## ABSTRACT

Credit card fraud detection is the most frequently occurring problem in the present world. This is due to the rise in both online transactions and e-commerce platforms. Credit Card Fraud generally happens when the card was stolen and used by unauthorized person for unauthorized purposes or even when the fraudster uses the credit card information for their use. Due to the increase in cybercrimes, the credit card fraud detection system was introduced to detect the fraudulent users. In this project, a model is built using machine learning classification algorithms such as Logistic Regression, Random Forest, Decision Tree, Naïve Bayes and Passive Aggressive algorithm for detecting the credit card frauds. The Credit Card dataset is collected from the Kaggle website and is uploaded and examined for fraud and non-fraud transactions. The model is trained and tested over the different splits ratios such as 6:4, 7:3 and 8:2 and is classified using different machine learning algorithms and the performance is measured using accuracy. The comparison between the different algorithms with different training and testing dataset is performed and as the best algorithm for each training and testing data are specified.



Keywords: Credit Card Fraud detection system, Random Forest, Logistic regression, Decision Tree, Naive Bayes Classifier, Passive Aggressive.

# 1. INTRODUCTION

## 1.1 OVERVIEW

At the current state of the world, financial organizations expand the availability of financial facilities by employing of innovative services such as credit cards, Automated Teller Machines (ATM), internet and mobile banking services. Besides, along with the rapid advances of e-commerce, the use of credit card has become a convenience and necessary part of financial life. Credit card is a payment card supplied to customers as a system of payment.

Credit card plays a very important rule in today's economy. It becomes an unavoidable part of household, business and global activities. Although using credit cards provides enormous benefits when used carefully and responsibly, significant credit and financial damages may be caused by fraudulent activities. Credit cards have been the main instruments for financial transactions in all online commercial activities since more decades. This makes credit card-based payment systems vulnerable to frauds. Since then credit card fraud has incurred losses of billions of credits and is increasing day by day.

Billions of losses are caused every year by the fraudulent credit card transactions. Fraud is old as humanity itself and can take an unlimited variety of different forms. The PWC global economic crime survey of 2017 suggests that approximately 48% of organizations experienced economic crime. Therefore, there's positively a requirement to resolve the matter of credit card fraud detection. The use of credit cards is prevalent in modern day society and credit card fraud has been kept on growing in recent years. Hugh financial losses have been fraudulent affects not only merchants and banks, but also individual person who is using the credits. Fraud may also affect the reputation and image of a merchant causing nonfinancial losses that, though difficult to quantify in the short term, may become visible in the long period. For example, if a cardholder is victim of fraud with a precise company, he might no longer trust their business and opt for a rival.

Fig 1.1 shows the Credit card fraud detection system which is a serious growing problem that occurs as unauthorized usage of card information, unexpected transaction behaviour, or any kind of transaction on an inactive card. A credit card is a small plastic card issued by a financial company that authorizes the cardholder to use it for payment of goods and services. The amount of purchase is recorded in the user's account and he has to repay the borrowed sum as well as any other charges agreed upon as understanding between the card company and the user. Credit card fraud is a wide-ranging term for theft and fraud committed using a Credit card as a fraudulent source of funds.

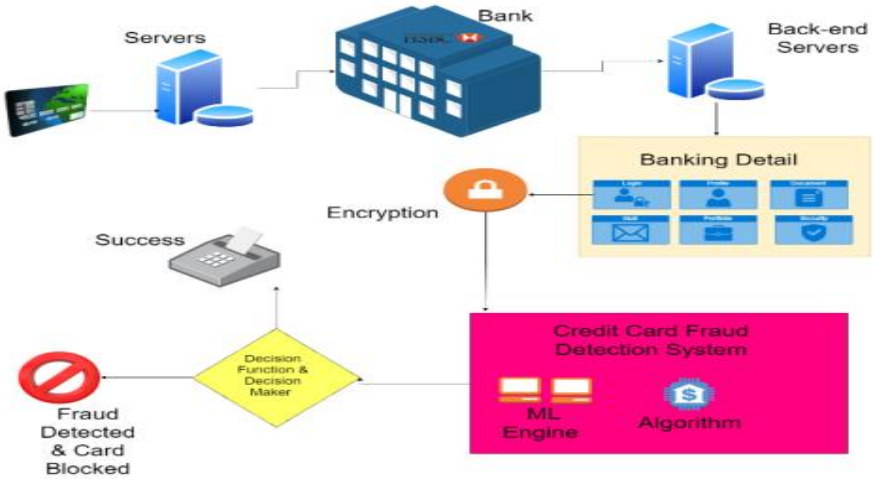



**Fig 1.1 Credit Card Fraud detection system**

### 1.1.1 Types of Credit Card Frauds

Main motive of credit card fraud is to illegally obtain physical possession or information of card. However, the modus operandi may differ in various cases. On the basis of instances of frauds that have been discussed in financial information sources, they can be categorized into two main categories described in this section.

#### 1.1.1.1 Obtaining Physical Cards Illegally

Fig 1.1.1.1 shows all the types of frauds by obtaining physical cards illegally are:

**1. Application Fraud:** Application fraud is when someone obtains a credit card using fake or false information by forging documents and providing fake telephone numbers of residence and place of employment.

**2. Lost and Stolen Card Fraud:** Physical security of credit card is an important factor. If a card is not adequately protected, then it can get accidently lost and fall in the hands of perpetrators. In some cases, an unattended card may be stolen with ill intention. These frauds can be used to launch other frauds.

**3. Counterfeit Cards:** Such frauds are committed through skimming actual credit card information and creating a forged magnetic tape having information about credit card.

**4. Mail Non-receipt Fraud:** This fraud is also known as "never received issue" or "intercept fraud." It occurs when a user is expecting a new card or a replacement, but a criminal gets its possession before the actual user and starts using it.

