A

Major Project report

on

NETWORK INTRUSION DETECTION

submitted to the Jawaharlal Nehru Technological University Hyderabad in

Partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology
in

Computer Science and Engineering

Submitted By

SHAIK SABIHA NAZREEN	**(17R91A05D9)**
RAMYA NADUKUDA	**(17R91A05D2)**
PRASHANTH TUMMALA	**(17R91A05E5)**
HARSHITH RAJ UPPULA	**(17R91A05E6)**

Under the Esteemed Guidance

Of

A.SRINIVASA REDDY

Asst.Professor

Department of Computer Science and Engineering

TEEGALA KRISHNA REDDY ENGINEERING COLLEGE

(Affiliated to JNTUH, Hyderabad, Approved by AICTE, Accredited by NBA & NAAC with 'A' Grade)

Medbowli, Meerpet, Saroornagar, Hyderabad– 500097.

2017-2021

TEEGALA KRISHNA REDDY ENGINEERING COLLEGE

(Sponsored by TKR Educational Society)

(Approved by AICTE, New Delhi and Affiliated to JNTUH, Accredited by NBA & NAAC 'A' Grade)

Medbowli, Meerpet (V), Balapur (M), Ranga Reddy (D), Telangana State – 500097

Mobile: 8498085218, E-mail: info@tkrec.ac.in, Website: www.tkrec.ac.in

College Code: R9

CERTIFICATE

This is to certify that the project report titled "**NETWORK INTRUSION DETECTION**" is being submitted by SHAIK SABIHA NAZAREEN, RAMYA NADUKUDA, HARSHITH RAJ UPPULA, PRASHANTH TUMMALA in IV B.Tech II semester Computer Science and Engineering is a record bonafide work carried out by them. The results embodied in this report have not been submitted to any other University for the award of any degree.

INTERNAL GUIDE

G.RANI

(Asst. Professor)

HEAD OF DEPARTMENT

Dr. Ch. V. PHANIKRISHNA

(Professor)

EXTERNAL

PRINCIPAL

Dr. K. VENKATA MURALI MOHAN

TEEGALA KRISHNA REDDY ENGINEERING COLLEGE
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project report titled "**Network Intrusion Detection"** is being submitted by **Shaik Sabiha Nazreen,Ramya Nadukuda,Harshith Raj Uppula,Prashanth Tummala** in IV B.Tech II semester Computer Science and Engineering is a record bonafide work carried out by them. The results embodied in this report have not been submitted to any other University for the award of any degree.

INTERNAL GUIDE
A.SRINIVASA REDDY
(Asst. Professor)

HEAD OF THE DEPARTMENT
Dr.CH. V. PHANI KRISHNA
(Professor)

EXTERNAL EXAMINAR

PRINCIPAL
Dr. K. VENKATA MURALI MOHAN

PROJECT EVALUATION CERTIFICATE

This is to certify that the project report titled "**Network Intrusion Detection"** is being submitted by **Shaik Sabiha Nazreen,Ramya Nadukuda,Harshith Raj Uppula,Prashanth Tummala** has been examined and adjudged as sufficient for the partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering of Jawaharlal Nehru Technological University, Hyderabad.

Internal Examiner : _____

(Signature with Date)

External Examiner : _____

(Signature with Date)

DECLARATION

We hereby declare that the project entitled **"Network Intrusion Detection"** submitted to the **JNTUH** in partial fulfillment of the requirements for the award of the degree of **B.TECH** in **CSE** is a record of an original work done by us under the guidance of A.SRINIVASA REDDY, Asst. Professor, CSE and this project work have not been submitted to any other university for award of any degree.

Submitted by

SHAIK SABIHA NAZREEN	(17R91A05D9)
RAMYA NADUKUDA	(17R91A05D2)
PRASHANTH TUMMALA	(17R91A05E5)
HARSHITH RAJ UPPULA	(17R91A05E6)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowned our efforts with success.

We extend my deep sense of gratitude to Principal Dr.K.VENKATA MURALI MOHAN, Teegala Krishna Reddy Engineering College, Meerpet, for permitting us to undertake this project.

We indebted to Dr.Ch.V.PHANI KRISHNA, Professor and Head of the Department, Computer Science and Engineering, Teegala Krishna Reddy Engineering College, Meerpet, for his support and guidance throughout our project.

We indebted to our guide,A.SRINIVASA REDDY,Asst.Professor, Department of Computer Science, Teegala Krishna Reddy Engineering College, Meerpet, for his constant support and guidance throughout our project.

We indebted to the project coordinator,Mrs.G.RANI Asst.Professor,Computer Science Engineering, Teegala Krishna Reddy Engineering College, Meerpet for their support and guidance throughout our project.

Finally, we express thanks to one and all who have helped us in successfully completing this project report. Furthermore, we would like thank my family and friends for their moral support and encouragement.

By

SHAIK SABIHA NAZREEN (17R91A05D9)

RAMYA NADUKUDA (17R91A05D2)

PRASHANTH TUMMALA (17R91A05E5)

HARSHITH RAJ UPPULA (17R91A05E6)

CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

Abbrevations	Acronyms
1.SDLC	Software Development Life Cycle
2.IDS	Intrusion Detection System
3.CI	Computational Intelligence
4.IEEE	Institute of Electrical and Electronics Engineers
5.GUI	Graphical User Interface
6.UML	Unified Modeling Language
7.KNN	K-Nearest Neighbors
8.NFR	Non-Functional Requirement
9.DFD	Data Flow diagram

ABSTRACT

An Intrusion Detection System (IDS) is a software that monitors a single or a network of computers for malicious activities (attacks) that are aimed at stealing or censoring information or corrupting network protocols. Most techniques used in today's IDS are not able to deal with the dynamic and complex nature of cyberattacks on computer networks.

Hence, efficient adaptive methods like various techniques of machine learning can result in higher detection rates, lower false alarm rates and reasonable computation and communication costs. We divide the schemes into methods based on classical artificial intelligence (AI) and methods based on computational intelligence (CI). We explain how various characteristics of CI techniques can be used to build efficient IDS.

INTRODUCTION

1. INTRODUCTION

1.1 MOTIVATION

Networks attacks are subject to attacks from malicious sources. Network attack is the intrusion or threat can be defined as any deliberate action that attempts unauthorized access of Information manipulation and by exploiting the existing vulnerabilities in the system. A Network attack is deliberate exploitation of computer systems, technology-dependent enterprises and networks. Network attacks use malicious code to alter computer code, logic or data, resulting in disruptive consequences that can compromise data and lead to cybercrimes, such as information and identity the theft.

1.2 PROBLEM DEFINITION

This system monitors the traffic on individual networks or subnets by continuously analyzing the traffic and comparing it with the known attacks in the library. If an attack is detected, an alert is sent to the system administration. It is placed mostly at important points in the network so that it can keep an eye on the traffic travelling to and from the different devices on the network. The IDS is placed along the network boundary or between the network and the server. An advantage of this system is that it can be deployed easily and at low cost, without having to be loaded for each system.

Hence, efficient adaptive methods like various techniques of machine learning can result in higher detection rates, lower false alarm rates and reasonable computation and communication costs. A set of techniques used for detection of abnormal behaviors on network. IDS is capable of detecting malicious activities that cannot be detected by conventional firewalls.It is a passive alert system, i.e. it only detects attacks but not prevents it.

1.3 OBJECTIVE OF PROJECT

The objective of this project is to detects attack based on irregularities in the pattern with respect to normal pattern will have the following functions:

- Anomaly IDS is capable of detecting new unknown threats.
- Computational Intelligence (CI) based algorithms are used to enhance the prediction of threats.

LITERATURE SURVEY

2.LITERATURE SURVEY

2.1 INTRODUCTION

An Anomaly based Intrusion Detection/Prevention System is a system for detecting computer intrusions by monitoring system activity and classifying it as either normal or anomalous. It consists of a statistical model of a normal network traffic which consists of the bandwidth used, the protocols defined for the traffic, the ports and devices which are part of the network. It regularly monitors the network traffic and compares it with the statistical model. In case of any anomaly or discrepancy, the administrator is alerted. An advantage of this system is they can detect new and unique attacks.

2.1.1 PURPOSE OF THE LITERATURE SURVEY

- Identifies gaps in current knowledge.
- Helps you to avoid reinventing the wheel by discovering the research already conducted on a topic.
- Sets the background on what has been explored on a topic so far.
- Increases your breadth of knowledge in your area of research.
- Helps you identify seminal works in your area.
- Allows you to provide the intellectual context for your work and position your research with other, related research.
- Provides you with opposing viewpoints.
- Helps you to discover research methods which may be applicable to your work.

2.2 EXISTING SYSTEM

- Present the attack in the from of signature and patterns.
- Patterns are then maintained as a database of attacks.
- These signature/patterns are compared with the data received on the network.

2.3 LIMITATIONS OF EXISTING SYSTEM

- Detect only those threats that are in the database.
- Can detect previously known attacks only.
- Updating is to much times consuming.

