A

Mini Project

On

Towards a Machine Learning-driven Trust Evaluation Model For Social Internet of Things: A Time-aware Approach

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

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

V.JAHNAVI (217R1A05R4)

B. SATWIK (217R1A05Q6)

J. VISHAL (217R1A05N1)

Under the Guidance of

K. RANJITH REDDY

(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 "**Towards a Machine Learning-driven Trust Evaluation Model for Social Internet of Things: A Time-aware Approach"** being submitted by **V.JAHNAVI (217R1A05R4), B.SATWIK (217R1A05Q6) and J.VISHAL (217R1A05N1)** 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 them under our guidance and supervision during the year 2024-25.

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

K. Ranjith Reddy
(Assistant Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. N. Bhaskar
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on _____

ACKNOWLEDGEMENT

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 our guide **Mr. K. Ranjith Reddy** Assistant Professor for his 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. J. Narasimha Rao, Mr. K. Ranjith Reddy, Dr. K. Maheswari, Mrs. K. Shilpa** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. N. Bhaskar,** Head of the department, 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.

V. JAHNAVI (217R1A05R4)

B. SATWIK (217R1A05Q6)

J. VISHAL (217R1A05N1)

ABSTRACT

As the Social Internet of Things (SIOT) becomes increasingly prevalent, establishing trust among interconnected devices and users becomes paramount. In this project, we propose a novel Machine Learning-driven Trust Evaluation Model (ML-TEM) tailored specifically for SIOT environments. Our approach integrates temporal dynamics into trust assessment, recognizing that trustworthiness can fluctuate over time due to evolving contexts and interactions. Leveraging a diverse array of features, including historical behavior, social relationships, and contextual information, our model employs advanced machine learning techniques to dynamically evaluate trust levels. We conduct extensive experiments using real-world SIOT datasets to validate the effectiveness and scalability of ML-TEM compared to existing trust evaluation methods. The results demonstrate that our time-aware approach significantly enhances trust assessment accuracy and adaptability in dynamic SIOT environments, thus contributing to the establishment of robust and reliable trust mechanisms crucial for fostering secure interactions within SIOT ecosystems

KEYWORDS: Machine learning, Behavioral patterns, Predictive analytics, Algorithms.

LIST OF FIGURES

LIST OF SCREEN SHOTS

TABLE OF CONTENTS

1. INTRODUCTION

INTRODUCTION

1.1 PROJECT SCOPE

The scope of this project encompasses the development of a Machine Learning-driven trust evaluation model tailored for the Social Internet of Things (SIOT), focusing on dynamic and adaptive trust assessments. Key activities include collecting diverse interaction data from SIOT devices and users, designing algorithms that incorporate temporal and contextual information, and establishing a robust evaluation framework to validate the model's effectiveness.

1.2 PROJECT PURPOSE

This project includes several innovative features, such as dynamic trust scoring that continuously updates based on real-time interactions. It employs time-aware analysis, considering how trust evolves, and incorporates contextual adaptability to account for variables like device type and user roles. Additionally, the model utilizes various machine learning techniques to enhance predictive accuracy and provides user-friendly visualization tools for a better understanding of trust dynamics.

1.3 PROJECT FEATURES

The primary purpose of this project is to enhance the security and reliability of interactions within the SIOT by improving the accuracy of trust evaluations. It aims to facilitate collaboration among devices and users, adapt to changing behaviors, and create a more intuitive user experience. Furthermore, the project seeks to contribute to the broader field of trust evaluation in IoT, offering insights and methodologies that can be applied to diverse SIOT scenarios.

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

The system analysis for the Machine Learning-driven trust evaluation model in the Social Internet of Things (SIOT) focuses on understanding the interactions among devices, users, and data architecture. It identifies key components such as data sources for user behavior and contextual information vital for trust assessments. The analysis evaluates machine learning algorithms for dynamically updating trust scores, while addressing security and privacy concerns. Additionally, it outlines the design of an intuitive user interface for visualizing trust metrics, ensuring users can easily interpret and manage their interactions. This analysis lays the groundwork for a robust and adaptive trust evaluation system tailored to the dynamic SIOT ecosystem.

2.1 PROBLEM DEFINITION

The increasing interconnectivity of devices in the Social Internet of Things (SIOT) presents significant challenges in establishing trust among diverse entities. Users and devices often lack reliable mechanisms to evaluate the trustworthiness of each other, leading to potential security risks, data breaches, and compromised interactions. Traditional trust assessment methods are often static and fail to adapt to the dynamic nature of SIOT, where relationships and user behaviors change frequently. Additionally, the absence of time-awareness in these evaluations limits their effectiveness in capturing the evolving nature of trust. This project aims to address these challenges by developing a Machine Learning-driven trust evaluation model that dynamically assesses trust levels based on real-time interactions and historical data, incorporating temporal and contextual factors to ensure accurate and reliable trust evaluations in a rapidly changing SIOT environment.

2.2 EXISTING SYSTEM

The existing systems for trust evaluation in the Social Internet of Things (SIOT) typically rely on static models that utilize predefined trust metrics and fixed criteria to assess the reliability of devices and users. These approaches often lack adaptability, failing to account for the dynamic and context-dependent nature of interactions in SIOT environments. Many existing models employ simplistic algorithms that do not incorporate temporal data, resulting in outdated trust assessments that can lead to erroneous conclusions. Furthermore, current systems may not effectively handle the vast amount of data generated by diverse devices, limiting their predictive accuracy. As a result, users face challenges in determining the trustworthiness of entities in real time, exposing them to security vulnerabilities and undermining the overall reliability of SIOT applications.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

- Data Quality and Availability
- Scalability Issues
- Model Complexity
- Dependence on Historical Data
- Privacy Concerns
- User Acceptance and Trust

2.3 PROPOSED SYSTEM

The proposed system aims to develop a Machine Learning-driven trust evaluation model that dynamically assesses trust levels in the Social Internet of Things (SIOT) by integrating real-time interaction data, historical behavior, and contextual information. This model utilizes advanced machine learning algorithms to process temporal data, allowing for continuous updates of trust scores that reflect the evolving nature of relationships among devices and users. By incorporating time-awareness and adaptability, the system provides more accurate and reliable trust evaluations, thereby enhancing security and facilitating safer interactions. Additionally, an intuitive user interface will be designed to visualize trust metrics, empowering users to make informed decisions about their interactions in the SIOT ecosystem.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- Time-Aware Trust Evaluation
- Dynamic Trust Adaptation
- Enhanced Security
- Contextual Trust Analysis
- Efficient Decision-Making
- Improved Interoperability

2.4 FEASIBILITY STUDY

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 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. All the resources are already available, it give an indication of the system is economically possible for development.

2.4.2 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.

2.4.3 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.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 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 : Pentium IV 2.4 GHz
- Hard Disk : 40GB and Above
- Memory : 512MB

2.5.2 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, 10.
- Languages : Python (Version 3.7.0)

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCITECTURE

This project architecture shows the procedure followed for trust evaluation model foe SIOT using machine learning, starting from input to final prediction.

