MALWARE ANALYSIS PROJECT

NETWORK INTRUSION DETECTION

(Machine Learning Based Approach)

Submitted By:

Ajay Sharma 2K18/CO/035

Akshansh Rewariya 2K18/CO/042

CONTENTS

INTRODUCTION

One of the most vital parts of system or network administration is to keep the network safe from intrusion. It can lead to a great loss for a company if it is penetrated by a malicious attacker like data breach, potential downtime, and customer trust loss.

1.1. What Is Intrusion Detection System (IDS)

A software or device that is used for monitoring systems or networks for activities that may be malicious and threatening or some violation of policy , is called an **Intrusion Detection System**. These violations or intrusion activities are reported to an administrator.

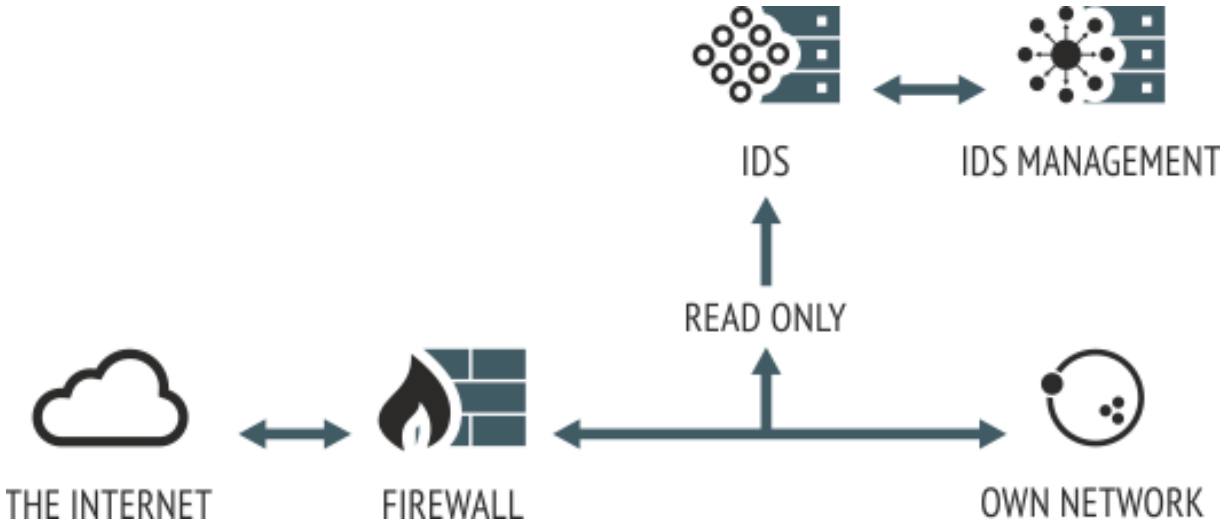


Fig.1: Intrusion Detection System

The IDS consists of the following three components:

- **Sensors** are used to track down the networking traffic or behavior and event triggering.
- **Consoles** are used to check event warnings and to control the sensors,
- **Detection Engine** tracks the events logged by the sensors in the database and sends notifications from the collected event details.

1.2. Types of IDS

These can be classified mainly into two types-

1. Network based Intrusion Detection

Any unauthorized activity on a computer network is termed as network intrusion , a software to detect these unwanted intrusions and to protect the network from unauthorized users. It keeps a check of the flow of traffic inside the network, a warning is also flagged.

A few preferences include:

- Brought into a network that is being targeted without effort and with minimal inconvenience.
- Perhaps imperceptible by assailants and are generally resistant to direct assaults.

