CHAPTER-1

INTRODUCTION

CHAPTER 1

INTRODUCTION

Nowadays Credit card usage has been drastically increased across the world, now people believe in going cashless and are completely dependent on online transactions. The credit card has made the digital transaction easier and more accessible. A huge number of dollars of loss are caused every year by the criminal credit card transactions. Fraud is as old as mankind itself and can take an unlimited variety of different forms. The PwC global economic crime survey of 2017 suggests that approximately 48% of organizations experienced economic crime. Therefore, there's positively a necessity to unravel the matter of credit card fraud detection. Moreover, the growth of new technologies provides supplementary ways in which criminals may commit a scam. The use of credit cards is predominant in modern day society and credit card fraud has been kept on increasing in recent years. Huge Financial losses have been fraudulent effects on not only merchants and banks but also the individual person who are using the credits. Fraud may also affect the reputation and image of a merchant causing non-financial losses that. For example, if a cardholder is a victim of fraud with a certain company, he may no longer trust their business and choose a competitor.

Credit cards are widely used due to the popularization of ecommerce and the development of mobile intelligent devices. Credit card has made an online transaction easier and more convenient. Fraud detection is a process of monitoring the transaction behaviour of a cardholder in order to detect whether an incoming transaction is done by the cardholder or others. Credit card fraud detection is a relevant problem that draws the attention onal intelligence communities, where a large number of automatic solutions have been proposed. In a real-world FDS, the massive stream of payment requests is quickly scanned by automatic tools that determine which transactions to authorize.

Classifiers are typically employed to analyze all the authorized transactions and alert the most suspicious ones. Alerts are then inspected by professional investigators that contact the cardholders to determine the true nature (either genuine or fraudulent) of each alerted transaction. By doing this, investigators f machine-learning and computatio provide a feedback to the system in the form of labelled transactions, which can be used to train or update the classifier, in order to preserve (or eventually improve) the fraud-detection performance over time. The vast majority of transactions cannot be verified by investigators for obvious time and cost constraints. These transactions remain unlabeled until customers discover and report frauds. Another important difference between what is typically done in the literature and the real-world operating conditions of Fraud-Detection System (FDS) concerns the measures used to assess the frauddetection performance. We use random forest to train the normal and fraud behaviour features. Random forest is a classification algorithm based on the votes of all base classifiers.

Fraud is a major issue that has an impact on austerity and nations. Credit cards are a key issue that has grown in popularity as a result of internet sales and daytime purchasing. This can take numerous forms, including looking for a credit card or being the victim of identity theft. There are two types of fraud: physical and virtual. As a result, the key goal was to figure out how to do behavioral analysis on this form of fraud. The banking association has devised a multi-packet fraud protection solution, comparable to the transportation verification system, credit card approval, and rules-based monitoring, to address this issue. A fraudster is known to be incontinent and will be blocked if a card is planted as a victim of fraud. The scammer made several attempts to obtain immoral information about the card of the seller, in a manner similar to phishing. Fraudulent selling reveals the relationship between reality and the discovery of anomalies in the features that reveal the packaging details of the fraud. The fraudster quickly recognizes the thresholds and takes advantage of and exploits stationary instruments.

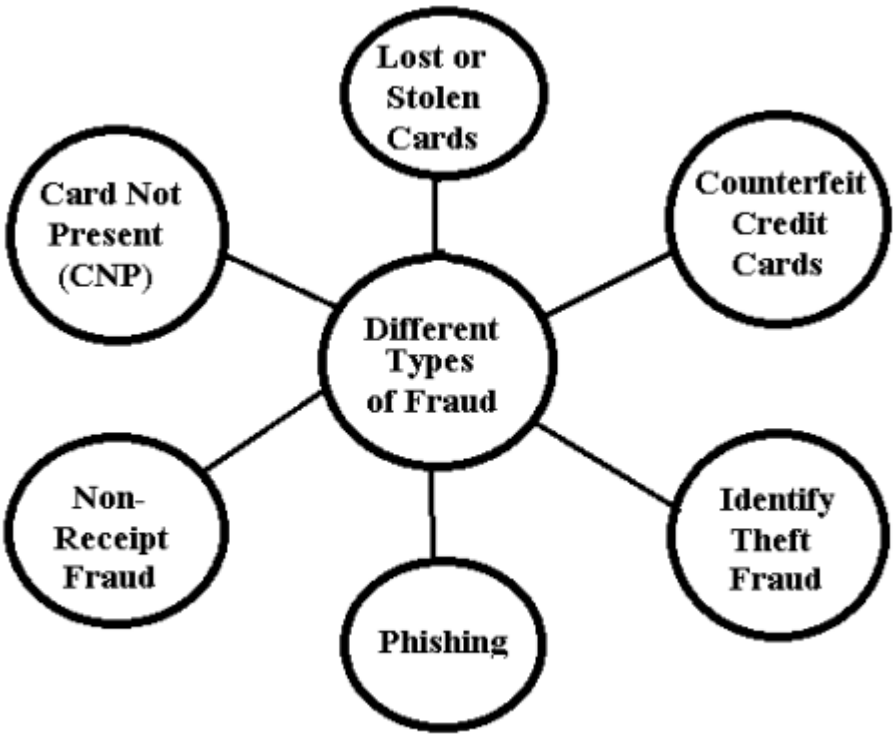

Fig 1: Different types of frauds

1.1 Problem Statement

Billions of dollars of loss are caused every year by the fraudulent credit card transactions. Fraud is old as humanity itself and can take an unlimited variety of different forms. The PwC global economic crime survey of 2017 suggests that approximately 48% of organizations experienced economic crime. Therefore, there is definitely an urge to solve the problem of credit card fraud detection. Moreover, the development of new technologies provides additional ways in which criminals may commit fraud. The use of credit cards is prevalent in modern day society and credit card fraud has been kept on growing in recent years.

Credit card fraud is a sober and major growing problem in banking industries. With the advent of the rise of many web services provided by banks, banking frauds are also on the increase. Banking systems always have a strapping security system in order to detect and prevent fraudulent activities of any category of transactions. Totally eliminating banking fraud is almost unfeasible, but we can however minimize the frauds and prevent them from happening by machine learning techniques like Data Mining. It represents how to utilize these data and find useful information from data has become an urgent need for detection of fraud. Therefore, data mining technology has become an effective method for detection of fraud. Thus we are developing a fraud detection system for credit cards using decision tree induction algorithm for security using Data Mining Technique.

1.2 Research Objective

We propose a Machine learning model to detect fraudulent credit card activities in online financial transactions. Analyzing fake transactions manually is impracticable due to vast amounts of data and its complexity. However, adequately given informative features, could make it is possible using Machine Learning. This hypothesis will be explored in the project. To classify fraudulent and legitimate credit card transaction by supervised

learning Algorithm such as Random Forest. To help us to get awareness about the fraudulent and without loss of any financially. Due to increasing use of E-trade, over there has been large use of credit cards for online shopping which brought about a large wide variety of frauds narrated to credit cards. The most important goal is to make a fraud detection set of rules, which unearths the fraud transactions with less time and lofty accuracy with the aid of making use of system acquiring based class algorithms. As technology is transferring send swiftly, the price by way of cash is reduced and online payment receives multiplied, this enables way for the fraudsters to make unidentified transactions[2]. There're a lot of two types of credit card frauds. One is theft of physical card, and different one is stealing sensitive facts from the card, for instance card number, cvv code, type of card and other. Through stealing credit card information, a fraudster can negate a massive wide variety of money or make a huge amount of buy prior to cardholder reveals out. As a consequence of that, corporations use several machine acquiring strategies to look which transactions are fake and which are not.

The surge in charges reaches as consumer spending plummets, leaving card issuers and consumers at a swiftly blooming risk of account fraud. The rise in such attempts reaches as millions of people failed to make card payments. Fraud detection entails monitoring and analyzing the behavior of several users in order to estimate detect or avoid undesirable behavior. In order to identify credit card fraud detection productively, we need to understand the various machine learning technologies, algorithms and types involved in detection of credit card frauds. The objective of this paper is to analyze several machine learning algorithms, for instance Random Forest (RF), KNearest Neighbor (KNN), Local Outlier Fuction (LOF) to work out which algorithm is the best for credit card fraud detection.

1.3 Project Scope and Limitations

In this proposed project we designed a protocol or a model to detect the fraud activity in credit card transactions. This system is capable of providing most of the essential features required to detect fraudulent and legitimate transactions

Objectives:

The main objective of credit card fraud detection using the Random Forest (CART) algorithm is to prevent and minimize losses due to fraudulent activities related to credit card transactions. The credit card companies can use this technique to detect fraudulent transactions in real-time or post-transaction analysis, allowing them to take necessary actions quickly to reduce the financial loss to the company and its customers.

The Random Forest (CART) algorithm is a popular machine learning algorithm that can be trained on a large dataset of credit card transactions to identify patterns and anomalies that indicate fraudulent activities.

The objectives of using the Random Forest (CART) algorithm for credit card fraud detection include:

- Minimizing financial losses due to fraudulent activities.
- Improving customer satisfaction by reducing the incidence of fraudulent transactions.
- Enhancing the reputation of credit card companies by providing reliable and secure payment services.
- Increasing the efficiency and accuracy of fraud detection and prevention.
- Reducing the workload and costs associated with manual fraud detection.
- Providing real-time fraud detection and prevention, enabling quick action to be taken against fraudulent activities.

Overall, the use of the Random Forest (CART) algorithm for credit card fraud detection can significantly improve the security and reliability of credit card transactions while reducing the risk of financial losses due to fraudulent activities.

Limitations

While the Random Forest (CART) algorithm is a powerful tool for credit card fraud detection, it does have some limitations. Here are some of the most common limitations of this approach:

- Imbalanced datasets: Random Forest (CART) algorithm is not effective when the dataset is imbalanced, where the number of fraudulent transactions is significantly less than the number of non-fraudulent transactions. This can result in an algorithm that is biased towards identifying non-fraudulent transactions and failing to detect fraudulent ones.
- Complexity: The Random Forest (CART) algorithm is a complex algorithm that can be difficult to interpret and explain, making it challenging for stakeholders to understand how the algorithm arrives at its decisions.
- Time-sensitive: In real-time fraud detection, the Random Forest (CART) algorithm may not be able to keep up with the volume and speed of transactions, which may cause delays and increase the risk of fraudulent activities.

CHAPTER-2

BACKGROUND WORK

CHAPTER 2

BACKGROUND WORK

2.1 Credit Card Fraud Detection using Machine Learning Algorithms

2.1.1 INTRODUCTION

Credit card generally refers to a card that is assigned to the customer (cardholder), usually allowing them to purchase goods and services within credit limit or withdraw cash in advance. Credit card provides the cardholder an advantage of the time, i.e., it provides time for their customers to repay later in a prescribed time, by carrying it to the next billing cycle.

Credit card frauds are easy targets. Without any risks, a significant amount can be withdrawn without the owner's knowledge, in a short period. Fraudsters always try to make every fraudulent transaction legitimate, which makes fraud detection very challenging and difficult task to detect.

With different frauds mostly credit card frauds, often in the news for the past few years, frauds are in the top of mind for most the world's population. Credit card dataset is highly imbalanced because there will be more legitimate transaction when compared with a fraudulent one.

As advancement, banks are moving to EMV cards, which are smart cards that store their data on integrated circuits rather than on magnetic stripes, have made some on-card payments safer, but still leaving card-not-present frauds on higher rates.

