CREDIT CARD FRAUD DEFECTION SYSTEM USING MACHINE LEARNING

PROJECT REPORT

Submitted by

MOHAMMED SHAHIL S
(20BCS0025)

Under the guidance of

Dr.R.MARUTHAVENI MCA.,M.Phil.,Ph.D.,

Assistant Professor, Department of Computer Science

In partial fulfilment for the award of the degree of

BACHELOR OF SCIENCE

IN

COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

Dr. SNS RAJALAKSHMI COLLEGE OF ARTS AND SCIENCE

(AUTONOMOUS)

Accredited with A+ Grade by NAAC

(Affiliated to Bharathiar University)

COIMBATORE-641 049

APRIL-2023

DEPARTMENT OF COMPUTER SCIENCE

Dr. SNS RAJALAKSHMI COLLEGE OF ARTS AND SCIENCE

(AUTONOMOUS)

Accredited with "A+ Grade by NAAC

(Affiliated to Bharathiar University)

COIMBATORE-641 049

BONAFIDE CERTIFICATE

This is to certify that this project work entitled.

CREDIT CARD FRAUD DEFECTION SYSTEM USING MACHINE LEARNING

is the bonafide record of project work done by

MOHAMMED SHAHIL S

(20BCS0025)

In partial fulfilment of the requirements for the award of the Degree of Bachelor of Science in Computer Science

INTERNAL GUIDE	**HEAD OF THE DEPARTMENT**

Submitted for Viva-Voce Examination held on ________________

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

DECLARATION

I hereby declare that this project work entitled as **CREDIT CARD FRAUD DEFECTION SYSTEM USING MACHINE LEARNING** submitted to **Dr.SNS Rajalakshmi College of Arts and Science, Coimbatore** is the record of original work done by myself under the guidance of **Dr.R.MARUTHAVENI MCA.,M.Phil.,Ph.D.,Assistant Professor** in **Department of Computer Science, Dr.SNS Rajalakshmi College of Arts and Science, Coimbatore** and this work has not formed the basis for the award of any degree to any candidate in any university.

Date:

Place: Coimbatore

Signature of the Candidate
MOHAMMED SHAHIL S
(20BCS0025)

ACKNOWLEDGEMENT

I convey my sincere thanks to all people who have contributed a lot for the successful completion of this project. I wish to convey my profound gratitude and special thanks our respected **Chairman Dr.S.N.SUBBRAMANIAN M.Tech., Ph.D., MBA., Ph.D., Dr. SNS Rajalakshmi College of Arts and Science, Coimbatore** for giving me an opportunity to take up this project.

I warmly acknowledge my sincere and devoted thanks to our beloved **Correspondent Dr.S.RAJALAKSHMI B.Sc., M.B.B.S., DGO, Dr. SNS Rajalakshmi College of Arts and Science,** for encouraging me throughout the course of my study and for providing the environment to complete this project.

I extend my thanks to our respected **Secretary Dr.S.NALIN VIMAL KUMAR ,B.E.,M.S., Ph.D.,Dr.SNS Rajalakshmi College of Arts and Science, Coimbatore** for his support rendered during this project.

I would like to thank our **CEO Dr.M.DANIEL M.Sc.,M.Phil., Ph.D., Principal Dr.R.Anitha MBA.,M.Phil., Ph.D.,MCA.,MBA(Fin).,NET** for providing all facilities in our college to carry out this project.

It is my prime duty to solemnly express my sense of gratitude to our **vice principal Dr.P.NARESH KUMAR M.Sc.,M.Phil., Ph.D.,** for providing all facilities in our college to carry out this project.

It is my prime duty to solemnly express my sense of gratitude to **Mr.B.MURUGESAKUMAR MCA.,M.Phil., Prof. & Head of the Department of Computer science** for his support and guidance.

I really deem this as a special privilege to convey my prodigious and everlasting thanks to my guide **Dr.R.MARUTHAVENI MCA.,M.Phil.,Ph.D.,Assistant professor in Computer Science** for her valuable guidance and personal interest in completing my project.I express my sincere thanks to all my staff members in Department of Computer Science of **Dr. SNS Rajalakshmi College of Arts and Science** for their support in my project.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Nowadays as we can see that there is a huge increase online payment and the payment is mostly done with the help of credit cards. It becomes a big problem for marketing company to overcome with the credit card fraudulent activities. Fraudulent can be done in many ways such as tax return in any other account, taking loans with wrong information etc. Therefore, we need an efficient fraudulent detection model to minimize fraudulent activity and to minimize their losses. There are a huge number of new techniques which provide different algorithms which help in detecting number of credit card fraudulent activity. Basic understanding of these algorithms will help us in making a significant credit card fraudulent detection model. This paper helps us in finding doubtful credit card transaction by proposing a machine learning algorithms. Credit Card Fraudulent detection comes under machine learning, and the objective is to reduce such type of fraudulent activity. This type of fraud is happening from past, and till now not much research has done here in this particular area. The types of credit fraud in transactions are bankruptcy fraud, behavioral fraud, counterfeit fraud, application fraud [3]. .There are experiments done before on credit card fraudulent activity on basis of meta-learning. There is certain limit of meta-learning. There are two features which is introduced here in our report is True Positive and False alarm. Both these features play an important role in catching fraudulent because the rate of determining fraudulent behavior is quick. For the better performance of model, we need a better classifier. Different classifier can be combined together with help of meta-learning.

1.2 OBJECTIVES

To run a suitable business, vendors need to make a profit, which can be calculated by subtracting the cost of doing business from the total sell price. Therefore, fraudulent become a business's tolerance among online payment, among financing company, gross margin is calculated by (sell price - cost of goods sold). The lower the margin, there will be low risk for fraudulent payment. In practice, whenever fraudulent occurs, the cardholder have to complain to the financing company and the debit from card is usually cancelled, which means there is a loss for either cardholder's bank or the finance company. Fraudulent turns as a financial risk to the financial company and the cardholder's bank. To overcome with fraudulent, fraudulent detection techniques should be used. The main objective is to prevent the customer from fraud because if this kind of things keep happening then people will not show there interest in taking credit card and using there facility which is given by the banks and other financial company.

Therefore, it's become an essential thing nowadays. People should also takes care of their personal information by keeping it to the limited source. The fraudulent activity start with the leaking of the someone personal information like credit number, one time password, registered mobile number and many more. The sharing of someone personal information should be reduced because fraudulent activity begin with the help of someone personal information like credit card number and many more.

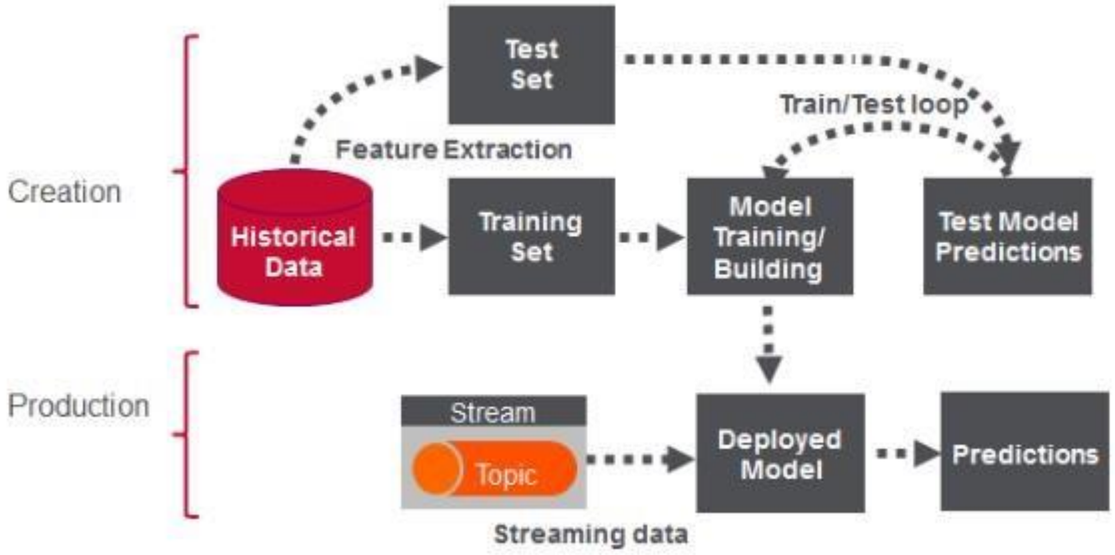


Figure 1.1: System Mechanism

1.3 EXISTING SYSTEM

The previous detecting technique takes a long time to catch fraud which is basically depend on the database, not that much accurate and not give the result in-time. After that algorithm which is used for the detection of credit card fraudulent is generally on basis of analysis, fraudulent detection based on credit card transaction made by cardholder and the credit rate for cardholders.

There are certain limits of meta-learning. There are two features which is introduced here in our report is True Positive and False alarm. Both these features play an important role in catching fraudulent because the rate of determining fraudulent behavior is quick. For the better performance of model, we need a better classifier. Different classifier can be combined together with help of meta-learning.

Previously attempts have been made to work out Credit Card Fraud Detection system using SVM (Select Vector Machine). SVM makes use of hyperplane to classify the data points in a collection. A good hyperplane associates greater number of data points within its margin [2].

This is not efficient for a large amount of data sets. As, in large amount of data sets there is a probability of redundant data which will take more time to process.

Therefore, it usually delayed in calculating the fraud or there might be probability to not calculate in time.

13.1 DISADVANTAGES OF EXISTING SYSTEM

- In case of fraud there is a high amount losses and thus because of this loss, card limit should be reduced.
- The fraudulent should be detected in real time and omission in false transactions is mandatory.
- Reasons of fraudulent should be identified from data available.

- System should be capable in identifying the trend of fraud transaction.

- Credit card fraudulent transaction should be based on web service scheme.

1.4 THE SYSTEM PROPOSED

- In this model we overcome with the issues in a significant way. Using Isolation random forest and local outlier factor algorithm we can detect the fraud in actual time and find out the way to minimize the fraud to produces an optimized result so that it will perform a better prediction. On the basis of customer's behavior, we can detect fraudulent. Here the local outlier factor is used.

- We have used logistic regression and random forest. We can get more accuracy like 0.99 etc…

- We are taking the dataset with help of simple GUI from our local directory where we downloaded the dataset.

- With the help of random forest algorithm and local outlier factor we are finding the data point which is different from its neighbor and can be a fraudulent transaction with its outlier behavior.

- We have two classification class which is named as class 0 and class 1.

- If there is legal transaction then the result will store in class 0 and if there is a fraudulent transaction then the result will store in class 1.

CHAPTER 2

LITERATURE SURVEY

In our paper we referred to various papers for improving the performance of routing, reduce delay of information, reduce packet loss rate, reduce link failure, to improve packet delivery rate, to reduce energy consumption. There are a huge number of new techniques which provide different algorithms which help in detecting number of credit card fraudulent activity. Basic understanding of these algorithms will help us in making a significant credit card fraudulent detection model. This paper helps us in finding doubtful credit card transaction by proposing a machine learning algorithms. There are two features which is introduced here in our report is True Positive and False alarm. Both these features plays an important role in catching fraudulent because the rate of determining fraudulent behavior is quick. As per today's Network plays an important role therefore it is mandatory for our models to be up to date to perform better detection capabilities. Whenever new fraudulent activity are detected then our model should be that much better to perform real time analysis. Other than traditional machine learning methods Fraudulent Detection System has been achieved through using Neural Networks [5]. To prevent personal information has become a huge task for financial company because there are a lot of attack on the system to steal someone personal information to perform fraudulent. Our model has two essential feature which will help in finding abnormal behavior in form of charts for different column such as time, amount etc.

