A

Major Project

on

INTERNET FINANCIAL FRAUD DETECTION BASED ON A DISTRIBUTED BIG DATA APPROACH WITH NODE2VEC

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

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

THATI YASHWANTH	(207R1A05P2)
TALAGAMA M DEEKSHIT	(207R1A05P1)
SUGAMANCHI NARENDER	(207R1A05P0)

UNDER THE GUIDANCE OF

Dr. J. NARASIMHARAO

(Associate 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.

2020-2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR
GROUP OF INSTITUTIONS
EXPLORE TO INVENT

CERTIFICATE

This is to certify that the project entitled **“Internet Financial Fraud Detection Based On A Distributed Big Data Approach With Node2vec”** being submitted by **T. YASHWANTH (207R1A05P2), T.M. DEEKSHIT (207R1A05P1) & S. NARENDER (207R1A05P0)** 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 2023-24.

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

Dr. J. NARASIMHARAO	**Dr. A. RAJI REDDY**
(Associate Professor)	DIRECTOR
INTERNAL GUIDE	

Dr. K. SRUJAN RAJU	**EXTERNAL EXAMINER**
HOD	

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 my guide Dr. **J. Narasimharao, Associate** 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 the Project Review Committee (PRC) **G. Vinesh Shanker, Dr. J. Narasimharao, Ms. Shilpa, & Dr. K. Maheswari** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** 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 also 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 theirconstant 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.

T. YASHWANTH	**(207R1A05P2)**
T. M. DEEKSHIT	**(207R1A05P1)**
S. NARENDER	**(207R1A05P0)**

ABSTRACT

The rapid development of information technologies like Internet of Things, Big Data, Artificial Intelligence, Blockchain, etc., has profoundly affected people's consumption behaviors and changed the development model of the financial industry. The financial services on Internet and IoT with new technologies has provided convenience and efficiency for consumers, but new hidden fraud risks are generated also. Fraud, arbitrage, vicious collection, etc., have caused bad effects and huge losses to the development of finance on Internet and IoT. However, as the scale of financial data continues to increase dramatically, it is more and more difficult for existing rule-based expert systems and traditional machine learning model systems to detect financial frauds from large-scale historical data. In the meantime, as the degree of specialization of financial fraud continues to increase, fraudsters can evade fraud detection by frequently changing their fraud methods.

In this article, an intelligent and distributed Big Data approach for Internet financial fraud detections is proposed to implement graph embedding algorithm Node2Vec to learn and represent the topological features in the financial network graph into low-dimensional dense vectors, to classify and predict the data samples of the large-scale dataset with the deep neural network intelligently and efficiently. The approach is distributed performed on the clusters of Apache Spark GraphX and Hadoop to process the large dataset in parallel. The groups of experimental results demonstrate that the proposed approach can improve the efficiency of Internet financial fraud detections with better precision rate and recall rate.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1.INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

The scope of an Internet financial fraud detection project encompasses defining the types of fraud to target, gathering relevant data sources such as transaction logs and user profiles, and preprocessing this data to ensure its quality. Real-time monitoring and detection systems are deployed to continuously assess transactions for suspicious patterns, with integration into existing financial infrastructure. Documentation and reporting facilitate transparency and accountability, while continuous improvement efforts refine the system over time.

1.2 PROJECT PURPOSE

The primary purpose of an Internet financial fraud detection project is to safeguard financial systems and protect users from fraudulent activities conducted online. By leveraging advanced technologies such as machine learning and data analytics, the project aims to detect and prevent various forms of financial fraud, including identity theft, credit card fraud, phishing scams, and unauthorized transactions. Ultimately, the project's goal is to enhance security, maintain trust in online financial services, and safeguard the integrity of digital transactions.

1.3 PROJECT FEATURES

This Internet financial fraud detection project involves gathering data from transactions and user profiles, cleaning it up, and spotting suspicious patterns. We then use this data to teach computer programs to recognize fraud, like unusual spending or logins from unexpected places. Once trained, these programs watch over transactions in real-time, sending alerts if they see something fishy. We integrate them into existing banking systems to keep things smooth for users. Regular checks ensure they're working well, and we make sure everything follows the rules and respects people's privacy. By doing this, we're aiming to make online transactions safer and more trustworthy for everyone.

2.SYSTEM ANALYSIS

2.SYSTEM ANALYSIS

2.1 PROBLEM DEFINITION

The problem definition for the project "Internet Financial Fraud Detection based on a Distributed Big Data Approach with Node2vec" involves addressing the challenges associated with detecting and preventing fraudulent activities in online financial transactions. Specifically, the project aims to develop a scalable and efficient fraud detection system capable of analyzing large volumes of financial transaction data in real-time. The system must accurately identify suspicious patterns and behaviors indicative of fraud while minimizing false positives and false negatives. By leveraging a distributed big data approach and incorporating the Node2vec algorithm for graph-based analysis, the project seeks to enhance the effectiveness and scalability of fraud detection in online financial ecosystems.

The increasing prevalence of online financial transactions has led to a corresponding rise in fraudulent activities such as identity theft, credit card fraud, phishing scams, and unauthorized transactions. Detecting and preventing such fraudulent activities is critical for maintaining trust in online financial systems and protecting both consumers and financial institutions from financial losses.

Overall, the project aims to develop a robust and reliable fraud detection system that enhances security, protects users, and preserves the integrity of online financial transactions. Through the integration of distributed big data techniques and the Node2vec algorithm for graph-based analysis, the project seeks to leverage the power of data-driven insights to combat financial fraud effectively in the digital age.

2.2 EXISTING SYSTEM

Allen et al. find that there are many credit channels in the United States and based on the research of American household credit models, and that household consumption, household income, credit banks and credit scale are obviously related. Kregel studies the development trend of consumer finance and finds that the development of Internet consumer finance companies must fully consider the current market legal environment, financial market and consumer behavior factors, etc. Internet consumer finance is directly related to the current development of the national financial system. Momparler et al. take the American Internet consumer finance company as the research object, study the risks and advantages of the Internet consumer finance platform, and design a related risk management model.

Ficawoyi et al. analyze the positive relationship between Internet exposure levels and credit card default through surveys on consumer finance and income nodes. The research points out that Internet access, low income, and male families are more likely to cause credit card defaults. Giudici et al. propose how to improve credit risk accuracy of P2P Internet financial platforms and of those who lend to small and medium enterprises. The augment traditional credit scoring methods are put forward with "alternative data" that consist of centrality measures derived from similarity networks among borrowers and deduced from their financial ratios. The experimental findings suggest that the proposed approach improves predictive accuracy as well as model explain ability.

2.2.1 DISADVANTAGES OF EXISTING SYSTEM

- The system doesn't support Resilient Distributed Datasets.
- There is no Directed Acyclic Graph method to find fraud accurately.

2.3 PROPOSED SYSTEM

Through studying many Internets financial fraud cases, two important characteristics are found: The pattern of Internet financial fraud continues to evolve and develop over time, not just repeating the existing individual behavior patterns appeared in historical cases; With the advancement of anti-fraud technology, it is getting harder for individuals to commit Internet financial fraud. It needs to be organized and conducted through related and connected groups. A graph is an abstract graph formed by a number of nodes and the edges connecting each node. It is usually used to describe a specific relationship between things. A relational network graph refers to a graph-based data structure composed of nodes and edges. Each node represents an entity, and each edge is the relationship between an entity and the other connected entity. The relationship network graph connects different entities together according to their relationships; thus, it could provide the ability to analyze problems from the perspective of "relationship".

In anti-fraud applications, entities in the network graph, such as people, equipment, mailboxes, card numbers, etc., can be represented by nodes, and the relationships between these nodes in the business can be represented by edges. Through continuous construction and reproduction of the associated relationships hidden covertly in Internet financial frauds, fraud characteristics can be detected and corresponding risk control strategies can be designed. The graph algorithms can characterize various high-risk features in Internet finance, such as batch attacks, intermediary participation, etc., which is more effective to identify abnormal group frauds from normal behaviors.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- Node2Vec is a graph embedding algorithm that introduces two biased random walk methods---BFS (Breadth First Search) and DFS (Depth First Search) on the basis of Deep Walk.
- AN INTELLIGENT AND DISTRIBUTED BIG DATA APPROACH FOR INTERNET FINANCIAL FRAUD DETECTION.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and a businessproposal 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.

Three key considerations involved in the feasibility analysis:

- ❖ Economic Feasibility
- ❖ Technical Feasibility
- ❖ Operational Feasibility

2.4.1 ECONOMIC FEASIBILITY

Economic Feasibility or Cost-benefit is an assessment of the economic justification for a computer-based project. As hardware was installed from the beginning & for lots of purposes thus the cost of the project of hardware is low. Since the system is network based, any number of employees connected to the LAN within that organization can use this tool at any time. The Virtual Private Network is to be developed using the existing resources of the organization. So the project is economically feasible.

2.4.2 TECHNICAL FEASIBILITY

According to Roger S. Pressman, Technical Feasibility is the assessment of the technical resources of the organization. The organization needs IBM compatible machines with a graphical web browser connected to the Internet and Intranet. The system is developed for platform independent environment. Java Server Pages, JavaScript, HTML, SQL server and WebLogic Server are used to develop the system. The technical feasibility has been carried out. The system is technically feasible for development and can be developed with the existing facility.

2.4.3 OPERATIONAL FEASIBILITY

Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the admin and helps him in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify 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
• RAM	-	4 GB (min)
• Hard Disk	-	20 GB
• Keyboard	-	Standard Windows Keyboard
• Mouse	-	Two or Three Button Mouse
• Monitor	-	SVGA

2.5.2 SOFTWARE REQUIREMENTS:

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

• Operating system	:	Windows 7 Ultimate.
• Coding Language	:	Python.
• Front-End	:	Python.
• Back-End	:	Django-ORM
• Designing	:	Html, CSS, JavaScript.
• Data Base	:	MySQL (WAMP Server).

3. ARCHITECTURE

3.ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction.

