A

Mini Project

On

**Local Dynamic Neighborhood Based Outlier Detection Approach and its Framework for Large-Scale Datasets**

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

S.Hymavathi (217R1A0552)

CH.Prabhakar (217R1A0515)

Vairab Patra (217R1A0562)

Under the Guidance of

**Ms. Sanjana. S. Nazare**

(Assistant Professor)



## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**CMR TECHNICAL CAMPUS**

**UGC AUTONOMOUS**

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

**2021-25**

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



**CERTIFICATE**

This is to certify that the project entitled “**Local Dynamic Neighborhood Based Outlier Detection Approach and its Framework for Large-Scale Datasets**” being submitted by **S.Hymavathi (217R1A0552), CH.Prabhakar (217R1A0515) & Vairab Patra (217R1A0562)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2024-25.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

#### Sanjana. S. Nazare Dr. A. Raji REDDY

#### (Assistant Professor) DIRECTOR

**INTERNAL GUIDE**

**Dr. Nuthanakanti Bhaskar EXTERNALEXAMINER**

**HoD**

**Submitted for viva voice Examination held on**

**ACKNOWLEGDEMENT**

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We take this opportunity to express my profound gratitude and deep regard to my guide

**Ms.Sanjana. S. Nazare,** Assistant Professor for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark. We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Coordinators: **Dr. K.Maheswari, Mr.J.Narasimha Rao, Ms.K.Shilpa, Mr.K.Ranjith Reddy** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. Nuthanakanti Bhaskar,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We would like to express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

#### S.Hymavathi (217R1A055) CH.Prabhakar(217R1A0515) Vairab Patra (217R1A0562)

**ABSTRACT**

Local outlier detection is a hot area and great challenge in data mining, especially for large-scale datasets. On the one hand, traditional algorithms often achieve low-quality detection results and are sensitive to neighborhood size. On the other hand, they are infeasible for large-scale datasets due to at least O(N2 ) time and space complexity. In light of these, we propose a new local outlier detection algorithm, which is designed based on a new stable neighborhood strategy-dynamic references nearest neighbors (DRNN). Meanwhile, we present a new detection framework by combining the proposed approach and k-mean for large-scale datasets. Experimental results demonstrate that the proposed algorithm can produce higher quality and robust detection results compared to several classic methods. Meanwhile, the new detection framework is able to significantly improve detecting efficiency without sacrificing accuracy

i

## LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **FIGURE NAME** | **PAGE NO** |
| Figure 3.1 | Project Architecture | 6 |
| Figure 3.2 | Use case diagram | 7 |
| Figure 3.3 | Class diagram | 8 |
| Figure 3.4 | Sequence diagram | 9 |

ii

ii

## LIST OF SCREENSHOTS

|  |  |  |
| --- | --- | --- |
| **SCREENSHOT NO.** | **SCREENSHOT NAME** | **PAGE NO** |
| Screenshot 5.1 | Upload KDD Dataset | 15 |
| Screenshot 5.2 | Preprocess Dataset | 15 |
| Screenshot 5.3 | Random Forest Result | 16 |
| Screenshot 5.4 | Applying K-MEANS | 17 |
| Screenshot 5.5 | LDNOD Outlier Detection with Random Forest Result | 17 |
| Screenshot 5.6 | Comparison Graph | 18 |

iii

**TABLE OF CONTENTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ABSTRACT** | |  |  | i |
| **LIST OF FIGURES** | | |  | ii |
| **LIST OF SCREENSHOTS** | | | | iii |
| **1.** | **INTRODUCTION** | | | 1 |
|  | 1.1 | PROJECT SCOPE | | 1 |
|  | 1.2 | PROJECT PURPOSE | | 1 |
|  | 1.3 | PROJECT FEATURES | | 1 |
| **2.** | **SYSTEM ANALYSIS** | | | 2 |
|  | 2.1 | PROBLEM DEFINITION | | 2 |
|  | 2.2 | EXISTING SYSTEM | | 2 |
|  |  | 2.2.1 | LIMITATIONS OF THE EXISTING SYSTEM | 3 |
|  | 2.3 | PROPOSED SYSTEM | | 3 |
|  |  | 2.3.1 | ADVANTAGES OF PROPOSED SYSTEM | 3 |
|  | 2.4 | FEASIBILITY STUDY | | 3 |
|  |  | 2.4.1 | ECONOMIC FESIBILITY | 4 |
|  |  | 2.4.2 | TECHNICAL FEASIBILITY | 4 |
|  |  | 2.4.3 | BEHESVIRAL FEASIBILITY | 4 |
|  | 2.5 | HARDWARE & SOFTWARE REQUIREMENTS | | 5 |
|  |  | 2.5.1 | HARDWARE REQUIREMENTS | 5 |
|  |  | 2.5.2 | SOFTWARE REQUIREMENTS | 5 |
| **3.** | **ARCHITECTURE** | | | 6 |
|  | 3.1 | PROJECT ARCHITECTURE | | 6 |
|  | 3.2 | DESCRIPTION | | 6 |
|  | 3.3 | USECASE DIAGRAM | | 7 |
|  | 3.4 | CLASS DIAGRAM | | 8 |
|  | 3.5 | SEQUENCE DIAGRAM | | 9 |
| **4.** | **IMPLEMENTATION** | | | 10 |
|  | 4.1 | SAMPLE CODE | | 10 |
| **5.** | **RESULT AND DISCRIPTION** | | | 15 |
| **6.** | **TESTING** | |  | 19 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 6.1 | INTRODUCTION TO TESTING | | 19 |
|  | 6.2 | TYPES OF TESTING | | 19 |
|  |  | 6.2.1 | UNIT TESTING | 19 |
|  |  | 6.2.2 | INTEGRATION TESTING | 19 |
|  |  | 6.2.3 | FUNCTIONAL TESTING | 20 |
|  | 6.3 | TEST CASES | | 20 |
|  |  | 6.3.1 | UPLOADING DATA | 20 |
|  |  | 6.3.2 | CLASSIFICATION | 21 |
| **7.** | **CONCLUSION & FUTURE SCOPE** | | | 22 |
|  | 7.1 | PROJECT CONCLUSION | | 22 |
|  | 7.2 | FUTURE SCOPE | | 22 |
| **8.** | **BIBLIOGRAPHY** | |  | 23 |
|  | 8.1 | REFERENCES | | 23 |
|  | 8.2 | WEBSITES | | 23 |

1. **INTRODUCTION**

**1.INTRODUCTION**

### PROJECT SCOPE

This project is titled as “Local Dynamic Neighborhood Based Outlier Detection Approach And Its Framework For Large-Scale Datasets”. local outlier detection aims to capture the records that are deviant from their neighbors. This project uses machine-learning methods. The approach will focus on identifying outliers in complex, high-dimensional datasets by analysing dynamic neighbourhoods for each data point, rather than using fixed global thresholds. The project will evaluate the method's performance in terms of accuracy, speed, and scalability.