**5. Assumed Identity:** All credit card issuance is checked for correct identification of the person to whom the card is being provided. In absence of fool-proof authentication mechanism, a fraudster may impersonate a naive person by obtaining and producing fake address proof and identity document.

**6. Doctored Cards:** One of the ways of fraud is to tamper information of an existing card with the help of a powerful electromagnet.

**7. Fake Cards:** Credit cards may be cloned by copying all the information encoded in magnetic strip and pasting into a new strip to get a fake card. Creation of fake cards can be done by someone who is skilled enough to forge the magnetic strip and the chip and break the complex security and even holograms of real credit cards.

**8. Account Takeover:** Such type of fraud is usually carried out online, where the fraudster talks to the credit card company to replace card by providing relevant documents and information. These attack vectors to physically obtain credit card in an illegal way have been summarized in below.


**Fig 1.1.1 Types of frauds by obtaining physical cards illegally**

### 1.1.1.2 Obtaining Card Information Illegally

Fig 1.1.1.2 shows that all types of frauds by obtaining credit card information illegally are:

**1. Credit Card Imprints**: Credit card imprints are taken as a measure of security deposit for a service usage like hotel or car rentals. A dishonest service provider or its employee may skim the information, which can be used in fraudulent transactions.

**2. CNP (Card-Not-Present) Fraud:** Card-Not-Present is a type of credit card fraud executed by obtaining card information like a cardholder's name, billing address, account number, three-digit security code, and card expiration date. Such theft of credit card data may occur through online phishing, tampered swipe machines, or shoulder surfing. CNP is generally used in online transactions where the perpetrator does not have to be physically present.

**3. Card ID Theft:** It is the most difficult fraud to detect where the details of credit card become known to a criminal, and this information is used to take over a card account or open a new one. Identity theft constitutes 71% of the most common type of fraud.

**4. Clean Frauds:** To commit this category of frauds, fraudster does a lot of homework in collecting the user's actual details and working principles of underlying Fraud Detection System. The system does not suspect such a transaction and thus the fraud occurs in a clean manner.

**5. Friendly Fraud:** These frauds are about repudiation. In absence of proper online authentication mechanisms, actual user may deny making a purchase after doing it. The user claims that the card has been stolen before the said transaction.

**6. Triangle Fraud:** As the name suggests, this fraud takes place in three recursive steps. The first step is to create a fake ecommerce store or website that offers popular items at very low price. Users are tempted to make purchases at these sites and their credit card details are stolen. In the second step, goods are purchased from other merchants using previously stolen cards and delivered to the purchaser. The third step is to use the stolen information to make purchases elsewhere. This indirection can help the attack remain hidden for a long time.


**Fig 1.1.2 Types of frauds by obtaining credit card information illegally**

Credit Card fraud is a wide-ranging term for theft and fraud committed using a credit card as a fraudulent source of funds in given transaction. Ever since starting my journey into data science, I have been thinking about ways to use data science for good while generating value at the same time. Thus, when I came across this data set on Kaggle dealing with credit card fraud detection, I was immediately hooked. The data set has 31 features, 28 of which have been anonymized and are labeled V1 through V28. The remaining three features are the time and the amount of the transaction as well as whether that transaction was fraudulent or not. Before it was uploaded to Kaggle, the anonymized variables had been modified in the form of a PCA (Principal Component Analysis). Furthermore, there were no missing values in the data set. Equipped with this basic description of the data, let's jump into some exploratory data analysis.

With the rise of e-commerce in the past decade, the use of credit cards has increased dramatically. The number of credit card transactions in 2011 in Malaysia was at about 320 million, and increased in 2015 to about 360 million. Along with the rise of credit card usage, the number of fraud cases has been constantly increased. While numerous authorization techniques have been in place, credit card fraud cases have not hindered effectively. Fraudsters favour the internet as their identity and location are hidden. The rise in credit card fraud has a big impact on the financial industry.

Credit card fraud is a huge ranging term for theft and fraud committed using or involving at the time of payment by using this card. The purpose may be to purchase goods without paying, or to transfer unauthorized funds from an account. Credit card fraud is also an add on to identity theft. As per the information from the United States Federal Trade Commission, the theft rate of identity had been holding stable during the mid 2000s, but it was increased by 21 percent in 2008. Even though credit card fraud, that crime which most people associate with ID theft, decreased as a percentage of all ID theft complaints In 2000, out of 13

billion transactions made annually, approximately 10 million or one out of every 1300 transactions turned out to be fraudulent.

Also, 0.05% (5 out of every 10,000) of all monthly active accounts was fraudulent. Today, fraud detection systems are introduced to control one-twelfth of one percent of all transactions processed which still translates into billions of dollars in losses. Credit Card Fraud is one of the biggest threats to business establishments today. However, to combat the fraud effectively, it is important to first understand the mechanisms of executing a fraud. Credit card fraudsters employ a large number of ways to commit fraud. In simple terms, Credit Card Fraud is defined as "when an individual uses another individuals' credit card for personal reasons while the owner of the card and the card issuer are not aware of the fact that the card is being used".

Card fraud begins either with the theft of the physical card or with the important data associated with the account, including the card account number or other information that necessarily be available to a merchant during a permissible transaction. Card numbers generally the Primary Account Number (PAN) are often reprinted on the card, and a magnetic stripe on the back contains the data in machine-readable format. It contains the following Fields:

- Name of card holder
- Card number
- Expiration date
- Verification/CVV code
- Type of card

There are more methods to commit credit card fraud. Fraudsters are very talented and fast moving people. In the Traditional approach, to be identified by this paper is Application Fraud, where a person will give the wrong information about himself to get a credit card. There is also the unauthorized use of Lost and Stolen Cards, which makes up a significant

area of credit card fraud. There are more enlightened credit card fraudsters, starting with those who produce Fake and Doctored Cards; there are also those who use Skimming to commit fraud. They will get this information held on either the magnetic strip on the back of the credit card, or the data stored on the smart chip is copied from one card to another. Site Cloning and False Merchant Sites on the Internet are getting a popular method of fraud for many criminals with a skilled ability for hacking. Such sites are developed to get people to hand over their credit card details without knowing they have been swindled.There is rapid increase in the credit card transaction which as led to substantial growth in fraudulent cases. Many data mining and statistical methods are used to detect fraud. Many fraud detection techniques are implemented using artificial intelligence, pattern matching. Detection of fraud using efficient and secure methods are very important.

