Name', index=['hour'], columns=['Attack
category'], aggfunc='count')
df_p1

heatmap_graph(df = df_p1, xlabel = 'Attack category', ylabel = 'Hour', title = 'Number of attacks
per hour and attack type')

heatmap_graph(df = df_p1/df_p1.sum(), xlabel = 'Attack category', ylabel = 'Hour', title =
'Porcentage of attacks per hour per attack type')

df_p2 = pd.pivot_table(df_pivot, values='Attack Name', index=['hour'], columns=['Destination
IP'], aggfunc='count')
heatmap_graph(df = df_p2/df_p2.sum(), xlabel = 'Destination IP', ylabel = 'Hour', title =
'Percentage of attacks per IP and hour')

df_p3 = pd.pivot_table(df_pivot, values='Attack Name', index=['Destination IP'],
columns=['Attack category'], aggfunc='count')
heatmap_graph(df = df_p3/df_p3.sum(), xlabel = 'Attack category', ylabel = 'Destination IP', title
= 'Number of attacks per IP and attack type')
```

```
for attack in list(newdf['Attack category'].unique()):
    df_attack = newdf[newdf['Attack category'] == attack].copy()
    statistic, pvalue = stats.ttest_ind(df_attack['Source Port'], df_attack['Destination Port'],
equal_var=False)
    print('p-value in T-test for ' + attack + ' attack: ' + str(pvalue))

df_crosstab = pd.crosstab(newdf['Attack category'], newdf['Destination Port'])
df_crosstab

chi2, p_value, dof, expected = chi2_contingency(df_crosstab)
print("p-value of Chi-square test for Attack category vs. Destination Port =", p_value)

plt.figure(figsize=(18,7))
sns.scatterplot(x='Source Port',y='Destination Port', hue='Attack category',data=newdf)
plt.show()

# Source ports
plt.figure(figsize=(16,5))
sns.stripplot(x='Attack category',y='Source Port',data=newdf)
plt.show()



# Destination ports
plt.figure(figsize=(16,5))
sns.stripplot(x='Attack category',y='Destination Port',data=newdf)
plt.show()

list(newdf['Source IP'].unique())

ips = list(newdf['Source IP'].unique())
f, axes = plt.subplots(2, 2)
f.set_figheight(10)
f.set_figwidth(15)

labels = list(newdf['Attack category'].unique())
for i, ip in enumerate(ips):
    sns.stripplot(x='Attack category',y='Destination Port',data=newdf[newdf['Source IP'] == ip],
order=labels, ax=axes[int(i/2)][i%2])
    axes[int(i/2)][i%2].set_xlabel('Attack category')
    axes[int(i/2)][i%2].set_ylabel('Destination Port')
    axes[int(i/2)][i%2].set_title('Destination Port distribution - Attacker IPv4 Address: ' + ip)
    axes[int(i/2)][i%2].set_xticklabels(labels,rotation=90)
plt.tight_layout()
plt.show()
```

```
ips = list(newdf['Destination IP'].unique())
f, axes = plt.subplots(5, 2)
f.set_figheight(25)
f.set_figwidth(15)

labels = list(newdf['Attack category'].unique())

