A Summer Internship Project Report on

CLASSIFICATION OF NETWORK TRAFFIC

Submitted to
The Department of Computer Science and Engineering
In partial fulfillment of the academic requirements of
Jawaharlal Nehru Technological University
For
The award of the degree of

Bachelor of Technology
in
Computer Science and Engineering
(2018-2022)
By

GOPU NAVEEN	18311A05T0
VADLAKONDA ROHITH	18311A05W8
CHITYALA UDHAY	18311A05R4

Under the Guidance of
Mr.CHANDU NAIK
ASSISANT PROFESSOR
Department of Cse

Sreenidhi Institute of Science and Technology
Yamnampet, Ghatkesar, R.R. District, Hyderabad - 501301

Affiliated to Jawaharlal Nehru Technology University,Hyderabad - 500085
Department of Computer Science And Engineering

Sreenidhi institute of Science And Technology

CERTIFICATE

This is to certify that this Internship Project report on **"CLASSIFICATION OF NETWORK TRAFFIC"** submitted by GOPU NAVEEN(18311A05T0), VADLAKONDA ROHITH (18311A05W8) , CHITYALA UDHAY (18311A05R4) **)** in the year 2020 in partial fulfillment of the academic requirements of Jawaharlal Nehru Technological University for the award of the degree of Bachelor of Technology in Computer Science and Engineering, is a bonafide work that has been carried out by them as part of their Summer Industry Internship –I Project , under our guidance. This report has not been submitted to any other institute or university for the award of any degree.

Internal Guide	Project Co-ordinator	Head of Department
Mr.A.Chandu Naik	Mrs.N.Shivani	Dr.Aruna Varanasi
Assistant Professor	Assistant Professor	Professor and HOD
Department of Cse	Department of Cse	Department of Cse

External Examiner
Date:

GOALSTREET

ORBIT SHIFTERS

http://internships.mygoalstreet.com/

https://www.orbitshifters.com/

Certificate Code: GSSIE2020MLB200801

SUMMER INTERNSHIPS

Project Completion Certificate

Date: 31/7/2020

To whomsoever it may concern,

This is to certify that "**GOPU NAVEEN**" bearing roll number **18311A05T0** respectively of **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY** has successfully completed the Capstone Mini project title "**CLASSIFICATION OF NETWORK TRAFFIC**" under the guidance of **NAVEEN KUMAR C** for a period from **25-May-20** to **25-Jul-20**.This project is a bonafide work undertaken by them towards partial fulfillment for the award of B.Tech in "**CSE**". During the course of completion of project, their performance was Good.

All the Best in your future endeavors.

Cheers!

Authorized Signatory
GoalStreet Internships

Note: This is a certificate copy, and the original copy can be found in the URL: internships.mygoalstreet.com/SIE2020/Certificates

ORBIT SHIFTERS Inc in association with GOALSTREET	Address: 4F2, 5th Floor, Ballad Estates, Tarnaka, Hyderabad. Ph.No.: +91-9246818840,

GOALSTREET
http://internships.mygoalstreet.com/

ORBIT SHIFTERS
Transforming Technologies
https://www.orbitshifters.com/

Certificate Code: GSSIE2020MLB200801

SUMMER INTERNSHIPS

Internship Completion Certificate

Date: 31/7/2020

To whomsoever it may concern,

We are glad to inform you that "GOPU NAVEEN" has been pursuing a 6 weeks Remote internship on "Machine Learning with Python" starting from 25-May-20. We hereby confirm that he/she has successfully completed the mentioned internship as part of GOAL STREET INTERNSHIPS.

During **his/her** Internship, **he/she** had worked under well-rounded mentors from the Industry for training in the particular discipline. Additionally, "**GOPU NAVEEN**" has also played a crucial role in working on a live "Capstone" project alongside fellow teammates to exhibit exemplary coordination and skill, and at the same time, was able to apply valuable knowledge and experience in facilitating the completion of the Project. The project, namely, "**CLASSIFICATION OF NETWORK TRAFFIC**" stands testimony to the skills and knowledge obtained as part of their journey here at GOALSTREET.

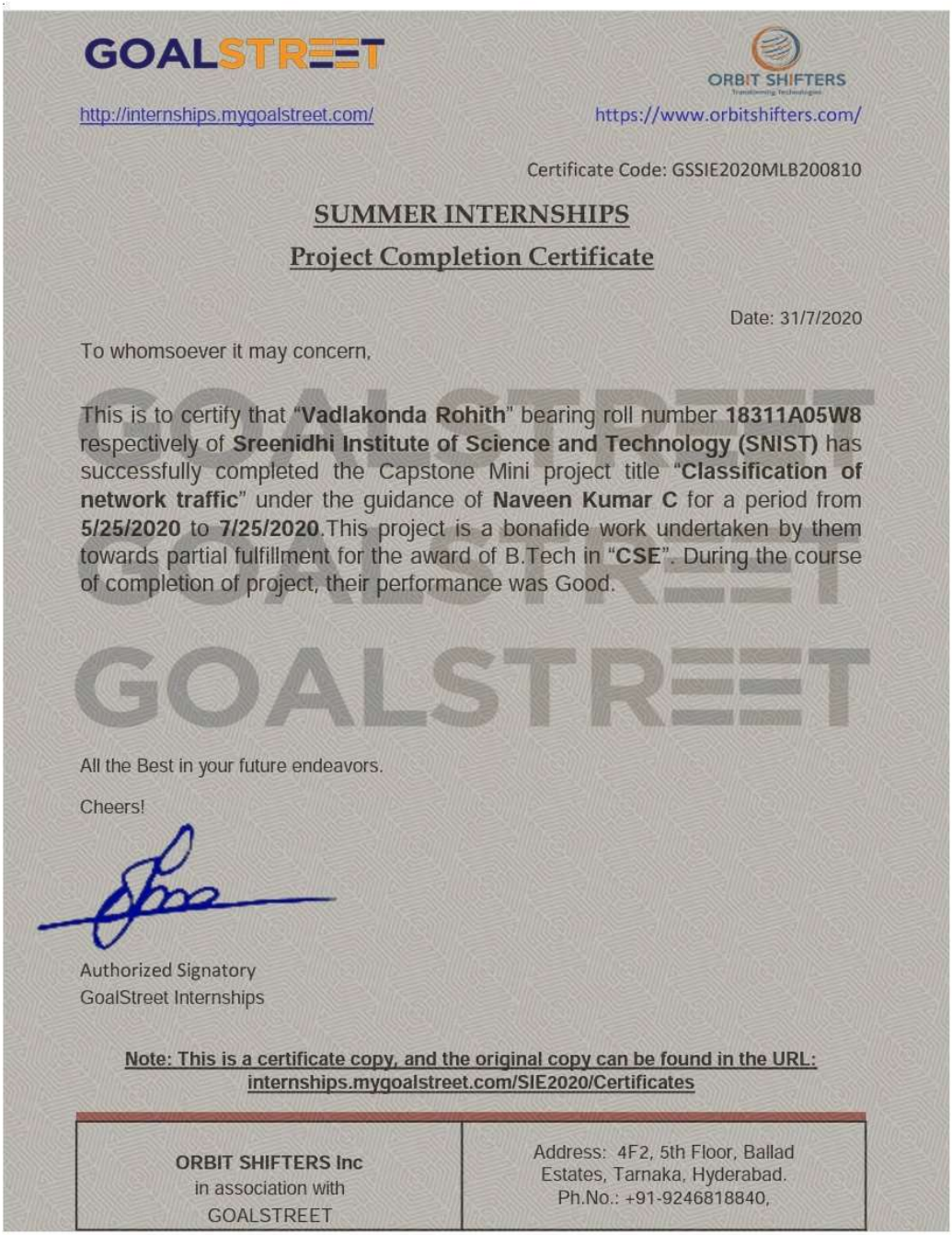
Internships have been a great opportunity for any student to learn experience and evolve as a better professional in this ever-changing engineering stream. We, on behalf of Orbit Shifters and GoalStreet Team, congratulate and wish you a bright career.

All the Best

Authorized Signatory
GoalStreet Internships

Note: This is a certificate copy, and the original copy can be found in the URL: internships.mygoalstreet.com/SIE2020/Certificates

ORBIT SHIFTERS Inc in association with **GOALSTREET**	Address: 4F2, 5th Floor, Ballad Estates, Tarnaka, Hyderabad. Ph.No.: +91-9246818840,



GOALSTREET

ORBIT SHIFTERS

http://internships.mygoalstreet.com/ https://www.orbitshifters.com/

Certificate Code: GSSIE2020MLB200810

SUMMER INTERNSHIPS

Project Completion Certificate

Date: 31/7/2020

To whomsoever it may concern,

This is to certify that "**Vadlakonda Rohith**" bearing roll number **18311A05W8** respectively of **Sreenidhi Institute of Science and Technology (SNIST)** has successfully completed the Capstone Mini project title "**Classification of network traffic**" under the guidance of **Naveen Kumar C** for a period from **5/25/2020** to **7/25/2020**.This project is a bonafide work undertaken by them towards partial fulfillment for the award of B.Tech in "**CSE**". During the course of completion of project, their performance was Good.

GOALSTREET

GOALSTREET

All the Best in your future endeavors.

Cheers!

Authorized Signatory
GoalStreet Internships

Note: This is a certificate copy, and the original copy can be found in the URL: internships.mygoalstreet.com/SIE2020/Certificates

ORBIT SHIFTERS Inc in association with GOALSTREET	Address: 4F2, 5th Floor, Ballad Estates, Tarnaka, Hyderabad. Ph.No.: +91-9246818840,

GOALSTREET
http://internships.mygoalstreet.com/

ORBIT SHIFTERS
Transforming Technologies
https://www.orbitshifters.com/

Certificate Code: GSSIE2020MLB200810

SUMMER INTERNSHIPS

Internship Completion Certificate

Date: 31/7/2020

To whomsoever it may concern,

We are glad to inform you that "**Vadlakonda Rohith**" has been pursuing a **6 weeks** Remote internship on "**Machine Learning using Python**" starting from **5/25/2020**. We hereby confirm that **he/she** has successfully completed the mentioned internship as part of GOAL STREET INTERNSHIPS.