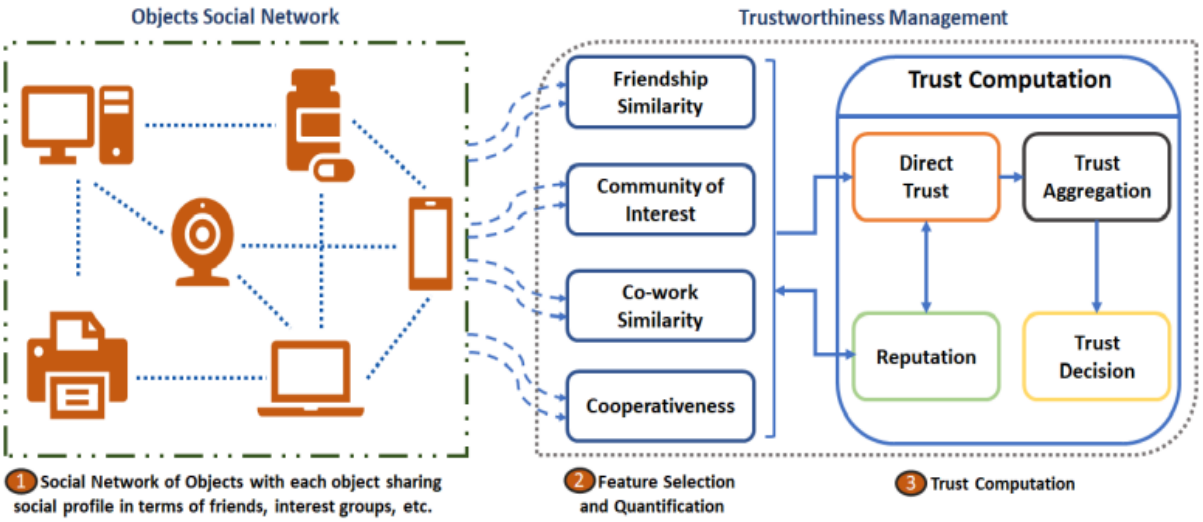


Figure 3.1: Project Architecture of Trust evaluation model using Machine learning

3.2 DESCRIPTION

Input Data: Input data is generally in .csv file format where the data is fetched and mapped in the data framed from the source columns.

Importing Data: Convert the dataset into a usable format and preview it. After uploading the dataset, this module reads the CSV file into a pandas Dataframe, which is a tabular data structure suited for data manipulation and analysis.

Preprocess: Prepare and clean the data, encode categorical variables, and visualize data distribution. To achieve the efficiency in computation we are going to normalize and clean the data values.

Training and test data: Training data is passed to the Bagging Classifier to train the model. Test data is used to test the trained model whether it is making correct predictions or not.

Run Algorithms: The purpose of choosing the Bagging classifier for this project the efficiency and accuracy and also run different algorithms to compare their accuracy.

3.3 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 upload, preprocess and test the data and to view the results. Whereas the all the process done within the system so results are stored or displayed by the system.

Figure 3.2: Trust evaluation model for SIOT using Machine Learning

3.4 CLASS DIAGRAM

Class Diagram is a collection of classes and objects.

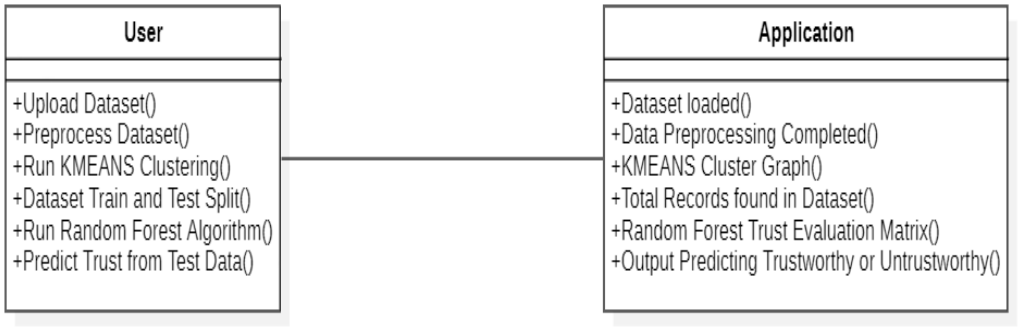


Figure 3.3: Class Diagram for Trust evaluation model using Machine Learning

3.5 SEQUENCE DIAGRAM

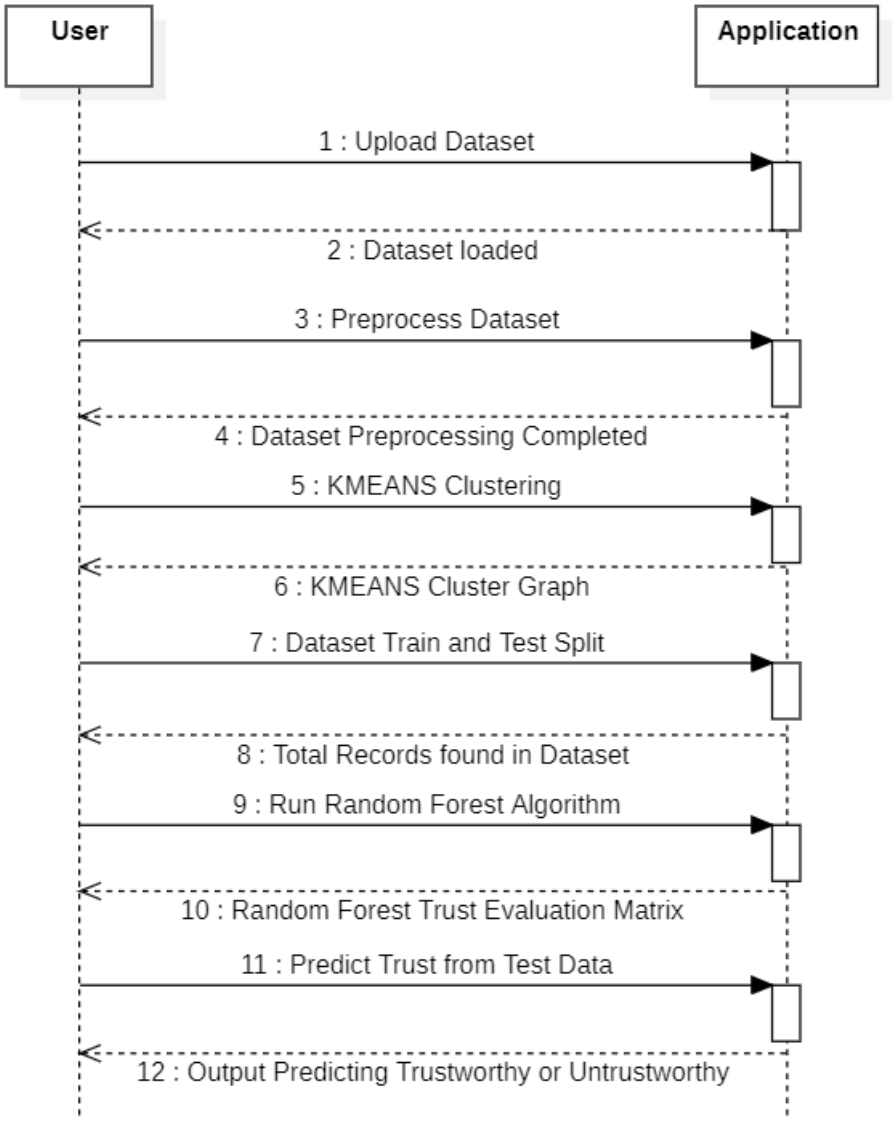


Figure 3.4: Sequence Diagram for Trust evaluation model for SIOT using Machine Learning

3.6 ACTIVITY DIAGRAM

It describes about flow of activity states.