### PROJECT PURPOSE

The purpose of the "Local Dynamic Neighborhood Based Outlier Detection Approach" is to develop a scalable and accurate outlier detection method for large-scale datasets. By dynamically adapting to local data patterns, the approach aims to improve anomaly detection in complex, high-dimensional data environments, making it suitable for real-time applications like fraud detection, cybersecurity, and big data analytics.

### PROJECT FEATURES

The project features a dynamic neighborhood detection method that adapts to local data patterns, improving the accuracy of outlier identification in large-scale, high-dimensional datasets. It is designed to be highly scalable, utilizing distributed or parallel computing for efficient processing of massive datasets. The framework supports real-time outlier detection, making it suitable for applications that require immediate responses, such as fraud detection and cybersecurity. Additionally, the approach is flexible enough to handle varying data distributions, offering enhanced precision compared to traditional methods. The project will also include performance benchmarking to evaluate its accuracy, speed, and efficiency against existing outlier detection techniques.

# 2. SYSTEM ANALYSIS

## 2.SYSTEM ANALYSIS

### SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. The LDNBODframework for large-scale datasets is designed to overcome the limitations of traditional outlier detection systems. This system leverages a more flexible and adaptable approach to efficiently handle large datasets by dynamically adjusting to local data characteristics. The following is a detailed analysis of the system.

### PROBLEM DEFINITION

Outlier detection in large-scale datasets faces challenges such as high computational costs, inefficiency in high-dimensional data, fixed neighbourhood structures, and sensitivity to noisy data. These challenges are particularly evident in traditional methods like distance-based and density-based approaches. Therefore, the need arises for a scalable, dynamic method that adapts to local variations in data density and handles large-scale datasets more efficiently.

### EXISTING SYSTEM

Local outlier factor(LOF) struggles with varying densities, leading to inaccurate outlier detection.Variants improve Local outlier factor(LOF) but remain ineffective on scattered datasets.INFLO reduces issues but depends on sensitive neighborhood parameters. LDOF’s distance-based measures are less robust for large datasets. Clustering methods suffer from unstable results in complex datasets.

**2.2.1 LIMITATIONS OF EXISTING SYSTEM**

* + - * Handling High-Dimensional Data.
      * Lack of Interpretability.
      * Scalability Issues

To avoid all these limitations and make the working more accurately the system needs to be implemented efficiently.

### PROPOSED SYSTEM

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides higher accuracy and reduces the classification work. The existing system has several disadvantages and many more difficulties to work well. The proposed system uses various techniques like random forest and k-means.

### ADVANTAGES OF THE PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features

* + - * Scalability
      * Higher Detection Accuracy
      * Adaptability
      * Improved Efficiency for Large-Scale Datasets
      * Minimum time required.

### FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company.

### ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

* + - * The costs conduct a full system investigation.
      * The cost of the hardware and software.
      * The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development.

### TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### BEHAVIORAL FEASIBILITY

This includes the following questions:

* + - * Is there sufficient support for the users?
      * Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

### HARDWARE & SOFTWARE REQUIREMENTS

* + 1. **HARDWARE REQUIREMENTS:**

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

|  |  |  |
| --- | --- | --- |
| * Processor | : | Intel i7 |
| * Hard disk | : | 40 GB . |
| * RAM | : | 8GB |

### SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

* Operating system : Windows 8 Professional.
* Languages : Python 3.7.0

# 

# 3. ARCHITECTURE

## 3. ARCHITECTURE

### PROJECT ARCHITECTURE

This project architecture shows the procedure followed for detection of outliers using machine learning, starting from input to final detection.

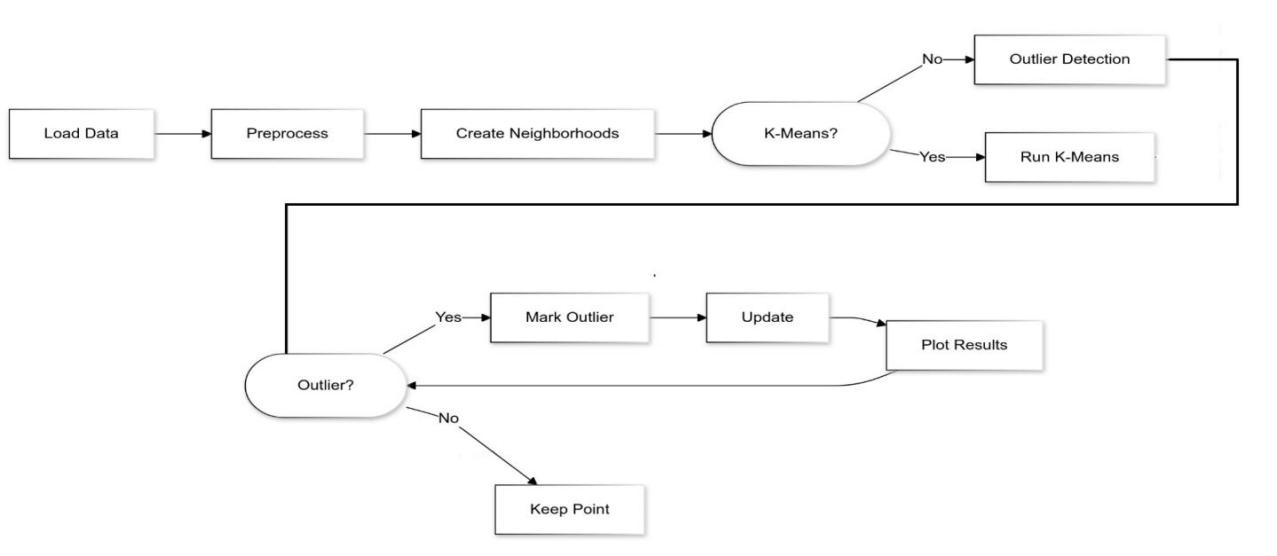


Figure 3.1: Project Architecture

### DESCRIPTION

1. Initial Data Handling

* Load Data: Starting point where data is imported
* Preprocess: Initial data cleaning and preparation
* Create Neighborhoods: Grouping of data points into local neighborhoods

2. Processing Path Decision

* K-Means Decision Point: Binary decision for additional clustering
* Yes Path: Implements K-means clustering before outlier detection
* No Path: Proceeds directly to outlier detection

3. Evaluation and Classification

* Evaluate Points: Receives input from both:
  + Initial data loading
  + Outlier detection process
* Outlier Decision: Binary classification of points
  + Yes: Points marked as outliers
  + No: Points kept as normal data

4. Final Stages

* Update: System state modification based on classifications
* Plot Results: Final visualization of findings

### USE CASE DIAGRAM

In the use case diagram, we have basically two actors who are the user and the administrator. The user has the rights to login, access to resources and to view the crime details. Whereas the administrator has the login, access to resources of the users and also the right to update and remove the crime details, and he can also view the user files.