2.2 CREDIT CARD FRAUDULENT DETECTION

We publish a Credit Card fraudulent detection model whose performance is evaluated on basis of anonymized data sets and found that detection model performance is good for this dataset. This is incorporated that this model creates two separate patterns for databases, one for fraud and other for legal transactions. The fraudulent detection model should be more accurate in order to detect the changing behavior of consumer and his behavior. We can predict this fraudulent by running our model after every fixed amount of transaction or after a fixed interval of time. AI provides procedure for various types of calculations which can be performed independently. If there is any outlier value in our dataset, then our model can detect it. Outlier value means the value which deviates by a long margin from their neighbor can perform abnormal behavior. That outlier behavior is the fraudulent transaction in dataset. We have also reduced redundancy of datasets by removing some of the redundant data from our dataset. Because our main aim is achieving the real time analysis and for that we need to reduce the datasets so that we can speed up our algorithm performance.

2.3 DATA SAMPLING

Since, Random forest algorithm is a machine learning algorithm therefore we need trained dataset to perform our mechanism. These trained datasets are then loaded to the main memory of the system. Our dataset has almost 300,000 value so it's a difficult task to load trained dataset in main memory. For that purpose we have removed the redundant datasets. We have trained our dataset from previous data, we did like this because our model should be trained on previous data and should be able detect fraudulent transaction of the current month, which will help in real world.

2.4 CREDIT CARD FRAUDULENT DETECTION USING HIDDEN MARKOV MODEL

In our paper we utilized HMM to identify fraudulent. We demonstrated the exchanges of MasterCard by utilizing HMM. For swiping reason, we have utilized the RFID gadget to demonstrate the shopping exchanges. We identified the misbehaviors by observing the conduct of the client. We include High security addresses page additionally, in case card is stolen, we have given another profile ID to the consumer and gave ONE TIME PASSWORD for security reasons. We have given right to the admin to block the card from obstructing in case card is lost. As our aim is to achieve the better accuracy but our dataset we could achieve up to 99.97%. As for fraudulent detection, the false alarm plays an important role, as whenever there is a fraud transaction it shows an outlier transaction which will differ from its neighbor or we can say that deviate from the given data point. We give more priority to fraudulent catching algorithm then the false alarm because our aim is to catch the fraudulent at the very first moment.

2.5 CREDIT CARD FRADULENT DETECTIONUSING DECISION TREE INDUCTION ALGORITHM

In Snehal Patiletal, describes the “Decision Tree Induction Algorithm” which is used for Credit Card Fraud Detection [1]. In this paper it discusses about the method, decision tree approach is a new cost sensitive technique compared with well-known traditional classification models on a real-world credit card fraud data set, which reduces the sum of misclassification cost, in selecting the splitting attribute at each of the non-terminal node become advance. Credit card fraud detection is to reduce the bank risks, also used to equalize the transaction information with credit card fraud transaction of historical profile pattern to predict the probability of being fraud on a new online transaction. In this model use of “Credit Card Fraud Detection Using Decision Tree for tracing Email and IP Address. By using this technique, we can able to find out the fraudulent customer/merchant through tracing the fake mail and IP address. If the mail is fake, the customer/merchant is suspicious and information about the owner/sender is traced through IP address.

As prediction of score is much important task according to our model therefore we are predicting the score on the basis of the given formula:

Score = 0.5 * TP + 0.5 * Deviation

Where, TP is True Positive value and Deviation is the deviation of outlier data from the standard data point.

On the basis of these score we made two classes 0 and 1. If the score is 1 it will move to class 1 and termed as legal transaction and if the score is 0 it will move to class 0 and termed as fraudulent transaction. At last, the accuracy is calculated on the basis of how many fraud transactions are there in our dataset and how many we predicted with the help of our model.

CHAPTER 3

CREDIT CARD FRAUDULENT DETECTION SYSTEM

Unusual pattern which is known as outliers which not fulfill the expected behaviors is known as Anomaly detection. Many business applications are based upon this technique, unusual patterns in network are identified. Its helps in detecting credit card fraudulent as well as operating system fraudulent. Jupiter notebook we are going to take the credit card fraud detection as the case study so that we can understand the concept in detail. Outlier value is those value which shows an abnormal behavior from its neighbor or we can say that from standard data point. Generally, Outlier data termed as fraudulent transaction. Our experiment based upon catching fraudulent activity with the help of false alarm. Our model has focused on the use of Isolated Random Forest and Local Outlier Factor, however previous works has also been done using Bayesian Regularization and Gradient Descent Adaptive learning algorithms[4]. There are many advantages of this system and one of the major advantage that we are recognizing the pattern and on the basis of pattern we made chart which will help to understand the fraudulent easily because it is easy to understand the data in the form of chart. We have plotted the chart for every features from V1 to V28. We should also keep the things in mind that other financial bank cannot read the other personal information. We can use this technique to find the scheme which will help in finding credit card fraudulent transaction. The advantages of this system is that it can work in an efficient way for the limited amount of data.
We examine the accuracy and it is quite satisfactory.

3.2 PROBLEM DEFINITION OF CREDIT CARD FRAUDULENT

As in increase of online payment, increase in use of credit card. Many company provides the facility of credit card payment. We can purchase a lot of things using our credit card. People started doing fraud in this field by using someone's credit card and using someone's personal information to issue credit card. Electronic data can be interchange in case of online payment to perform fraudulent. We cannot prevent credit card fraudulent with the help of credit card billing but we need to prevent the fraudulent also. If we will talk about the success story of all the existing system, it is not that much efficient in finding the fraudulent. So, it's become essential to make a system which can find the fraudulent at the very first time and help customer to reduce the fraudulent in their all online transaction and they can get the notification at the very first time that their credentials are using by some other people. This will help him to overcome with this kind of fraud activity at early and can think to modify their losses. There should be limitation on the credit card that we cannot make transaction above this much amount in on day or at a time. This will reduce the amount of losses.

We have two analyzer as random forest algorithm and local factor outlier which will determine the nature of fraudulent whether it is a legal or fraudulent transaction. These will also help us in calculating the score prediction which will represent a more balancing result.

As the result is the mean between the legal and fraud transaction and then the accuracy is determined on basis of them. The dataset should be more balance because it plays an important role. It's also become a mandatory to split our dataset in train dataset and test dataset. Because for building a problem model these two things are essential. So, our goal is to train the dataset and test the class classifier on the basis of this datasets.

BLOCK DIAGRAM

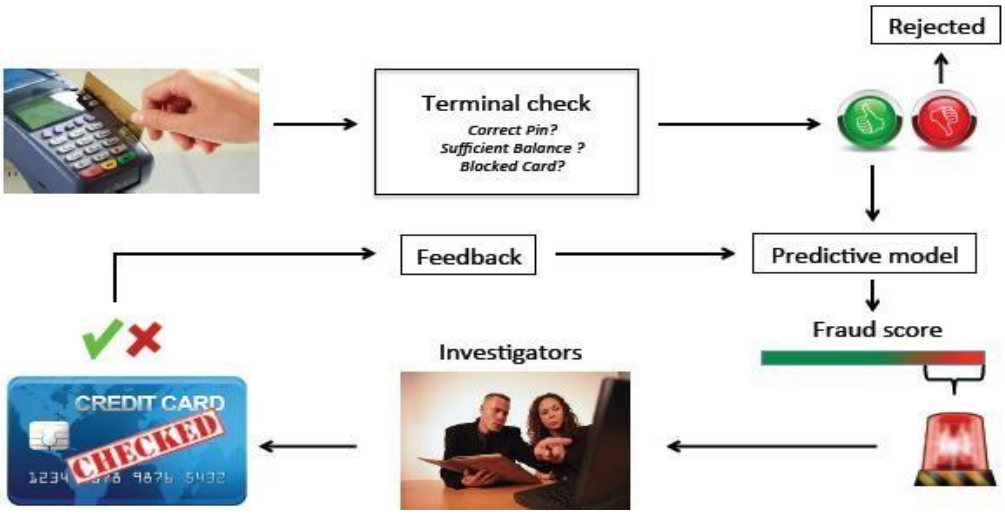


Figure 3.1: Block Diagram

3.4 METHODOLOGY

The task which is performed for the prediction of transaction and labelled as fraud is detected on the basis of binary classification. We make two class for the prediction of fraud: class 0 and class 1.

Class 0 if there is no fraud and class 1 to catch the fraud. This can be done with the help of binary classification.

3.4.1 WHAT ARE ANOMALIES?

Anomalies can be categorized as following:

- Point Anomalies: Point anomaly is a single instance of data. The credit card fraudulent detection technique is based on "amount spend".

- Contextual Anomalies: The best example of contextual anomaly is time-series data.

- Collective Anomalies: Here, Detection of anomaly is based on a set of data instances collectively. Therefore, a set of data will help in detecting fraudulent anomaly. If someone try to theft personal data from server it will come under collective anomaly and named as cyber-attack.

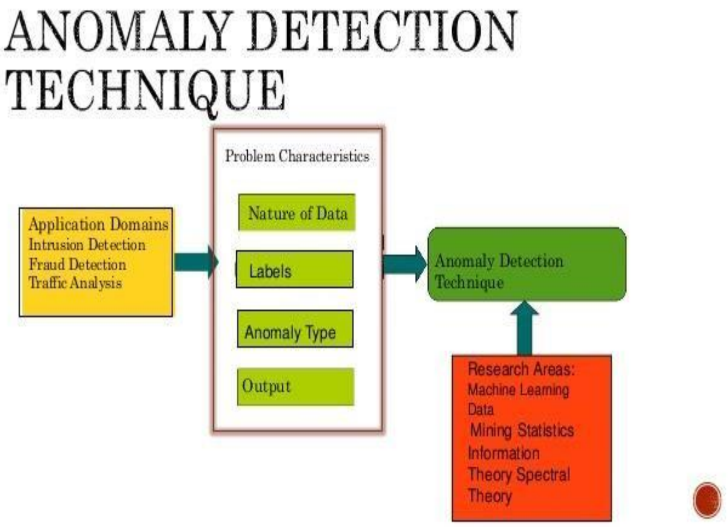


Figure 3.2: Anomaly Detection Technique

3.4.2 ANOMALY DETECTION

Identifying an unobserved pattern in new observation is the main area of concern. It's include training of dataset.

3.4.3 NOISE REMOVAL

Noise removal is the process of removing noise from meaningful data, noise is unnecessary data along with the meaningful data.

3.5 ANOMALY DETECTION TECHNIQUES

The various Anomaly Detection Techniques are as follows

3.5.1 SIMPLE STATISTICAL METHODS

The simple way by which we can determine the irregularities in dataset by determining the deviation of data point from common statistical distribution, for example mean, mode and median.

Anomaly data point is that deviates by a certain standard deviation from mean. To compute average data point we need a rolling window across data points which is known as moving average which is used to find low pass filter.

3.5.2 CHALLENGES WITH SIMPLE STATISTICAL METHODS

The low pass filter allows us to identify anomalies in simple use cases, but there are some framework where this method fails to determine anomaly data point. Data which contain noise data which can be named as abnormal data, as the boundary between normal and abnormal are not accurate. Therefore, it’s a big problem to identify threshold value because the moving average can’t apply in that framework.

3.6 CREDIT CARD FRAUDULENT DETECTION SYSTEMS

All the credit card fraudulent detecting models are evaluated and compared using this model.

Accuracy - It is characterized as a bit of all the quantity of exchanges which are distinguished effectively.

Methodology - This indicates the instrument pursued by the credit card FDS.

True Positive or TP - Legal and fraud transaction are detected on this basis. Genuine transaction only counted here.

False Positive or FP - Legal and fraud transaction are detected on this basis. Fraud transaction only counted here.

Supervised Learning - In this supervised data is fed in the machine.

3.7 FUNCTIONALITIES

Many organization and banks will take the benefit from this model. Because this will be a significant model for the prediction of credit card fraudulent. This will detect the consumer behaviors and his last transaction and predict whether the consumer is fraud or not. We use random forest and local outlier factor for the fraudulent. We need to have controls over the algorithm in order to fit with the data set. It will help our application to improve and to be more efficient in order to detect the fraudulent transactions and help us in solving problems.