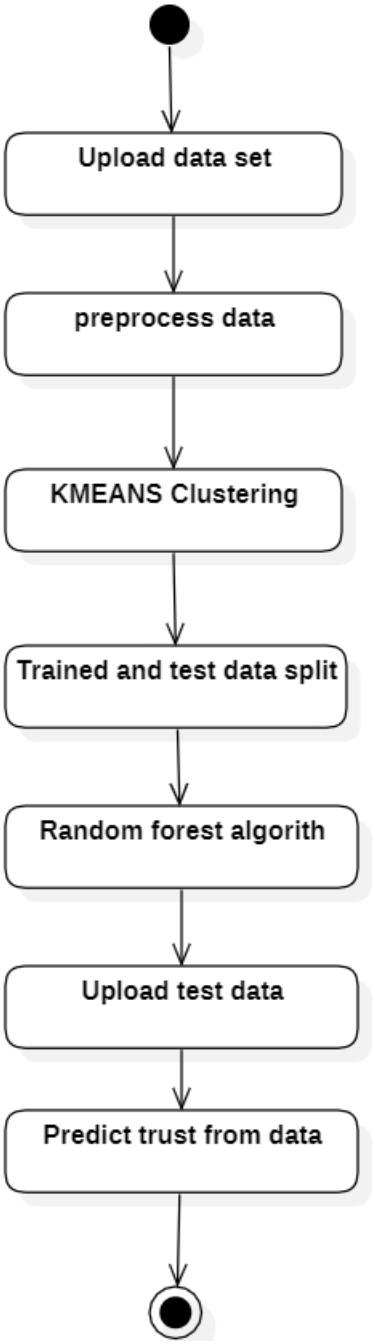


Figure 3.5: Activity Diagram for Trus using Machine Learning

4. IMPLEMENTATION

4.1 SAMPLE CODE

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
from tkinter.filedialog import askopenfilename
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
from sklearn import tree
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import LinearSVC
from sklearn.metrics import classification_report,f1_score,precision_score,recall_score
from sklearn.linear_model import LogisticRegression
import seaborn as sns
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import BaggingClassifier

main = tkinter.Tk()
main.title("Consumer COnduct")
main.geometry("1300x1200")

global le1, le2, le3, le4, cls, extension_acc

def upload():
    global filename
    global data
    text.delete('1.0', END)
    filename = askopenfilename(initialdir = "Dataset")
```

```
    pathlabel.config(text=filename)
    text.insert(END,"Dataset loaded\n\n")

def importdata():
    global filename
    global df
    df = pd.read_csv(filename,encoding = 'latin1')
    text.insert(END,"Data Information:\n"+str(df.head())+"\n")
    text.insert(END,"Columns Information:\n"+str(df.columns)+"\n")

def preprocess():
    global df
    global x,y
    X = df.drop(columns=['Clicked on Ad'])
    label_names = np.array(['No','Yes'])
    y = df['Clicked on Ad'].values
    feature_names = np.array(list(X))
    x = np.array(X)
    sns.countplot(df["Clicked on Ad"])
    plt.show()

def plotCorrelationMatrix(df, graphWidth):
    #filename = df.dataframeName
    df = df.dropna('columns') # drop columns with NaN
    df = df[[col for col in df if df[col].nunique() > 1]] # keep columns where there are
more than 1 unique values
    if df.shape[1] < 2:
        text.insert(END,f'No correlation plots shown: The number of non-NaN or constant
columns ({df.shape[1]}) is less than 2')
        return
    corr = df.corr()
    plt.figure(num=None, figsize=(graphWidth, graphWidth), dpi=80, facecolor='w',
edgecolor='k')
    corrMat = plt.matshow(corr, fignum = 1)
    plt.xticks(range(len(corr.columns)), corr.columns, rotation=90)
    plt.yticks(range(len(corr.columns)), corr.columns)
```

```
    plt.gca().xaxis.tick_bottom()
    plt.colorbar(corrMat)
    plt.show()

def ttmodel():
    global le1, le2, le3, le4
    global x,y
    global df
    global X_train,X_test,y_train,y_test
    X_train,X_test,y_train,y_test = train_test_split(x,y,test_size=0.33,random_state=5)
    le1 = LabelEncoder()
    le2 = LabelEncoder()
    le3 = LabelEncoder()
    le4 = LabelEncoder()

    x[:,4] = le1.fit_transform(x[:,4])
    x[:,5] = le2.fit_transform(x[:,5])
    x[:,7] = le3.fit_transform(x[:,7])
    x[:,8] = le4.fit_transform(x[:,8])

    X_train[:,4] = le1.fit_transform(X_train[:,4])
    X_train[:,5] = le2.fit_transform(X_train[:,5])
    X_train[:,7] = le3.fit_transform(X_train[:,7])
    X_train[:,8] = le4.fit_transform(X_train[:,8])

    X_test[:,4] = le1.fit_transform(X_test[:,4])
    X_test[:,5] = le2.fit_transform(X_test[:,5])
    X_test[:,7] = le3.fit_transform(X_test[:,7])

    X_test[:,8] = le4.fit_transform(X_test[:,8])

    text.insert(END,"Train Shape: "+str(X_train.shape)+"\n")
    text.insert(END,"Test Shape: "+str(X_test.shape)+"\n")

    plotCorrelationMatrix(df, len(df.columns))
```

```
def mlmodels():
    global x,y
    global X_train,X_test,y_train,y_test,cls, extension_acc
    global lr_acc,svc_acc,rfc_acc,gnb_acc,dtc_acc
    clf_lr = LogisticRegression(random_state=0)
    clf_lr.fit(X_train,y_train)
    pred = clf_lr.predict(X_test)
    lr_acc=clf_lr.score(X_test, y_test)
    text.insert(END,"LOGIT Accuracy: "+str(clf_lr.score(X_test, y_test))+"\n")
    text.insert(END,"LOGIT recall_score: "+str(recall_score(y_test,pred))+"\n")
    text.insert(END,"LOGIT precision_score: "+str(precision_score(y_test,pred))+"\n")
    text.insert(END,"LOGIT f1_score: "+str(f1_score(y_test,pred))+"\n\n")

    clf_svc = LinearSVC(random_state=0)
    clf_svc.fit(X_train,y_train)
    clf_svc.score(X_test,y_test)
    pred = clf_svc.predict(X_test)
    svc_acc=clf_svc.score(X_test, y_test)
    text.insert(END,"SVC Accuracy: "+str(clf_svc.score(X_test, y_test))+"\n")
    text.insert(END,"SVC recall_score: "+str(recall_score(y_test,pred))+"\n")
    text.insert(END,"SVC precision_score: "+str(precision_score(y_test,pred))+"\n")
    text.insert(END,"SVC f1_score: "+str(f1_score(y_test,pred))+"\n\n")

    clf_gnb = GaussianNB()
    clf_gnb.fit(X_train,y_train)
    clf_gnb.score(X_test,y_test)
    pred = clf_gnb.predict(X_test)
    gnb_acc=clf_gnb.score(X_test, y_test)
    text.insert(END,"Naive Bayes Accuracy: "+str(clf_gnb.score(X_test, y_test))+"\n")
    text.insert(END,"Naive Bayes recall_score: "+str(recall_score(y_test,pred))+"\n")
    text.insert(END,"Naive Bayes precision_score:
"+str(precision_score(y_test,pred))+"\n")
    text.insert(END,"Naive Bayes f1_score: "+str(f1_score(y_test,pred))+"\n\n")

    clf_rfc = RandomForestClassifier(random_state=0)
    clf_rfc.fit(X_train,y_train)
```