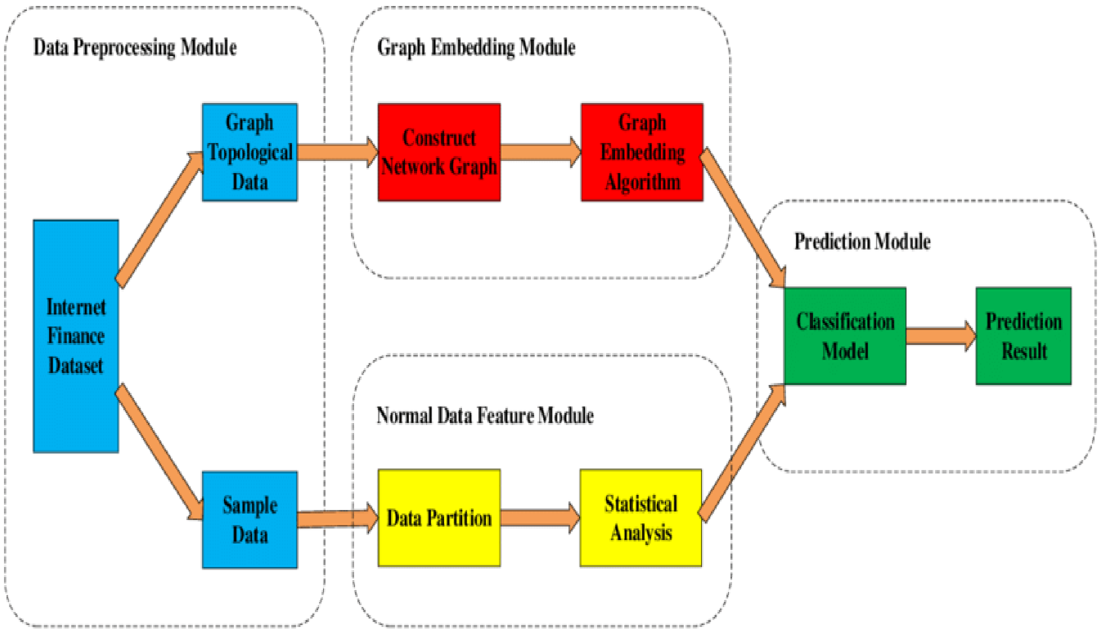


Figure 3.1: Project Architecture of Internet Financial Fraud Detection Based on a Distributed Big Data Approach with Node2vec.

3.2 DESCRIPTION

The architecture of our Internet financial fraud detection project, rooted in a distributed big data approach with Node2Vec, orchestrates a comprehensive framework for the effective identification and prevention of fraudulent activities. At its core, the system begins with the ingestion of raw financial data from diverse sources, employing distributed data storage solutions like Apache Hadoop or Apache Spark to handle the sheer volume of incoming information. Following data preprocessing to ensure accuracy and consistency, the Node2Vec algorithm is employed to convert the financial data into a structured graph representation, where entities such as users, transactions, and accounts become nodes, and their relationships are captured as edges. Leveraging distributed processing frameworks like Apache Spark, the architecture conducts parallel processing on the graph, generating Node2Vec embeddings that encapsulate both structural and contextual information crucial for anomaly detection. These embeddings serve as feature vectors for machine learning models, which are trained to discern patterns indicative of fraudulent behavior. Real-time monitoring, facilitated by tools like Apache Kafka, is implemented to detect and respond to suspicious activities as they occur.

The system is designed for scalability, fault tolerance, and seamless integration with external systems, ensuring a robust and adaptable solution for combating internet financial fraud. The architecture culminates in alerting mechanisms and detailed reporting, providing stakeholders with insights into detected anomalies and the actions taken in response.

3.3 USE CASE DIAGRAM

The use case diagram for our Internet financial fraud detection project encapsulates a detailed representation of the system's functionalities and user interactions. At its core are two primary actors: the end-users, denoted as the User, and External Systems, representing any integrated external databases or systems. The User actor engages in several key use cases, including logging into the system, viewing the real-time dashboard, uploading raw financial data, initiating the fraud detection process, receiving real-time alerts for potential fraud, and generating comprehensive reports summarizing detected anomalies and the corresponding actions taken. The diagram establishes associations between these actors and their associated use cases, illustrating the intricate interactions within the system.

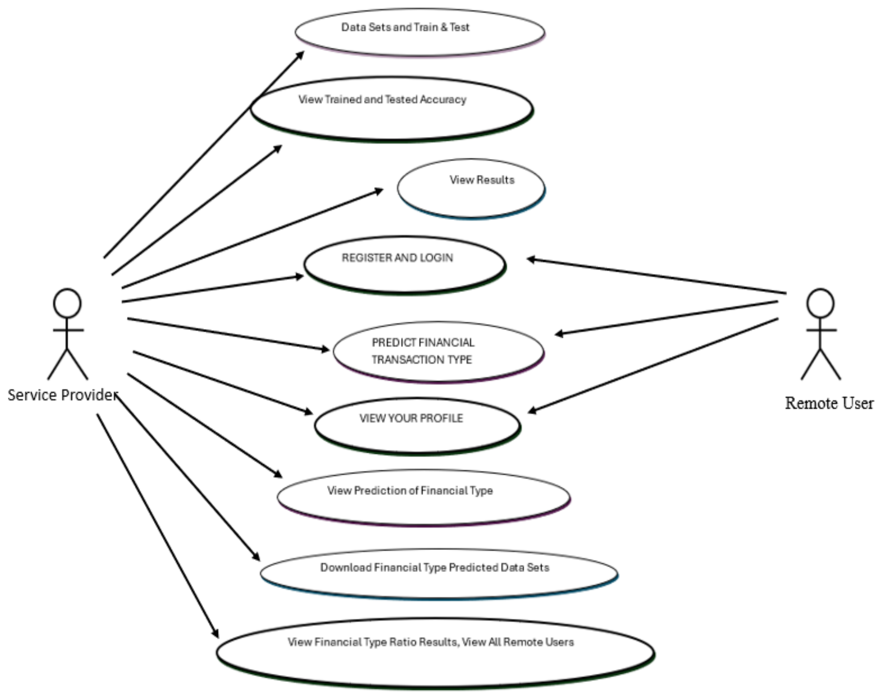

Data Sets and Train & Test
View Trained and Tested Accuracy
View Results
REGISTER AND LOGIN
PREDICT FINANCIAL TRANSACTION TYPE
VIEW YOUR PROFILE
View Prediction of Financial Type
Download Financial Type Predicted Data Sets
View Financial Type Ratio Results, View All Remote Users
Service Provider
Remote User

Figure 3.2: Use Case Diagram of Internet Financial Fraud Detection Based on a Distributed Big Data Approach with Node2vec.

3.4 CLASS DIAGRAM

The class diagram for our Internet financial fraud detection project serves as a blueprint detailing the structure and relationships between essential entities within the system. At its core is the User class, embodying the end-users who interact with the system. This class is characterized by attributes such as user ID, username, and authentication details, while methods like login () and view Dashboard () encapsulate user interactions with the system. The External Systems class is introduced to represent external databases or systems that seamlessly integrate with our fraud detection platform, enhancing its capabilities.

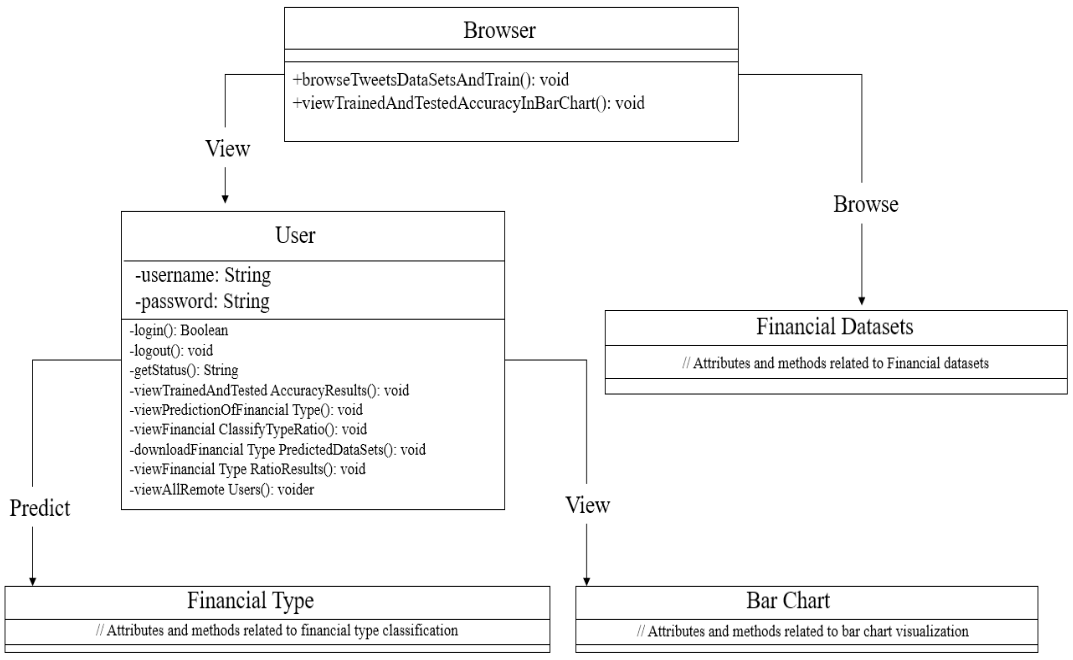

Browser
+browseTweetsDataSetsAndTrain(): void
+viewTrainedAndTestedAccuracyInBarChart(): void
View
Browse
User
-username: String
-password: String
-login(): Boolean
-logout(): void
-getStatus(): String
-viewTrainedAndTested AccuracyResults(): void
-viewPredictionOfFinancial Type(): void
-viewFinancial ClassifyTypeRatio(): void
-downloadFinancial Type PredictedDataSets(): void
-viewFinancial Type RatioResults(): void
-viewAllRemote Users(): voider
Financial Datasets
// Attributes and methods related to Financial datasets
Predict
View
Financial Type
// Attributes and methods related to financial type classification
Bar Chart
// Attributes and methods related to bar chart visualization

Figure 3.3: Class Diagram of Internet Financial Fraud Detection Based on a Distributed Big Data Approach with Node2vec.