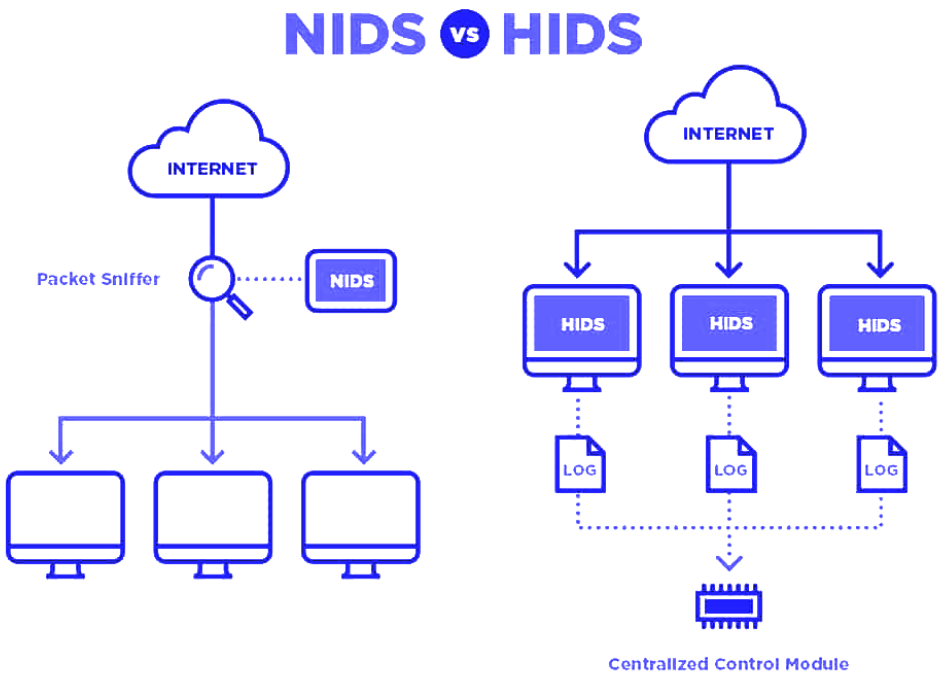


Fig.2

2. Host based Intrusion Detection

It is used to monitor and look into the details about harmful activities on the system of a remote individual. A warning may be triggered if it sees any malignant change in the system. It detects intrusion by keeping an eye on any suspicious change in files in OS in various places.

Features	HIDS	NIDS
Management	Harder to manage due to the heterogeneity of the environment and its high number in large networks with many hosts	Simple to manage due to its homogeneity and a few NIDS are sufficient to monitor a large network with many hosts
Analyse encrypted network traffic	YES	NO
IDS evasion techniques	Harder to perform than on NIDS[6]	Evasion techniques like fragmentation will easily work with NIDS when they have no possibility to reconstruct locally the fragmented network packets
Knows if an attack was successful or not on a host	YES	NO
Protection against targeted attacks	Can be disabled during the attack of a host or by specific denial-of-service attacks	Easier than HIDS to protect against targeted attacks and can run in stealth mode
Detects large network attacks	NO	YES
Uses computing resources of the monitored host	YES	NO

Fig. 3 : Advantages and Disadvantages of NIDS and HIDS

1.3. Importance of Network Intrusion Detection Systems

In order to ensure reliable and trusted knowledge exchange between different organisations a major call for great level security and protection is there for a healthy working network .With the evolving science and technology, attacks like Cyber breaches are to be tackled with greater severity. An intrusion detection system fills in the gap with secure protection of these networks and makes our lives a lot easier.