```
    clf_rfc.score(X_test,y_test)
    pred = clf_rfc.predict(X_test)
    rfc_acc=clf_rfc.score(X_test, y_test)
    text.insert(END,"Random Forest Accuracy: "+str(clf_gnb.score(X_test, y_test))+"\n")
    text.insert(END,"Random Forest recall_score: "+str(recall_score(y_test,pred))+"\n")
    text.insert(END,"Random Forest precision_score:
"+str(precision_score(y_test,pred))+"\n")
    text.insert(END,"Random Forest f1_score: "+str(f1_score(y_test,pred))+"\n\n")

    clf_dtc = DecisionTreeClassifier(random_state=0)
    clf_dtc.fit(X_train,y_train)
    clf_dtc.score(X_test,y_test)
    pred = clf_dtc.predict(X_test)
    dtc_acc=clf_rfc.score(X_test, y_test)
    text.insert(END,"Decision Tree Accuracy: "+str(clf_dtc.score(X_test, y_test))+"\n")
    text.insert(END,"Decision Tree recall_score: "+str(recall_score(y_test,pred))+"\n")
    text.insert(END,"Decision Tree precision_score:
"+str(precision_score(y_test,pred))+"\n")
    text.insert(END,"Decision Tree f1_score: "+str(f1_score(y_test,pred))+"\n\n")

    cls = BaggingClassifier()
    cls.fit(x,y)
    cls.score(X_test,y_test)
    pred = cls.predict(X_test)
    extension_acc =cls.score(X_test, y_test)
    text.insert(END,"Extension Bagging Classifier Accuracy: "+str(extension_acc)+"\n")
    text.insert(END,"Extension Bagging Classifier recall_score:
"+str(recall_score(y_test,pred))+"\n")
    text.insert(END,"Extension Bagging Classifier precision_score:
"+str(precision_score(y_test,pred))+"\n")
    text.insert(END,"Extension Bagging Classifier f1_score:
"+str(f1_score(y_test,pred))+"\n\n")

def predict():
    global cls
    global le1, le2, le3, le4
```

```
    text.delete('1.0', END)
    filename = filedialog.askopenfilename(initialdir="Dataset")
    text.insert(END,filename+" loaded\n\n")
    dataset = pd.read_csv(filename,encoding='latin1')
    dataset.fillna(0, inplace = True)
    dataset = dataset.values
    XX = dataset[:,0:dataset.shape[1]]
    XX[:,4] = le1.fit_transform(XX[:,4])
    XX[:,5] = le2.fit_transform(XX[:,5])
    XX[:,7] = le3.fit_transform(XX[:,7])
    XX[:,8] = le4.fit_transform(XX[:,8])

    prediction = cls.predict(XX)
    print(prediction)
    for i in range(len(prediction)):
        if prediction[i] == 0:
            text.insert(END,"Test DATA : "+str(dataset[i])+" ===> PREDICTED AS
CONSUMER NOT CLICKED ON ADD\n\n")
        if prediction[i] == 1:
            text.insert(END,"Test DATA : "+str(dataset[i])+" ===> PREDICTED AS
CONSUMER CLICKED ON ADD\n\n")

def graph():
    global lr_acc,svc_acc,rfc_acc,gnb_acc,dtc_acc, extension_acc

    height = [lr_acc,svc_acc,rfc_acc,gnb_acc,dtc_acc, extension_acc]
    bars = ('Logit', 'SVC','RFC','GNB','DT', 'Extension BaggingClassifier')
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, height)
    plt.xticks(y_pos, bars)
    plt.show()

font = ('times', 16, 'bold')
title = Label(main, text='A Model for prediction of consumer conduct using machine
learning algorithm')
title.config(bg='dark salmon', fg='black')
```

```
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)

font1 = ('times', 14, 'bold')
upload = Button(main, text="Upload Dataset", command=upload)
upload.place(x=900,y=100)
upload.config(font=font1)

pathlabel = Label(main)
pathlabel.config(bg='dark orchid', fg='white')
pathlabel.config(font=font1)
pathlabel.place(x=900,y=150)

ip = Button(main, text="Data Import", command=importdata)
ip.place(x=900,y=200)
ip.config(font=font1)

pp = Button(main, text="Data Preprocessing", command=preprocess)
pp.place(x=900,y=250)
pp.config(font=font1)

tt = Button(main, text="Train and Test Model", command=ttmodel)
tt.place(x=900,y=300)
tt.config(font=font1)

ml = Button(main, text="Run Algorithms", command=mlmodels)
ml.place(x=900,y=350)
ml.config(font=font1)

gph = Button(main, text="Accuracy Graph", command=graph)
gph.place(x=900,y=400)
gph.config(font=font1)