The global credit card fraud in 2015 reached to a staggering USD $21.84 billion. Loss from credit card fraud affects the merchants, where they bear all costs, including card issuer fees, charges, and administrative charges. Since the merchants need to bear the loss, some goods are priced higher, or discounts and incentives are reduced. Therefore, it is imperative to reduce the loss, and an effective fraud detection system to reduce or eliminate fraud cases is important. There have been various studies on credit card fraud detection. Machine learning and related methods are most commonly used, which include artificial neural networks, rule-induction techniques, decision trees, logistic regression, and support vector machines. These methods are used either standalone or by combining several methods together to form hybrid models.

## 1.2 MOTIVATION

Nowadays most of the transactions take place online, meaning that credit cards and other online payment systems are involved. It is convenient both for the company and for the consumer. Consumers save time because they don't have to go to the store to make their purchases and companies save money by not owing physical stores and avoiding expensive rental payments. It seems that the digital age brought some highly useful features which changed the way that both companies and consumers interact with each other but with one cost. The customers prefer the most accepted payment mode via credited card for the convenient way of paying bills, online shopping is easiest way. At the same time the fraud transaction risks using credit card is a main problem which should be avoided. So there are number of deep learning techniques to avoid these risks effectively. In existing research they modelled the sequence of operations in credit card fraud transaction processing using a single algorithm and show how it can be used for the detection of frauds. In order to provide better accuracy by using machine learning algorithms in fraud detection in proposed work.

## 1.3 PROBLEM DEFINITION

The Credit Card Fraud Detection Problem includes modelling past credit card transactions with the knowledge of the ones that turned out to be a fraud. This model is then used to identify whether a new transaction is fraudulent or not. We enumerate different properties a fraud detection system should have in order to generate proper results are the system should be able to handle skewed distributions, adapt themselves to new kinds of fraud, overlapping data and there should be a proper means to handle the noise. Classification techniques like Random Forest, Logistic Regression, Decision Tree, Naive Bayes Classifier and Passive Aggressive algorithms are used. The performance is measured using accuracy and the classification model with high accuracy is defines as the best algorithm for detecting the frauds.

## 1.4 AIM OF THE PROJECT

The aim of the project is to develop a credit card fraud detection system with high accuracy using machine learning algorithms that helps in detecting the frauds easily. It can also used to detect 99% of the fraudulent transactions while minimizing the incorrect fraud classifications.

## 1.5 ORGANIZATION OF REPORT

The central idea of the report is to develop a credit card fraud detection system using deep learning algorithms.

**Chapter 1** consists of brief introduction about the credit card fraud detection systems and the types of frauds followed by the motivation behind the report, problem definition and the aim of the project.

**Chapter 2** consists of the literature survey that briefly explains the works done previously. This is the major part of research as it improves the quality of results.

**Chapter 3** consists of the system requirements that are necessary for implementing the project.

**Chapter 4** consists of the methodology and the machine learning algorithms that are used to build the classification model.

**Chapter 5** consists of the implementation process of the project, which consists of collecting data, uploading them and splitting the data and then building the classification model.

**Chapter 6** consists of results that shows the accuracies and the comparison graph of the accuracy for all machine learning algorithms

**Chapter 7** gives the conclusion of the overall report.

# 2. LITERATURE SURVEY

## 2.1 Credit Card Fraud Detection using Machine Learning and Data Science

Credit Card Fraud Detection using Machine Learning and Data Science is a journal published in the year 2019 in the International Journal of Engineering Research and Technology on Computer Science and Engineering by S P Maniraj and Aditya Saini, Swarna Deep Sarkar and Shadab Ahmed.

Fig 2.1 shows a credit card fraud detection system is developed in order to detect the frauds using machine learning techniques. In this paper, they have focused on analysing and pre-processing data sets as well as the deployment of multiple anomaly detection algorithms such as Local Outlier Factor and Isolation Forest algorithm on the PCA transformed Credit Card transaction data are used and the performance is measured using precision, recall, f1-Score, and accuracy.

