**ABSTRACT**

The project is mainly focussed on credit card fraud detection in real world. A phenomenal growth in the number of credit card transactions, has recently led to a considerable rise in fraudulent activities. The purpose is to obtain goods without paying, or to obtain unauthorized funds from an account. Implementation of efficient fraud detection systems has become imperative for all credit card issuing banks to minimize their losses. One of the most crucial challenges in making the business is that neither the card nor the cardholder needs to be present when the purchase is being made. This makes it impossible for the merchant to verify whether the customer making a purchase is the authentic cardholder or not. With the proposed scheme, using random forest algorithm the accuracy of detecting the fraud can be improved. Classification process of random forest algorithm to analyse data set and user current dataset. Finally optimize the accuracy of the result data. The performance of the techniques is evaluated based on accuracy, sensitivity, and specificity, and precision. Then processing of some of the attributes provided identifies the fraud detection and provides the graphical model visualization. The performance of the techniques is evaluated based on accuracy, sensitivity, specificity, and precision.

**TABLE OF CONTENTS**

**ABSTRACT I**

**INDEX WITH PAGE NO II**

**LIST OF FIGURES V**

**LIST OF SCREENSHOTS VI**

**INDEX NAME PAGE NO**

**1. INTRODUCTION 1**

1.1 CREDIT CARD FRAUD

**2. SYSTEM ANALYSIS 3**

2.1 EXISTING SYSTEM

2.1.1 DISADVANTAGES

2.2 PROPOSED SYSTEM

2.2.1 ADVANTAGES

**3. REQUIREMENTS SPECIFICATIONS 4**

3.1 HARDWARE REQUIREMENTS

3.2 SOFTWARE REQUIREMENTS

3.2.1 DATASET USED

**4. FEASIBILITY STUDY 5**

4.1 TECHNICAL FEASIBILITY

4.2 ECONOMICAL FEASIBILITY

4.3 OPERATIONAL FEASIBILITY

**5. SYSTEM ARCHITECTURE 6**

5.1 ARCHITECTURE DIAGRAM 5.2 ER DIAGRAM

5.3 CLASS DIAGRAM

5.4 USECASE DIAGRAM

5.5 SEQUENCE DIAGRAM

5.6 DATAFLOW DIAGRAM

**6. SYSTEM MODULES 12**

6.1 MODULE 1

6.2 MODULE 2

6.3 MODULE 3

6.4 MODULE 4

**7.ALGORITHM UTILIZED 15**

7.1 RANDOM FOREST

7.1.1 BASIC MODEL OF RF CLASSIFIER

7.1.2 WORKING OF RFA

7.1.3 CONFUSION MATRIX

7.1.4 STEPS OF RFA

7.2 ADVANTAGES OF RFA

**8. TESTING 21**

**8.1 TYPES OF TESTING**

8.1.1 WHITE BOX TESTING

8.1.2 BLACK BOX TESTING

8.1.3 UNIT TESTING

8.1.4 INTEGRATION TESTING

8.1.5 VALIDATION TESTING

8.1.6 OUTPUT TESTING

8.1.7 USER ACCEPTANCE TESTING

8.1.7a ALPHA TESTING

8.1.7b BETA TESTING

**8.2 TESTING USED IN THIS PROJECT**

**9. OUTPUTS 28**

**10.COMPARISION OF RFA 32**

**11. CONCLUSION 33**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIG.NO** | **PARTICULARS** | **PAGE NO.** |
| 1 | Architecture Diagram | 6 |
| 2 | ER Diagram | 7 |
| 3 | Class Diagram | 8 |
| 4 | Use Case Diagram | 9 |
| 5 | Sequence Diagram | 10 |
| 6 | Data Flow Diagram | 11 |
| 7 | Modules Diagram | 14 |
| 8 | Random Forest Classifier | 15 |
| 9 | White Box Testing | 22 |
| 10 | Black Box Testing | 23 |
| 11 | Unit Testing | 23 |
| 12 | Integration Testing | 24 |
| 13 | Validation Testing | 25 |
| 14 | User Acceptance Testing | 26 |

**LIST OF SCREENSHOTS**

|  |  |  |
| --- | --- | --- |
| **SCREENSHOT NO.** | **PARTICULARS** | **PAGE NO.** |
| 1 | Understanding Dataset | 28 |
| 2 | Describing Dataset | 28 |
| 3 | Imbalanced Data | 28 |
| 4 | Fraudulent Transaction | 29 |
| 5 | Normal Transaction | 29 |
| 6 | X and Y values | 29 |
| 7 | Parameters | 30 |
| 8 | Correlation Matrix | 30 |
| 9 | Confusion Matrix | 31 |
| 10 | Comparison of RFA | 32 |

**1. INTRODUCTION**

**1.1 WHAT IS CREDIT CARD FRAUD?**

Credit card fraud can be divided into 2 types: inner card fraud and external card fraud. Inner card fraud intends to defraud the cash. Usually it is the collusion between merchants and cardholders, using false transactions to defraud banks cash. External card fraud is mainly embodied at using the stolen, fake or counterfeit credit card to consume, or using cards to get cash in disguised forms, such as buying the expensive, small volume commodities or the commodities that can easily be changed into cash. This paper is mainly devoted to the investigation of the external card fraud, which accounts for the majority of credit card frauds. In general, a fraud is defined as a crime committed with intention to damage a person and is also a violation. Fraud may be committed for various reasons: for entertainment, to exploit a business / an organization, to take revenge, to cause financial loss, to damage identity etc. Also, there are several types of frauds: bankruptcy frauds, identity thefts, health frauds, religious frauds, credit card frauds, insurance frauds, forgery, tax frauds and many more. Here considering only the credit card frauds, they can be of two kinds: a) offline credit card frauds and b) online credit card frauds. Offline credit card frauds are those where an individual’s credit card is lost or stolen. If any attacker or hacker, hack the details and use it to commit illegal actions is referred as online frauds. With the rapidly developing technology, usage of internet is drastically increasing. Substantially, this is leading to many credit-card fraudulent activities.

Under the vulnerable situations, the users must limit amount of information they are sharing to reduce the exposure and chances of getting attacked by a hacker.

Certain steps can be followed to protect themselves in case of vulnerabilities and insecurities regarding their identity: firstly, the users must take considerable care and attention to all their documents consisting of their identity, accounts and other sensitive information. Information sharing between friends and relatives is other reason from research that causes credit card frauds because of current lifestyle conditions.