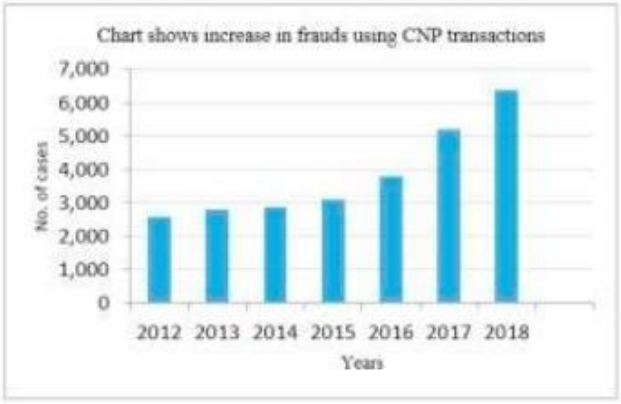

Fig 2: Fraud ratio

In 2019 Sahayasakila V, D.Kavya Monisha, Aishwarya, Sikhakolli Venkatavisalakshiswshai Yasaswi have clarified the Twain significant algorithmic strategies which are the Whale Optimization Techniques (WOA) and SMOTE (Manufactured Minority Oversampling Techniques). They were meant to improve the union speed and to address the information lop-sidedness issue. The class irregularity issue is survived utilizing the SMOTE strategy and the WOA procedure. The Destroyed procedure separates every one of the exchanges which are orchestrated is again re-tested to check the information exactness and is enhanced utilizing the WOA procedure. The calculation speed, unwavering quality, what's more, proficiency of the framework.

In 2018 Navanushu Khare and Saad Yunus Sait have clarified their work on choice trees, Random Forest, SVM, and strategic relapse. They have taken the profoundly slanted dataset and dealt with such kind of dataset. The execution assessment depends on exactness, affectability, explicitness, and exactness. The outcomes show that the exactness for the Logistic Regression is 97.7%, for Decision Trees is 95.5%, for Random Forest is 98.6%, for SVM classifier is 97.5%. They have reasoned that the Random Forest calculation has the most noteworthy precision among the other calculations and is considered as the best calculation to recognize the misrepresentation. They likewise reasoned that the SVM calculation has an information lop-sidedness issue and doesn't give better outcomes to identify Visa misrepresentation

2.1.2 Merits and Demerits

Merits

- In this protocol, a sender can share multiple frames and then wait for the acknowledgment.
- This protocol has much better efficiency in comparison, with low time delay.
- This protocol requires sorting for increased efficiency and applies full-duplex transmission

Demerits

- Does not have the 'amount' feature is the transaction amount which helps to identify the fraud
- It's clear that this method is almost unusable, especially for real-time object detection

2.1.3 IMPLEMENTATION

- Firstly, we use clustering method to divide the cardholders into different clusters/groups based on their transaction amount, i.e., high, medium and low using range partitioning.
- Using Sliding-Window method, we aggregate the transactions into respective groups, i.e., extract some features from window to find cardholder's behavioural patterns. Features like maximum amount, minimum amount of transaction, followed by the average amount in the window and even the time elapsed.
- Every time a new transaction is fed to the window the old once are removed and step-2 is processed for each group of transactions. (Algorithm for Sliding-Window based method to aggregate are referred from [1]).
- After pre-processing, we train different classifiers on each group using the cardholders' behavioral patterns in that group and extract fraud features. Even when we apply classifiers on the dataset, due to imbalance (shown in fig 4) in the dataset, the classifiers do not work well on the dataset. Thus, we perform SMOTE (Synthetic Minority Over-Sampling Technique) operation on the dataset.
- Oversampling does not provide any good results.
- Thus, there are two different ways of dealing with imbalance dataset i.e., consider Matthew

Coefficient Correlation of the classifier on the original dataset or we make use of one-class classifiers.

- Finally, the classifier that is used for training the group is applied to each cardholder in that group. The classifier with the highest rating score is considered as cardholder's recent behavioral pattern.
- Once the rating score [1] is obtained, now we append a feedback system, wherein the current transaction and updated rating score are given back to the system (for further comparison) to solve the problem of concept drift.

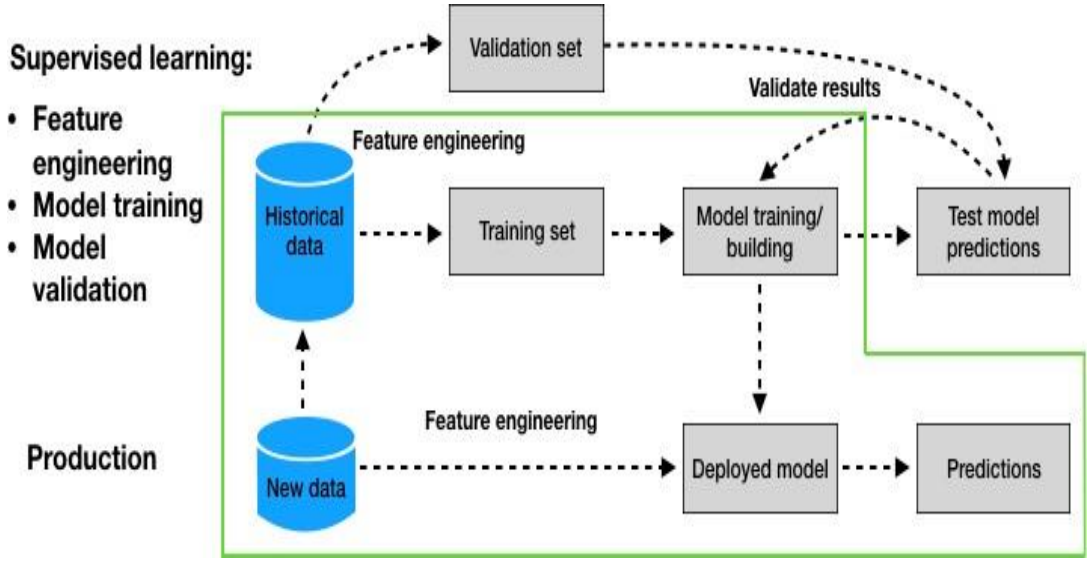

Fig 3: Flow chart

2.1.4 Uml Diagram:

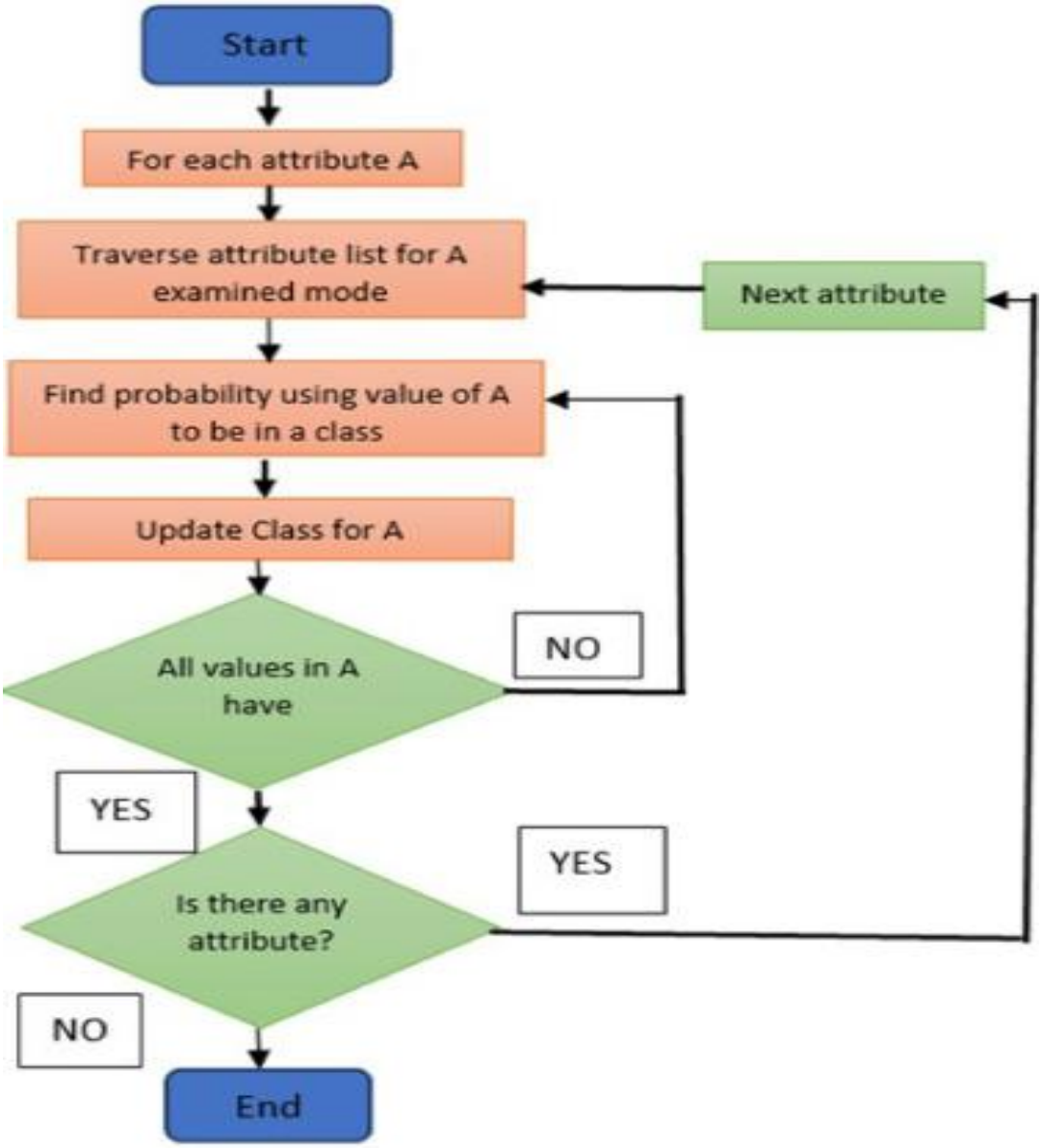

Fig 4: UML diagram

2.2 Credit Card Detection using KNN

2.2.1 INTRODUCTION

In credit card transactions, fraud is defined as the unlawful and unwanted use of an account by someone who is not the account's authorized user. This misuse, as well as the behavior of such fraudulent operations, can be investigated in order to reduce it and prevent such occurrences in the future. In simple terms, credit card fraud occurs when a person uses another person's credit card for personal gain while the owner and card-issuing authorities are unaware of the transaction. It is currently one of the most serious risks to enterprises. However, to fight the fraud completely, it is essential to first understand the structure of executing a fraud. Credit card fraudsters opt many numbers of ways to commit fraud. Card fraud occurs when a physical card is stolen or when critical account information is stolen, such as the card account number or other information that must be available to conduct a transaction.

K-nearest neighbors (KNN) algorithm is a type of supervised machine learning algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry.