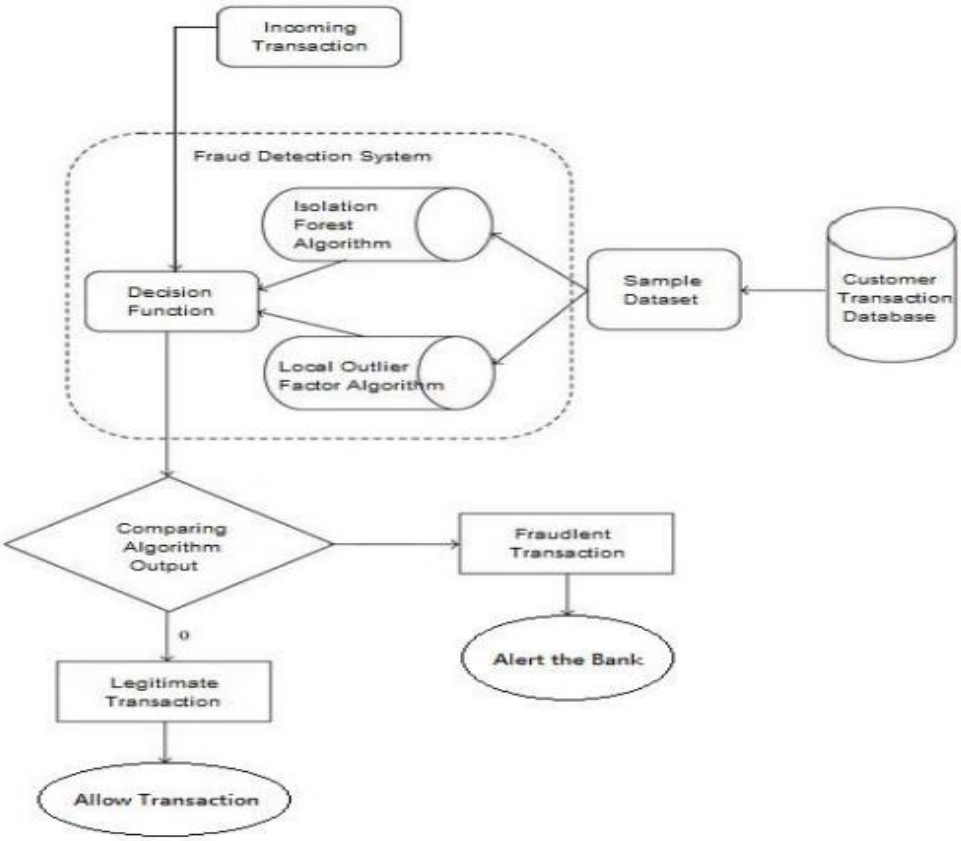



**Fig 2.1 Credit card fraud detection using local outlier and Isolation Forest algorithms**

Here, the ‘CreditCard’ dataset from Kaggle, it can be used for implementation and to evaluate the effectiveness and performance of detecting credit card fraud transaction using Local Outlier factor and Isolation Forest Algorithm. In these dataset, there are 31 columns out of which 28 are named as v1-v28 to protect sensitive data. The other columns represent time, amount and class. Time shows the time gap between the first transaction and the following one. Amount is the amount of money transacted. Class 0 represents a valid transaction and 1 represents a fraudulent one.

And now, the model is build using these multiple anomaly detection algorithms such as isolation forest, Local outlier factor algorithms and by evaluating the performance it is proved that one is best as Isolation forest can be classifies the system and does reach over 99.6% accuracy, its precision remains only at 28% when a tenth of the data set is taken into consideration. However, when the entire dataset is fed into the algorithm, the precision rises to 33%. The accuracy and detection rate of this model is high when compared to other algorithms.

### 2.2 An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine

An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine is a conference paper published in the year 2020 in International Conference on Information Technology by Altyeb Altaher Taha and Sharaf Jameel Malebary.

Fig 2.2 shows a model is developed by training and testing data by data preprocessing and features selection and by using the algorithm like GBM (Gradient Boosting Machine) model and five-fold cross and the performance is measured using the accuracy of the model.

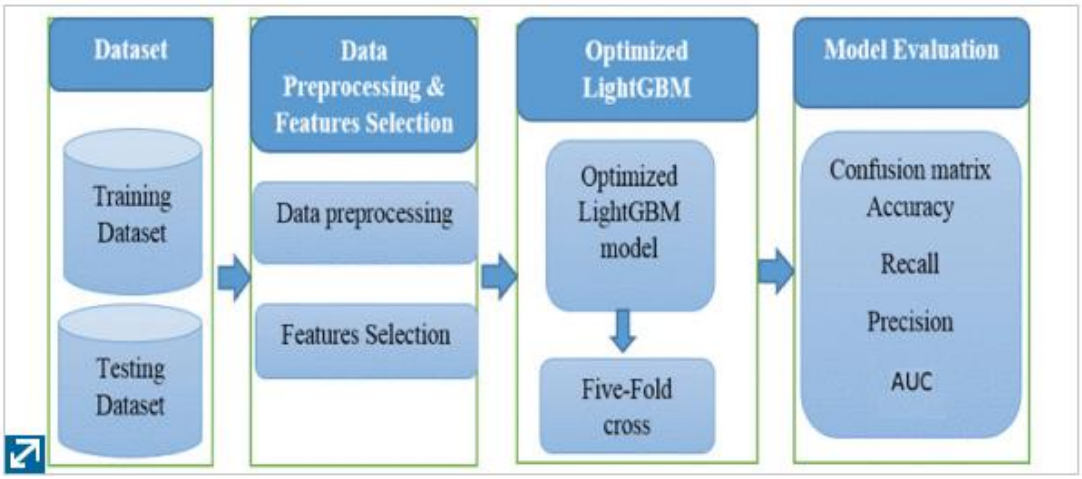



**Fig 2.2 Intelligent approach to credit card fraud detection**

UCSD-FICO Data Mining Contest 2009 dataset, which is a real data set of e-commerce transactions. The data set consists of 94,683 transactions, 2,094 of which are fraudulent. This datasets are taken and are cleaned and pre-processed to extract the features and by using GBM model can achieved the highest AUC (92.88%), Accuracy, Precision and F1-score. the importance and value of adopting an efficient parameter optimization strategy for enhancing the predictive performance of the proposed approach.

### 2.3 An Experimental Study with Imbalanced Classification Approaches for Credit Card Fraud Detection

An Experimental Study with Imbalanced Classification Approaches for Credit Card Fraud Detection is a conference paper published in the year 2019 in International Conference on Computer Science by Sara Makki, Zainab Assaghir, Yehia taher, rafiqul Haque, Mohand Said Hacid and Hassan Zeineddine.

Here, a credit card fraud detection is developed in order to detect the frauds using eight different machine learning classification algorithms like C4.0 algorithm, Naive bayes, Logistic Regression, SVM, ANN, Bayesian belief

network , Artificial Immune system and KNN are used and the performance is measured using precision, recall, f1-Score, and accuracy.

The dataset used in our experiment contains credit card fraud labeled data. It contains ten million credit card transactions described by 8 variables listed here: custID is an auto increasing integer value that represents the customer ID. This variable is removed later as it has no relevance for detecting fraud as Gender, state , cardholder , balance, numTrans, numIntTrans, creditLine, fraudRisk and taking the values 0 denoting legitimate transaction, and 1 denoting fraudulent transaction.