3.8 ACCURACY

The Fraudulent Detection is done on basis of previous transaction history of consumer. We will detect out of whole transaction how much result in fraud. Then we will identify whether a new transaction made by customer is fraudulent or not. With the help of this model we achieve 99.97% accuracy in finding fraudulent transactions.

3.9 OBSERVATION

The data set contains 492 frauds out of almost 300,000. This results a probability of 17.2% fraudulent cases. This identified that there is much more fraud customer. The data sets consists of column which start from v1 and end as V28. There are much features present from V1 to V28. Furthermore, there is no missing value present in datasets. The datasets has column name as Time & Amount. The analysis is done on the basis of ranges present in this two columns.

The datasets contains the numerical value which can be called as PCA transformation. Due to security issue, unfortunately we cannot take the original features and information about data. Column V1 to V28 are taken as principal components. The features which is not transformed with PCA are “Time” and “Amount”.

“Time” plays an important role here as it is used to determine the time between each transaction and it is calculated in seconds.

“Amount” is another feature which is used to determine the transactional Amount.

“Class” is the most important feature here in our model which is response variable and it takes the value as 1 and 0. It gives value 1 in case of fraud and value 0 in case of legal transaction. The main goal of this model is to predict the credit card fraudulent, for all transaction which is received as online payment to check whether the transaction is legal or not. If the transaction is genuine then it is consider as legal transaction and the transaction which has fraudulent should be recognize as fraud transaction. All this is performed with the help of random forest algorithm and local outlier factor to make an assumption of true probability and false probability. The result obtain after this algorithm performed successfully is then plotted as graph and heat-map. This model is also tested for different test cases and also compared with the previous all model and the accuracy is also compared.

3.10 MODEL PREDICTION

Now it is time to start building the model. The types of algorithms we are going to use to perform anomaly detection on this data sets are as follows:

3.10.1 ISOLATION FOREST ALGORITHM

Anomalies are detected with the help of Isolation Random Forest algorithm. This algorithm tells the fact that anomalies are data points that are distinct and few. These properties in results describes that, isolation mechanism suspects anomalies. On the basis of above all we came to know that this method is different from all methods which exists in past and more accurate as well. This introduces isolation algorithm is more efficient technique for anomalies detection rather previous algorithm. Moreover, this algorithm takes very less memory and time complexity is also very less. We make binary tree which is small as compare to the datasets.

When both good and bad behaviors present in datasets then Machine learning algorithms should work better to balance the system, and predict the pattern.

3.10.2 WORKING PRINCIPLES OF ISOLATED RANDOM FOREST

The Isolation Random Forest algorithm works by randomly selecting a feature from datasets and then randomly find a split value from minimum and maximum value. According to logic applied, the difference between anomaly observations and normal observation is of few cases. We require more condition in isolating normal observations. The conditions required to differentiate between normal and anomaly observation is used to calculate score. The score is used to make binary decision tree which has child nodes as 0 and 1. Then, finally if 0 is obtain then there is no fraud and if 1 is obtain then there is a fraudulent.

3.10.3 LOCAL OUTLIER FACTOR (LOF)

Local Outlier Factor (LOF) is an outlier algorithm which provide mechanism to compute the deviation of given data point from its neighbors. It consists outlier samples which has a low density as compare to its neighbors. The outlier value is chosen on basis of greater and minimum value present in the cluster of datasets and different from its neighbors. If the outlier value is mismatching from its neighbors, then it would have been caught by the system and result in fraudulent. The conditions required to differentiate between normal and anomaly observation is used to calculate score. The score is used to make binary decision tree which has child nodes as 0 and 1. Then, finally if 0 is obtain then there is no fraud and if 1 is obtain then there is a fraudulent. Therefore, Local outlier factor helps us finding the fraudulent data which is not fitted well within its neighbors. It also helps us in finding the deviation of outlier data from the standard deviation which is followed by all the neighbors.

3.10.4 OBSERVATIONS

- Isolation Forest can detect 73 errors where as Local Outlier Factor can detect 97 errors in order SVM can detect 8516 errors
- Isolation Forest has a better accuracy which is 99.74% than LOF which is 99.65% and SVM has 70.09
- When we compare error precision & recall for these 3 models, the Isolation Forest performance is much better than that of LOF as we can see that the detection of fraud cases is around 27 % in case of Isolation Forest where as in case LOF detection rate of just 2 % and in case of SVM of 0%.

CHAPTER 4

INTRODUCTION OF MACHINE LEARNING

AI is a mechanism which features algorithms and calculations based on a normal human intelligence to address a problem. The AI behaves and approaches a problem in a similar way that a normal human brain would. Its working mechanism is influenced by human thinking. A collection of expectation and result is achieved by AI by portraying information in a form termed as ‘test information’ without making use of any predetermined models or being trained in that particular domain. Problems catering to non-related dimensions such as email sifting, PC vision, location of system gate crashers are addressed. Thus it is assertive that it is not possible to train an AI to address a particular domain, instead an AI trained with general problem solving abilities, builds up its own algorithms for a set of problems.

An AI engine is allocated with responsibility of prediction or analysis using a PC framework and set of data. For this an AI engine is allocated with packages of scientific methods, logistic calculations, data sets and knowledge about the field of the problems for performing. At the initial stage AI makes use of various algorithm to perform exploratory analysis for marking out various features of the given problem. Information mining is one of the necessary tool used by various AI models for this purpose. Moreover, the entire operation of AI is carried based on unsupervised learning model which leaves a very less room for training a robust AI for only a problem specific solution. However, for business purposes modifications are performed before its application.

4.2 OVERVIEW OF MACHINE LEARNING

The name was authored in 1959 by Arthur Samuel Tom M. Mitchell gave a generally cited, increasingly formal meaning of the calculations contemplated in the AI field. This meaning of the assignments in which AI is concerned offers an in a general sense operational definition as opposed to characterizing the field in psychological terms. This pursues Alan Turing's proposition in his paper "Registering Machinery and Intelligence", in which the inquiry "Can machines believe?" is supplanted with the inquiry "Can machines do what we (as speculation elements) can do?" In Turing's proposition the different attributes that could be controlled by a reasoning machine and the different ramifications in building one are uncovered.

Before the introduction of machine learning a general assumption was that a robot needs to learn everything from a human brain to function appropriately. But as efforts were made to do so, it was realized that it is very difficult to make a robot to learn everything from a human brain as the human brain is very much sophisticated. An idea was then proposed that rather than teaching a robot everything we know, it is easier to make the robot learn on its own. Thus was the birth of the term of 'machine learning' to describe this idea. Machine learning uses different approaches and algorithms to train a model. Methods applicable to models vary widely based on certain features. The type of dataset we are working upon largely determines how we approach while training the model. Based on the dataset we will feed to the algorithm, the training model would vary. The size, type and dynamism of the dataset will decide what type of training model we would build. Finally on deciding upon the training model, modifications need to be made to achieve the proper objective function to generate proper set of output that we wish to achieve. The stages of machine learning process are rather termed as ingredients than steps, because the machine learning is an iterative process. The iterative process is repeated each time to achieve maximum optimization and efficiency.

4.3 MACHINE LEARNING-BASED APPROCHES

The following is a concise outline of mainstream AI based systems for inconsistency identification.

4.3.1 DENSITY BASED DETECTION OF ANOMALY

It derives its working mechanism from KNN algorithm

Assumption - Relevant data locates themselves around a common point in close proximity whereas irregular data are placed at a distance. The data points are clustered at a closed proximity based on a density score, which may be derived using Euclidian distance or appropriate methods based on the data. Classification is made on two basis:

K closest neighbor: In this method the basic clustering mechanism is dependent on separation measurements of each data points which determines the clustering or similarities of each information considered.

Relative thickness of the information - Also known as Least Outlier Fraction (LOF).

Calculation is performed on the basis of separation metric.

4.3.2 CLUSTERING BASED DETECTION OF ANOMALY

Clustering is an exceptional algorithm known for its optimization and robust nature. For this reason, it is widely used in unsupervised learning

Assumption - Data points that are similar tends to get gather around specific points. The relative distance of each cluster is achieved by its shortest distance from the centroid of the space.

K means is widely used in data classification. It makes use of k means algorithm to cluster closely related data in close proximity forming clusters.

4.3.3 SVM BASED DETECTION OF ANOMALY

- A support vector machine is one of the most important algorithm used for classification purposes

- The SVM uses methods to determine a soft boundary to distinguish data clusters. Data closely related falls within the parameter of a closed boundary. This results in formation of multiple clusters. SVM is widely used for binary classifications also. Most of the SVM algorithms works based on unsupervised learning.

- The yield of an abnormality locator are mostly numeric scalar qualities for distinguishing areas of explicit edges.

In this Jupiter journal we are going to assume the acknowledgment card misrepresentation recognition as the contextual investigation for understanding this idea in detail utilizing the accompanying Anomaly Detection Techniques in particular

4.4 DATASET

A dataset corresponds to a collection of data which may or may not be related to each other. A dataset can consist of data related to a particular domain. It may consist information for a single member or a group of member. For ex personal and other relevant details of an employee can be termed as a dataset, whereas collection of the information of all the employees working for that company is also a dataset. Thus the purpose of the problem defines the size of the dataset. A dataset consists of multiple columns often termed as parameters and multiple rows known as tuples. Individual data pieces are also termed as datum. For example, in a data set consisting of employee details of a company: columns such as salary, date of joining, title etc. can be termed as attributes. Whereas a row consisting of all the information of a particular employee is called a tuple. In simple words a dataset can be termed as a collection of inter-related tables containing information. The relational parameters are often on numerical or logical basis. Modification of the dataset needs to be made while keeping these interdependencies in mind. Datasets which are too large to be operated on by traditional database methods are termed as Big Data. With rising data generation, the need for new algorithms and tools to cope up with thousands of gigabytes of data have given rise to Big Data Analytics. Modified and robust algorithms to optimally operate on the varied range of data is in development. Other than that data is also classified on basis of its dynamisms. A static data requires a single set of algorithms to operate upon whereas a real time data requires a dynamic algorithm to suit the operational needs as and when required.

4.4.1 DATASET DETAILS

- Time
- Number of seconds slipped by between this exchange and the primary exchange in the dataset
- V1 up to V28
- It might be consequence of a PCA Dimensionality decrease to secure client personalities and touchy features (v1-v28)

4.4.2 AMOUNT

- Transaction amount
- Class
- 1 for fraudulent transactions, 0 otherwise

CHAPTER 5

SYSTEM SPECIFICATION

The necessity for the most part dependent on two classes: they are practical portray every single required usefulness for framework administrations which are given by the customers. Non useful necessities characterize the framework properties and compels. The equipment prerequisites indicate the equipment functionalities and required speed and limit of the fringe.
The product prerequisites incorporate programming expected to create and run the framework.

5.2 HARDWARE SPECIFICATION

• System	- Core i5
• Mobile	- Android
• Monitor	- RGB Colour
• Hard Disk	- 2 TB
• Mouse	- Microsoft
• Ram	- 8GB

5.3 SPECIFICATION OF THE SOFTWARE

• Operating system	- Win 10
• Dataset	- csv
• Language	- Python

5.4 SOFTWARES USED

- Python 3.5

- NumPy 1.11.3
- Matplotlib 1.5.3
- Pandas 0.19.1 □ Seaborn 0.7.1
- SciPy
- Scikit-learn 0.18.1

CHAPTER 6

DESIGN ENGINEERING

The UML is used for business and production based works. The task of using UML is to provide a solution or working of a product or model using visual representation. UML involves usage of lock diagrams and flow chart to depict the interrelation and workflow of a model. Sometimes it is also used for planning purposes or analysis as a reference for further development of a project

- Provides direction with regards to the requests of the group exercise.
- Software ancient rarities create.
- Directs of errand to individual designers and group.
- Offer the criteria to check & estimate the task's item & exercise.