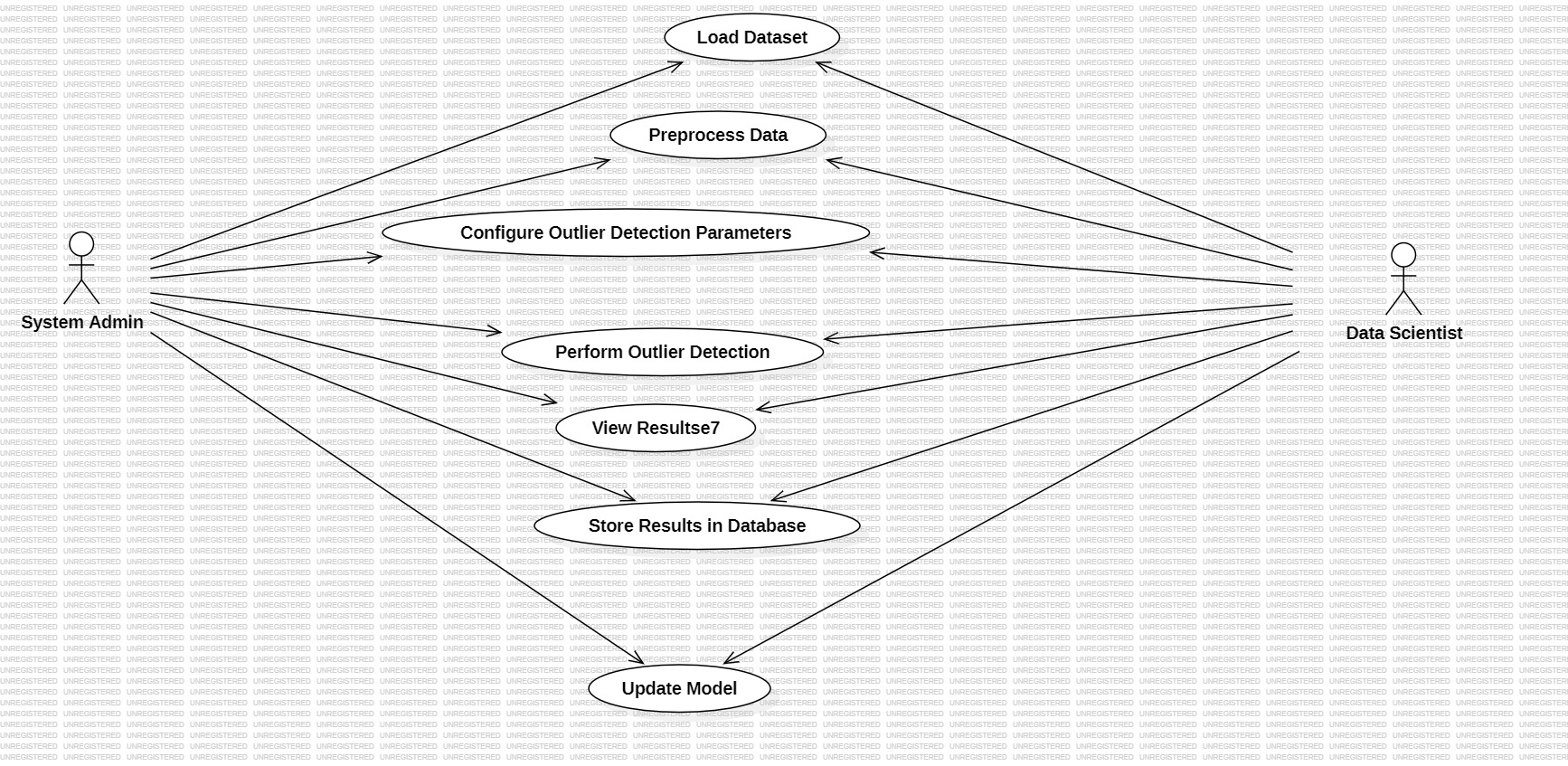


Figure 3.2: Use Case Diagram

### CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling 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), and the relationships among the classes. It explains which class contains information.