So, this has to be avoided and proper precaution must be taken. While the old documents or devices are disposed or discarded, users must make sure that there is no personal information that can be possibly revealed or tracked by anyone. Bank account information like transactions and balances should be regularly checked by the users to avoid money loss that could be caused by attackers. Secondly, the use of computer systems and internet must be developed among the people. For example, just by installing anti-virus software is not sufficient enough to not get attacked by any malware. Without proper security across the network, the information over the internet must not be shared. Users should not access for their personal information from public wireless networks or others accounts and must carry a strong WPA key to their Wi-Fi access points. Thus, users need to look for security policies and keep their account protected from possible threats. Basic security precautions like effective control over cookies, anonymous browsing, reducing computer information with-holdings, considering network address protection, using strong encrypting tools and changing passwords of all accounts often and also using passwords that are hard-to-crack can help the users from being attacked. Not just the users that need to take care of their personal credit card information; the respective banks and organizations that are issuing credit cards must also take initiative actions and work collectively considering two important things: firstly, they must bring out strict policies over public to secure genuine credit card holders and their privacy. And secondly, they must organize awareness about the secure way of using credit-cards and attacks possible among the public by prevention campaigns from several smaller targets to bigger population. If there are people who are already attacked by fraud persons, then those victims should take immediate actions related to situation. If a credit card related to a bank account of a person is hacked and fraud is detected by the victim, in order to protect his account from further financial destructions, he/she must contact the bank or bank’s anti-fraud centre or credit card agent, requesting to block their card and monitor the details. Also, then should report and register a complaint with the police department for investigation. So, it is definitely a challenging task to battle against the damage caused for one’s loss. The information that is shared over the internet is never completely deleted. This is also a possible reason for cyber attacks. Hence it is very important to contact the sources of information and request them to delete the information as after effects of any fraud is difficult to recover and time consuming for any victim.

**2. SYSTEM ANALYSIS**

**2.1 Existing System:**

In existing System, a research about a case study involving credit card fraud detection, where data normalization is applied before Cluster Analysis and with results obtained from the use of Cluster Analysis and Artificial Neural Networks on fraud detection has shown that by clustering attributes neuronal inputs can be minimized. And promising results can be obtained by using normalized data and data should be MLP trained. This research was based on unsupervised learning. Significance of this paper was to find new methods for fraud detection and to increase the accuracy of results. The data set for this paper is based on real life transactional data by a large European company and personal details in data is kept confidential. Accuracy of an algorithm is around 50%. Significance of this paper was to find an algorithm and to reduce the cost measure. The result obtained was by 23% and the algorithm they found was Bayes minimum risk.

**2.1.1 Disadvantages:**

1. In this paper a new collative comparison measure that reasonably represents the gains and losses due to fraud detection is proposed.

2. A cost sensitive method which is based on Bayes minimum risk is presented using the proposed cost measure.

**2.2 PROPOSED SCHEME:**

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 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.

**2.2.1 ADVANTAGES OF PROPOSED SYSTEM:**

1. Random forest ranks the importance of variables in a regression or classification problem in a natural way can be done by Random Forest.

2. 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. REQUIREMENT SPECIFICATIONS**

The requirements specification is a technical specification of requirements for the software products. It is the first step in the requirements analysis process it lists the requirements of a particular software system including functional, performance and security requirements. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals.

**3.1 HARDWARE REQUIREMENTS**

* Processor - Intel
* RAM - 4 Gb
* Hard Disk - 260 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse

**3.2 SOFTWARE REQUIREMENTS**

* Python
* Anaconda
* OS - Windows 7, 8 and 10 (32 and 64 bit)
* Dataset used Credit.csv

**3.2.1 Details about credit.csv dataset**

The datasets contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions. It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise

**4. FEASIBILITY STUDY**

**4.1 TECHNICAL FEASIBILITY:**

It is evident that necessary hardware software is available for development and implementation of proposed system.

It uses Anaconda.

**4.2 ECONOMICAL FEASIBILITY:**

The cost for the proposed system is comparatively less to other existing software’s.

**4.3 OPERATIONAL FEASIBILITY:**

In this project, it requires to configure the necessary software to work on the software.

**5. SYSTEM ARCHITECTURE**

**5.1 ARCHITECTURE DAIGRAM:**

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 factors 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 test and train dataset.