The UML intestinally process autonomous and can be attached with regards to various procedures. All things considered, It is the most reasonable for utilize driven, intuitive and gradual improvement forms. A case for such procedure is Rational Unified Process (RUP).

6.2 ACTIVITY DIAGRAM

It portray the work process conduct of a framework. It ought to be utilized related to other displaying methods, for example, connection and state chart.

displaying methods, for example, connection and state chart.

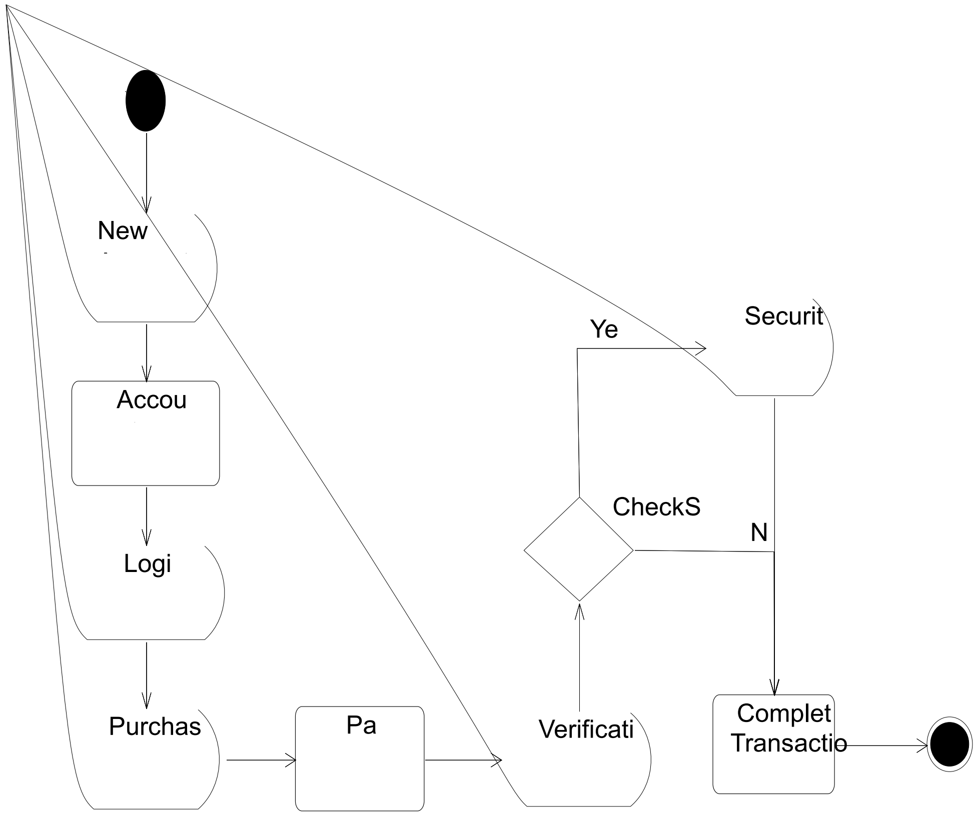


Figure 6.1: Activity Diagram

6.3 USE CASE DIAGRAM

Use case chart show the relationship, among performers and clients. Use cases are utilized in pretty much every task they are useful in uncovering prerequisites and arranging the venture. Amid the underlying phase of a task most use cases ought to be characterized yet as a venture proceeds with more become an obvious

Use case diagrams are used to describe association of actors along with the working model. It is often used to describe a static state of a model. Use case model consists mainly of two components: The one who interacts with the system (Actor) and the system in consideration.

Use case charts are3 formally incorporate into two displaying dialects they are Unified Modelling Language (UML) and System Modelling Language (SML).

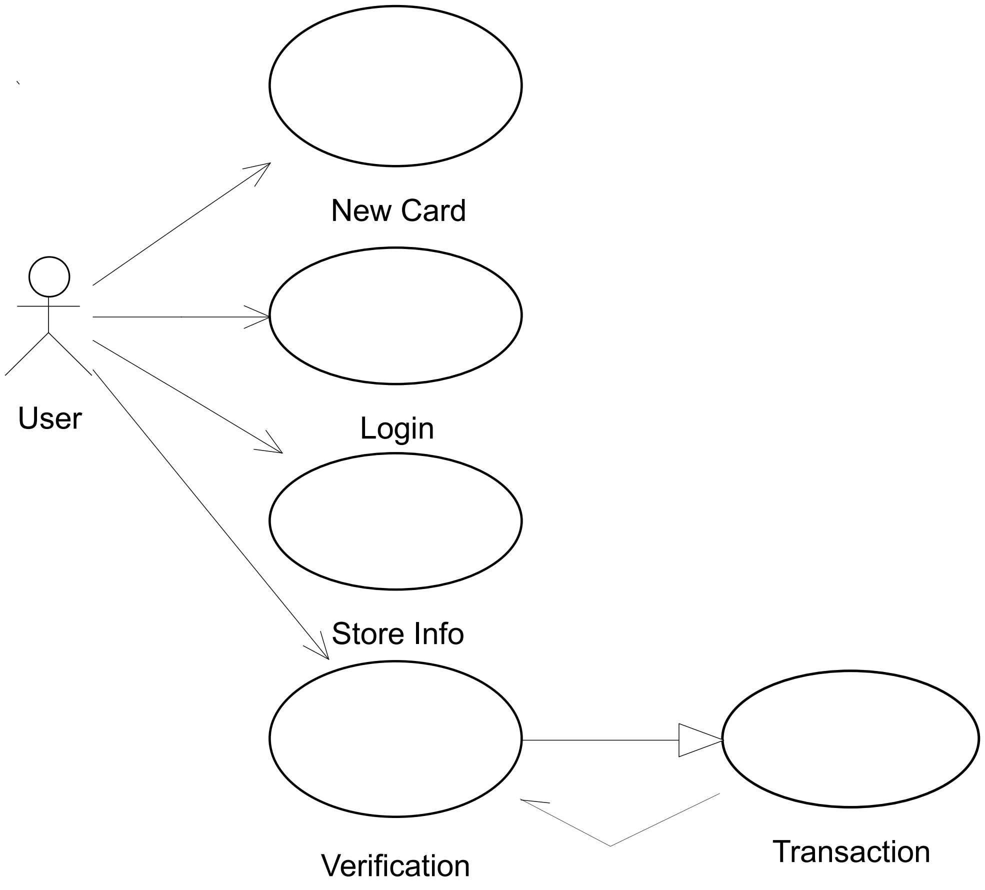


Figure 6.2: Use Case Diagram

6.5 CLASS FEATURE

Class diagram makes us of inter-related structures which consists of package, entities, objects and variables. This depicts relationship between each of the entities through associations, containment and inheritance etc. Using class diagram it becomes easier to understand the holistic working of entities in work along with their inner functionalitites. It is widely used in Object Oriented Software designs.

6.6 THE DATA-FLOW-DIAGRAM

Data Flow Diagram is used to represent the requirements of a system in graphical form. It depicts what the data flow is rather than how they are being processed.It is known as bubbler chart. It defines important transaction in a system as a part of requirement of the model. It is used in the starting phase of a design process for reference of further development on the basis of the current workflow.

It is depicted by collection of bubbles and lines. The bubbles represents the transactions and operations whereas the lines demonstrates the connection/flow between each transactions. It is independent of hardware, software and datasets used and is a general outline in simple words.

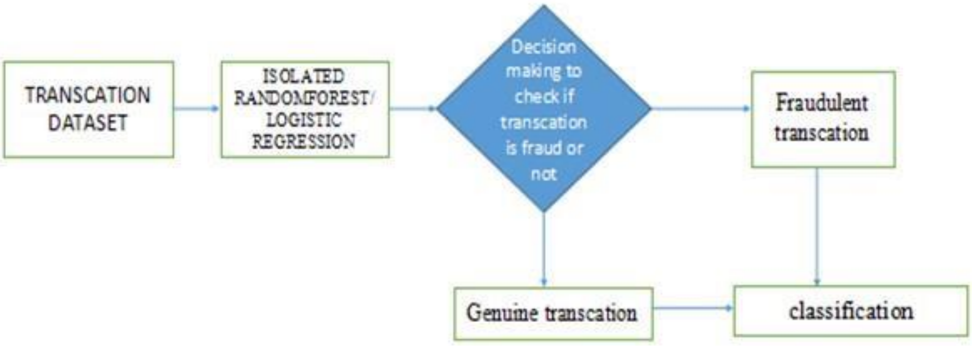

TRANSCATION DATASET
ISOLATED RANDOMFOREST/ LOGISTIC REGRESSION
Decision making to check if transcation is fraud or not
Fraudulent transcation
Genuine transcation
classification

Figure 6.5: Data Flow Diagram

6.7 COMPONENT DIAGRAM

Segment chart shows the abnormal state bundle structure of the code itself. Conditions among parts are demonstrated including source code segments, double code segments, and executable segments. A few parts exist at arrange time, at connection time, at run time well as at more than one time

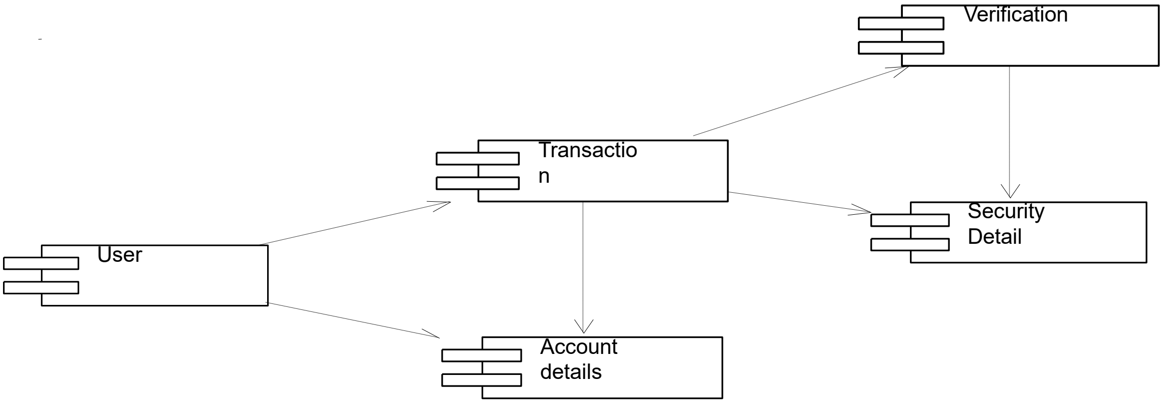


Figure 6.6: Component Diagram

6.8 DEPLOYMENT DIAGRAM

Arrangement graphs shows the design of run time handling components and the product parts, procedures, and items that live on them. Programming part occasion speak to run time appearances of code units.

A deployment diagram is UML diagram type that shows the execution architecture of a system,including nodes such as hardware or software execution environments,and the middleware connecting them deployment diagram are typically used to visualizethe physical hardware and software of a system.

There are two types of nodes in a deployment diagram;device nodes are computing resources with processing capabilities and the ability to execute programs

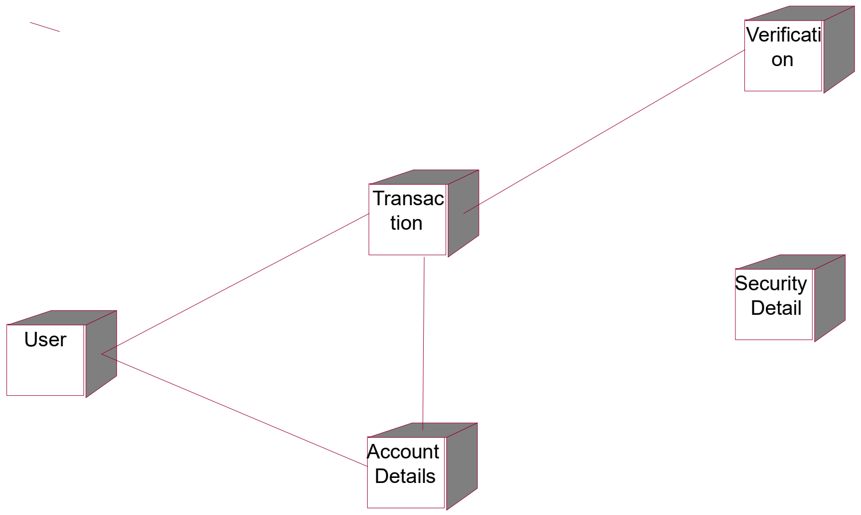


Figure 6.7: Deployment Diagram

CHAPTER 7

IMPLEMENTAION

Implementation phase brings out the design tweaked out into a operational system. Hence this can be deliberated to be most precarious juncture in accomplishing the efficacious system and in convincing the user faith that system will operate and be effective. This phase encompasses vigilant planning & design, examination of prevailing system and constraints on execution, design & scheming of methods to change over.