3.5 SEQUENCE DIAGRAM

The sequence diagram for our Internet financial fraud detection project unfolds a step-by-step depiction of the interactions and messaging sequences between different components and actors within the system. It commences with the User actor initiating the sequence by logging into the system, triggering the authentication process. Subsequently, the User engages in the crucial step of uploading raw financial data. As this data enters the system, the sequence illustrates the intricate process of data preprocessing, graph representation using Node2Vec, and the parallel distributed processing facilitated by frameworks like Apache Spark. The diagram vividly captures the dynamic generation of Node2Vec embeddings, which serve as feature vectors for the subsequent machine learning models.

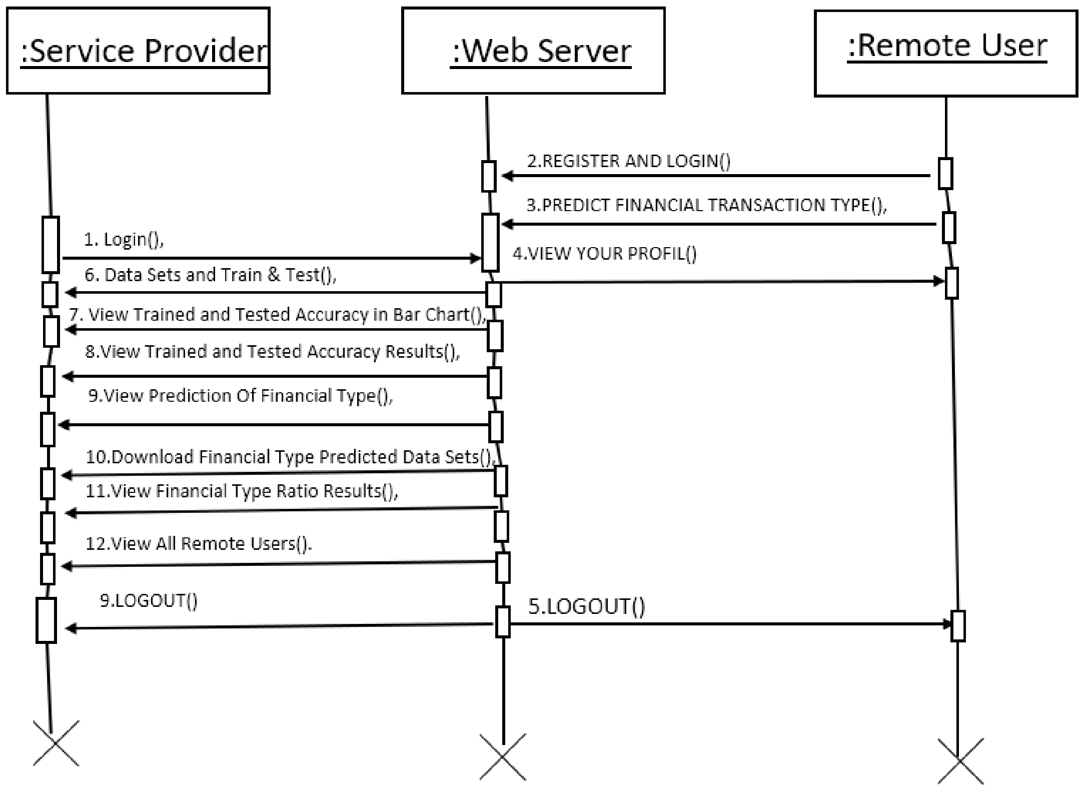

:Service Provider
:Web Server
:Remote User
2.REGISTER AND LOGIN()
3.PREDICT FINANCIAL TRANSACTION TYPE(),
4.VIEW YOUR PROFIL()
1. Login(),
6. Data Sets and Train & Test(),
7. View Trained and Tested Accuracy in Bar Chart(),
8.View Trained and Tested Accuracy Results(),
9.View Prediction Of Financial Type(),
10.Download Financial Type Predicted Data Sets(),
11.View Financial Type Ratio Results(),
12.View All Remote Users().
9.LOGOUT()
5.LOGOUT()

Figure 3.4: Sequence Diagram of Internet Financial Fraud Detection Based on a Distributed Big Data Approach with Node2vec.

3.6 ACTIVITY DIAGRAM

Activity diagram for an "Internet Financial Fraud Detection" system involves several key steps. The process begins with data collection, where various types of data related to online transactions are gathered, including transaction details, user behavior data, and historical fraud data. Once the model is trained, it can be used for fraud prediction, where it evaluates whether a new transaction is fraudulent or not based on the extracted features. Finally, if a transaction is predicted to be fraudulent, an alert is generated, which may involve notifying relevant authorities or taking preventive measures. This process concludes with the end of the activity diagram, signifying the completion of the fraud detection process.

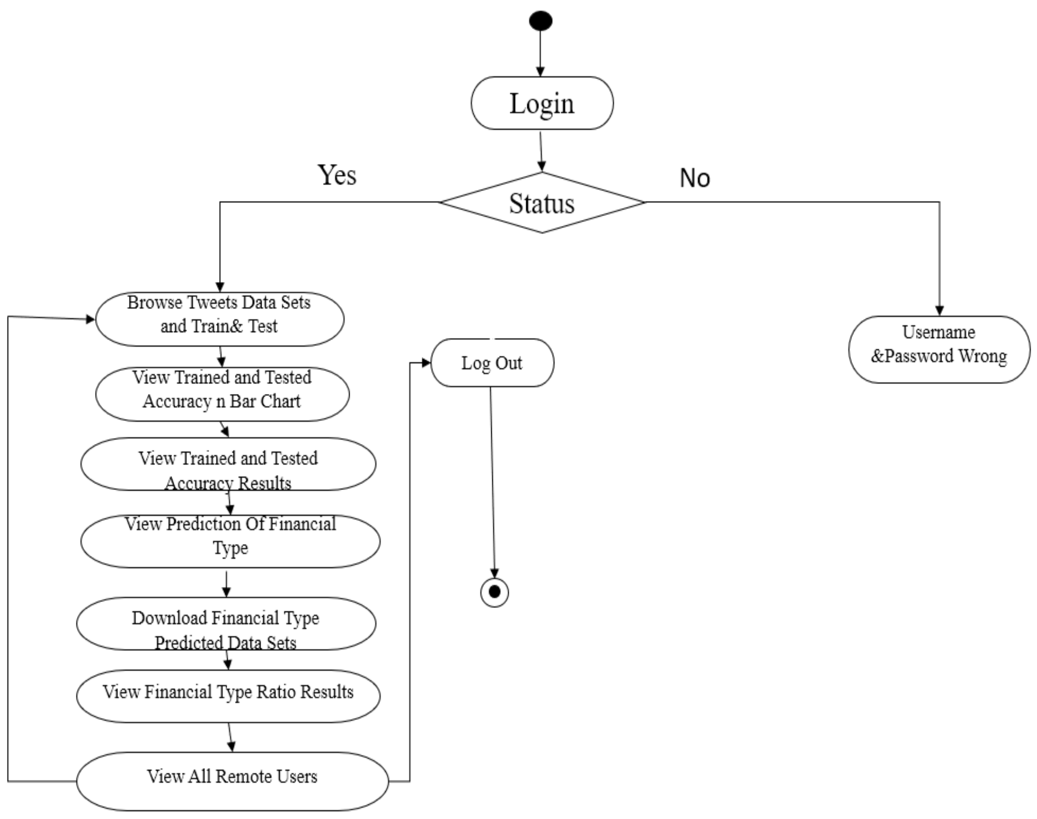

Login
Yes
Status
No
Browse Tweets Data Sets and Train& Test
View Trained and Tested Accuracy n Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Financial Type
Download Financial Type Predicted Data Sets
View Financial Type Ratio Results
View All Remote Users
Log Out
Username &Password Wrong

Figure 3.5: Activity Diagram of Internet Financial Fraud Detection Based on a Distributed Big Data Approach with Node2vec.