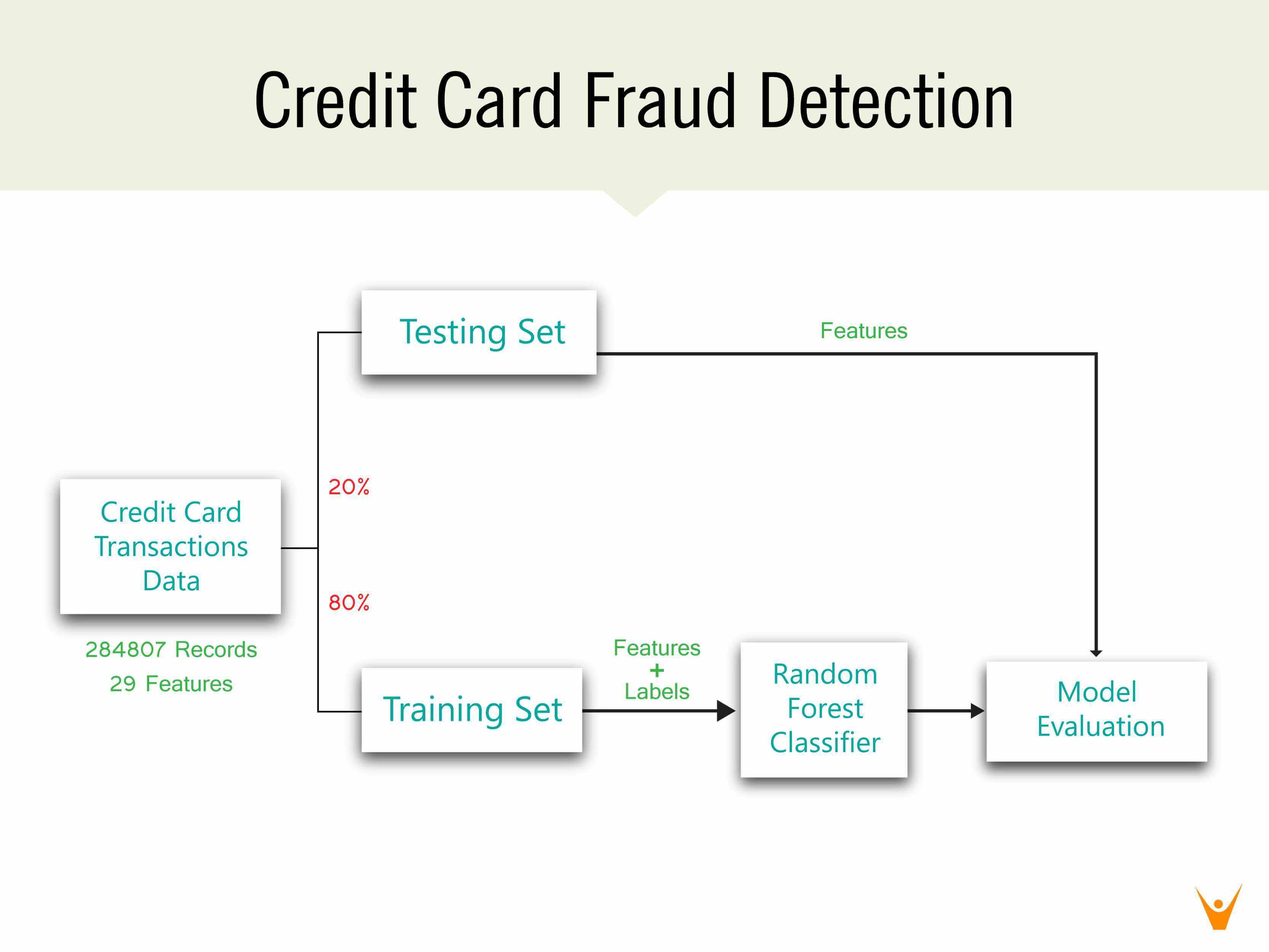


FIG-1

**5.2 ER DIAGRAM:**

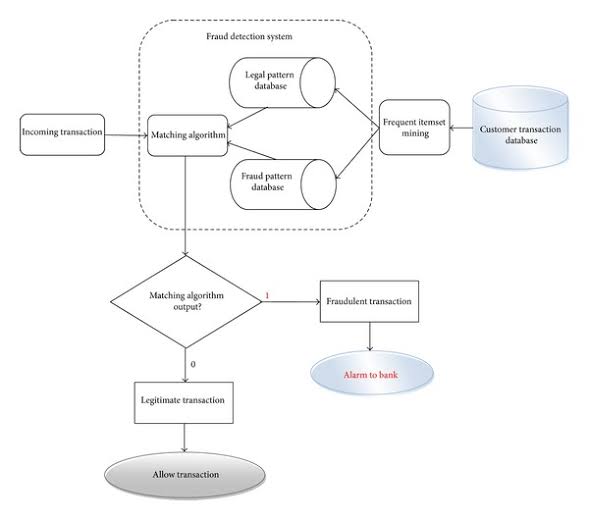


FIG-2

**5.3 CLASS DIAGRAM:**

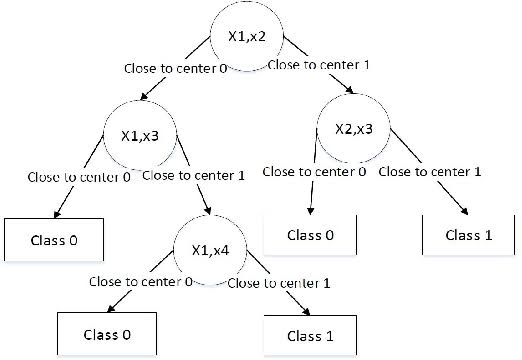


FIG-3

**5.4 USE CASE DIAGRAM:**

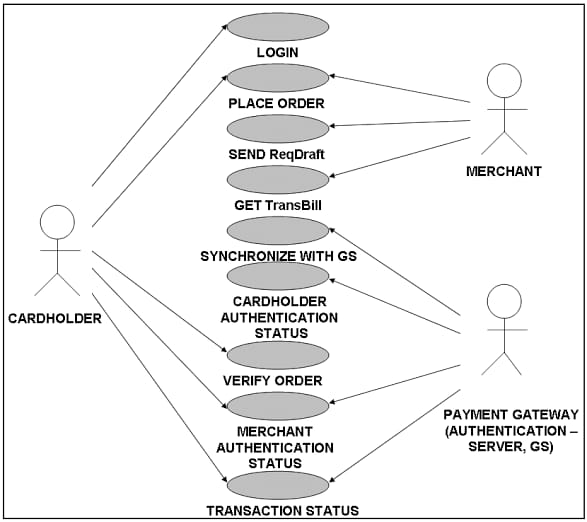


FIG-4

**5.5 SEQUENCE DIAGRAM:**

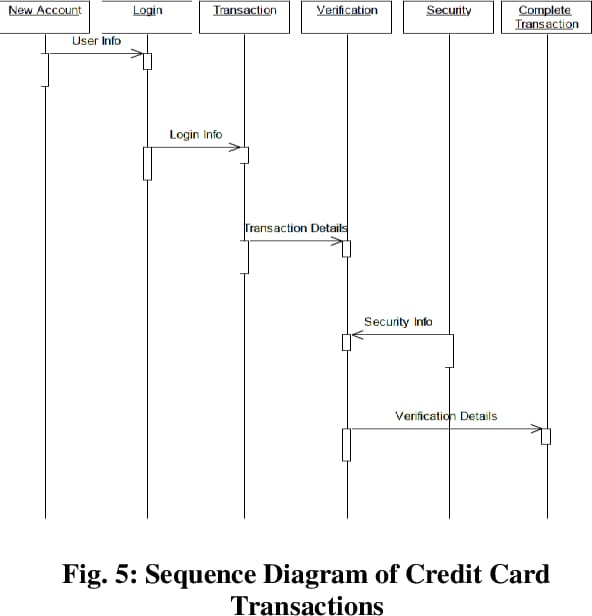


FIG-5

**5.6 DATAFLOW DIAGRAM(DFD):**

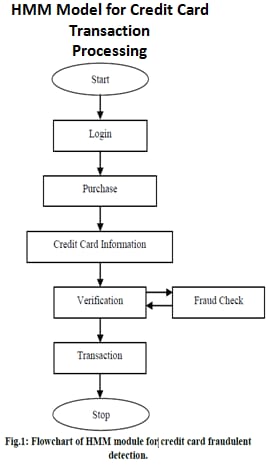


FIG-6

**6. SYSTEM MODULES**

**6.1 MODULE 1: DATA COLLECTION:**

Data used in this paper is a set of product reviews collected from credit card transactions records. This step is concerned with selecting the subset of all available data that you will be working with. ML problems start with data preferably, lots of data (examples or observations) for which you already know the target answers. Data for which you already know the target answer is called labelled data.

**6.2 MODULE 2: DATA PRE-PROCESSING:**

Organize your selected data by formatting, cleaning and sampling from it.

Three common data pre-processing steps are:

Formatting: The data you have selected may not be in a format that is suitable for you to work with. The data may be in a relational database and you would like it in a flat file, or the data may be in a proprietary file format and you would like it in a relational database or a text file.