7.2 PROCEDURE FOLLOWED DURING IMPLEMENTATION

The application – Credit Card Fraud Detection which is in itself the complete & full-fledged GUI enabled application to envisage/foresee the authenticity & legitimacy of a transaction has been implemented, as per the following steps:

- Install Anaconda from an reliable source.

- Import packages: pandas, Scipy, Matplotlib, Seaborn

- Load the dataset, a dataset is the pool of data for analytical/critical purpose, a (.CSV) file.

- Reconnoiter and get through the dataset through data. shape, data. describe.

- Split the dataset into training dataset and testing dataset.

- Plot histogram of the dataset to epitomize/depict numerical data.

- Determine the count of fraud cases by checking if class is 0 or 1.

- In the similar procedure, get the correlation matrix.

- Next, there is a need to determine the local outlier factor.

- This is followed by use of random forest algorithm to find accurate results.

- The GUI is developed using PyQt library.

- The PyQt library, provides tools to achieve a complete GUI enabled application, similar to swings in java environment.

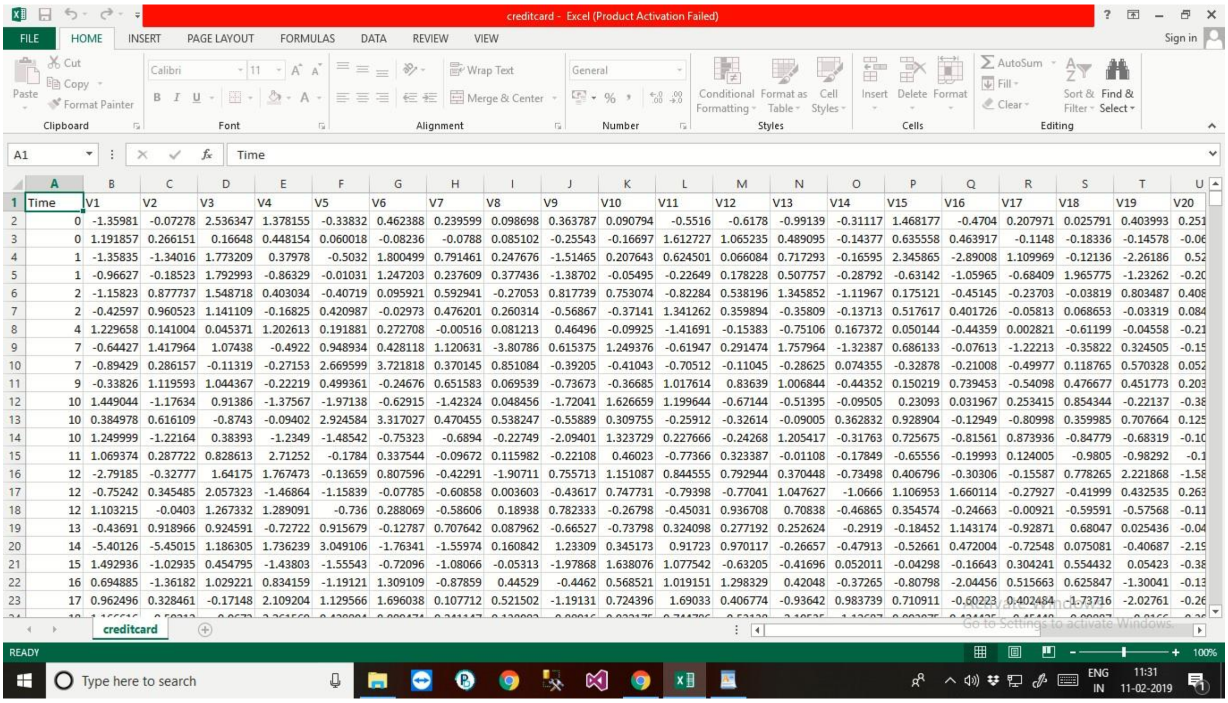
- Define the constructor in the file.

- Write down the entire implementation inside, thus encapsulating everything inside a GUI-enabled python file.

- Machine learning alorithms use historical data as input to predict new output values

7.2.1 DATASET DESIGN

The dataset holds information about credit card transactions which has been made in a span of two days. The number of frauds have been calculated as 492 out of 284,807 transactions. The details have been given in form of positive and non-positive numerical values. The dataset contains 31 features which has been labelled as V1-V28 due to confidential reasons. The feature which has been reveled are Time and Amount of transaction. Here time denotes the number of seconds elapsed from the first transaction of Day 1. Amount of transaction consists of positive value denoting deposit and non-positive value denoting withdrawal.

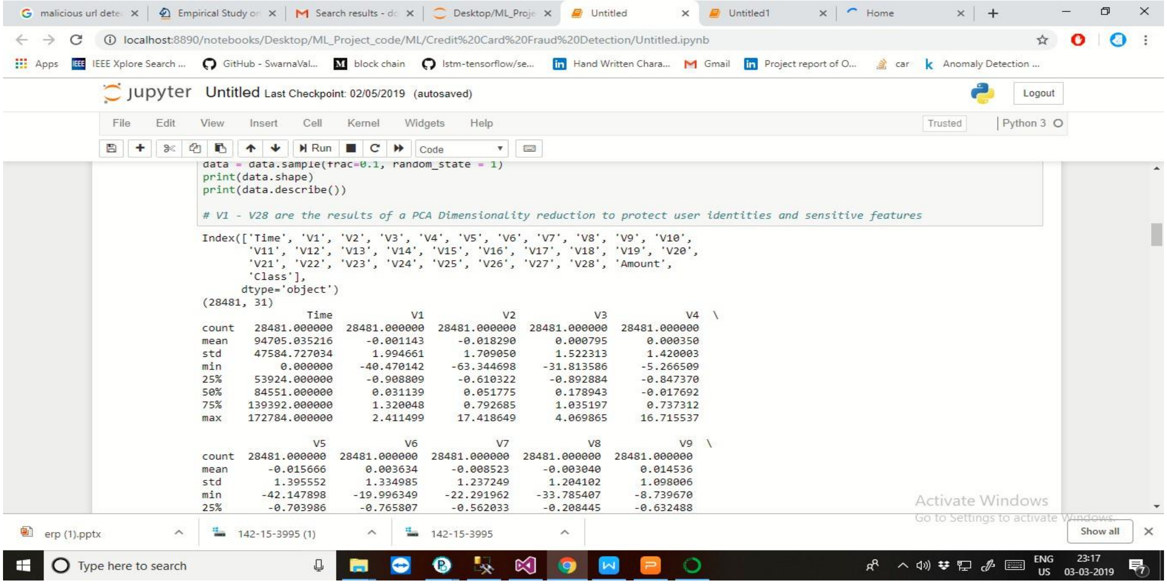


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20
2	0	-1.35981	-0.07278	2.536347	1.378155	-0.33832	0.462388	0.239599	0.098698	0.363787	0.090794	-0.5516	-0.6178	-0.99139	-0.31117	1.468177	-0.4704	0.207971	0.025791	0.403993	0.251
3	0	1.191857	0.266151	0.16648	0.448154	0.060018	-0.08236	-0.0788	0.085102	-0.25543	-0.16697	1.612727	1.065235	0.489095	-0.14377	0.635558	0.463917	-0.1148	-0.18336	-0.14578	-0.06
4	1	-1.35835	-1.34016	1.773209	0.37978	-0.5032	1.800499	0.791461	0.247676	-1.51465	0.207643	0.624501	0.066084	0.717293	-0.16595	2.345865	-2.89008	1.109969	-0.12136	-2.26186	0.52
5	1	-0.96627	-0.18523	1.792993	-0.86329	-0.01031	1.247203	0.237609	0.377436	-1.38702	-0.05495	-0.22649	0.178228	0.507757	-0.28792	-0.63142	-1.05965	-0.68409	1.965775	-1.23262	-0.20
6	2	-1.15823	0.877737	1.548718	0.403034	-0.40719	0.095921	0.592941	-0.27053	0.817739	0.753074	-0.82284	0.538196	1.345852	-1.11967	0.175121	-0.45145	-0.23703	-0.03819	0.803487	0.408
7	2	-0.42597	0.960523	1.141109	-0.16825	0.420987	-0.02973	0.476201	0.260314	-0.56867	-0.37141	1.341262	0.359894	-0.35809	-0.13713	0.517617	0.401726	-0.05813	0.068653	-0.03319	0.084
8	4	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.00516	0.081213	0.46496	-0.09925	-1.41691	-0.15383	-0.75106	0.167372	0.050144	-0.44359	0.002821	-0.61199	-0.04558	-0.21
9	7	-0.64427	1.417964	1.07438	-0.4922	0.948934	0.428118	1.120631	-3.80786	0.615375	1.249376	-0.61947	0.291474	1.757964	-1.32387	0.686133	-0.07613	-1.22213	-0.35822	0.324505	-0.15
10	7	-0.89429	0.286157	-0.11319	-0.27153	2.669599	3.721818	0.370145	0.851084	-0.39205	-0.41043	-0.70512	-0.11045	-0.28625	0.074355	-0.32878	-0.21008	-0.49977	0.118765	0.570328	0.052
11	9	-0.33826	1.119593	1.044367	-0.22219	0.499361	-0.24676	0.651583	0.069539	-0.73673	-0.36685	1.017614	0.83639	1.006844	-0.44352	0.150219	0.739453	-0.54098	0.476677	0.451773	0.203
12	10	1.449044	-1.17634	0.91386	-1.37567	-1.97138	-0.62915	-1.42324	0.048456	-1.72041	1.626659	1.199644	-0.67144	-0.51395	-0.09505	0.23093	0.031967	0.253415	0.854344	-0.22137	-0.38
13	10	0.384978	0.616109	-0.8743	-0.09402	2.924584	3.317027	0.470455	0.538247	-0.55889	0.309755	-0.25912	-0.32614	-0.09005	0.362832	0.928904	-0.12949	-0.80998	0.359985	0.707664	0.125
14	10	1.249999	-1.22164	0.38393	-1.2349	-1.48542	-0.75323	-0.6894	-0.22749	-2.09401	1.323729	0.227666	-0.24268	1.205417	-0.31763	0.725675	-0.81561	0.873936	-0.84779	-0.68319	-0.10
15	11	1.069374	0.287722	0.828613	2.71252	-0.1784	0.337544	-0.09672	0.115982	-0.22108	0.46023	-0.77366	0.323387	-0.01108	-0.17849	-0.65556	-0.19993	0.124005	-0.9805	-0.98292	-0.1
16	12	-2.79185	-0.32777	1.64175	1.767473	-0.13659	0.807596	-0.42291	-1.90711	0.755713	1.151087	0.844555	0.792944	0.370448	-0.73498	0.406796	-0.30306	-0.15587	0.778265	2.221868	-1.58
17	12	-0.75242	0.345485	2.057323	-1.46864	-1.15839	-0.07785	-0.60858	0.003603	-0.43617	0.747731	-0.79398	-0.77041	1.047627	-1.0666	1.106953	1.660114	-0.27927	-0.41999	0.432535	0.263
18	12	1.103215	-0.0403	1.267332	1.289091	-0.736	0.288069	-0.58606	0.18938	0.782333	-0.26798	-0.45031	0.936708	0.70838	-0.46865	0.354574	-0.24663	-0.00921	-0.59591	-0.57568	-0.11
19	13	-0.43691	0.918966	0.924591	-0.72722	0.915679	-0.12787	0.707642	0.087962	-0.66527	-0.73798	0.324098	0.277192	0.252624	-0.2919	-0.18452	1.143174	-0.92871	0.68047	0.025436	-0.04
20	14	-5.40126	-5.45015	1.186305	1.736239	3.049106	-1.76341	-1.55974	0.160842	1.23309	0.345173	0.91723	0.970117	-0.26657	-0.47913	-0.52661	0.472004	-0.72548	0.075081	-0.40687	-2.19
21	15	1.492936	-1.02935	0.454795	-1.43803	-1.55543	-0.72096	-1.08066	-0.05313	-1.97868	1.638076	1.077542	-0.63205	-0.41696	0.052011	-0.04298	-0.16643	0.304241	0.554432	0.05423	-0.38
22	16	0.694885	-1.36182	1.029221	0.834159	-1.19121	1.309109	-0.87859	0.44529	-0.4462	0.568521	1.019151	1.298329	0.42048	-0.37265	-0.80798	-2.04456	0.515663	0.625847	-1.30041	-0.13
23	17	0.962496	0.328461	-0.17148	2.109204	1.129566	1.696038	0.107712	0.521502	-1.19131	0.724396	1.69033	0.406774	-0.93642	0.983739	0.710911	-0.60223	0.402484	-1.73716	-2.02761	-0.26