The KNN algorithm obtained following results:
Precision: 92.95%
Recall: 80.43%
Accuracy: 99.95%

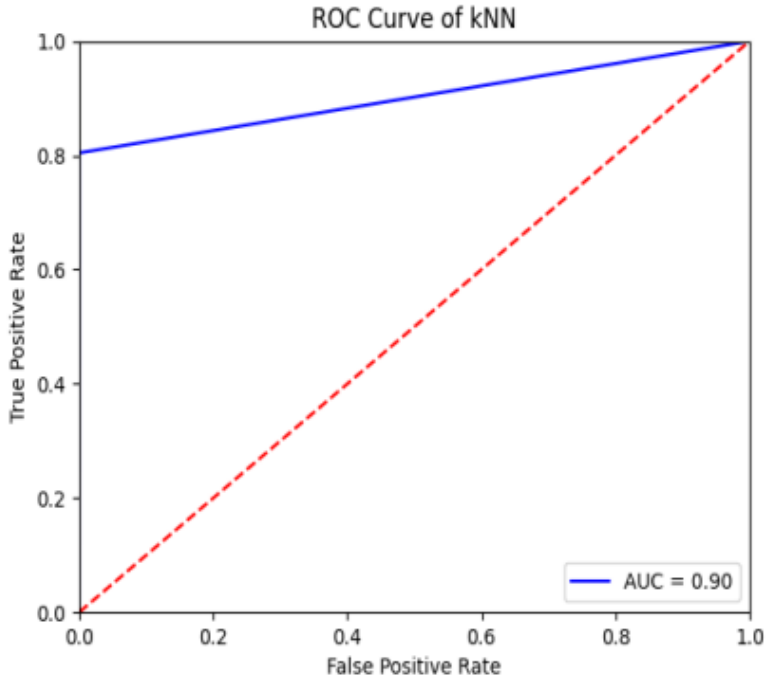

Fig 5: KNN ROC curve

In the above graph, false positive rate at point 0.0 - the true positive rate is at point 0.8. This algorithm shows the highest accuracy i.e. 99.95% and give best results than other algorithms

2.2.2 Merits and Demerits

Merits

- It's easy to understand and simple to implement.
- It can be used for both classification and regression problems.
- It's ideal for non-linear data since there's no assumption about underlying data.

Demerits

- Less importance of variables in a regression or classification problem in a natural way can be done by Random Forest Algorithm.
- Less accuracy, recall score and f1 score

2.2.3 Implementation

The proposed techniques emphasizes on detecting Credit Card Fraudulent transactions whether it is a genuine/non fraud or a fraud transaction and the approaches used to separate fraud and non-fraud are KNN, Decision Tree, Logistic regression, XGBoost, Random forest and Finally we will observe which approach is best for detecting credit card frauds

2.2.3.1 SYSTEM ARCHITECTURE:

The system architecture has following steps:

- Import of Necessary Packages
- Read the Dataset
- Exploratory Data Analysis i.e. finding null values, duplicate values etc.
- Selecting Features (X) and the Target (y) columns
- Train Test Split will split the whole dataset into train and test data
- Build the model i.e. Training the model
- Test the model i.e. Model prediction
- Evaluation of the system i.e. Accuracy score, F1- score

Various anomaly detection algorithms have exploited the concept of nearest neighbour analysis. Three primary elements influence the performance of the KNN algorithm:

- The distance metric used to locate the nearest neighbors.
- The distance rule that is used to classify k nearest neighbours.
- The fresh sample was classified based on the number of neighbours it had.

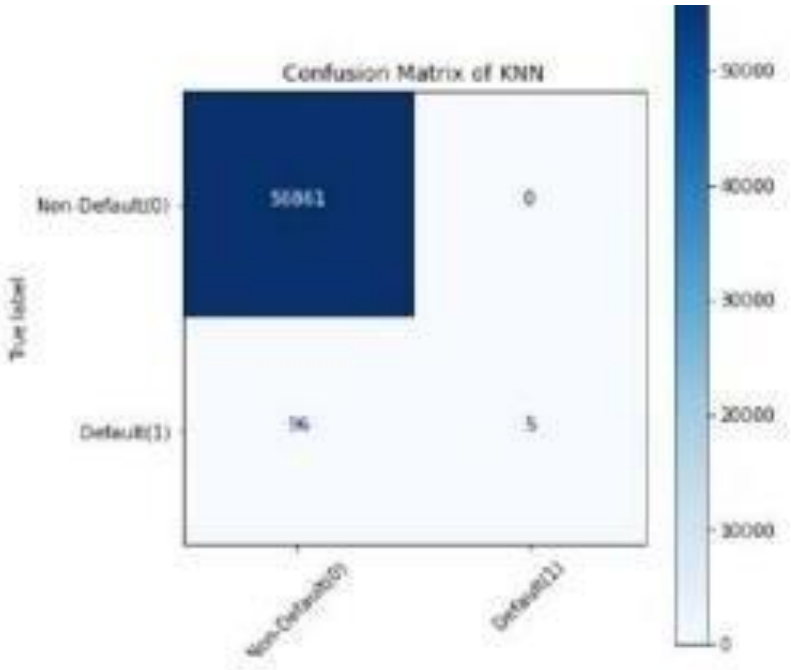

Fig 6: Confusion matrix of KNN

2.2.3.2 Data flow diagram

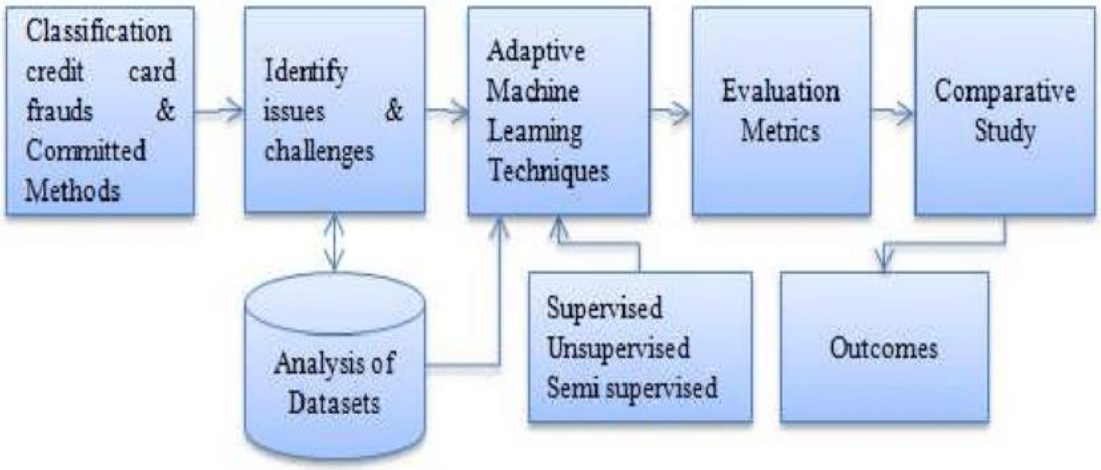

Fig 7: Data flow diagram

2.2.4 Conclusion

In credit card fraud detection, we frequently deals with highly imbalanced datasets. For the chosen dataset from Kaggle, we show that our proposed algorithms are able to detect fraud transactions with very high accuracy and low false positive rate.

Hence for better performance, our result shows that classification of algorithms done by preprocessing data rather than raw data. Because of applying preprocessing technique and K-Means algorithm on the dataset, output of algorithms is with high accuracy and give best results. Hence the comparison was done and it was concluded that K-Nearest Neighbor gives the best results. This was established using accuracy, precision and recall. Balancing dataset and feature selection is important in achieving significant results.

In future, to enhance the system, other machine learning algorithms or artificial neural networks approaches can be used to detect frauds in credit card

2.3. Different algorithms techniques :

Transactional fraud detection based on artificial intelligence techniques for visa transactions is highly caught attention in the research world. There is data available in bulks is to preprocess, analyze, and derive a suitable conclusion of it. Often it is said to be that there is a need to derive the best results out of the data that is available based on different algorithms. Here, there are few methods which are discussed that have been proposed until now.

2.3.1 Kernel-Based:

Hash Functions in general, the hash technique is acquainted to develop a group of hash functions. This assistance in mapping the high-dimensional information to the lower-dimensional articulations, and this procedure is done inside the hash code space [9]. The main advantage in this process is that it increases the pace at which the nearest neighbor scanning in a huge data set for finding the nearest fraud detections which consist of many fraudulent examples. Kernel plays an important role in constructing all the basic hash functions. It has been proved that it tackles the linear inseparable problem.

$$h(x) = \operatorname{sgn}\left(\sum_{j=0}^{m} k\left(x_{(j)}, x\right) a_j - b\right)$$

Here, denotes the hash function, denotes the kernel, and x(j)...x(m)denotes the sample training set. After obtaining these values, KSH model is trained with labeled records. For every tested dataset, first map the hash function with KSH.

Fraud Detection for Credit Card Transactions Using …

model and it helps in finding the most similar training samples that are labeled as fraudulent. In most of the cases, it uses k-nearest neighbor in fraud detection

2.3.2 Bayesian Network:

Bayesian network is utilized to assemble a coordinated non-cyclic chart which installed by a conditional probability distribution to build the non-cyclic graph [10].Consider. four random variables A, B, C, and D. Given are the minimal joint probabilities for each pair of factors, that is the probabilities of the form P(A, B), P(A, C)and so on, and the conditional probability P(A, B|C, D).P(A, C|B, D) is calculated as

$$P(A, C|B, D) = \frac{P(A, B, C, D)}{P(B, D)} = \frac{P(A, B|C, D)P(C, D)}{P(B, D)}$$

2.3.3 Clustering Mode:

The clustering models identify the records and grouping the records as per the cluster to which they have a place. Clustering model firmly corresponded to information which is fixated on distribution models. While performing cluster analysis, first separate the arrangement of information into clusters. Clustering is addition-ally utilized in perceiving applications. To find the credit card fraud, the clustering calculation is basic and most extreme predominant calculation than other AI calculations [8]. K-means clustering technique is an unsupervised clustering model. This technique is mainly used for clustering particular classes in a data set.

$$J = \sum_{i=1}^{k} \sum_{n \varepsilon Sj} \left| x_n - \mu_j \right|^2$$

Here, it is the vector representing nth data point, and is the centroid of the data points in so.

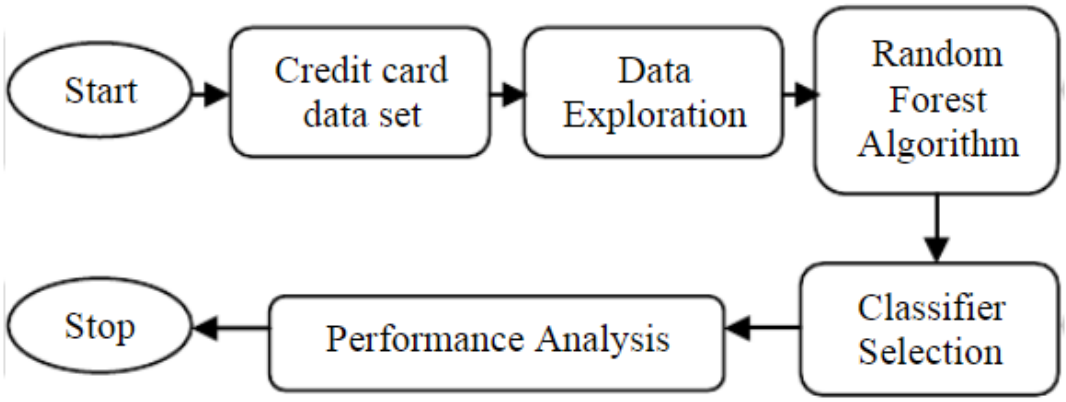

Fig 8: Data flow diagram

CHAPTER-3

PROPOSED SYSTEM

CHAPTER 3
PROPOSED SYSTEM

3.1 Objective of proposed model

In proposed System, we are applying random forest algorithm for classification of the credit card dataset. Random Forest is an algorithm for classification and regression. Summarily, it is a collection of decision tree classifiers. Random forest has an advantage over decision tree as it corrects the habit of over fitting to their training set. A subset of the training set is sampled randomly so that to train each individual tree and then a decision tree is built, each node then splits on a feature selected from a random subset of the full feature set. Even for large data sets with many features and data instances training is extremely fast in random forest and because each tree is trained independently of the others. The Random Forest algorithm has been found to provide a good estimate of the generalization error and to be resistant to over fitting.