4.IMPLEMENTATION

4. SAMPLE CODE

Service Provider-Views.py

```
from django.db.models import  Count, Avg
from django.shortcuts import render, redirect
from django.db.models import Count
from django.db.models import Q
import datetime
import xlwt
from django.http import HttpResponse
import numpy as np


import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from sklearn.pipeline import Pipeline

#to data preprocessing
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder

#NLP tools
import re
import nltk
nltk.download('stopwords')
nltk.download('rslp')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer

#train split and fit models
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from nltk.tokenize import TweetTokenizer
from sklearn.ensemble import VotingClassifier
#model selection
from sklearn.metrics import confusion_matrix, accuracy_score, plot_confusion_matrix,
classification_report

# Create your views here.
```

```
from Remote_User.models import
ClientRegister_Model,Financial_Fraud_Prediction,detection_ratio,detection_accuracy
def serviceproviderlogin(request):
    if request.method  == "POST":
        admin = request.POST.get('username')
        password = request.POST.get('password')
        if admin == "Admin" and password =="Admin":
            detection_accuracy.objects.all().delete()
            return redirect('View_Remote_Users')

    return render(request,'SProvider/serviceproviderlogin.html')

def View_Financial_Type_Ratio(request):
    detection_ratio.objects.all().delete()
    rratio = ""
    kword = 'Fraud'
    print(kword)
    obj = Financial_Fraud_Prediction.objects.all().filter(Q(Prediction=kword))
    obj1 = Financial_Fraud_Prediction.objects.all()
    count = obj.count();
    count1 = obj1.count();
    ratio = (count / count1) * 100
    if ratio != 0:
        detection_ratio.objects.create(names=kword, ratio=ratio)

    ratio1 = ""
    kword1 = 'No Fraud'
    print(kword1)
    obj1 = Financial_Fraud_Prediction.objects.all().filter(Q(Prediction=kword1))
    obj11 = Financial_Fraud_Prediction.objects.all()
    count1 = obj1.count();
    count11 = obj11.count();
    ratio1 = (count1 / count11) * 100
    if ratio1 != 0:
        detection_ratio.objects.create(names=kword1, ratio=ratio1)
    obj = detection_ratio.objects.all()
    return render(request, 'SProvider/View_Financial_Type_Ratio.html', {'objs': obj})

def View_Remote_Users(request):
    obj=ClientRegister_Model.objects.all()

return render(request,'SProvider/View_Remote_Users.html',{'objects':obj})

def ViewTrendings(request):
    topic =
Financial_Fraud_Prediction.objects.values('topics').annotate(dcount=Count('topics')).order_by('-
dcount')
```

```
    return  render(request,'SProvider/ViewTrendings.html',{'objects':topic})

def charts(request,chart_type):
    chart1 = detection_ratio.objects.values('names').annotate(dcount=Avg('ratio'))
    return render(request,"SProvider/charts.html", {'form':chart1, 'chart_type':chart_type})

def charts1(request,chart_type):
    chart1 = detection_accuracy.objects.values('names').annotate(dcount=Avg('ratio'))
    return render(request,"SProvider/charts1.html", {'form':chart1, 'chart_type':chart_type})

def View_Prediction_Of_Financial_Type(request):
    obj =Financial_Fraud_Prediction.objects.all()
    return render(request, 'SProvider/View_Prediction_Of_Financial_Type.html', {'list_objects':
obj})

def likeschart(request,like_chart):
    charts =detection_accuracy.objects.values('names').annotate(dcount=Avg('ratio'))
    return render(request,"SProvider/likeschart.html", {'form':charts, 'like_chart':like_chart})


def Download_Trained_DataSets(request):

    response = HttpResponse(content_type='application/ms-excel')
    # decide file name
    response['Content-Disposition'] = 'attachment; filename="Predicted_Data.xls"'
    # creating workbook
    wb = xlwt.Workbook(encoding='utf-8')
    # adding sheet
    ws = wb.add_sheet("sheet1")
    # Sheet header, first row

row_num = 0
    font_style = xlwt.XFStyle()
    # headers are bold
    font_style.font.bold = True
    # writer = csv.writer(response)
    obj = Financial_Fraud_Prediction.objects.all()
    data = obj  # dummy method to fetch data.
    for my_row in data:
        row_num = row_num + 1
        ws.write(row_num, 0, my_row.Customer_Email, font_style)
        ws.write(row_num, 1, my_row.customerPhone, font_style)
        ws.write(row_num, 2, my_row.customerDevice, font_style)
        ws.write(row_num, 3, my_row.customerIPAddress, font_style)
        ws.write(row_num, 4, my_row.customerBillingAddress, font_style)
        ws.write(row_num, 5, my_row.No_Transactions, font_style)
        ws.write(row_num, 6, my_row.No_Orders, font_style)
        ws.write(row_num, 7, my_row.No_Payments, font_style)
        ws.write(row_num, 8, my_row.Prediction, font_style)
    wb.save(response)
    return response

def train_model(request):
    detection_accuracy.objects.all().delete()
    dataset = pd.read_csv('Transaction_Datasets.csv', encoding='latin-1')

    dataset.rename(columns={'Fraud': 'label', 'customerBillingAddress': 'caddress'}, inplace=True)

    def apply_results(label):
        if (label == 0):
            return 0  # False
        elif (label == 1):
            return 1  # True

    dataset['results'] = dataset['label'].apply(apply_results)
    dataset.drop(['label'], axis=1, inplace=True)
    results = dataset['results'].value_counts()

    cv = CountVectorizer()

    x = dataset["caddress"]
    y = dataset["results"]

    # x = cv.fit_transform(x)


x = cv.fit_transform(dataset['caddress'].apply(lambda x: np.str_(x)))

    models = []
    from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.20)
    X_train.shape, X_test.shape, y_train.shape

    print("Naive Bayes")

    from sklearn.naive_bayes import MultinomialNB

    NB = MultinomialNB()
    NB.fit(X_train, y_train)
    predict_nb = NB.predict(X_test)
    naivebayes = accuracy_score(y_test, predict_nb) * 100
    print("ACCURACY")
    print(naivebayes)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, predict_nb))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, predict_nb))
    detection_accuracy.objects.create(names="Naive Bayes", ratio=naivebayes)


    # SVM Model
    print("SVM")
    from sklearn import svm

    lin_clf = svm.LinearSVC()
    lin_clf.fit(X_train, y_train)
    predict_svm = lin_clf.predict(X_test)
    svm_acc = accuracy_score(y_test, predict_svm) * 100
    print("ACCURACY")
    print(svm_acc)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, predict_svm))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, predict_svm))
    detection_accuracy.objects.create(names="SVM", ratio=svm_acc)

    print("Logistic Regression")

    from sklearn.linear_model import LogisticRegression

    reg = LogisticRegression(random_state=0, solver='lbfgs').fit(X_train, y_train)

y_pred = reg.predict(X_test)
    print("ACCURACY")
    print(accuracy_score(y_test, y_pred) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, y_pred))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, y_pred))
    detection_accuracy.objects.create(names="Logistic Regression", ratio=accuracy_score(y_test,
y_pred) * 100)

    print("Decision Tree Classifier")
    dtc = DecisionTreeClassifier()
    dtc.fit(X_train, y_train)
    dtcpredict = dtc.predict(X_test)
    print("ACCURACY")
    print(accuracy_score(y_test, dtcpredict) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, dtcpredict))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, dtcpredict))
    detection_accuracy.objects.create(names="Decision Tree Classifier",
ratio=accuracy_score(y_test, dtcpredict) * 100)
```

```
    print("SGD Classifier")
    from sklearn.linear_model import SGDClassifier
    sgd_clf = SGDClassifier(loss='hinge', penalty='l2', random_state=0)
    sgd_clf.fit(X_train, y_train)
    sgdpredict = sgd_clf.predict(X_test)
    print("ACCURACY")
    print(accuracy_score(y_test, sgdpredict) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, sgdpredict))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, sgdpredict))
    detection_accuracy.objects.create(names="SGD Classifier", ratio=accuracy_score(y_test,
sgdpredict) * 100)


    labeled = 'Processed_data.csv'
    dataset.to_csv(labeled, index=False)
    dataset.to_markdown

    obj = detection_accuracy.objects.all()
    return render(request,'SProvider/train_model.html', {'objs': obj})
```

Remote user-views.py

```
from django.db.models import Count
from django.db.models import Q
from django.shortcuts import render, redirect, get_object_or_404
import datetime
import openpyxl

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from sklearn.pipeline import Pipeline

#to data preprocessing
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder

#NLP tools
import re
import nltk
nltk.download('stopwords')
nltk.download('rslp')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer
```

```
#train split and fit models
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from nltk.tokenize import TweetTokenizer
from sklearn.ensemble import VotingClassifier
#model selection
from sklearn.metrics import confusion_matrix, accuracy_score, plot_confusion_matrix,
classification_report
# Create your views here.
from Remote_User.models import
ClientRegister_Model,Financial_Fraud_Prediction,detection_ratio,detection_accuracy

def login(request):


    if request.method == "POST" and 'submit1' in request.POST:

        username = request.POST.get('username')
        password = request.POST.get('password')
        try:
            enter = ClientRegister_Model.objects.get(username=username,password=password)
            request.session["userid"] = enter.id

            return redirect('ViewYourProfile')
        except:
            pass

    return render(request,'RUser/login.html')

def Add_DataSet_Details(request):

    return render(request, 'RUser/Add_DataSet_Details.html', {"excel_data": ''})


def Register1(request):

    if request.method == "POST":
        username = request.POST.get('username')
        email = request.POST.get('email')
        password = request.POST.get('password')
        phoneno = request.POST.get('phoneno')
        country = request.POST.get('country')
        state = request.POST.get('state')
        city = request.POST.get('city')
        ClientRegister_Model.objects.create(username=username, email=email, password=password,
phoneno=phoneno,
                                    country=country, state=state, city=city)

        return render(request, 'RUser/Register1.html')
    else:
        return render(request,'RUser/Register1.html')

def ViewYourProfile(request):
    userid = request.session['userid']
    obj = ClientRegister_Model.objects.get(id= userid)
    return render(request,'RUser/ViewYourProfile.html',{'object':obj})


def Predict_Financial_Type(request):
    if request.method == "POST":
        customerBillingAddress = request.POST.get('customerBillingAddress')
        if request.method == "POST":

            Customer_Email= request.POST.get('Customer_Email')
            customerPhone= request.POST.get('customerPhone')
            customerDevice= request.POST.get('customerDevice')
            customerIPAddress= request.POST.get('customerIPAddress')
            customerBillingAddress= request.POST.get('customerBillingAddress')
            No_Transactions= request.POST.get('No_Transactions')
            No_Orders= request.POST.get('No_Orders')
            No_Payments= request.POST.get('No_Payments')



        dataset = pd.read_csv('Transaction_Datasets.csv', encoding='latin-1')

        dataset.rename(columns={'Fraud': 'label', 'customerBillingAddress': 'caddress'},
inplace=True)

        def apply_results(label):
            if (label == 0):
                return 0  # False
            elif (label == 1):
                return 1  # True

        dataset['results'] = dataset['label'].apply(apply_results)
        dataset.drop(['label'], axis=1, inplace=True)
        results = dataset['results'].value_counts()

        cv = CountVectorizer()

        x = dataset["caddress"]
        y = dataset["results"]

        # x = cv.fit_transform(x)
```

```
x = cv.fit_transform(dataset['caddress'].apply(lambda x: np.str_(x)))

models = []
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.20)
X_train.shape, X_test.shape, y_train.shape

print("Naive Bayes")

from sklearn.naive_bayes import MultinomialNB
NB = MultinomialNB()
NB.fit(X_train, y_train)
predict_nb = NB.predict(X_test)
naivebayes = accuracy_score(y_test, predict_nb) * 100
print(naivebayes)
print(confusion_matrix(y_test, predict_nb))
print(classification_report(y_test, predict_nb))
models.append(('naive_bayes', NB))

# SVM Model
print("SVM")
from sklearn import svm
lin_clf = svm.LinearSVC()
lin_clf.fit(X_train, y_train)
predict_svm = lin_clf.predict(X_test)
svm_acc = accuracy_score(y_test, predict_svm) * 100
print(svm_acc)
print("CLASSIFICATION REPORT")
print(classification_report(y_test, predict_svm))
print("CONFUSION MATRIX")
print(confusion_matrix(y_test, predict_svm))
models.append(('svm', lin_clf))

print("Logistic Regression")

from sklearn.linear_model import LogisticRegression
reg = LogisticRegression(random_state=0, solver='lbfgs').fit(X_train, y_train)
y_pred = reg.predict(X_test)
print("ACCURACY")
print(accuracy_score(y_test, y_pred) * 100)
print("CLASSIFICATION REPORT")
print(classification_report(y_test, y_pred))
print("CONFUSION MATRIX")
print(confusion_matrix(y_test, y_pred))
models.append(('logistic', reg))

print("Decision Tree Classifier")
```