Cleaning: Cleaning data is the removal or fixing of missing data. There may be data instances that are incomplete and do not carry the data you believe you need to address the problem. These instances may need to be removed. Additionally, there may be sensitive information in some of the attributes and these attributes may need to be removed from the data entirely.

Sampling: There may be far more selected data available than you need to work with. More data can result in much longer running times for algorithms and larger computational and memory requirements. You can take a smaller representative sample of the selected data that may be much faster for exploring and prototyping solutions before considering the whole dataset.

**6.3 MODULE 3: FEATURE EXTRATION:**

Next thing is to do Feature extraction is an attribute reduction process. Unlike feature selection, which ranks the existing attributes according to their predictive significance, feature extraction actually transforms the attributes. The transformed attributes, or features, are linear combinations of the original attributes. Finally, our models are trained using Classifier algorithm. We use classify module on Natural Language Toolkit library on Python. We use the labelled dataset gathered. The rest of our labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify pre-processed data. The chosen classifiers were Random forest. These algorithms are very popular in text classification tasks.

**6.4 MODULE 4: Evaluation Model:**

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and over fitted models. There are two methods of evaluating models in data science, Hold-Out and Cross-Validation. To avoid over fitting, both methods use a test set (not seen by the model) to evaluate model performance. Performance of each classification model is estimated base on its averaged. The result will be in the visualized form. Representation of classified data in the form of graphs. Accuracy is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

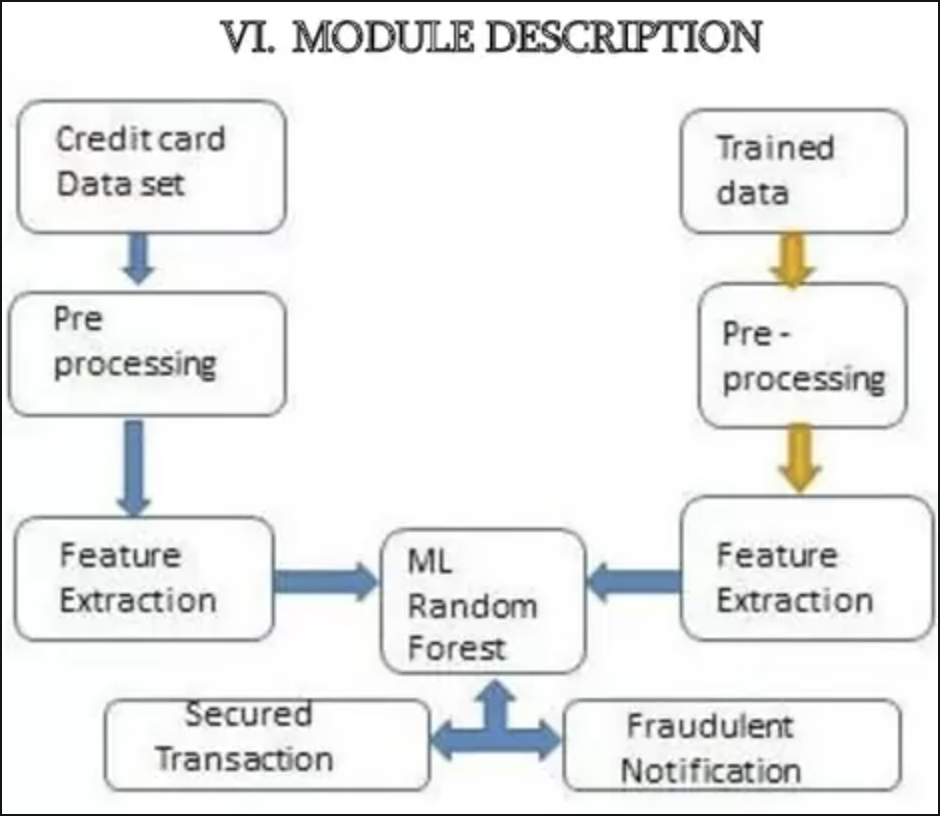


FIG-7

**7. Algorithm Utilized**

**7.1 Random Forest**:

Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name “Random Forest”. The random forest algorithm can be used for both regression and classification tasks.

**7.1.1 BASIC MODEL OF RANDOM FOREST CLASSIFIER:**

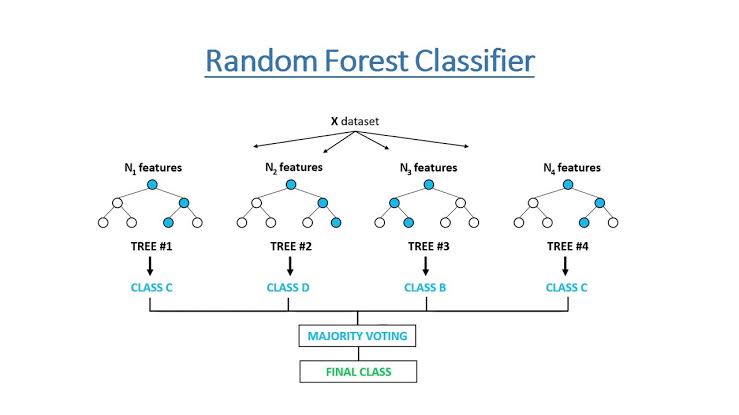


FIG-8

**7.1.2 WORKING OF RANDOM FOREST ALGORITHM:**

The following are the basic steps involved in performing the random forest algorithm

1. Pick N random records from the dataset.

2. Build a decision tree based on these N records.

3. Choose the number of trees you want in your algorithm and repeat steps 1 and 2.

4. For classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

5.Generate Confusion Matrix.

**7.1.3 What is Confusion Matrix?**

A confusion matrix is basically a summary of prediction results or a table which is used to describe the performance of the classifier on a set of test data where true values are known. It provides visualization of an algorithm’s performance and allows easy identification of classes. Thus, resulting in the computing of most performance measures by giving insights not only the errors being made by the classification model but also tells the type of errors being made. Trained Data and Testing Data is represented in a confusion matrix which portrays:

* **TP: True** Positive which denotes the real data where customers are subjected to fraud and are used for training and were accurately predicted.
* **TN: True** Negative denotes the data which was not predicted and doesn’t match with the data which was subjected to the fraud.
* **FP: False** Positive is predicted but there is no possibility of the data to be subjected to the fraud.
* **FN: False** Negative is not predicted but there is an actual possibility of the data who is subjected to fraud.