Random Forest Algorithm is used to detect the accuracy of the fraud in the transaction. Random choice forests are another name for random forests. These are a categorization, regression, and other tasks that use an ensemble learning strategy that involves teaching a greater number of decision trees and then determining the norm of the classifications (categorization) or the overall prediction (regression) of each tree. Random Forest is a supervised classification technique that uses ensemble learning. Ensemble model is a type of machine learning in which multiple versions of a same algorithm are combined to create a far more effective predictive model.

3.2 ADVANTAGES OF PROPOSED SYSTEM

Random forest ranks the importance of variables in a regression or classification problem in a natural way can be done by Random Forest. • The 'amount' feature is the transaction amount. Feature 'class' is the target class for the binary classification and it takes value 1 for positive case (fraud) and 0 for negative case (not fraud).

3.3 Algorithm Used for Proposed

Random Forest Algorithm:

Random forests is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forests create decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance. Python SKLEARN inbuilt contains support for CART with all decision trees and random forest classifier.

Random forests have a variety of applications, such as recommendation engines, image classification and feature selection. It can be used to classify loyal loan applicants, identify fraudulent activity, and predict diseases. It lies at the base of the Boruta algorithm, which selects important features in a dataset

How Random forest Algorithm Working In Fraud Detection

In the case of credit card fraud detection, the Random Forest algorithm works by combining multiple decision trees to create a more robust and accurate model for classification.

Each decision tree in the Random Forest is built on a subset of the training data and considers only a subset of the available features. This helps to reduce overfitting and increases the diversity of the trees in the forest.

During training, the Random Forest algorithm randomly selects a subset of the available features for each decision tree. This means that each tree in the forest has a

different set of features to work with, which helps to increase the diversity of the forest and reduce the correlation between the trees.

When a new credit card transaction is presented to the model for classification, it is evaluated by each of the decision trees in the forest. Each tree makes a prediction on whether the transaction is fraudulent or legitimate based on its own subset of features. The final prediction of the Random Forest is then based on the majority vote of the individual trees.

Random Forest algorithm has several advantages in the case of credit card fraud detection. It can handle imbalanced data, where the number of fraudulent transactions is much lower than the number of legitimate transactions. It can also handle missing values and noisy data. Furthermore, Random Forest algorithm is able to capture non-linear relationships between features and interactions between them, making it well-suited for this type of classification problem.

steps to use the Random Forest algorithm for credit card fraud detection:

1 Data Collection: Collect the transaction data from credit card companies or other sources. The data should include information such as transaction amount, transaction time, transaction location, and other relevant details.

2 Data Preprocessing: The collected data needs to be cleaned and preprocessed before applying the algorithm. This involves handling missing data, outliers, and transforming categorical variables into numerical variables.

3 Feature Selection: Select the most relevant features that contribute to the classification of fraudulent transactions. This step can be done by using statistical tests, correlation analysis, or domain knowledge.

4 Train/Test Split: Split the data into training and testing sets. The training set is used to train the Random Forest model, while the testing set is used to evaluate the model's performance.

5 Model Training: Train the Random Forest model on the training set. This involves specifying the number of trees in the forest and other hyperparameters. The model will learn to classify transactions as either fraudulent or legitimate based on the selected features.

6 Model Evaluation: Evaluate the performance of the Random Forest model on the testing set. This involves computing metrics such as accuracy, precision, recall, F1 score, and ROC curve.

7 Hyperparameter Tuning: Fine-tune the hyperparameters of the Random Forest algorithm to improve its performance. This can be done by using techniques such as grid search or randomized search.

8 Model Deployment: Deploy the trained Random Forest model in a production environment for real-time credit card fraud detection.

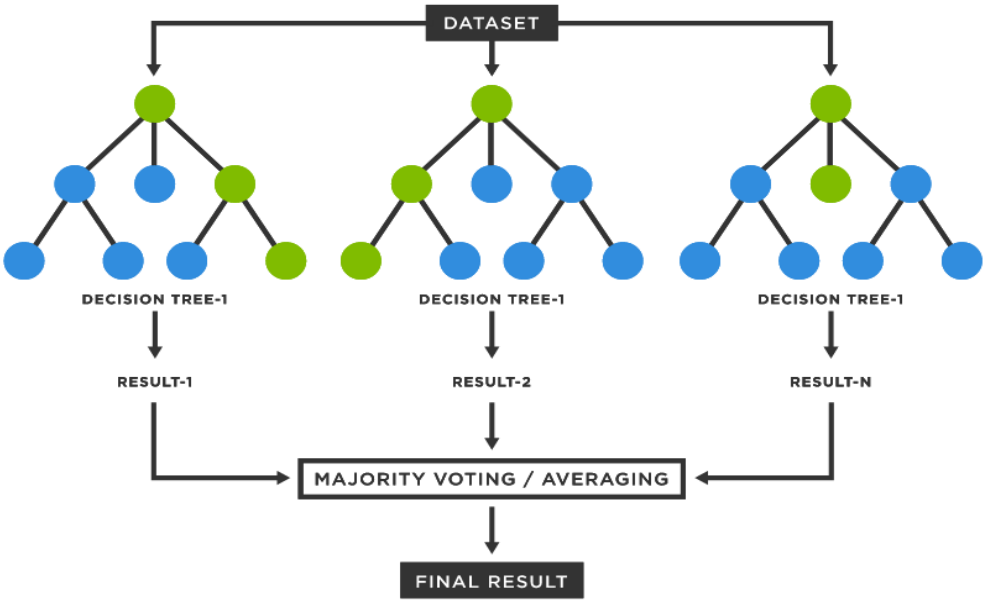

Fig 9: Random forest

3.4 ADVANTAGES OF USING RANDOM FOREST

Pros of using random forest for classification and regression.

1. The random forest algorithm is not biased, since, there are multiple trees and each tree is trained on a subset of data. Basically, the random forest algorithm relies on the power of "the crowd"; therefore, the overall biasedness of the algorithm is reduced.
2. This algorithm is very stable. Even if a new data point is introduced in the dataset the overall algorithm is not affected much since new data may impact one tree, but it is very hard for it to impact all the trees.
3. The random forest algorithm works well when you have both categorical and numerical features

3.5 Designing

3.5.1 UML DIAGRAMS:

UML represents Unified Modeling Language. UML is an institutionalized universally useful showing dialect in the subject of article situated programming designing. The fashionable is overseen, and become made by way of, the Object Management Group.

The goal is for UML to become a regular dialect for making fashions of item arranged PC programming. In its gift frame UML is contained two noteworthy components: a Meta-show and documentation. Later on, a few types of methods or system can also likewise be brought to; or related with, UML.

The Unified Modeling Language is a popular dialect for indicating, Visualization, Constructing and archiving the curios of programming framework, and for business demonstrating and different non-programming frameworks.

The UML speaks to an accumulation of first-rate building practices which have verified fruitful in the showing of full-size and complicated frameworks.

The UML is a essential piece of creating gadgets located programming and the product development method. The UML makes use of commonly graphical documentations to specific the plan of programming ventures.

GOALS:

The Primary goals inside the plan of the UML are as in step with the subsequent:

1. Provide clients a prepared to-utilize, expressive visual showing Language on the way to create and change massive models.
2. Provide extendibility and specialization units to make bigger the middle ideas.
3. Be free of specific programming dialects and advancement manner.
4. Provide a proper cause for understanding the displaying dialect.

5. Encourage the improvement of OO gadgets exhibit.

6. Support large amount advancement thoughts, for example, joint efforts, systems, examples and components.

USE CASE DIAGRAM:

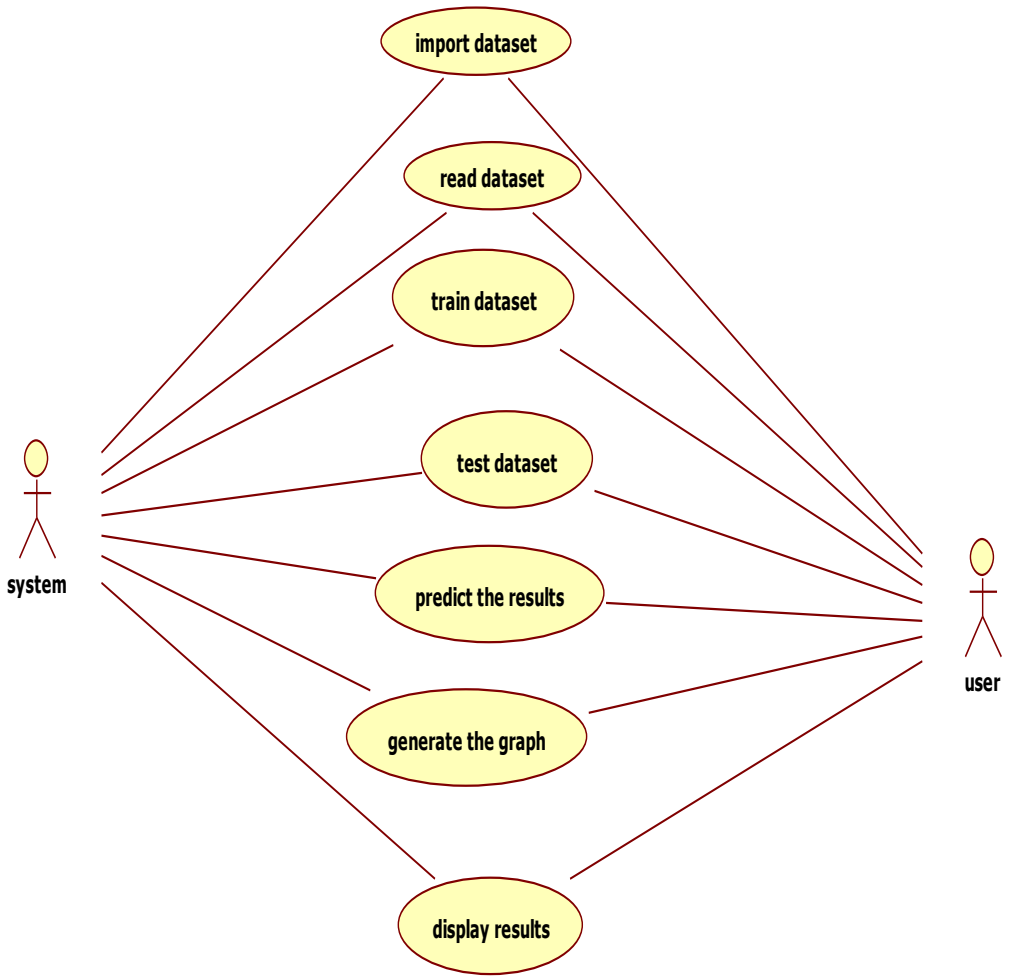

Fig 10: Use case diagram

CLASS DIAGRAM:

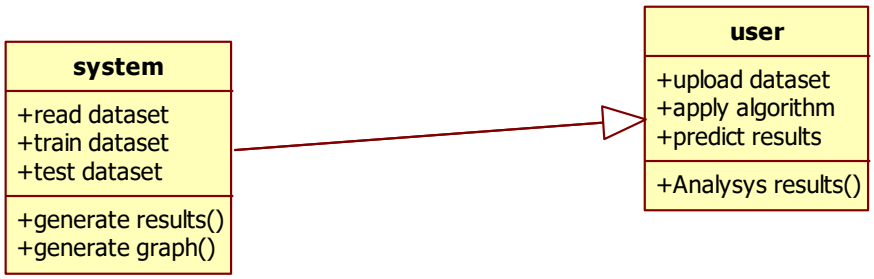

Fig 11: Class diagram

SEQUENCE DIAGRAM:

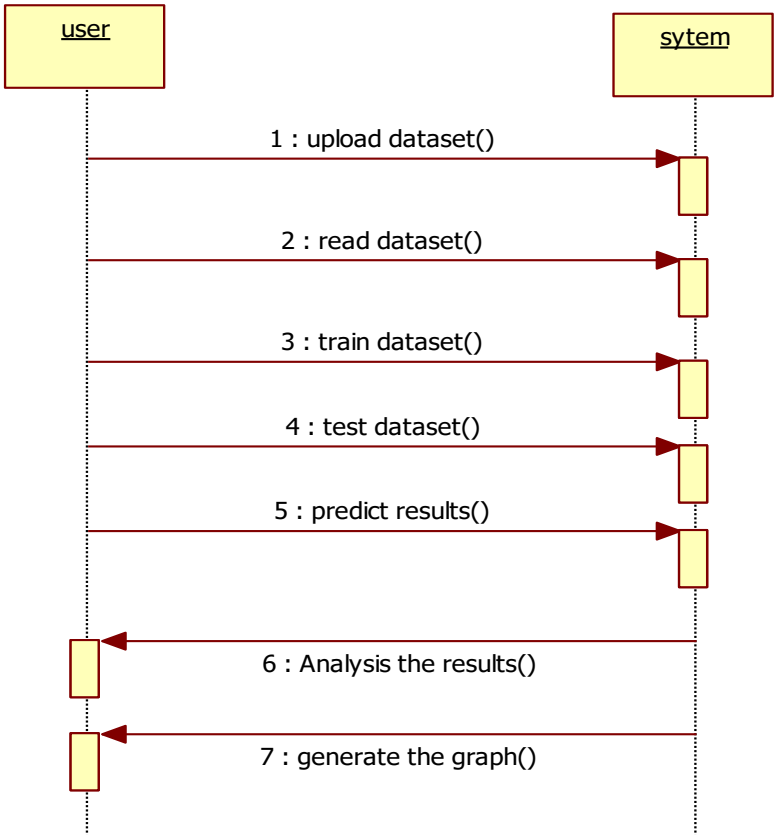

Fig 12: Sequence diagram

COLLABORATION DIAGRAM:

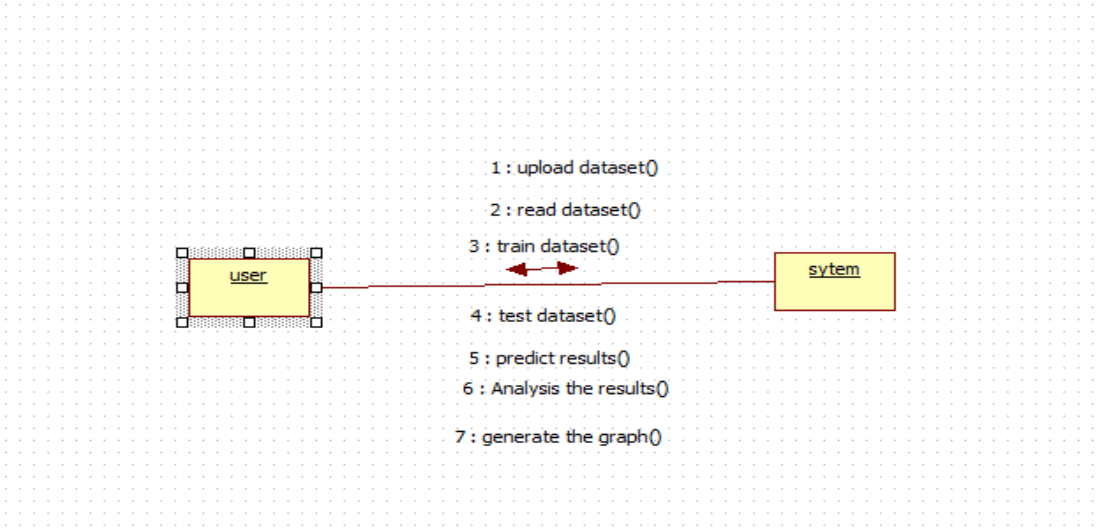

Fig 13: Collaboration diagram

ACTIVITY DIAGRAM:

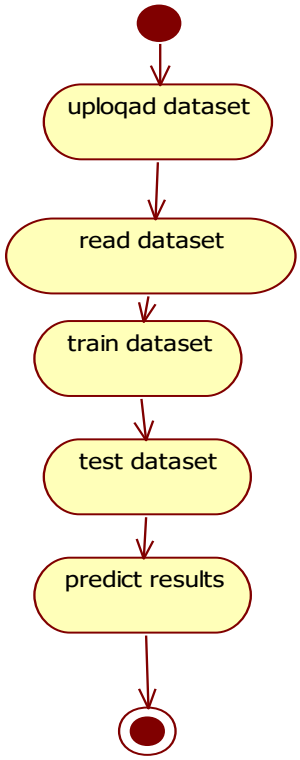

Fig 14: Activity diagram

3.6. SYSTEM ARCHITECTURE:

First the credit card dataset is taken from the source and cleaning and validation is performed on the dataset which includes removal of redundancy, filling empty spaces in columns, converting necessary variable into ors or classes then data is divided into 2 parts, one is training dataset and another one is test data set. Now the original sample is randomly partitioned into teat and train dataset.

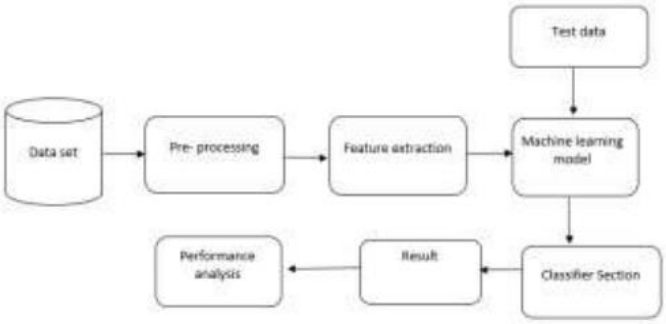

Fig 15: System Architecture

DEPLOYMENT DIAGRAM:

Fig 16: Deployment diagram

3.7 Stepwise Implementation and Code

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
import matplotlib.pyplot as plt
import numpy as np
from tkinter.filedialog import askopenfilename
import numpy as np
import pandas as pd
# data processing, CSV file I/O (e.g. pd.read_csv)
# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory
from sklearn import *
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.ensemble import RandomForestClassifier
#from sklearn.tree import export_graphviz
#from IPython import display

main = tkinter.Tk()
main.title("Credit Card Fraud Detection") #designing main screen
main.geometry("1300x1200")
global filename
global cls
```

```
global X, Y, X_train, X_test, y_train, y_test
global random_acc # all global variables names define in above lines
global clean
global attack
global total

def traintest(train):     #method to generate test and train data from dataset
X = train.values[:, 0:29]
Y = train.values[:, 30]
print(X)
print(Y)
X_train, X_test, y_train, y_test = train_test_split(
X, Y, test_size = 0.3, random_state = 0)
return X, Y, X_train, X_test, y_train, y_test

def generateModel(): #method to read dataset values which contains all five features data
global X, Y, X_train, X_test, y_train, y_test
train = pd.read_csv(filename)
X, Y, X_train, X_test, y_train, y_test = traintest(train)
text.insert(END,"Train & Test Model Generated\n\n")
text.insert(END,"Total Dataset Size : "+str(len(train))+"\n")
text.insert(END,"Split Training Size : "+str(len(X_train))+"\n")
text.insert(END,"Split Test Size : "+str(len(X_test))+"\n")

def upload(): #function to upload tweeter profile
global filename

filename = filedialog.askopenfilename(initialdir="dataset")
text.delete('1.0', END)
text.insert(END,filename+" loaded\n");
```

```
def prediction(X_test, cls):  #prediction done here
y_pred = cls.predict(X_test)
for i in range(50):
print("X=%s, Predicted=%s" % (X_test[i], y_pred[i]))
return y_pred

# Function to calculate accuracy
def cal_accuracy(y_test, y_pred, details):
accuracy = accuracy_score(y_test,y_pred)*100
text.insert(END,details+"\n\n")
text.insert(END,"Accuracy : "+str(accuracy)+"\n\n")
return accuracy

def runRandomForest():
headers =
["Time","V1","V2","V3","V4","V5","V6","V7","V8","V9","V10","V11","V12","V13
","V14","V15","V16","V17","V18","V19","V20","V21","V22","V23","V24","V25","
V26","V27","V28","Amount","Class"]
global random_acc
global cls
global X, Y, X_train, X_test, y_train, y_test
cls =
RandomForestClassifier(n_estimators=50,max_depth=2,random_state=0,class_weight
='balanced')

cls.fit(X_train, y_train)
text.insert(END,"Prediction Results\n\n")
prediction_data = prediction(X_test, cls)
random_acc = cal_accuracy(y_test, prediction_data,'Random Forest Accuracy')
#str_tree = export_graphviz(cls, out_file=None, feature_names=headers,filled=True,
```

```
special_characters=True, rotate=True, precision=0.6)
#display.display(str_tree)

def predicts():
global clean
global attack
global total
clean = 0;
attack = 0;
text.delete('1.0', END)
filename = filedialog.askopenfilename(initialdir="dataset")
test = pd.read_csv(filename)
test = test.values[:, 0:29]
total = len(test)
text.insert(END,filename+" test file loaded\n");
y_pred = cls.predict(test)
for i in range(len(test)):
if str(y_pred[i]) == '1.0':
attack = attack + 1
text.insert(END,"X=%s, Predicted = %s" % (test[i], 'Contains Fraud Transaction Signature')+"\n\n")
else:
clean = clean + 1
text.insert(END,"X=%s, Predicted = %s" % (test[i], 'Transaction Contains Cleaned Signatures')+"\n\n")

def graph():
height = [total,clean,attack]
bars = ('Total Transactions','Normal Transaction','Fraud Transaction')
y_pos = np.arange(len(bars))
```

```
plt.bar(y_pos, height)
plt.xticks(y_pos, bars)
plt.show()

font = ('times', 16, 'bold')
title = Label(main, text='Credit Card Fraud Detection Using Random Forest Tree Based Classifier')
title.config(bg='greenyellow', fg='dodger blue')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)

font1 = ('times', 12, 'bold')
text=Text(main,height=20,width=150)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=50,y=120)
text.config(font=font1)

font1 = ('times', 14, 'bold')
uploadButton = Button(main, text="Upload Credit Card Dataset", command=upload)
uploadButton.place(x=50,y=550)
uploadButton.config(font=font1)

modelButton = Button(main, text="Generate Train & Test Model", command=generateModel)

modelButton.place(x=350,y=550)
modelButton.config(font=font1)

runrandomButton = Button(main, text="Run Random Forest Algorithm", command=runRandomForest)
```