2.4PROPOSED SYSTEM

- Based on the statistical analysis.
- Detects attack based on irregularities in the pattern with respect the normal pattern.
- Creates a model of the normal behavior of the system.
- Then look for the activities that are different from the created model.

2.5MERITS OF PROPOSED SYSTEM

- Anomaly IDS is capable of detecting new unknown threats.
- Computational Intelligence (CI) based algorithms are used to enhance the prediction of threats.

2.6FEASIBILITY STUDY

Feasibility study is an analysis of the viability idea. The studies provide thorough analysis of the system. The outcome of the feasibility studies will indicate whether can proceed or not to develop the system.

Technical Feasibility

A large part of determining resources has to do with assessing technical feasibility. It considers the technical requirements of the proposed project.The technical requirements are then compared to the technical capability of organisation.The systems project is considered technically feasible if the internal technical capability is sufficient to support the project requirements.

The tools that will use for this system are:

- Python as the main development platform.
- Another language such as CSS and JavaScript,HTML,Bootstrap.

Operational Feasibility

Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented.

Operational feasibility is a measure of how well a proposed system solves the problems,and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

Economic Feasibility:

To decide whether a project is economically feasible, we have to consider various factors as:

- Cost benefit analysis
- Long-term returns
- Maintenance costs

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit .

SOFTWARE DEVELOPMENT LIFE CYCLE

3. SOFTWARE DEVELOPMENT LIFE CYCLE

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality software. The SDLC aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

- SDLC is the acronym of Software Development Life Cycle.
- It is also called as Software Development Process.
- SDLC is a framework defining tasks performed at each step in the software development process.
- ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software.

Fig. 3.0: Software Development Life Cycle

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

The following figure is a graphical representation of the various stages of a typical SDLC.

A typical Software Development Life Cycle consists of the following stages:

Stage 1: Planning and Requirement Analysis

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

Stage 2: Defining Requirements

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Designing the Product Architecture

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

Stage 4: Building or Developing the Product

In this stage of SDLC the actual development starts and the product is built. The programming
code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high-level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

Stage 5: Testing the Product

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Stage 6: Deployment in the Market and Maintenance

Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

SDLC Models

There are various software development life cycle models defined and designed which are followed during the software development process. These models are also referred as Software Development Process Models". Each process model follows a Series of steps unique to its type to ensure success in the process of software development.

Following are the most important and popular SDLC models followed in the industry:

- Waterfall Model
- Iterative Model
- Spiral Model
- V-Model
- Big Bang Model

Other related methodologies are Agile Model, RAD Model, Rapid Application Development and Prototyping Models.

3.1 SOFTWARE PROCESS MODEL

A software process model is a simplified representation of a software process. Each model represents a process from a specific perspective.These generic models are abstractions of the process that can be used to explain different approaches to the software development. They can be adapted and extended to create more specific processes.

3.1.1 Waterfall Model

Description:

The Waterfall Model is a linear sequential flow in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of software implementation. This means that any phase in the development process begins only if the previous phase is complete.

Fig. 3.1.1: Waterfall Model

The waterfall approach is the earliest approach and most widely known that was used for software development.

- Easy to explain to the users.
- Structures approach.
- Stages and activities are well defined.
- Helps to plan and schedule the project.
- Verification at each stage ensures early detection of errors/misunderstanding.
- Each phase has specific deliverables.

Requirements: This project requires the python language .

Design: This Project makes use of the HTML,CSS,JS for the designing purpose.

Execution: This project can be executed with the help of FLASK Frame Work.

Testing: This project had successfully overcome all the testing scenarios. Release: This project can be deployed into the cloud for better understanding of corporate actions.

SYSTEM ANALYSIS

4. SYSTEM ANALYSIS

4.1 REQUIREMENTS

Requirements analysis is done in order to understand the problem the software system is to solve. The problem could be automating an existing manual process, developing a new automated system, or a combination of the two. For large systems that have many features, and that need to perform many different tasks, understanding the requirements of the system is a major task. The emphasis in requirements analysis is on identifying what is needed from the system, not how the system will achieve its goals. This task is complicated by the fact that there are often at least two parties involved in software development-a client and a developer. The developer usually does not understand the client's problem domain and the client does not understand the issues involved in the system software systems developed by the developers. Hence causes a communication gap between them.

4.2 FUNCTIONAL REQUIREMENTS

Functional Requirements are the functions that define the system. Functional Requirements are captured in use cases. Functional Requirements of the system for those requirements which are expressed in the natural language style.Functional Requirements are those which represent what are the project functions about. Some of the functional requirements are as follows:

- Build accurate, and complete data from conflicting data retrieved from multiple sources.
- For every search robot looks for new data available for relevant corporate action.

Corporate Actions: Jarvis will scrap the corporate actions from different available sources and displays.

Graphical Analysis: User can have different perspectives of the Graphical Analysis for understanding the impact of corporate actions with stock market and the graphical data can be downloaded.

Upcoming Corporate Actions: Jarvis will scrap the upcoming corporate actions from different available sources and displays.

Download: The scraped data is available for downloading.

4.3 NON-FUNCTIONAL REQUIREMENTS

- Security
- Performance
- Accuracy
- Efficiency
- User interface

4.4 SOFTWARE REQUIREMENTS

- Operating System : Windows 10
- Backend : Python 3.7

4.5 HARDWARE REQUIREMENTS

- Processor : Intel Based Systems.
- Hard Disk : 500 GB.
- RAM : 4 GB.

5. SYSTEM DESIGN

5. SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE:

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

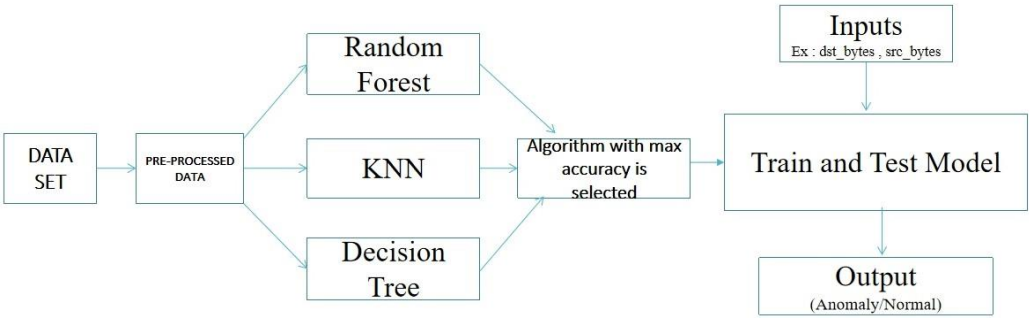


Fig. 5.1: System Architecture

As shown in the figure,first dataset need to be uploaded and preprocessed,given to algorithms and highest accuracy algorithm is trained and tested ,input given to test model and output is ,the attack is normal or abnormal.

5.2 UML DIAGRAMS:

UML stands for Unified Modeling Language. UML is a standardized general purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

Goals:

The Primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modelling language.
- Encourage the growth of OO tools market.

5.2.1 Class Diagram:

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes,their attributes,operations(or methods).

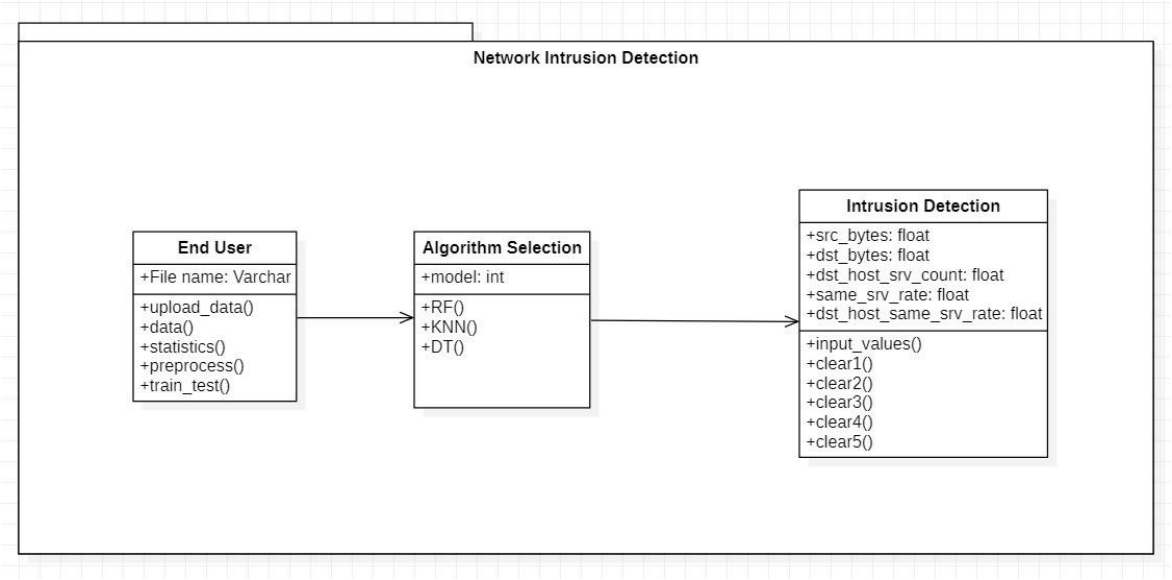


Fig. 5.2.1: class diagram

Class Name:

End user

Attributes:

File name:Attribute to store datset

Operations:

Upload_data():To upload the dataset.