**7.1.4 Detailed Steps of credit card fraud detection system using random forest algorithm**:

**1.Importing all the necessary Libraries:**

# 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

**2.Loading the Data:**

data = pd.read\_csv("credit.csv")

**3.Understanding the Data:**

data.head()

**4. Describing the Data:**

print(data.shape)

print(data.describe())

**5. 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])))

**6.Print the amount details for Fraudulent Transaction**

print(“Amount details of the fraudulent transaction”)

fraud.Amount.describe()

**7.Print the amount details for Normal Transaction:**

print(“details of valid transaction”)

valid.Amount.describe()

**8.Plotting the Correlation Matrix:**

# Correlation matrix

corrmat = data.corr()

fig = plt.figure(figsize = (12, 9))

sns.heatmap(corrmat, vmax = .8, square = True)

plt.show()

**9.Separating the X and the Y values:**

# dividing the X and the Y from the dataset

X = data.drop(['Class'], axis = 1)

Y = data["Class"]

print(X.shape)

print(Y.shape)

# getting just the values for the sake of processing

# (its a numpy array with no columns)

xData = X.values

yData = Y.values

**10.Training and Testing Data Bifurcation:**

# Using Skicit-learn to split data into training and testing sets

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

# Split the data into training and testing sets

xTrain, xTest, yTrain, yTest = train\_test\_split(

xData, yData, test\_size = 0.2, random\_state = 42)

**11.Building a Random Forest Model using skicit learn:**

# Building the Random Forest Classifier (RANDOM FOREST)

from sklearn.ensemble import RandomForestClassifier

# random forest model creation

rfc = RandomForestClassifier()

rfc.fit(xTrain, yTrain)

# predictions

yPred = rfc.predict(xTest)

**12.Building all kinds of evaluating parameters:**

# Evaluating the classifier

# printing every score of the classifier

# scoring in anything

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))

**13.Visulalizing the Confusion Matrix:**

# printing the confusion matrix

LABELS = ['Normal', 'Fraud']

conf\_matrix = confusion\_matrix(yTest, yPred)

plt.figure(figsize =(12, 12))

sns.heatmap(conf\_matrix, xticklabels = LABELS,

yticklabels = LABELS, annot = True, fmt ="d");

plt.title("Confusion matrix")

plt.ylabel('True class')

plt.xlabel('Predicted class')

plt.show()

**7.2 ADVANTAGES OF USING RANDOM FOREST ALGORITHM**

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. The random forest algorithm also works well when data has missing values or it has not been scaled well.

**8. TESTING**

Testing is a process of executing a program with intent of finding an error.

Testing presents an interesting anomaly for the software engineering.

The goal of the software testing is to convince system developer and customers that the software is good enough for operational use.

Testing is a process intended to build confidence in the software.

Testing is a set of activities that can be planned in advance and conducted systematically.

Testing is a set of activities that can be planned in advance and conducted systematically.

Software testing is often referred to as verification & validation.

**8.1 Types of testing**

The various types of testing are:

* + - White Box Testing
    - Black Box Testing
    - Unit Testing
    - Integration Testing
    - Validation Testing
    - Output Testing
    - User Acceptance Testing

**8.1.1 White box testing**

It is also called as glass-box testing. It is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineer can derive test cases that: Guarantee that all independent parts within a module have been exercised at least once.

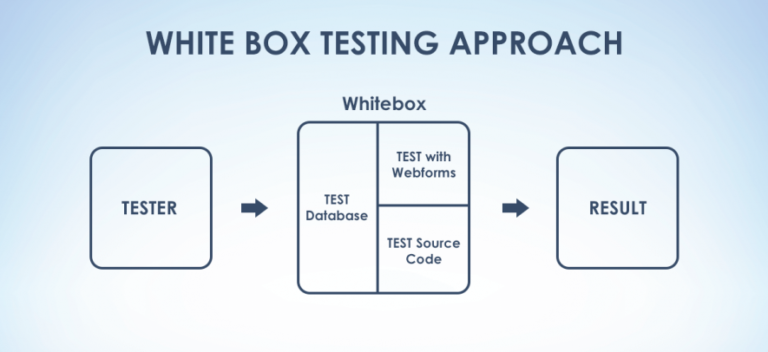


FIG-9

**8.1.2 Black box testing**

It’s also called as behavioral testing. It focuses on the functional requirements of the software. It is complementary approach that is likely to uncover a different class of errors than white box errors. A black box testing enables a software engineering to derive a set of input conditions that will fully exercise all functional requirements for a program.

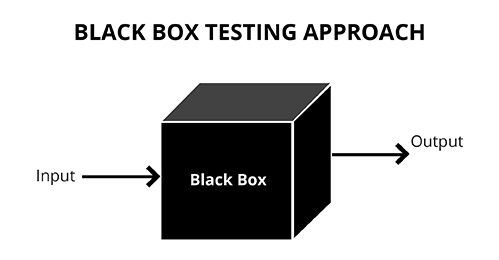


FIG-10

**8.1.3 Unit testing**

In this testing, we test each module individually and integrate with the overall system. Unit testing focuses verification efforts on the smallest unit of software design in the module. This is also known as module testing.

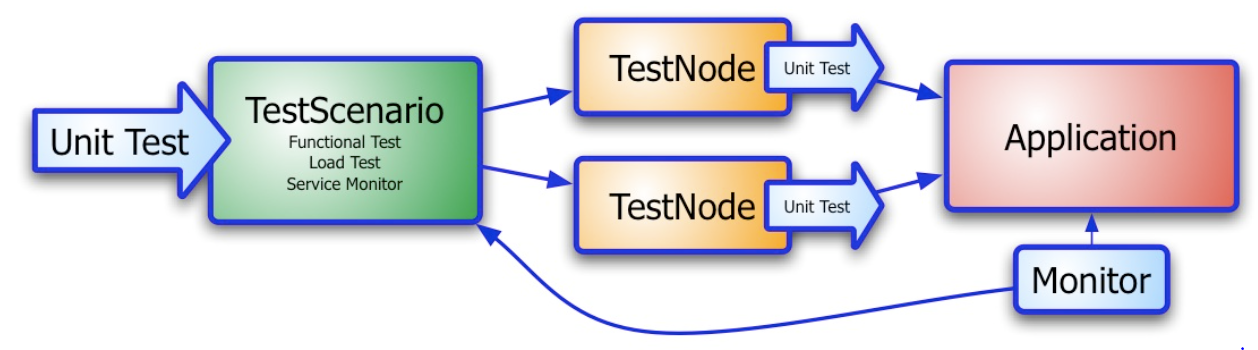


FIG-11

The module of the system is tested separately. This testing is carried out during programming stage itself. In this testing step each module is found to working satisfactorily as regard to the expected output from the module. There are some validation checks for fields also. It is very easy to find error debut in the system.