predictButton = Button(main, text="Predict Consumer Conduct from Test Data",
command=predict)
```

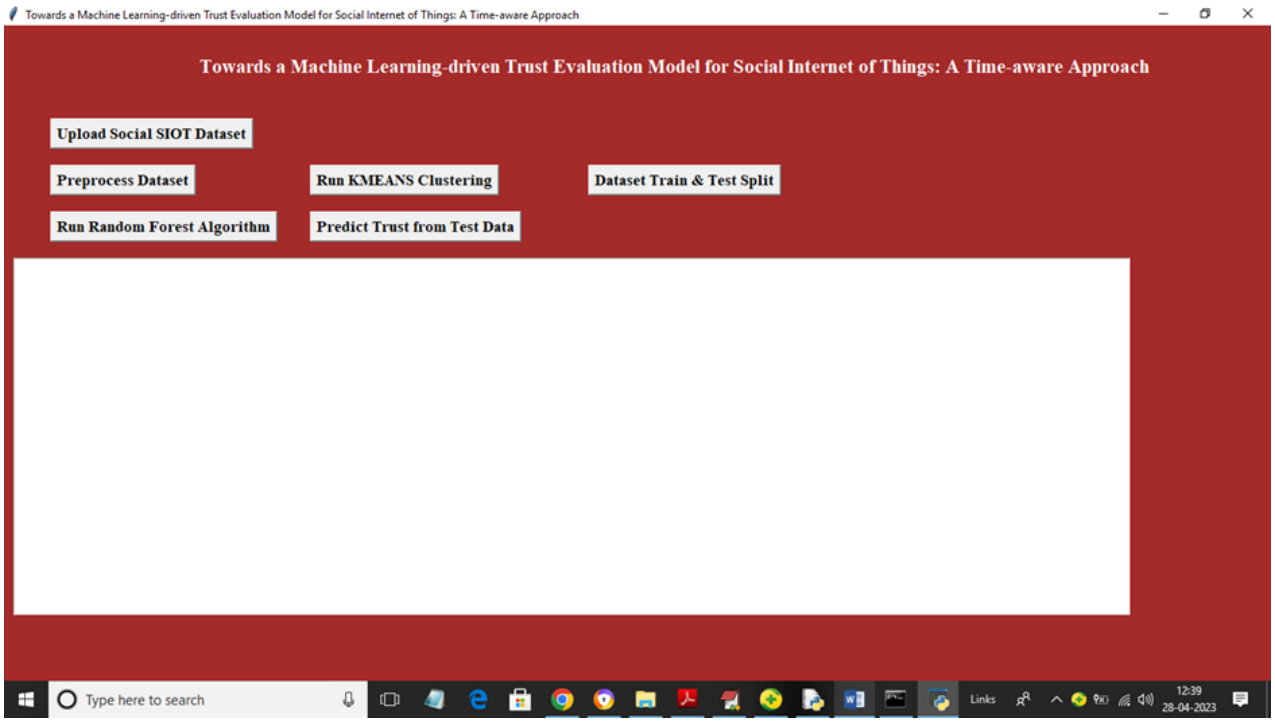
```
predictButton.place(x=900,y=450)
predictButton.config(font=font1)

font1 = ('times', 12, 'bold')
text=Text(main,height=30,width=110)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=10,y=100)
text.config(font=font1)