Data():Read the dataset.

Statistics():Gives count of rows and columns.

Preprocess():It process data remove redundancy data and inconsistent data.

Train_test():It train the algorithm.

Class Name:

Algorithm selection

Attributes:

Model:Gives highest accuracy of an algorithms.

Operation:

RF():Datset given to random forest algorithm gives accuracy.

KNN():Datset given to random forest algorithm gives accuracy.

Decisiontree():Datset given to random forest algorithm gives accuracy.

Class Name:

Intrusion detection

Attributes:

Src_bytes:input value

Dest_bytes: input value

Operations:

Input(): It takes input to give attack is normal or abnormal.

5.2.2 Use Case Diagram:

UML provides the use case diagram to facilitate the process of requirements gathering. The use case diagram models the interactions between the system's external clients and the use cases of the system.

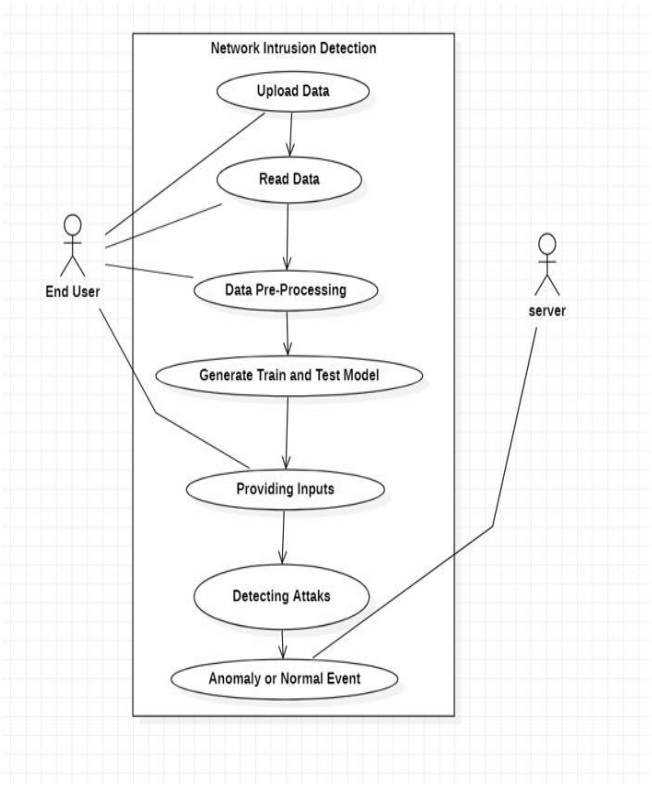


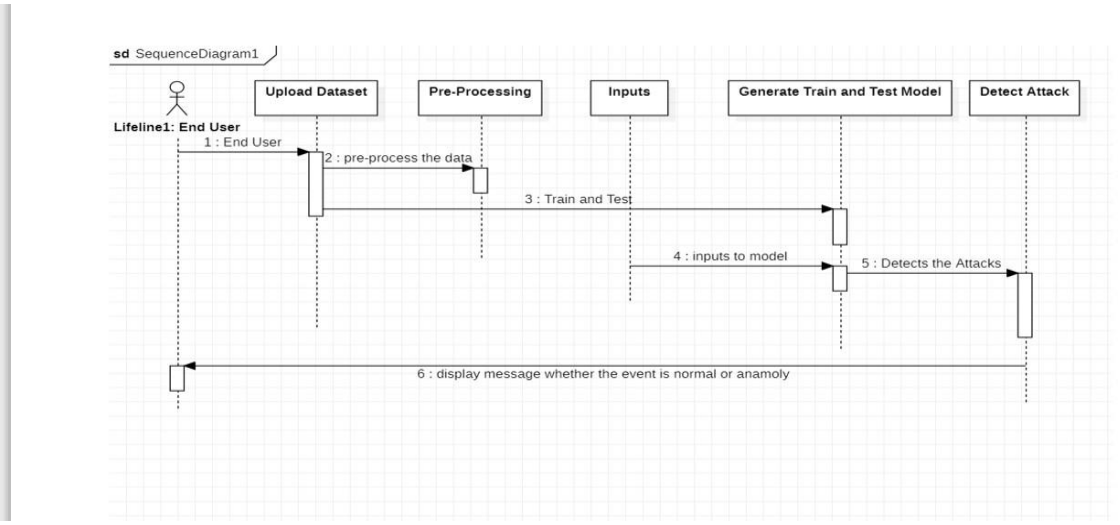
Fig. 5.2.2:use case diagram

Participating Actor:end user

- Enduser upload dataset and read the data.
- Data is processed to remove redundancy data and inconsistentdata.
- Datset given to algorithms .
- And highest accuracy of algorithm is trained and tested.
- Input is given to test model to give output is normal or abnormal.

5.2.3 Sequence Diagram:

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

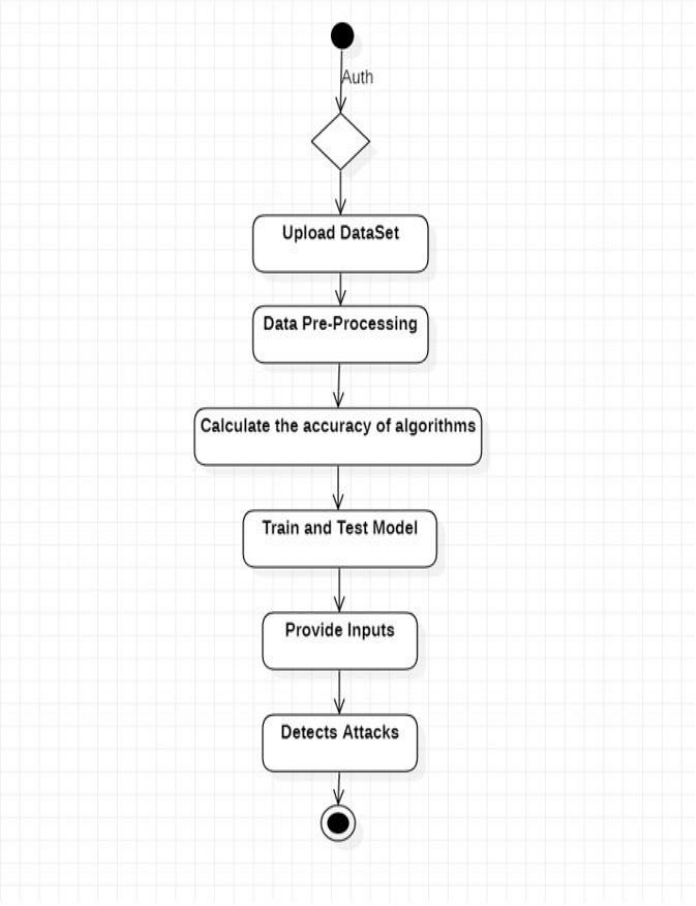


5.2.3 sequence diagram

- Enduser upload dataset and read the data.
- Data is processed to remove redundancy data and inconsistentdata.
- Datset given to algorithms .
- And highest accuracy of algorithm is trained and tested.
- Input is given to test model to give output is normal or abnormal.

5.2.4 Activity Diagram:

Activity diagram is another important behavioral diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity. Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction.