Figure7.1: Dataset Design

7.2.2 DATA DESCRBE

The shape and characteristics of the data values belonging to each column has been described in the following step. The data describe stage belongs to starting stage of exploratory analysis stage.



```
data = data.sample(frac=0.1, random_state = 1)
print(data.shape)
print(data.describe())

# V1 - V28 are the results of a PCA Dimensionality reduction to protect user identities and sensitive features
```

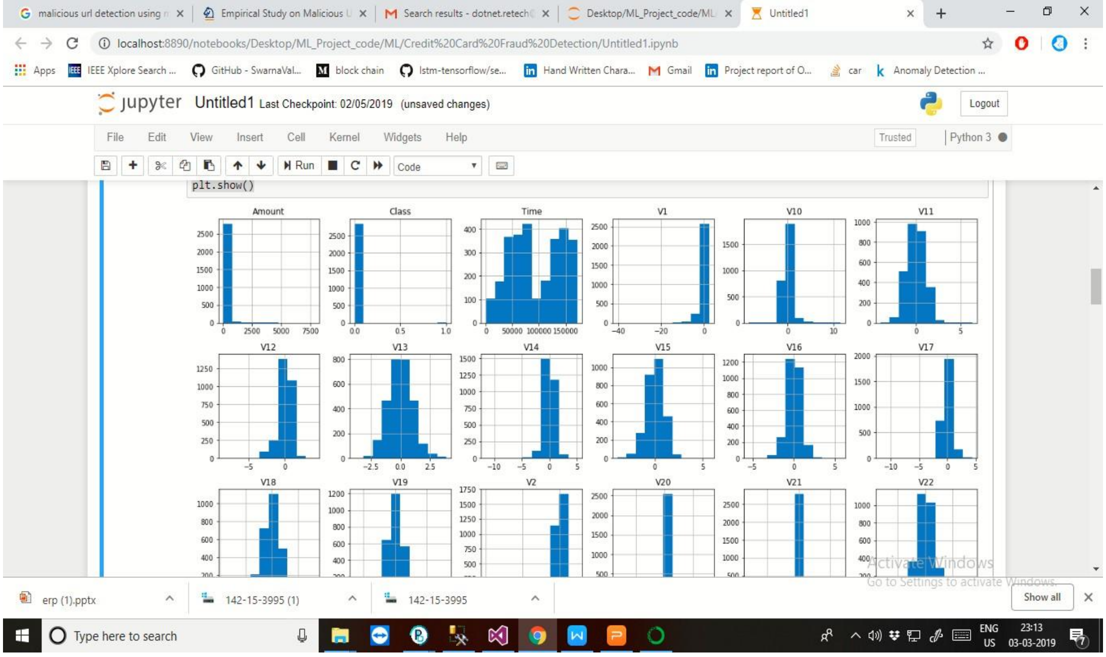
```
Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10',
       'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20',
       'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount',
       'Class'],
      dtype='object')
(28481, 31)
                Time            V1            V2            V3            V4  \
count   28481.000000  28481.000000  28481.000000  28481.000000  28481.000000
mean    94705.035216     -0.001143     -0.018290      0.000795      0.000350
std     47584.727034      1.994661      1.709050      1.522313      1.420003
min         0.000000    -40.470142    -63.344698    -31.813586     -5.266509
25%     53924.000000     -0.908809     -0.610322     -0.892884     -0.847370
50%     84551.000000      0.031139      0.051775      0.178943     -0.017692
75%    139392.000000      1.320048      0.792685      1.035197      0.737312
max    172784.000000      2.411499     17.418649      4.069865     16.715537

                 V5            V6            V7            V8            V9  \
count  28481.000000  28481.000000  28481.000000  28481.000000  28481.000000
mean      -0.015666      0.003634     -0.008523     -0.003040      0.014536
std        1.395552      1.334985      1.237249      1.204102      1.098006
min      -42.147898    -19.996349    -22.291962    -33.785407     -8.739670
25%       -0.703986     -0.765807     -0.562033     -0.208445     -0.632488
```

Figure 7.2: Data Describe

7.2.3 PREPROCESSING

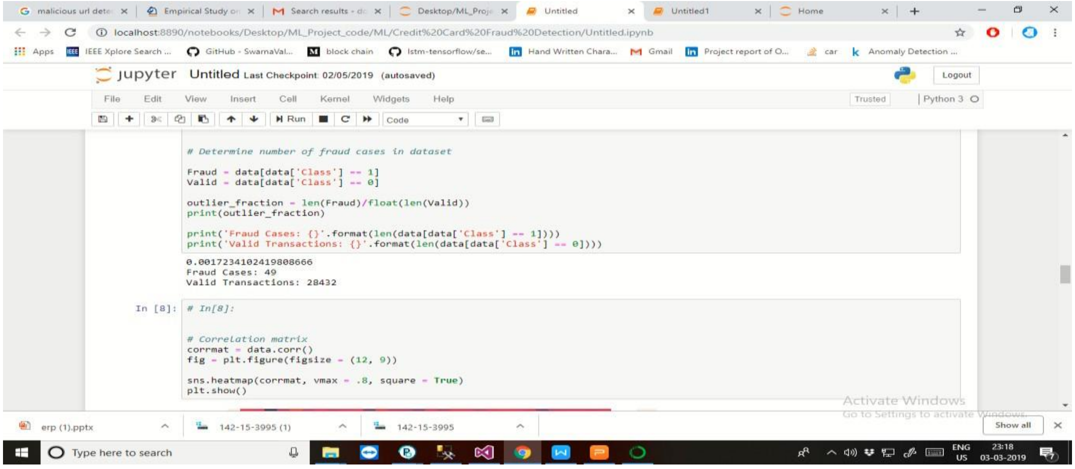
The data values has been plotted using histogram describing the numerical distribution of the data values.



```
plt.show()
```

Figure 7.3: Histogram

.2.4 FIND FRAUD



```
# Determine number of fraud cases in dataset

Fraud = data[data['Class'] == 1]
Valid = data[data['Class'] == 0]

outlier_fraction = len(Fraud)/float(len(Valid))
print(outlier_fraction)

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

```
0.0017234102419808666
Fraud Cases: 49
Valid Transactions: 28432
```

```
In [8]: # In[8]:


        # Correlation matrix
        corrmat = data.corr()
        fig = plt.figure(figsize = (12, 9))

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

localhost:8890/notebooks/Desktop/ML_Project_code/ML/Credit%20Card%20Fraud%20Detection/Untitled.ipynb
jupyter Untitled Last Checkpoint: 02/05/2019 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Activate Windows
Go to Settings to activate Windows.
Type here to search
Show all

Figure 7.4: Fraud Diagram

The above picture describes the number of fraud that has been detected by the model. The number of fraud detected has varied for two different algorithms.

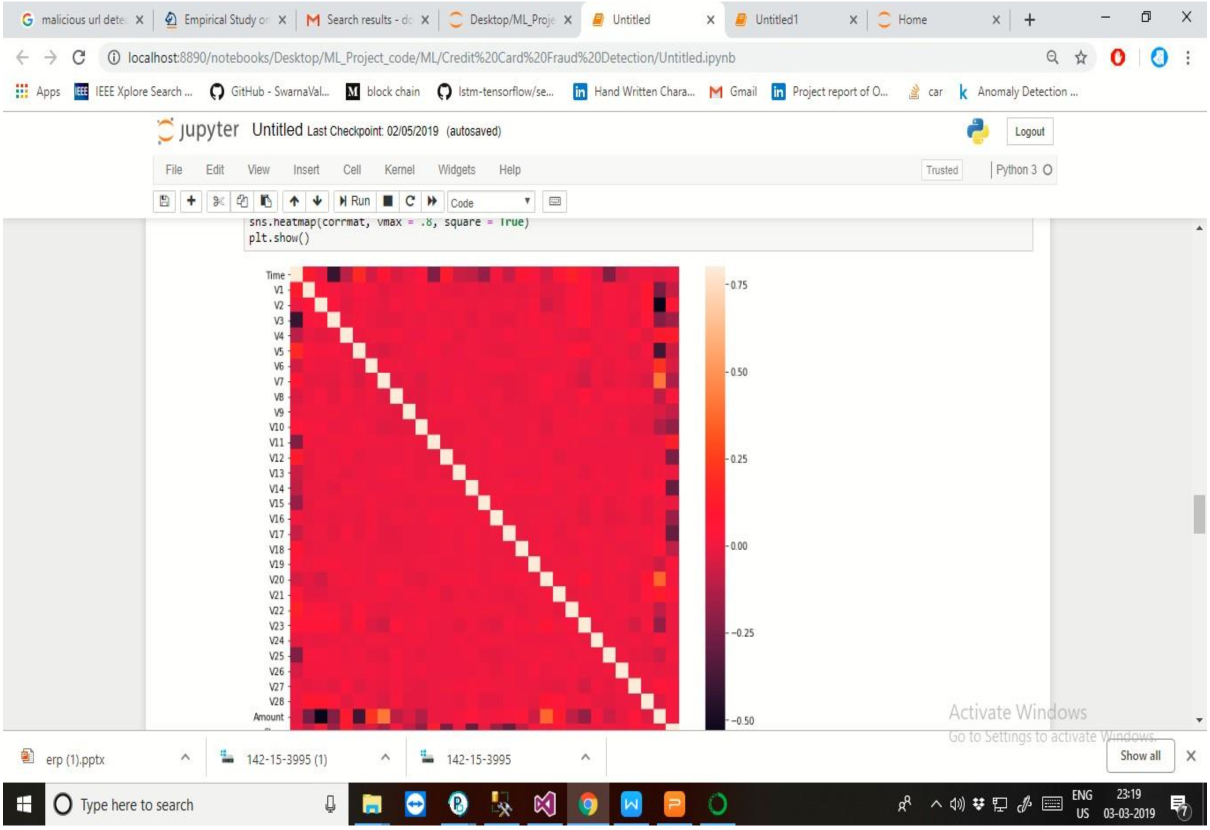
7.2.5 HEATMAP

The heat map has been plotted based on correlation matrix of the features. A correlation matrix describes the relation of features with each other. The level of correlation has been ranged from 0.0-1.0 with 1(white shade) denoting the features to be equilateral and 0(black shade) denoting the features with no interrelation.

A heatmap is graphical representation of data that uses a system of color-codigto represent different values.