```
runrandomButton.place(x=650,y=550)
runrandomButton.config(font=font1)

predictButton = Button(main, text="Detect Fraud From Test Data", command=predicts)
predictButton.place(x=50,y=600)
predictButton.config(font=font1)

graphButton = Button(main, text="Clean & Fraud Transaction Detection Graph", command=graph)
graphButton.place(x=350,y=600)
graphButton.config(font=font1)

exitButton = Button(main, text="Exit", command=exit)
exitButton.place(x=770,y=600)
exitButton.config(font=font1)

main.config(bg='LightSkyBlue')
main.mainloop()
```

3.7.1 Explanation of code:

This is a program written in Python using the Tkinter library for building a GUI. The program uses machine learning algorithms, particularly the Random Forest algorithm, to detect fraud in credit card transactions. The program loads a dataset and generates a model from the dataset. The program then predicts the results and displays the accuracy of the Random Forest algorithm. The program also allows the user to upload a test dataset and predict the results for fraud and normal transactions. Finally, the program displays a graph showing the number of clean transactions and fraudulent transactions.

The program starts by importing the necessary libraries such as Tkinter, Pandas, NumPy, and Scikit-learn. The GUI window is then created using the Tkinter library. The window is given a title, a size, and a background color.

The program then defines several functions for uploading the dataset, generating a model from the dataset, predicting the results, calculating the accuracy of the model, and displaying the results in the GUI window. The program also defines a function for displaying a graph of the results.

The main function of the program is the Random Forest algorithm, which is used to detect fraudulent transactions. The algorithm is trained on the dataset and is then used to predict the results for a test dataset.

Finally, the program displays a graph of the results, showing the number of clean transactions and fraudulent transactions. The user can also view the details of the predictions in the GUI window.

3.7.2 Dataset:

In this project we are using python Random Forest inbuilt Cart algorithm to detect fraud transaction from credit card dataset, we downloaded this dataset from 'kaggles' web site from below URL

Dataset URL: https://www.kaggle.com/mlg-ulb/creditcardfrau

To provide privacy to users transaction data kaggles peoples have converted transaction data to numerical format using PCA Algorithm. Below are some example from dataset

"Time","V1","V2","V3","V4","V5","V6","V7","V8","V9","V10","V11","V12","V13","V14","V15","V16","V17","V18","V19","V20","V21","V22","V23","V24","V25","V26","V27","V28","Amount","Class"

Time: Number of seconds elapsed between this transaction and the first transaction in the dataset

V1-V28: May be result of a PCA Dimensionality reduction to protect user identities and sensitive features(v1-v28)

Amount: amount of the transaction

Class: is the target variable that indicates whether a transaction is fraudulent (Class=1) or legitimate

(Class=0).

0,-1.3598071336738,-0.0727811733098497,2.53634673796914,1.37815522427443,-0.338320769942518,0.462387777762292,0.239598554061257,0.0986979012610507,0.363786969611213,0.0907941719789316,-0.551599533260813,-0.617800855762348,-0.991389847235408,-0.311169353699879,1.46817697209427,-0.470400525259478,0.207971241929242,0.0257905801985591,0.403992960255733,0.251412098239705,-0.018306777944153,0.277837575558899,-0.110473910188767,0.0669280749146731,0.128539358273528,-0.189114843888824,0.133558376740387,-0.0210530534538215,149.62,"0"

0,1.19185711131486,0.26615071205963,0.16648011335321,0.448154078460911,0.0600176492822243,-0.0823608088155687,-0.0788029833323113,0.0851016549148104,-0.255425128109186,-

0.166974414004614,1.61272666105479,1.06523531137287,0.48909501589608,-0.143772296441519,0.635558093258208,0.463917041022171,-0.114804663102346,-0.183361270123994,-0.145783041325259,-0.0690831352230203,-0.225775248033138,-0.638671952771851,0.101288021253234,-0.339846475529127,0.167170404418143,0.125894532368176,-0.00898309914322813,0.0147241691924927,2.69,"0"

406,-2.3122265423263,1.95199201064158,-1.60985073229769,3.9979055875468,-0.522187864667764,-1.42654531920595,-2.53738730624579,1.39165724829804,-2.77008927719433,-2.77227214465915,3.20203320709635,-2.89990738849473,-0.595221881324605,-4.28925378244217,0.389724120274487,-1.14074717980657,-2.83005567450437,-0.0168224681808257,0.416955705037907,0.126910559061474,0.517232370861764,-0.0350493686052974,-0.465211076182388,0.320198198514526,0.0445191674731724,0.177839798284401,0.2611450025676