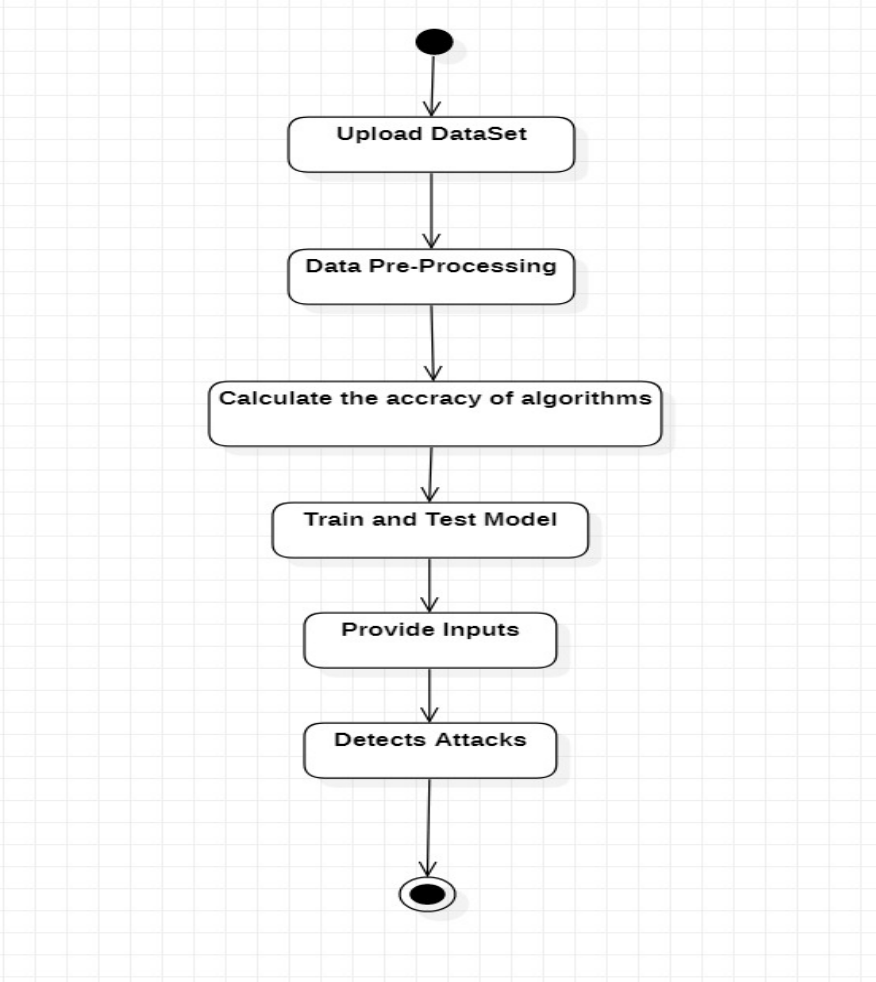
5.2.4 Activity diagram

Activity Diagram cases:

- Enduser upload dataset and read the data.
- Data is processed to remove redundancy data and inconsistentdata.
- Datset given to algorithms .
- And highest accuracy of algorithm is trained and tested.
- Input is given to test model to give output is normal or abnormal.

5.2.5 **State Diagram**:

Statechart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. Statechart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events.



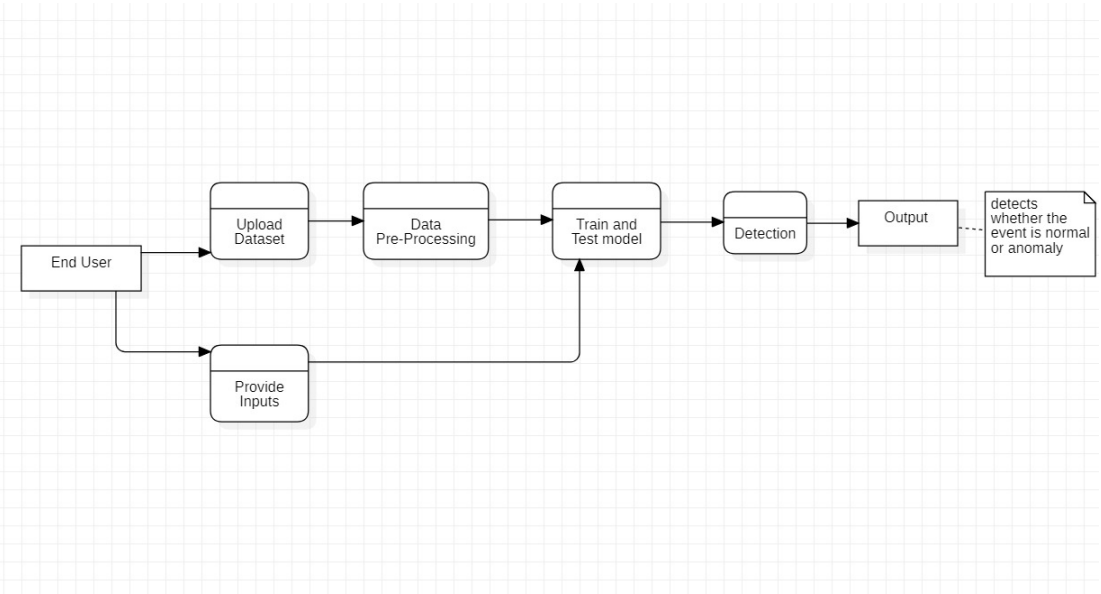
5.2.5 State Diagram

State Diagram cases:

- Enduser upload dataset and read the data.
- Data is processed to remove redundancy data and inconsistentdata.
- Datset given to algorithms .
- And highest accuracy of algorithm is trained and tested.
- Input is given to test model to give output is normal or abnormal.

5.2.6 Data Flow diagram:

A data-flow diagram is a way of representing a flow of a data of a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart .



5.2.6 Data Flow Diagram

Data flow diagram cases:

- Enduser upload dataset and read the data.
- Data is processed to remove redundancy data and inconsistentdata.
- Datset given to algorithms .
- And highest accuracy of algorithm is trained and tested.
- Input is given to test model to give output is normal or abnormal.

IMPLEMENTATION

6.IMPLEMENTATION

6.1 METHODS OF IMPLEMENTATION

6.1.1 Python

A Brief History of Python:

Python is a widely used general-purpose, high-level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

In the late 1980s, history was about to be written. It was that time when working on Python started. Soon after that, Guido Van Rossum began doing its application based work in December of 1989 by at Centrum Wiskunde & Informatica (CWI) which is situated in Netherland. It was started firstly as a hobby project because he was looking for an interesting project to keep him occupied during Christmas. The programming language which Python is said to have succeeded is ABC Programming Language, which had the interfacing with the Amoeba Operating System and had the feature of exception handling. He had already helped to create ABC earlier in his career and he had seen some issues with ABC but liked most of the features. After that what he did as really very clever. He had taken the syntax of ABC, and some of its good features. It came with a lot of complaints too, so he fixed those issues completely and had created a good scripting language which had removed all the flaws. The inspiration for the name came from BBC's TV Show – 'Monty Python's Flying Circus', as he was a big fan of the TV show and also he wanted a short, unique and slightly mysterious name for his invention and hence he named it Python! He was the "Benevolent dictator for life" (BDFL) until he stepped down from the position as the leader on 12th July 2018. For quite some time he used to work for Google, but currently, he is working at Dropbox.

The language was finally released in 1991. When it was released, it used a lot fewer codes to express the concepts, when we compare it with Java, C++ & C. Its design philosophy was quite good too. Its main objective is to provide code readability and advanced developer productivity. When it was released it had more

than enough capability to provide classes with inheritance, several core data types exception handling and functions.

Features of Python:

Python is known for its general purpose nature that makes it applicable in almost each domain of software development. Python as a whole can be used in any sphere of development.

Here, we are specifing applications areas where python can be applied.

1) Web Applications

We can use Python to develop web applications. It provides libraries to handle internet protocols such as HTML and XML, JSON, Email processing, request, beautifulSoup, Feedparser etc. It also provides Frameworks such as Django, Pyramid, Flask etc to design and delvelop web based applications. Some important developments are: PythonWikiEngines, Pocoo, PythonBlogSoftware etc.

2) Desktop GUI Applications

Python provides Tk GUI library to develop user interface in python based application. Some other useful toolkits wxWidgets, Kivy, pyqt that are useable on several platforms. The Kivy is popular for writing multitouch applications.

3) Software Development

Python is helpful for software development process. It works as a support language and can be used for build control and management, testing etc.

4) Scientific and Numeric

Python is popular and widely used in scientific and numeric computing. Some useful library and package are SciPy, Pandas, IPython etc. SciPy is group of packages of engineering, science and mathematics.

5) Business Applications

Python is used to build Bussiness applications like ERP and e-commerce systems. Tryton is a high level application platform.

6) Console Based Application

We can use Python to develop console based applications. For example: IPython.

7) Audio or Video based Applications

Python is awesome to perform multiple tasks and can be used to develop multimedia applications. Some of real applications are: TimPlayer, cplay etc.

8) 3D CAD Applications

To create CAD application Fandango is a real application which provides full features of CAD.

9) Enterprise Applications

Python can be used to create applications which can be used within an Enterprise or an Organization. Some real time applications are: OpenErp, Tryton, Picalo etc.

10) Applications for Images

Using Python several application can be developed for image. Applications developed are: VPython, Gogh, imgSeek etc.