**8.1.4 Integration testing**

Data can be lost across an interface; one module can have an adverse effort on the other sub functions when combined may not produces the desired major functions.

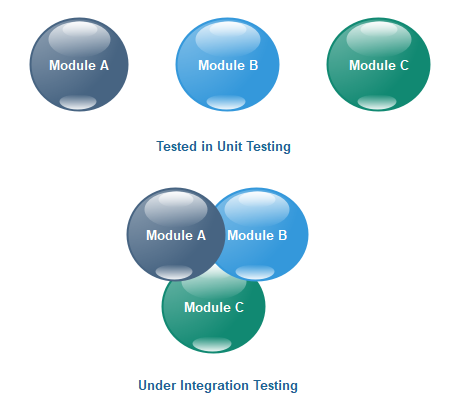


FIG-12

Integrated testing is the systematic testing for constructing the uncover errors within the interface. The testing was done with sample data. The Developed system has run successfully for this sample data. The need for integrated test is to find the overall system performance.

**8.1.5 Validation testing**

At the culmination of the black box testing, software is completely assembled as a package, interfacing errors have been uncovered and corrected and a final series of software tests. That is, validation tests begin, validation testing can be defined many ways but a simple definition is that validation succeeds when the software functions in manner that can be reasonably expected be the customer.

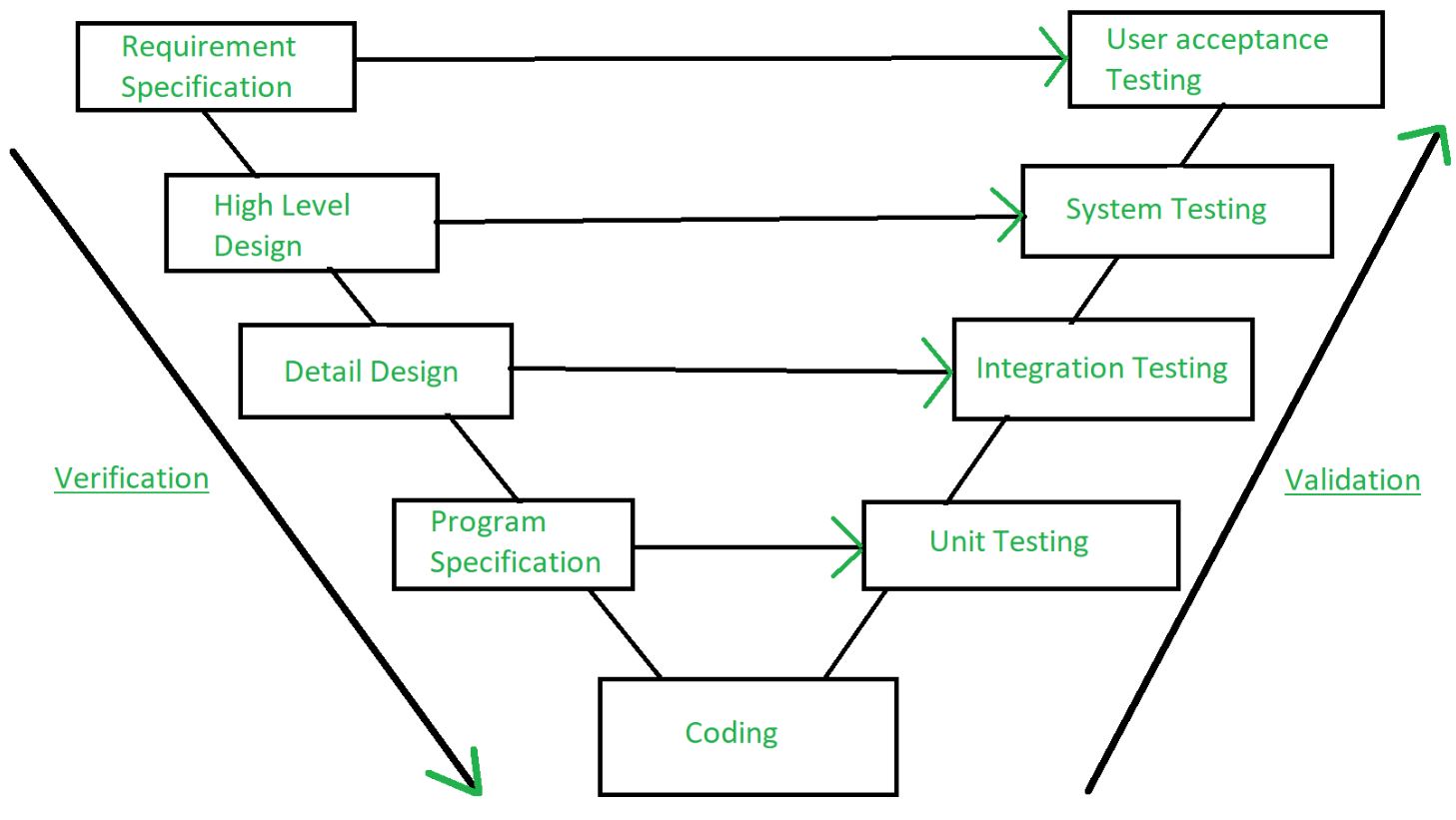


FIG-13

After validation test has been conducted one of the two possible conditions exists. The functions or performance characteristics confirm to specification and are accepted.

**8.1.6 Output testing**

After performance of the validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output in the specific format. Asking the user about the format required by system tests the output displayed or generated by the system under consideration.

Here the output format is considered the of screen display. The output format on the screen is found to be correct as the format was designed in the system phase according to the user need. For the hard copy also the output comes out as specified by the user. Hence the output testing does not result in any correction in the system.

**8.1.7 User acceptance testing**

Some of my friends were who tested this module suggested that this was really a user friendly application and giving good processing speed.