During **his/her** Internship, **he/she** had worked under well-rounded mentors from the Industry for training in the particular discipline. Additionally, "**Vadlakonda Rohith**" has also played a crucial role in working on a live "Capstone" project alongside fellow teammates to exhibit exemplary coordination and skill, and at the same time, was able to apply valuable knowledge and experience in facilitating the completion of the Project. The project, namely, "**Classification of network traffic**" stands testimony to the skills and knowledge obtained as part of their journey here at GOALSTREET.

Internships have been a great opportunity for any student to learn experience and evolve as a better professional in this ever-changing engineering stream. We, on behalf of Orbit Shifters and GoalStreet Team, congratulate and wish you a bright career.

All the Best

Authorized Signatory
GoalStreet Internships

Note: This is a certificate copy, and the original copy can be found in the URL: internships.mygoalstreet.com/SIE2020/Certificates

ORBIT SHIFTERS Inc in association with **GOALSTREET**	Address: 4F2, 5th Floor, Ballad Estates, Tarnaka, Hyderabad. Ph.No.: +91-9246818840,

GOALSTREET

ORBIT SHIFTERS

http://internships.mygoalstreet.com/

https://www.orbitshifters.com/

Certificate Code: GSSIE2020MLB200816

SUMMER INTERNSHIPS

Project Completion Certificate

Date: 31/7/2020

To whomsoever it may concern,

This is to certify that "**Udhay Chityala**" bearing roll number **18311A05R4** respectively of **Sreenidhi institute of science and technology** has successfully completed the Capstone Mini project title "**Classification of network traffic**" under the guidance of **C.Naveen kumar** for a period from **25-May-20** to **25-Jul-20**.This project is a bonafide work undertaken by them towards partial fulfillment for the award of B.Tech in "**CSE**". During the course of completion of project, their performance was Good.

All the Best in your future endeavors.

Cheers!

Authorized Signatory
GoalStreet Internships

Note: This is a certificate copy, and the original copy can be found in the URL: internships.mygoalstreet.com/SIE2020/Certificates

ORBIT SHIFTERS Inc in association with GOALSTREET	Address: 4F2, 5th Floor, Ballad Estates, Tarnaka, Hyderabad. Ph.No.: +91-9246818840,

GOALSTREET
http://internships.mygoalstreet.com/

ORBIT SHIFTERS
Transforming Technologies
https://www.orbitshifters.com/

Certificate Code: GSSIE2020MLB200816

SUMMER INTERNSHIPS

Internship Completion Certificate

Date: 31/7/2020

To whomsoever it may concern,

We are glad to inform you that "Udhay Chityala" has been pursuing a 6 weeks Remote internship on "Machine Learning With Python" starting from 25-May-20. We hereby confirm that he/she has successfully completed the mentioned internship as part of GOAL STREET INTERNSHIPS.

During **his/her** Internship, **he/she** had worked under well-rounded mentors from the Industry for training in the particular discipline. Additionally, "**Udhay Chityala**" has also played a crucial role in working on a live "Capstone" project alongside fellow teammates to exhibit exemplary coordination and skill, and at the same time, was able to apply valuable knowledge and experience in facilitating the completion of the Project. The project, namely, "**Classification of network traffic**" stands testimony to the skills and knowledge obtained as part of their journey here at GOALSTREET.

Internships have been a great opportunity for any student to learn experience and evolve as a better professional in this ever-changing engineering stream. We, on behalf of Orbit Shifters and GoalStreet Team, congratulate and wish you a bright career.

All the Best

Authorized Signatory
GoalStreet Internships

Note: This is a certificate copy, and the original copy can be found in the URL: internships.mygoalstreet.com/SIE2020/Certificates

ORBIT SHIFTERS Inc in association with **GOALSTREET**	Address: 4F2, 5th Floor, Ballad Estates, Tarnaka, Hyderabad. Ph.No.: +91-9246818840,

DECLARATION

We, **GOPU NAVEEN (18311A05T0), VADLAKONDA ROHITH (18311A05W8) and CHITYALA UDHAY (18311A05R4)** students of **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY, YAMNAMPET, GHATKESAR, COMPUTER SCIENCE AND ENGINEERING** solemnly declare that the Internship project work, titled **"CLASSIFICATION OF NETWORK TRAFFIC"** is submitted to **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY** for partial fulfillment for the award of degree of Bachelor of technology in **COMPUTER SCIENCE AND ENGINEERING**.

It is declared to the best of our knowledge that the work reported does not form part of any dissertation submitted to any other University or Institute for award of any degree.

ACKNOWLEDGEMENT

I would like to express my gratitude to all the people behind the screen who helped me to transform an idea into a real application.

I would like to express my heart-felt gratitude to my parents without whom I would not have been privileged to achieve and fulfill my dreams. I am grateful to our principal, **Dr. T. Ch. Siva Reddy,** who most ably run the institution and has had the major hand in enabling me to do my project.

I profoundly thank **Dr. ARUNA VARANASI**, Head of the Department of Computer Science & Engineering who has been an excellent guide and also a great source of inspiration to my work.

I would like to thank my internal guide Mr.A.Chandu Naik for her technical guidance, constant encouragement and support in carrying out my project at college.

The satisfaction and euphoria that accompany the successful completion of the task would be great but incomplete without the mention of the people who made it possible with their constant guidance and encouragement crowns all the efforts with success. In this context, I would like thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased my task.

GOPU NAVEEN **18311A05T0**
VADLAKONDA ROHITH **18311A05W8**
CHITYALA UDHAY **18311A05R4**

CLASSIFICATION OF NETWORK TRAFFIC

Abstract

In our society, information systems are everywhere. They are used by corporations to store proprietary and other sensitive data, by families to store financial and personal information, by universities to keep research data and ideas, and by governments to store defense and security information.

It is very important that the information systems that house this vitally sensitive information be secure. In order for information systems to be secure, it is paramount that they utilize robust security mechanisms.

Commonly found security mechanisms are passwords on accounts, encryption of sensitive data, virus protection, and intrusion detection. An Intrusion Detection System (IDS) monitors activity at an access point and can log or prevent activities that are marked as intrusions. Intrusions occur when malicious activity gains access to or affects the usability of a computer resource.

The goal of this project is two-fold. First, we attempt to find the most effective machine learning model for identifying network attacks.

Although scalability and performance are major considerations in every commercial product, our results are targeted at minimizing false positives and negatives.

INDEX

INTRODUCTION

- We know that in this world information systems are everywhere like universities,government offices,stock markets etc.
- Protection of information is very important in equal with maintaining information.
- Now a days we are maintaining many security mechanisms like passwords,firewalls etc,but still there are many chances of instrusion attacks.this is results in stealing of information.
- In the current scenario, our single aadhar card can give our whole information.
- An Intrusion Detection System (IDS) monitors activity at an access point and can log or prevent activities that are marked as intrusions.
- Intrusions occur when malicious activity gains access to or affects the usability of a computer resource.

Example :In our universities there is Admin who does allow other intruders without his intervention.

1.1 SCOPE

- Today we know information systems are everywhere.
- Protection of our information systems is very important.
- The main aim of this project is to detect an Instruder who enters through macilious attacks.i.e Intrusion Detection

1.2 EXISTING SYSTEM

The existing system uses Support Vector Machine (SVM) one of the main issues with this is that it need the data to be linearly separable.
The system also does not provide enough preprocessing and visualization or Exploratory Data Analysis(EDA).

Disadvantages of Existing System:

- The limitations of available systems are not sufficient to deal with the complex data.
- In this section, we present some of the limitations that are present in the existing system.
- The model suffers from overfitting due to no generalization of data.
- The error on test data is high due to overfitting.
- The system also requires data extensive data preprocessing and Exploratory Data Analysis(EDA) inorder to perform feature engineering.

1.3 PROPOSED SYSTEM

We aim to build other classification models like logistic regression, Naïve Bayes, Decision Trees and others and also fine tune the parameters of the model. These models would be trained on a data set which will be engineered carefully after performing the feature engineering.

Advantages:

- The requirement is to come up with novel features based on the functional understanding of the dataset. It is important to keep in mind to avoid correlated features during this process. Each feature should only improve the information contained in the dataset.

- visualize the dataset without cleaning the data and understand the distribution of the dataset.

- Find the attributes that needs to be handled based on the data visualization task.

1.4 DESCRIPTION OF DATASET

- For our research used the NSL KDD dataset [10], it is a better version of the KDD Cup 99 dataset. One of the major drawbacks with the KDD Cup 99 dataset is a large number of duplicate observations in test and train, the NSL KDD dataset overcomes these limitations hence, it suits our purpose of building robust predictive models.
- For each observation in the NSL KDD dataset, there are 41 features,3 are nominal, 4 are binary and the remaining 34 are continuous variables. It has 23 traffic classes in the training dataset and 30 in the test dataset.
- These attacks can be clustered into four main categories DOS, probing, U2R and R2L. The features are classified into 3 broad types
- 1) basic features,
- 2) content-based features and
- 3) traffic-based features.
- The attack information of the NLS-KDD dataset is listed in Table 1 and Table 2.

Table 1. Dataset network intrusion details.

Traffic	Train	Test
Normal	67,343	9,711
Dos	45,927	7,458
U2R	52	67
R2L	995	2,887
Probe	11,656	2,421

Table 2. Subcategories of intrusions under each broader class intrusion (The highlighted attacks are only present in the test dataset).

