“Jnana Sangama”, Belagavi-590018, Karnataka

Report
on

“Detect Transaction Anomalies in Credit Card System using Machine Learning”

Submitted in partial fulfillment of the requirements for the award of
the degree of Bachelor of Engineering
in
Computer Science & Engineering

Submitted by

USN	Name
1BI19CS011	AKASH JAIN
1BI19CS015	AMAN ADITYA PANDEY
1BI19CS026	ARINDAM DUTTA
1BI19CS141	SHIVASHANKAR TADAKI

Under the Guidance of
Dr. MAYA B S

Assistant Professor
Dept. of CS&E, BIT

Bengaluru-560004

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BANGALORE INSTITUTE OF TECHNOLOGY

K.R. Road, V.V. Puram, Bengaluru-560 004
2022-23

“Jnana Sangama”, Belagavi-590018, Karnataka

BANGALORE INSTITUTE OF TECHNOLOGY

Bengaluru-560 004

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate

This is to certify that the project work entitled “**Detect Transaction Anomalies in Credit Card System using Machine Learning**” carried out by

USN	Name
1BI19CS011	AKASH JAIN
1BI19CS015	AMAN ADITYA PANDEY
1BI19CS026	ARINDAM DUTTA
1BI19CS141	SHIVASHANKAR TADAKI

bonafide students of VIII semester B.E. for the partial fulfillment of the requirements for the Bachelor's Degree in Computer Science & Engineering of the **VISVESVARAYA TECHNOLOGICAL UNIVERSITY** during the academic year 2022-23. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said degree.

Dr. Maya B S	Dr. Girija J	Dr. Aswath M U
Assistant Professor	Prof. & Head Dept. of CSE	Principal, BIT

External Viva

Name of the Examiners	Signature with date
1.	
2.	

ACKNOWLEDGEMENT

The knowledge & satisfaction that accompany the successful completion of any task would be incomplete without mention of people who made it possible, whose guidance and encouragement crowned my effort with success. We would like to thank all and acknowledge the help we have received to carry out this final year project.

We would like to convey our sincere thanks to our college the **Bangalore Institute of Technology, Dr. Aswath M U, Principal,** for being kind enough to provide an opportunity and platform to complete and present our final year project "**Detect Transaction Anomalies in Credit Card System using Machine Learning".**

We would also like to thank **Dr. Girija J,** Professor and Head of the Department for Computer Science and Engineering, Bangalore Institute of Technology, for her constant encouragement and making us believe in ourselves and ultimately present our final year project "**Detect Transaction Anomalies in Credit Card System using Machine Learning".**

We would also like to express our gratitude to the project coordinators, **Dr. Gunavathi H S & Prof. Mamatha V**, Assistant Professor for coordinating the project activities.

We are most humbled to mention the enthusiastic influence provided during the development and ideation phase by our guide **Dr. Maya B S**, Assistant Professor, Department of Computer Science & Engineering for her ideas, time to time suggestions, and co-operation shown during the venture and help make our final year project a success.

AKASH JAIN (1BI19CS011)
ARINDAM DUTTA (1BI19CS026)
AMAN ADITYA PANDEY(1BI19CS015)
SHIVASHANKAR TADAKI (1BI19CS141)

ABSTRACT

Credit card fraud detection is currently a prevalent issue worldwide, driven by the increasing number of online transactions and e-commerce platforms. Instances of credit card fraud occur when unauthorized individuals gain access to stolen card information and use it for fraudulent purposes. This ongoing problem necessitates the implementation of credit card fraud detection systems. The primary focus of this project revolves around utilizing machine learning algorithms for fraud detection. Specifically, the Random Forest algorithm and the Adaboost algorithm are employed. The performance evaluation of these algorithms is based on metrics To visually represent the results, a ROC curve is generated using the confusion matrix. By comparing the Random Forest and Adaboost algorithms, the algorithm with the highest is identified as the most effective for fraud detection.

Digital transactions can take place over the phone or on the internet. For executing a transaction, very basic information is required such as expiry date, card number, card verification number etc. Cardholders provide this information through phone or the internet. Fraudsters apply several techniques and attempt to steal the credit card information of the customers so that they can use it for doing fraudulent transactions. It is a very serious, and costly problem for financial service providers. Billions of dollars are subject to fraudulent transactions every year. The fraudulent transaction is an issue of concern for all the credit card providers or by expansion for all the financial systems that provide the facilities for online transactions to their customers. It is usually the result of someone stealing the credit card information of the customers which also impact the brand value of the credit card service providers and the merchants. The identification of fraudulent activities involves the constant monitoring of user populations to detect, understand, or prevent undesirable behaviors, including fraud, intrusion, and defaulting. This is a highly significant issue that requires the attention of fields like machine learning and data science, as it can be effectively addressed through automated solutions. From a learning standpoint, this problem poses several challenges, especially due to factors such as class imbalance. The majority of transactions are legitimate, greatly outnumbering fraudulent ones. Furthermore, transaction patterns frequently exhibit changes in their statistical characteristics over time.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INRODUCTION

1.1 Overview

Despite the promising progress made in detecting anomalies in day-to-day transactions, identifying frauds remains a challenging task due to semantic gap between the various predefined fraud detection models and diversities in implementing them. The hybrid approach made it possible to train generative models for the system to maintain the privacy as the transaction data is kept in a decentralized manner using the concept of federated learning.

In this figure 1.1, The issue at hand is highly pertinent and requires the dedicated efforts of communities like machine learning and data science. The application of automated solutions holds immense potential in addressing this problem. From a learning perspective, this challenge is particularly complex due to various factors, including class imbalance. The presence of imbalanced classes adds an additional layer of difficulty to the problem, making it a noteworthy.

Digital transactions can take place over the phone or on the internet. For executing a transaction, very basic information is required such as expiry date, card number, card verification number etc. So accidental disclosure of any one of them leads to serious disruptions.

Fig 1.1. Introduction to Credit Card Fraud Detection

1.2 Objectives

- Design and implement the technique for detecting anomalies in the credit card systemusing the existing algorithms.
- Customizing the existing bank authentication system. Implement the messageauthentication system for successful transaction to the user.
- Ability to identify new customer behavior patterns and adapt to changes.
- Unlike rule-based systems, algorithms are to be aligned with a constantly changing environment and financial conditions.

1.3 Purpose, Scope, and Applicability

1.3.1 Purpose

- Fraud detection costs huge money loss to different financial companies and consumers soit have become essential for banks and financial institutions to minimize their losses.

- With digital crime and online fraud of all kinds on the rise, it‟s more important than ever for organizations to take firm and clear steps to prevent payment card fraud through advanced technology and strong security measures.

1.4 Scope

The scope of this project is to classify the transactions are fraudulent or not,

- Obtain factual and accurate information that will lead to an appropriate credit decision.
- Predict fragility based on the transaction amount, location, and other transaction related data.

1.5 Applications

- Customer Service
- Banking
- Payment Gateways
- Merchandise Payments

Chapter 2

LITERATURE SURVEY

2.1 Review of Machine Learning Approach on Credit Card Fraud Detection [1]

Proposed Idea:

- The idea is utilizing the real-time datasets to train the model in a privacy-preserving manner.
- A Federated learning(decentralized) framework Fig.2.1 with ANN can enhance the capability of the ML model to detect fraudulent transactions

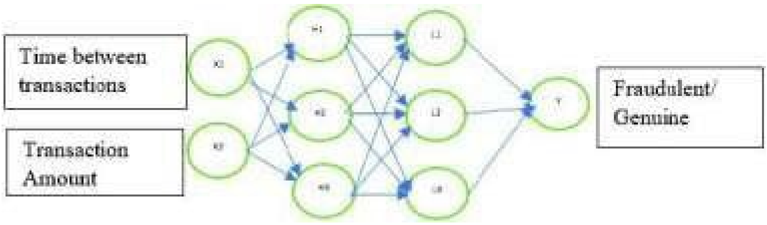


Fig 2.1. Flowchart of the process of detection of CC fraud detection

Conclusion: The efficient method to identify an affected person so as to save the users is proposed in this paper . Also, Privacy of customers of the banks are preserved.

Drawback: Although data is not shared centrally, even the trained model will be going to learn patterns that can be possibly decoded by hackers. Therefore, while keeping the limitations in place, there still needs to be work done for gaining the confidence of banks and financial institutes to adopt this technology.

2.2 Credit Card Fraud Detection using Machine Learning Algorithm [2]

Proposed Idea:

- Customers are grouped based on their transactions and extract behavioral patterns to develop a profile for every cardholder.
- Then different classifiers are applied on three different groups (low ,medium, high) ,later rating scores are generated for every type of classifier.
- This dynamic changes in Fig 2.2 parameters lead the system to adapt to new cardholder's transaction behaviors timely. Followed by a feedback mechanism to solve the problem of concept drift.

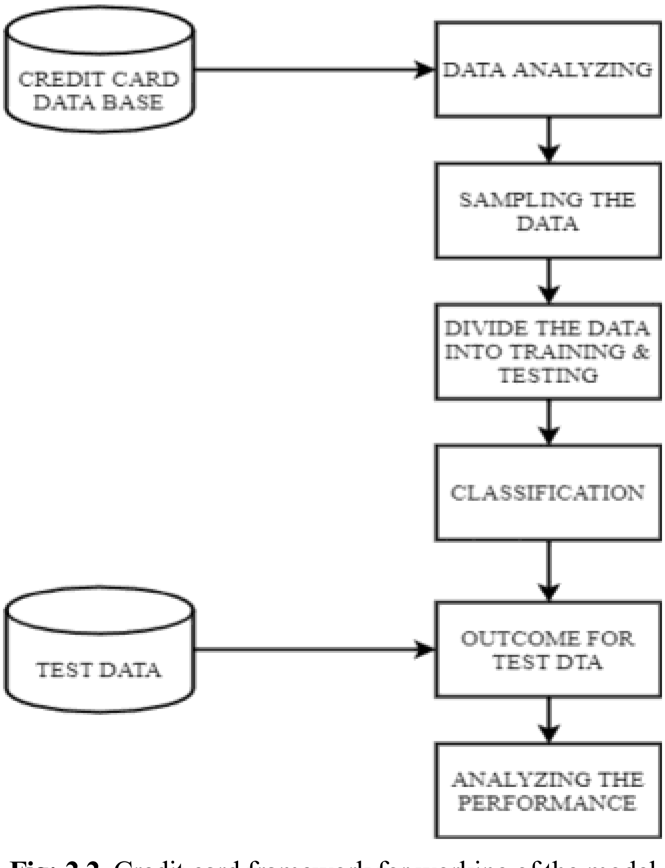


Fig: 2.2. Credit card framework for working of the model

Conclusion: It tackles the problem of concept drift.

2.3 Implementing Machine Learning in detecting transaction anomalies [3]

Proposed Idea:

- The main aim of this paper is to classify the transactions that have both the fraudand non-fraud transactions in the dataset using algorithms like that the Random Forest and the Adaboost algorithms. Then these two algorithms are compared to choose the algorithm that best detects the credit card fraud transactions includes many steps from gathering dataset to deploying model and performing analysis based on results.
- In this model Fig 2.3 we take the Kaggle dataset and pre-processing is to be done for the dataset. We split the data into the training data and the testing data.
- We use the training data to prepare the Random Forest and the Adaboost models. Then we develop both the models. Finally, the accuracy, precision, recall, and F1-score is calculated for the models.

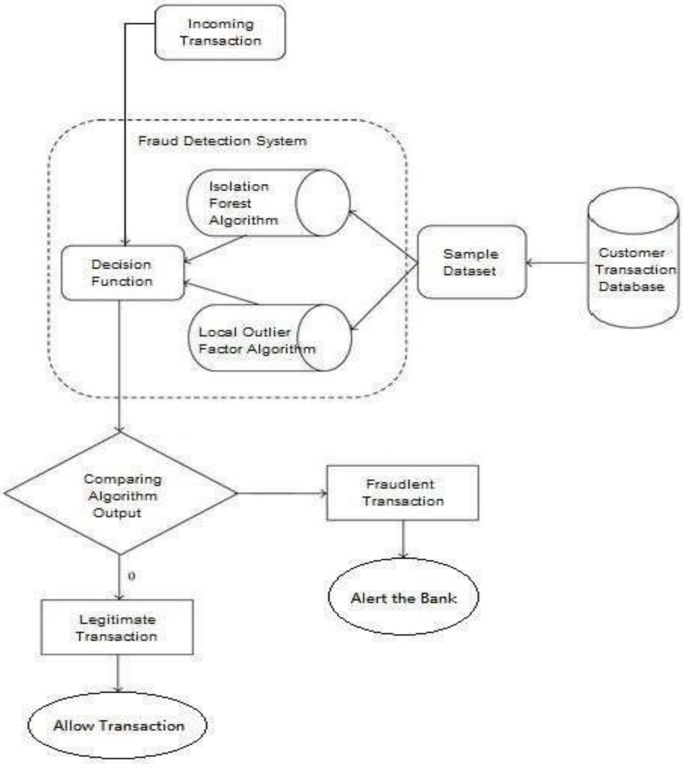


Fig 2.3. Work Flowchart of CC fraud detection data

Main Drawback: To achieve good accuracy and low computation time they selected just 10 features from the dataset.

2.4 GA algorithm for feature selection and enhancement [4]

Proposed Idea:

- we implement a feature selection algorithm that is based on the Genetic Algorithm (GA) using the RF method in its fitness function. The RF method is used becauseit can handle many input variables, can automatically handle missing values, and is not affected by noisy data.
- A GA-based FS in order to increase the performance of ML based models applied to the domain of intrusion detection systems Fig 2.4. of Evolutionary Algorithm (EA) that is often used to solve several optimization tasks with a reduced computational.

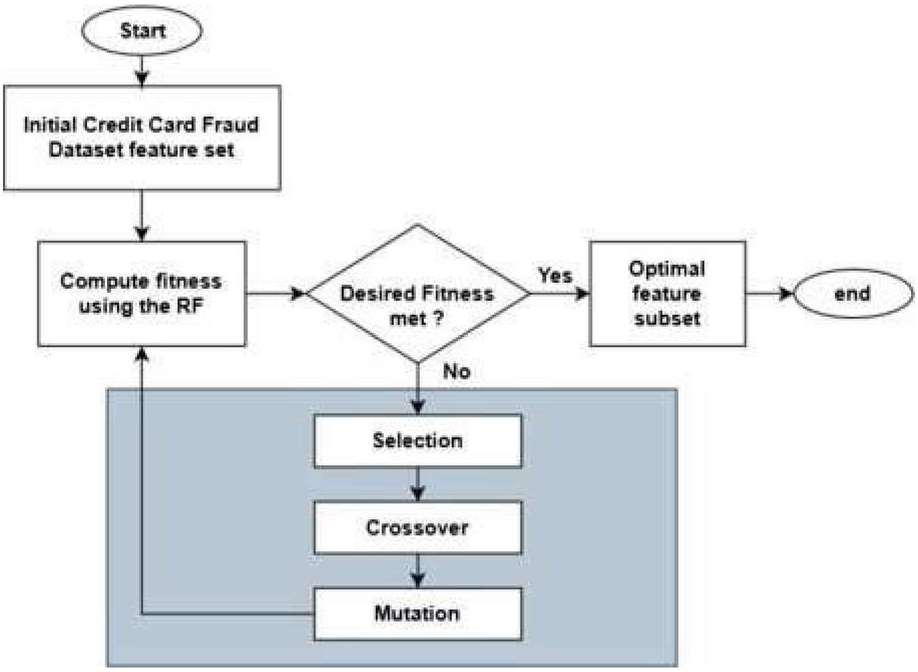


Fig 2.4 Architecture of CC fraud detection

2.5 Implementing Hybrid model in the Fraud Detection [5]

Proposed Idea:

- In this paper, we have studied the behavior pattern based on their previous transaction records. They classify all the attributes of transaction and construct the logical graph of behavior profile (LGBP).