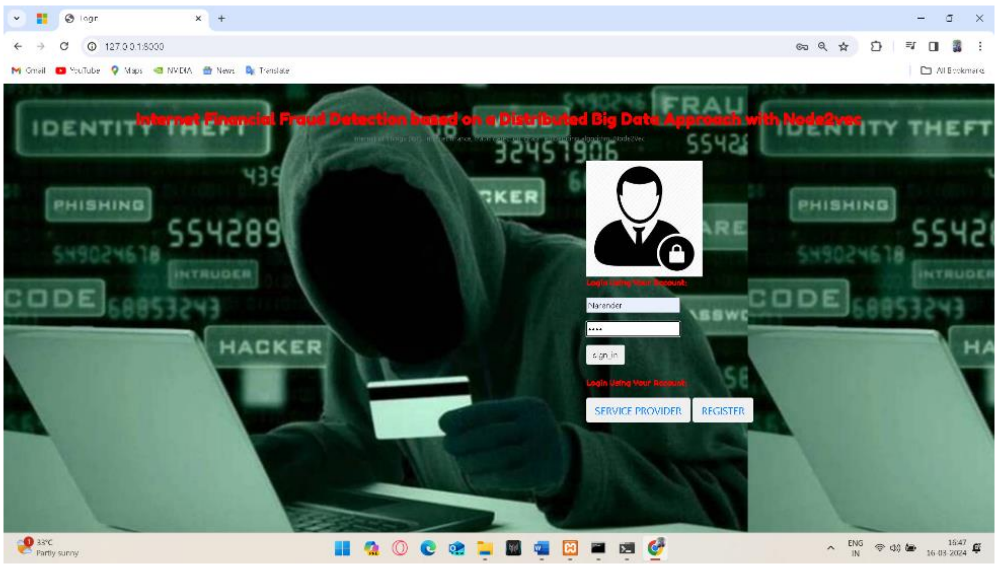
```
        dtc = DecisionTreeClassifier()
        dtc.fit(X_train, y_train)
        dtcpredict = dtc.predict(X_test)
        print("ACCURACY")
        print(accuracy_score(y_test, dtcpredict) * 100)
        print("CLASSIFICATION REPORT")
        print(classification_report(y_test, dtcpredict))
        print("CONFUSION MATRIX")
        print(confusion_matrix(y_test, dtcpredict))

        classifier = VotingClassifier(models)
        classifier.fit(X_train, y_train)
        y_pred = classifier.predict(X_test)

        customerBillingAddress1 = [customerBillingAddress]
        vector1 = cv.transform(customerBillingAddress1).toarray()
        predict_text = classifier.predict(vector1)

        pred = str(predict_text).replace("[", "")
        pred1 = pred.replace("]", "")

        prediction = int(pred1)

        if prediction == 0:
            val = 'No Fraud'
        elif prediction == 1:
            val = 'Fraud'

        Financial_Fraud_Prediction.objects.create(Customer_Email=Customer_Email,customerPhon
e=customerPhone,customerDevice=customerDevice,customerIPAddress=customerIPAddress,cust
omerBillingAddress=customerBillingAddress,
        No_Transactions=No_Transactions,
        No_Orders=No_Orders,
        No_Payments=No_Payments,
        Prediction=val)

        return render(request, 'RUser/Predict_Financial_Type.html',{'objs': val})
    return render(request, 'RUser/Predict_Financial_Type.html')
```

5. SCREENSHOTS

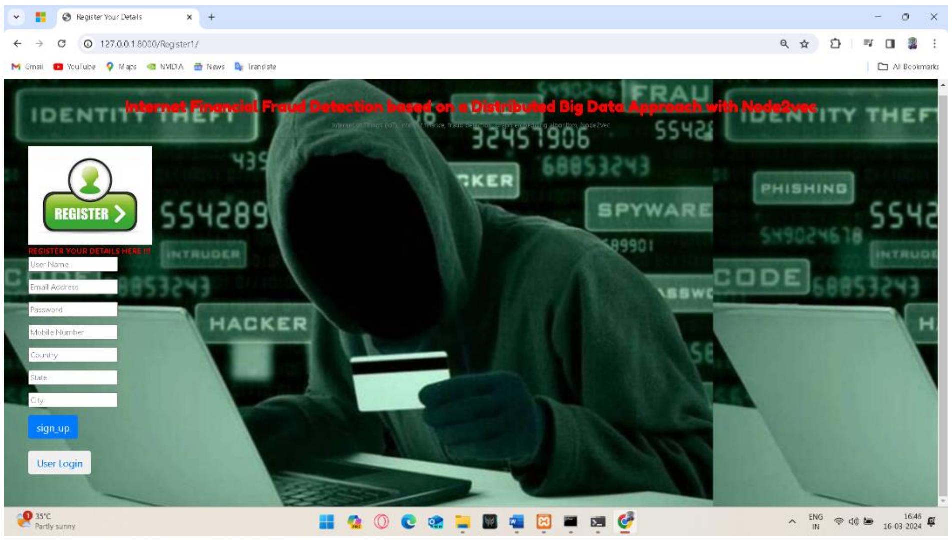
5. SCREENSHOTS

Step 1: To implement this project there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like REGISTER AND LOGIN, PREDICT FINANCIAL TRANSACTION TYPE, VIEW YOUR PROFILE.

- Access the login page where you can securely connect to a remote server. Register your credentials to create a personalized account, enabling seamless sign-in as both a user and a service provider.

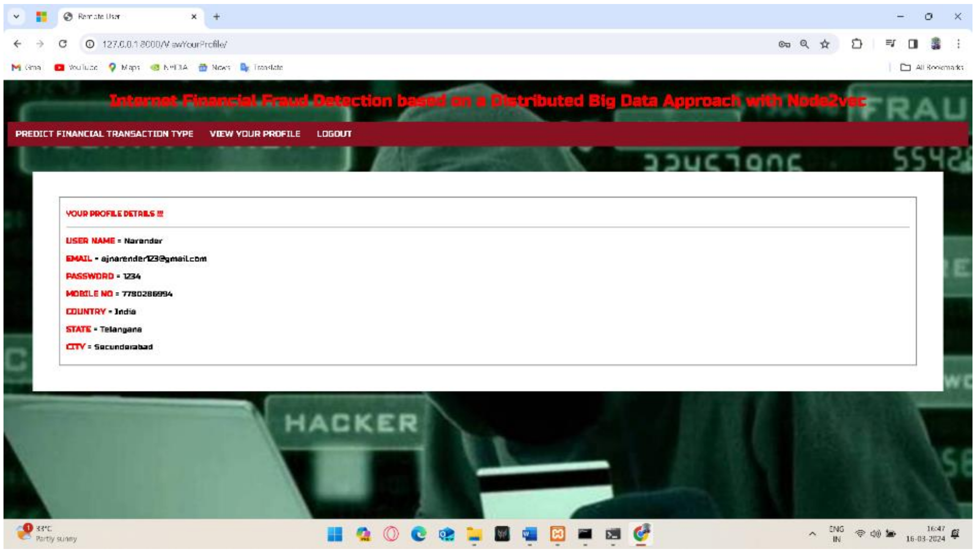

Internet Financial Fraud Detection based on a Distributed Big Data Approach with Node2vec
Login Using Your Account:
sign in
Login Using Your Account:
SERVICE PROVIDER
REGISTER

Screenshot 5.1: Home page


Register Your Details
127.0.0.1:8000/Register1/
Internet Financial Fraud Detection based on a Distributed Big Data Approach with Node2vec
REGISTER
REGISTER YOUR DETAILS HERE !!!
User Name
Email Address
Password
Mobile Number
Country
State
City
sign_up
User Login

Screenshot 5.2: Register Remote user

- ➢ Once logged into your remote account, conveniently access and review your profile details, reflecting the information you provided during the registration process as a remote user


Remote User
127.0.0.1:8000/ViewYourProfile/
Internet Financial Fraud Detection based on a Distributed Big Data Approach with Node2vec
PREDICT FINANCIAL TRANSACTION TYPE
VIEW YOUR PROFILE
LOGOUT

YOUR PROFILE DETAILS !!!