Category	Attacks
DoS	back, land, neptune, pod, smurf, teardrop, mailbomb,processtable, udpstorm,apache2,worm
R2L	fpt-erite,guess-passwd, imap, multihop, phf, spy, warezmaster, xlock,xsnoop,snmpguess,snmpgetattack,httptunnel,sendmail,named
U2R	buffer-overflow, loadmodule, perl, rootkit, sqlattack, xterm, ps
Probe	ipsweep, nmap, portsweep, satan, mscan, saint

2.SYSTEM ANALYSIS

2.1 SOFTWARE REQUIREMENTS

- Operating System : Windows 7 , Windows 8, (or higher versions)
- Language : Python 3.5 and other libraries likes numpy, pandas, matplotlib, seaborn and scikitlearn.
- Mozilla Firefox(or any browser)

2.2 HARDWARE REQUIREMENTS

- Processor : Pentium 3,Pentium 4 and higher
- RAM : 2GB/4GB RAM and higher
- Hard disk : 40GB and higher

3.SYSTEM DESIGN

3.1 Archirectural design

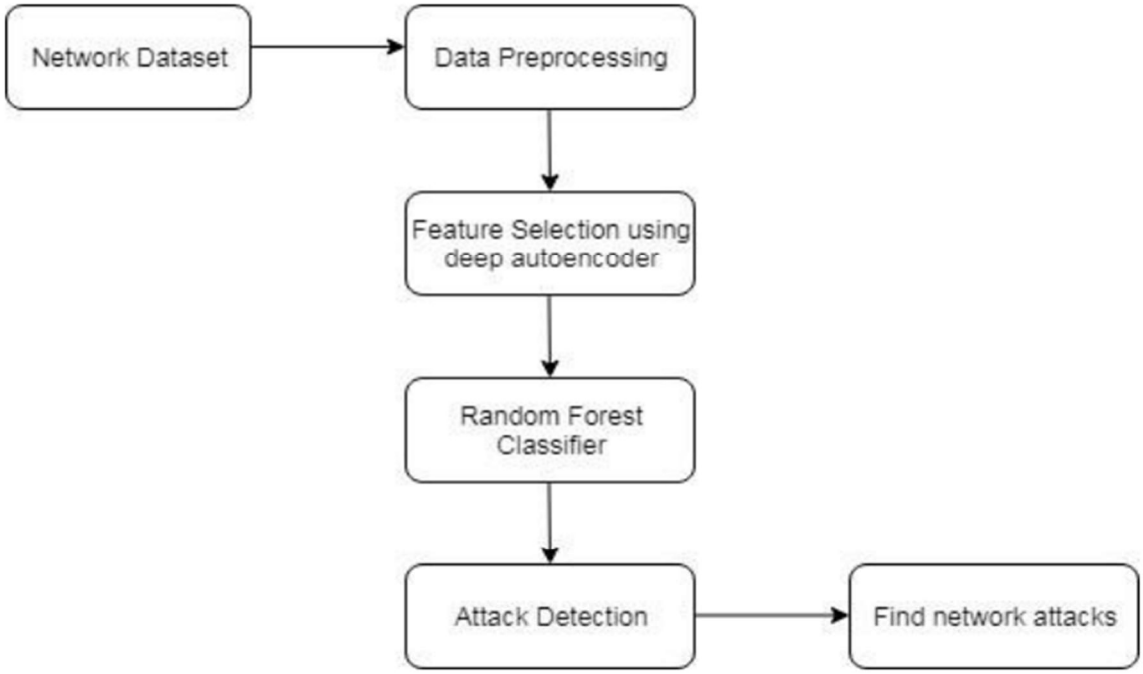


Fig. 1. Proposed System Architecture

The Proposed system consists of mainly three steps. In this section, we discuss each step in detail.
1. Data Preprocessing: This phase is made up of preprocessing, normalization and transformation.
a) Preprocessing Neural network-based classification only uses numerical values for training and testing.dataset consists of different data types. Hence a preprocessing stage is needed to convert the nonnumerical values to numerical values.

Two main tasks in pre-processing are:
1) Converting the non-numerical features in the dataset to numerical values.
2) Convert the attack types into its numeric categories b) Normalization

3.2 UML DIAGRAMS

3.2.1 USE CASE DIAGRAM

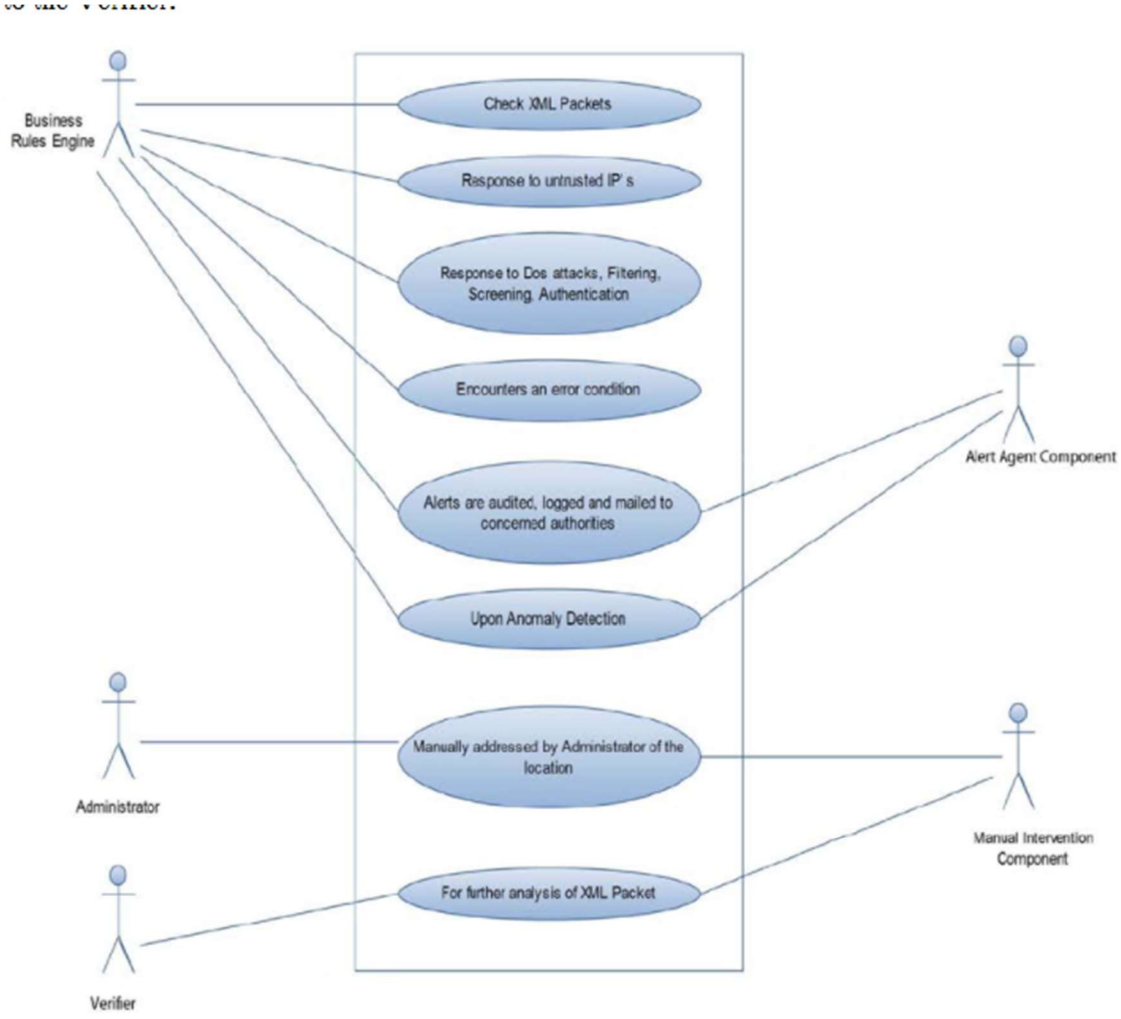


Fig. 2. Use-Case diagram drawn between Business Rules Engine, Alert Agent, Verifier and Manual Intervention using UML 5.0

3.2.2 Class Diagrams

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 modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