- They don't require predictive model Fig 2.5 and their outlier detection mechanism helps to detect the card fraud using less memory and computation requirements.

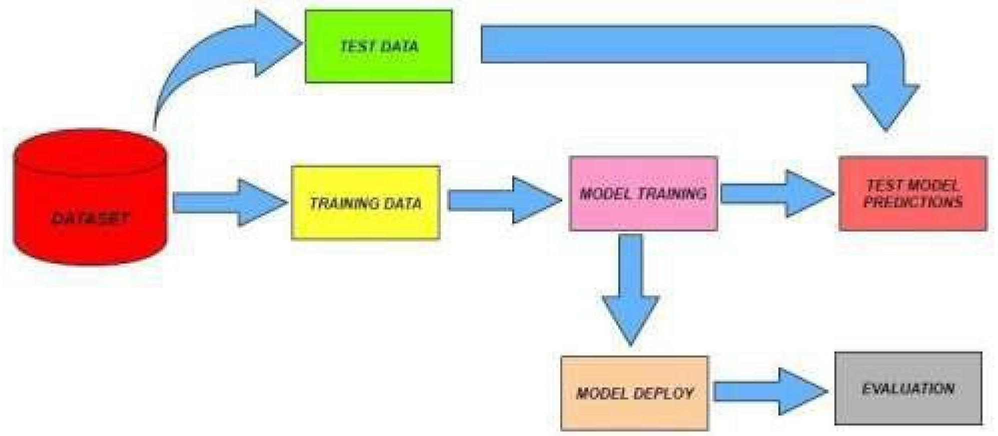


Fig 2.5. Framework of the CC fraud detection model

Conclusion: We have done a comparative study of the results of K-Nearest Neighborsand Convoluted Neural Network and the hybrid model of both. Among the individual models, The KNN had the highest accuracy rate of 90.66%, followed by CNN with 88.12% accuracy. Upon hybridization, the resultant model had accuracy of 98%.The accuracy of the Convolutional Neural Network increased by 10% when made into a hybrid model with K-Nearest Neighbors, and would only improve if trained over larger balanced dataset.

2.6 Credit Card Fraud Detection using Deep Learning [6]

Proposed Idea:

- In this paper, we put forth a method of Fraud Detection which is completely based on Deep Learning. We first compare it with all the renowned methods such as Random Forest, Support Vector Machines Fig 2.6, etc.

- Finally, we come across a conclusion that Neural Networks, even though harder to train, would be a perfect fit for the Model.

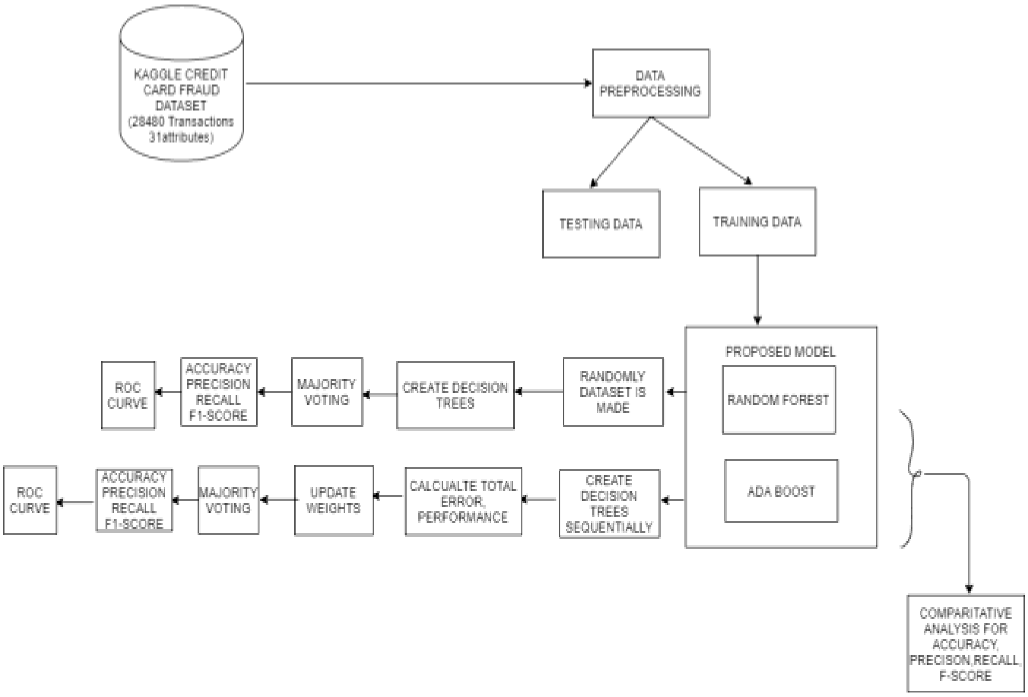

KAGGLE CREDIT CARD FRAUD DATASET (28480 Transactions 31attributes)
DATA PREPROCESSING
TESTING DATA
TRAINING DATA
PROPOSED MODEL
RANDOM FOREST
ADA BOOST
RANDOMLY DATASET IS MADE
CREATE DECISION TREES
MAJORITY VOTING
ACCURACY PRECISION RECALL F1-SCORE
ROC CURVE
CREATE DECISION TREES SEQUENTIALLY
CALCUALTE TOTAL ERROR, PERFORMANCE
UPDATE WEIGHTS
MAJORITY VOTING
ACCURACY PRECISION RECALL F1-SCORE
ROC CURVE
COMPARITATIVE ANALYSIS FOR ACCURACY, PRECISON,RECALL, F-SCORE

Fig 2.6. Architecture of the CC fraud detection framework model

Drawback: we can conclude that, Kera"s based Deep Learning Neural Network proves to be a great alternative to other classifiers mentioned above. Also, no matter how accurate the trained model of the network might be, it will not show accurate results unless the skewness of the data is reduced.

2.7 Credit Card Fraud Detection using Machine Learning and Data Science [7]

Proposed idea:

- This article has listed out the most common methods of fraud along with their detection methods(outliers) and reviewed recent findings in this field.

- This paper has also explained in detail Fig 2.7, how machine learning can be appliedto get better results in fraud detection along with the algorithm, pseudocode, explanation its implementation and experimentation results.

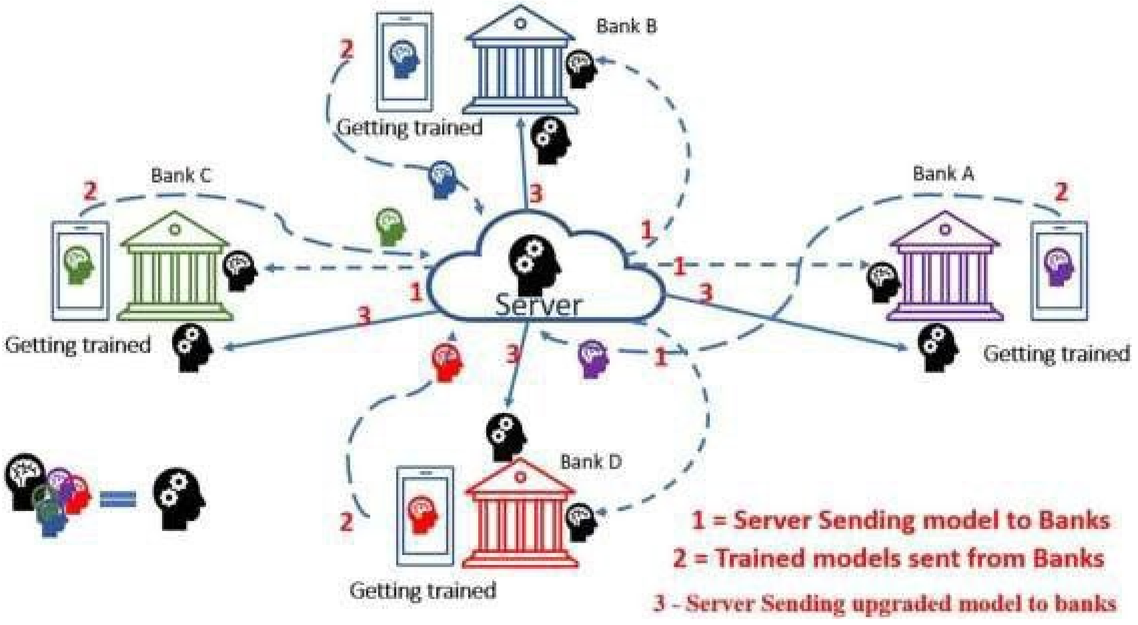

Bank B
Getting trained
Bank C
Getting trained
Bank A
Getting trained
Server
Bank D
Getting trained
1
2
3
1 = Server Sending model to Banks
2 = Trained models sent from Banks
3 - Server Sending upgraded model to banks

Fig 2.7. Workflow of training and testing our models

Drawbacks: This idea is difficult to implement in real life because it requires the cooperation from banks, which aren‟t willing to share information due to their market competition, and due to legal reasons and protection of data of their users.

2.8 Autonomous credit card fraud detection using machine learning approach [8]

Proposed idea:

- This paper proposes a Machine Learning models such as Naive bayes, SVM, ANN, and LSTM-RNN which have been utilized to detect fraud in the credit card system.
- The suggested system‟s performance is measured using sensitivity, precision, accuracy, and error rate.
- Traditional techniques are no longer effective in the age of big data. As a result,the team developed a model for detecting credit card fraud based on the Long Short-Term Memory technique using an actual data set of credit card fraud.
- This model was created to improve current detection tactics as well as detection accuracy in light of big data.
- It used deep learning techniques Fig 2.8 to quickly and effectively identify patterns, overcoming the difficulty of recognizing unexpected and sophisticated fraud practices.

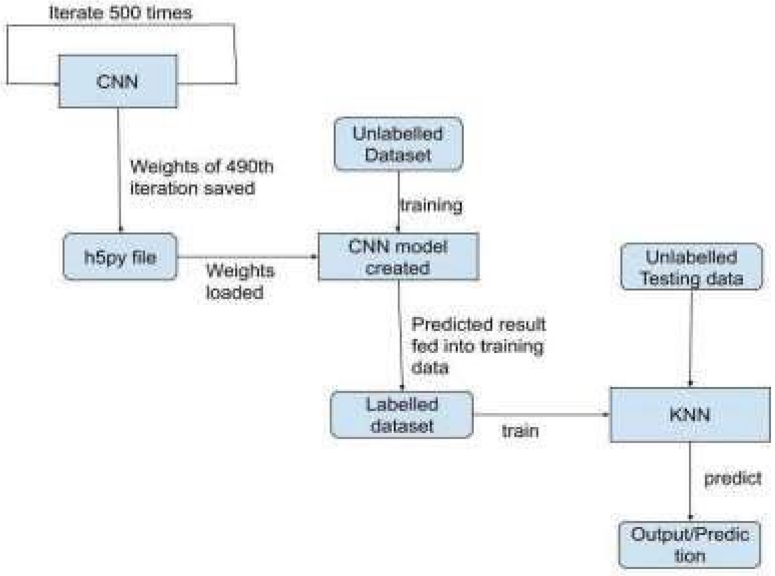


Fig 2.8. Overall process of proposed methodology

Drawbacks: The watershed algorithm used is highly sensitive to noise. If there is an image with noise, then it will influence the segmentation.

2.9 Imbalanced Classification Approaches for Credit Card Fraud Detection [9]

Proposed Idea:

- This research paper presents a comprehensive investigation into the imbalance classification problem, focusing on rigorous experimentation and performance comparison of various solutions.
- Two techniques are employed in imbalanced classification approaches Fig 2.9. The first technique is employed on data as a prepossessing step to balance classes, like oversampling, under sampling, etc. The second technique is used within the classification algorithm like Cost-Sensitive (CS) approaches or One-Class Classification (OCC).

1. Random Oversampling (RO) – It is used to balance classes by simply replicating observations as needed until the balance between classes is reached.
2. One Class Classification (OCC) – This approach uses only one class of the data(usually the minority class) and learns its characteristics.

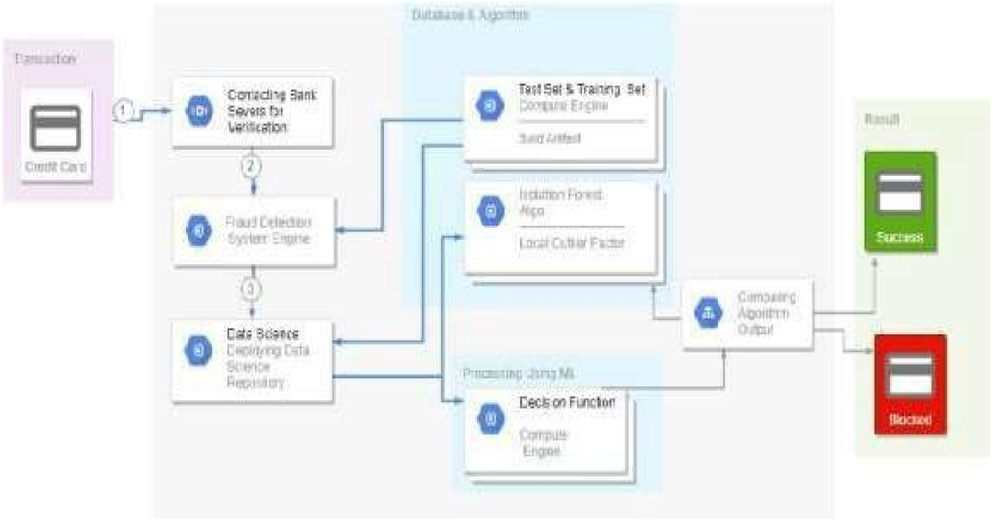


Fig 2.9. Framework of CC fraud detection

2.10 Credit Card Fraud Detection Using AdaBoost and Majority Voting [10]

Proposed idea:

- This paper proposes single and hybrid machine learning algorithms for financial applications. Various financial applications and financial statements are reviewed.