77,-0.143275874698919,0,"1"

The above bold names are the column names of this dataset and others decimal values are the content of dataset and in above 3 rows last column contains class label where 0 means transaction values are normal and 1 means contains fraud values.

Using above 'CreditCardFraud.csv' file we will train Random Forest algorithm and then we will upload test data file and this test data will be applied on Random Forest train model to predict whether test data contains normal or fraud transaction signatures.

When we upload test data then it will contains only transaction data no class label will be there application will predict and give the result. See below test data file.

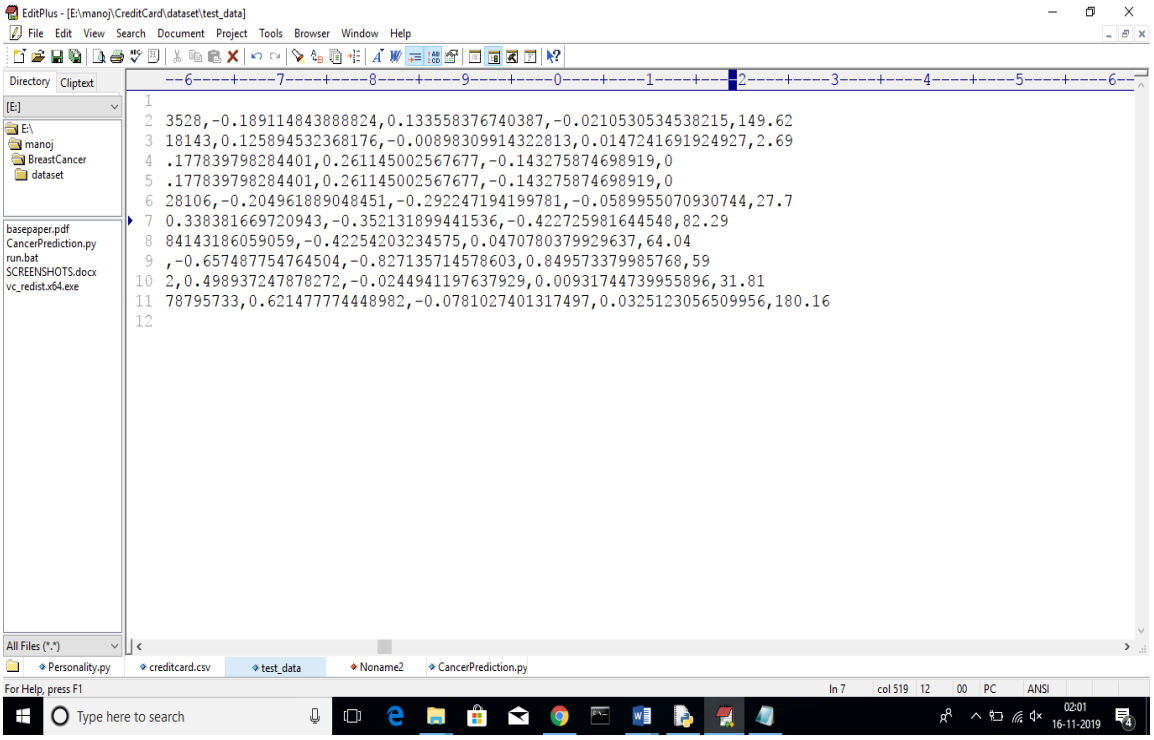

Fig 17: In above screen in test data file there are no 0 or 1 values, application will predict from this test data using random forest and give the result

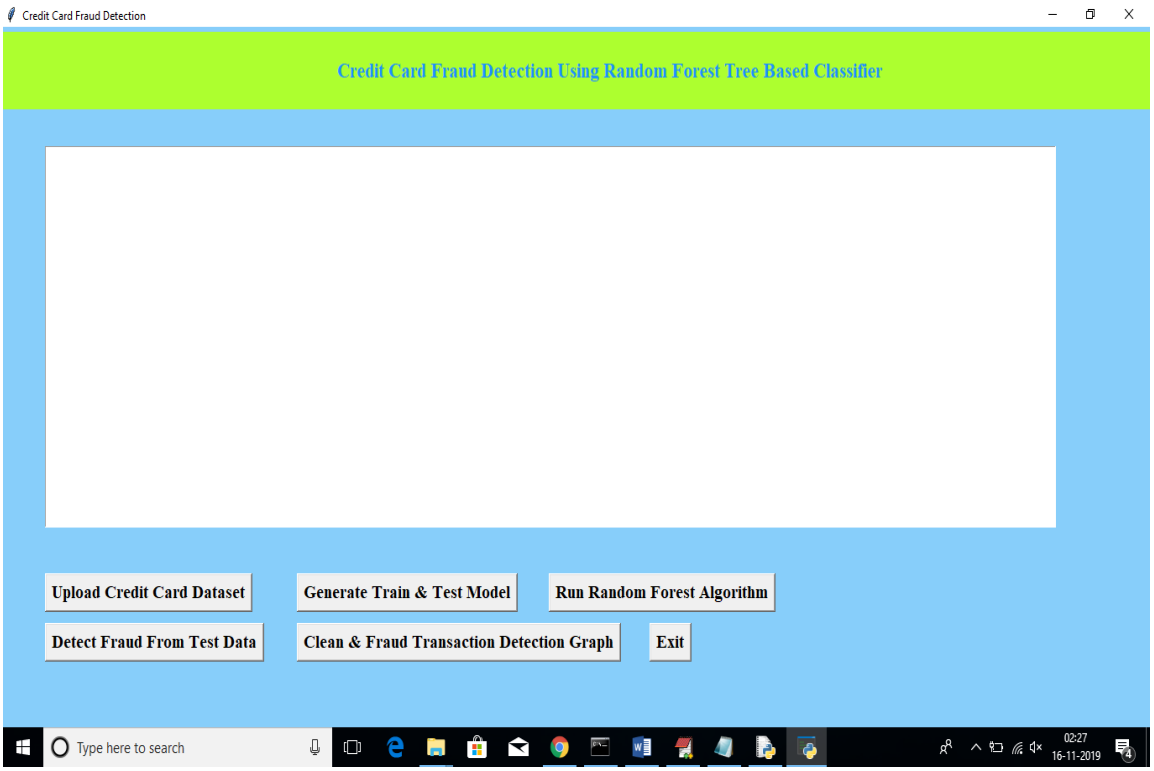

Fig 18: 'Upload Credit Card Dataset' button to upload dataset.

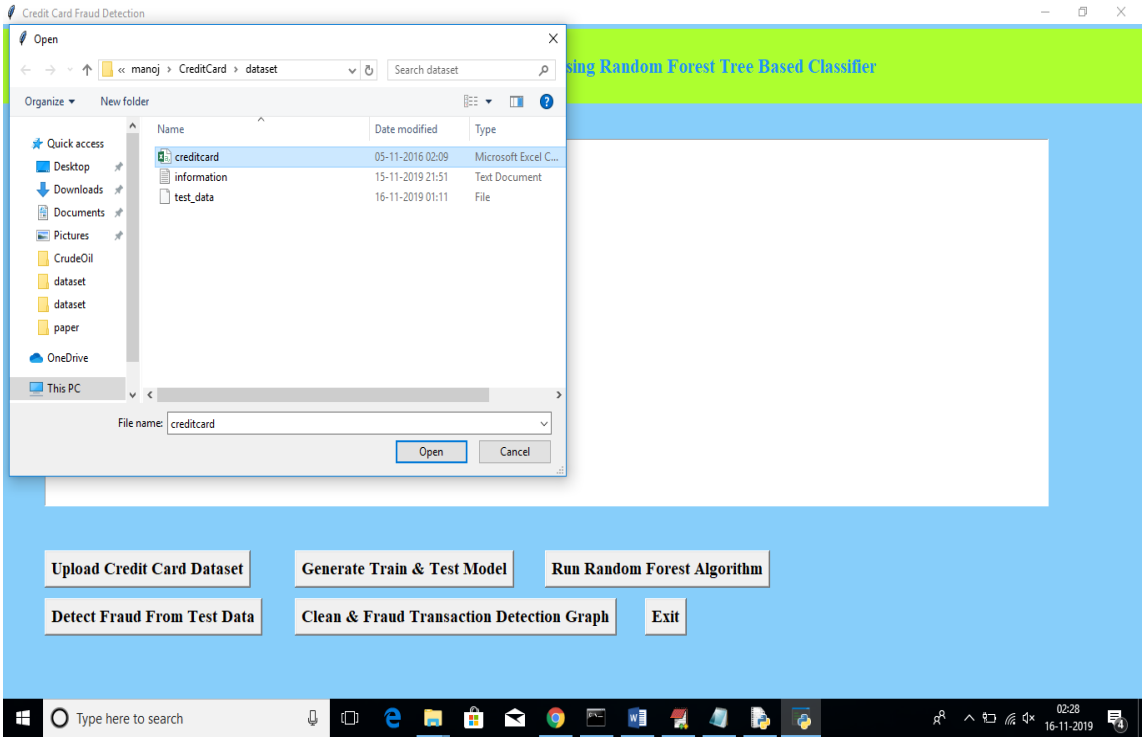

Fig 19: After uploading dataset will get below screen.

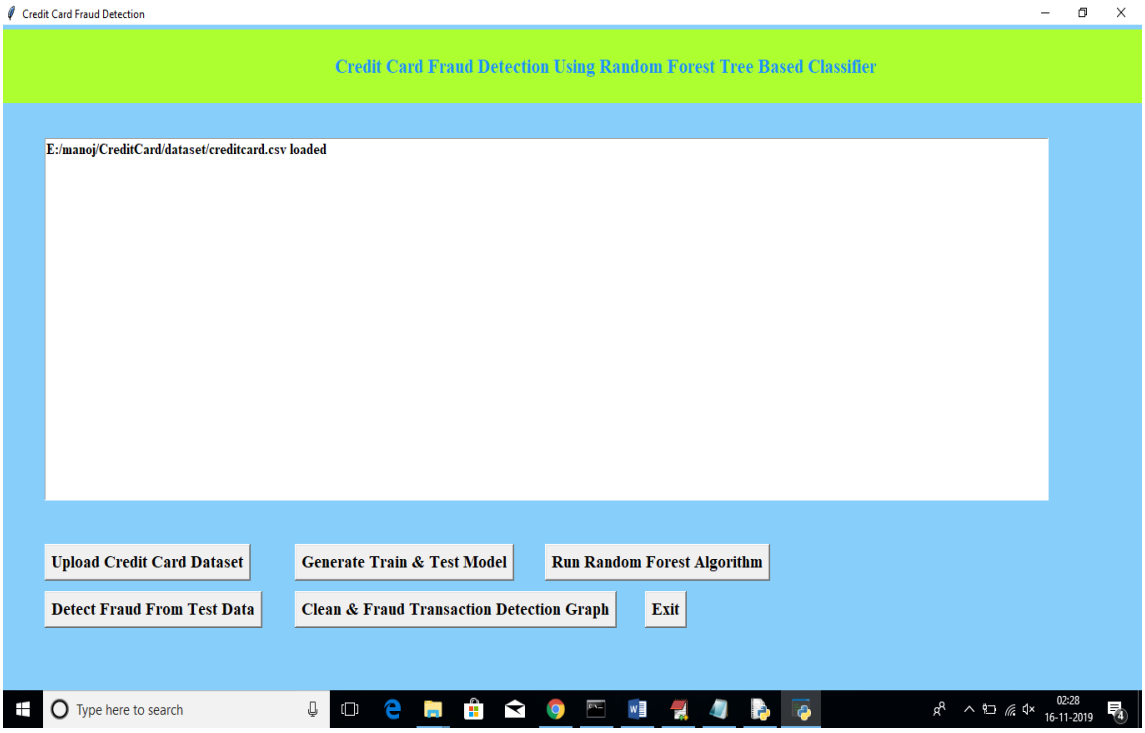

Fig 20: Now click on 'Generate Train & Test Model' to generate training model for Random Forest Classifier.

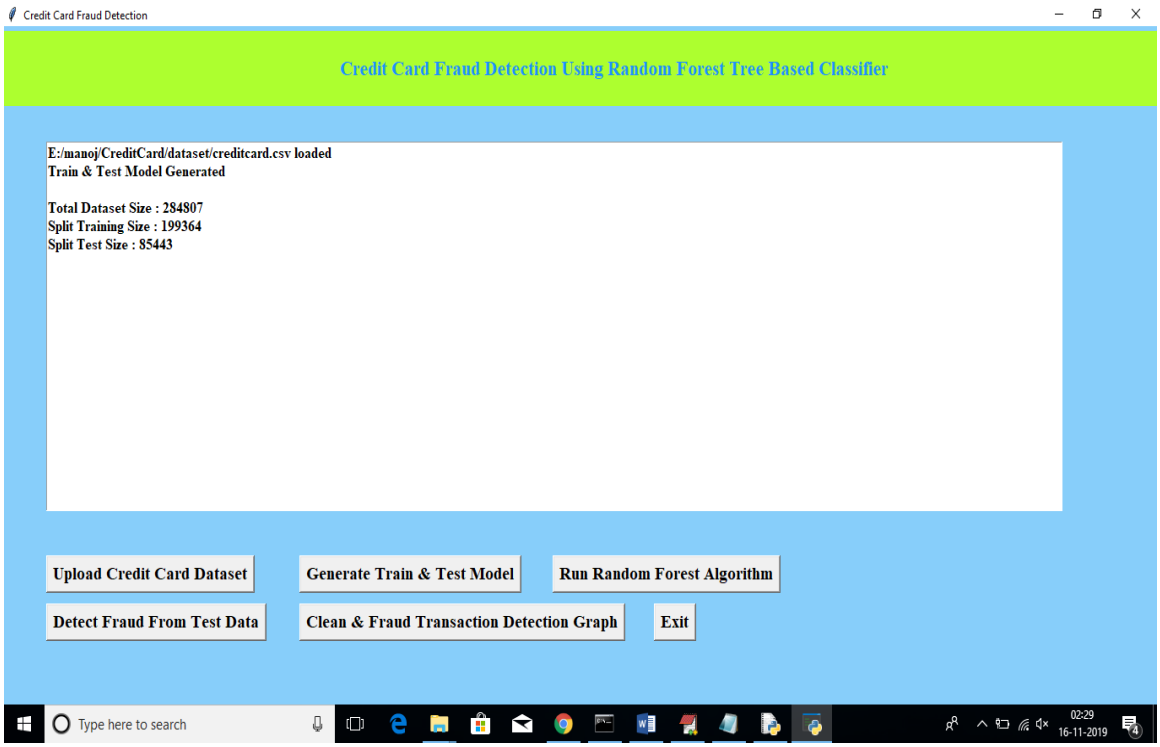

Fig 21: we can see total records available in dataset . Now click on "Run Random Forest Algorithm' button to generate Random Forest model on train and test data.

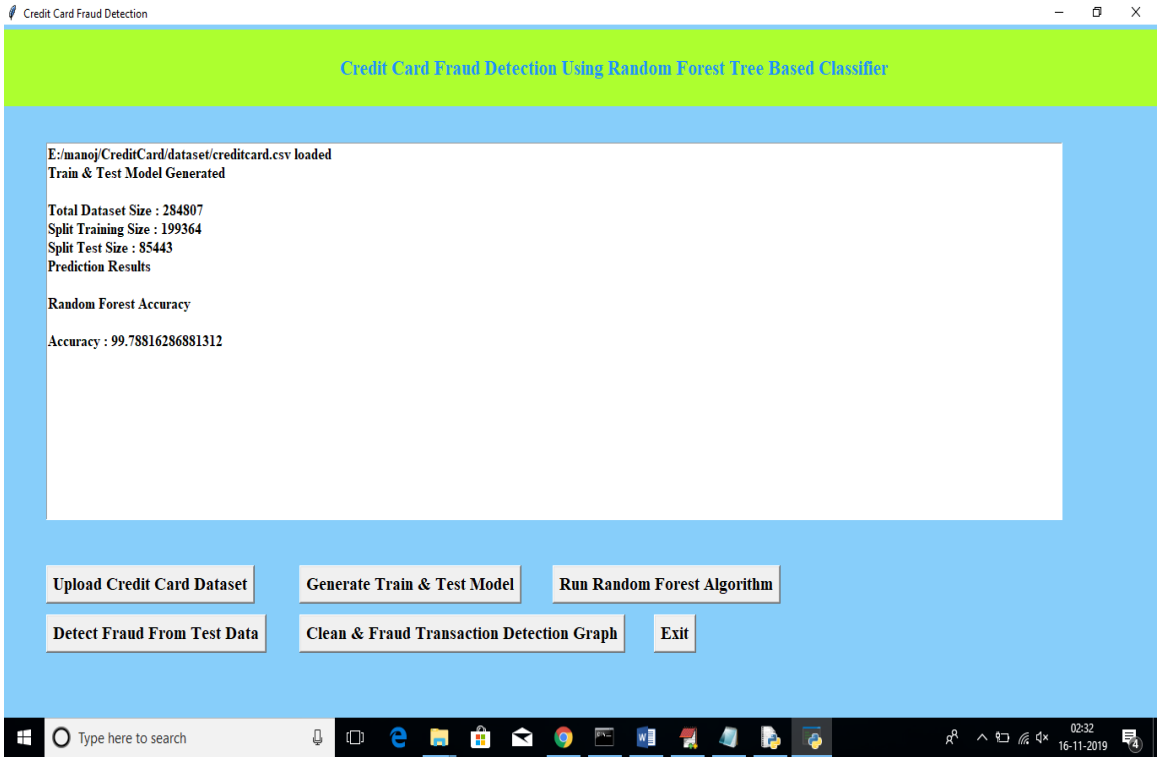

Fig 22: Random Forest generate 99.78% percent accuracy . Now click on 'Detect Fraud From Test Data' button to upload test data and to predict whether test data contains normal or fraud transaction.

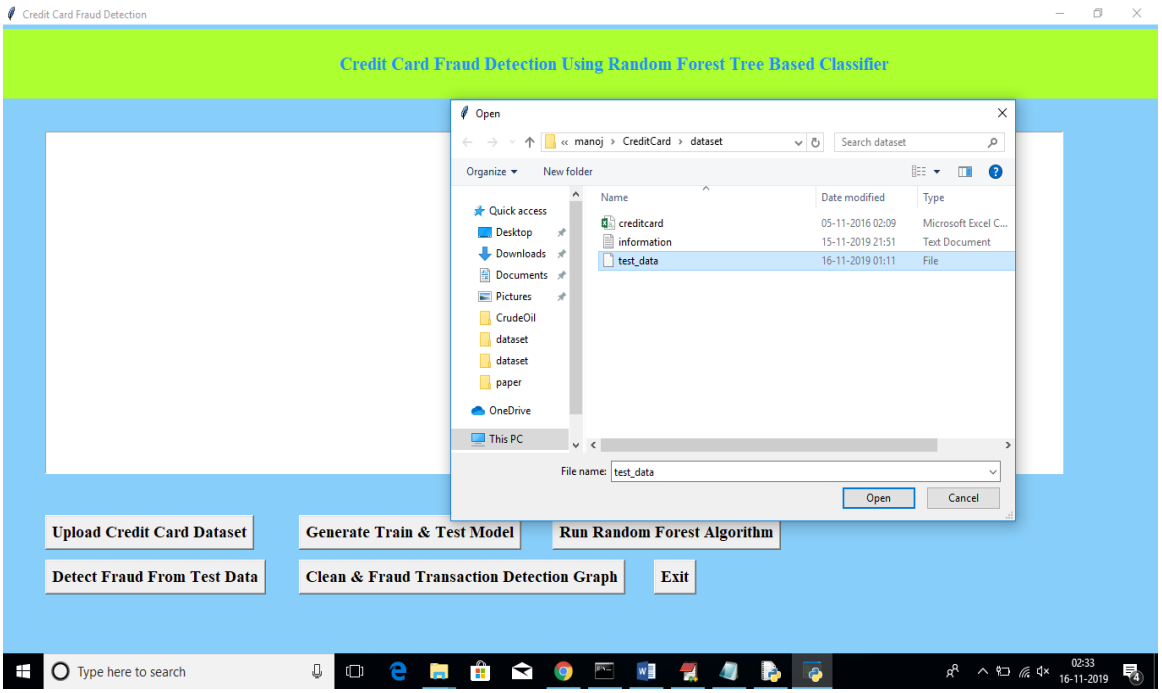

Fig 23: uploading test dataset.

Fig 24: Information says whether transaction contains cleaned or fraud signatures.

Now click on ‘Clean & Fraud Transaction Detection Graph’ button to see total test transaction with clean and fraud signature in graphical format. See below screen.

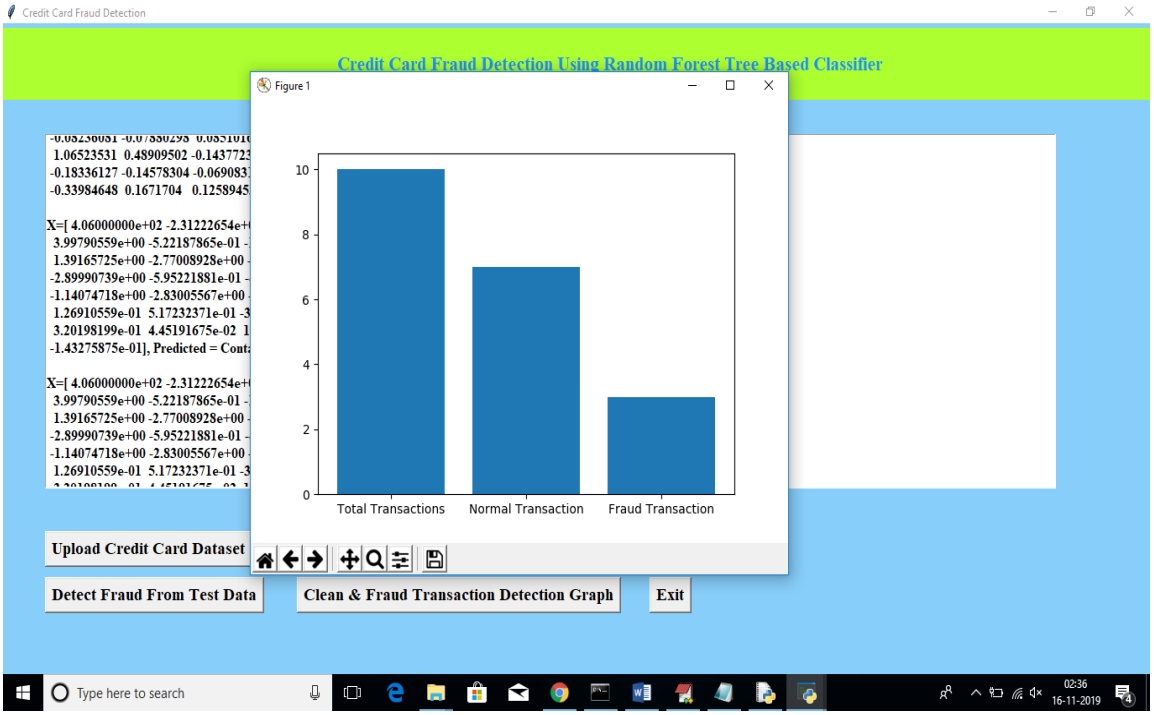

Fig 25: Final output shows number of fraud and normal transactions.

CHAPTER 4

RESULTS AND DISCUSSION

CHAPTER 4

RESULTS AND DISCUSSION

4.1 COMPARISON OF EXISTING SOLUTIONS

In order to compare various techniques we calculate the true positive, true negative, false positive and false negative generated by a system or an algorithm and use these in quantitative measurements to evaluate and compare performance of different systems. True Positive (TP) is number of transactions that were fraudulent and were also classified as fraudulent by the system. True Negative (TN) is number of transactions that were legitimate and were also classified as legitimate. False Positive (FP) is number of transactions that were legitimate but were wrongly classified as fraudulent transactions. False Negative (FN) is number of transactions that were fraudulent but were wrongly classified as legitimate transactions by the system.

4.2 DATA COLLECTION AND PERFORMANCE METRICS

Using above 'CreditCardFraud.csv' file we will train Random Forest algorithm and then we will upload test data file and this test data will be applied on Random Forest train model to predict whether test data contains normal or fraud transaction signatures. When we upload test data then it will contains only transaction data no class label will be there application will predict and give the result. See below test data file

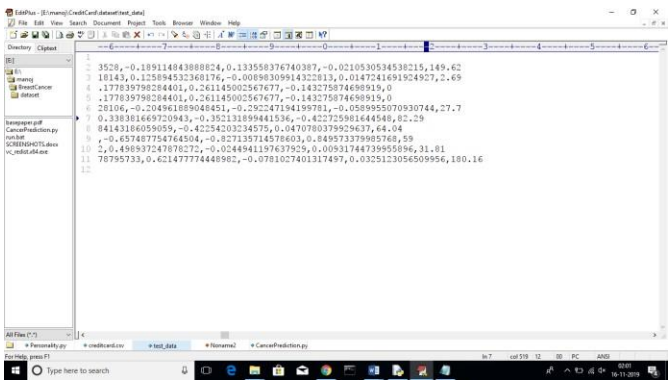

Fig 26: Data set

CHAPTER 5

CONCLUSION

CHAPTER 5
CONCLUSION

5.1 CONCLUSION AND FUTURE SCOPE

The Random Forest algorithm will perform better with a larger number of training data, but speed during testing and application will suffer. Application of more pre-processing techniques would also help. The SVM algorithm still suffers from the imbalanced dataset problem and requires more preprocessing to give better results at the results shown by SVM is great but it could have been better if more preprocessing have been done on the data.

Random Forest Algorithm in credit card fraud detection system and the final optimization results indicates the optimal accuracy for Random Forest Algorithm is 98.6%. Although random forest obtains good results on given data set, there are still some problems such as imbalanced data. Our future work will focus on solving these problems.