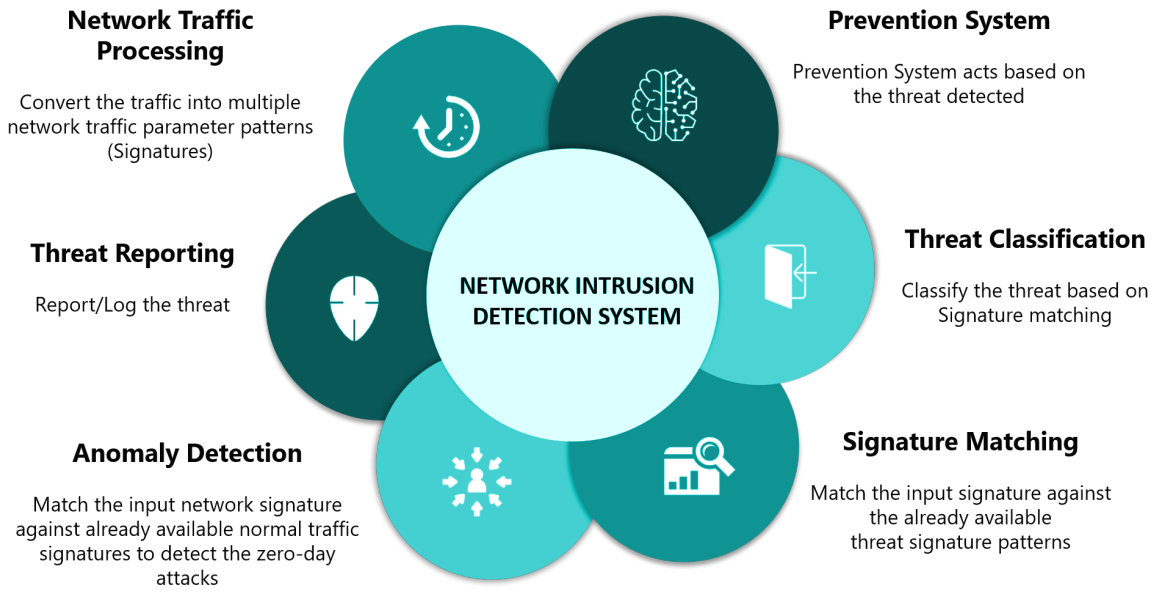


Fig. 4 : Network Intrusion Detection System

1.4. Types of Attacks

1. Denial-of-Service (DOS)

Majorly a type of cyber attack where a harmful actor means to shut down a network by manipulating the working of the network, leading it inaccessible to its intended users. They are carried out by over-flooding the target's network until the system's ability to control and handle it ,is neutralised. Ultimately the system denies the service due to overload.

Attack type	Service	Mechanism	Effect of the attack
Apache2	http	Abuse	Crashes httpd
Back	http	Abuse/ Bug	Slows down server response
Land	http	Bug	Freezes the machine
Mail bomb	N/A	Abuse	Annoyance

Fig.5 : Some examples of DoS

There is a kind of DoS attack that comes from many circulated sources making it way more difficult to handle , called distributed denial of service ., like , a botnet DDoS attack.

Indications for this attack can be

- A slow network performance,like, long loading time for documents or sites
- //failure to stack a specific site, for example, your web property
- loss of availability of connection across devices that are on a similar network

2. Remote to Local attack (R2L)

In this, a network is assaulted by sending packets over an organization network, at that point machines' weaknesses are exploited to wrongfully increase nearby access as a user.

Attack type	Service	Mechanism	Effect of the attack
Dictionary	Telnet, rlogin, pop, ftp, imap	Abuse feature	Gains user access
Ftp-write	FTP	Misconfig	Gains user access
Guest	Telnet, rlogin	Misconfig	Gains users access
Imap	Imap	Bug	Gains root access

Fig.6 : Some examples of R2L

3. User to Root attack (U2R)

In the beginning of the attack , the hacker gets admitted to the user account by illegally accessing it and further grants the root privileges to them by legally accessing the machine and hence making it hard to be found

Attack type	Service	Mechanism	Effect of the attack
Eject	User session	Buffer overflow	Gains root shell
Ffbconfig	User session	Buffer overflow	Gains root shell
Fdformat	User session	Buffer overflow	Gains root shell
Loadmodule	User session	Poor environment sanitation	Gains root shell

Fig.7 : Some examples of U2R

4. Probe attack

The hacker examines a networking device in order to accumulate data or find known weaknesses of the machines. There are various kinds of tests attacks including: some of them misuse the device's authentic features, and sometimes the machine's weaknesses are used..

Msacn and saint looks…

Nmap

Attack type	Service	Mechanism	Effect of the attack
Ipsweep	ICMP	Abuse of feature	Identifies active machines
Mscan	Many	Abuse of feature	Looks for known vulnerabilities
Nmap	Many	Abuse of feature	Identifies active ports on a machine
Saint	Many	Abuse of feature	Looks for known vulnerabilities

Fig.8 : Some examples of Probing

1.5. Motivation

With the Internet being such an integral part of our daily lives, security issues such as internet security and connectivity continue to arise.Intrusion and denial-of-service threats are the major threat to information security and availability.An intrusion detection system with a 100 percent success rate is difficult or virtually impossible to build. Today, most devices have a lot of security bugs.Hackers are now using artificial learning methods to discover fresh paths into the

networks. To recognise potential intruders, quick identification of these attacks would help & the implementation of an appropriate and precise intrusion detection system would also benefit.

1.6. Objective

The key reason behind conducting malware analysis is to extract data from the sample of malware, which will help to respond to a malware event. The aim of malware research is to assess, identify, and contain the potential risks of a malware.

Our purpose of this project would be to develop a network intrusion system, a predictive model which is able to distinguish and tell between good(normal) or bad (network intrusions) connections.

For the aforementioned objective, we will be using KDDCUP-99 dataset for training slew of machine learning models to get better understanding of network intrusions.