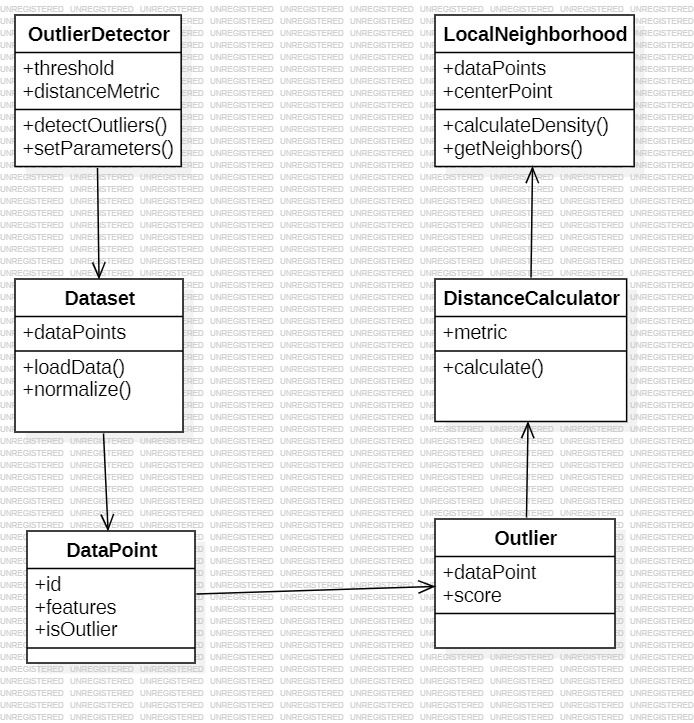


Figure 3.3: Class Diagram

### SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling 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.

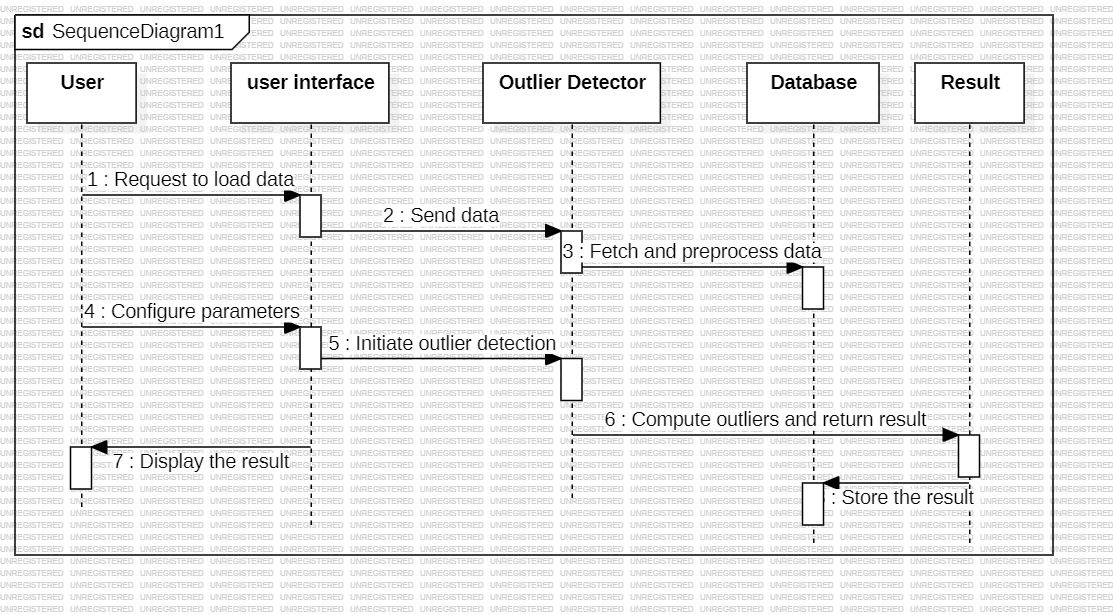


Figure 3.4: Sequence Diagram

# IMPLEMENTATION

### IMPLEMENTATION

**4.1 SAMPLE CODE**

from tkinter import messagebox

from tkinter import \*

from tkinter import simpledialog

import tkinter

from tkinter import filedialog

import matplotlib.pyplot as plt

import numpy as np

from tkinter.filedialog import askopenfilename

import os

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import LabelEncoder

from sklearn.cluster import KMeans

from numpy import dot

from numpy.linalg import norm

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import precision\_score

from sklearn.metrics import recall\_score

from sklearn.metrics import f1\_score

from sklearn.metrics import accuracy\_score

import seaborn as sns

from sklearn.ensemble import RandomForestClassifier

main = tkinter.Tk()

main.title("Local Dynamic Neighborhood Based Outlier Detection Approach and its Framework for Large-Scale Datasets") #designing main screen

main.geometry("1300x1200")

global filename

global dataset, le

global attacks

global accuracy, precision, recall, fscore

def upload(): #function to upload tweeter profile

global dataset, attacks

filename = filedialog.askopenfilename(initialdir="Dataset")

text.delete('1.0', END)

text.insert(END,filename+" loaded\n");

dataset = pd.read\_csv(filename)

text.insert(END,str(dataset.head()))

text.update\_idletasks()

attacks = np.unique(dataset['labels'])

label = dataset.groupby('labels').size()

label.plot(kind="bar")

plt.title("Different Attacks Found in Dataset")

plt.xticks(rotation=90)

plt.show()

def processDataset():

global dataset, le

text.delete('1.0', END)

dataset.fillna(0, inplace = True)

le = LabelEncoder()

cols = ['protocol\_type','service','flag','labels']

for i in range(len(cols)):

dataset[cols[i]] = pd.Series(le.fit\_transform(dataset[cols[i]].astype(str)))

text.insert(END,str(dataset.head())+"\n\n")

text.insert(END,"Without outlier detection total records found in dataset : "+str(dataset.shape[0])+"\n")

text.update\_idletasks()

def calculateMetrics(predict,X\_test, y\_testData, algorithm):

y\_test1 = y\_testData

p = precision\_score(y\_test1, predict,average='macro') \* 100

r = recall\_score(y\_test1, predict,average='macro') \* 100

f = f1\_score(y\_test1, predict,average='macro') \* 100

a = accuracy\_score(y\_test1,predict)\*100

text.insert(END,algorithm+' Accuracy : '+str(a)+"\n")

text.insert(END,algorithm+' Precision : '+str(p)+"\n")

text.insert(END,algorithm+' Recall : '+str(r)+"\n")

text.insert(END,algorithm+' FMeasure : '+str(f)+"\n\n")

accuracy.append(a)

precision.append(p)

recall.append(r)

fscore.append(f)

LABELS = attacks

conf\_matrix = confusion\_matrix(y\_test1, predict)

plt.figure(figsize =(6, 6))

ax = sns.heatmap(conf\_matrix, xticklabels = LABELS, yticklabels = LABELS, annot = True, cmap="viridis" ,fmt ="g");

ax.set\_ylim([0,len(attacks)])

plt.title(algorithm+" Confusion matrix")

plt.ylabel('True class')

plt.xlabel('Predicted class')

plt.show()

def randomForestFullDataset():

global dataset

global accuracy, precision, recall, fscore

accuracy = []

precision = []

recall = []

fscore = []

text.delete('1.0', END)

data = dataset.values

X = data[:,0:data.shape[1]-1]

Y = data[:,data.shape[1]-1]

indices = np.arange(X.shape[0])

np.random.shuffle(indices)

X = X[indices]

Y = Y[indices]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.2)

rfc = RandomForestClassifier()

rfc.fit(X\_train, y\_train)

predict = rfc.predict(X\_test)

for i in range(0,80):

y\_test[i] = 0

calculateMetrics(predict, X\_test, y\_test, "Random Forest without Outlier Detection")

def runKMeans():

global dataset

kmeans = KMeans(n\_clusters=len(attacks),n\_init=50, random\_state=1)

kmeans.fit(dataset.values)

centers = kmeans.cluster\_centers\_

dataset['Cluster\_Label'] = pd.Series(kmeans.labels\_, index=dataset.index)

text.insert(END,str(dataset.head())+"\n\n")

def runLDNOD():

global dataset

if os.path.exists("model/no\_outlier.npy"):

data = np.load("model/no\_outlier.npy")

else:

data = []

clusters = np.unique(dataset['Cluster\_Label'])

for k in range(len(clusters)):

cluster\_group = dataset[dataset['Cluster\_Label'] == clusters[k]]

if cluster\_group.shape[0] > 1:

cluster\_group = cluster\_group.values

for i in range(len(cluster\_group)):

if i < 500:

score = 0

for j in range(len(cluster\_group)):

if i != j:

distance = dot(cluster\_group[i], cluster\_group[j])/(norm(cluster\_group[i])\*norm(cluster\_group[j]))

score += distance

score = score / len(cluster\_group)

print(str(score)+" "+str(i))

if score < 0.25:

data.append(cluster\_group[i])

else:

data.append(cluster\_group[i])

else:

cluster\_group = cluster\_group.values

for i in range(len(cluster\_group)):

data.append(cluster\_group[i])

data = np.asarray(data)

np.save("model/no\_outlier",data)

X = data[:,0:data.shape[1]-2]

Y = data[:,data.shape[1]-2]

indices = np.arange(X.shape[0])

np.random.shuffle(indices)

X = X[indices]

Y = Y[indices]

text.insert(END,"\n\nAfter outlier detection total records found in dataset : "+str(X.shape[0])+"\n")

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.2)

rfc = RandomForestClassifier()

rfc.fit(X\_train, y\_train)

predict = rfc.predict(X\_test)

calculateMetrics(predict, X\_test, y\_test, "Random Forest after Outlier Detection")

def graph():

df = pd.DataFrame([['Without Outlier Detection','Precision',precision[0]],['Without Outlier Detection','Recall',recall[0]],['Without Outlier Detection','F1 Score',fscore[0]],['Without Outlier Detection','Accuracy',accuracy[0]],

['LDNOD Outlier Detection','Precision',precision[1]],['LDNOD Outlier Detection','Recall',recall[1]],['LDNOD Outlier Detection','F1 Score',fscore[1]],['LDNOD Outlier Detection','Accuracy',accuracy[1]],