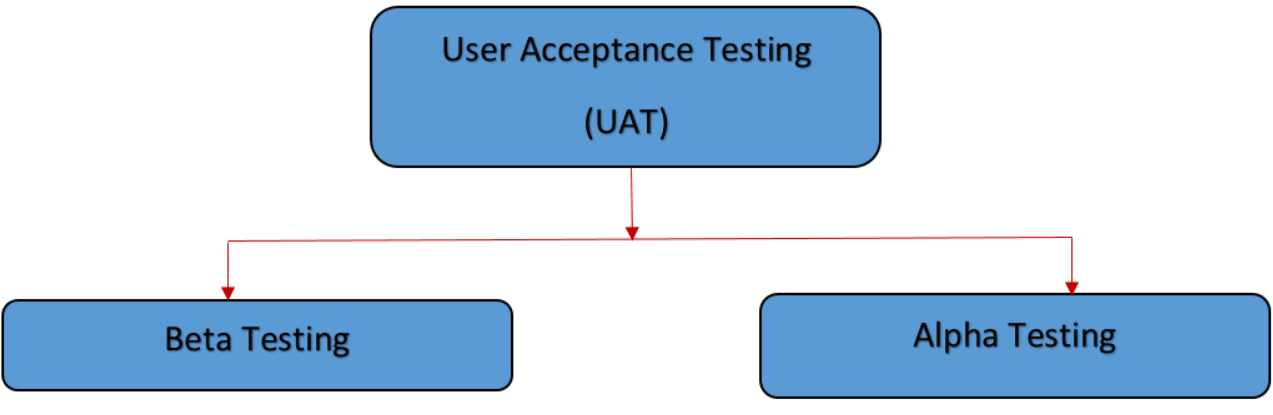


FIG-14

**8.1.7a Alpha Testing**

**Alpha Testing** is a type of acceptance testing; performed to identify all possible issues and bugs before releasing the final product to the end users. Alpha testing is carried out by the testers who are internal employees of the organization. The main goal is to identify the tasks that a typical user might perform and test them.

To put it as simple as possible, this kind of testing is called alpha only because it is done early on, near the end of the development of the software, and before beta testing. The main focus of alpha testing is to simulate real users by using a black box and white box techniques.

**8.1.7b Beta Testing**

**Beta Testing** is performed by "real users" of the software application in "real environment" and it can be considered as a form of external UAT It is the final test before shipping a product to the customers. Direct feedback from customers is a major advantage of Beta Testing. This testing helps to test products in customer's environment.

Beta version of the software is released to a limited number of end-users of the product to obtain feedback on the product quality. Beta testing reduces product failure risks and provides increased quality of the product through customer validation.

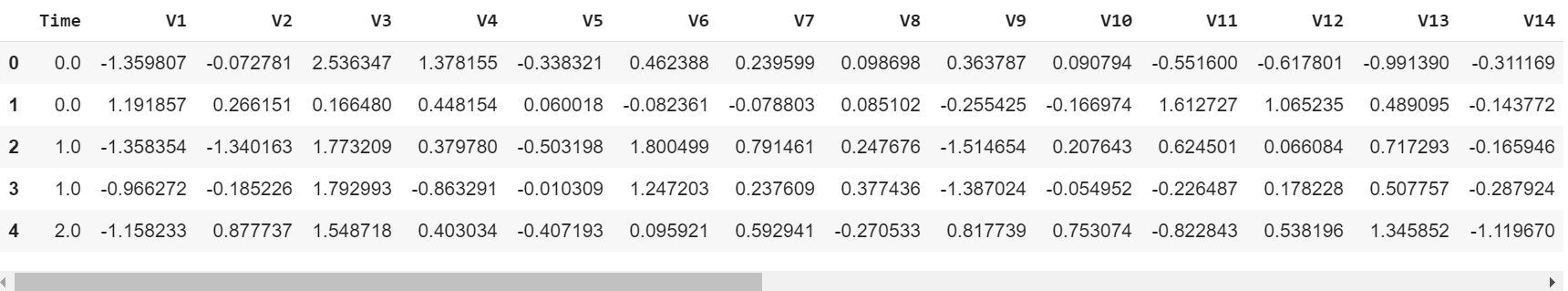
**8.2 Testing used in this project:**

* Unit Testing
* Integration Testing
* Validation Testing
* Output Testing
* User Acceptance Testing

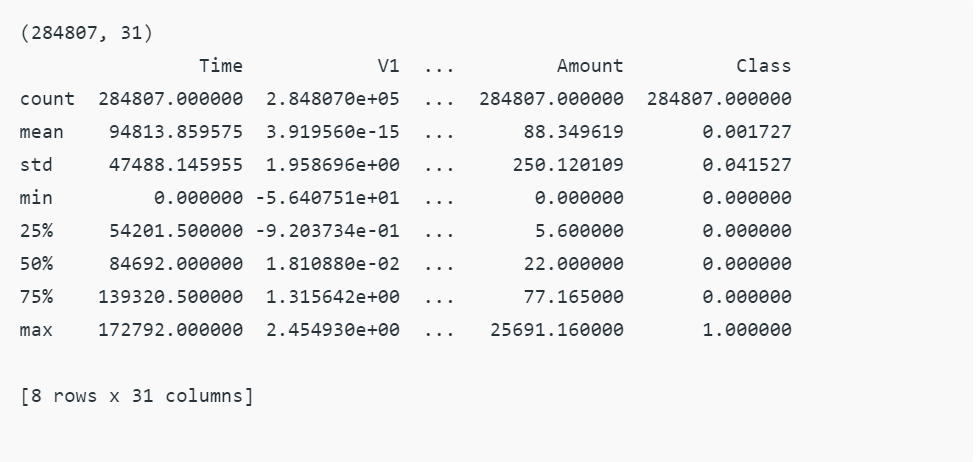
Alpha Testing

Beta Testing

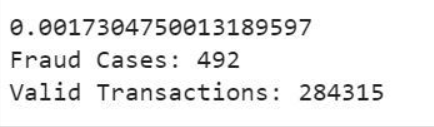
**9. OUTPUTS**



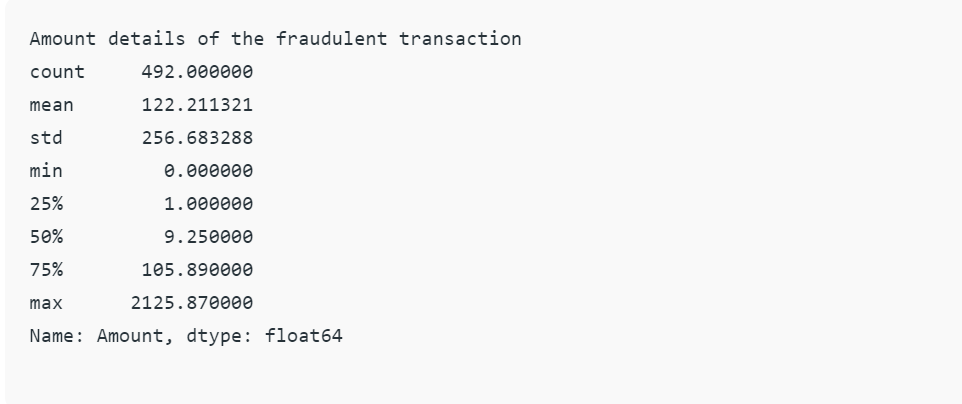
SS-1 (Understanding Dataset**)**



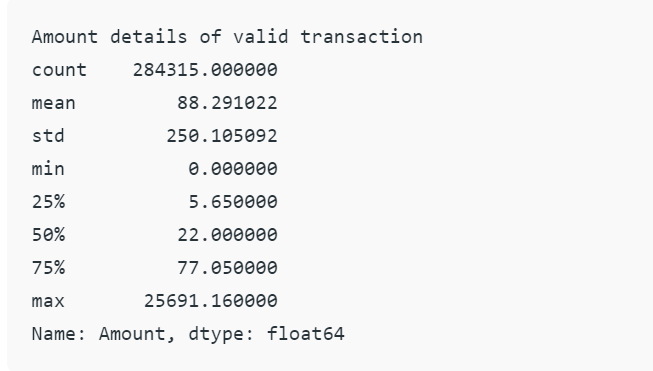
SS-2 (Describing Dataset)