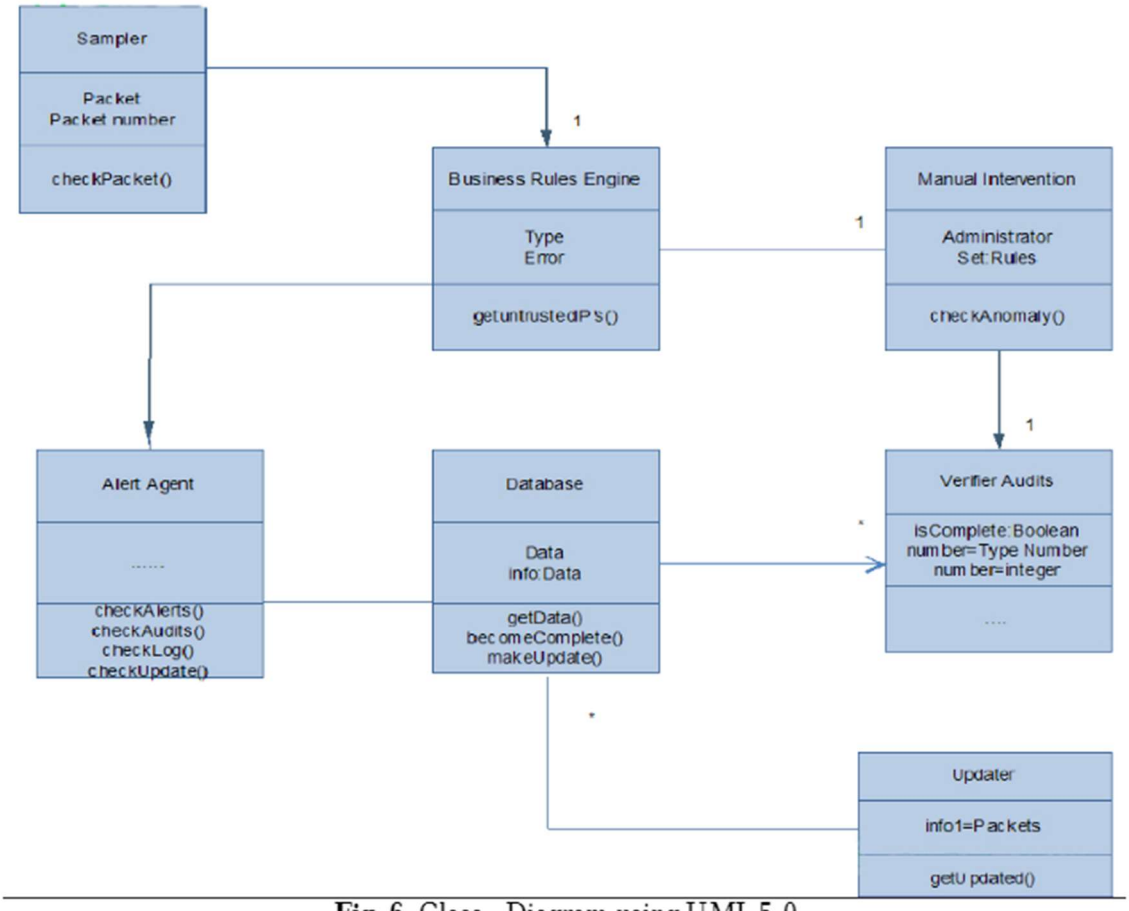


Fig. 6. Class –Diagram using UML 5.0

3.2.3 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

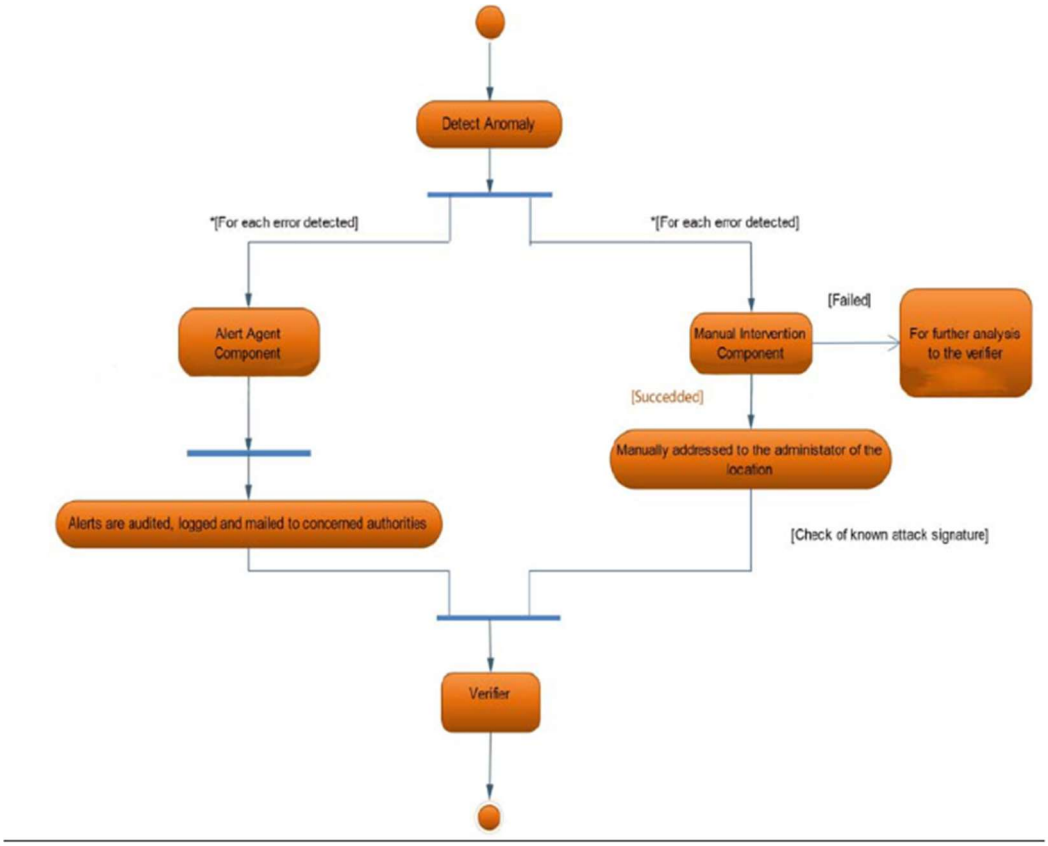


Fig. 3. Activity Diagram between the Business Rules Engine, Manual Intervention, Alert Agent and Verifier using UML 5.0

4.SYSTEM IMPLEMENTION

- Some of the machine learning algorithms used to implement are:
- Logistic Regression
- Decision Tree Classifier
- Random Forest Classifier
- K-Neighbours Classifier
- Gaussian NB
- The above algorithms are used to detect there is malicious attack or not.
- The one with high accuracy is the best algorithm.
- First we imported libraries.
- Then imported the dataset.
- Then we dropped less important features after understanding.
- Describe function used to know if there are any missing values.
- Then we are encoding the categorical data.
- Then we are dividing dataset into training and testing sets.
- Next we are scaling the features.
- Then appling different machine learning algorithms and calculating their accuracy.

IMPLEMENTATION CODE IN PYTHON:

IMPORTING LIBRARIES

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Feature Selection

Here we some less important features

```
dataset.drop(['protocol_type','service','land','urgent',],axis=1,inplace=True)
```

```
dataset.drop(['hot','lnum_file_creations'],axis=1,inplace=True)

temp=pd.get_dummies(dataset['flag'],drop_first=True)

dataset=pd.concat([dataset,temp],axis=1)

dataset.drop('flag',axis=1,inplace=True)

temp1=pd.get_dummies(dataset['label'],drop_first=True)


dataset=pd.concat([dataset,temp1],axis=1)

dataset.drop('label',axis=1,inplace=True
```

Encoding the categorical data

```
x=dataset.iloc[:,0:44]
y=dataset.iloc[:,44]
```

Splitting into Training and Testing data

```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=1)
```

Scaling the training and testing data

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()

x_train=sc.fit_transform(x_train)
x_test=sc.transform(x_test)
```

Applying Logistic Regression Algorithm

```
from sklearn.Linear_model import LogisticRegression
reg=LogisticRegression()
reg.fit(x_train,y_train)
pred_logistic_reg=reg.predict(x_test)
```

Calculating accuracy of Logistic Regression

```
from sklearn.metrics import confusion_matrix,accuracy_score
score_logistic_reg=accuracy_score(y_test,pred_logistic_reg)*100
```

Applying Decision Tree Classifier Algorithm and its accuracy score

```
from sklearn.tree import DecisionTreeClassifier
```

```
classifier=DecisionTreeClassifier(criterion='entropy',random_state=0)

classifier.fit(x_train,y_train)


y_pred_decision_tree=classifier.predict(x_test)

score_decision_tree=accuracy_score(y_test,y_pred_decision_tree)**100
```

Applying Random Forest Classifier Algorithm

```
from sklearn.ensemble import RandomForestClassifier

rfclassifier=RandomForestClassifier(n_estimators=10,criterion='entropy',random_
state=0)

rfclassifier.fit(x_train,y_train)

y_pred_random_forest=rfclassifier.predict(x_test)

score_random_forest=accuracy_score(y_test,y_pred_random_forest)
```

Applying k neighbors Algorithm

```
from sklearn.neighbors import KNeighborsClassifier

knn=KNeighborsClassifier(n_neighbors=5)

knn.fit(x_train,y_train)

y_pred_knn=knn.predict(x_test)
```

```
score_knn=accuracy_score(y_test,y_pred_knn)*100
```

Applying Gaussion NB Algorithm

```
from sklearn.naivebayes import GaussianNB

model=GaussianNB()

model.fit(x_train,y_train)

y_pred_naive_bayes=model.fit(x_test)
score_naive_bayes=accuracy_score(y_test,y_pred_naive_bayes)
```

5.OUTPUT SCREENS

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
path=r"C:\Users\gopun\Downloads\project 13\isd_project.csv"
dataset=pd.read_csv(path)
```