6.1.2 SAMPLE CODE

```
import tkinter as tk
from tkinter import messagebox,simpledialog,filedialog
from tkinter import *
import tkinter
from imutils import paths
from tkinter.filedialog import askopenfilename

import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import cross_val_score
from sklearn.feature_selection import SelectFromModel
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
from sklearn.model_selection import
train_test_split,KFold,cross_val_score,GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier

import warnings
warnings.filterwarnings('ignore')
root= tk.Tk()
root.title("Network Intrusion Detection")
root.geometry("1300x1200")
def upload_data():
    global filename
    filename = askopenfilename(initialdir = "Dataset")
    text.insert(END,"Dataset loaded\n\n")
```

```
def data():
    global filename
    global df
    text.delete('1.0',END)
    df = pd.read_csv(filename)
    text.insert(END,"Complete Dataset\n\n")
    text.insert(END,df)
    return  df
def statistics():
    global df
    text.delete('1.0',END)
    text.insert(END,"Top FIVE rows of the Dataset\n\n")
    text.insert(END,df.head())
    stats=df.describe()
    text.insert(END,"\n\nStatistical Measurements for Data\n\n")
    text.insert(END,stats)
  def preprocess():
    global df
    text.delete('1.0',END)
    df = df.drop(['protocol_type','service','flag'],axis=1)
    df=df.loc[:, (df==0).mean() < .7]
    df['label'] = df['label'].map({'anomaly':1,'normal':0})
    text.insert(END,"\t\t\tPreprocessed Data\n\n")
    text.insert(END,df)
def train_test():
    global df
    global x_train, x_test, y_train, y_test
    text.delete('1.0',END)
    x=df.iloc[:,:-1]
    y=df.iloc[:,-1]
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=25)
    text.insert(END,"Train and Test model Generated\n\n")
    text.insert(END,"Total Dataset Size : "+str(len(df))+"\n")
    text.insert(END,"Training Size : "+str(len(x_train))+"\n")
```

```
    text.insert(END,"Test Size : "+str(len(x_test))+"\n")
    return x_train,x_test,y_train,y_test


# Machine Learning Models

def RF():
    global x_train,x_test,y_train,y_test
    global new_x_train,new_x_test
    text.delete('1.0',END)
    text.insert(END,"\t\t\t\tRandom Forest Classifier\n\n")
    model= RandomForestClassifier(random_state=25)
    model= model.fit(x_train,y_train)
    features = pd.DataFrame()
    features['Feature'] = x_train.columns
    features['Importance'] = model.feature_importances_
    features.sort_values(by=['Importance'], ascending=False, inplace=True)
    text.insert(END,"Selected Important Features by *feature_importances_* &
*SelectFromModel*\n\n")
    selector = SelectFromModel(model, prefit=True)
    train_reduced = selector.transform(x_train)
    columns=x_train.columns[selector.get_support()]
    new_x_train=pd.DataFrame(train_reduced,columns=columns)
    test_reduced = selector.transform(x_test)
    new_x_test=pd.DataFrame(test_reduced,columns=columns)
    rf = RandomForestClassifier(random_state=25)
    rf.fit(new_x_train, y_train)
    pred=rf.predict(new_x_test)
    acc=accuracy_score(y_test,pred)
    cm=confusion_matrix(y_test,pred)
    CR=classification_report(y_test,pred)
    output = rf.predict(new_x_test).astype(int)
    df_output = pd.DataFrame()
```

```
    df_output['Network_Intrusion_Predicted'] = np.vectorize(lambda s: 'Anomaly' if
s==1 else 'Normal')(output)
    final_pred=pd.concat([new_x_test,df_output],axis=1 )
    final_pred.to_csv('Network_Prediction_submission@RF.csv',index=False)
    text.insert(END,features[:5])
    text.insert(END,"\n\nConfusion Matrix:\n"+str(cm)+"\n\n")
    text.insert(END,"Accuracy Score:\n"+str(np.round(acc*100,4))+' %'+"\n\n")
    text.insert(END,"Predicted Values on Test Data:\n"+str(pred)+"\n\n")
    text.insert(END,"Classification Report:\n"+str(CR))
    text.insert(END,"\n\nFinal Predicted values on New Data:\n\n")
    text.insert(END,final_pred)
    text.insert(END,"\n\nCheck the Project Directory for Submission CSV file\n\n")
    text.insert(END,"@@@----------Thank You ---------- @@@")


def KNN():
    global  x_train,x_test,y_train,y_test
    global new_x_train,new_x_test,new_data
    text.delete('1.0',END)
    text.insert(END,"\t\t\t\tLogistic Regression\n\n")
    model = KNeighborsClassifier()
    model.fit(new_x_train, y_train)
    pred=model.predict(new_x_test)
    acc=accuracy_score(y_test,pred)
    cm=confusion_matrix(y_test,pred)
    CR=classification_report(y_test,pred)
    text.insert(END,"\n\nConfusion Matrix:\n"+str(cm)+"\n\n")
    text.insert(END,"Accuracy:\n"+str(np.round(acc*100,4))+' %'+"\n\n")
    text.insert(END,"Predicted Values on Test Data:\n"+str(pred)+"\n\n")
    text.insert(END,"Classification Report:\n"+str(CR))
    output = model.predict(new_x_test).astype(int)
    df_output = pd.DataFrame()
    df_output['Network_Intrusion_Predicted'] = np.vectorize(lambda s: 'Anomaly' if
s==1 else 'Normal')(output)
```

```
    final_pred=pd.concat([new_x_test,df_output],axis=1 )
    final_pred.to_csv('Network_Prediction_submission@KNN.csv',index=False)
    text.insert(END,"\n\nFinal Predicted values on New Data:\n\n")
    text.insert(END,final_pred)
    text.insert(END,"\n\nCheck the Project Directory for Submission CSV file\n\n")
    text.insert(END,"@@@----------Thank You ---------- @@@")



def DT():
    global  x_train,x_test,y_train,y_test
    global new_x_train,new_x_test,new_data
    text.delete('1.0',END)
    text.insert(END,"\t\t\t\tDecision Tree Classifier\n\n")
    model = DecisionTreeClassifier()
    model.fit(new_x_train, y_train)
    pred=model.predict(new_x_test)
    acc=accuracy_score(y_test,pred)
    cm=confusion_matrix(y_test,pred)
    CR=classification_report(y_test,pred)
    text.insert(END,"\n\nConfusion Matrix:\n"+str(cm)+"\n\n")
    text.insert(END,"Accuracy:\n"+str(np.round(acc*100,4))+' %'+"\n\n")
    text.insert(END,"Predicted Values on Test Data:\n"+str(pred)+"\n\n")
    text.insert(END,"Classification Report:\n"+str(CR))
    output = model.predict(new_x_test).astype(int)
    df_output = pd.DataFrame()
    df_output['Network_Intrusion_Predicted'] = np.vectorize(lambda s: 'Anomaly' if
s==1 else 'Normal')(output)
    final_pred=pd.concat([new_x_test,df_output],axis=1 )
    final_pred.to_csv('Network_Prediction_submission@DT.csv',index=False)
    text.insert(END,"\n\nFinal Predicted values on New Data:\n\n")
    text.insert(END,final_pred)
    text.insert(END,"\n\nCheck the Project Directory for Submission CSV file\n\n")
    text.insert(END,"@@@----------Thank You ---------- @@@")
```

```
def input_values():
    text.delete('1.0',END)
    global x_train,x_test,y_train,y_test
    global new_x_train,new_x_test

    global src_bytes
    src_bytes = float(entry1.get())

    global dst_bytes
    dst_bytes = float(entry2.get())

    global dst_host_srv_count
    dst_host_srv_count = float(entry3.get())

    global same_srv_rate
    same_srv_rate = float(entry4.get())

    global dst_host_same_srv_rate
    dst_host_same_srv_rate  = float(entry5.get())


list1=[[src_bytes,dst_bytes,dst_host_srv_count,same_srv_rate,dst_host_same_srv_rat
e]]

    dt = DecisionTreeClassifier(random_state=25)
    dt.fit(new_x_train, y_train)


    Prediction_result  = dt.predict(list1)
    text.insert(END,"Decision Tree Classifier having greater accuracy score\n\n")
    text.insert(END,"New values are predicted from Decision Tree Classifier\n\n")
    text.insert(END,"Predicted Network Intrusion Status for the New inputs\n\n")
```

```
    text.insert(END,np.vectorize(lambda s: 'Anomaly' if s==1 else
'Normal')(Prediction_result))

font = ('times', 14, 'bold')
title = Label(root, text='Network Intrusion Detection Using Machine Learning')
title.config(font=font)
title.config(height=2, width=120)
title.place(x=0,y=5)

font1 = ('times',13 ,'bold')
button1 = tk.Button (root, text='Upload Data',width=13,command=upload_data)
button1.config(font=font1)
button1.place(x=60,y=100)

button2 = tk.Button (root, text='Read Data',width=13,command=data)
button2.config(font=font1)
button2.place(x=60,y=150)

button3 = tk.Button (root, text='Statistics',width=13,command=statistics)
button3.config(font=font1)
button3.place(x=60,y=200)

button3 = tk.Button (root, text='Preprocessing',width=13,command=preprocess)
button3.config(font=font1)
button3.place(x=60,y=250)

button4 = tk.Button (root, text='Train & Test',width=13,command=train_test)
button4.config(font=font1)
button4.place(x=60,y=300)

title = Label(root, text='Application of ML models')
title.config(font=font1)
title.config(width=25)
title.place(x=250,y=70)
```

```
button5 = tk.Button (root, text='Random Forest',width=15,bg='pale
green',command=RF)
button5.config(font=font1)
button5.place(x=300,y=100)

button6 = tk.Button (root, text='KNN',width=15,bg='sky blue',command=KNN)
button6.config(font=font1)
button6.place(x=300,y=150)

button7 = tk.Button (root, text='Decision Tree',width=15,bg='orange',command=DT)
button7.config(font=font1)
button7.place(x=300,y=200)



title = Label(root, text='Enter Input values for the New Prediction')
title.config(bg='black', fg='white')
title.config(font=font1)
title.config(width=40)
title.place(x=60,y=380)


def clear1(event):
   entry1.delete(0, tk.END)

font2=('times',10)
entry1 = tk.Entry (root) # create 1st entry box
entry1.config(font=font2)
entry1.place(x=60, y=450,height=30,width=150)
entry1.insert(0,'src_bytes')
entry1.bind("<FocusIn>",clear1)

def clear2(event):
```

```
    entry2.delete(0, tk.END)

font2=('times',10)
entry2 = tk.Entry (root) # create 1st entry box
entry2.config(font=font2)
entry2.place(x=315, y=450,height=30,width=150)
entry2.insert(0,'dst_bytes')
entry2.bind("<FocusIn>",clear2)



def clear3(event):
    entry3.delete(0, tk.END)

font2=('times',10)
entry3 = tk.Entry (root) # create 1st entry box
entry3.config(font=font2)
entry3.place(x=60, y=500,height=30,width=150)
entry3.insert(0,'dst_host_srv_count')
entry3.bind("<FocusIn>",clear3)

def clear4(event):
    entry4.delete(0, tk.END)

font2=('times',10)
entry4 = tk.Entry (root) # create 1st entry box
entry4.config(font=font2)
entry4.place(x=315, y=500,height=30,width=150)
entry4.insert(0,'same_srv_rate')
entry4.bind("<FocusIn>",clear4)

def clear5(event):
    entry5.delete(0, tk.END)

font2=('times',10)
```

```
entry5 = tk.Entry (root) # create 1st entry box
entry5.config(font=font2)
entry5.place(x=60, y=550,height=30,width=150)
entry5.insert(0,'dst_host_same_srv_rate')
entry5.bind("<FocusIn>",clear5)


Prediction = tk.Button (root,
text='Prediction',width=15,fg='white',bg='green',command=input_values)
Prediction.config(font=font1)
Prediction.place(x=180,y=650)
font1 = ('times', 11, 'bold')
text=Text(root,height=32,width=90)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set,xscrollcommand=scroll.set)
text.place(x=550,y=70)
text.config(font=font1)
#root.config(bg='grey')
root.mainloop()
```