1.7. Problem Statement

In this project we will study network intrusion by using machine learning classifier models. We will analyse our data and apply various models on them to gain better understanding. We will use following models:

- Decision Tree
- Random Forest
- Gaussian Naive Bayes
- K-Nearest Neighbour

Accuracy for all the models will be calculated and will help in predicting whether connections are malicious or not.

LITERATURE SUMMARY

2.1. Related Research to Solve the Problem

In the field of network intrusion detection , ML methodologies and AI philosophies are in effect broadly utilized by the analysts and the researchers because of their speculation capacities that

assists with understanding the specialized technical information about the interruptions that don't have any predefined or reoccurring designs. To detect network intrusions, many studies have utilized rules of association [1,10].

Neural networks have been widely used in network intrusion to recognise both patterns of anomaly and misuse. The problem with neural networks is that training in a broad dataset like KDDCup 1999 takes a lot of time.[2,3]

Decision trees and Naïve Bayes are widely used methodologies in IDS.Because of its simplicity, quick adaptability and most notably,to achieve high classification accuracy, Decision Trees are used.At the same time, instead of similar characteristics that could degrade its efficiency, Naïve Bayes assumes the conditional independence of the data function.[4,5,6]

A study conducted by Nguyen and Choi [7] suggested a classification model by applying 10 widely used classifier algorithms.

In misuse, anomaly and hybrid identification, Zhang et al.[8] implemented the random forest algorithm, where they observed that their proposed hybrid system achieves a higher detection rate with a low false positive rate than the others.

Tavallaee et al.[9] recently reviewed the existing state of scientific practice in the IDS region and offered an overview of their observation by outlining the common pitfalls of 267 experiments in the survey.

DATASET

The KDD-99 dataset is used in our project. The dataset being very detailed and humongous, we chose the skimmed 10% dataset of the original to work on. This dataset was prepared by Massachusetts Institute of Technology Lincoln Lab. Its original purpose was to study Network Intrusion.

Dataset Partitioning:

Partitioning of the dataset was done in a ratio of 80:20 for training and testing. Training dataset contained 116469 samples and Testing dataset contained 29117 samples.

Analysing Dataset:

Simple analysis of dataset shows the outcome which is either good (normal) or bad (malicious)

KDD-99 Dataset

KDD-99 dataset is a benchmark for ML based intrusion detection, however , it suffers from several inefficiencies. These include:

- Class imbalance
- Under representation of widely used attack types such as u2r and r2l

We have classified all the intrusions into classes so as to get a better understanding of what malware is causing more harm.

The intrusions studied in this dataset are:

PROBE	*R2L*	*U2R*	*DOS*
ipsweep.	*ftp_write.*	*buffer_overflow.*	*back.*
portsweep.	*guess_passwd.*	*loadmodule.*	*land.*
nmap.	*imap.*	*perl.*	*neptune.*
satan.	*multihop.*	*rootkit.*	*pod.*
	phf.		*smurf.*
	spy.		*teardrop.*
	warezclient.		
	warezmaster.		

METHODOLOGY

Data Cleaning:

The dataset used in this analysis (KDD-99) is very detailed and feature rich. Still it suffers from many inconsistencies. So it was imperative for us to do the data cleaning process on it.

For the purpose of this experiment, we have chosen and skimmed 10% KDD-99 dataset. We inspected the dataset for the inconsistencies.

Further, for better model training and testing we removed null values and duplicates from our dataset. Luckily, no null values were found.

Below is the code snippet of our data cleaning:

```
[ ]  da.drop_duplicates(subset=features, keep='first', inplace=True)
     print('No. of null values: ')
     num_null=len(da[da.isnull().any(1)])
     print(num_null)
     da.shape

     No. of null values:
     0
     (145586, 42)
```

Feature Extraction:

Function extraction is a dimensionality reduction mechanism by which an initial collection of raw data is condensed for processing to more manageable classes. The method of extracting features is effective if the amount of resources available for processing is minimized without losing substantial information. The extraction of features may also reduce the redundancy for given analysis.

In the dataset, rows with redundant and ineffective data were chosen to be dropped since they served no use to our analysis.