LR (Logistic Regression), C5.0 decision tree algorithm, SVM and ANN are the best methods according to the three considered performance measures (Accuracy, Sensitivity and AUPRC). Our experimental study revealed that the approaches normally used to solve imbalance problems may have unpleasant consequences when the imbalance is extreme, such as generating a significant number of false positives.

**Table 2.1** Comparison between the research papers

| S. No | Title of the paper | Name of Journal, Year of Publication & authors | Techniques or methodology used | Performance parameters used | Advantages | Draw backs |
|---|---|---|---|---|---|---|
| 1 | Credit Card Fraud Detection using Machine Learning and Data Science | International Journal of Engineering Research and Technology on Computing Science, and Engineering **Year:** 2019 **Authors:** S P Maniraj and Aditya | Local Outlier Factor Algorithm<br><br>Isolation Forest algorithm | Performance metrics precision, recall, f1-Score, and accuracy are used | Comparison between two algorithms is performed<br><br>Isolation forest is specified as the best detection algorithm with an accuracy of 93% | In high dimensions, the Local Outlier Factor algorithm detection accuracy gets effected<br><br>Precision is also |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | Saini, Swarna Deep Sarkar and Shadab Ahmed. | | | | low Local Outlier factor algorithm |
| 2 | An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine | International Conference on Information Technology **Year:** 2020 **Authors:** Altyeb Altaher Taha and Sharaf Jameel Malebary | It consists of three main phases: pre-processing, feature selection, and classification GBM (Gradient Boosting Machine) And five fold across are used in classification | Confusion matrix, Accuracy, Recall, Precision and AUC are calculated. | The accuracy and AUC of this GBM algorithm is high when compared to other algorithms | It has more complexity to detect fraud detection |
| 3 | An Experimental Study with Imbalanc | International Conference on Computer Science | C4.0 algorithm, Naïve bayes, Logistic Regression, SVM, ANN, | Performance metric such as accuracy, Recall, F-measure | This model of a LR (Logistic Regression), C5.0 decision tree algorithm, SVM and ANN | The time consumption of the |

| | | | | | | |
|---|---|---|---|---|---|---|
| | ed Classification Approaches for Credit Card Fraud Detection | **Year:** 2019 **Authors:** Sara Makki, Zainab Assaghir, Yehia taher, rafiqul Haque, Mohand Said Hacid and Hassan Zeineddine | Bayesian belief network, Artificial Immune system and KNN are used | and Precision are used. | are the best methods according to the three considered performance measures (Accuracy, Sensitivity and AUPRC). | NSA training phase is difficult to deal with.<br><br>Difficult to finding optimized parameters such as the costs for CS approaches, k for the KNN |

# 3. SYSTEM REQUIREMENT SPECIFICATIONS

The Systems Requirements are of two types as below namely hardware and software requirements.

## 3.1 SOFTWARE REQUIREMENTS

- Operating System : Windows 10.
- Coding Language : Python.
- Integrated Development Environment : PyCharm

## 3.2 HARDWARE REQUIREMENTS

- System : Intel i3 processor.
- Hard Disk : 1 TB.
- Monitor : 15 VGA Colour.
- Ram : 8GB

# 4. METHODOLOGY

For a Credit Card Fraud detection System to work, the data has to be collected as csv file of these frauds and non-frauds transactions from the creditcard dataset of kaggle. Then, it can be splitting into training and testing the data. Building the Classification models using different machine learning algorithms like Random Forest, Logistic Regression, Naive Bayes, GBM, Passive Aggressive and by comparing all the performance measures of accuracy as results are analyzed. The block diagram of the log file-based intrusion detection system using machine learning is shown below in the Fig 4.1.

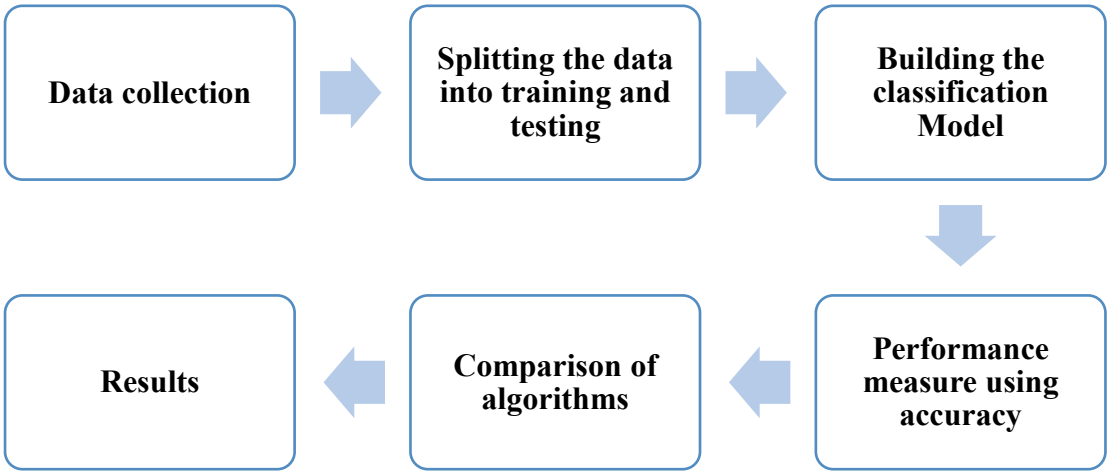



**Fig 4.1 Block diagram for Credit Card Fraud Detection**

## 4.1 DATA COLLECTION

Data can be collected as a 'creditcard' dataset of csv file from the kaggle website. In this dataset, there are 31 columns out of which 28 are named as v1-v28

to protect sensitive data. The other columns represent time, amount and class. Time shows the time gap between the first transaction and the following one. Amount is the amount of money transacted. Class 0 represents a valid transaction and 1 represents a fraudulent one.