FUTURE SCOPE

It is evident from the above review that several machine learning algorithms are used to detect fraud, but the findings are not satisfactory. As a result, we'd like to use deep learning algorithms to reliably detect credit card fraud.

CHAPTER 6

REFERENCES

CHAPTER 6

REFERENCES

[1] J. T. Quah and M. Sriganesh, “Real-time credit card fraud detection using Computational intelligence,” Expert Syst. Appl., vol. 35, no. 4, pp. 1721–1732, 2008.

[2] H. He and E. A. Garcia, “Learning from imbalanced data,” IEEE Trans. Knowl. Data Eng., vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[3] D. Sánchez, M. A. Vila, L. Cerda, and J. M. Serrano, “Association rules applied to credit card fraud detection,” Expert Syst. Appl., vol. 36, no. 2, pp. 3630– 3640, 2009.

[4] M. Krivko, “A hybrid model for plastic card fraud detection systems,” Expert Syst. Appl., vol. 37, no. 8, pp. 6070–6076, 2010.

[5] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, “Data mining for credit card fraud: A comparative study,” Decision Support Syst., vol. 50, no. 3, pp. 602–613, 2011.

[6] R. Elwell and R. Polikar, “Incremental learning of concept drift in non stationary environments,” Trans. Neural Netw., vol. 22, no. 10, pp. 1517–1531, 2011.

[7] S. Jha, M. Guillen, and J. C. Westland, “Employing transaction aggregation strategy to detect credit card fraud,” Expert Syst. Appl., vol. 39, no. 16, pp. 12650–12657, 2012.

[8] C. Alippi, G. Boracchi, and M. Roveri, “Just-in-time classifiers for recurrent concepts,” IEEE Trans. Neural Netw. Learn. Syst., vol. 24, no. 4, pp. 620–634, Apr. 2013.

[9] M. Carminati, R. Caron, F. Maggi, I. Epifani, and S. Zanero, BankSealer: A Decision Support System for Online Banking Fraud Analysis and Investigation, Berlin, Germany: Springer, 2014, pp. 380–394.

[10] C. Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, “Detecting credit card fraud using periodic features,” in Proc. 14th Int. Conf. Mach. Learn. Appl., Dec. 2015, pp. 208–213.

[11] Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, “Credit card fraud detection and concept-drift adaptation with delayed supervised information,” in Proc. Int.

[12] Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Cali-brating probability with undersampling for unbalanced classification," in Proc. IEEE Symp. Ser. Computat. Intell., Dec. 2015, pp. 159–166.

[13] N. Mahmoudi and E. Duman, "Detecting credit card fraud by modified fisher discriminant analysis," Expert Syst. Appl., vol. 42, no. 5,pp. 2510–2516, 2015.

[14] Alippi, G. Boracchi, and M. Roveri, "Hierarchical change-detection tests," IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 2, pp. 246–258, Feb. 2016.

[15] Andrea Dal Pozzolo, GiacomoBoracchi, Olivier Caelen, CesareAlippi, Fellow, IEEE, and GianlucaBontempi, Senior Member, IEEE, "Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy", IEEE Transactions On Neural Networks And Learning Systems, vol 10, pp 216-23, 2018

[16] Sudhamathy G: Credit Risk Analysis and Prediction Modelling of Bank Loans Using R, vol. 8, no-5, pp. 1954-1966.

[17] LI Changjian, HU Peng: Credit Risk Assessment for ural Credit Cooperatives based on Improved Neural Network, International Conference on Smart Grid and Electrical Automation vol. 60, no. - 3, pp 227-230, 2017.

[18] Wei Sun, Chen-Guang Yang, Jian-Xun Qi: Credit Risk Assessment in Commercial Banks Based On Support Vector Machines, vol.6, pp 2430-2433, 2006.

[19].https://www.kaggle.com/code/hassanamin/credit-card-fraud-detection-using-random- forest/notebook

[20].https://www.tutorialspoint.com/what-is-a-neural-network-in-machine-learning

[21].https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm

[22].U. Fiore, A. De Santi's, F. Perla, P. Zanetti, F. Palmieri, Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. Inf. Sci. 479, 448–455(2019)

[23].N. Carneiro, G. Figueira, M. Costa, A data mining based system for credit-card fraud detectionin e-tail. Decis. Supp. Syst. 95, 91–101 (2017)

[24].A.C. Bahnsen, D. Aouada, A. Stojanovic, B. Ottersten, Feature engineering strategies for creditcard fraud detection. Exp. Syst. Appl. 51, 134–142 (2016)

[25].M. Zareapoor, P. Shamsolmoali, Application of credit card fraud detection: based on baggingand ensemble classifier. Proc. Comput. Sci. 48, 679–685 (2015)

GitHub Link

https://github.com/nivas-6052/Credit-Card-Fraud-Detection-Using-Random-Forest-Cart-Algorithm/upload