  1. **Single Models** - For credit card fraud detection, Random Forest (RF), Support Vector Machine,(SVM) and Logistic Regression (LOR) will be examined.

  2. **Hybrid Models** - Hybrid models are combination of multiple individual models. A hybrid model consisting of the Multilayer Perceptron (MLP) neural network, SVM, LOR, and Harmony Search (HS) optimization.

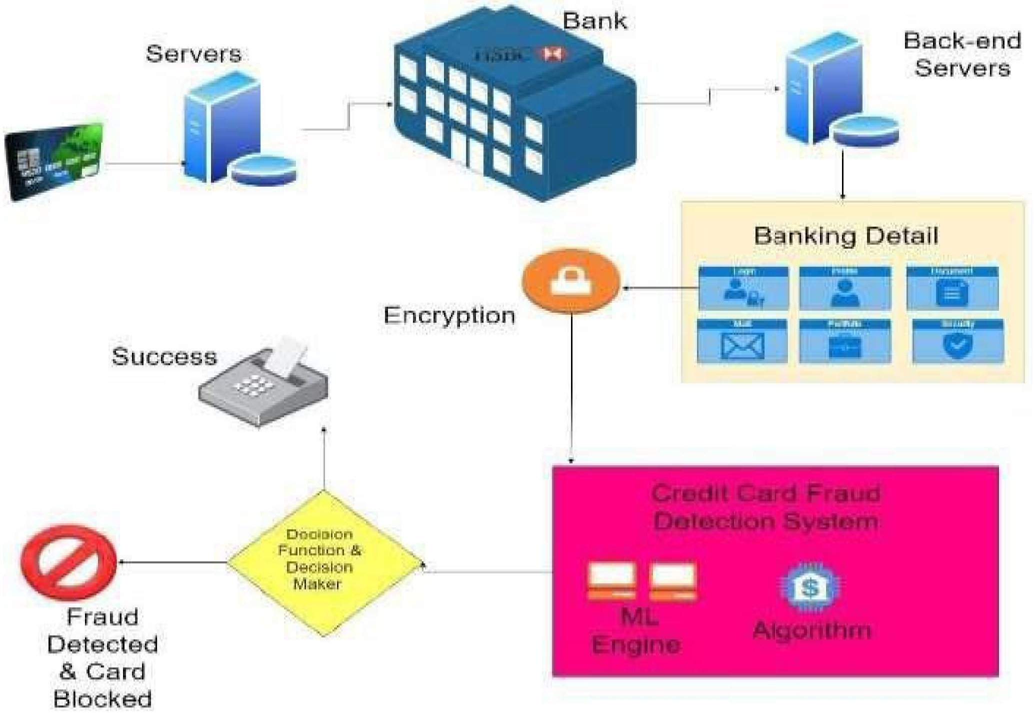

Servers
Bank
Back-end Servers
Banking Detail
Encryption
Success
Decision Function & Decision Maker
Credit Card Fraud Detection System
ML Engine
Algorithm
Fraud Detected & Card Blocked

Fig 2.10. Block diagram of proposed system

Conclusion: A total of twelve algorithms are used in this experimental study. They are used in conjunction with the Ada Boost and Majority Voting methods.

Chapter 3

REQUIREMENT ENGINEERING

3.1 Software and Hardware Tools Used

3.1.1 Software Requirements

Python 3	Python is a high-level, general-purpose programming language that is interpreted rather than compiled. It is widely used and known for its simplicity and readability. Python 3, the latest major version of Python, adheres to the design philosophy of emphasizing code readability.
Pip	Pip is a software package-management system that is implemented in Python. It serves as a tool for installing, managing, and uninstalling software packages in the Python ecosystem.
NumPy	NumPy is a Python library that enhances the Python programming language by introducing robust support for handling large, multi-dimensional arrays and matrices. It offers an extensive collection of high-level mathematical functions that enable efficient operations on these arrays.
Anaconda	Anaconda is a comprehensive distribution of the Python and R programming languages, specifically designed for scientific computing and data analysis. It provides a bundled collection of commonly used libraries, tools.
Google Collab	Collaboratory, often referred to as Colab, is a product developed by Google Research. It is an online platform that enables users to write and execute Python code directly through a web browser.

3.1.2 Hardware Requirements

- Processor: Intel i5 10th gen or more.
- Ethernet connection (LAN) Oral wireless adapter (Wi-Fi).
- GPU - NVIDIA GeForce MX150- 4GBorabove.
- Memory (RAM): Minimum4 GB; Recommended16GB or above.

3.2 Conceptual Modeling

3.2.1 Use Case Diagram

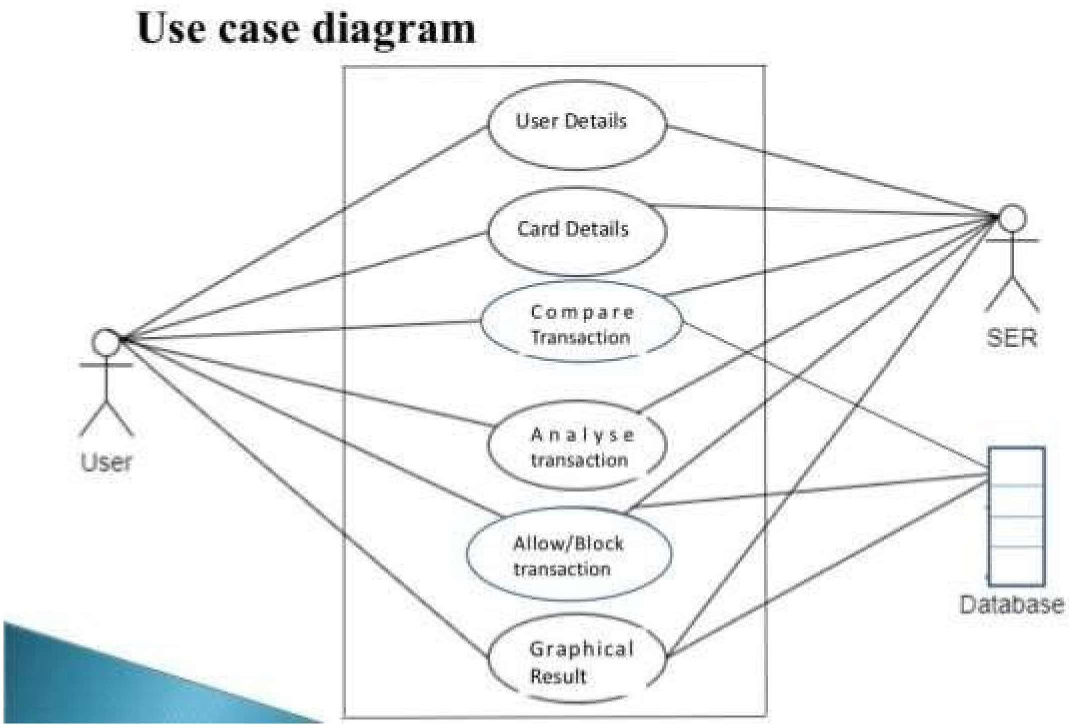


Fig 3.1 Use case diagram

In a credit card use case, various scenarios can arise when a customer initiates a transaction. These transactions can fall into two categories: those that are successfully processed and settled through the payment system and those that are not originally submitted through the payment gateway. The use case involves a validation process to verify the authenticity and integrity of the transaction. Any anomalies detected during this validation process trigger appropriate actions, such as blocking the transaction and providing relevant feedback to the user. The goal is to ensure that only legitimate and secure transactions are processed, while potential fraudulent or suspicious activities are promptly identified and handled accordingly.

3.2.2 Sequence Diagram

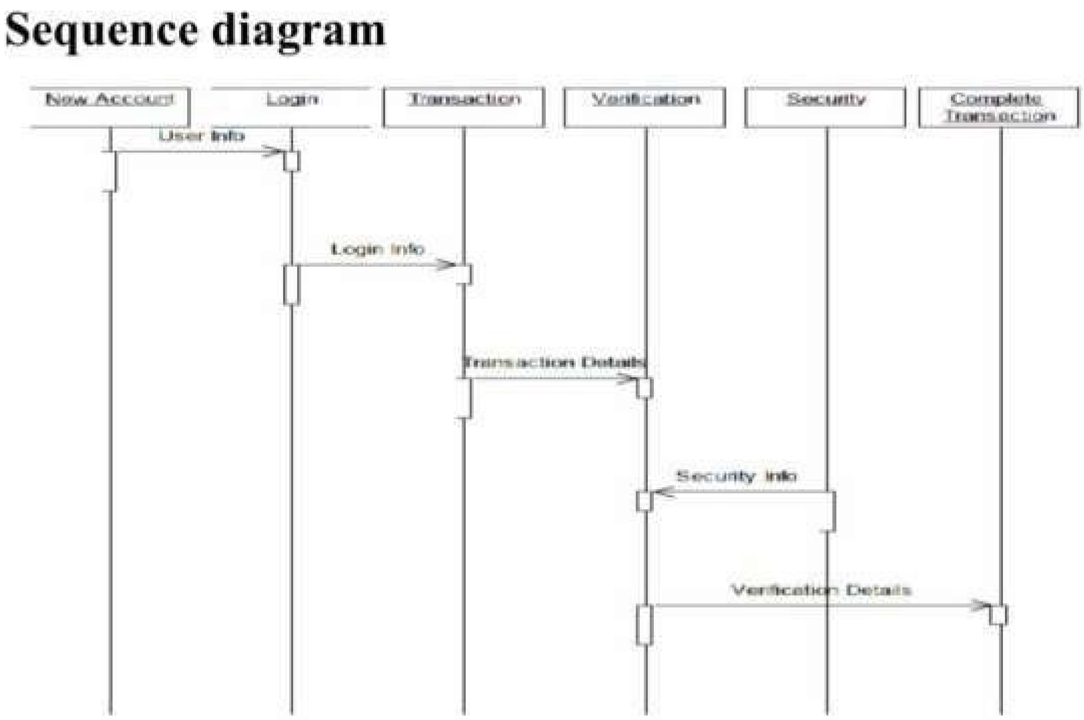


Fig 3.2 Sequence Diagram

The sequence diagram serves as a visual representation of the sequential flow of steps within a system, also known as an event diagram. It aids in conceptualizing various dynamic scenarios by illustrating the communication and interactions between different lifelines or entities involved in the system. The diagram captures the sequence of events in a time-ordered manner, depicting how the lifelines participate in the runtime of the system.

In the specific case of credit card processing, the sequence diagram showcases the steps involved in the process. It begins with the user entering their credit card details, followed by subsequent actions taken by the security system to verify the user's information. The diagram provides a clear visualization of the chronological order of events and the interactions between the user and the security system during the credit card verification process.

3.2.3 Activity Diagram

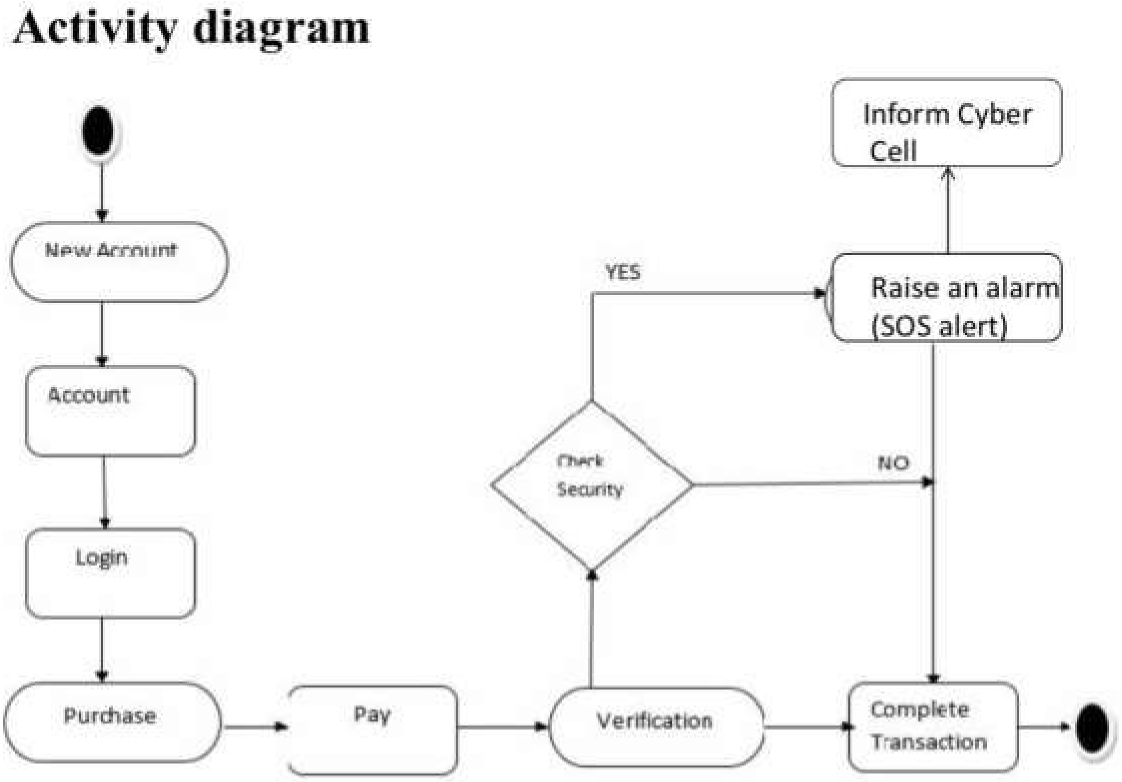


Fig 3.3 Activity Diagram

In UML, the activity diagram is used to demonstrate the flow of control within the system rather than the implementation. It models the concurrent and sequential activities.

The activity diagram helps in envisioning the workflow from one activity to another. It put emphasis on the condition of flow and the order in which it occurs. The flow can be sequential, branched, or concurrent, and to deal with such kinds of flows, the activity diagram has come up with a fork, join, etc.

Here in this case, It depicts the use of a credit card while purchasing any product and how the different activities are being done at the same moment – security check, verification etc and accordingly final decision will be made internally whether to complete the transaction or raise an alarm in case of fraud detection.

3.2.4 State Diagram

As seen in the figure 3.4, the state diagram displays the state in which our system will be at finite instances of time.