## 4.2 SPLITTING THE DATA INTO TRAINING AND TESTING

The data has been splitting into training and testing data to build the classification model by using different machine learning algorithms. The total dataset size of transactions is 2,84,807. It can be splits into 6:4, 7:3 and 8:2 ratios of training and testing of the data.

## 4.3 BUILD THE CLASSIFICATION MODEL

Classification is a process of dividing the dataset into different categories by adding labels based on some condition. Classification is usually done to perform predictive analytics i.e., to predict whether the activity is fraud or non-fraud. The supervised classification algorithm is a classification algorithm where the labelled data is used to train the model. The classification algorithms that are used in project are:

### 4.3.1 Random Forest Algorithm

Random forest is a supervised machine learning algorithm which is used for both classification and regression. Here in this project, random forest is used for classification. Here, the labelled data is trained to the model and the new data is given for classification. Radom forest, the name itself says that it's a forest that is a combination of trees. So, basically random forest is a combination of decision tress. It is an ensemble model which uses combination of different supervised machine learning algorithm. Random Forest breaks a complex problem into smaller ones and solves them.

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. 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. 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.

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.

The Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

- It overcomes the problem of overfitting by averaging or combining the results of different decision trees.
- Random forests work well for a large range of data items than a single decision tree does.
- Random forest has less variance then single decision tree.

- Random forests are very flexible and possess very high accuracy.
- Scaling of data does not require in random forest algorithm. It maintains good accuracy even after providing data without scaling.
- Random Forest algorithms maintains good accuracy even a large proportion of the data is missing.
- Construction of Random forests are much harder and time-consuming than decision trees.
- More computational resources are required to implement Random Forest algorithm.
- It is less intuitive in case when we have a large collection of decision trees.
- The prediction process using random forests is very time-consuming in comparison with other algorithms.

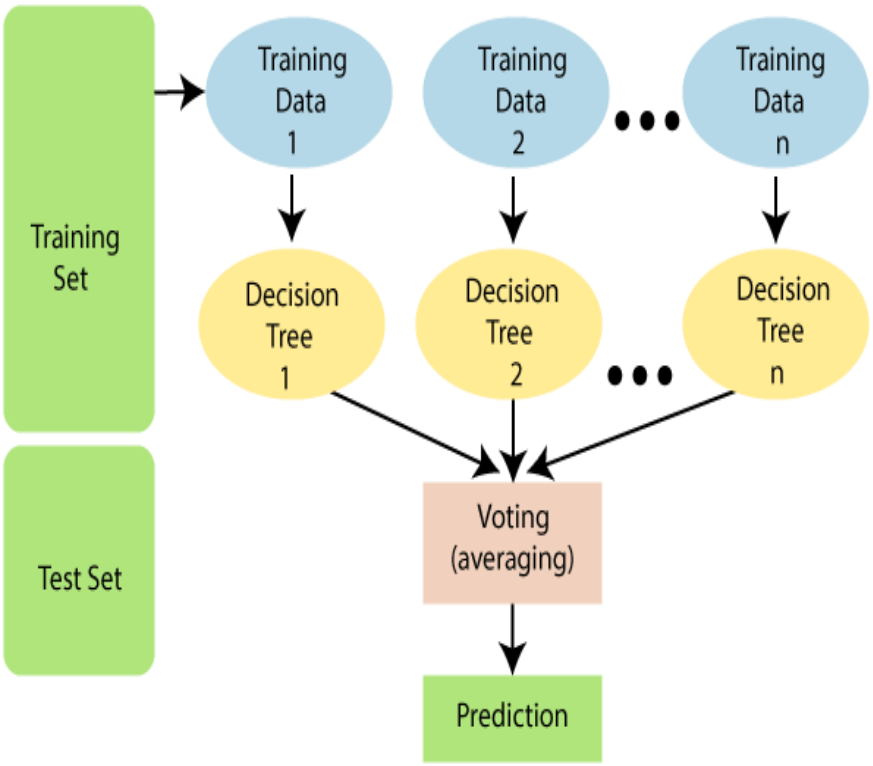



**Fig 4.3.1 Random Forest Algorithm**

### 4.3.2 Logistic Regression Algorithm

Logistic regression is a supervised regression algorithm used for both classification and regression where the dependent variable (output) is categorical or discrete i.e., 0/1, True/ False, Yes/No etc., Logistic regression is a modified version of linear regression where a straight line in linear regression is converted into a sigmoid curve also known as S-Curve as the output of logistic regression is either 1 or 0. Here, in this algorithm a threshold value is set and if the test values is greater than the given threshold value it is classified as 1 and if the value is less than the threshold value then it is set to 0. The equation of logistic regression is

$$\hat{Y} = \frac{e^{b_0+b_1X_1+\cdots+b_kX_k}}{e^{b_0+b_1X_1+\cdots+b_kX_k}+1}$$

Where,

$b_1$, $b_2$, $b_3$, ....are the coefficients of the independent variables X.