main.config(bg='peach puff')
main.mainloop()
```

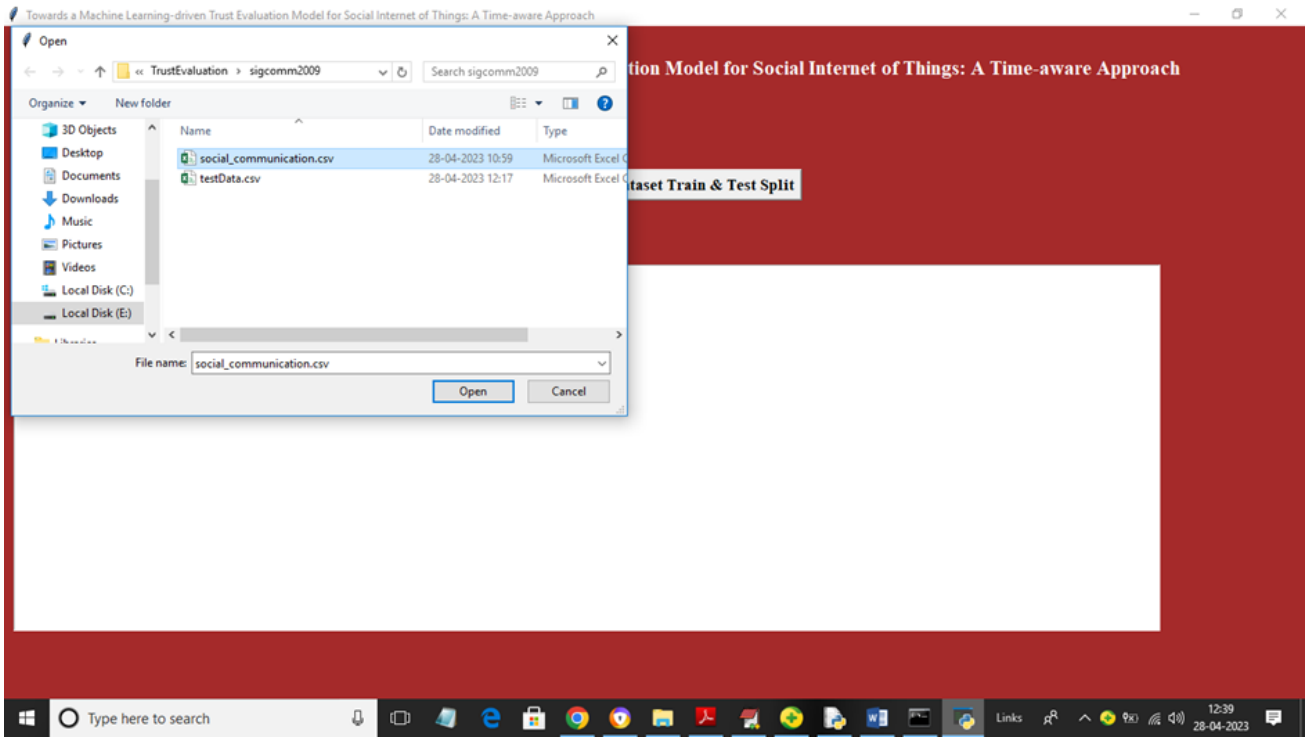
4. SCREENSHOTS

5.1 UPLOAD DATA RESULT



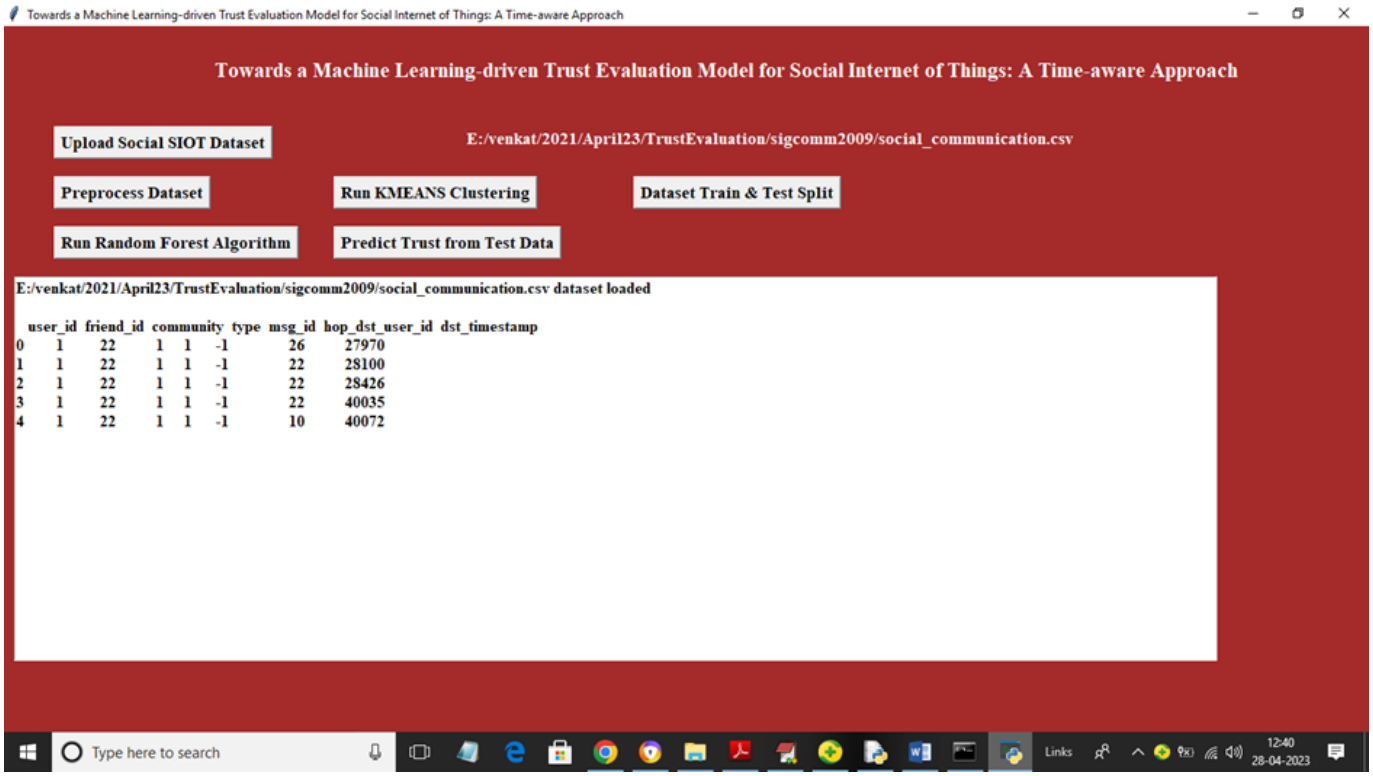
Screenshot 5.1: Upload data result of **TRUST EVALUATION MODEL** using Machine Learning

5.2 IMPORT DATA



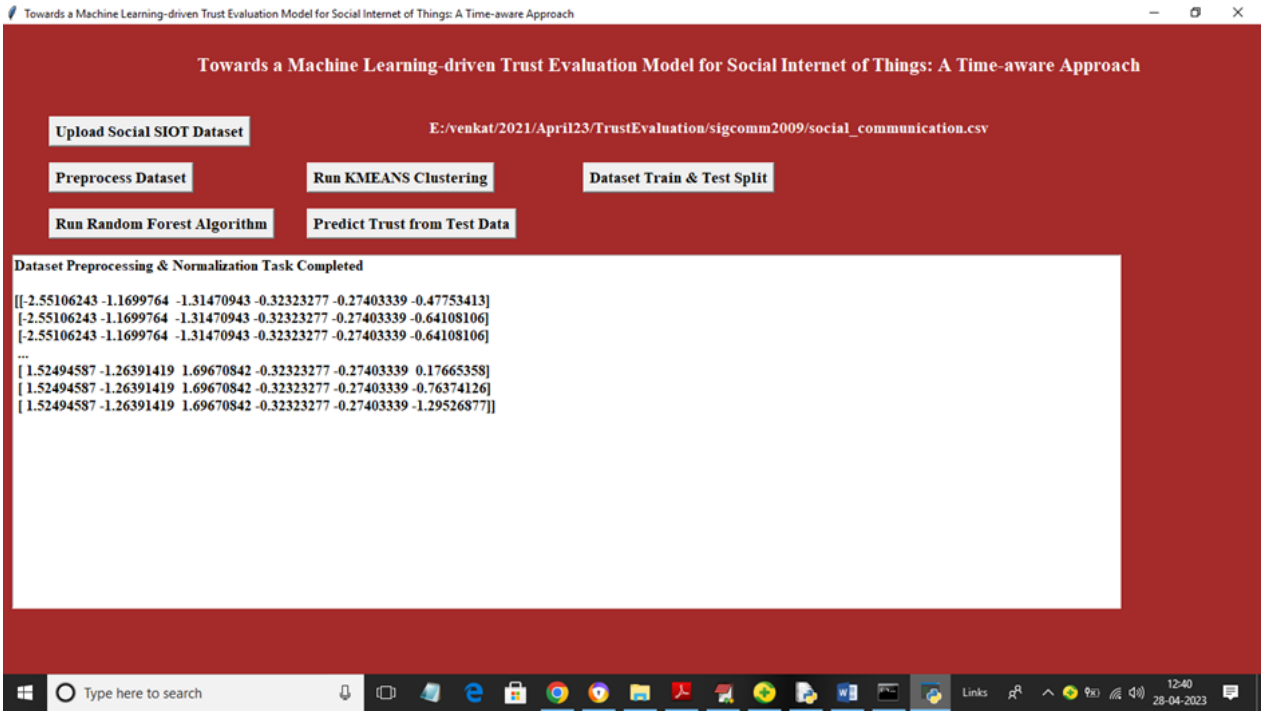
Screenshot 5.2: Import data result of TRUST EVALUATION MODEL using Machine Learning

5.3 PREPROCESS DATA RESULT



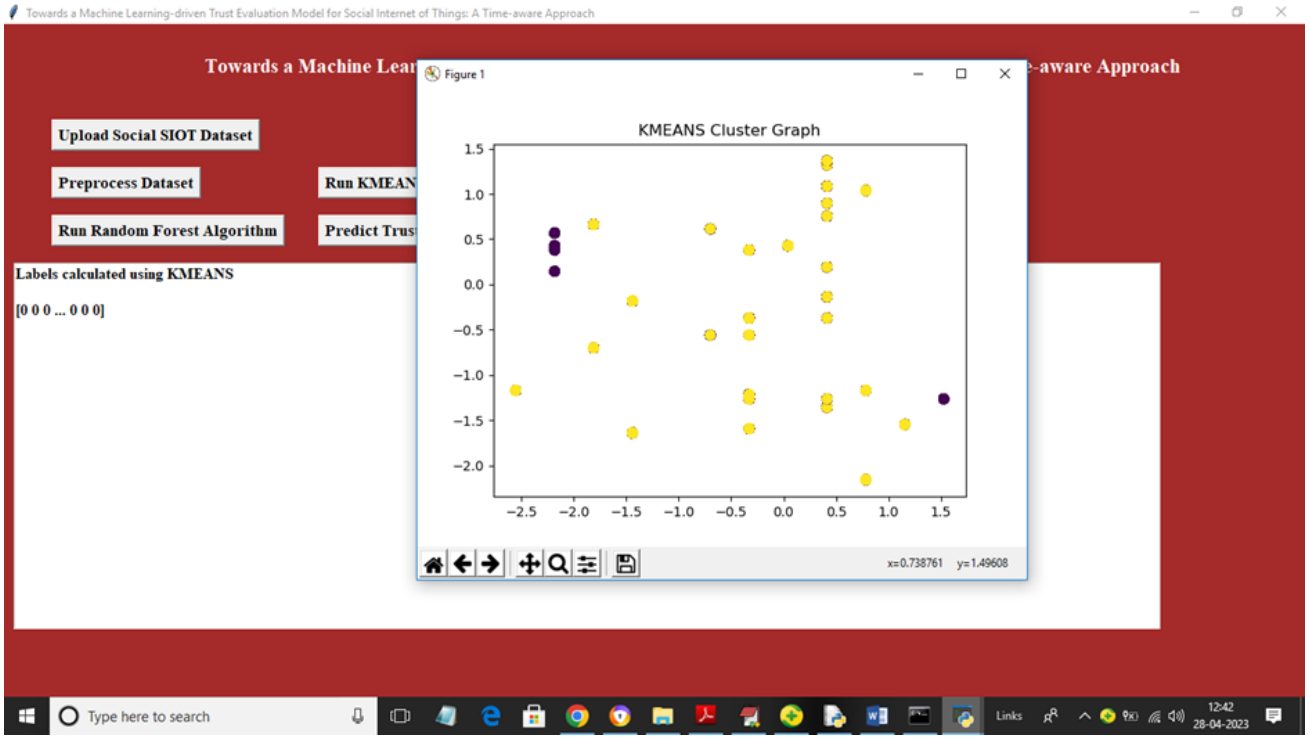
Screenshot 5.3: Preprocess data result of TRUST EVALUATION MODEL using Machine Learning

5.4 TRAIN AND TEST DATA RESULT



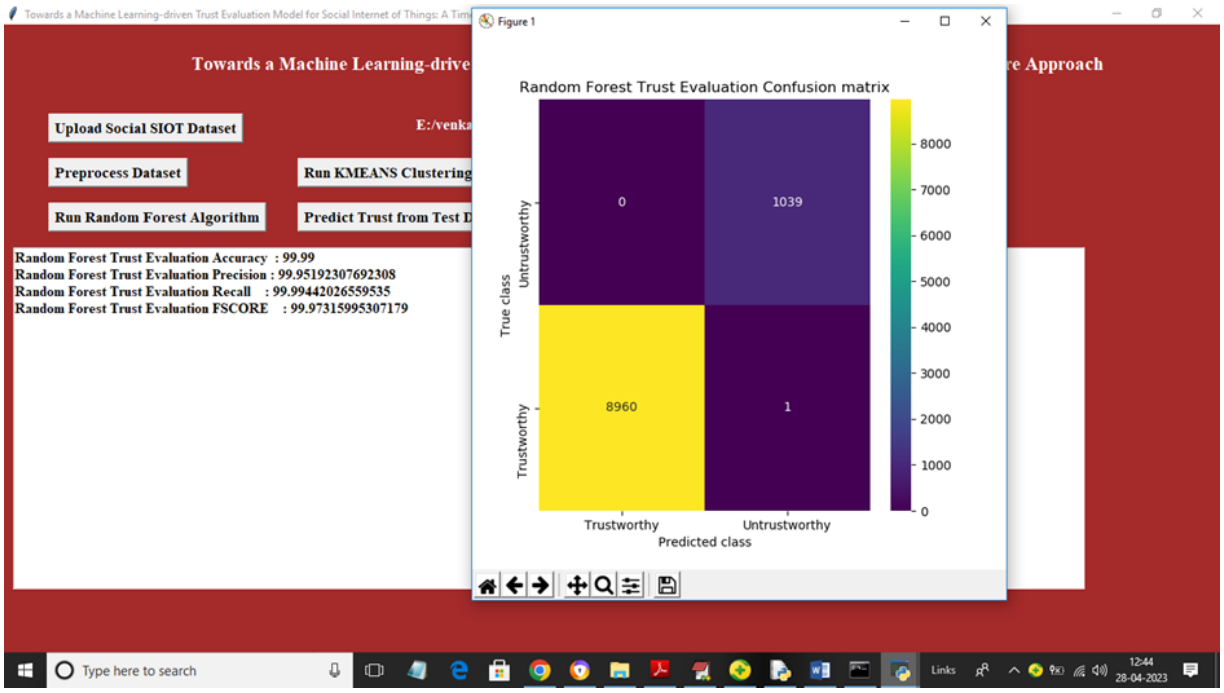
Screenshot 5.4: Run KMEANS on data result of TRUST EVALUATION MODEL using Machine Learning

5.5 RUN ALGORITHMS RESULT



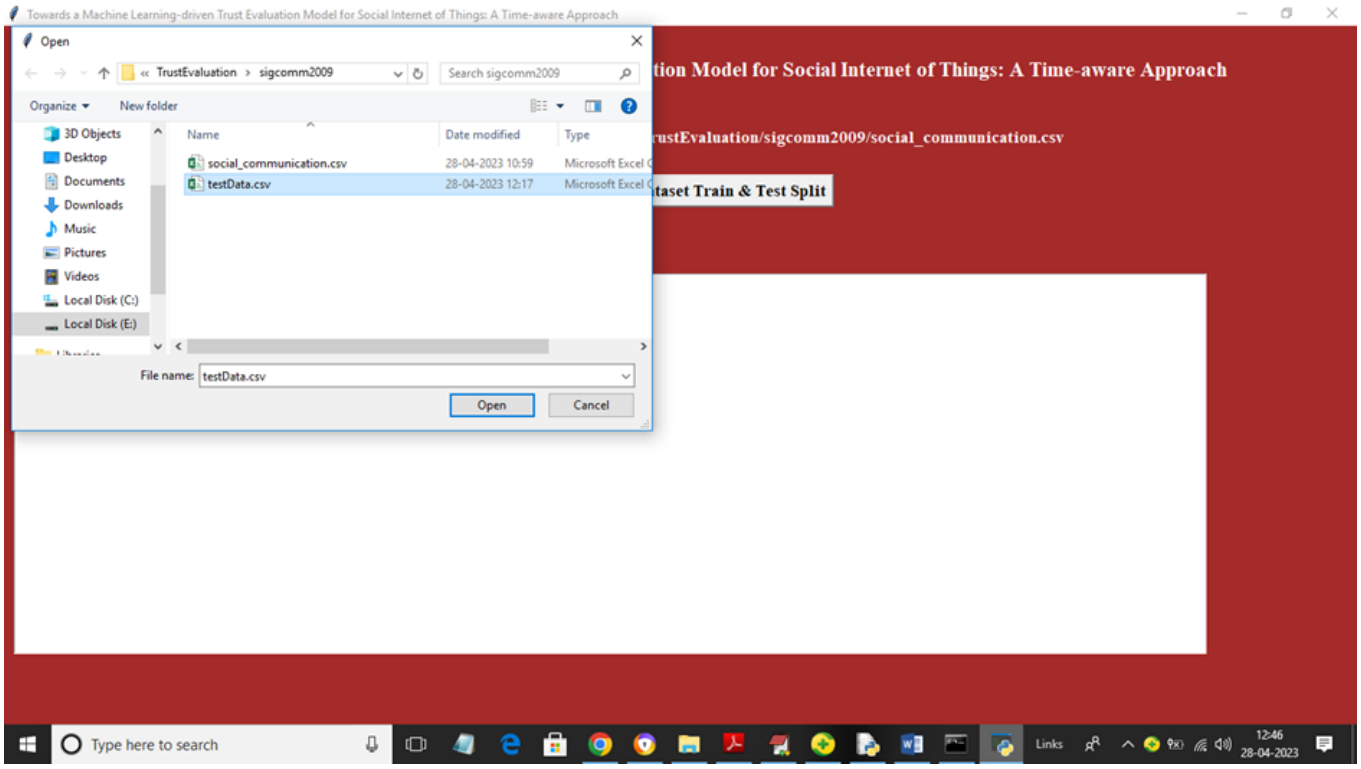
Screenshot 5.5: Graph obtained from KMEASN Alogorithm

5.6 ACCURACY GRAPH RESULT



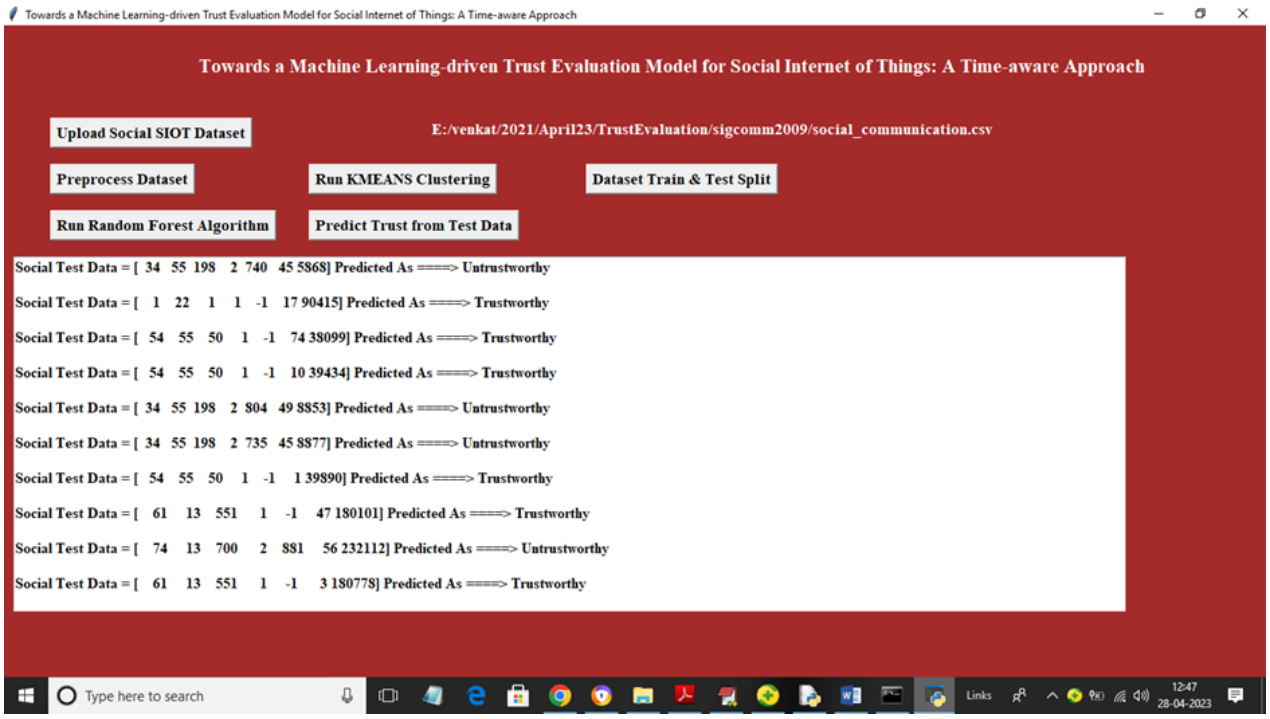
Screenshot 5.6: Run Random Forest on data result of TRUST EVALUATION MODEL using Machine Learning

5.7 UPLOAD TEST DATA



Screenshot 5.7: Uploading test data result of TRUST EVALUATION MODEL using Machine Learning

5.8 FINAL OUTPUT



Screenshot 5.8: Final output

5. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discoverevery conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, 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.

6.2 TYPES OF TESTING

6.2.1 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.

6.2.2 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.

6.2.3 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.

6.3 TEST CASES

6.3.1 UPLOADING DATASET

Test case ID	Test case name	Purpose	Test Case	Output
1	User uploads Data	Use it for evaluation	The user uploads the SIOT data as a csv file.	Uploaded successfully
2	User uploads 2nd data	Use it for prediction	The user uploads the different consumers behavior data as a csv file	Uploaded successfully