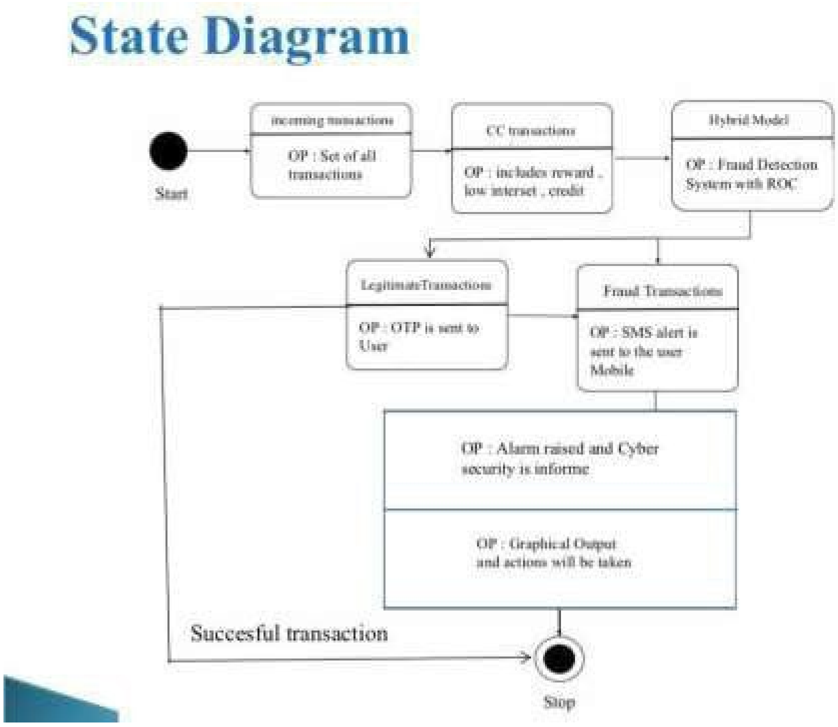

State Diagram
Start
incoming transactions
OP : Set of all transactions
CC transactions
OP : includes reward , low interset , credit
Hybrid Model
OP : Fraud Detection System with ROC
LegitimateTransactions
OP : OTP is sent to User
Fraud Transactions
OP : SMS alert is sent to the user Mobile
OP : Alarm raised and Cyber security is informe
OP : Graphical Output and actions will be taken
Succesful transaction
Stop

Fig 3.4. State Diagram

The state diagram is also called the State chart or State Transition diagram, which shows the order of states underwent by an object within the system. It captures the software system's behavior. It models the behavior of a class, a subsystem, a package, and a complete system.

Here in this case, it shows the processing of the model where when a transaction is being made and it checks whether it is being done by credit card or other card and then further steps are being followed – otp verification, sms alert, cyber cell and the respective actions.

3.2.5 Class Diagram

Base Class:

- User

Sub Classes:

- Input1
- Input2
- Prediction
- Detection
- Display1
- Display2

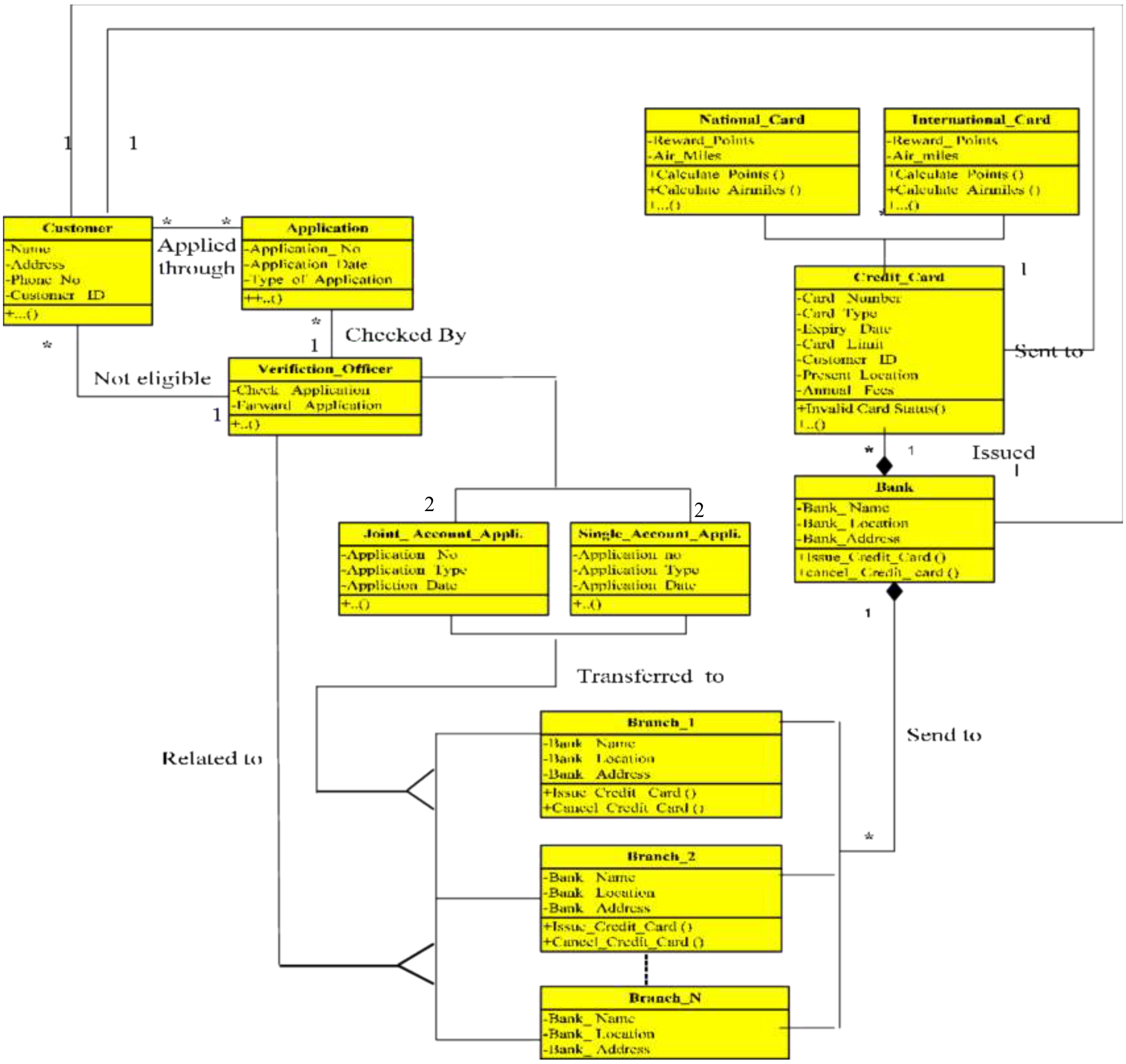


Fig 3.5. Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system.

It tends out to be an efficient way of modeling the interactions and collaborations in the external entities and the system. It models event-based systems to handle the state of an object. It also defines several distinct states of a component within the system. Each object/component has a specific state.

Here in this case it shows the interactions between various entities involved in the processing of the payment using credit card and what are their roles at respective steps when a transaction is beingmade.

3.3 Software Requirements Specification

3.3.1 Functional Requirements

- **Policy**: Should be able to implement the policy which is the core element of the RL as it alone can define the behavior of the agent. It should map the perceived states of the environment to the actions taken on those states.

- **Reward Signal:** The environment should be able to send an immediate signal to the learning agent at each state, and this signal is a reward signal. The agent's main objective is to maximize the total number of rewards for good actions.

- **Value Function:** The value functionality should be able to give information about how good the situation and action are and how much reward an agent can be expected. A value function should specify the good state and action for the future.

3.3.2 Non-Functional Requirements

- **Accuracy & Performance**: Learning algorithm accuracy and precision is important, and output is compared to the real true result.

- **Transparency**: It is often not clear how results are derived, causing issues in trust and transparency.
- **Reliability**: Further effort has to be put for reliability in the system, like looking at the reliability of individual ML predictions, focusing on reliability estimation.
- **Testability:** Systematic testing of the outcome of ML systems is necessary. Major focus should be given on applying ML systems to improve software testing strategies.

3.3.3 Domain Requirements

- The user must be able to input the required card details and obtain the output, to check the card is fraud or not.
- and must gather the data of all the fraud as well as non-fraudulent transactions throughout the month

Chapter 4

PROJECT PLANNING

As shown in figure 4.1 different phases involved in the project are mentioned along with their timeline.

- **Problem analysis**: We decided upon the problem to be solved for the project/ decideuponthe topic for the project.
- **Planning phase**: In this phase, we laid out a plan to complete various activities related to the project.
- **Project literature survey:** In this phase, we have surveyed different types of technical paper related to the problem which we have picked for our project.
- Study of DL and ML concepts: We tried to learn the deep learning concepts required for the project.
- **Collection of datasets**: We collected the ultrasound images datasets of Ovarian Images and numerical data from Kaggle.
- **System Design and architecture phase:** We design the architecture of the proposed system.
- **Implementation phase**: Here, we implement our proposed system using Python. After the implementation is completed, the next phase is to Test the system.
- **Testing:** Use a diverse set of test data that covers various scenarios, including legitimate transactions, known fraudulent patterns, and edge cases. This helps validate the system's ability to detect fraudulent activity accurately while minimizing false positives.
- **Documentation:** Provide an overview of the fraud detection system, including its purpose, objectives, and high-level architecture. Describe the components, data flows, and integration points with other systems. Document the sources of data used for fraud detection, such as transaction records, user profiles, and historical data. Explain the preprocessing steps involved in cleaning, transforming, and normalizing the data for analysis.
- The last phase is preparation of the final report, in this phase we provide a final and detailed report of our project.

4.1 Gantt Chart

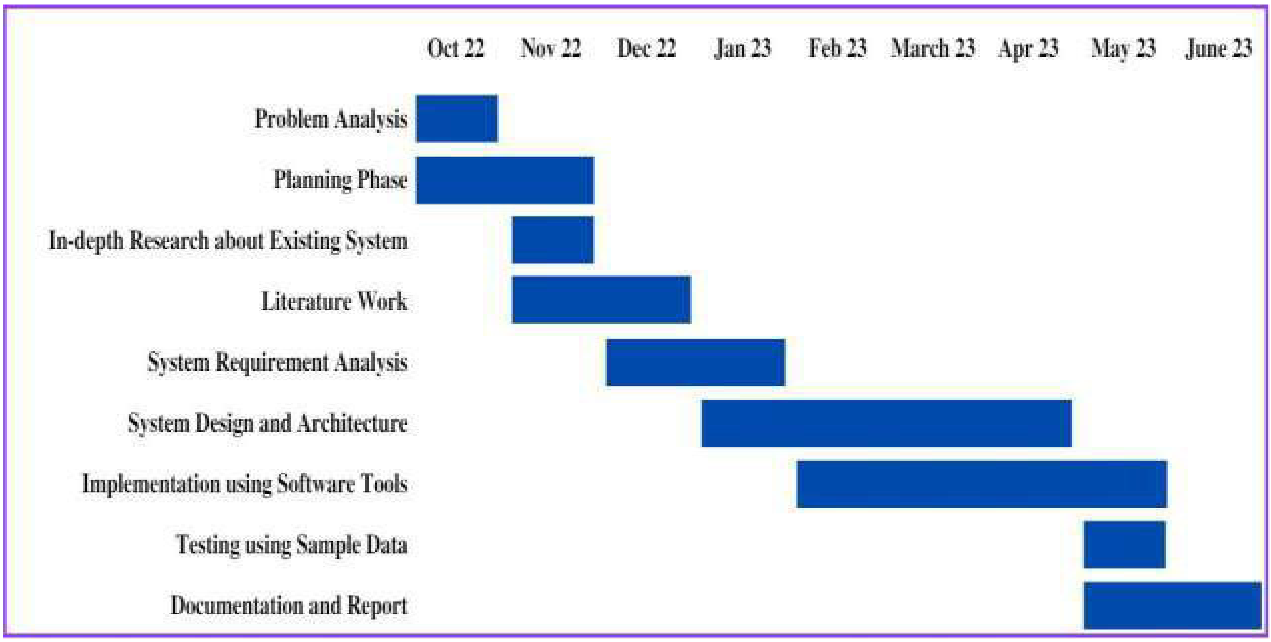


Fig 4.1 Gantt Chart

A Gantt chart is a type of bar chart that illustrates a project schedule. This chart lists the tasks to be performed on the vertical axis, and time intervals on the horizontal axis. The width of the horizontal bars in the graph shows the duration of each activity. In the fig 4.1, the Gantt chart of the current project is represented, where the planning stage started in the month of September 2022 and the is proposed to end by June 2023.

Chapter 5

SYSTEM DESIGN

5.1 System Architecture

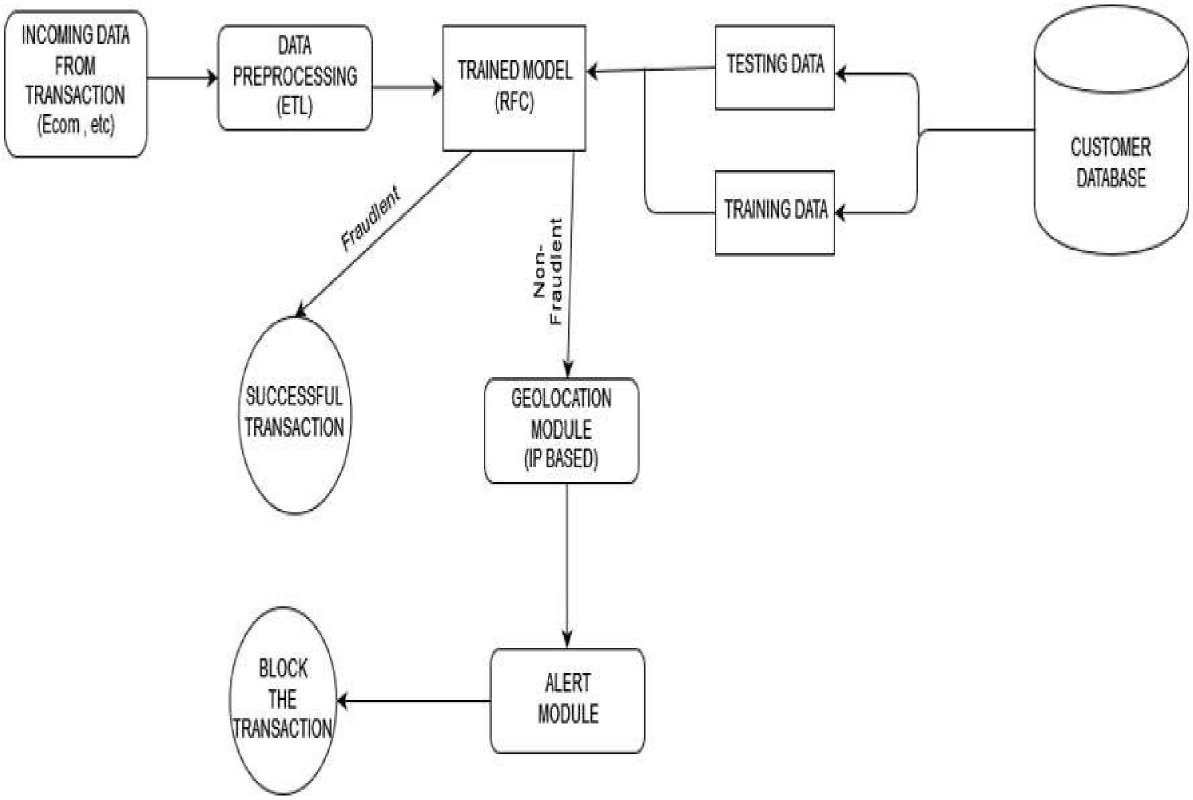


Fig.5.1 System Architecture of Credit Card Fraud Detection

This is the architecture of our system. It is the actual representation of the algorithm which we implemented. The first step is to read the data set and then it is sent for sampling. Training and testing of the data set is done.