Heatmap are used in various forms of analytics but are most commonly used to show user behavior on specific webpages or webpage templates.

The major benefit of bitmap visulazation is that it enables data to be presented visually which allows us to easily consume information and make more sense it.

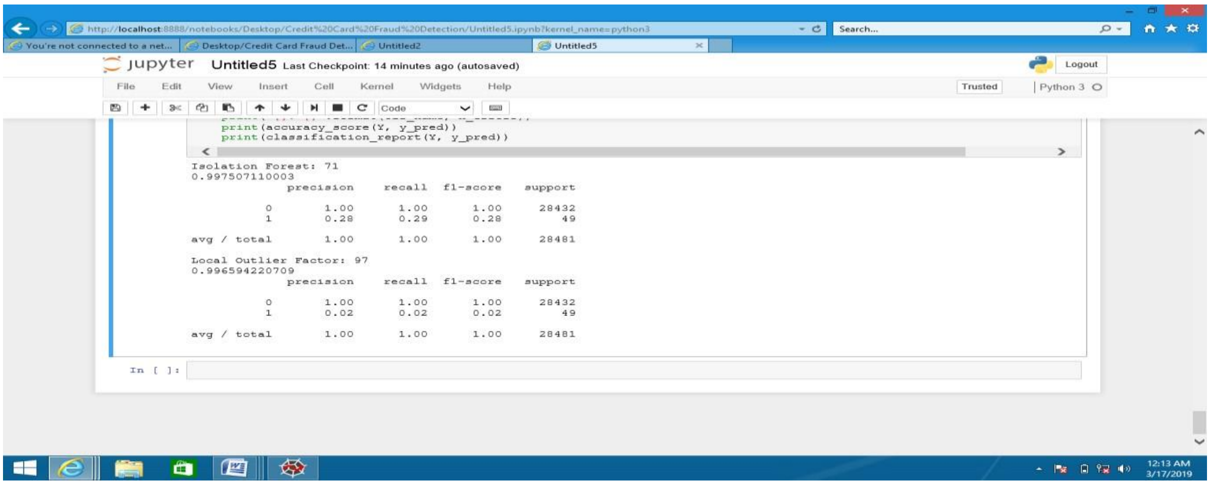


```
sns.heatmap(cormat, vmax = .8, square = True)
plt.show()
```

Figure 7.5: Heat Map Diagram

2.6 PREDICTION

The prediction that has been achieved using the Isolation Forest Algorithm and Local Outlier Factor Algorithm has been shown below



```
print(accuracy_score(Y, y_pred))
print(classification_report(Y, y_pred))
```

```
Isolation Forest: 71
0.997507110003
             precision    recall  f1-score   support

          0       1.00      1.00      1.00     28432
          1       0.28      0.29      0.28        49

avg / total       1.00      1.00      1.00     28481

Local Outlier Factor: 97
0.996594220709
             precision    recall  f1-score   support

          0       1.00      1.00      1.00     28432
          1       0.02      0.02      0.02        49

avg / total       1.00      1.00      1.00     28481
```

Figure 7.6: Accuracy Diagram

CHAPTER 8

SOFTWARE TESTING

In a generalized way, we can say that the system testing is a type of testing in which the main aim is to make sure that system performs efficiently and seamlessly. The process of testing is applied to a program with the main aim to discover an unprecedented error, an error which otherwise could have damaged the future of the software. Test cases which brings up a high possibility of discovering and error is considered successful. This successful test helps to answer the still unknown

8.2 TESTING

Table 8.1: Tabulated Results

Test Case (sample split)	Assumption	Description	Expected Output	Actual Output		Log Message
				Isolation Forest AlgorithmAlgorithm I Accuracy(%)	Local Outlier Factor - Algorithm II Accuracy(%)	
10:90	Algorithm-I will perform better	Check for accuracy at 10% training of data	99.70505	99.75071	99.65942	Success
15:85	Algorithm-II will perform better	Check for accuracy at 15% training of data	99.71675	99.75421	99.67931	Fail
20:80	Algorithm-II will perform better	Check for accuracy at 20% training of data	99.73485	99.69628	99.77352	Success
25:75	Algorithm-I will perform better	Check for accuracy at 25% training of data	99.73311	99.77107	99.69523	Success
30:70	Algorithm-I will perform better	Check for accuracy at 30% training of data	99.73425	99.77645	99.69218	Success

The test cases has been based on the following sample split (train: test) :- (10:90), (15:85), (20:80), (25:75) and (30:70).
Outlier Fraction: Describes the ratio of outlier values to the real values in the dataset

Data Shape: Describes the number of rows and columns in the training sample.

Isolation Forest Algorithm Accuracy: Describes the accuracy achieved on the test dataset using Isolation Forest Algorithm
Local Outlier Factor Accuracy: Describes the accuracy achieved on the test dataset using Local Outlier Factor

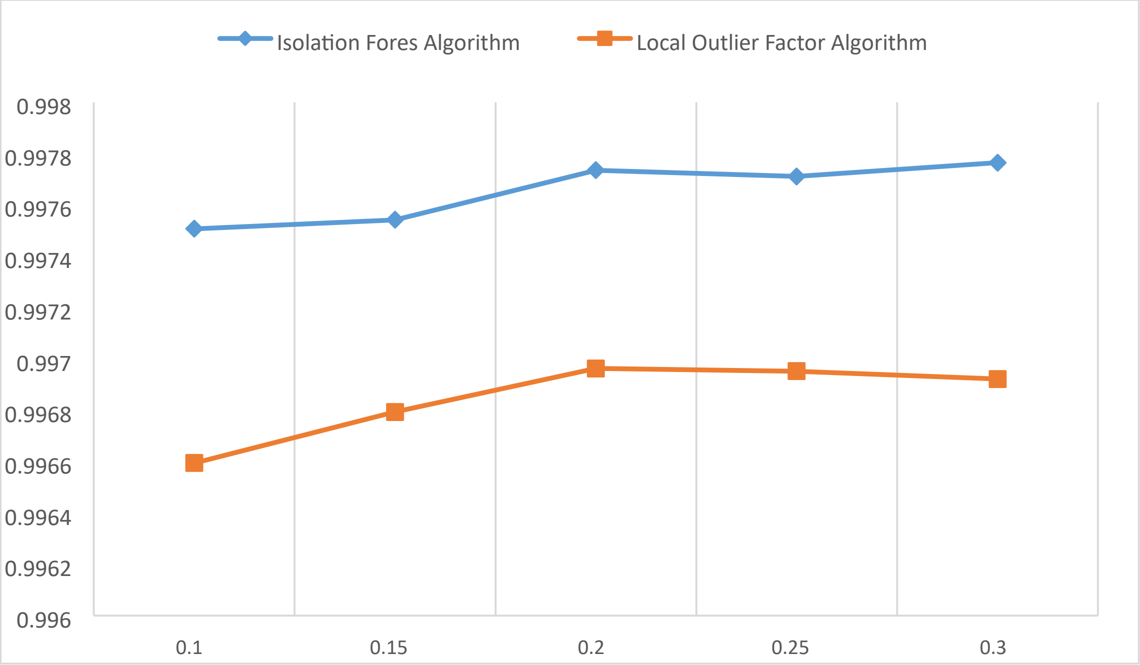


Figure 8.1: Comparison Chart

As we tested the application under different test conditions, the application gave appropriate results. The above chart depicts the accuracy based on two algorithms used, i.e. the Isolation Forest Algorithm and the Local Outlier Factor Algorithm.

The Outlier Fraction values tend to come out different under different cases used.The above chart is self-explanatory and depicts the testing results in a characterized manner.

CHAPTER 9

CONCLUSION

In this model, we discussed about credit card fraud detection using machine learning. The proposed model has been extensively tested on different types of transactions. The results were promising, almost all the fraudulent transactions could be detected successfully, and the proposed methodology has been compared with existing method and the results shows that proposed method performs superior than existing methods.

In this model, we detected the fraudulent transactions and recognized which illustrates the robustness of the proposed system. This proposed model took the trained dataset and performed classification on basis of them, if the transaction was legal then it moved to class 0 and if the transaction was fraud then it moved to class 1, and significantly improve the detection accuracy.

The proposed method works efficiently in various platform, vivid environment and is a fullfledged cross platform application. The system has depicted robust, scalable and accurate performance to the degree that efficiency is taken into consideration in the Credit Card Fraud Detection System.

The system takes into consideration various factors and has been fulfilling or meeting all the project specifications documented.

CHAPTER 10

APPLICATION AND FUTURE ENHANCEMENT

Implementation is the most critical phase to attain a fruitful system and providing the users assurance that the new system is feasible and operative. Each module is tried and tested discretely using the data and substantiated in the mode indicated as per program specification, system and the environment is tested as per user requirement.

The frequently techniques for fraud detection are Nave Bayes, support vector machine and the k - nearest neighbor algorithm. Here, this document has trained various machine learning practices and techniques used in detection of fraud in credit card and assess each methodology based on certain design measures and criteria.

Nevertheless, if there is a need to contrivance a platform that performs real-time credit card fraud detection, it is imperious to reach precisions of 95%, as the odds of false positives along with false negatives is else quite elevated to be used for business application. Impending task must subsequently be focused at exploring further relevant features to enhance, execution of a thorough optimization, and doing real-time tests. Other than the major fraud practices some other types of frauds are done such as through phishing, skimming, credit card generator etc.[6]. Also the possibility regrettably not pursued for timing issues is to refine the metrics in form of commercial forfeiture resolution system, the tenacity of model wouldn't be to capitalize on the count of transactions precisely organized, but instead minimize the costs associated with following up on fraudulent transactions based on the confidence of the model and the associated financial loss. Finally, approaches for dealing with the 'refused' examples are to be explored.

10.2 FUTURE ENHANCEMENTS

There is a very strong possibility of the system being adopted as a norm for the major banking and financial services applications as fraud detection and prevention is the major checkpoint in financial and banking sector.

The above system is also likely to be embedded in other applications based, modified as per platform-specific/application specific environment. The banks, financial and retail institutes have faced huge losses owing to cause of a robust and accurate system to predict and prevent the fraudulent transactions going on in an institution.
This in-turn affects the business capabilities and consumer trust of the company.

Thus, the organizations have moved their focus onto implementing a system which can depict inconsistent transactions, providing banks a privilege to act upon it take necessary measures.