6.1.3 SAMPLE DATA:

duration	protocol_ty	service	flag	src_bytes	dst_bytes	land	wrong_frag	urgent	hot	num_failed	logged_in	num_compr	root_shell	su_attempt	num_root	num_file_cr	num_shells	num_access	num_outbo	is_host_
0	tcp	ftp_data	SF	491	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	udp	other	SF	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	232	8153	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	199	420	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	remote_job	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	287	2251	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	tcp	ftp_data	SF	334	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	tcp	name	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	netbios_ns	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	300	13788	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	icmp	eco_i	SF	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	233	616	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	343	1178	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	tcp	mtp	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	253	11905	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
5607	udp	other	SF	147	105	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	mtp	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
507	tcp	telnet	SF	437	14421	0	0	0	0	0	1	3	0	0	0	0	0	1	0
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	227	6588	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	215	10499	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
30	0	tcp	http	SF	215	10499	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
31	0	tcp	http	SF	241	1400	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
32	0	icmp	eco_i	SF	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
33	0	tcp	finger	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
34	0	tcp	http	SF	303	555	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
35	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
36	0	udp	domain_u	SF	45	45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
37	1	udp	private	SF	105	147	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
38	0	udp	domain_u	SF	43	43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
39	0	tcp	supdup	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
40	0	tcp	http	SF	324	2302	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
41	0	tcp	uucp_path	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
42	0	tcp	ftp_data	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
43	0	tcp	Z39_50	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
44	2	tcp	smtp	SF	1591	372	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
45	9052	udp	other	SF	146	105	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
46	0	tcp	http	SF	290	3006	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
47	0	tcp	csnet_ns	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
48	0	udp	private	SF	28	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	
49	0	tcp	http	SF	255	861	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
50	0	tcp	ftp_data	SF	334	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
51	0	tcp	ftp_data	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
52	0	tcp	http	SF	302	498	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
53	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
54	0	udp	private	SF	28	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	
55	0	tcp	http	SF	220	1398	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
56	0	udp	domain_u	SF	43	69	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
57	0	udp	domain_u	SF	44	133	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
58	0	icmp	eco_i	SF	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
59	0	tcp	uucp	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

60	0	tcp	finger	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
61	0	udp	domain_u	SF	43	43	0	0	0	0	0	0	0	0	0	0	0	0	0	0
62	0	tcp	ftp_data	SF	641	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
63	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
64	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
65	0	tcp	http	SF	310	2794	0	0	0	0	0	1	0	0	0	0	0	0	0	0
66	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
67	0	tcp	smtp	SF	696	333	0	0	0	0	0	1	0	0	0	0	0	0	0	0
68	0	tcp	netbios_dgr	RSTR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
69	0	tcp	http	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70	0	tcp	netbios_dgr	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
71	0	tcp	supdup	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
72	0	tcp	http	SF	221	2878	0	0	0	0	0	1	0	0	0	0	0	0	0	0
73	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
74	0	icmp	urp_i	SF	181	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
75	0	udp	domain_u	SF	42	42	0	0	0	0	0	0	0	0	0	0	0	0	0	0
76	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
77	0	tcp	ftp_data	SF	12	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
78	0	tcp	auth	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
79	0	tcp	http	SF	210	2790	0	0	0	0	0	1	0	0	0	0	0	0	0	0
80	0	tcp	http	SF	321	5683	0	0	0	0	0	1	0	0	0	0	0	0	0	0
81	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
82	0	tcp	http	SF	329	3982	0	0	0	0	0	1	0	0	0	0	0	0	0	0
83	0	tcp	uucp_path	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
84	0	tcp	domain	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
85	0	icmp	eco_i	SF	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
86	0	tcp	http	SF	320	379	0	0	0	0	0	1	0	0	0	0	0	0	0	0
87	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
88	0	tcp	http	SF	225	3762	0	0	0	0	0	1	0	0	0	0	0	0	0	0
89	0	tcp	ftp	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table.4.7 Sample Data Table

4.8 ALGORITHM:
Decision Tree Algorithm:

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too.

The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).

In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

Types of decision trees are based on the type of target variable we have. It can be of two types:

1. Categorical Variable Decision Tree: Decision Tree which has a categorical target variable then it called a Categorical variable decision tree.
2. Continuous Variable Decision Tree: Decision Tree has a continuous target variable then it is called Continuous Variable Decision Tree.

Example:- Let's say we have a problem to predict whether a customer will pay his renewal premium with an insurance company (yes/ no). Here we know that the income of customers is a significant variable but the insurance company does not have income details for all customers. Now, as we know this is an important variable, then we can build a decision tree to predict customer income based on occupation, product, and various other variables. In this case, we are predicting values for the continuous variables.

Random Forest classifier:

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.

The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds. In data science speak, the reason that the random forest model works so well is:

A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.

The low correlation between models is the key. Just like how investments with low correlations (like stocks and bonds) come together to form a portfolio that is greater than the sum of its parts, uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions. The reason for this wonderful effect is that the trees protect each other from their individual errors (as long as they don't constantly all err in the same direction). While some trees may be wrong, many other trees will be right, so as a group the trees are able to move in the correct direction. So the prerequisites for random forest to perform well are:

- There needs to be some actual signal in our features so that models built using those features do better than random guessing.
- The predictions (and therefore the errors) made by the individual trees need to have low correlations with each other.

K-Nearest Neighbhour Alogorithm:

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
- It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
- Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

SYSTEM TESTING

7. SYSTEM TESTING

7.1 INTRODUCTION

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During software development. During testing, the program is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to perform.

The following are the Testing Objectives:

- Testing is a process of executing a program with the intent of finding an error.
- A good test has a high probability of finding an as yet undiscovered error.
- A successful test is one that uncovers an as yet undiscovered error.

Each module can be tested using the following two Strategies:

Black Box Testing:

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has been uses to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structure or external database access
- Performance errors
- Initialization and termination errors.

White Box testing:

In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. It has been uses to generate the test cases in the following cases:

- Guarantee that all independent paths have been executed.
- Execute all logical decisions on their true and false Sides.
- Execute internal data structures to ensure their validity.

7.2 DESIGN OF TEST CASES AND 7SCENARIOS

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

Unit Testing:

Unit Testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements.

- Guarantee that all independent paths have been executed.
- Execute all logical decisions on their true and false Sides.
- Execute all loops at their boundaries and within their operational bounds.
- Execute internal data structures to ensure their validity.

Integrating Testing:

Integration testing ensures that software and subsystems work together a whole. It tests the interface of all the modules to make sure that the modules behave properly when integrated together.

System Testing:

Involves in-house testing of the entire system before delivery to the user. It's aim is to satisfy the user the system meets all requirements of the client's specifications.

Acceptance Testing:

It is a pre-delivery testing in which entire system is tested at client's site on real world data to find errors.

Compile Testing:

It was a good idea to do our stress testing early on, because it gave us time to fix some of the unexpected deadlocks and stability problems that only occurred when components were exposed to very high transaction volumes.