After the feature selection, the data will be sent to the algorithm which is the Random Forest the a Classifier. The resultant data is stored in test sample data. The prediction of outcome is done based on test sample data & the result of the algorithm.

Later the performance & accuracy results are plotted. It has a feature which validates the results if the transaction is legitimate then the transaction is said to be true or else it is false. In case of false transactions the user is made aware of it.

5.2 Component Design / Module Decomposition

- **Data loading and preprocessing:** This component involves loading the dataset and preprocessing it to remove any missing values or outliers.
- **Data segmentation:** This component involves dividing the data into two segments - fraudulent and non-fraudulent transactions.
- **Exploratory data analysis:** This component involves analyzing the dataset to identify any patterns or correlations between the variables.
- **Model building:** This component involves building a Random Forest Classifier model to predict fraudulent transactions.
- **Model evaluation:** This component involves evaluating the performance of the model using metrics such as accuracy, precision.

5.3 Module description :

1. Data preprocessing:

Data pre-processing is used to remove the noisy data ,inconsistent data, remove the missing values.

The ETL steps included are

The Extract step involves gathering data from various sources

The Transform step involves cleaning and processing the data to ensure its quality and consistency.

Finally, in the Load step, the transformed and cleaned data is loaded into a data warehouse or database for analysis.

2. Training Module:

It consists of the sample output data and the corresponding sets of input data that have an influence on the output. Model validation is a set of processes and activities designed to ensure that an ML/AI model is performing as it should, including both its design objectives and its utility for the end user.

3. Geolocation Module:

An IP-based geolocation module is a component or service that uses the IP address of a device to determine its geographical location using IPv4 and locate the nearest Cybercell Station . Cybercell team can use geolocation data to detect and prevent fraudulent transactions.

4. Alert Module:

The messaging module can be integrated into a larger system or can be a standalone application. The messaging module can support various communication channels like SMS, MMS, email, push notifications, instant messaging, and social media messaging.

5.4 Interface Design

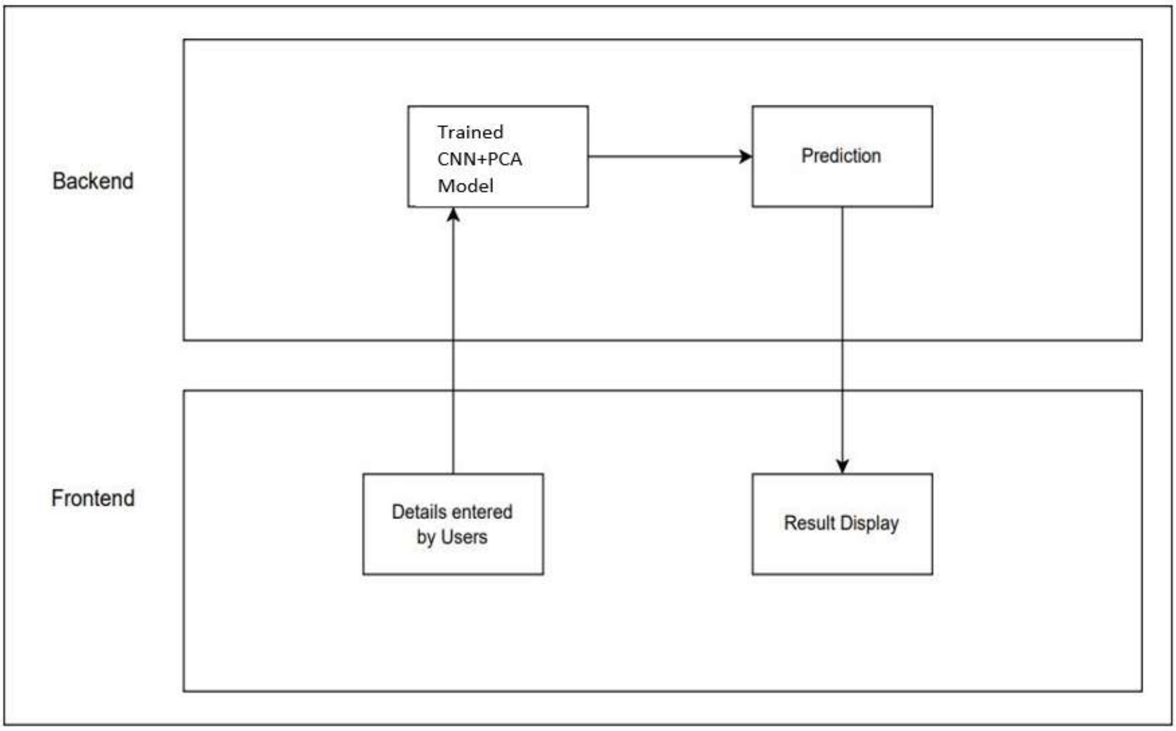


Fig.5.3 Interface Design of Credit Card Fraud Detection

User interface is the front-end application view to which user interacts in order to use the a software.

There are text boxes to enter user details .

There are buttons to predict the transactios and to reset the entered data.

5.5 Data Structure Design

1. Pandas DataFrame: The credit card dataset is loaded into a Pandas DataFrame, which is a two-dimensional table-like data structure with labeled rows and columns. It is used to manipulate and analyze the data, as well as perform operations such as selecting and dropping columns, splitting data into input features and output features, and filtering rows.
2. NumPy arrays: The input features and output feature of the credit card dataset are converted into NumPy arrays, which are homogeneous arrays of fixed size with efficient element-wise operations. They are used to store and manipulate numerical data and are a popular data structure in machine learning.
3. StandardScaler: A StandardScaler object is used to scale the input features to have zero mean and unit variance. It is applied to the NumPy arrays of the training and testing sets and is a useful data structure for preprocessing data before applying machine learning models.
4. Lists: Lists are used to specify the column names to load from the CSV file and to define the input data point to make predictions on.
5. Variables: Variables are used to store the various objects and data structures, such as the X_train, y_train, and model variables.

5.6 Algorithm Design

Collect the dataset: Gather the credit card data you will use for training and testing your model.
Prepare the data: Preprocess the data, clean it, and transform it into a format that can be used a by the algorithm.
Split the data: Split the data into a training set and a testing set. The training set will be used to train the algorithm, and the testing set will be used to evaluate its performance.
Train the model: Fit the Random Forest Classifier to the training data.
Make predictions: Use the trained model to make predictions on the testing data .

Chapter 6

IMPLEMENTATION

6.1 Implementation Approaches

- First, the necessary packages are imported, including the RandomForestClassifier from sklearn.ensemble.
- The credit card transaction data is loaded from a CSV file using Pandas, and a list of columns to use for analysis is created.
- The data is then subset using the list of columns, and its shape and descriptive statistics are printed.
- The fraction of fraud cases in the dataset is calculated and printed, along with the number of fraudulent and valid transactions.
- The amount details of the fraudulent and valid transactions are printed.
- A correlation matrix is created using the data, and it is displayed as a heatmap using Seaborn.
- The data is divided into predictors (X) and target (Y), and then converted to Numpy arrays.
- The data is then split into training and testing sets using the train_test_split function from sklearn.model_selection.
- A Random Forest Classifier model is created using RandomForestClassifier() from sklearn.ensemble, and then fit to the training data using fit() method.
- The model is used to make predictions on the testing data using predict() method.
- The performance of the model is then evaluated using various metrics including accuracy, precision, recall, F1-score, and Matthews correlation coefficient. This is done using methods from sklearn.metrics, such as accuracy_score(), precision_score(), recall_score(), f1_score(), and matthews_corrcoef(). The results of each metric are printed to the console.

6.2 Coding Details and Code Efficiency

- **Import the necessary packages –**

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import gridspec
```

- **Loading the datasets using Pandas –**

```
data = pd.read_csv("credit.csv")
```

- **Describing the data –**

```
print(data.shape)
print(data.describe())
```

- **Imbalance in the data –**

```
fraud = data[data['Class'] == 1]
valid = data[data['Class'] == 0]
outlierFraction = len(fraud)/float(len(valid))
print(outlierFraction)
print('Fraud Cases: {}'.format(len(data[data['Class'] == 1])))
print('Valid Transactions: {}'.format(len(data[data['Class'] == 0])))
```

- **Print the amount details for Fraudlent transactions –**

```
print(“Amount details of the fraudulent transaction”)
fraud.Amount.describe()
```

- **Print the amount details for Normal transactions –**

```
print(“details of valid transaction”)
valid.Amount.describe()
```

- **Plotting the correlation matrix –**

```
corrmat = data.corr()
fig = plt.figure(figsize = (12, 9))
sns.heatmap(corrmat, vmax = .8, square = True)
plt.show()
```

- **Separating the X and Y values –**

```
X = data.drop(['Class'], axis = 1)
Y = data["Class"]
print(X.shape)
print(Y.shape)
xData = X.values
yData = Y.values
```

- **Training and Testing Data Bifurcation –**

```
from sklearn.model_selection import train_test_split
xTrain, xTest, yTrain, yTest = train_test_split(xData, yData, test_size = 0.2, random_state = 42)
```

- **Building a Random Forest model using scikit learn –**

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()
rfc.fit(xTrain, yTrain)
yPred = rfc.predict(xTest)
```

- **Building all types of evaluating parameters –**

```
from sklearn.metrics import classification_report, accuracy_score
from sklearn.metrics import precision_score, recall_score
from sklearn.metrics import f1_score, matthews_corrcoef
from sklearn.metrics import confusion_matrix

n_outliers = len(fraud)
n_errors = (yPred != yTest).sum()
print("The model used is Random Forest classifier")

acc = accuracy_score(yTest, yPred)
print("The accuracy is {}".format(acc))

prec = precision_score(yTest, yPred)
print("The precision is {}".format(prec))
```

```
rec = recall_score(yTest, yPred)
print("The recall is {}".format(rec))

f1 = f1_score(yTest, yPred)
print("The F1-Score is {}".format(f1))

MCC = matthews_corrcoef(yTest, yPred)
print("The Matthews correlation coefficient is{}".format(MCC))
```

- **Get user input for new data –**

```
import json
with open('C:\\Users\\shivu\\Downloads\\data.json') as f:
data = json.load(f)
time=1234
cvv=(data['cvv'])
ccno=(data['ccno'])
expiry=(data['expiry'])
category=(data['category'])
amount=(data['Amount'])
 user_input = [[time, cvv, ccno, expiry,category,amount]]
```

- **Create a pandas dataframe from the user input –**

```
new_data = pd.DataFrame(user_input, columns=listofcol[:-1])
prediction = rfc.predict(new_data)
print(prediction)
```

- **SMS alert feature -**

```
import requests
ip_address =  requests.get('https://api.ipify.org').text
response = requests.get(f'https://ipinfo.io/{ip_address}/json')
data = response.json()
IP_Address= data['ip']
Location= data['city']
Latitude= data['loc'].split(',')[0]
Longitude= data['loc'].split(',')[1]
```

```
 if(prediction):

from twilio.rest import Client

account_sid = 'AC4effaaefcad352a976272a2e8e74f4a3'
auth_token = '28669e5fe9f3002c4796a7aa2b958ca2'
client = Client(account_sid, auth_token)

message = client.messages \ .create(body="hello shivu,ip address of hacker is:" +
                IP_Address+" location:" + Location + " latitude and longitude: "+
                Latitude+ " " + Longitude, from_='+16205434655',  to='+919945471308')
print(message.sid)
```

- **Calling Feature –**

```
account_sid = 'AC4effaaefcad352a976272a2e8e74f4a3'
auth_token = '28669e5fe9f3002c4796a7aa2b958ca2'
client = Client(account_sid, auth_token)

call = client.calls.create( url='http://demo.twilio.com/docs/voice.xml' to='+919945471308',
          from_='+16205434655'  )
print(call.sid)
}

else:
from flask import Flask, render_template
        app = Flask(_name_)
@app.route('/')
def index():
   message = 'Hello, world!'
   return render_template('index1.html', message=message)
 if _name_ == '_main_':
   app.run(debug=True)
```

- **Check CC Number HTML page –**

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Document</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css"
rel="stylesheet" crossorigin="anonymous">
</head>

<body>
   <form style="width:50vh; margin: auto; position:relative; text-align: center;top: 30vh;border:2px
solid grey;padding:2rem">
      <span class="input-group-text" id="basic-addon1">Credit Card Number :</span>
        <input  id="inputName" name="name"   type="text" class="form-control" placeholder="Enter the
credit card number" aria-label="Enter the credit card number" aria-describedby="basic-addon1"><br>

        <input type="submit" class="btn btn-success" value="submit">
      </form>
      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.min.js"
integrity="sha384-
Y4oOpwW3duJdCWv5ly8SCFYWqFDsfob/3GkgExXKV4idmbt98QcxXYs9UoXAB7BZ"
crossorigin="anonymous"></script>
   <script>
var ccNum=''

// Get the form element
var form = document.querySelector('form');

// Add an event listener to the form's submit button
form.addEventListener('submit', function(event) {
  // Prevent the form from submitting
  event.preventDefault();

// Get the values of the input fields
var ccnumber = document.getElementById('inputName').value;
```

```
  ccNum = ccnumber;

  if (isValidCreditCardNumber(ccNum)) {
  alert('Valid credit card number!');
} else {
  alert('Invalid credit card number!');
}

  // Do something with the values
  console.log('Name:', ccnumber);


});
      function isValidCreditCardNumber(ccNum) {

  // Remove any whitespace from the input string
  ccNum = ccNum.replace(/\s/g, '');

  // Check if the input is a string of 16 digits


  if (!/^\d{16}$/.test(ccNum)) {
    return false;
  }
  // Apply the Luhn algorithm to the input string
  var sum = 0;
  var digit;
  var even = false;

  for (var i = ccNum.length - 1; i >= 0; i--) {
    digit = parseInt(ccNum.charAt(i), 10);
    if (even) {
      digit *= 2;
      if (digit > 9) {
        digit -= 9;
      }
    }
    sum += digit;
    even = !even;
  }