In [3]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26013 entries, 0 to 26012
Data columns (total 42 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   duration                     26013 non-null  int64  
 1   protocol_type                26013 non-null  object 
 2   service                      26013 non-null  object 
 3   flag                         26013 non-null  object 
 4   src_bytes                    26013 non-null  int64  
 5   dst_bytes                    26013 non-null  int64  
 6   land                         26013 non-null  int64  
 7   wrong_fragment               26013 non-null  int64  
 8   urgent                       26013 non-null  int64  
 9   hot                          26013 non-null  int64  
 10  num_failed_logins            26013 non-null  int64  
 11  logged_in                    26013 non-null  int64  
 12  lnum_compromised             26013 non-null  int64  
 13  lroot_shell                  26013 non-null  int64  
 14  lsu_attempted                26013 non-null  int64  
 15  lnum_root                    26013 non-null  int64  
 16  lnum_file_creations          26013 non-null  int64  
 17  lnum_shells                  26013 non-null  int64  
 18  lnum_access_files            26013 non-null  int64  
 19  lnum_outbound_cmds           26013 non-null  int64  
 20  is_host_login                26013 non-null  int64  
 21  is_guest_login               26013 non-null  int64  
 22  count                        26013 non-null  int64  
 23  srv_count                    26013 non-null  int64  
 24  serror_rate                  26013 non-null  float64
 25  srv_serror_rate              26013 non-null  float64
 26  rerror_rate                  26013 non-null  float64
 27  srv_rerror_rate              26013 non-null  float64
 28  same_srv_rate                26013 non-null  float64
 29  diff_srv_rate                26013 non-null  float64
 30  srv_diff_host_rate           26013 non-null  float64
 31  dst_host_count               26013 non-null  int64  
 32  dst_host_srv_count           26013 non-null  int64  
 33  dst_host_same_srv_rate       26013 non-null  float64
 34  dst_host_diff_srv_rate       26013 non-null  float64
 35  dst_host_same_src_port_rate  26013 non-null  float64
 36  dst_host_srv_diff_host_rate  26013 non-null  float64
 37  dst_host_serror_rate         26013 non-null  float64
 38  dst_host_srv_serror_rate     26013 non-null  float64
 39  dst_host_rerror_rate         26013 non-null  float64
 40  dst_host_srv_rerror_rate     26013 non-null  float64
 41  label                        26013 non-null  object 
dtypes: float64(15), int64(23), object(4)
memory usage: 8.3+ MB
```

In [4]:
```
dataset.drop(['protocol_type','service','land','urgent',],axis=1,inplace=True)
```

In [5]:
```
dataset.drop(['hot','lnum_file_creations'],axis=1,inplace=True)
```

In [6]:
```
temp=pd.get_dummies(dataset['flag'],drop_first=True)
dataset=pd.concat([temp,dataset],axis=1)
dataset.drop('flag',axis=1,inplace=True)
```

In [7]:
```
x=dataset.iloc[:,0:44]
y=dataset.iloc[:,44]
```

In [8]:
```
x.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26013 entries, 0 to 26012
Data columns (total 44 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   REJ                          26013 non-null  uint8  
 1   RSTO                         26013 non-null  uint8  
 2   RSTOS0                       26013 non-null  uint8  
 3   RSTR                         26013 non-null  uint8  
 4   S0                           26013 non-null  uint8  
 5   S1                           26013 non-null  uint8  
 6   S2                           26013 non-null  uint8  
 7   S3                           26013 non-null  uint8  
 8   SF                           26013 non-null  uint8  
 9   SH                           26013 non-null  uint8  
 10  duration                     26013 non-null  int64  
 11  src_bytes                    26013 non-null  int64  
 12  dst_bytes                    26013 non-null  int64  
 13  wrong_fragment               26013 non-null  int64  
 14  num_failed_logins            26013 non-null  int64  
 15  logged_in                    26013 non-null  int64  
 16  lnum_compromised             26013 non-null  int64  
 17  lroot_shell                  26013 non-null  int64  
 18  lsu_attempted                26013 non-null  int64  
 19  lnum_root                    26013 non-null  int64  
 20  lnum_shells                  26013 non-null  int64  
 21  lnum_access_files            26013 non-null  int64  
 22  lnum_outbound_cmds           26013 non-null  int64  
 23  is_host_login                26013 non-null  int64  
 24  is_guest_login               26013 non-null  int64  
 25  count                        26013 non-null  int64  
 26  srv_count                    26013 non-null  int64  
 27  serror_rate                  26013 non-null  float64
 28  srv_serror_rate              26013 non-null  float64
 29  rerror_rate                  26013 non-null  float64
 30  srv_rerror_rate              26013 non-null  float64
 31  same_srv_rate                26013 non-null  float64
 32  diff_srv_rate                26013 non-null  float64
 33  srv_diff_host_rate           26013 non-null  float64
 34  dst_host_count               26013 non-null  int64  
 35  dst_host_srv_count           26013 non-null  int64  
 36  dst_host_same_srv_rate       26013 non-null  float64
 37  dst_host_diff_srv_rate       26013 non-null  float64
 38  dst_host_same_src_port_rate  26013 non-null  float64
 39  dst_host_srv_diff_host_rate  26013 non-null  float64
 40  dst_host_serror_rate         26013 non-null  float64
 41  dst_host_srv_serror_rate     26013 non-null  float64
 42  dst_host_rerror_rate         26013 non-null  float64
 43  dst_host_srv_rerror_rate     26013 non-null  float64
dtypes: float64(15), int64(19), uint8(10)
memory usage: 7.0 MB
```

In [9]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=
1)
```

In [10]:
```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.transform(x_test)
```

In [11]:
```
from sklearn.linear_model import LogisticRegression

reg=LogisticRegression()
reg.fit(x_train,y_train)
pred_logistic_reg=reg.predict(x_test)
```

```
F:\Anaconda\lib\site-packages\sklearn\linear_model\_logistic.py:940: Converge
nceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
sion
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
```

In [14]:
```
from sklearn.metrics import confusion_matrix,accuracy_score
score_logistic_reg=accuracy_score(y_test,pred_logistic_reg)*100
```

In [15]:
```
score_logistic_reg
```

Out[15]:
```
99.34648898001025
```

In [16]:
```
from sklearn.tree import DecisionTreeClassifier
classifier=DecisionTreeClassifier(criterion='entropy',random_state=0)
classifier.fit(x_train,y_train)
```

Out[16]:
```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='entropy',
                       max_depth=None, max_features=None, max_leaf_nodes=Non
e,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=0, splitter='best')
```

In [19]:
```
y_pred_decison_tree=classifier.predict(x_test)
score_decison_tree=accuracy_score(y_test,y_pred_decison_tree)*100
```

In [20]:
```
score_decison_tree
```

Out[20]:
```
99.6668375192209
```

In [21]:
```
from sklearn.ensemble import RandomForestClassifier
rfclassifier=RandomForestClassifier(n_estimators=10,criterion='entropy',random
_state=0)
rfclassifier.fit(x_train,y_train)
```

Out[21]:
```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='entropy', max_depth=None, max_features='aut
o',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=10,
                       n_jobs=None, oob_score=False, random_state=0, verbose=
0,
                       warm_start=False)
```

In [23]:
```
y_pred_random_forest=rfclassifier.predict(x_test)
score_random_forest=accuracy_score(y_test,y_pred_random_forest)*100
```

In [24]:
```
score_random_forest
```

Out[24]:
```
99.75653511019989
```

In [25]:
```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=5)
knn.fit(x_train,y_train)
```

Out[25]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                     weights='uniform')
```

In [26]:
```
y_pred_knn=knn.predict(x_test)
score_knn=accuracy_score(y_test,y_pred_knn)*100
```

In [27]:
```
score_knn
```

Out[27]:
```
99.39774474628396
```

In [28]:
```
from sklearn.naive_bayes import GaussianNB

model=GaussianNB()
model.fit(x_train,y_train)
```

Out[28]:
```
GaussianNB(priors=None, var_smoothing=1e-09)
```

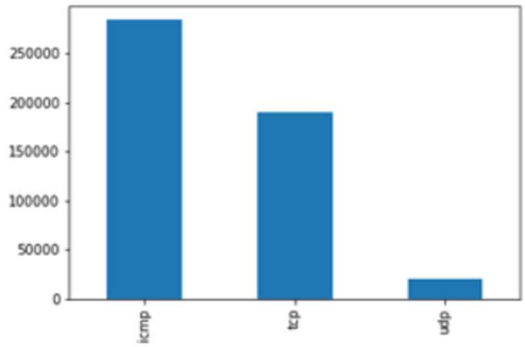
In [29]:
```
y_pred_naive_bayes=model.predict(x_test)