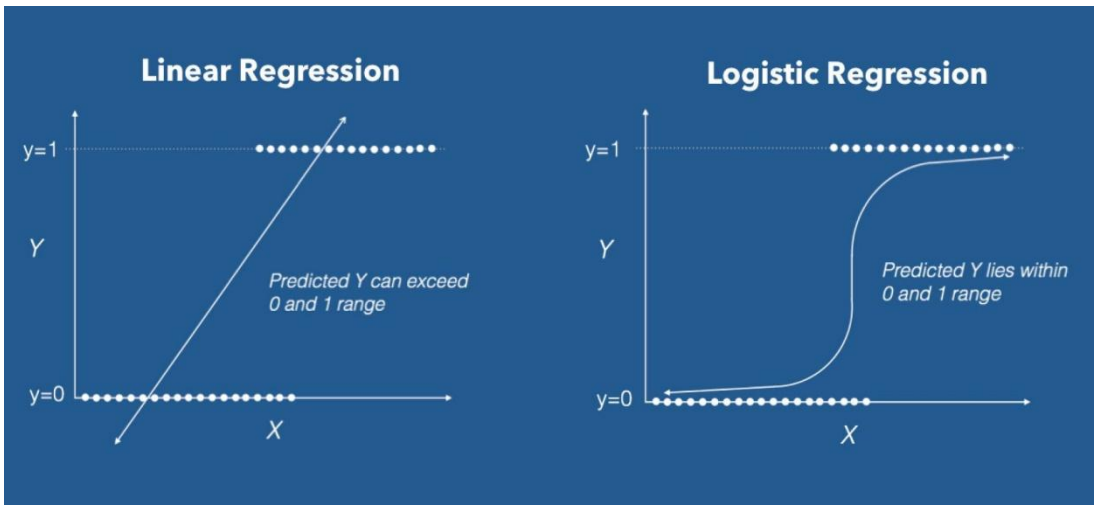



**Fig 4.3.2 Logistic Regression**

Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability. We can call a Logistic Regression a Linear Regression model but the Logistic Regression uses a more complex cost function, this cost function can be defined as the 'Sigmoid function' or also known as the 'logistic function' instead of a linear function.

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression).

Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1". In the logistic model, the log-odds (the logarithm of the odds) for the value labeled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value).

The corresponding probability of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labeling; the function that converts log- odds to probability is the logistic function, hence the name. The unit of measurement for the log-odds scale is called a logit, from logistic unit, hence the alternative names.

Analogous models with a different sigmoid function instead of the logistic function can also be used, such as the profit model; the defining characteristic of the logistic model is that increasing one of the independent variables multiplicatively scales the odds of the given outcome at a constant rate, with each independent variable having its own parameter; for a binary dependent variable this generalizes the odds ratio.

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. Sometimes logistic regressions are difficult to interpret; the Intellectus Statistics tool easily allows you to conduct the analysis, then in plain English interprets the output.

**Binary logistic regression major assumptions:**

- The dependent variable should be dichotomous in nature (e.g., presence vs. absent).
- There should be no outliers in the data, which can be assessed by converting the continuous predictors to standardized scores, and removing values below -3.29 or greater than 3.29.
- There should be no high correlations (multicollinearity) among the predictors. This can be assessed by a correlation matrix among the predictors. Tabachnick and Fidell (2013) suggest that as long correlation coefficients among independent variables are less than 0.90 the assumption is met.
- At the center of the logistic regression analysis is the task estimating the log odds of an event. Mathematically, logistic regression estimates a multiple linear regression function defined as:

  logit(p)

  for i = 1…n .

  **Overfitting:** When selecting the model for the logistic regression analysis, another important consideration is the model fit. Adding independent variables to a logistic regression model will always increase the amount of variance explained in the log odds (typically expressed as $R^2$). However, adding

more and more variables to the model can result in overfitting, which reduces the generalizability of the model beyond the data on which the model is fit.

**Reporting the $R^2$:** Numerous pseudo-$R^2$ values have been developed for binary logistic regression. These should be interpreted with extreme caution as they have many computational issues which cause them to be artificially high or low. A better approach is to present any of the goodness of fit tests available; Hosmer-Lemeshow is a commonly used measure of goodness of fit based on the Chi-square test.Logistic regression uses an equation as the representation, very much like linear regression.

Input values (x) are combined linearly using weights or coefficient values (referred to as the Greek capital letter Beta) to predict an output value (y). A key difference from linear regression is that the output value being modeled is a binary values (0 or 1) rather than a numeric value.

Below is an example logistic regression equation:

$$y = e^{(b0 + b1*x)} / (1 + e^{(b0 + b1*x)})$$

Where y is the predicted output, b0 is the bias or intercept term and b1 is the coefficient for the single input value (x). Each column in your input data has an associated b coefficient (a constant real value) that must be learned from your training data.

Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1. Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets. Logistic Regression can be used to classify the observations

using different types of data and can easily determine the most effective variables used for the classification.

### 4.3.3 Decision Tree Algorithm

Decision tree is a supervised classification algorithm which uses graphical representation to classify the data based on the solutions obtained when a decision is made using certain conditions. It has a tree like structure which consists of root nodes, internal/decision nodes and leaf nodes where a root node is termed as a parent node and the leaf node is termed as a child node and is a terminal node and the branches has the conditions. Here, a root node is taken and is split based on some condition thus deriving a sub tree and the process is repeated until the data is classified and also, we perform pruning when there are unwanted branches in the tree.

The root node i.e., the attribute that best classifies the decision tree is chosen by calculating the Information Gain (IG). The attribute with highest IG is considered as the root node.

**Information gain** = Entropy(s) – [(Weighted avg) * Entropy (each feature)]

Where,

**Entropy(s)** = $- P(\text{Yes})\log_2 P(\text{Yes}) - P(\text{No})\log_2 P(\text{No})$

Where, S = Total number of Samples

P(Yes) = probability of yes

P(No) = probability of no

Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute. Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data.

Decision tree learning methods use branching method to illustrate every possible outcome of a decision. They can work with discrete-value attributes and continuous value attributes as well. The learned trees are then represented in the form of if-then rules.

Three basic elements of the tree are decision node, branch and leaf node. Decision node specifies a test over some attribute. Each branch represents one of the possible values for this attribute. At last, leaf node represents the class to which the object belongs. There exist various decision tree algorithms. An ID3 algorithm uses greedy search approach. The tests are selected using information gain criteria. In ID3 algorithm, data may be overfitted and overclassified. ID3 does not handle missing values and numeric attributes. C4.5 is an improved version of ID3, given by Quinlan.