USER NAME = Narender

EMAIL = ajnarender123@gmail.com

PASSWORD = 1234

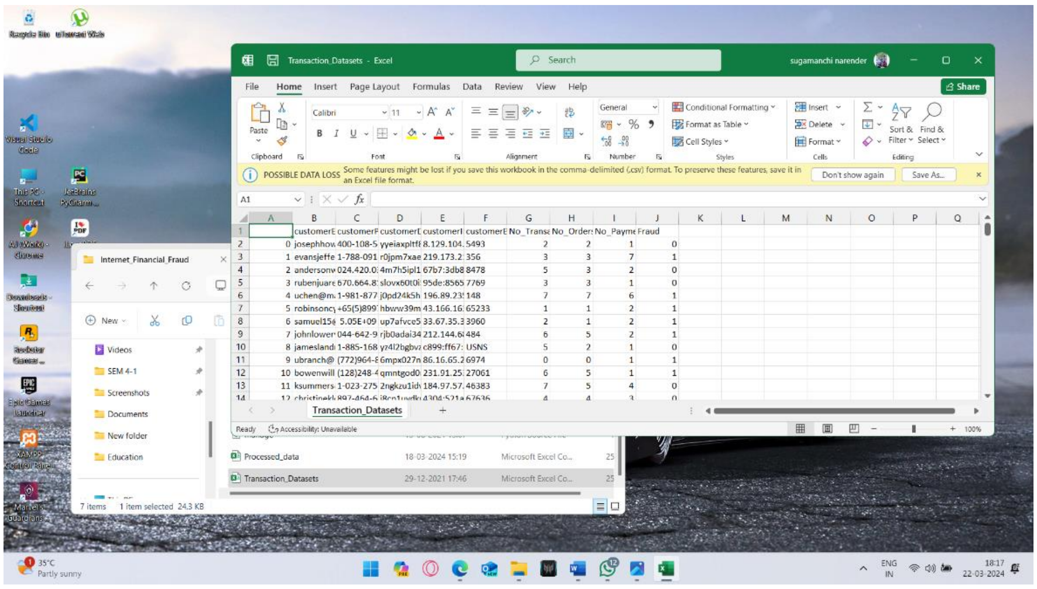
MOBILE NO = 7780286994

COUNTRY = India

STATE = Telangana

CITY = Secunderabad

Screenshot 5.3: View your profile



Transaction_Datasets - Excel

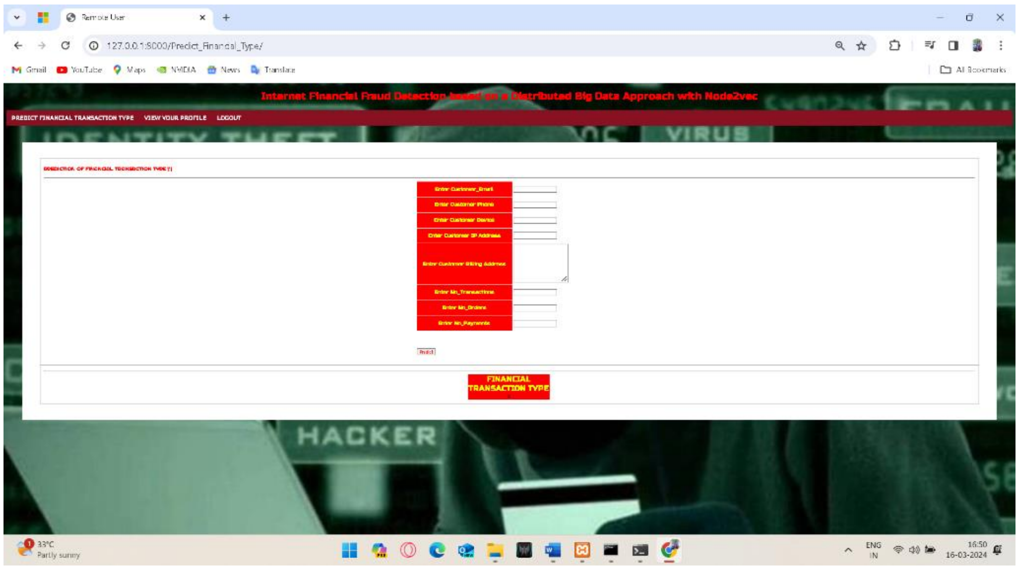
POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

	A	B	C	D	E	F	G	H	I	J
1		customerE	customerF	customerI	customerI	customerE	No_Transa	No_Order:	No_Payme	Fraud
2	0	josephhow	400-108-5	yyeiaxpltff	8.129.104.	5493	2	2	1	0
3	1	evansjeffe	1-788-091	r0jpm7xae	219.173.2	356	3	3	7	1
4	2	andersonw	024.420.0:	4m7h5ipl1	67b7:3db8	8478	5	3	2	0
5	3	rubenjuare	670.664.8:	slovx60t0i	95de:8565	7769	3	3	1	0
6	4	uchen@m.	1-981-877	j0pd24k5h	196.89.23	148	7	7	6	1
7	5	robinsoncy	+65(5)899	hbww39m	43.166.16	65233	1	1	2	1
8	6	samuel15(	5.05E+09	up7afvce5	33.67.35.3	3960	2	1	2	1
9	7	johnlower	044-642-9	rjb0adai34	212.144.6	484	6	5	2	1
10	8	jameslandi	1-885-168	yz4l2bgbvz	c899:ff67:	USNS	5	2	1	0
11	9	ubranch@	(772)964-8	6mpx027n	86.16.65.2	6974	0	0	1	1
12	10	bowenwill	(128)248-4	qmntgod0	231.91.25.	27061	6	5	1	1
13	11	ksummers	1-023-275	2ngkzu1idy	184.97.57.	46383	7	5	4	0
14	12	christinekl	897-464-6	i8cn1uvdk	4304:521a	67636	4	4	3	0

Processed_data	18-03-2024 15:19	Microsoft Excel Co...
Transaction_Datasets	29-12-2021 17:46	Microsoft Excel Co...

Screenshot 5.4: Transactional Datasets

➢ For predicting financial transaction types, you require transactional datasets where you input all transactional details such as customer email, phone, device, IP address, billing address, number of transactions, number of orders, and number of payments. Utilizing various algorithms, the system predicts whether the transaction is fraudulent or not upon pressing the "predict" button.



127.0.0.1:8000/Predict_Financial_Type/

Internet Financial Fraud Detection based on a Distributed Big Data Approach with Node2vec

PREDICT FINANCIAL TRANSACTION TYPE VIEW YOUR PROFILE LOGOUT

PREDICTION OF FINANCIAL TRANSACTION TYPE !!!

Enter Customer_Email

Enter Customer Phone

Enter Customer Device

Enter Customer IP Address

Enter Customer Billing Address

Enter No_Transactions

Enter No_Orders

Enter No_Payments

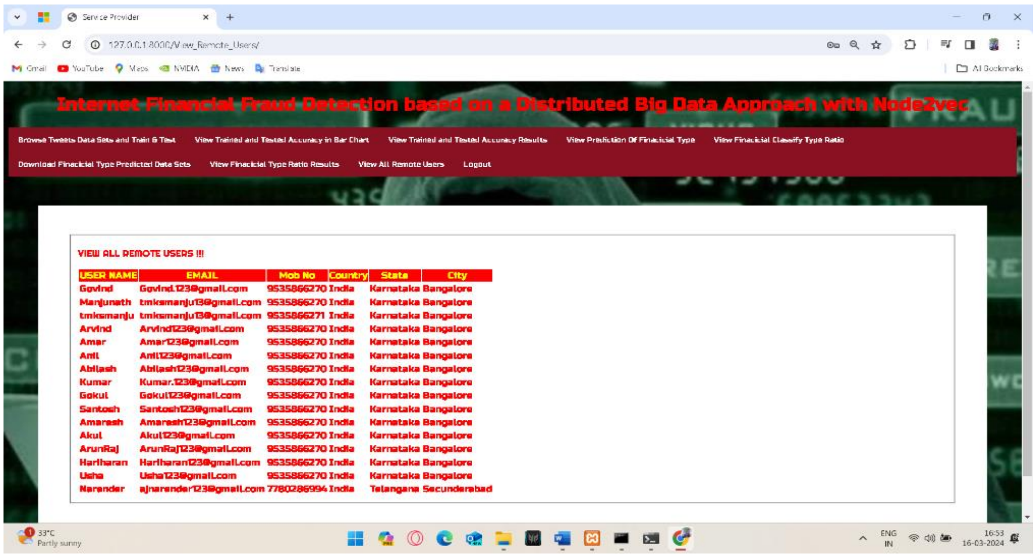
Predict

FINANCIAL TRANSACTION TYPE

Screenshot 5.5: Predicting Financial Transaction Type.

Step 2: The Service Provider has to login by using Admin as user name and password. After login successful he can do some operations such as Login, Browse Tweets Data Sets and Train & Test, View Trained and Tested Accuracy in Bar Chart, View Trained and Tested Accuracy Results, Prediction of Financial Type, Download Financial Type Predicted Data Sets, View Financial Type Ratio Results, View All Remote Users.

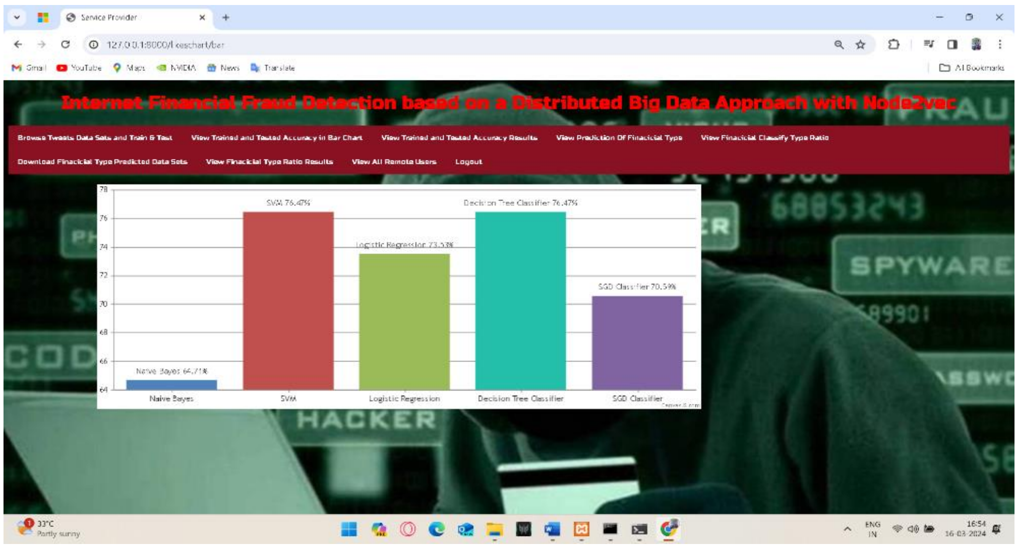
- ➢ These datasets consist of browse tweets from remote users, and they are utilized for both training and testing purposes.