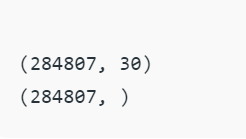
SS-3 (Imbalanced Data)



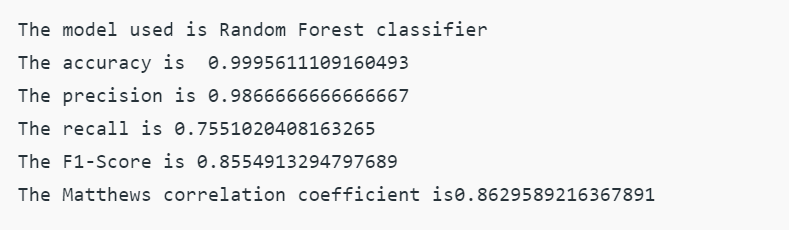
SS-4 (Fraudulent Transaction)



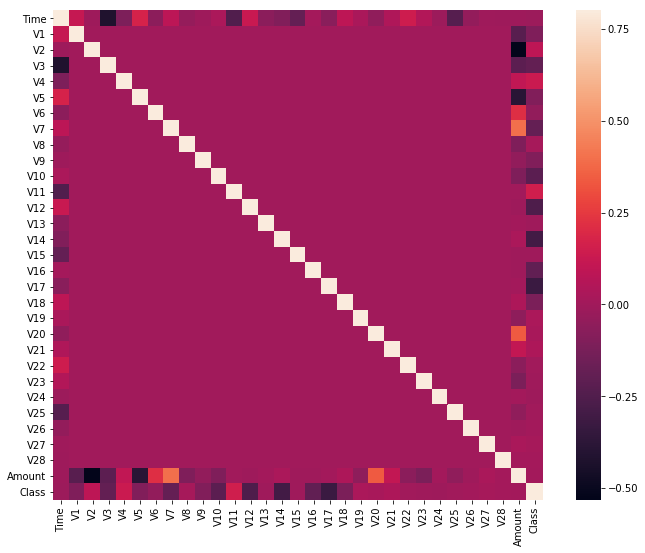
SS-5 (Normal Transaction)



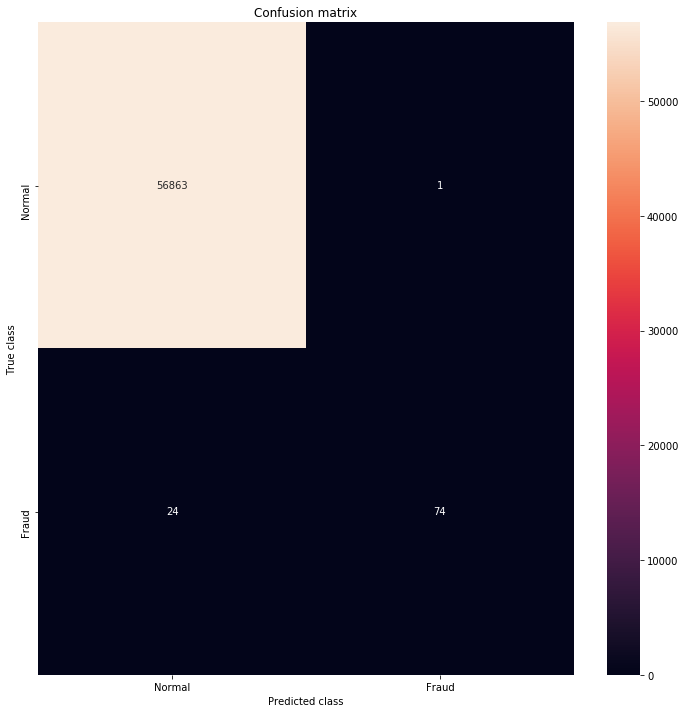
SS-6 (X and Y values)



SS-7 (Parameters)

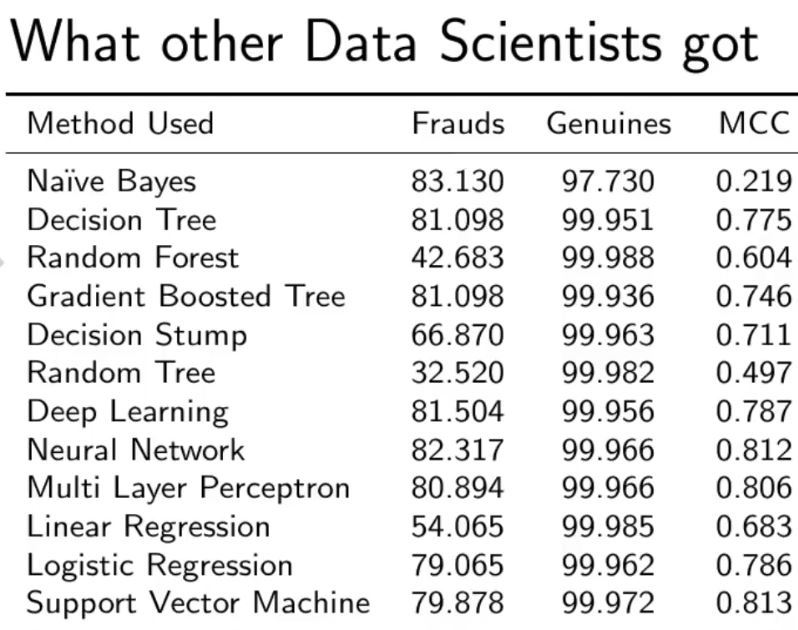


SS-8 (Correlation Matrix)



SS-9 (Confusion Matrix)

**10. COMPARSION OF RANDOM FOREST ALGORITHM WITH OTHER MACHINE LEARNING ALGORITHMS**



SS-10 (Comparison of RFA)

As you can see with our Random Forest Model we are getting a better result even for the recall which is the most tricky part.

**11. CONCLUSION**

* 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.