Execution Test:

This program was successfully loaded and executed. Because of good programming there was no execution error.

Output Test:

The successful output screens are placed in the output screens section. Testing is the process of finding differences between the expected behavior specified by system models and the observed behavior of the system.

Testing Approaches:

Testing can be done in two ways:

- Bottom up approach
- Top down approach

Bottom up Approach:

Testing can perform starting from smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded within the larger system. When bottom level modules are tested attention turns to those on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower level modules.

Top down approach:

This type of testing starts from upper level modules. Since th the detailed activities usually performed in the lower level routines are not provided stubs are written. A stub is a module shell called by upper level module and that when reached properly will return a message to the calling module indicating that proper interaction occurred. No attempt is made to verify the correctness of the lower level module.

7.3 VALIDATION

The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed.

OUTPUT SCREEN

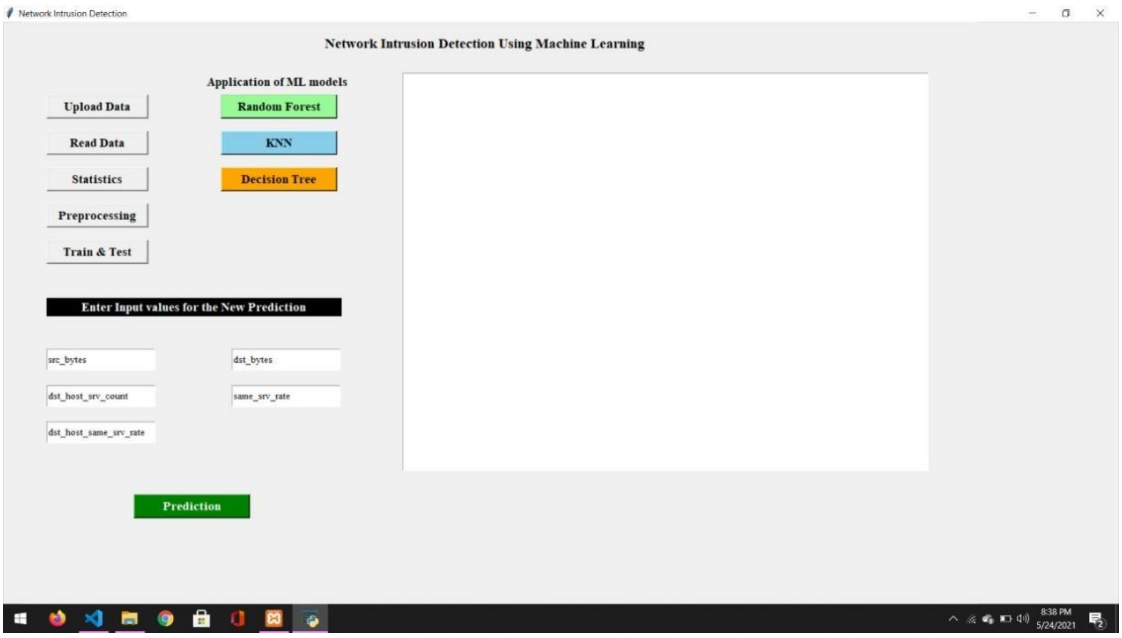


Fig.8.1.Upload dataset

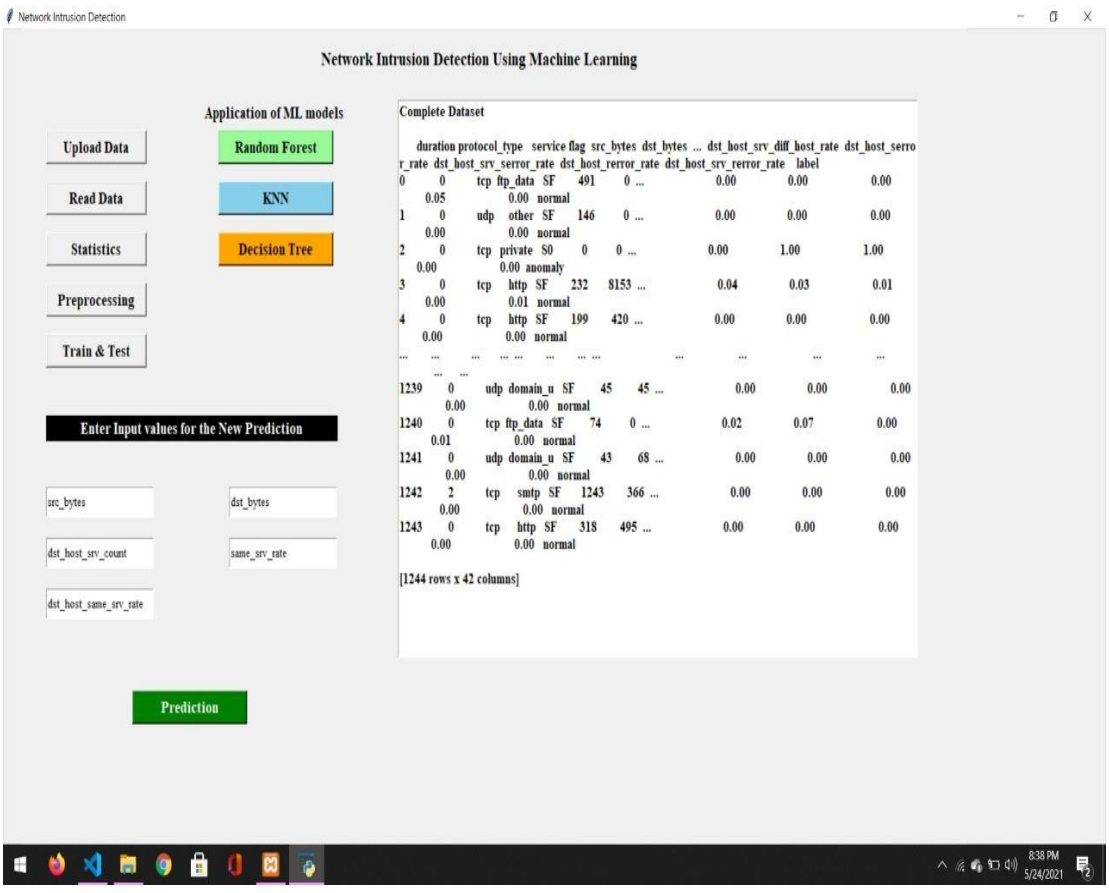


Fig.8.2.Read dataset

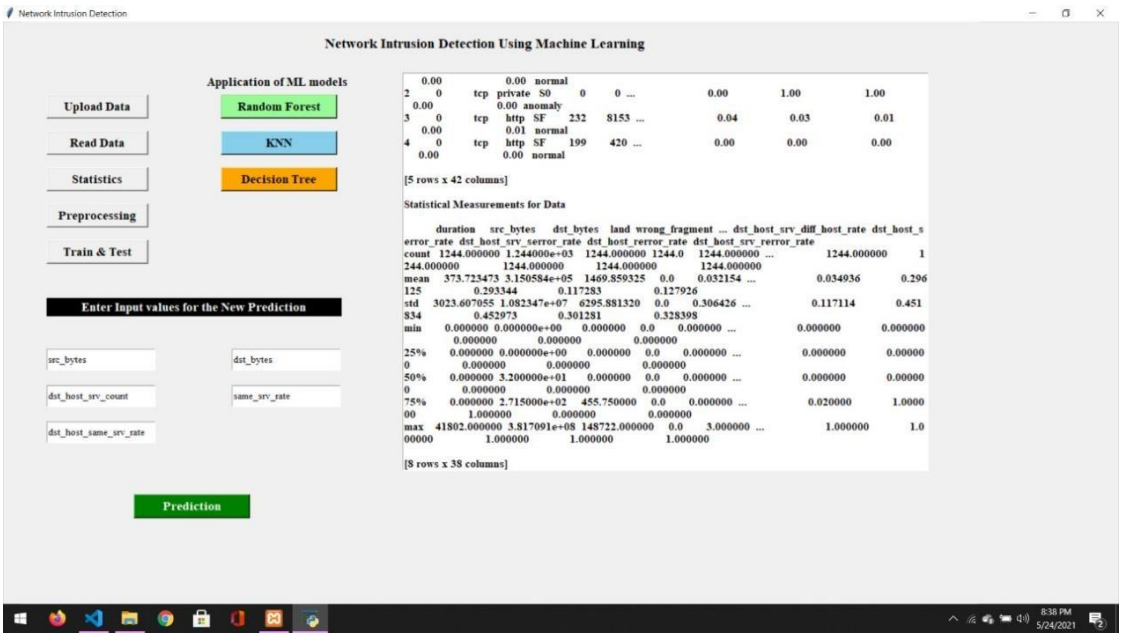


Fig.8.3Statistics of dataset

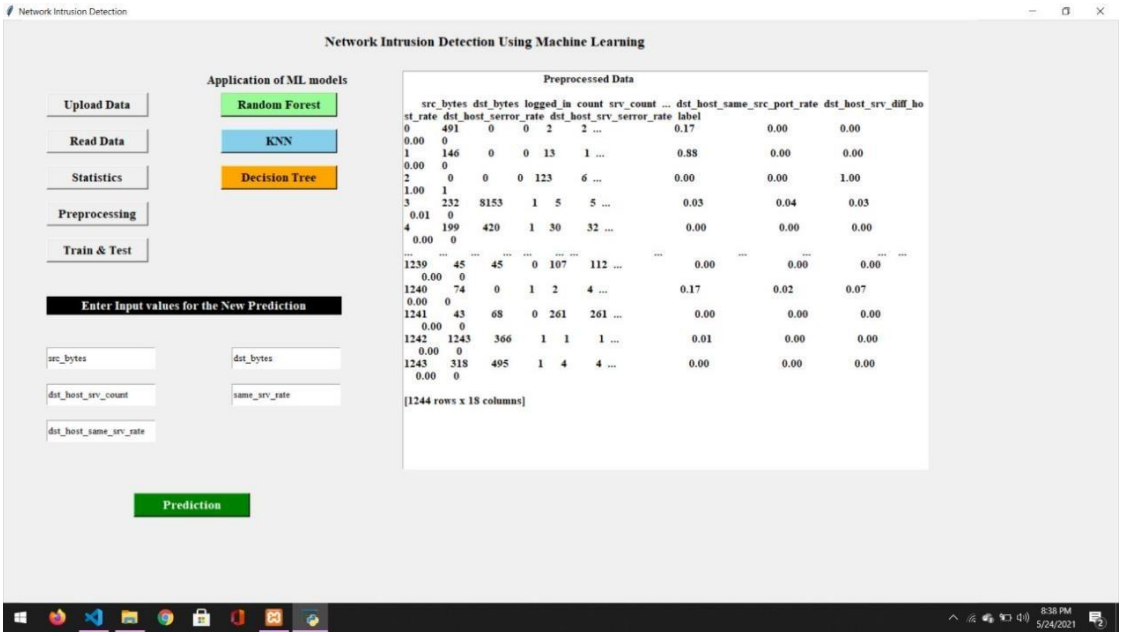


Fig.8.4 Preprocessing data

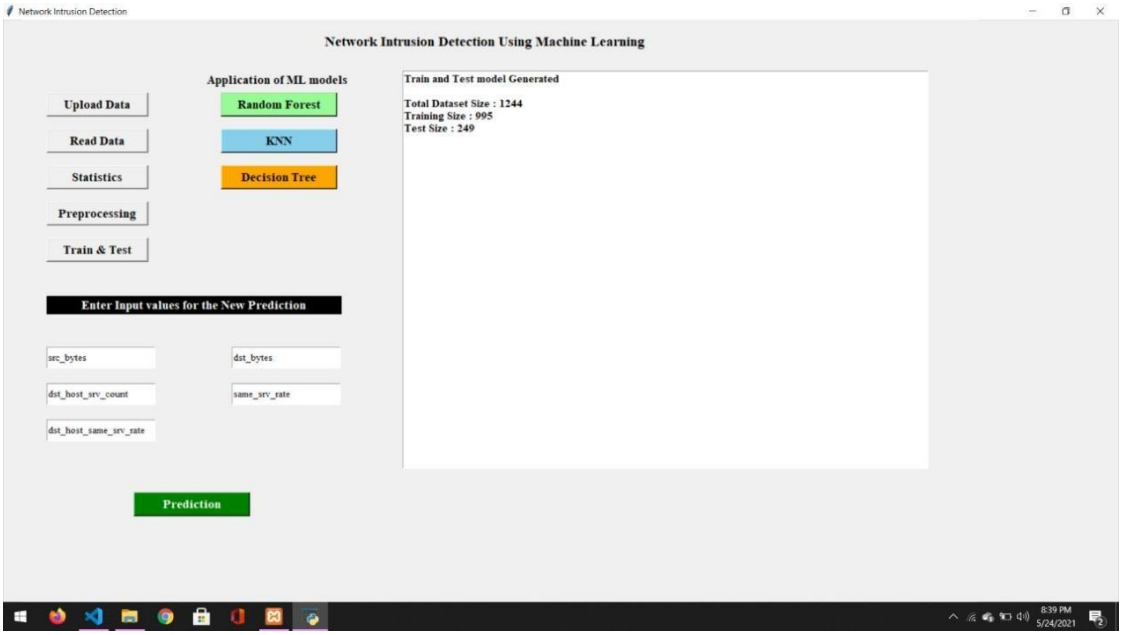


Fig.8.5.Train and test model

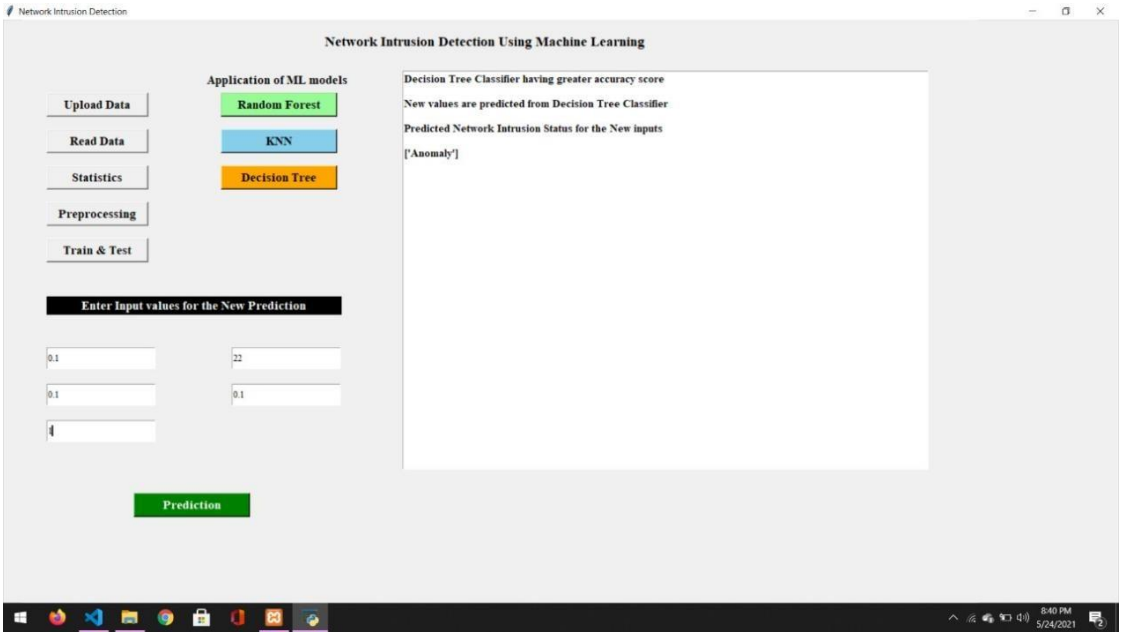


Fig.8.6 Result

CONCLUSION

9. CONCLUSION

It is desirable for anomaly-based network intrusion detection system to achieve high classification accuracy and reduce the process complexity of extracting the rules from training data. A canonical correlation analysis is used to optimize the features toward designing the scale to detect the intrusions. The selection of optimal features simplifies the process of FAIS.