Service Provider
127.0.0.1:8000/View_Remote_Users/
Internet Financial Fraud Detection based on a Distributed Big Data Approach with Node2vec
Browse Tweets Data Sets and Train & Test
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Finacicial Type
View Finacicial Classify Type Ratio
Download Finacicial Type Predicted Data Sets
View Finacicial Type Ratio Results
View All Remote Users
Logout

VIEW ALL REMOTE USERS !!!

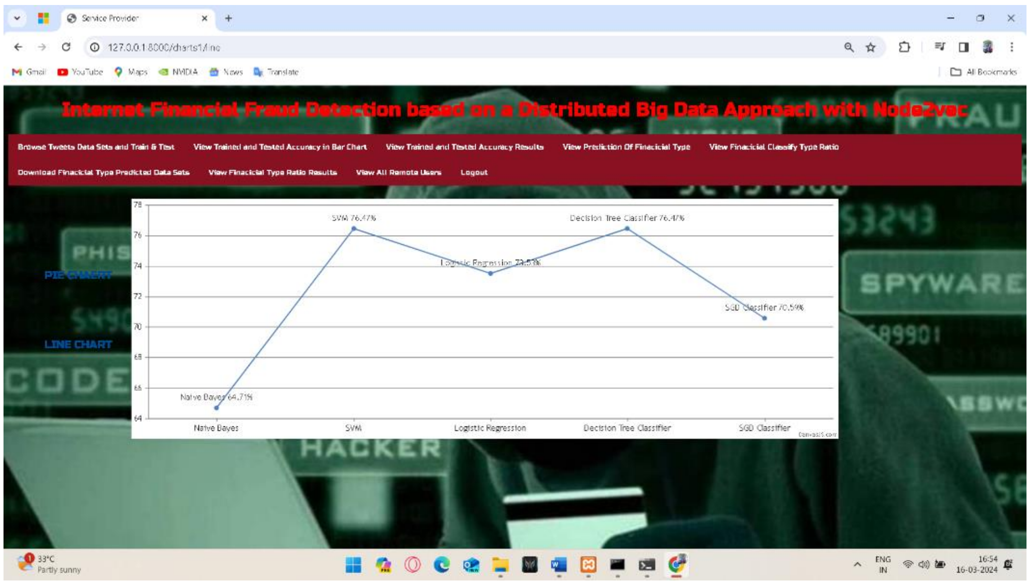
USER NAME	EMAIL	Mob No	Country	State	City
Govind	Govind.123@gmail.com	9535866270	India	Karnataka	Bangalore
Manjunath	tmksmanju13@gmail.com	9535866270	India	Karnataka	Bangalore
tmksmanju	tmksmanju13@gmail.com	9535866271	India	Karnataka	Bangalore
Arvind	Arvind123@gmail.com	9535866270	India	Karnataka	Bangalore
Amar	Amar123@gmail.com	9535866270	India	Karnataka	Bangalore
Anil	Anil123@gmail.com	9535866270	India	Karnataka	Bangalore
Abilash	Abilash123@gmail.com	9535866270	India	Karnataka	Bangalore
Kumar	Kumar.123@gmail.com	9535866270	India	Karnataka	Bangalore
Gokul	Gokul123@gmail.com	9535866270	India	Karnataka	Bangalore
Santosh	Santosh123@gmail.com	9535866270	India	Karnataka	Bangalore
Amaresh	Amaresh123@gmail.com	9535866270	India	Karnataka	Bangalore
Akul	Akul123@gmail.com	9535866270	India	Karnataka	Bangalore
ArunRaj	ArunRaj123@gmail.com	9535866270	India	Karnataka	Bangalore
Hariharan	Hariharan123@gmail.com	9535866270	India	Karnataka	Bangalore
Usha	Usha123@gmail.com	9535866270	India	Karnataka	Bangalore
Narender	ajnarender123@gmail.com	7780286994	India	Telangana	Secunderabad

Screenshot 5.6: Browse Tweets Datasets Consist and train & test.



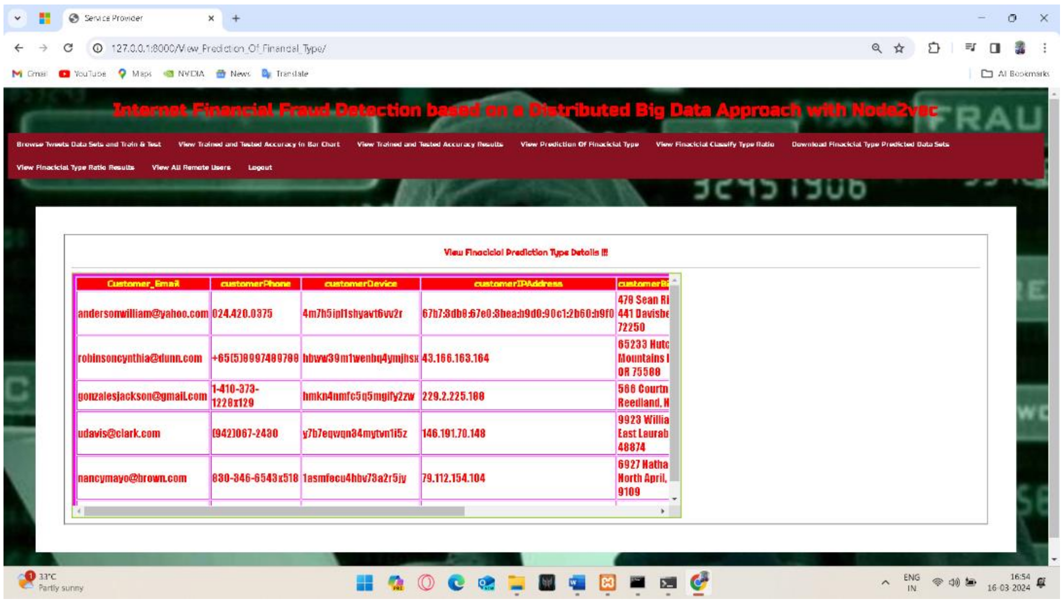
Screenshot 5.7: Comparison Graph Based on Accuracies.

- Trained and Tested Accuracy Results of various algorithms representing in pie chart and line chart.


Internet Financial Fraud Detection based on a Distributed Big Data Approach with Node2vec
Browse Tweets Data Sets and Train & Test
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Finacicial Type
View Finacicial Classify Type Ratio
Download Finacicial Type Predicted Data Sets
View Finacicial Type Ratio Results
View All Remote Users
Logout
PIE CHART
LINE CHART
Naive Bayes 64.71%
SVM 76.47%
Logistic Regression 73.53%
Decision Tree Classifier 76.47%
SGD Classifier 70.59%
Naive Bayes
SVM
Logistic Regression
Decision Tree Classifier
SGD Classifier

Screenshot 5.8: Train and test accuracy results.

- Prediction Of Financial Type whether the user have done any fraud or not.

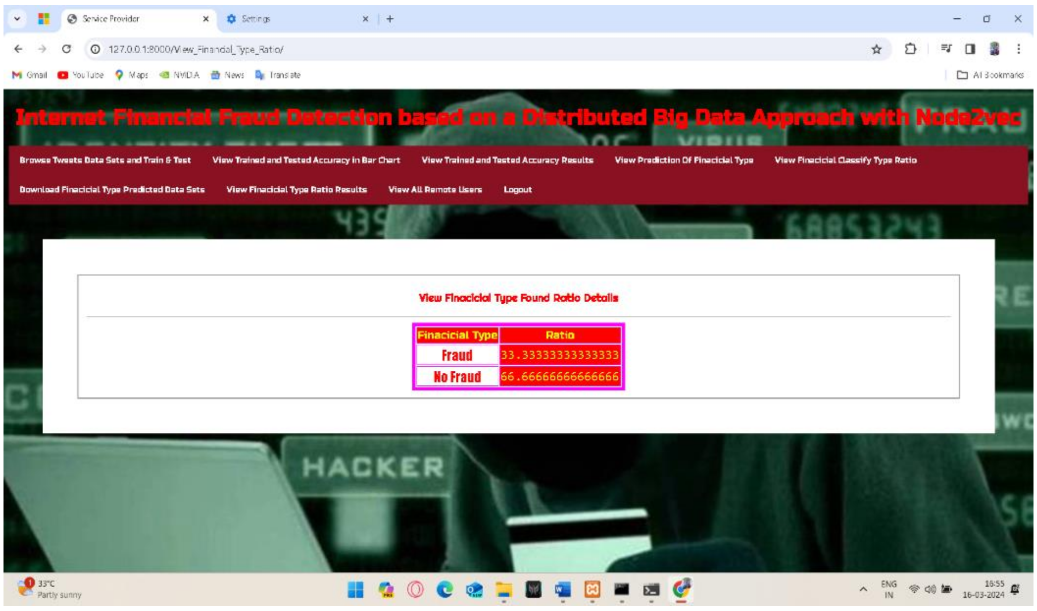

Internet Financial Fraud Detection based on a Distributed Big Data Approach with Node2vec
Browse Tweets Data Sets and Train & Test
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Finacicial Type
View Finacicial Classify Type Ratio
Download Finacicial Type Predicted Data Sets
View Finacicial Type Ratio Results
View All Remote Users
Logout

View Finacicial Prediction Type Details !!!