  // If the sum is divisible by 10, the input string is a valid credit card number
  return (sum % 10 === 0);
}
    </script>
</body>
</html>
```

- **Home HTML page**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="style.css">
  <title>Credit Card Fraud Detection</title>
      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css"
   rel="stylesheet" integrity="sha384-
   KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJOZ"
   crossorigin="anonymous">

</head>
<body>
    <!-- <div id="main_register_page">
        <div id="signup">
          <h2 style="padding: 1rem">Your Details Please</h2>
          <div className="input-group mb-3">
            <span className="input-group-text" id="basic-addon1">
              <i className="fas fa-user-lock" />
            </span>
            <input
              type="text"
              className="form-control"
              placeholder="Username"
              aria-label="Username"
              aria-describedby="basic-addon1"

            />
          </div>
          <div className="input-group mb-3">
            <span className="input-group-text" id="basic-addon1">
  <i className="fas fa-user-lock" />
  </span>
  <input
    type="text"
    className="form-control"
    placeholder="Username"
    aria-label="Username"
    aria-describedby="basic-addon  />
</div>
<div className="input-group mb-3">
  <span className="input-group-text" id="basic-addon1">
    <i className="fas fa-user-lock" />
  </span>
  <input
    type="text"
    className="form-control"
    placeholder="Username"
    aria-label="Username"
    aria-describedby="basic-addon1"  />
</div>
<div className="input-group mb-3">
  <span className="input-group-text" id="basic-addon1">
    <i className="fas fa-user-lock" />
  </span>
  <input
    type="text"
    className="form-control"
    placeholder="Username"
    aria-label="Username"
    aria-describedby="basic-addon1" />
</div>
<div className="input-group mb-3">
  <span className="input-group-text" id="basic-addon1">
    <i className="fas fa-user-lock" />
  </span>
  <input
    type="text"
    className="form-control"
    placeholder="Username"
    aria-label="Username"
```

```
          aria-describedby="basic-addon1"  />
        </div>
        <div className="input-group mb-3">
          <span className="input-group-text" id="basic-addon1">
            <i className="fas fa-user-lock" />
          </span>
          <input
            type="text"
            className="form-control"
            placeholder="Username"
            aria-label="Username"
            aria-describedby="basic-addon1"  />
        </div>
        <button id="contact_button" >
          <i className="far fa-paper-plane"/> Register
        </button>
      </div>
    </div>   -->
    <!-- <form id="myForm">

      <label for="inputField">Enter a message:</label>
<input type="text" id="inputField">

<label for="inputField">Enter a message:</label>
<input type="text" id="inputField">

<label for="inputField">Enter a message:</label>
<input type="text" id="inputField">

<label for="inputField">Enter a message:</label>
<input type="text" id="inputField">

<label for="inputField">Enter a message:</label>
<input type="text" id="inputField">
<button type="submit">Render Message</button>
</form> -->

<form  id="myForm" method="post" action="/add" style="width:50vh;margin: auto;position:relative;text-align: center;top: 10vh;border:2px solid grey;padding:2rem;display: block;">
  <!-- <label for="inputField">Enter a cvv:</label>
  <input type="number" id="inputField1" name="message1"> -->
  <span class="input-group-text" id="basic-addon1">Enter a CVV:</span>
      <input  type="number" id="inputField1" name="message1" class="form-control" placeholder="Enter the credit card number" aria-label="Enter the credit card number" aria-describedby="basic-addon1"><br>

  <!-- <label for="inputField">Enter a ccno:</label>

  <input type="number" id="inputField2" name="message2"> -->
  <span class="input-group-text" id="basic-addon1">Enter a CCno:</span>
      <input  type="number" id="inputField2" name="message2" class="form-control" placeholder="Enter the credit card number" aria-label="Enter the credit card number" aria-describedby="basic-addon1"><br>

  <!-- <label for="inputField">Enter a expiry:</label>

  <input type="text" id="inputField3" name="message3"> -->
  <span class="input-group-text" id="basic-addon1">Enter a expiry:</span>
  <input  type="text" id="inputField3" name="message3" class="form-control" placeholder="Enter the credit card number" aria-label="Enter the credit card number" aria-describedby="basic-addon1"><br>
<!--
  <label for="inputField">Enter a cateogory:</label>

  <input type="text" id="inputField4" name="message4"> -->
  <span class="input-group-text" id="basic-addon1">Enter a cateogory (1):</span>
  <input type="text" id="inputField4" name="message4" class="form-control" placeholder="Enter the credit card number" aria-label="Enter the credit card number" aria-describedby="basic-addon1"><br>
<!--
  <label for="inputField">Enter a Amount:</label>

  <input type="number" id="inputField5" name="message5"> -->
  <span class="input-group-text" id="basic-addon1">Enter a Amount:</span>
  <input type="number" id="inputField5" name="message5" class="form-control" placeholder="Enter the credit card number" aria-label="Enter the credit card number" aria-describedby="basic-addon1"><br>

  <button type="submit" class="btn btn-success" >Generate json</button>
  <div id="output"></div>
</form>
<h2></h2>
<!-- <h2>{{message1}}</h2> -->

<a href=""></a>
```

```
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.7/dist/umd/popper.min.js"
integrity="sha384-zYPOMqeu1DAVkHiLqWBUTcbYfZ8osu1Nd6Z89ify25QV2/QExE"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.min.js"
integrity="sha384-oOpwW3duJdCWv5ly8SCFYWqFDsfob/3GkgExXKV4idmAB7BZ"
crossorigin="anonymous"></script>
 <script src="./src.js" >
  </script>
</body>
</html>
```

- **CSS page code**

```
#header{
  width: 100%;
  height: 100vh;
  background-image: url('./images/fraud.png');
  background-position: center;
  background-size: cover;
}
.btn2
{
    display: block;
    margin: 20rem auto;
    width: fit-content;
    border: 1px solid #ff004f;
    padding:14px 50px ;
    border-radius: 6px;
    text-decoration: none;
    color: blueviolet;
    transition:background 0.5s ;
    position:absolute ;
    top: 0;

}
```

```
.btn1
{
     display: block;
    margin: 20rem auto;
    width: fit-content;
    border: 1px solid #ff004f;
    padding:14px 50px ;
    border-radius: 6px;
    text-decoration: none;
    color: blueviolet;
    transition:background 0.5s ;
    position:absolute ;
    top:10%;
    right: 5%;
}
.App {
    text-align: center;
  }
    .App-logo {
    height: 40vmin;
    pointer-events: none;
  }
    @media (prefers-reduced-motion: no-preference) {
    .App-logo {
      animation: App-logo-spin infinite 20s linear;
  }
  .App-header {
    background-color: #282c34;
    min-height: 100vh;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    font-size: calc(10px + 2vmin);
    }
```

```
.App-link {
  color: #61dafb;
}

@keyframes App-logo-spin {
  from {
    transform: rotate(0deg);
  }
  to {
    transform: rotate(360deg);
  }
}

/* *********navbar******** */
.navbar-nav
{
  margin: auto;
  width: 100%;
  display:  flex;
  justify-content:flex-end;
}
.nav-item
{
  margin-right: 7rem;



}

*{
  margin:0;
  padding: 0;
  box-sizing: border-box;
  /* overflow: hidden; */
}
/* *****home************* */
```

```
.bg1
{
  width: 50%;
  height: 100%;
  background-color: rgba(0,0,255, 0.09);

}
.bg2
{
  width: 50%;
  height: 100%;

}
 .main_body
{
  /* border: 2px solid red; */
  height:100vh;
  position: relative;
  /* top: -15rem; */
  /* z-index: -1; */

}
.item{
  font-family: 'Poppins', sans-serif;
font-size:xx-large;
  text-align: center;
  position:sticky;
  top: 50%;
  left: 30vw;
  width: 40%;
  /* margin-left: auto; */
}

/* ***************contact***************** */
#header
{
  display: grid;
  grid-template-rows: 1fr ;
  grid-template-columns:repeat(auto-fit,minmax(19rem,1fr));
  row-gap: 12px;

  column-gap: 12px;

}
#header div
{

  background-color: rgba(243, 245, 247, 0.79);

  padding: 1rem;

  margin: 1rem;

  box-shadow: 6px 2px 9px grey;

}
#header div  input
{
  border-radius: 4px;
text-align: center;
  width: 80%;
}

#signup
{
  background-color: rgba(243, 245, 247, 0.79);
  box-shadow: 6px 2px 9px grey;
padding: 1rem;
  width: 60%;
  margin: auto;
  margin-top: 7rem;

  margin-bottom: 7rem;
```

```
}
#signup input
{
  width: 50%;
}

#main_contact_page
{
  overflow:scroll;

  background-color: rgba(0,0,255, 0.09);
  height: 100vh;
}
#contact_button
{
  background-color: green;
  width: 20%;
  border-radius: 3px;
  margin: 0.5rem;

  padding: 0.5rem;
  color: whitesmoke;

}
.butt
{
  position: relative;
  top: -6rem;
  width: 10%;
left:19rem
}
#contact_button:hover
{
  opacity: 0.6;
}
#flex_name
{
  /* margin: 0; */
  display: flex;
  justify-content: space-between;
}

/* *********register*********** */
#main_register_page
{
overflow-y: scroll;
  background-color: rgba(0,0,255, 0.09);
  height: 100vh;
}

input{
  border: none;
}


/* ***************login************ */

#login
{
  background-color: rgba(0,0,255, 0.09);
  height: 100vh;
  text-align: center;
}
#login input
{
  box-shadow: 4px 2px 9px grey;
   border-radius: 2px;
  border:1px solid black;
  margin: .7rem;
  text-align: center;

}
#login i{
  /* background-color: gray; */
  width: max-content;
}
.wrapper
{
  box-shadow: 6px 2px 9px grey;

  background-color: white;
  display: flex;
  width: 60%;
  margin: auto;
margin-top: 7rem;

}
.left
{
  margin:auto;
  margin-top: 7rem;
  display: flex;
  flex-direction: column;
  min-width:15rem;
}
form{
  display: flex;
  flex-direction: column;
}

#login_button
{
  background-color: green;
  color: whitesmoke;
  width: 30%;
}
#login_button:hover
{
  opacity: 0.6;
}
```

```
 /* *******thank you************** */
 .item2
 {
   text-align: center;
   display: flex;
   flex-direction: column;
 }
 .item2 Link ,a{

   /* border: 2px solid red; */
   /* position: relative; */
   /* z-index: 50; */
   text-decoration: none;
 }
 /* .item2 Link:hover,.item2 a:hover{

   text-decoration:underline;
   cursor: pointer;
 }  */

 /* //////////////////
  */
```

- **Index Page**

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <link rel="stylesheet" href="./style.css">

   <title>Home</title>
</head>
<body  >
   <div style="position: relative;height: 100vh;width: 100vw;">
      <div id="header" >
```

```
</div>
<div>
 <!-- <button style="position:relative ;top: 50%;right: 100%;color: blue;" ><a
href="/index.html">Click</a></button> -->
            <button type="button" class="btn1"><a href="./home.html">Click</a></button>
            <button type="button" class="btn2"><a href="./CheckCC.html">Check CC</a></button>

 </div>
 </div>
</body>
</html>
```

- **JavaScript Code**

```
const form = document.getElementById("myForm");
const output = document.getElementById("output");

form.addEventListener("submit", function(event) {
  event.preventDefault(); // prevent the form from submitting

  const input1 = form.elements["message1"].value;
  const input2 = form.elements["message2"].value;
  const input3 = form.elements["message3"].value;
  const input4 = form.elements["message4"].value;
  const input5 = form.elements["message5"].value;

  const data = {
 cvv:input1,
  ccno:input2,
  expiry:input3,
  category:input4,
  Amount:input5
};

const jsonString = JSON.stringify(data);
console.log(jsonString)

const jsonObject = JSON.parse(jsonString);
const blob = new Blob([JSON.stringify(jsonObject)], {type: "application/json"});
const url = URL.createObjectURL(blob);
const link = document.createElement("a");
link.href = url;
link.download = "data.json";
link.click();