6.3.2 TRUST EVALUATION

Test case ID	Test case name	Purpose	Input	Output
1	Prediction test 1	To check if app performs its task	A csv file is given	accuracy is predicted.
2	Prediction test 2	To check if app performs its task	A another csv fileis given	predicted output and accuracy.

7. CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

In conclusion, the proposed machine learning-driven trust evaluation model addresses the critical challenges of trust management in the Social Internet of Things (SIOT) by integrating advanced techniques and a time-aware approach. By leveraging historical interaction data and temporal patterns, the model dynamically assesses trustworthiness, adapting to changes in device behavior. The incorporation of supervised and unsupervised learning captures both explicit and implicit trust indicators, enhancing the comprehensiveness of evaluations. Key features, such as time-decay functions and Recurrent Neural Networks (RNNs), improve accuracy in reflecting temporal trust dynamics. Experimental results on simulated datasets indicate significant improvements in trust evaluation accuracy and resilience against malicious behavior compared to traditional methods. Furthermore, the system's scalability ensures efficient assessments in large SIOT networks, paving the way for more reliable and adaptive trust management systems. This advancement ultimately enhances security and collaboration within the SIOT ecosystem, making it a crucial step towards more robust and trustworthy inter-device interactions.

7.2 FUTURE SCOPE