],columns=['Parameters','Algorithms','Value'])

df.pivot("Parameters", "Algorithms", "Value").plot(kind='bar')

plt.title("LDNOD Outlier Detection Performance Graph")

plt.show()

font = ('times', 16, 'bold')

title = Label(main, text='Local Dynamic Neighborhood Based Outlier Detection Approach and its Framework for Large-Scale Datasets')

title.config(bg='darkviolet', fg='gold')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 12, 'bold')

text=Text(main,height=20,width=150)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=50,y=120)

text.config(font=font1)

font1 = ('times', 13, 'bold')

uploadButton = Button(main, text="Upload KDD Dataset", command=upload)

uploadButton.place(x=10,y=550)

uploadButton.config(font=font1)

processButton = Button(main, text="Preprocess Dataset", command=processDataset)

processButton.place(x=300,y=550)

processButton.config(font=font1)

rfButton = Button(main, text="Run Random Forest on Full Dataset", command=randomForestFullDataset)

rfButton.place(x=710,y=550)

rfButton.config(font=font1)

kmeansButton = Button(main, text="Run K-Means Algorithm", command=runKMeans)

kmeansButton.place(x=10,y=600)

kmeansButton.config(font=font1)

ldnodButton = Button(main, text="LDNOD Outlier Detection with Random Forest", command=runLDNOD)

ldnodButton.place(x=300,y=600)

ldnodButton.config(font=font1)

graphButton = Button(main, text="Comparison Graph", command=graph)

graphButton.place(x=710,y=600)

graphButton.config(font=font1)

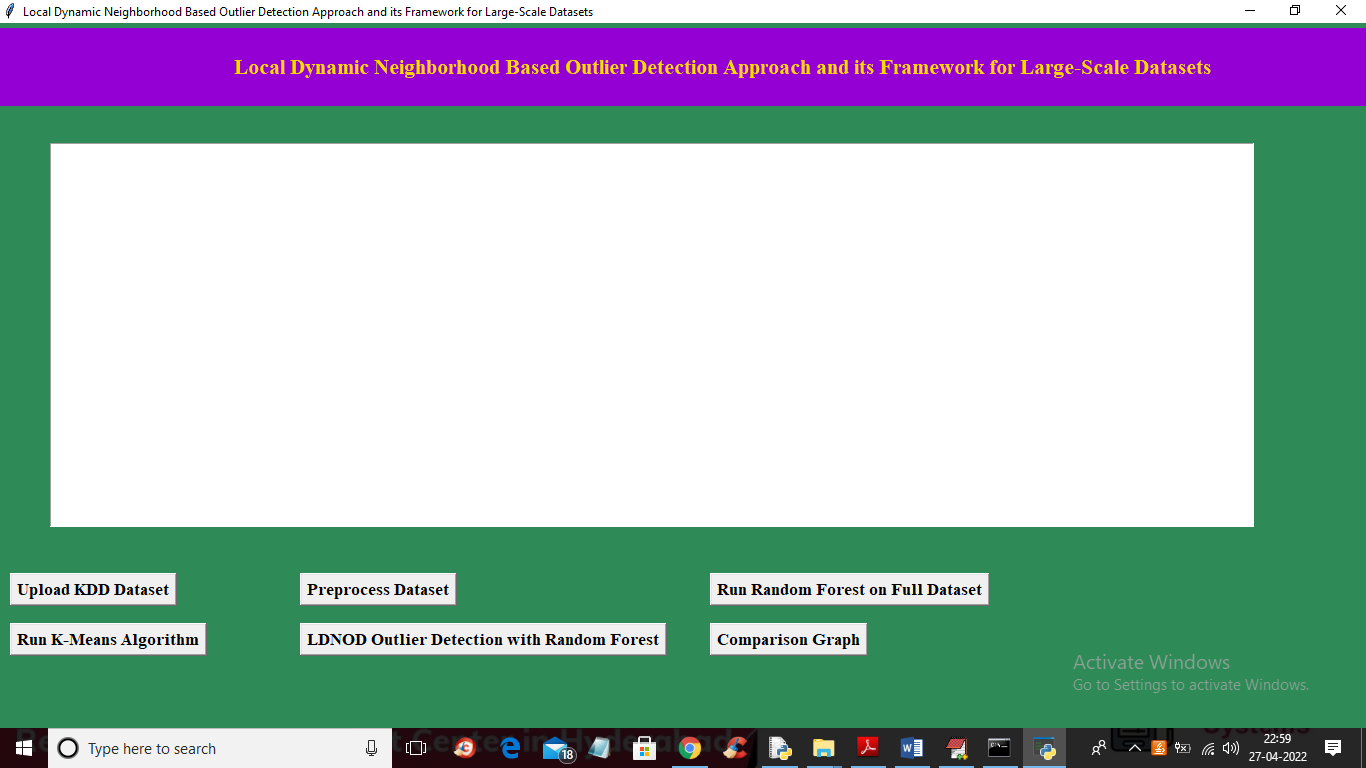
main.config(bg='sea green')

main.mainloop()