for i, ip in enumerate(ips):
    sns.stripplot(x='Attack category',y='Destination Port',data=newdf[newdf['Destination IP'] == ip], order=labels, ax=axes[int(i/2)][i%2])
    axes[int(i/2)][i%2].set_xlabel('Attack category')
    axes[int(i/2)][i%2].set_ylabel('Destination Port')
    axes[int(i/2)][i%2].set_title('Destination Port distribution - Target IPv4 Address: ' + ip)
    axes[int(i/2)][i%2].set_xticklabels(labels,rotation=90)
plt.tight_layout()
plt.show()
```

CHAPTER 7

SYSTEM TESTING

INTRODUCTION

The cause of testing is to detect mistakes. Making an attempt out is the technique of looking for to realize each viable fault or weakness in a piece product. It presents a method to determine the performance of add-ons, sub-assemblies, assemblies and/or a completed product. It is the method of excising program with the intent of constructing certain that the application procedure meets its necessities and client expectations and does no longer fail in an unacceptable process. There are rather plenty of forms of scan. Each experiment sort addresses a special trying out requirement.

TYPES OF TESTING:

Unit testing:

Unit checking out involves the design of scan circumstances that validate that the Internal application good judgement is functioning safely, and that program inputs produce legitimate outputs. All decision branches and interior code float must be validated. It's the checking out of character application items of the application. It is achieved after the completion of a person unit earlier than integration. It is a structural checking out, that relies on competencies of its construction and is invasive. Unit exams participate in common exams at component level and scan a distinct business approach, utility, and/or process configuration. Unit assessments be certain that every specified course of an industry method performs appropriately to the documented requisites and involves clearly outlined inputs and anticipated results. In our project every input can produce the expected result without any error.

Integration testing:

Integration Testing are designed to scan built-in program accessories to determine within the occasion that they evidently run as one software. Trying out is occasion driven and is more concerned with the fundamental final result of screens or fields. Integration assessments reveal that despite the fact that the accessories had been for my part pleasure, as proven through effectively unit checking out, the combo of accessories is correct and regular. Integration checking out is chiefly aimed at exposing the issues that come up from the performance of different components. The attacks can be identified easily with the help of integrated components such as IP addresses and time of the day together into single component.

Functional testing:

Functional Testing checks provide systematic demonstrations that capabilities established are to be had as particular by means of the business and technical specifications, method documentation, and consumer manuals. Functional testing is intended to validate that an application does what it is supposed to do. For example, functional tests may test an application's authentication mechanism to check that legitimate users can authenticate successfully while invalid login attempts are rejected. Functional testing is working on below mentioned data:

- ✓ Legitimate input: identified lessons of legitimate input ought to be accredited.
- ✓ Noisy voice : recognized lessons of unacceptable IP signals must be rejected.
- ✓ Capabilities : recognized features ought to be exercised in terms of day, time and hour.
- ✓ Output : recognize the attacks and detect the pattern.

Systems/Procedures:

Performance of the system here was invoked Individual and systematic concerning method flows; data fields, predefined processes, and successive strategies have to be regarded for trying out the approach taken by most organisations to counter cyber attacks is defensive and reactionary. Threats are only removed and analysed once they are detected; at which point, the harm is done the network has already been breached and valuable information compromised. Intrusion detection and prevention, such as anti-virus software and firewalls combined with access controls such as passwords, are the common tools and measures employed by the majority of organisations. Considering how sophisticated and multi-faceted cyber crimes have been in recent years, and how numerous the attacks that don't make the tabloids are, however, it's safe to say reactionary responses are damage control measures, at best, and are largely ineffective. The solution, however, is far easier to hypothesise than actually implement. Governments and organisations need to start predicting cyber attacks and threats and commit to attack simulations across their systems without delay.

System testing:

System testing is based on approach descriptions and flows, emphasizing pre-driven system links and integration aspects. The focus is only given to a few key elements:

- ✓ Preventative security: Strong passwords that prevent USB devices from accessing open ports.
- ✓ Network design security: Minimising vulnerabilities and isolating them to prevent a network-wide compromise in the event of a breach.
- ✓ Active security: Encryption, protocol-specific deep packet inspection, layer three firewalls and powerful antivirus software.
- ✓ Detective security: Evaluating activity registers and logs to identify a threat in real-time and monitoring intrusion detection systems.

- ✓ Corrective security: Limiting the extent of the damage if an incident occurs by updating antivirus and firewall software and having a configuration parameter backup policy.