The future scope of this project includes expanding the trust evaluation model to accommodate larger, more complex Social Internet of Things (SIOT) networks, incorporating advanced machine learning techniques like deep learning for more precise trust predictions. Integration with blockchain technology could further enhance security and transparency in trust management. Additionally, the model can be extended to support real-time trust evaluations in highly dynamic environments, such as smart cities and autonomous systems, while also considering multi-dimensional trust factors like privacy, energy efficiency, and user preferences. This would enable broader applications in critical domains such as healthcare, transportation, and industrial IoT.

8. BIBLIOGRAPHY

8. BIBILOGRAPHY

8.1 REFERENCES

1. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4), 2347-2376.
2. Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. Computer networks, 54(15), 2787-2805.
3. Dargie, W., & Poellabauer, C. (2010). Fundamentals of wireless sensor networks: Theory and practice. John Wiley & Sons.
4. Sookhak, M., Talebian, H., Shojafar, M., Mozaffari, M., Alinejad-Rokny, H., & Elhoseny, M. (2019). Secure data communication and trust management in fog-and-cloud integrated IoT environments. IEEE Internet of Things Journal, 7(3), 2270-2282.
5. Pires, G., Garcia, N. M., & Cerqueira, E. (2017). The internet of things: A survey of topics and trends. Information Systems Frontiers, 20(3), 495-510.
6. Qureshi, B. A., Shah, M. A., Yousafzai, A., Khan, A. N., & Kim, S. W. (2019). Blockchain-based secure and trustable internet of things architecture for healthcare. IEEE Access, 7, 47003-47013.
7. Yan, Z., Zhang, P., & Vasilakos, A. V. (2014). A survey on trust management for internet of things. Journal of network and computer applications, 42, 120-134.
8. Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., & Zhao, W. (2017). A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. IEEE Internet of Things Journal, 4(5), 1125-1142.

8.2 WEBSITES

[1] https://github.com/Vishaljakkam/Trust-evaluation-model/blob/main/code