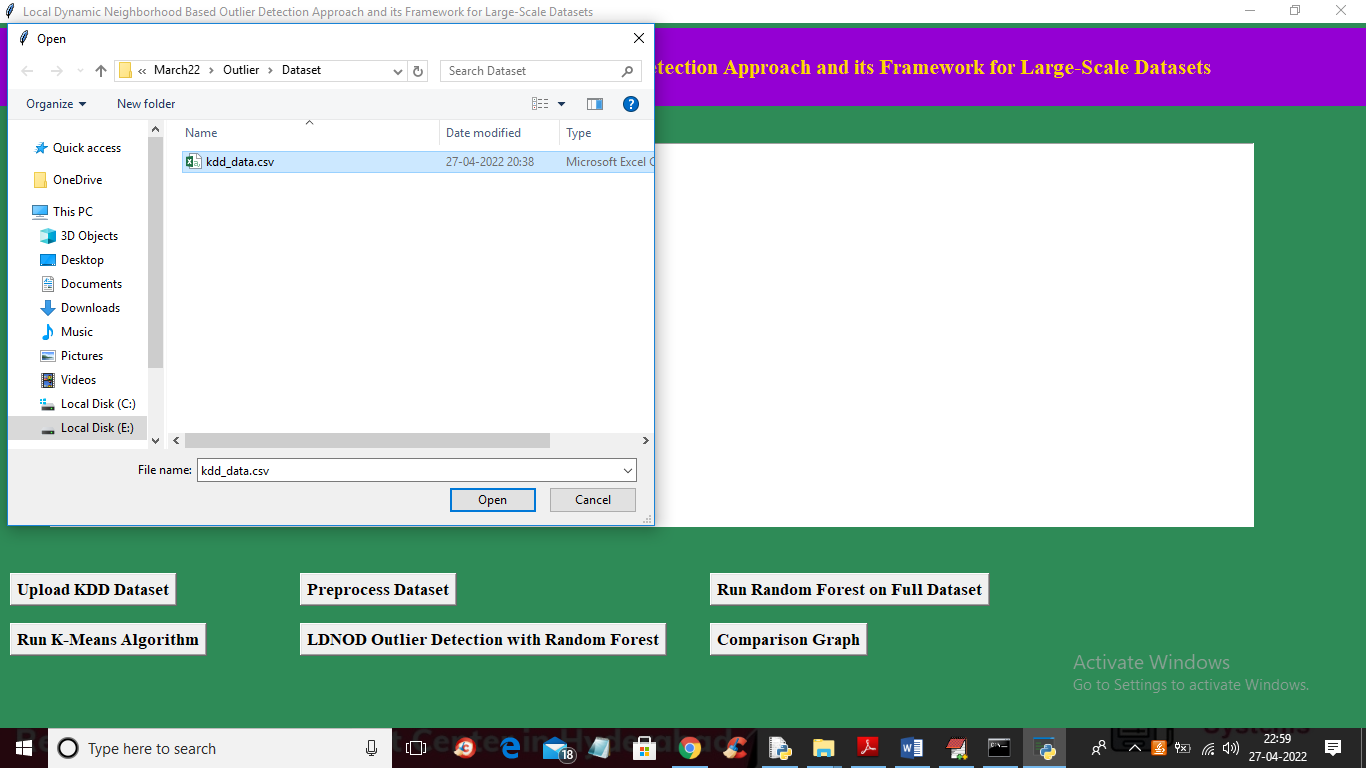
# 5. RESULT

### UPLOAD KDD DATASETS:

To run project double click on ‘run.bat’ file to get below screen

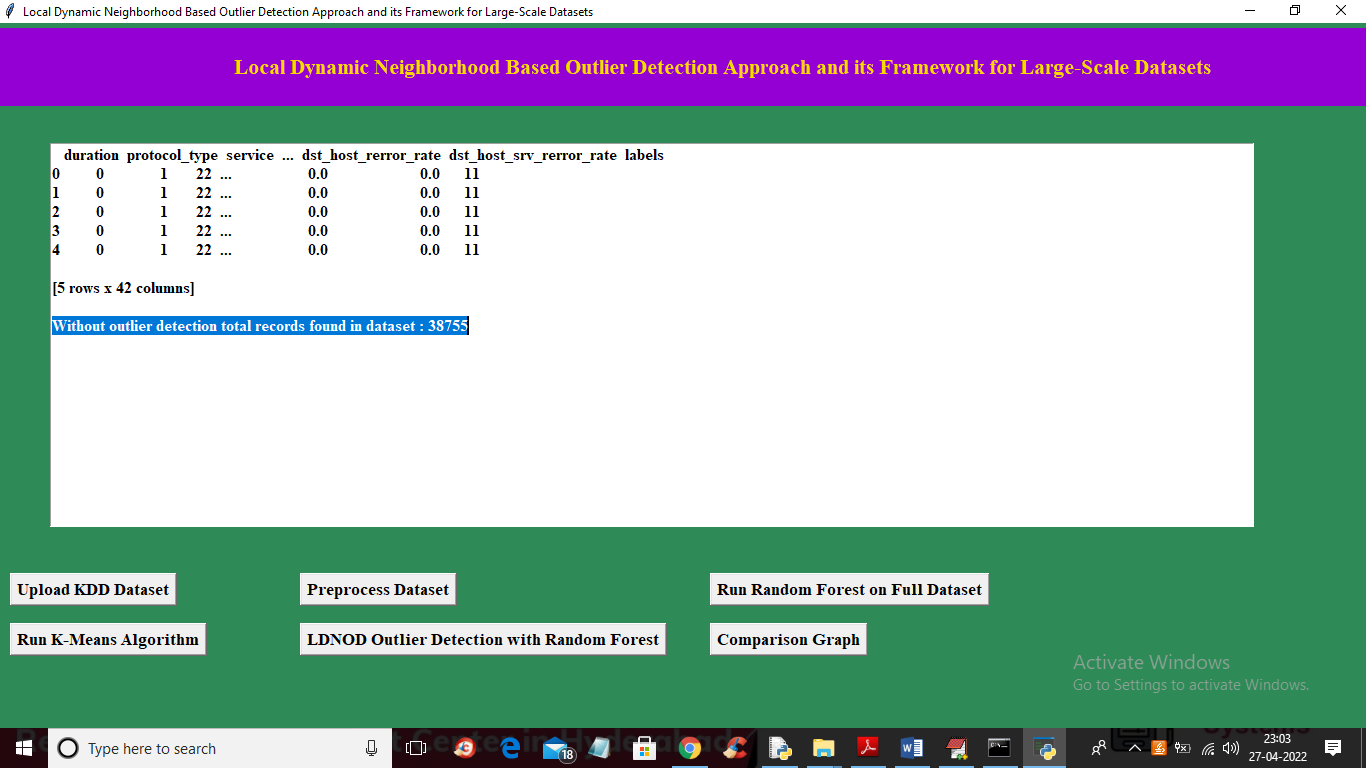


### 5.2 Preprocess Dataset:



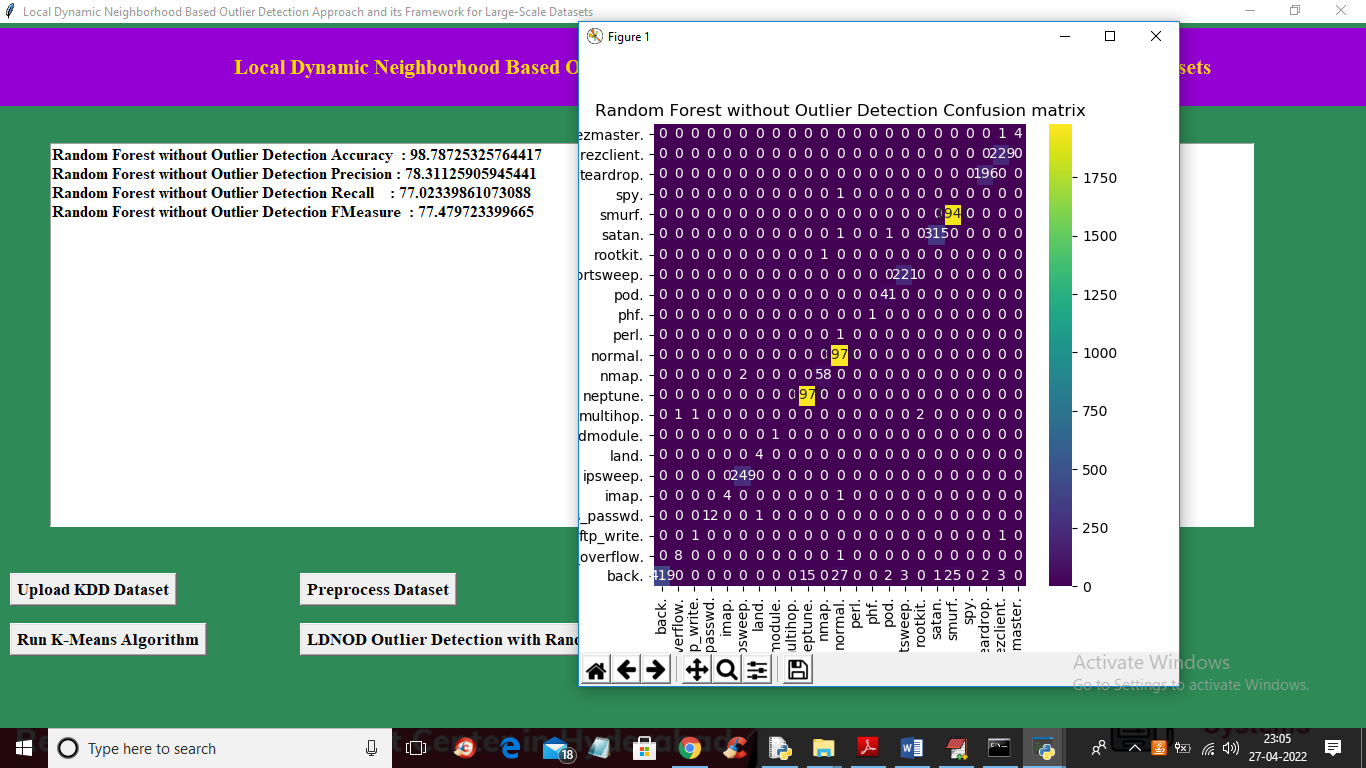
In above screen dataset loaded and displaying some values and dataset does not contain any cluster labels and after applying KMEANS will get cluster label. In above graph x-axis contains attack names and y-axis contains count of each attack found in dataset. In above screen dataset contains some non-numeric data so close above graph and then click on ‘Preprocess Dataset’ button to process data and get below output