APPENDIX

SOURCE CODE

```
import pandas import matplotlib import

seaborn import scipy print('Python:

{}'.format(sys.version)) print('Numpy:

{}'.format(numpy.__version__)) print('Pandas:

{}'.format(pandas.__version__))

print('Matplotlib:

{}'.format(matplotlib.__version__))

print('Seaborn:

{}'.format(seaborn.__version__)) print('Scipy:

{}'.format(scipy.__version__))
```

```
# In[2]:
# import the necessary
packages import numpy
as np import pandas as
pd import
matplotlib.pyplot as plt
import seaborn as sns

# In[3]:
# Load the dataset from the csv file using
pandas data = pd.read_csv('creditcard.csv')


# In[4]:
# Start exploring the dataset
print(data.columns)

# Print the shape of the data data =
data.sample(frac=0.1, random_state = 1)
print(data.shape) print(data.describe())

# Plot histograms of each
parameter  data.hist(figsize =
(20, 20)) plt.show()

# In[7]:
# Determine number of fraud cases in dataset
Fraud = data[data['Class'] == 1] Valid = data[data['Class'] ==
0] outlier_fraction = len(Fraud)/float(len(Valid))
```

```
print(outlier_fraction) print('Fraud Cases:
{}'.format(len(data[data['Class'] == 1]))) print('Valid
Transactions: {}'.format(len(data[data['Class'] == 0]))) from
sklearn.metrics import classification_report, accuracy_score
from sklearn.ensemble import IsolationForest from
sklearn.neighbors import LocalOutlierFactor

# define random
states state = 1

# define outlier detection tools to be compared
classifiers = {
"Isolation Forest": IsolationForest(max_samples=len(X),
                                   contamination=outlier_fraction,
random_state=state),
"Local Outlier Factor": LocalOutlierFactor(
n_neighbors=20,
contamination=outlier_fraction)}

# In[15]: # Fit the
model
plt.figure(figsize=(
9, 7)) n_outliers =
len(Fraud)

for i, (clf_name, clf) in enumerate(classifiers.items()):
   # fit the data and tag outliers
if clf_name == "Local Outlier
Factor":
      y_pred = clf.fit_predict(X)
scores_pred =
clf.negative_outlier_factor_    else:
      clf.fit(X)        scores_pred =
clf.decision_function(X)        y_pred =
clf.predict(X)

# Reshape the prediction values to 0 for valid, 1 for fraud.
   y_pred[y_pred == 1] = 0
y_pred[y_pred == -1] = 1
n_errors = (y_pred != Y).sum()

# Run classification metrics
print('{}: {}'.format(clf_name,
n_errors))    print(accuracy_score(Y,
y_pred))
print(classification_report(Y,
y_pred))

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

```
# In[9]:
# Get all the columns from the dataFrame
      columns = data.columns.tolist()

# Filter the columns to remove data we do not want
      columns = [c for c in columns if c not in ["Class"]]

# Store the variable we'll be
      predicting on target = "Class" X
      = data[columns]
       Y =
       data[target
       ] # Print
       shapes
       print(X.sha
       pe)
        print(Y.shape)
```

GRAPHICAL USER INTERFACE CODE

```
import sys import
numpy import pandas
import
matplotlib.pyplot as plt
```

```
import seaborn import
scipy

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


import sys,os from PyQt5 import
QtWidgets,QtGui,QtCore import csv
from PyQt5.QtWidgets import QDialog, QApplication, QPushButton,
QVBoxLayout

from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as
FigureCanvas
from matplotlib.backends.backend_qt5agg import
NavigationToolbar2QT as NavigationToolbar import matplotlib.pyplot
as plt # import LogisticRegression_diabetes
# from QtGui import QLabelBox,QMainWindow,QLineEdit

class
MainWindow(QtWidgets.QMainWindow):
def __init__(self,*args,**kwargs):
super().__init__(*args,**kwargs)
self.showMaximized()
```

```
        self.fontLiber =   QtGui.QFont("LIBER
BASKERVILLE",int(30),QtGui.QFont.Bold)
self.fontLiber1 =   QtGui.QFont("LIBER
BASKERVILLE",int(10),QtGui.QFont.Bold)
        self.titleBox=QtWidgets.QLabel("CREDIT CARD FRAUD
DETECTION")

self.titleBox.setStyleSheet("background:rgba(125,0,0,0);color:rgba(0,0,0,2
55)")         self.titleBox.setFont(self.fontLiber)

        self.uploadFileLabel=QtWidgets.QLabel("Upload
File")         self.uploadFileLabel.setFont(self.fontLiber1)
self.uploadImageLabel=QtWidgets.QLabel()
pixmap = QtGui.QPixmap('FileUpload.png')
self.uploadImageLabel.setPixmap(pixmap)

        self.uploadBtn=QtWidgets.QPushButton("Upload")
self.uploadBtn.setFont(self.fontLiber1)
self.uploadBtn.clicked.connect(self.uploadFn)



        self.uploadLayout=QtWidgets.QGridLayout()

self.uploadLayout.addWidget(self.uploadImageLabel,0,0,QtCore.Qt.AlignC
ente
r)
```

```
self.uploadLayout.addWidget(self.uploadFileLabel,0,1,QtCore.Qt.AlignCen
ter)
self.uploadLayout.addWidget(self.uploadBtn,0,2,QtCore.Qt.AlignCenter)
self.uploadLayout.setHorizontalSpacing(40)

        self.lineEdit1=QtWidgets.QLineEdit("")
self.lineEdit2=QtWidgets.QLineEdit("")
self.predictBtn=QtWidgets.QPushButton("Predict")
self.predictBtn.clicked.connect(self.predictFn)
self.predictBtn.setFont(self.fontLiber1)

        self.figure1 = plt.figure()
self.canvas1 = FigureCanvas(self.figure1)
        # self.toolbar = NavigationToolbar(self.canvas, self)
self.button1 = QPushButton('Plot')
self.button1.clicked.connect(self.plot1)

        self.figure2 = plt.figure()
self.canvas2 = FigureCanvas(self.figure2)
        # self.toolbar = NavigationToolbar(self.canvas, self)
self.button2 = QPushButton('Plot')
self.button2.clicked.connect(self.plot2)

        self.layout1=QtWidgets.QGridLayout()
self.layout1.addWidget(self.canvas1,0,0,QtCore.Qt.AlignCenter)
self.layout1.addWidget(self.button1,0,1,QtCore.Qt.AlignCenter)
self.layout1.setHorizontalSpacing(40)
self.layout2=QtWidgets.QGridLayout()
self.layout2.addWidget(self.canvas2,0,0,QtCore.Qt.AlignCenter)
self.layout2.addWidget(self.button2,0,1,QtCore.Qt.AlignCenter)
self.layout2.setHorizontalSpacing(40)

        # set the layout        self.layout =
QtWidgets.QHBoxLayout()
self.layout.addLayout(self.layout1)
self.layout.addLayout(self.layout2)



        self.mainLayout=QtWidgets.QGridLayout()
self.mainLayout.addWidget(self.titleBox,0,0,QtCore.Qt.AlignCenter)
self.mainLayout.addLayout(self.uploadLayout,1,0,QtCore.Qt.AlignCenter)
        # self.mainLayout.addLayout(self.layout,2,0,QtCore.Qt.AlignCenter)



        self.mainWidget=QtWidgets.QWidget()
self.mainWidget.setLayout(self.mainLayout)
self.setCentralWidget(self.mainWidget)
self.varText=1

    def plot1(self):
        # data = [random.random() for i in range(10)]
self.data.hist(figsize = (20, 20))
        # plt.show()
self.figure.clear()        ax =
self.figure.add_subplot(111)
```

```
ax.plot(self.data, '*-')
self.canvas.draw()


    def plot2(self):
        pass


    def uploadFn(self):
        self.fname = QtGui.QFileDialog.getOpenFileName(self, 'Open file',"",
"Image files (*.csv *.CSV ")
if(self.fname):
            self.data = pd.read_csv(self.fname)
print(self.data.columns)
            QtGui.QMessageBox.information(self, "Message", " File Upload
successfully ")



    def predictFn(self):
        self.data.hist(figsize = (20, 20))        plt.show()
self.data = self.data.sample(frac=0.1, random_state =
1)        print(self.data.shape)
print(self.data.describe())

        # V1 - V28 are the results of a PCA Dimensionality reduction to
protect user identities and sensitive features

        Fraud = self.data[self.data['Class'] == 1]
        Valid = self.data[self.data['Class'] == 0]

        outlier_fraction = len(Fraud)/float(len(Valid))
    print(outlier_fraction)

        print('Fraud Cases: {}'.format(len(self.data[self.data['Class'] == 1])))
    print('Valid Transactions: {}'.format(len(self.data[self.data['Class'] == 0])))

        corrmat = self.data.corr()
    fig = plt.figure(figsize = (12, 9))

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

        # In[9]:



        # Get all the columns from the dataFrame
    columns = self.data.columns.tolist()

        # Filter the columns to remove data we do not want
    columns = [c for c in columns if c not in ["Class"]]

        # Store the variable we'll be predicting on
    target = "Class"

X     = self.data[columns]
Y     = self.data[target]
```

```
        # Print shapes
print(X.shape)
print(Y.shape)


        # # Unsupervised Outlier Detection
        #
        # Now that we have processed our data, we can begin deploying our machine learning algorithms.  We will use the following techniques:
        #
        # **Local Outlier Factor (LOF)**
        #
        # The anomaly score of each sample is called Local Outlier Factor. It measures the local deviation of density of a
        # given sample with respect to its neighbors. It is local in that the anomaly score depends on how isolated the
        # object is with respect to the surrounding neighborhood.
        #
        #
        # **Isolation Forest Algorithm**
        #
        # The IsolationForest isolates observations by randomly selecting a feature and then randomly selecting
        # a split value between the maximum and minimum values of the selected feature.
        #
        # Since recursive partitioning can be represented by a tree structure, the number of splittings required to
        # isolate a sample is equivalent to the path length from the root node
to the terminating node.
        #
        # This path length, averaged over a forest of such random trees, is a
measure of normality and our decision function.
        #
        # Random partitioning produces noticeably shorter paths for
anomalies.
Hence, when a forest of random trees
        # collectively produce shorter path lengths for particular samples,
they are highly likely to be anomalies.


        from sklearn.metrics import classification_report,
accuracy_score         from sklearn.ensemble import
IsolationForest        from sklearn.neighbors import
LocalOutlierFactor

        # define random
states        state = 1

        # define outlier detection tools to be compared
classifiers = {
            "Isolation Forest":
IsolationForest(max_samples=len(X),
contamination=outlier_fraction,
random_state=state),            "Local Outlier Factor":
```

```
LocalOutlierFactor(                n_neighbors=20,
contamination=outlier_fraction)}


        # In[15]:



        # Fit the model
        #plt.figure(figsize=(9, 7))
n_outliers = len(Fraud)


        for i, (clf_name, clf) in enumerate(classifiers.items()):

            # fit the data and tag outliers
if clf_name == "Local Outlier Factor":
                y_pred = clf.fit_predict(X)
scores_pred = clf.negative_outlier_factor_
else:
                clf.fit(X)                scores_pred =
clf.decision_function(X)                y_pred =
clf.predict(X)

            # Reshape the prediction values to 0 for valid, 1 for fraud.
            y_pred[y_pred == 1] = 0
y_pred[y_pred == -1] = 1

            n_errors = (y_pred != Y).sum()
```

```
           # Run classification metrics
print('{}: {}'.format(clf_name, n_errors))
print(accuracy_score(Y, y_pred))
          # print(classification_report(Y, y_pred))
       #plt.show()


if __name__ == '__main__':
   currentApp = QtWidgets.QApplication(sys.argv)
currentWindow = MainWindow()
currentWindow.show()

   sys.exit(currentApp.exec_())
```

BIBLIOGRAPHY

[1] V. Bhusari S. Patil, “Study of Hidden Markov Model in Credit Card Fraudulent

Detection”, International Journal of Computer Applications (0975 – 8887) Volume 20– No.5, April 2011

[2] Priya Ravindra Shimpi, Prof. Vijayalaxmi Kadroli Angrish, “Survey on Credit Card

Fraud Detection Techniques”, International Journal Of Engineering And Computer Science ISSN: 2319-7242 [3] Salvatore J. Stolfo, Wei Fan, WenkeLee, “Cost-based

Modeling for Fraud and Intrusion Detection Results from the JAM Project”, In

Proceedings of the ACM SIGMOD Conference on Management of Data, pages 207– 216, 2014.

[3] Delamaire. L. Abdou, HAH and Pointon. J,”Credit card f raud and detection techniques”, Banks and Bank Systems, Volume 4, Issue 2, 2009

[4] Suman, Nutan, “Review Paper on Credit Card Fraud Detection”, International Journal of Computer Trends and Technology (IJCTT) – volume 4 Issue 7–July 2013.

[5] Renu, Suman, “Analysis on Credit Card Fraud Detection Methods”, International

Journal of Computer Trends and Technology (IJCTT) – volume 8 number 1 – Feb 2014.

[7] Sushmito Ghosh and Douglas L. Reilly, “Credit Card Fraud Detection with a NeuralNetwork” Proc. IEEE First Int. Conf. on Neural Networks, 2014.

[6] Deepak Pawar, SwapnilRabse, Sameer Paradkar, NainaKaushi, “Detection of Fraud in Online Credit Card Transactions”, International Journal of Technical Research and Applications e-ISSN: 2320-8163.