score_naive_bayes=accuracy_score(y_test,y_pred_naive_bayes)*100
```

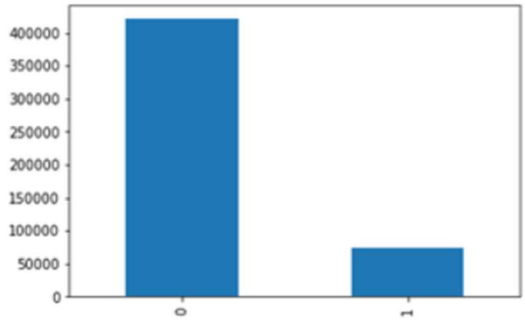
In [30]:
```
score_naive_bayes
```

Out[30]:
```
89.85135827780624
```

Visualizing Categorical Features using bar graph

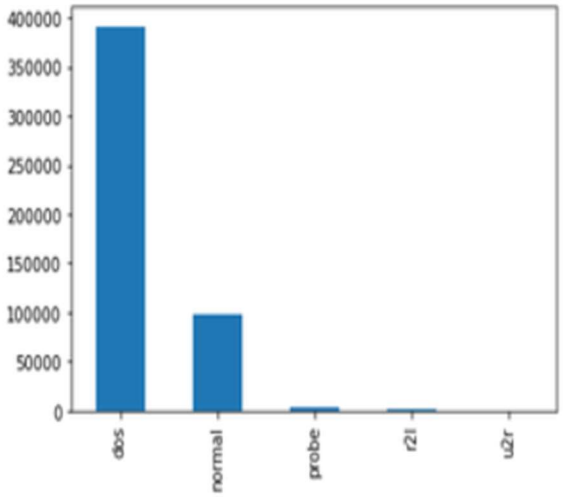


Protocol type: We notice that ICMP is the most present in the used data, then TCP and almost 20000 packets of UDP type



logged_in (1 if successfully logged in; 0 otherwise): We notice that just 70000 packets are successfully logged in.

Target Feature Distribution:



Attack Type(The attack types grouped by attack, it's what we will predict)

6.METHODOLOGY

A general AI-based NIDS methodology A NIDS developed using ML methods usually involves following three major steps as depicted in Figure 5, that is, (i) Data preprocessing phase, (ii) Training phase, and (iii) Testing phase. For all the proposed solutions, the dataset is first preprocessed to transform it into the format suitable to be used by the algorithm. This stage typically involves encoding and normalization. Sometimes, the dataset requires cleaning in terms of removing entries with missing data and duplicate entries, which is also performed during this phase. The preprocessed data is then divided randomly into two portions, the training dataset, and the testing dataset. Typically, the training dataset comprises almost 80% of the original dataset size and the remaining 20% forms testing dataset.53,54 The ML algorithm is then trained using the training dataset in the training phase. The time taken by the algorithm in learning depends upon the size of the dataset and the complexity of the proposed model. Normally, the training time for the DL models requires more training time due to its deep and complex structure. Once the model is trained, it is tested using the testing dataset and evaluated based on the predictions it made. In the case of NIDS models, the network traffic instance will be predicted to belong to either benign (normal) or attack class.

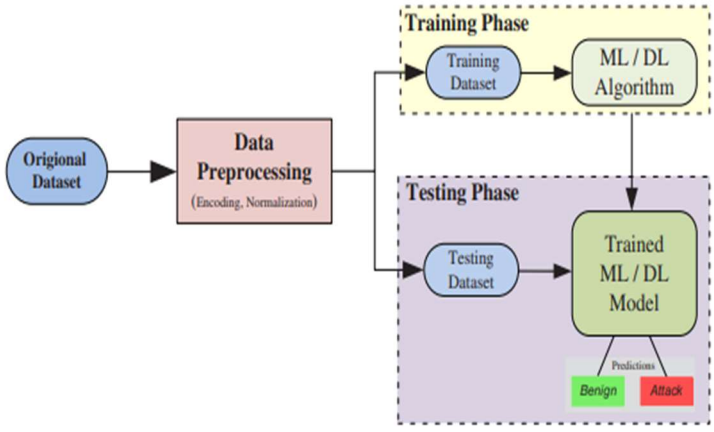


FIGURE 5 Generalized machine learning-/deep learning-based network-based intrusion detection system methodology

7.CONCLUSION AND EXPERIENCE:

- The internship was very knowledgable and interesting.
- We gained the knowledge of machine learning and its application in Real Time.
- For the above dataset Random Forest Classifier is the best algorithm because it was having higher accurcy of 99.7%
- By using machine learning we are able to predict whether there is any malicious attack or not.
- It was very helpful for us.
- The overall environment of the online class room was pleasant and joyful.
- The lab work was quite useful and helped us in gaining the inputs to build our project.
- We had a great time with our fellow members and learnt a lot from them.

8.BIBLIOGRAPHY

https://archive.ics.uci.edu/ml/machine-learning-databases/kddcup99-mld/

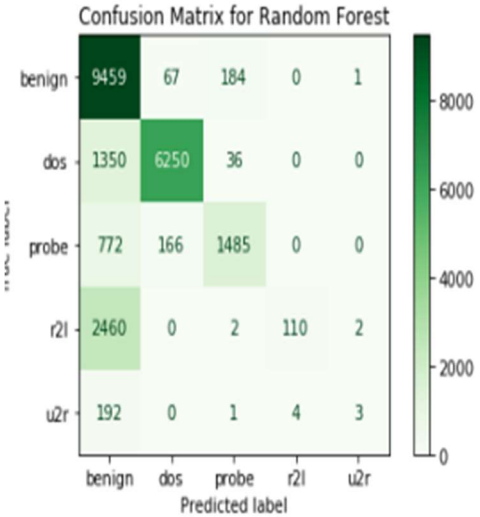
https://scikit-learn.org/stable about the machine learning techniques we used scikit learn.

www.wikipedia.com

9.1 APPENDIX-A:MACHINE LEARNING ALGORITHMS

Random Forest

- Random Forest is a supervised learning algorithm that can be used for both classification and regression. It is an easy and flexible algorithm.
- This algorithm is based on randomly selected sets of decision trees. It creates decision trees based on randomly selected data samples and gets a prediction from each tree and selects the best solution by means of voting.
- It uses the majority wins theory. In order to use this algorithm, we need to import RandomForestClassifier from the scikit-learn library.
- Scikit-learn library provides free supervised and unsupervised machine learning algorithms for python. n_estimater is an important parameter where you need to specify the number of trees in the forest. This is an optional parameter and default is 100.
- The confusion matrix is used to check the quality and performance of a model and below is the confusion matrix chart from this algorithm.

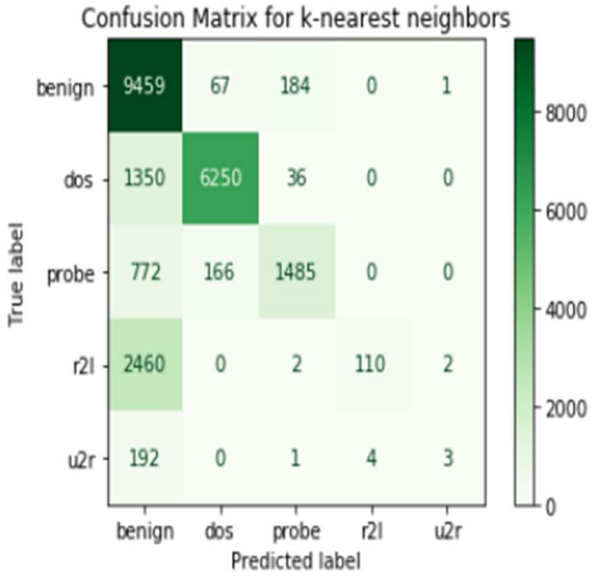


This confusion matrix of the Random Forest algorithm clearly shows that 9459 items are correctly identified as a benign class. Similarly, 6050 are correctly classified as dos, 1485 as a probe, 110 as r2l and 3 as u2r.

KNN

- k-nearest neighbors Classification (KNN) is a simple algorithm that classifies values based on similarity measures such as distance. It is used for both classification and regression predictive problems.
- However, it is mostly used in classification problems. This is simple and popular because it is easy to interpret the output. Similarly, calculation time is faster.
- It works by finding the distance between a point and data with the selected specific number closet to the point then votes for the most frequent label.
- Parameter n_neighbors is very important as it changes the accuracy rate varies. n_neighbors are a number of neighbors to use by default for neighbor's queries.
- There is no specific method to choose the best value for it. The SIT719 218191552 PRADEEP THAPA 3 default value of it is 5. One of the ways to find out better value for n_neighbors is to iterate n_neighbors value with some range (1 to 40) and check the accuracy.
- Once you know the accuracy of different values of n_neighbors, you can choose the higher accuracy value.

I have used an arbitrary value of 7 for n_neighbors and here is my confusion matrix.



This confusion matrix of the k-nearest neighbors algorithm clearly shows that 9459 items are correctly identified as a benign class. Similarly, 6250 are correctly classified as dos, 1485 as a probe, 110 as r2l and 3 as u2r.

SVM

- A Support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems.
- It is a fast and dependable classification algorithm that performs very well with a limited amount of data. The support vector machine takes a pair of (x,y) coordinates and outputs the hyperplane that best separates the tags which is the decision boundary.
- The decision boundary is anything that falls to one side of it we will classify as class A and anything that falls to other as class B. It uses kernel as a parameter where it specifies the kernel type to be used in the algorithm.
- It must be one of 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed' or a callable. The default kernel parameter value is rbf.

LOGISTIC REGRESSION

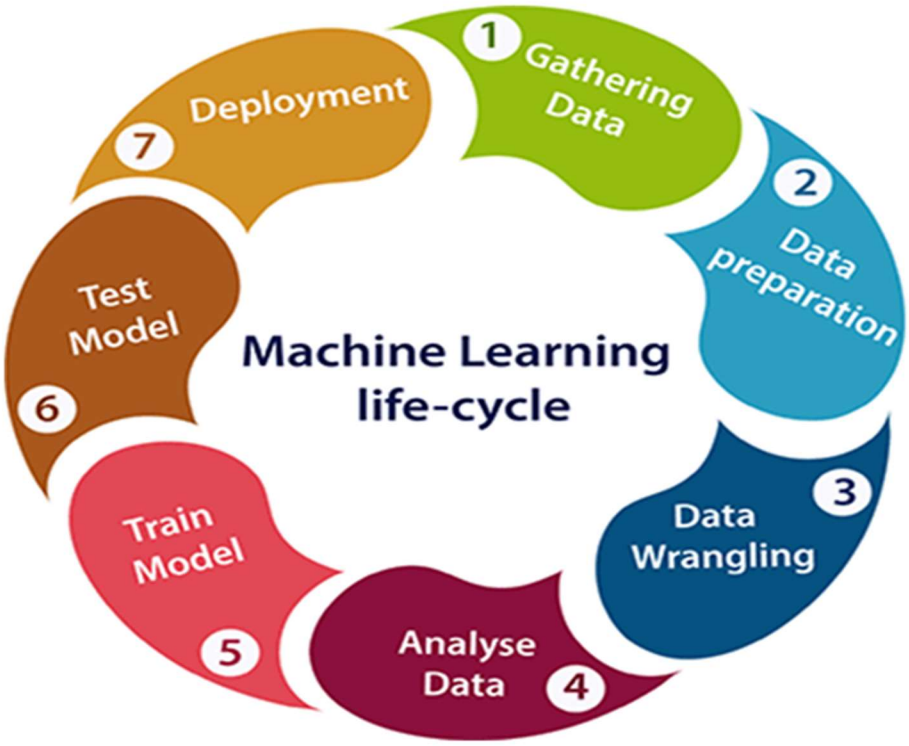
- Logistic Regression is used for classification. This algorithm determines the probability of observation to be part of a certain class or not.
- The probability is expressed with a value between 0 and 1 in which 1 means the observation is very likely to be part of that category and 0 means the observation is not the part of that category.