Customer_Email	customerPhone	customerDevice	customerIPAddress	customerBi
andersonwilliam@yahoo.com	024.420.0375	4m7h5ipl1shyavt6vv2r	67b7:3db8:67e0:3bea:b9d0:90c1:2b60:b9f0	478 Sean Ri 441 Davisbe 72250
robinsoncynthia@dunn.com	+65(5)8997489788	hbvw39m1wenbq4ymjhsx	43.166.163.164	65233 Hutc Mountains I OR 75588
gonzalesjackson@gmail.com	1-410-373-1228x129	hmkn4nmfc5q5mgify2zw	229.2.225.188	566 Courtn Reedland, N
udavis@clark.com	(942)067-2430	y7b7eqvqn34mytvn1i5z	146.191.70.148	9923 Willia East Laurab 48874
nancymayo@brown.com	830-346-6543x518	1asmfecu4hbv73a2r5jy	79.112.154.104	6927 Natha North April, 9109

Screenshot 5.9: Prediction of Financial Type.

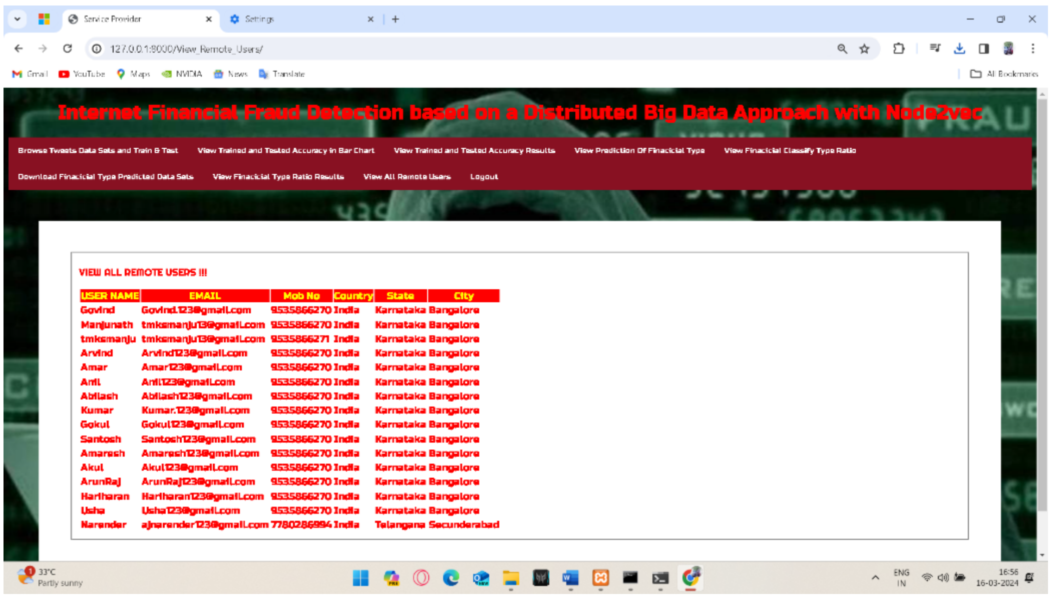
➢ Financial Classify Type Ratio of fraud and not fraud by the users. where you also can Download Financial Type Predicted Data Sets.


127.0.0.1:8000/View_Financial_Type_Ratio/
Internet Financial Fraud Detection based on a Distributed Big Data Approach with Node2vec
Browse Tweets Data Sets and Train & Test
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Finacicial Type
View Finacicial Classify Type Ratio
Download Finacicial Type Predicted Data Sets
View Finacicial Type Ratio Results
View All Remote Users
Logout

View Finacicial Type Found Ratio Details

Finacicial Type	Ratio
Fraud	33.33333333333333
No Fraud	66.66666666666666

Screenshot 5.10: Financial Classify Type Ratio


127.0.0.1:8000/View_Remote_Users/
Internet Financial Fraud Detection based on a Distributed Big Data Approach with Node2vec
Browse Tweets Data Sets and Train & Test
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Finacicial Type
View Finacicial Classify Type Ratio
Download Finacicial Type Predicted Data Sets
View Finacicial Type Ratio Results
View All Remote Users
Logout

VIEW ALL REMOTE USERS !!!

USER NAME	EMAIL	Mob No	Country	State	City
Govind	Govind.123@gmail.com	9535866270	India	Karnataka	Bangalore
Manjunath	tmksmanju13@gmail.com	9535866270	India	Karnataka	Bangalore
tmksmanju	tmksmanju13@gmail.com	9535866271	India	Karnataka	Bangalore
Arvind	Arvind123@gmail.com	9535866270	India	Karnataka	Bangalore
Amar	Amar123@gmail.com	9535866270	India	Karnataka	Bangalore
Anil	Anil123@gmail.com	9535866270	India	Karnataka	Bangalore
Abilash	Abilash123@gmail.com	9535866270	India	Karnataka	Bangalore
Kumar	Kumar.123@gmail.com	9535866270	India	Karnataka	Bangalore
Gokul	Gokul123@gmail.com	9535866270	India	Karnataka	Bangalore
Santosh	Santosh123@gmail.com	9535866270	India	Karnataka	Bangalore
Amaresh	Amaresh123@gmail.com	9535866270	India	Karnataka	Bangalore
Akul	Akul123@gmail.com	9535866270	India	Karnataka	Bangalore
ArunRaj	ArunRaj123@gmail.com	9535866270	India	Karnataka	Bangalore
Hariharan	Hariharan123@gmail.com	9535866270	India	Karnataka	Bangalore
Usha	Usha123@gmail.com	9535866270	India	Karnataka	Bangalore
Narender	ajnarender123@gmail.com	7780286994	India	Telangana	Secunderabad

Screenshot 5.11: View All Remote Users.

6.TESTING

6. TESTING

6.1 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, subassemblies, assemblies and or afinished 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 tests. Each test type addresses a specific testing requirement.

6.2 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.3 TYPES OF TESTING

6.3.1 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 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

6.3.3 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 are focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identifying Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing iscomplete, additional tests are identified and the effective value of current tests is determined.

6.3.4 INTEGRATION TESTING

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

The following are the types of Integration Testing:

1. Top-Down Integration

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

2. Bottom-up Integration

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom-up integration strategy may be implemented with the following steps:

- ❖ The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
- ❖ The cluster is tested.

6.3.5 USER ACCEPTANCE TESTING

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

6.3.6 OUTPUT TESTING

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

6.4 TEST CASES

6.4.1 CLASSIFICATION

S.NO	Test Case	ExceptedResult	Result	Remarks (IF fails)
1.	Transactional fraud Identification	The system accurately identifies Transactional frauds.	Pass	Successful detection of Transactional frauds
2.	False Positive Minimization	Ensure the system does not misclassify harmless frauds	Pass	Minimization of legitimate transactions falsely flagged as fraud
3.	Browse tweets datasets	Show our dataset	Pass	Fails if there is no dataset in the database.
4.	Train & Test Machine learning model	Successful training &Accurate prediction of fraud	Pass	If Model accuracy too low.
5.	Predict Financial type.	Display Review with true results	Pass	Results not True Fail
6.	Show Detection process	Display Detection process	Pass	Results Not True Fail
7.	View Financial type ratio	Display financial Type ratio	Pass	If Results not Displayed Fail.
8.	Data Privacy Compliance	System complies with data privacy regulations and protects sensitive information	Pass	If Privacy regulations are not followed
9.	Cross-validation	Validate model performance across different datasets and time periods	Pass	If Model performs poorly on unseen data
10.	Scalability	System can handle increasing volume of transactions	Pass	If System becomes slow with large datasets

7.CONCLUSION

7. CONCLUSION AND FUTURE SCOPE

7.1 PROJECT CONCLUSION

The occurrences of Internet financial fraud cases have caused huge losses to commercial banks or financial institutions. In order to enhance the efficiency of financial fraud detections, an intelligent and distributed Big Data approach is proposed in this article. The approach mainly includes four modules: data pre-processing module, normal data feature module, graph embedding module, prediction module. The graph embedding algorithm Node2Vec is implemented on Spark GraphX and Hadoop to learn and represent the topological features of each vertex in the network graph into a low-dimensional dense vector, so as to improve the classification effectiveness of deep neural network and predict the fraudulent samples of the dataset. The experiments evaluate the indicators of precision rate, recall rate, and the results show that due to the Node2Vec properties of structural equivalence and homophily, the features of samples can be better learned and represented and the proposed approach is better than the comparative methods. In future work, the inductive graph embedding network algorithms, such as GraphSage, Pins age, etc., would be improved and implemented to effectively learn the features of newly generated vertices in a dynamic network graph, so as to achieve the better effect of financial fraud detection.

7.2 FUTURE SCOPE

In the future, the "Internet Financial Fraud Detection based on a Distributed Big Data Approach with Node2vec" project holds significant potential for further development. This includes improving the accuracy of fraud detection models, exploring advanced graph analytics techniques like Graph Neural Networks, and implementing real-time fraud prevention mechanisms. Continued focus on regulatory compliance, user education, and ethical AI practices will be essential. By embracing innovation and staying responsive to evolving fraud threats, the project can continue to enhance online financial security and contribute to a safer digital economy.

8.BIBLOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

1. A. Srivastava, A. Kundu, and S. Sural, “Credit card fraud detection using hidden Markov model,” IEEE Transactions on Dependable and Secure Computing, vol. 5, no. 1, pp. 37-48, 2008.
2. C. C. Aggarwal, Y. Zhao, and S. Y. Philip, “Outlier detection in graph streams,” in Proc. IEEE 27th International Conference on Data Engineering, 2011, pp. 399-409.
3. E. L. Paula, M. Ladeira, and R. N. Carvalho, “Deep learning anomaly detection as support fraud investigation in Brazilian exports and anti-money laundering,” in Proc. 15th IEEE International Conference on Machine Learning and Applications, pp. 954-960, 2016.
4. A. Grover and J. Leskovec, “Node2vec: scalable feature learning for networks,” in Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 855-864.
5. V. Jambulapati and J. Stavins, “Credit card act of 2009: what did banks do?” *Banking & Finance*, vol. 46, no. 9, pp. 21-30, 2014.

8.2 GITHUB LINK

https://github.com/SugamanchiNarender/INTERNEaT-FINANCIAL-FRAUD-DETECTION