Transformation of categorical values:

Transforming a Categorical Variable turns a categorical variable into a numeric one. Many machine learning models only handle numeric values so they categorical variables must be transformed

```
[ ]  da['protocol_type'] = da['protocol_type'].astype('category')
     da['service'] = da['service'].astype('category')
     da['flag'] = da['flag'].astype('category')
     cc = da.select_dtypes(['category']).columns
     da[cc] = da[cc].apply(lambda x: x.cat.codes)
```

Many categorical variables were found in our dataset. So transformation of categorical variables was of prime concern. This is shown in the above figure.

Standardization:

We then followed by Standardization of Data. It is done before feeding data from our dataset to our Machine Learning models. To gain the gist of it: Standardization leads to a convenient, same and singular format in which data is assembled.

```
[ ]  stan_scaler = StandardScaler()
     reX = stan_scaler.fit_transform(X)
     names_inputed =features[0:39]
     da = pd.DataFrame(data=reX, columns=names_inputed)
```

Above figure shows our data standardization.

Normalization:

We applied normalization to our dataset. Below is a snippet of it:

```
[ ]  n_fun = Normalizer()
     X = n_fun.fit_transform(X)
```

After data processing, we trained and tested our Machine Learning models on the dataset.

For our project on this topic, following machine learning algorithms applied on this subset will be:

- **Decision tree**

  The Decision Tree is a classification machine learning model. Analysing the data with simple rules is used to predict the class of given object.

  Here we have selected Gini as the best suited attribute to split the records. Further we have adjusted class weights according to the proportions of each class frequencies. Following is the code snippet:

  ```
  [ ]  kfv(DecisionTreeClassifier(criterion='gini',class_weight='balanced'))
       acc_des_tree = res_project['acc']/5
  ```

- **Random forest**

Random Forest is the supervised learning technique. We have used a Random forest classifier with 100 trees as the optimal number for best performance and precise results.

```
kfold_validation(RandomForestClassifier(n_estimators=100))
acc_random_tree = res_project['acc'] /
```

- **Gaussian Naive Bayes**

  A Naive Bayes classification model is a probabilistic classification model.It is a classifier in a machine learning algorithm that is used to differentiate between related objects based on certain attributes.

```
[ ] kfold_validation(GaussianNB())
    acc_naive_bayes = res_project['acc'] / 5
```

- **KNN**

  The K-Nearest Neighbors (KNN) is another algorithm we will use in this project.
  It is used for predictive problems of classification and regression purposes. However it is primarily used in industry for predictive classification problems.
  For using the KNN model, we have chosen K = 2 which was found to be the best case for this approach.

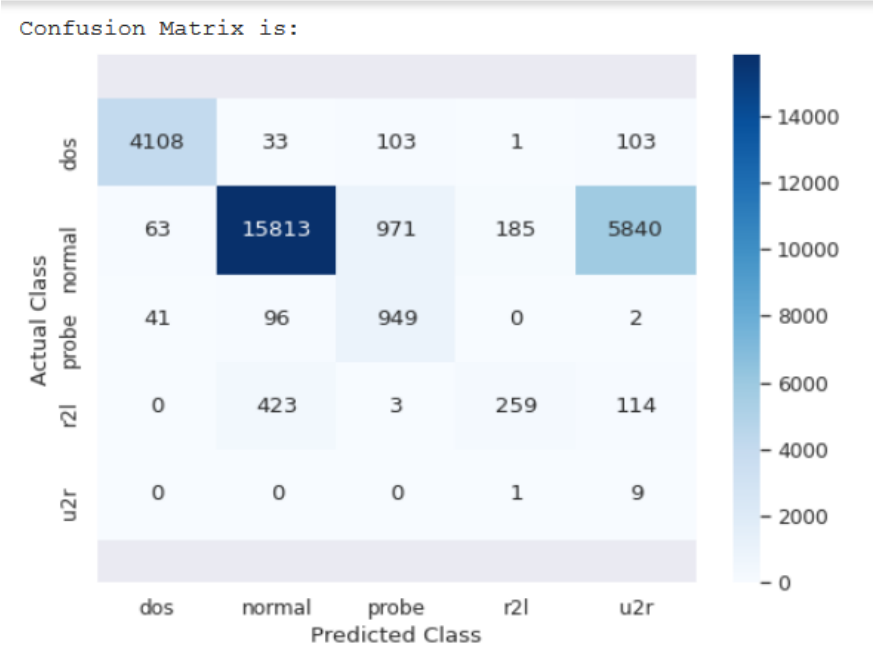
```
[ ] kfold_validation(KNeighborsClassifier(3))
    acc_knn = res_project['acc'] / 5
```

RESULT ANALYSIS

We conducted our experiments on the KDD-99 dataset by dividing it into a ratio of 80:20. 80% of the dataset was used for the training and rest for the testing. Further, K-fold Validation (K=5) is performed to estimate the skill of machine learning models.