- When it comes to classification, we are determining the probability of observation to be part of a certain class or not. Therefore, we wish to express the probability with a value between 0 and 1. The sigmoid function is used to generate these values between 0 and 1.
- Random_state is one of the parameters that this algorithm takes which is a random number generator. The default random state is none.

9.2 APPENDIX-B
MACHINE LEARNING LIFECYCLE

Machine learning has given the computer systems the abilities to automatically learn without being explicitly programmed. But how does a machine learning system work? So, it can be described using the life cycle of machine learning. Machine learning life cycle is a cyclic process to build an efficient machine learning project. The main purpose of the life cycle is to find a solution to the problem or project.

Machine learning life cycle involves seven major steps, which are given below:

- **Gathering Data**
- **Data preparation**
- **Data Wrangling**
- **Analyse Data**
- **Train the model**
- **Test the model**
- **Deployment**



- The most important thing in the complete process is to understand the problem and to know the purpose of the problem. Therefore, before starting the life cycle, we need to understand the problem because the good result depends on the better understanding of the problem.

1. Gathering Data:

Data Gathering is the first step of the machine learning life cycle. The goal of this step is to identify and obtain all data-related problems.

In this step, we need to identify the different data sources, as data can be collected from various sources such as **files**, **database**, **internet**, or **mobile devices**. It is one of the most important steps of the life cycle. The quantity and quality of the collected data will determine the efficiency of the output. The more will be the data, the more accurate will be the prediction.

This step includes the below tasks:

- **Identify various data sources**
- **Collect data**
- **Integrate the data obtained from different sources**

By performing the above task, we get a coherent set of data, also called as a **dataset**. It will be used in further steps.

2. Data preparation

After collecting the data, we need to prepare it for further steps. Data preparation is a step where we put our data into a suitable place and prepare it to use in our machine learning training.

In this step, first, we put all data together, and then randomize the ordering of data.

This step can be further divided into two processes:

- **Data exploration:**
  It is used to understand the nature of data that we have to work with. We need to understand the characteristics, format, and quality of data.
  A better understanding of data leads to an effective outcome. In this, we find Correlations, general trends, and outliers.
- **Data pre-processing:**
  Now the next step is preprocessing of data for its analysis.

---

3. Data Wrangling

Data wrangling is the process of cleaning and converting raw data into a useable format. It is the process of cleaning the data, selecting the variable to use, and transforming the data in a proper format to make it more suitable for analysis in the next step. It is one of the most important steps of the complete process. Cleaning of data is required to address the quality issues.

It is not necessary that data we have collected is always of our use as some of the data may not be useful. In real-world applications, collected data may have various issues, including:

- **Missing Values**
- **Duplicate data**
- **Invalid data**
- **Noise**

So, we use various filtering techniques to clean the data.

It is mandatory to detect and remove the above issues because it can negatively affect the quality of the outcome.

---

4. Data Analysis

Now the cleaned and prepared data is passed on to the analysis step. This step involves:

- **Selection of analytical techniques**
- **Building models**
- **Review the result**

The aim of this step is to build a machine learning model to analyze the data using various analytical techniques and review the outcome. It starts with the determination of the type of the problems, where we select the machine learning techniques such as **Classification**, **Regression**, **Cluster analysis**, **Association**, etc. then build the model using prepared data, and evaluate the model.

Hence, in this step, we take the data and use machine learning algorithms to build the model.

---

5. Train Model

Now the next step is to train the model, in this step we train our model to improve its performance for better outcome of the problem.

We use datasets to train the model using various machine learning algorithms. Training a model is required so that it can understand the various patterns, rules, and, features.

---

6. Test Model

Once our machine learning model has been trained on a given dataset, then we test the model. In this step, we check for the accuracy of our model by providing a test dataset to it.

Testing the model determines the percentage accuracy of the model as per the requirement of project or problem.

---

7. Deployment

The last step of machine learning life cycle is deployment, where we deploy the model in the real-world system.

If the above-prepared model is producing an accurate result as per our requirement with acceptable speed, then we deploy the model in the real system. But before deploying the project, we will check whether it is improving its performance using available data or not. The deployment phase is similar to making the final report for a project.

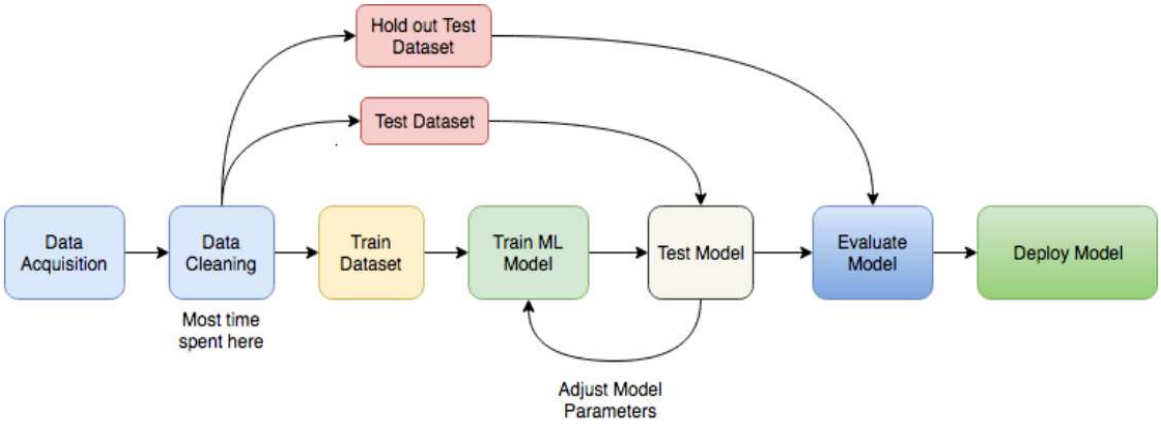
9.3 APPENDIX-B: UNIFIED MODELING LANGUAGE

The Unified Modeling Language (UML) is a general-purpose visual modeling language that is used to specify, visualize, construct, and document the artifacts of a software system.

It captures decisions and understanding about systems that must be constructed. It is used to understand, design, browse, configure, maintain, and control information about such systems.

It is intended for use with all development methods, lifecycle stages, application domains, and media.

The modeling language is intended to unify past experience about modeling techniques and to incorporate current software best practices into a standard approach.



UML includes semantic concepts, notation, and guidelines. It has static, dynamic, environmental, and organizational parts. It is intended to be supported by interactive visual modeling tools that have code generators and report writers. The UML specification does not define a standard process but is intended to be useful with an iterative development process. It is intended to support most existing object-oriented development processes.