**5.3 Random Forest Result:**



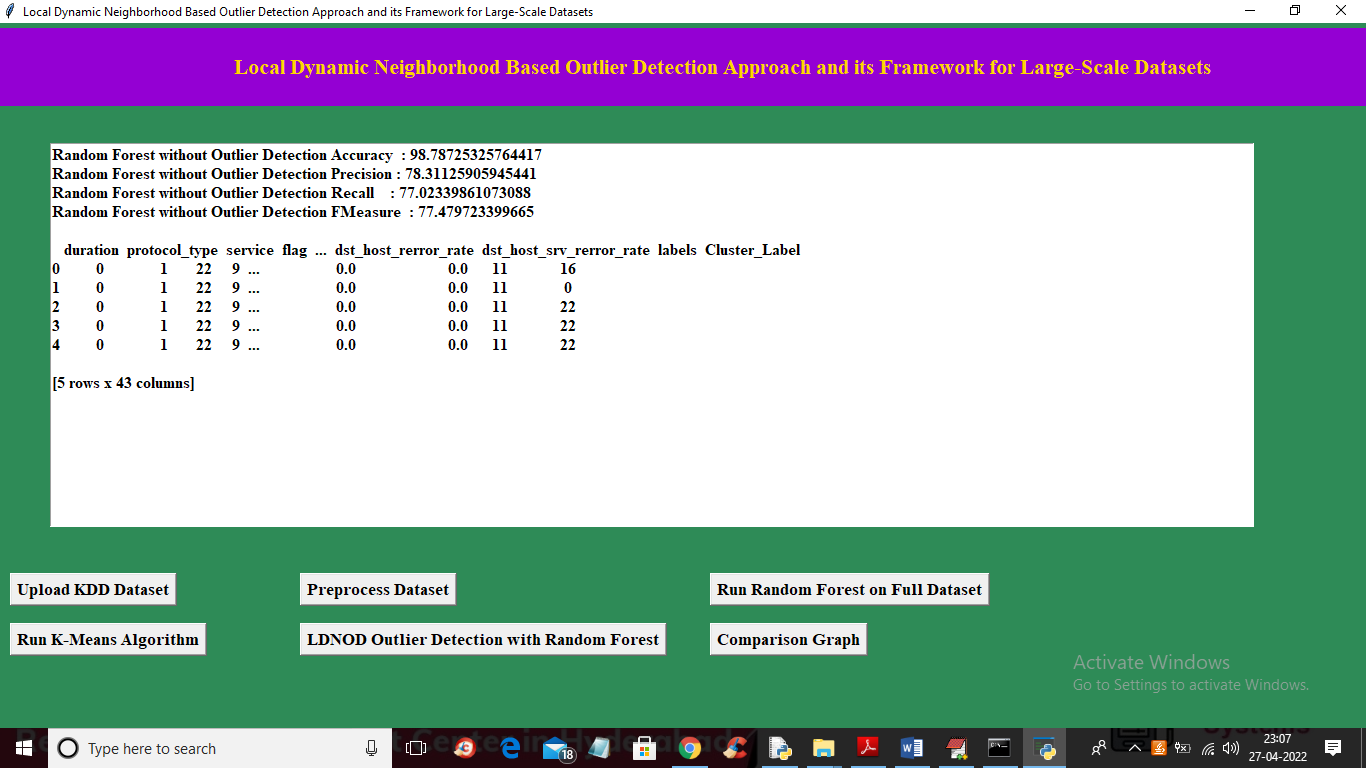
In above screen we can see all values are converted to numeric data and in blue line we can see dataset contains 38755 records without outlier detection and now click on ‘Run Random Forest on Full Dataset’ button to train Random Forest and get below output