Following are the results of our analysis:

1. **Gaussian Naive Bayes Model** :




```
Accuracy score is:
0.6171498249376537
==================================================
Precision score is:
0.9664085999165005
==================================================
Recall score is:
0.6171498249376537
==================================================
F1-score is:
0.7102650962788861
```

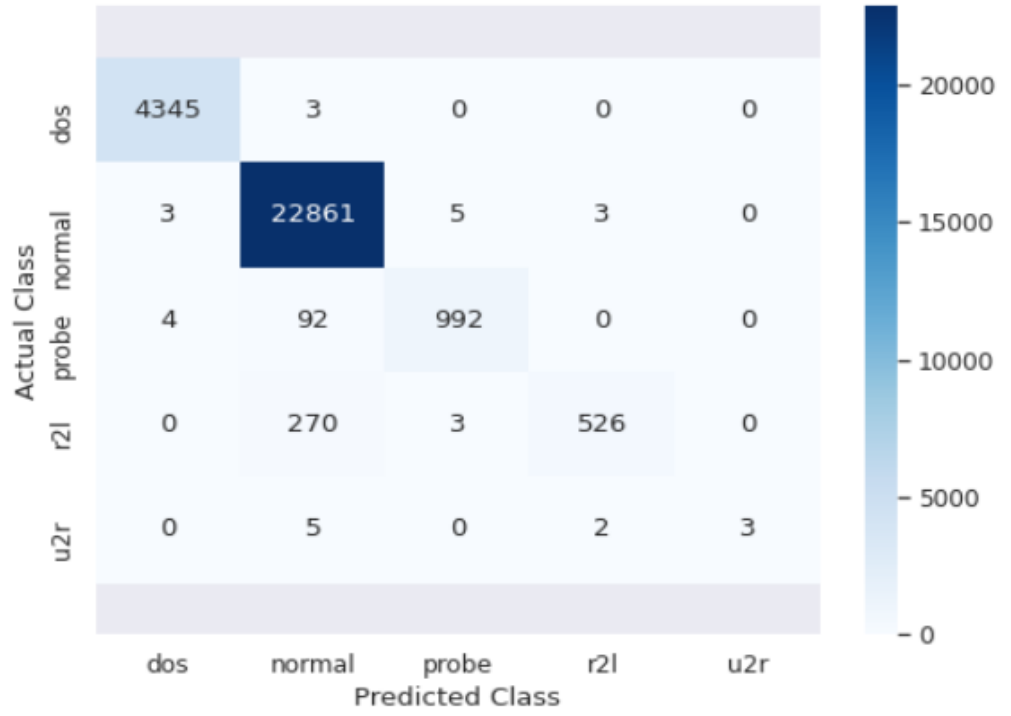
2. **Decision Tree Model:**

Confusion Matrix is:


```
Accuracy score is:
0.9204726202709372
================================================
Precision score is:
0.9828689316716357
================================================
Recall score is:
0.9204726202709372
================================================
F1-score is:
0.9326618470942462
```

3. **Random Forest Model:**

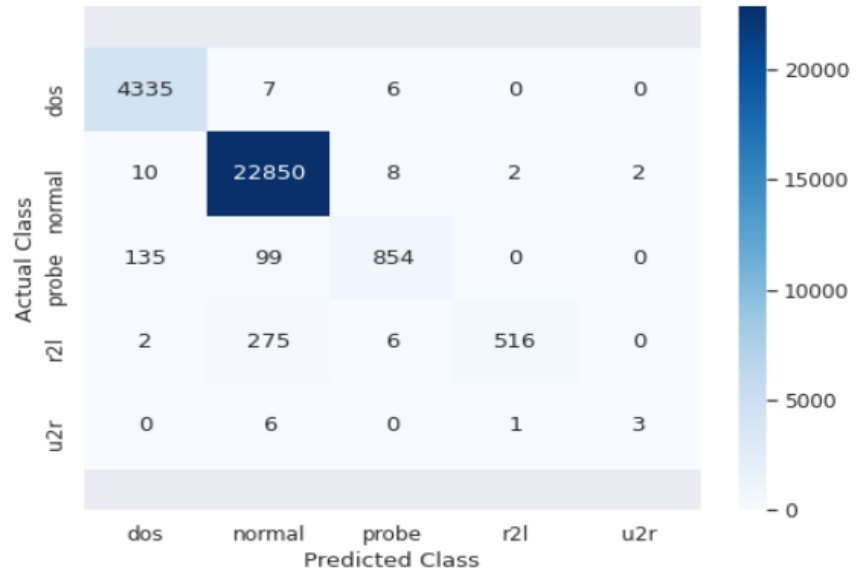
Confusion Matrix is:



Accuracy score is:
0.9275887292746692
==========
Precision score is:
0.9933370544464408
==========
Recall score is:
0.9275887292746692
==========
F1-score is:
0.9420921892655911

4. **K Nearest Neighbour Model (K=3):**

```
Confusion Matrix is:
```



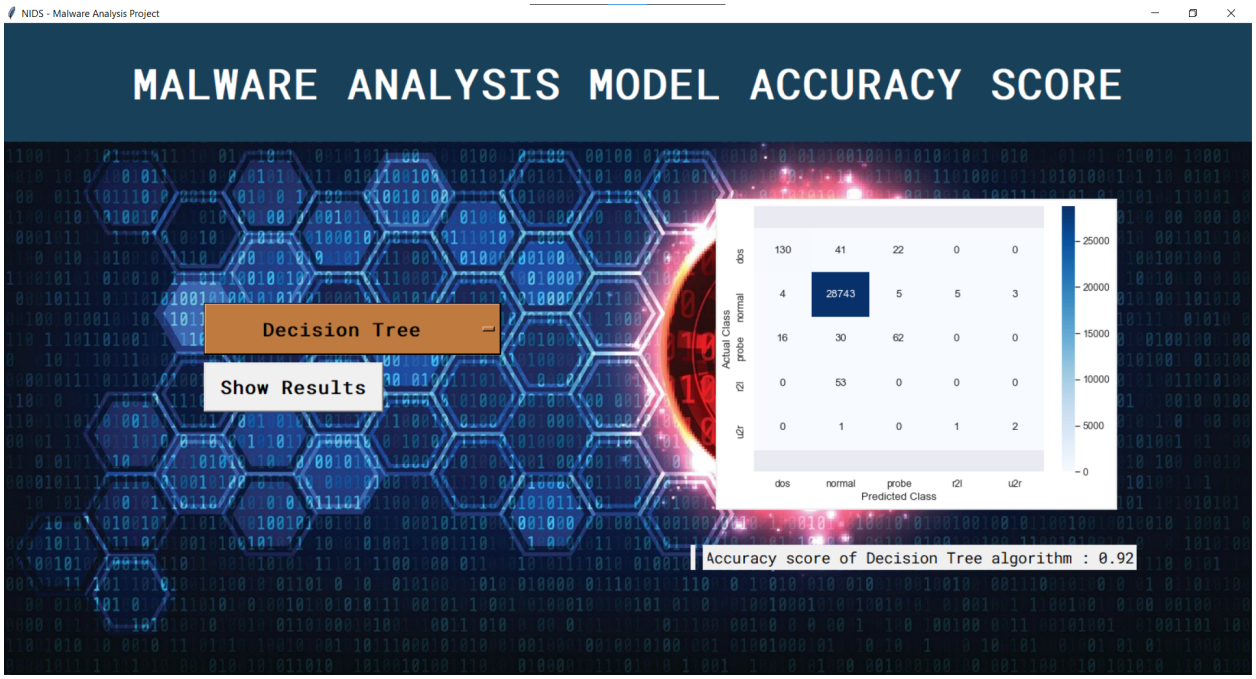

```
==================================================

Accuracy score is:
0.9802521291073001
==================================================
Precision score is:
0.9890022320956418
==================================================
Recall score is:
0.9802521291073001
==================================================
F1-score is:
0.9835317710753337
```

Testing Accuracy achieved by our models:

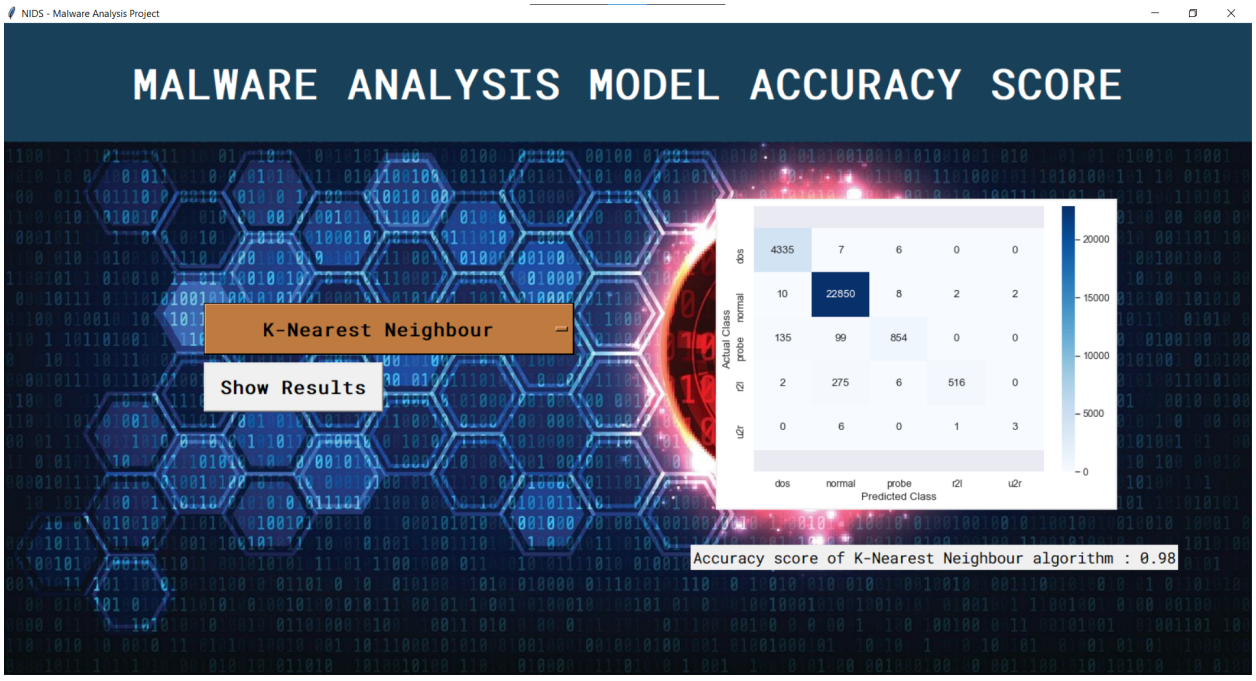
1. **Gaussian Naive Bayes Model** : 61.71%
2. **Decision Tree Model** : 92.04%
3. **Random Forest Model** : 92.75%
4. **K-Nearest Neighbour Model** : 98.02%

K-Nearest Neighbour model was found to be the best performing model with the **accuracy score of 98.02%.**

IMPLEMENTATION



Above figure shows our GUI implementation of the project. This GUI shows the model's accuracy score.



CONCLUSION

Successful Experiments were performed to analyse our dataset using machine learning models. The Models used in this project are:

1. Gaussian Naive Bayes Model.
2. Decision Tree Model.
3. Random Forest Model.
4. K-Nearest Neighbour Model.

The best performing model of the aforementioned list was K Nearest Neighbour with the accuracy score of **98.02%.**

REFERENCES

1) https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.138.8956&rep=rep1&type=pdf
2) https://www.researchgate.net/publication/286697952_Neural_Networks_for_Intrusion_Detection_and_Its_Applications
3) https://www.researchgate.net/publication/3950039_Intrusion_detection_using_neural_networks_and_support_vector_machines
4) https://www.researchgate.net/publication/220998586_Naive_Bayes_vs_decision_trees_in_intrusion_detection_systems
5) https://www.researchgate.net/publication/4325362_Effective_Value_of_Decision_Tree_with_KDD_99_Intrusion_Detection_Datasets_for_Intrusion_Detection_System
6) https://www.semanticscholar.org/paper/NETWORK-INTRUSION-DETECTION-USING-NA%C3%8FVE-BAYES-Panda-Patra/1a5c191da4aa733c80311ef4057c16dc899819cd
7) https://arxiv.org/pdf/1007.1268
8) https://www.researchgate.net/publication/3421958_Random-Forests-Based_Network_Intrusion_Detection_Systems
9) https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.379.6674&rep=rep1&type=pdf
10) https://www.researchgate.net/publication/2452599_A_Framework_for_Constructing_Features_and_Models_for_Intrusion_Detection_Systems