The UML captures information about the static structure and dynamic behavior of a system. A system is modeled as a collection of discrete objects that interact to perform work that ultimately benefits an outside user. The static structure defines the kinds of objects important to a system and to its implementation, as well as the relationships among the objects. The dynamic behavior defines the history of objects over time and the communications among objects to accomplish goals.

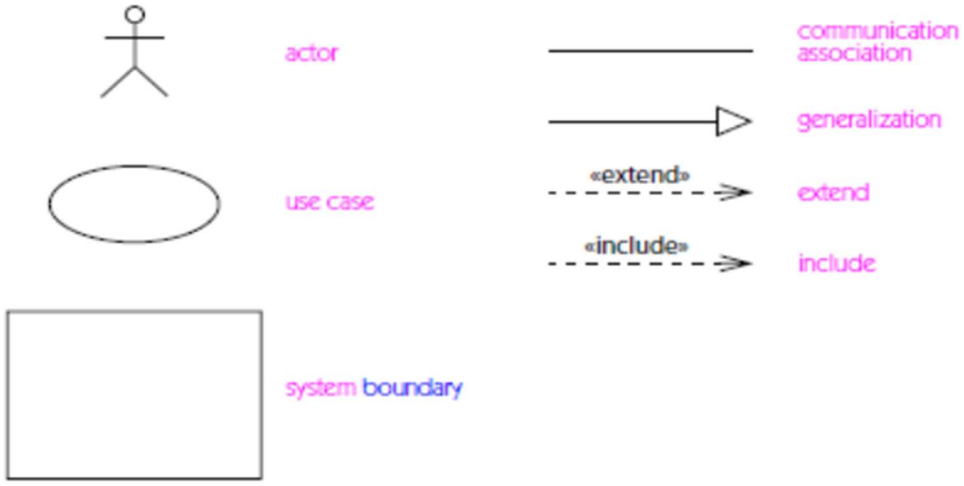
Modeling a system from several separate but related viewpoints permits it to be understood for different purposes.

The UML also contains organizational constructs for arranging models into packages that permit software teams to partition large systems into workable pieces, to understand and control dependencies among the packages, and to manage the versioning of model units in a complex development environment. It contains constructs for representing implementation decisions and for organizing run-time elements into components.

UML is not a programming language. Tools can provide code generators from UML into a variety of programming languages, as well as construct reverse engineered models from existing programs. The UML is not a highly formal language intended for theorem proving. There are a number of such languages, but they are not easy to understand or to use for most purposes. The UML is a general-purpose modeling language. For specialized domains, such as GUI layout, VLSI circuit design, or rule-based artificial intelligence, a more specialized tool with a special language might be appropriate. UML is a discrete modeling language.

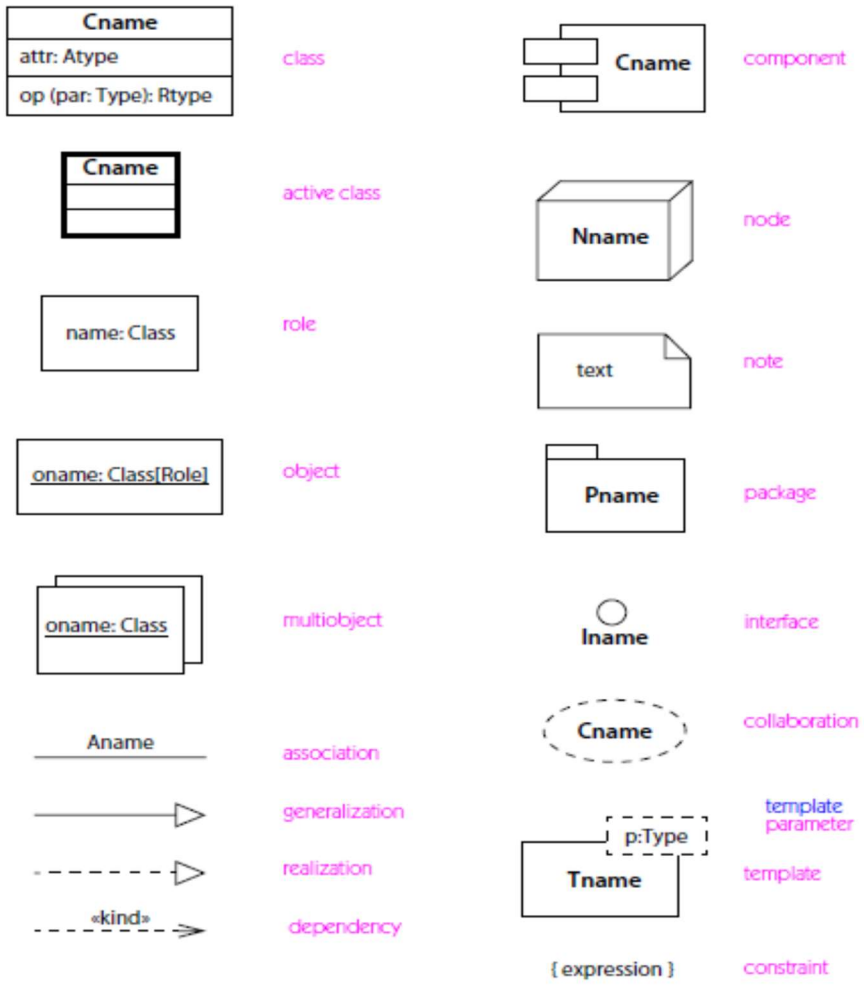
It is not intended to model continuous systems such as those found in engineering and physics. UML is intended to be a universal general-purpose modeling language for discrete systems such as those made of software, firmware, or digital logic.

Icons on use case diagrams

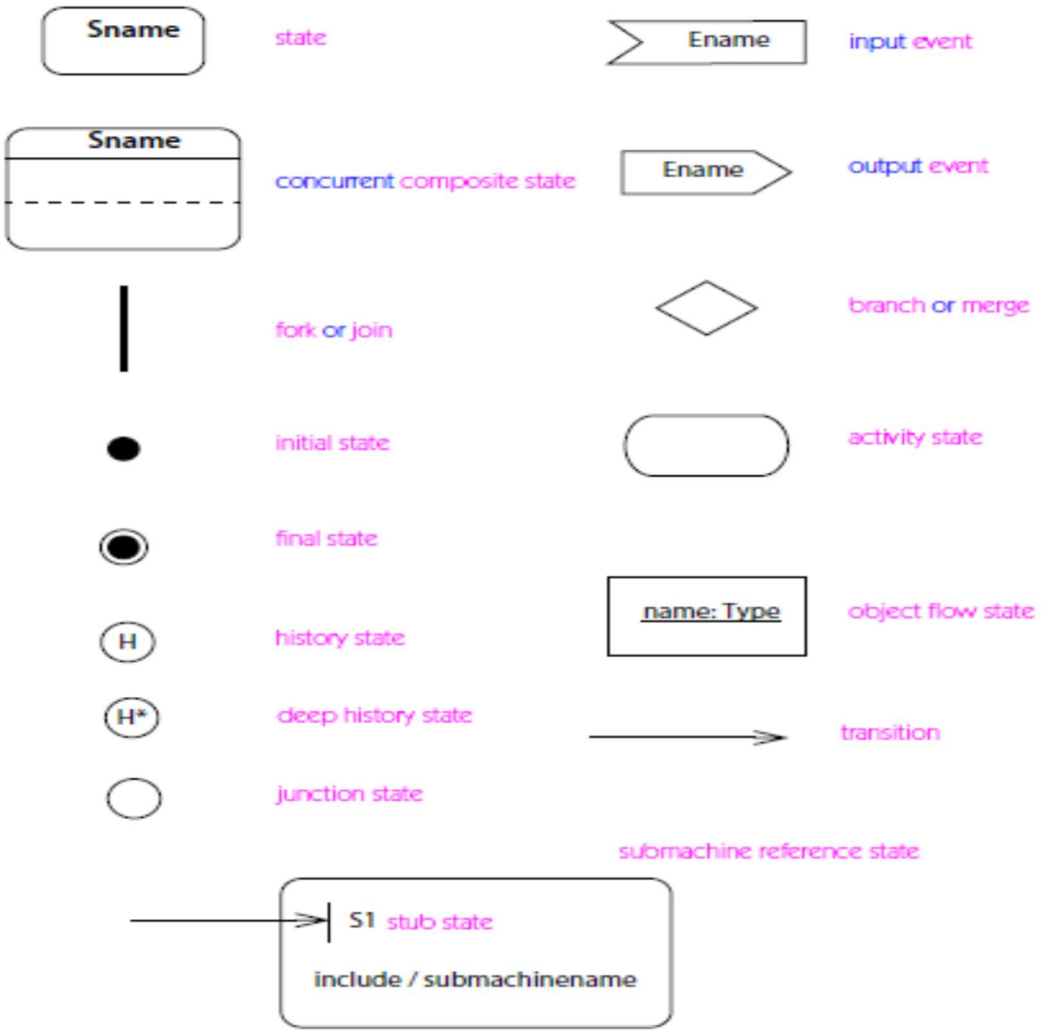


Icons on class, component, deployment, and collaboration diagrams

Icons on class, component, deployment, and collaboration diagrams



Icons on statechart and activity diagrams



9.3 APPENDIX-C : EVALUATION METRICS

This section explains the most commonly used evaluation metrics for measuring the performance of ML and DL methods for IDS. All the evaluation metrics are based on the different attributes used in the Confusion Matrix, which is a two-dimensional matrix providing information about the Actual and Predicted class128 and includes;

i. True Positive (TP): The data instances correctly predicted as an Attack by the classifier.
ii. ii. False Negative (FN): The data instances wrongly predicted as Normal instances.
iii. iii. False Positive (FP): The data instances wrongly classified as an Attack.
iv. iv. True Negative (TN): The instances correctly classified as Normal instances.

The diagonal of confusion matrix denotes the correct predictions while nondiagonal elements are the wrong predictions of a certain classifier. Table 5 depicts these attributes of confusion matrix. Further, the different evaluation metrics used in the recent studies are

• Precision: It is the ratio of correctly predicted Attacks to all the samples predicted as Attacks.

•Precision = TP /(TP + FP.)

Recall: It is a ratio of all samples correctly classified as Attacks to all the samples that are actually Attacks. It is also called a Detection Rate.

Recall = Detection Rate = TP /(TP + FN).

• False alarm rate: It is also called the false positive rate and is defined as the ratio of wrongly predicted Attack samples to all the samples that are Normal.

False Alarm Rate = FP /(FP + TN).

• True negative rate: It is defined as the ratio of the number of correctly classified Normal samples to all the samples that are Normal.

True Negative Rate = TN /(TN + FP).

• Accuracy: It is the ratio of correctly classified instances to the total number of instances. It is also called as Detection Accuracy and is a useful performance measure only when a dataset is balanced.
Accuracy = TP + TN/(TP + TN + FP + FN).

TABLE 5 Confusion matrix

		Predicted class	
		Attack	**Normal**
Actual Class	**Attack**	True Positive	False Negative
	Normal	False Positive	True Negative

Team No: 17		Title	Guide Name
Roll No	Name		
18311A05T0	G.NAVEEN	CLASSIFICATION OF NETWORK USING MACHINE LEARNING	Mr.A.CHANDU NAIK
18311A05R4	CH.UDAY		
18311A05W8	V.ROHITH		

ABSTRACT

In our society, information systems are everywhere. They are used by corporations to store proprietary and other sensitive data, by families to store financial and personal information, by universities to keep research data and ideas, and by governments to store defense and security information.

It is very important that the information systems that house this vitally sensitive information be secure. In order for information systems to be secure, it is paramount that they utilize robust security mechanisms.

Commonly found security mechanisms are passwords on accounts, encryption of sensitive data, virus protection, and intrusion detection. An Intrusion Detection System (IDS) monitors activity at an access point and can log or prevent activities that are marked as intrusions. Intrusions occur when malicious activity gains access to or affects the usability of a computer resource.

The goal of this project is two-fold. First, we attempt to find the most effective machine learning model for identifying network attacks.

Although scalability and performance are major considerations in every commercial product, our results are targeted at minimizing false positives and negatives

	Guide	HOD
Student 1- G.NAVEEN	**Mr.A.CHANDU NAIK**	**DR.ARUNA VARANASI**
Student 2- CH.UDAY	**Assistant Professor**	**Professor & HOD**
Student 3- V.ROHITH	**Department of CSE**	**Department of CSE**

Team No: 17		Title	Guide Name
Roll No	Name		
18311A05T0	G.NAVEEN	CLASSIFICATION OF NETWORK TRAFFIC USING MCHINE LEARNING	MR.A.CHANDU NAIK
18311A05R4	CH.UDAY		
18311A05W8	V.ROHITH		

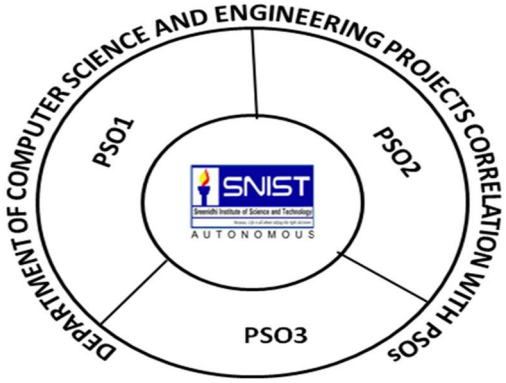
Batch No.	Roll No.	Product/app	Ethics	research	social science	safety
E-19	18311A05T0					✓
	18311A05R4					
	18311A05W8					

SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Projects Correlation with POs

PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
M	M	M	M	H	M	M	L	H	M	M	H



H **High**
M **Moderate**
L **Low**

GUIDE:MR.A.CHANDU NAIK
ASSISTANT PROFFESSOR
Department of computer science

HOD
DR.ARUNA VARANASI
PROFFESOR&HOD
Department of computer science