});
```

Chapter 7

TESTING

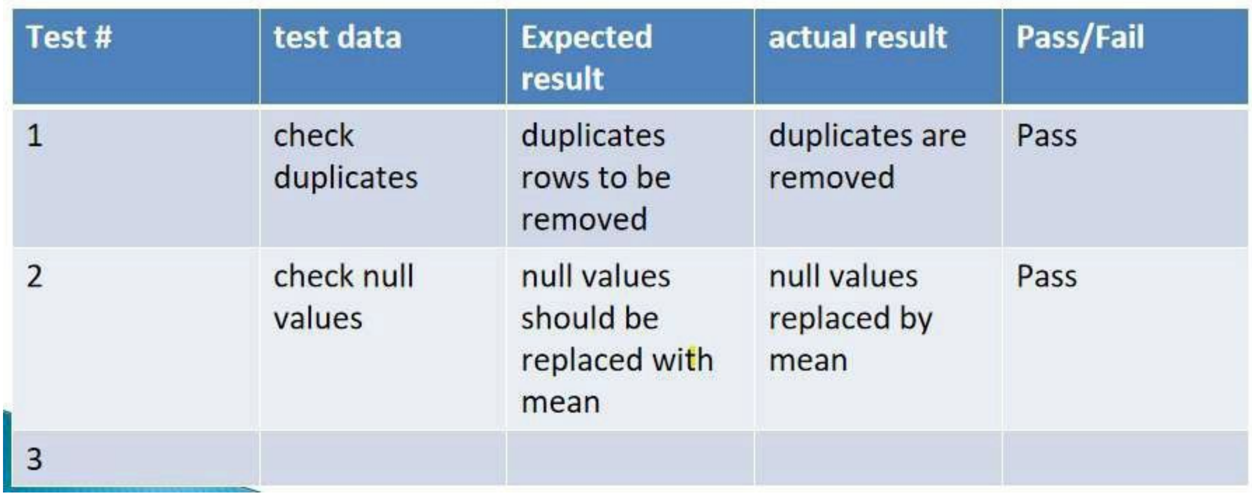
7.1 Unit Testing

Data Load:

Preprocessing

Test case 1:

preprocessing( ): Duplicate values are dropped and null values are replaced by mean.



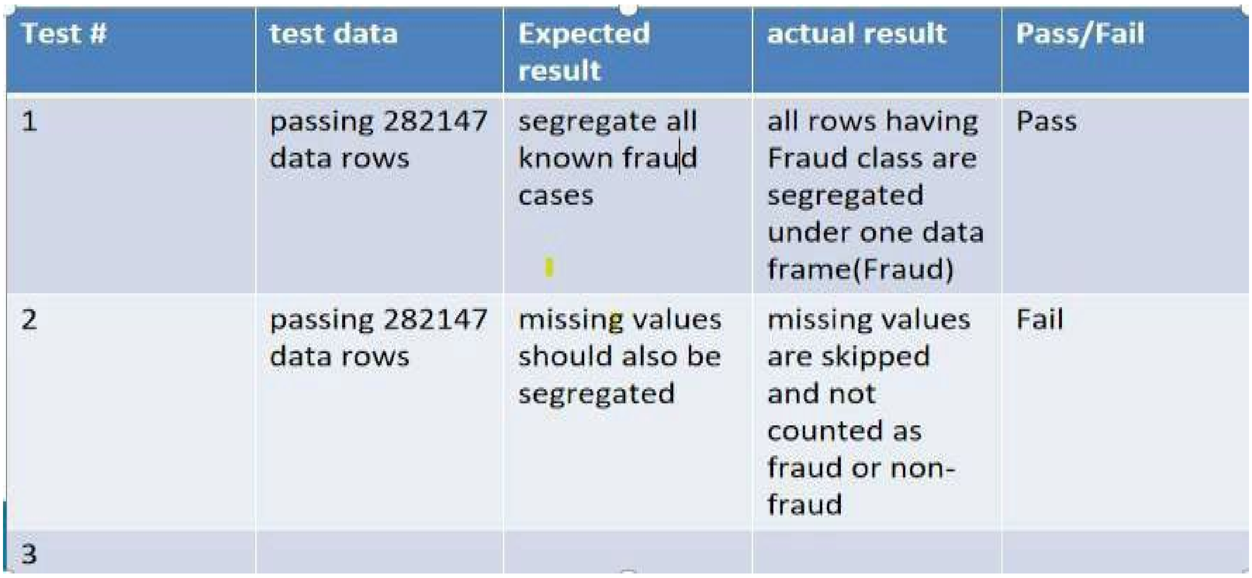
Test #	test data	Expected result	actual result	Pass/Fail
1	check duplicates	duplicates rows to be removed	duplicates are removed	Pass
2	check null values	null values should be replaced with mean	null values replaced by mean	Pass
3				

Segregation:

Data Segregation

Test case 2:

segregation( ): 282147 data values are segregated into fraud and non fraud.



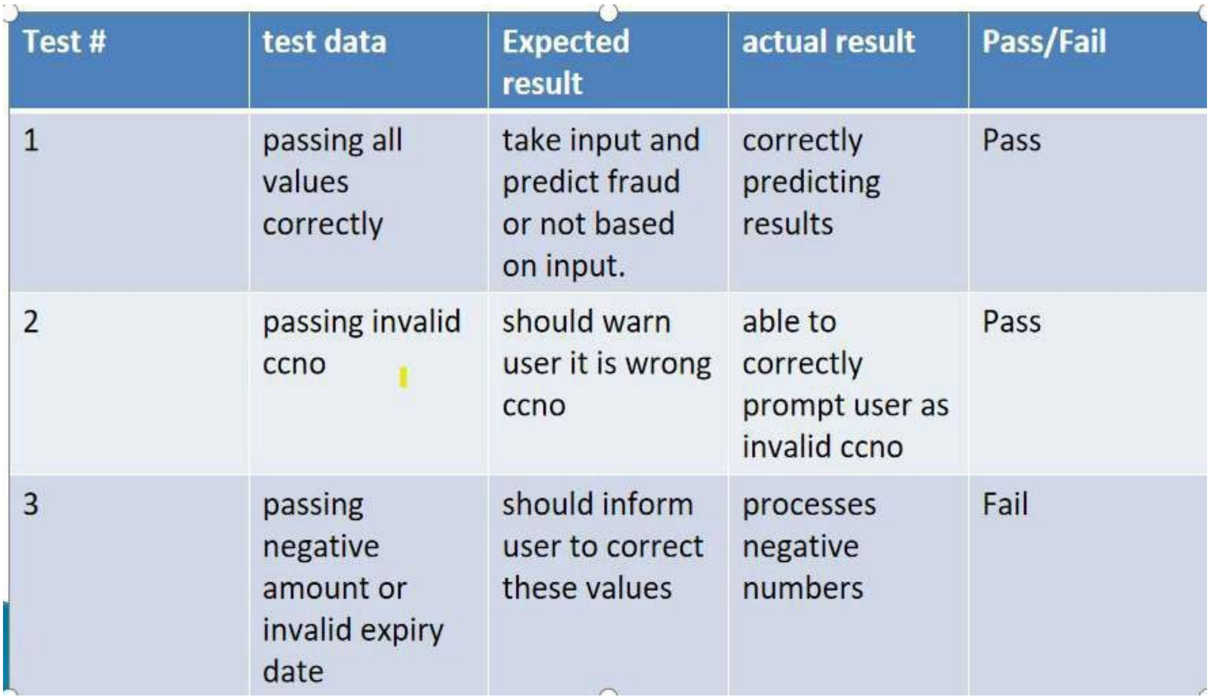
Test #	test data	Expected result	actual result	Pass/Fail
1	passing 282147 data rows	segregate all known fraud cases	all rows having Fraud class are segregated under one data frame(Fraud)	Pass
2	passing 282147 data rows	missing values should also be segregated	missing values are skipped and not counted as fraud or non-fraud	Fail
3				

Getting input from user:

Data input

Test case 3:

inputTaking( ): cvv,ccno,category,amount,expiry taking values.



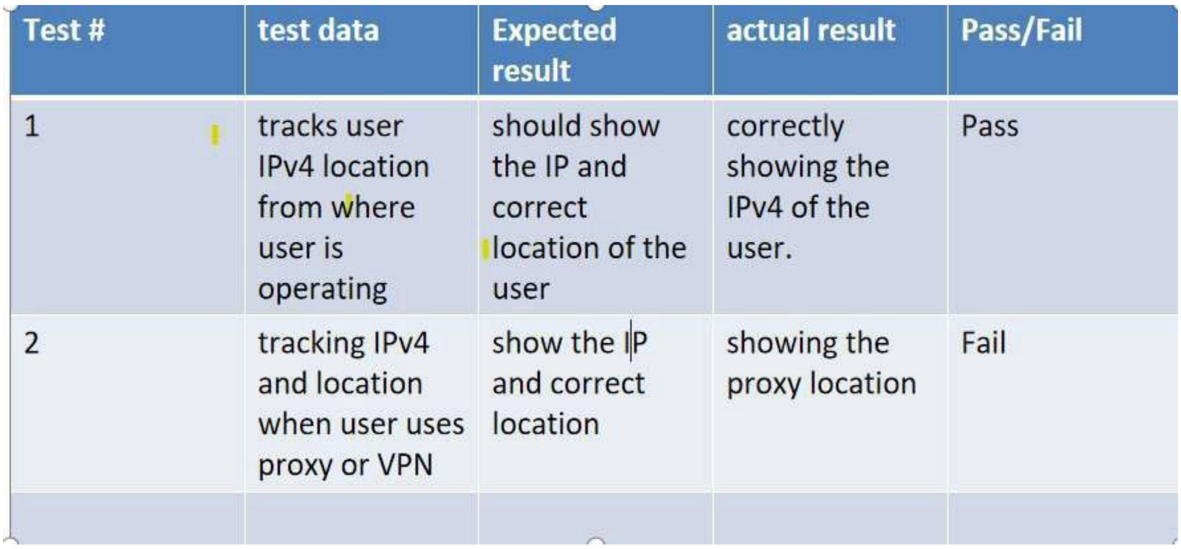
Test #	test data	Expected result	actual result	Pass/Fail
1	passing all values correctly	take input and predict fraud or not based on input.	correctly predicting results	Pass
2	passing invalid ccno	should warn user it is wrong ccno	able to correctly prompt user as invalid ccno	Pass
3	passing negative amount or invalid expiry date	should inform user to correct these values	processes negative numbers	Fail

Getting IP address:

IP tracker

Test case 4:

IP tracking( ): tracks the users ip location (reqiures internet for demo).



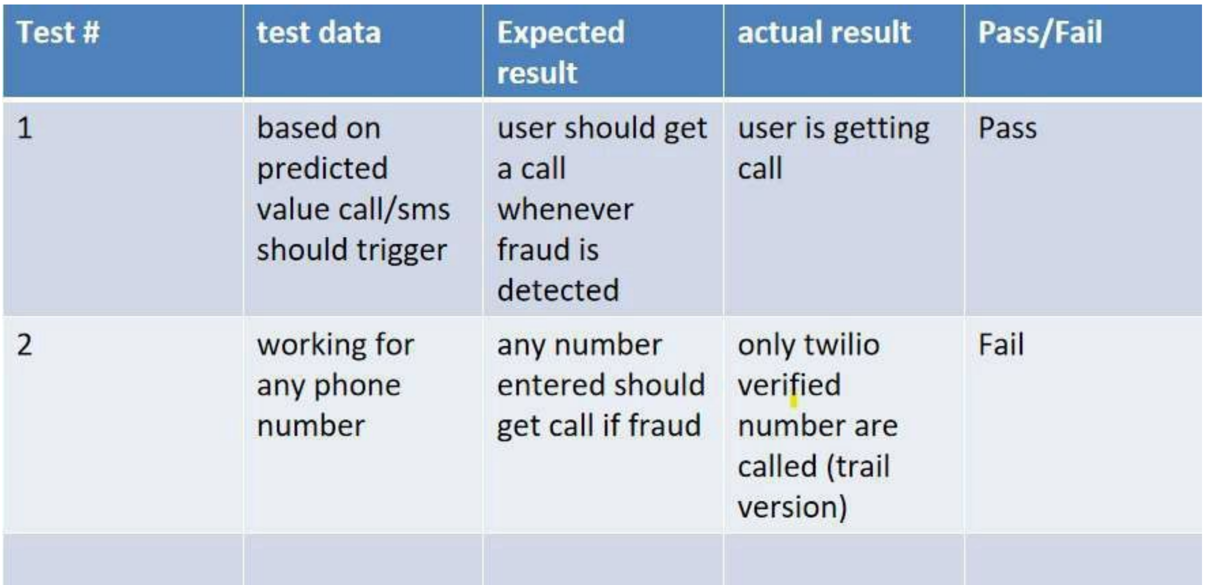
Test #	test data	Expected result	actual result	Pass/Fail
1	tracks user IPv4 location from where user is operating	should show the IP and correct location of the user	correctly showing the IPv4 of the user.	Pass
2	tracking IPv4 and location when user uses proxy or VPN	show the IP and correct location	showing the proxy location	Fail

Triggering call and SMS:

Alert Component

Test case 5:

Alert( ): alerts the user if fraud is detected(reqiures internet for demo).

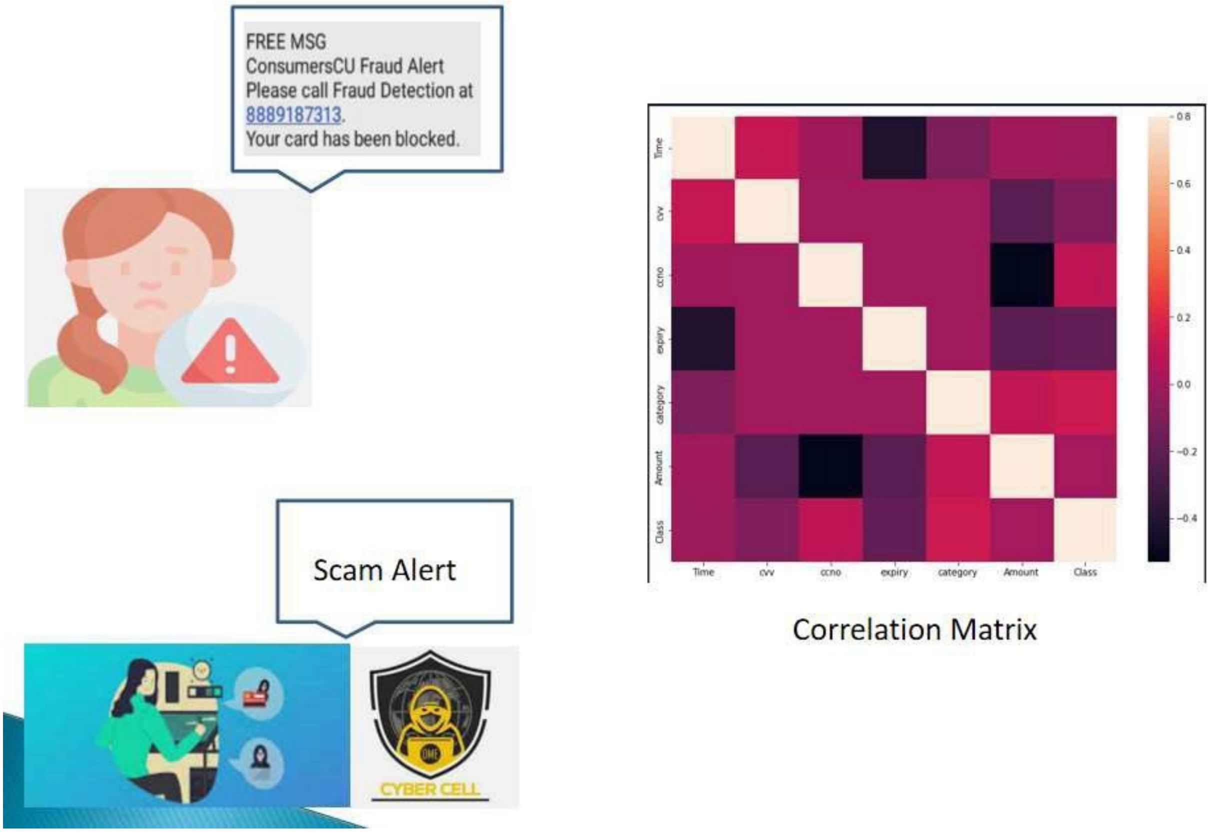


Test #	test data	Expected result	actual result	Pass/Fail
1	based on predicted value call/sms should trigger	user should get a call whenever fraud is detected	user is getting call	Pass
2	working for any phone number	any number entered should get call if fraud	only twilio verified number are called (trail version)	Fail

Chapter 8

RESULTS DISCUSSION AND PERFORMANCE ANALYSIS

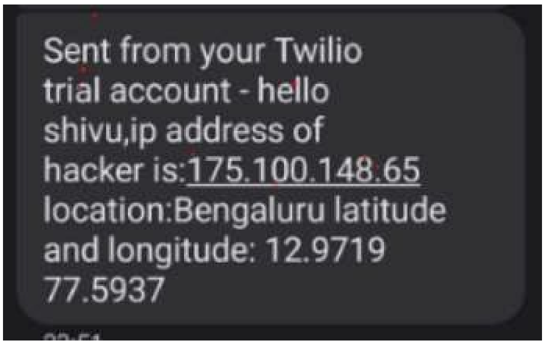
8.1 Results



FREE MSG
ConsumersCU Fraud Alert
Please call Fraud Detection at 8889187313.
Your card has been blocked.

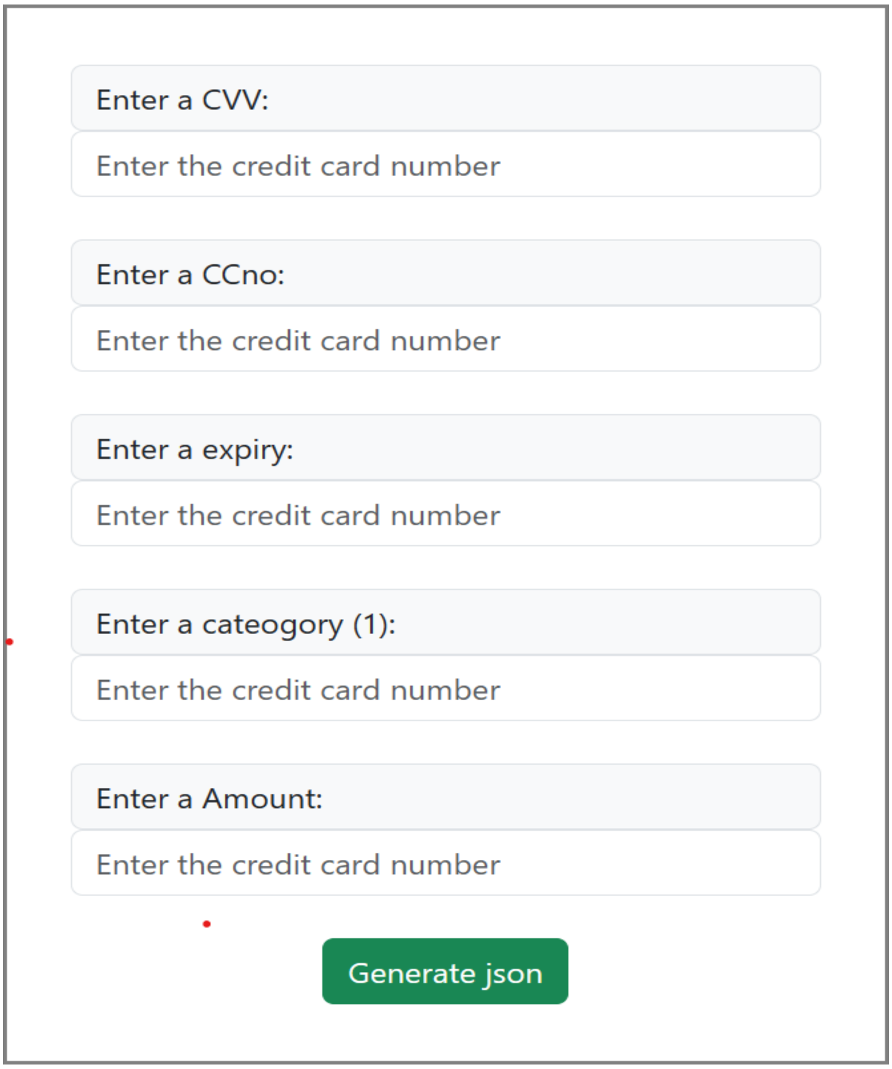
Fig.8.1a Illustration of correlation matrix and alert message

The message that was sent to the customer after detecting the fraud detection



Sent from your Twilio trial account - hello shivu,ip address of hacker is:175.100.148.65 location:Bengaluru latitude and longitude: 12.9719 77.5937

Fig.8.1b Actual message sent to the customer



Enter a CVV:

Enter the credit card number

Enter a CCno:

Enter the credit card number

Enter a expiry:

Enter the credit card number

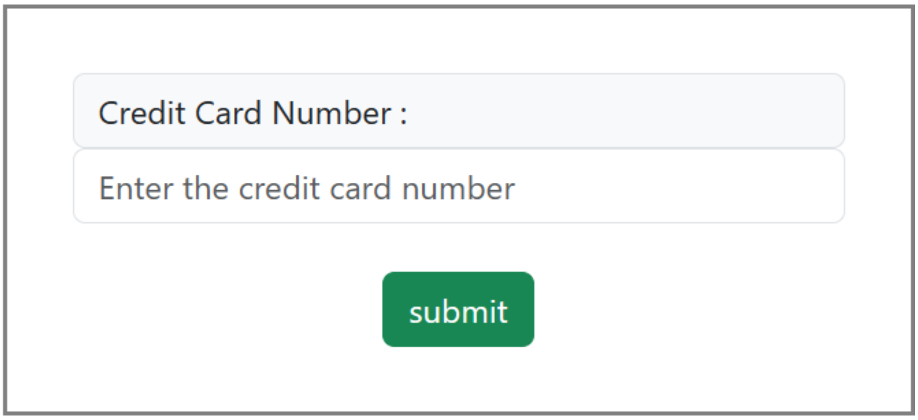
Enter a cateogory (1):

Enter the credit card number

Enter a Amount:

Enter the credit card number

Fig.8.2a Input form for Credit card details



Credit Card Number :

Enter the credit card number

Fig.8.2b Verification of credit card number

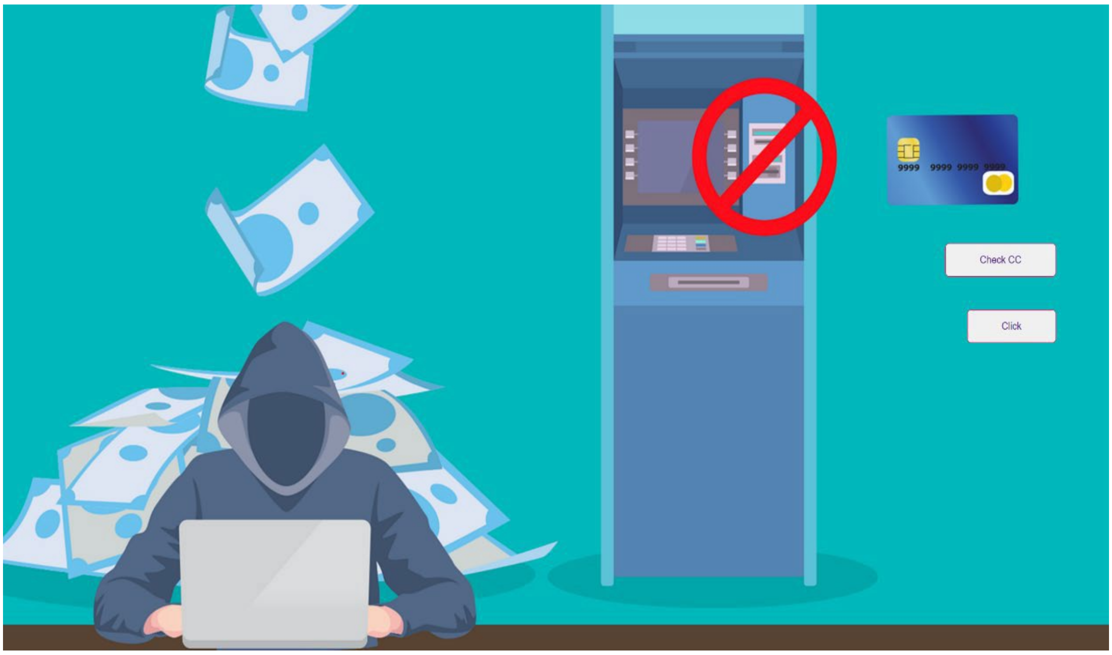
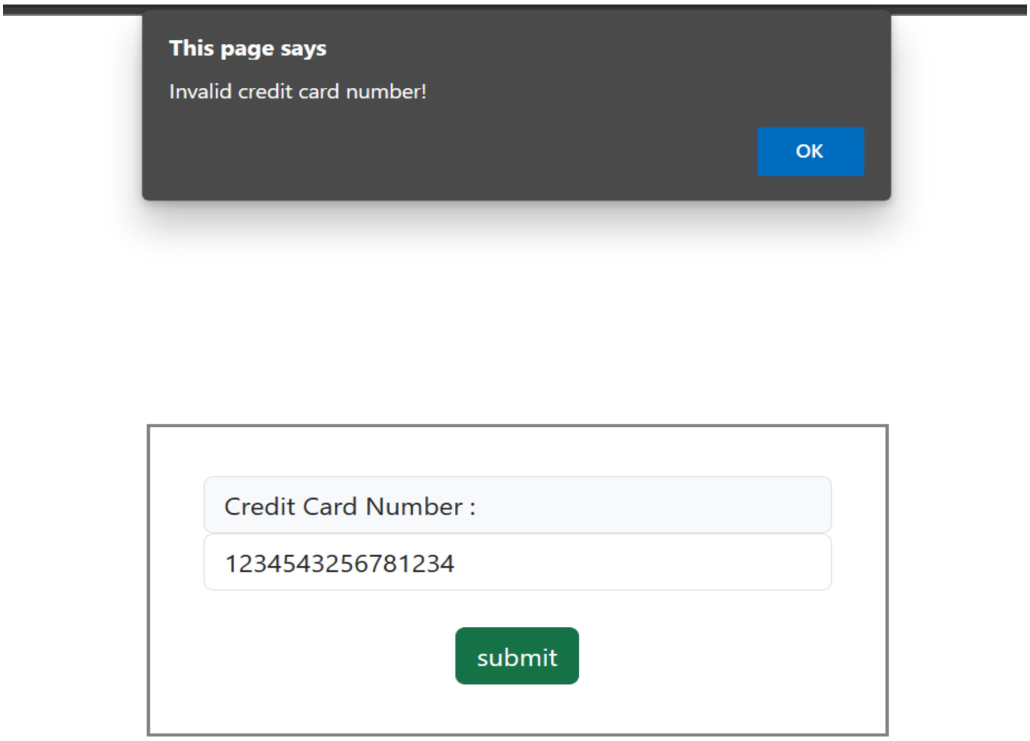


Fig.8.3 Home page of credit card fraud detection



This page says

Invalid credit card number!

Credit Card Number :

1234543256781234

Fig.8.4 Checking for valid credit card number

Chapter 9

Application and Conclusion

9.1 Appliction

1. **Reducing financial losses:** By promptly identifying and preventing fraudulent transactions, credit card fraud detection systems help minimize financial losses for both credit card companies and customers, limiting the impact of unauthorized purchases.

2. **Enhancing customer trust**: The implementation of robust fraud detection systems assures customers that their credit card information is being protected. This boosts their confidence in using credit cards for online and offline transactions, fostering trust and loyalty.

3. **Improving business reputation:** By effectively combating credit card fraud, businesses can safeguard their reputation. This helps maintain customer trust, leading to increased sales and sustained loyalty.

4. **Reducing manual labor:** Automated fraud detection systems streamline the identification of fraudulent transactions, minimizing the need for manual review and intervention. This saves time, effort, and reduces the potential for human error.