White Box Testing:

This testing is a trying out wherein where the application tester has competencies of the interior workings, constitution and software language. White Box Penetration testing is a method where some information is known about the network or the application. A Penetration Tester will be provided credentials to access web applications and whitelisted so that they are not blocked by firewalls or intrusion detection systems. In the internal network test case, the penetration tester will be provided with a range of IP addresses and a foothold within the network. They may also be provided a network diagram or other information about the devices on the network.

This type of penetration testing may provide the most in-depth testing and the best idea of what a well-informed attacker could exploit. Testers can identify vulnerabilities and target the most critical systems or areas of the application to identify any security weaknesses. White-box testers thoroughly study the code and other internal aspects of the given software, determine all the valid or invalid inputs. Using this data, they then verify the outputs against the expected outcomes. They check the statements and conditions, the code paths, and data-flows to ensure there are no hidden errors or defect-prone elements.

Black Box Testing:

Black box testing, a form of testing that is performed with no knowledge of a system's internals, can be carried out to evaluate the functionality, security, performance, and other aspects of an application. Dynamic code analysis is an example of automated black box security testing. Black box evaluators define test cases and interact with the software like a user would to validate that it does what it should, how it should.

LEVELS OF TESTING

- ✓ Unit testing strategy
- ✓ Unit checking out is most commonly performed as a part of a mixed code and unit.
- ✓ experiment part of the software life cycle, though it be not exceptional for coding and unit checking out to be performed as two targeted phases.

Test strategy and approach:

- ✓ Field testing out can be carried out manually and sensible assessments shall be written in element.
- ✓ Test objectives.
- ✓ each field must be work correctly.

- ✓ each page must be activated through the specified link.
- ✓ Features to be tested Verify that the entries are of the correct format No duplicate entries should be allowed.

Test Results:

All of the scan circumstances recounted above passed efficiently. No defects encountered.

Acceptance Testing

User Acceptance testing trying out is a crucial section of any mission and requires enormous participation by the tip user. It additionally ensures that the procedure meets the functional specifications. specific elements of *confidentiality, integrity, authentication, availability, authorization and non-repudiation*. Actual security requirements tested depend on the security requirements implemented by the system.

Test Results:

The entire test cases recounted above passed effectually. No defects Encountered.

VIII. CONCLUSION

Prediction of Cyber Attacks by using Machine Learning came under cybersecurity. Cybersecurity is critical because it helps to protect us from cyber attacks and other threats. By being aware of the risks and taking steps to mitigate them, we can help to keep our data and systems safe. The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. We detect Most targeted Destination IP Address, Most Logical Ports attacked, Most Frequently/common type of Attack, Different time of the day(odd, hours, day or night) and finally Find the Pattern. The best accuracy score will be finding out by comparing each algorithm with type of all network attacks for future prediction results by finding best connections.

IX.FUTURE ENHANCEMENT

A career in cybersecurity is a lucrative, future transforming, and twice the career scope for the next generation. It is an in-demand career with abundant queries and twisting predictions. The cybersecurity job employment rate is estimated at 100% and the worldwide cybersecurity market will reach approx. 6 billion in 2023.

cybersecurity could be an eye-opener and how emerging technologies including Artificial Intelligence (AI), Machine Learning, cloud computing, and the Internet of Things (IoT) will strike the future of cybersecurity. cybersecurity businesses will make key pieces of infrastructure, those which would make appealing targets during a cyberwar, stronger to digital intrusions.

Meaning definitely involves putting up more security to commute systems, management networks, and relatable databases.

The future of the cybersecurity industry will reverse in the next coming years. Here's what the probable future of cyber security might look like:

- Contribution of Machine learning and AI

- Quantum computing raise the capacity of hacker and defender.

- Obsolete technologies impact cybersecurity after each upgradation.

- Nature of cybersecurity threats.

- Nature of cybersecurity practice.

IX- REFERENCES

1. Assistant Professor/Lecturer (on Deputation), Department of Computer Science and Engineering Annamalai University, Annamalainagar, Tamil Nadu, India.
2. AasthaJoshi “cyber attack prediction Using Combined Features & SVM Algorithm”, National Conference on August 2013.
3. AnkurSapra, Nikhil Panwar, SohanPanwar “cyber crimes”, International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 2, pp. 341-345, February 2013.
4. BjörnSchuller, Manfred Lang, Gerhard Rigoll “security detection Automatic by the ip Signal”, National Journal on 2013, Volume 3, Issue 2, pp. 342-347.