The experiments were conducted using a benchmark KDD'99 Dataset. Decision Tree algorithm in machine learning has given highest accuracy of 98.3936% where as Random Forest and KNN algorithms has 97.992% and 94.7791% accuracy respectively. So Train and Test model follows Decision Tree algorithm to provide accurate and low FAIS rate.

10. FUTURE ENHANCEMENT

Intrusion Prevention:is another area that will grow dramatically in the future. Intrusion prevention is in its infancy.

Source Determination:means determining the origin of network traffic. Given how easy it is to spoof IP addresses, any source IP address in conventional IP packets must be viewed with suspicion. Tools that fabricate packets, inserting any desired IP address into the IP headers, are freely available on the Internet.

Target Detection: We're also likely to see more widespread use of target detection in the future.

11. BIBLIOGRAPHY

- https://en.wikipedia.org/wiki/Intrusion_detection_system
- https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning
- https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html#:~:text=Decision%20Tree
- https://www.idrbt.ac.in/assets/alumni/PT-2015/Vigneshwaran
- https://www.researchgate.net/publication/255670538_An_Intrusion_Detection_System_For_Academic_Institutions
- https://www.google.com/search?q=sdlc&rlz=1C1ONGR_enIN955IN955&oq=sdlc&aqs=chrome.0.69i59j0i20i263i433l2j0i433l3j0j0i433j0l2.1165j0j7&sourceid=chrome&ie=UTF-8
- https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm#Algorithm
- https://www.simplilearn.com/feasibility-study-article
- https://www.geeksforgeeks.org/types-software-testing/
- https://www.google.com/search?q=network+intrusion+detection&rlz=1C1ONGR_enIN955IN955&oq=network&aqs=chrome.0.69i59j69i57j35i39j0i67i1
- https://www.idrbt.ac.in/assets/alumni/PT-2015/Vigneshwaran