5. **Compliance with regulations:** Credit card fraud detection systems assist businesses in adhering to regulatory requirements designed to protect customer data and prevent fraud. Compliance with these regulations ensures businesses avoid penalties and legal consequences.

9.2 Conclusion

- In this project, Machine Learning techniques like Logistic Regression, Decision Tree and Random Forest were used to detect the fraud in credit card system.
- Sensitivity, Specificity, Accuracy and Error rate are used to evaluate the performance for the proposed system.
- There is a very strong possibility of the system being adopted as a norm for the major banking and financial services applications as fraud detection and prevention is the major checkpoint in financial and banking sector
- The above system is also likely to be embedded in other applications based, modified as per platform-specific/application specific environment.

institution.

- This in-turn affects the business capabilities and consumer trust of the company. Thus, the organizations have moved their focus onto implementing a system which can depict inconsistent transactions, providing banks a privilege to act upon it take necessary measures

9.3 Future Scope of the Work

The credit card fraud detection project has a wide range of potential future scopes for the given improvement and development. Here are some potential areas that can be explored:

1. **Real-time detection:** Enhancing the system to perform real-time analysis of transactions, enabling the identification of fraudulent activity as it happens. This would help prevent fraudulent transactions from being completed successfully.

2. **Enhanced data analysis:** Improving data analysis techniques by incorporating advanced machine learning algorithms and utilizing comprehensive datasets with a wider range of variables. This would enhance the accuracy and effectiveness of fraud detection.

3. **Improved fraud prediction:** Developing more accurate fraud prediction models that can anticipate fraudulent activity before it occurs. Leveraging artificial intelligence and machine learning to analyze transaction patterns and identify potential fraudulent patterns.

4. **Integration with blockchain technology:** Exploring the integration of the credit card fraud detection system with blockchain technology. This integration can provide an additional layer of security and transparency in detecting and preventing fraudulent transactions.

These potential areas of improvement and development can further enhance the effectiveness and efficiency of credit card fraud detection systems, ensuring a higher level of security for users and minimizing financial losses due to fraudulent activities.

REFERENCES

[1] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and G. N. Surname, “Random forest for credit card fraud detection” , IEEE15thInternationalConference on Networking, Sensing and Control (ICNSC) , pp.123-345,2022 .

[2] Satvik Vats, Surya Kant Dubey, Naveen Kumar Pandey, “Tool for Effective Detection of Fraud in Credit Card System” , published in International Journal of Communication Network SecurityISSN:2231– 1882, Volume-2, Issue-1, 2022.

[3] Rinky D. Patel and Dheeraj Kumar Singh, “Credit Card Fraud Detection & Prevention of Fraud Using Genetic Algorithm”, published by International Journal of Soft Computing and Engineering(IJSCE)ISSN: 2231-2307, Volume-2, Issue-6, January 2019.

[4] Rinky D. Patel and Dheeraj Kumar Singh, “Credit Card Fraud Detection Prevention of Fraud Using Genetic Algorithm” , published by International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307,Volume-2, Issue-6, January 2019.

[5] Wen-Fang YU, Na Wang, “Research on Credit Card Fraud Detection Model Based on Distance Sum” , published by IEEE International Joint Conference on Artificial Intelligence, pp.243-256 , 2020.

[6] Andreas L. Prodromitids and Salvatore J. Stolfo; "Agent-Based Distributed Learning Applied to Fraud Detection"; Department of Computer Science-Columbia University, pp.145-167 , 2021.

[7] P. K. Chan, W. Fan, A. L. Prodromidis, and S. J. Stolfo, „„Distributed data mining in credit card fraud detection,"" IEEE Intel. Syst. Appl., pp. 67–74,2022.

[8] Soltani, N., Akbari, M.K., SargolzaeiJavan, M., “An user- based model for credit card fraud detection based on artificial immune system,” Artificial Intelligence and Signal Processing (AISP),16th CSI International Symposium on., IEEE, pp. 029- 033, 2021.

[9] S. Ghosh and D. L. Reilly, “Credit card fraud detection with a neural network”, Proceedings of the 27th Annual Conference on System Science, Volume 3: Information Systems: DSS/ Knowledge Based Systems, pp.621-630,2021.

[10] Masoumeh Zareapoor, seeja.K.R, M.Afshar.Alam, “Analysis of Credit Card Fraud Detection Techniques: based on Certain Design Criteria”, International Journal of Computer Applications , pp. 975 – 8887, Volume52–No.3, 2021.