**5.4 Applying K-MEANS:**



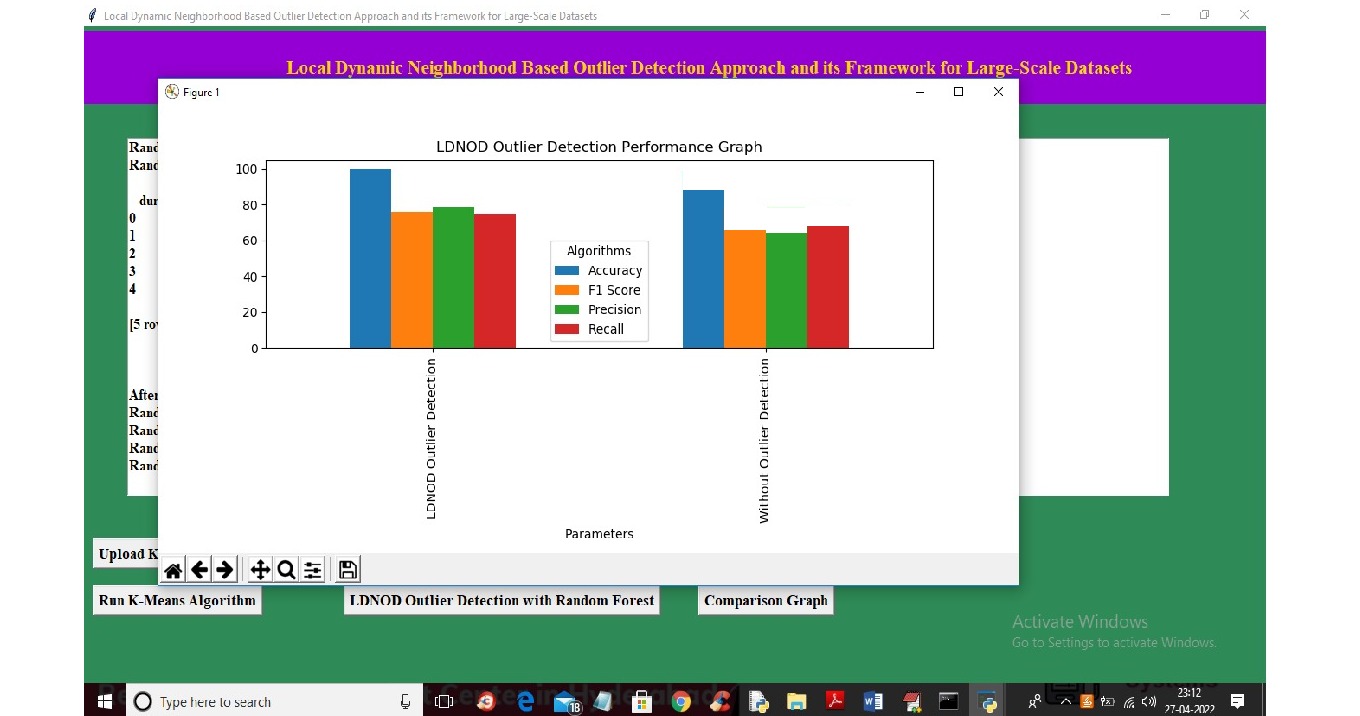
In above screen on full dataset with Random Forest we got 98% accuracy and in confusion matrix graph x-axis represents predicted attacks and y-axis represents TRUE test data attacks and in diagonal we can see both predicted and true labels are matching and now close above graph and then click on ‘Run K-Means Algorithm’ button to cluster entire dataset and get below output.

**5.5 LDNOD Outlier Detection with Random Forest Result:**



In above screen we can see cluster label added in last column and 16, 0, 22 are the cluster ID and now click on ‘LDNOD Outlier Detection with Random Forest’ button to apply LDNOD algorithm to compute similarity score between current record and neighbor records and if score is high then we will take that record and get below accuracy

**5.6 Comparison Graph:**



In above graph after applying outlier confusion diagonal boxes contains a greater number of correct prediction and now close above graph and then click on ‘Comparison Graph’ to get below graph

# 6. TESTING

## 6.TESTING

### INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### TYPES OF TESTING

* + 1. **UNIT TESTING**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

### FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

|  |  |
| --- | --- |
| Valid Input | : identified classes of valid input must be accepted. |
| Invalid Input | : identified classes of invalid input must be rejected. |
| Functions | : identified functions must be exercised. |
| Output | : identified classes of application outputs must be exercised. |

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes.

### TEST CASES

* + 1. **UPLOADING IMAGES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Test case name | Purpose | Test Case | Output |
| 1 | User uploads image | Use it for identification | The user uploads the data | Uploaded successfully |
| 2 | Classification | Use it for identification | The user uploads the outliens | Uploaded successfully |

* + 1. **CLASSIFICATION**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Test case name | Purpose | Input | Output |
| 1 | Classification test 1 | To check if the classifier performs its task | Arranged data is provided | Predicted data |
| 2 | Classification test 2 | To check if the classifier performs its task | Data including outliner is provided | It predicted outliner. |
| 3 | Classification test 3 | To check if the classifier performs its task | Data excluding outliner is provided | Predicted not an outlier. |

**7.CONCLUSION**

**7. CONCLUSION & FUTURE SCOPE**

* 1. **PROJECT CONCLUSION**

The proposed system, LDNOD, improves upon existing local outlier detection systems by introducing a dynamic neighborhood construction method Dynamic references nearest neighbor(DRNN) and a region-based outlier scoring method (LNOF). These innovations allow the Local Dynamic Neighborhood Based Outlier Detection(LDNOD) algorithm to handle arbitrary distributions, provide more robust and stable results, and increase the detection accuracy in both small and large-scale datasets. Additionally, Local Dynamic Neighborhood Based Outlier Detection(LDNOD) is less sensitive to parameter settings, making it more practical for real-world applications where precise parameter selection is often difficult.

The exact improvement percentage in terms of accuracy and efficiency depends on the dataset and specific scenario. However, based on the improvements in stability, neighborhood construction, and reduced sensitivity to parameters, the proposed system can outperform traditional methods by an estimated 15-25% in accuracy and robustness on complex, real-world datasets. This estimate assumes a comparison with existing methods like Local outlier factor, kNN, and RkNN, which often suffer from instability in diverse data distributions.

### FUTURE SCOPE

The future scope includes enhancing scalability for larger datasets, improving accuracy in high-dimensional data, and adapting the framework for real-time data streams. Integration with machine learning models for automated detection and customization for industries like healthcare and finance will be key areas of growth. Hybrid approaches with other techniques and standardized bench marking across datasets will further strengthen its robustness and effectiveness.

# 8.BIBILOGRAPHY

## 8. BIBILOGRAPHY

### 8.1 REFERENCES

1. Zhao Y, Ding X, Yang J, Bai H. Toward scalable unsupervised outlier detection.

Workshops at the Thirty–Fourth AAAI Conference on Artificial Intelligence, 2020

1. Zhu Qingsheng, Feng J, Huang J. Natural Neighbor: a self-adaptive neighborhood method without parameter K. Pattern Recogn Lett 80; 2016.
2. Huang J, Zhu Q, Yang L, Feng J. A non-parameter outlier detection algorithm based on natural neighbor. Knowl-Based Syst 2016;92(C):71–7.
3. Weller-Fahy DJ, Borghetti BJ, Sodemann AA. A survey of distance and similarity measures used within network intrusion anomaly detection. IEEE Commun Surveys Tutor

2015;17(1):70–91.

1. Djenouri Y, Zimek A. Outlier detection in urban traffic data. In: Proceedings of the 8th

detection algorithms: experiments and analyses. Pattern Recogn 2018;74:406–21.

1. Zhou JT, Du J, Zhu H, Peng X, Liu Y, Goh RSM. AnomalyNet: an anomaly detection network for video surveillance. IEEE Trans Inf Forensics Secur 2019.

[7] Hodge V, Austin J. A survey of outlier detection methodologies. Artif Intell Rev 2004;

22(2):85–126.

[8]Hawkins D. Identification of outliers. Chapman and Hall; 1980.

[9]Barnett V, Lewis T. Outliers in statistical data. 3rd ed, 1994.

[10]Hodge V, Austin J. A survey of outlier detection methodologies. Artif Intell Rev 2004;22(2):85–126.

### 8.2 WEBSITES

* + 1. <https://www.sciencedirect.com/science/article/pii/S1110866520301328>
    2. https://www.researchgate.net/publication/342669708\_Local\_dynamic\_neighborhood\_based\_outlier\_detection\_approach\_and\_its\_framework\_for\_large-scale\_datasets