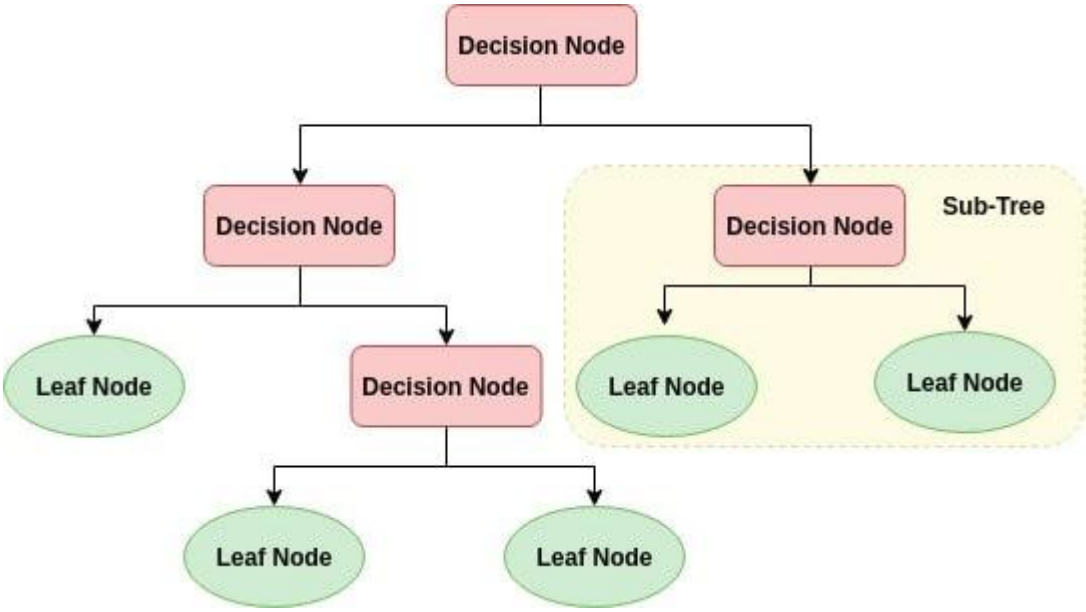



**Fig 4.3.3 Representation of Decision Tree Algorithm**

It accepts both discrete and continuous values and splits the tree based on the gain ratio. It also solves the over-fitting problem by using error based pruning technique. J48 is an open source implementation of C 4.5 in Weka. It reduces the chances of overfitting.

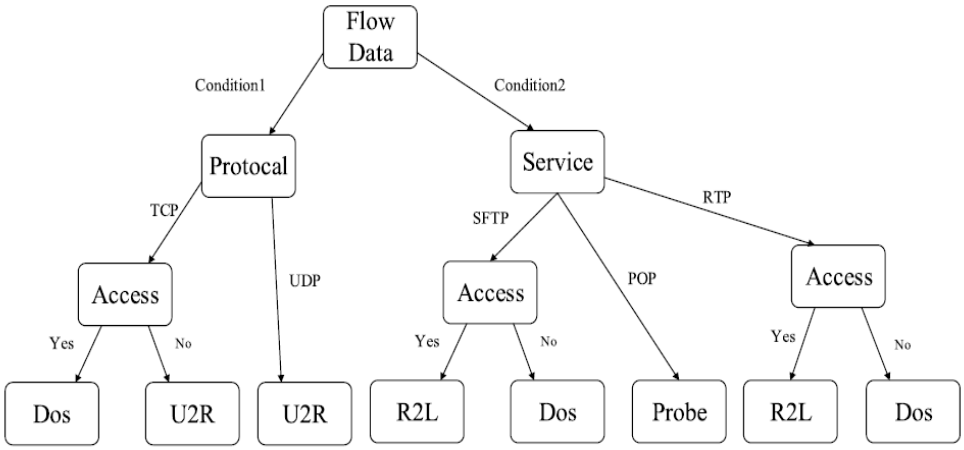



**Fig 4.3.4 Example of Decision Tree**

However, for noisy data, overfitting may happen. CART algorithm splits the tree based on towing criteria. It also handles both categorical and numerical values. It uses cost-complexity based pruning and handles missing values. Logistic Model Tree (LMT) uses a decision tree having linear regression model. Most of these algorithms operate from root to leaf to arrive at some decision.

The following measures are used for choosing the best attribute during classification: Entropy and Information gain. Entropy characterizes the impurity of an arbitrary collection of examples whereas Information gain measures how well a given attribute separates the training examples according to their target classification.

A decision tree is a tree structure in which each internal node represents a test on one property and each branch represents a test output, with each leaf node representing a category. In machine learning, the decision tree is a predictive model, it represents a mapping between object attributes and object values. Each node in the tree represents an object, each divergence path represents a possible attribute value, and each leaf node corresponds to the value of the object represented by the path from the root node to the leaf node. The decision tree only has a single output, if complex output is required, an independent decision tree to handle different outputs can be established.

### 4.3.4 Naive Bayes Classifier Algorithm

Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which help in building the fast machine learning models that can make quick predictions.

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)}$$

Where,

**P(A|B) is Posterior probability**: Probability of hypothesis A on the observed event B.

**P(B|A) is Likelihood probability**: Probability of the evidence given that the probability of a hypothesis is true.

**P(A) is Prior Probability**: Probability of hypothesis before observing the evidence.

**P(B) is Marginal Probability**: Probability of Evidence.

- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.
- Naive Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
- It can be used for Binary as well as Multi-class Classifications.
- It performs well in Multi-class predictions as compared to the other Algorithms.
- It is the most popular choice for text classification problems.

Naive Bayes algorithm is made up of two words Naive and Bayes**.** It is called naive because it assumes that the presence of particular feature is independent of the presence of other features. It is called bayes because it depends on principal of Baye's Theorem.

Naive Bayes is a supervised learning algorithm which helps in building the fast machine learning models that can make quick predictions. It is known to outperform even with highly sophisticated classification method.

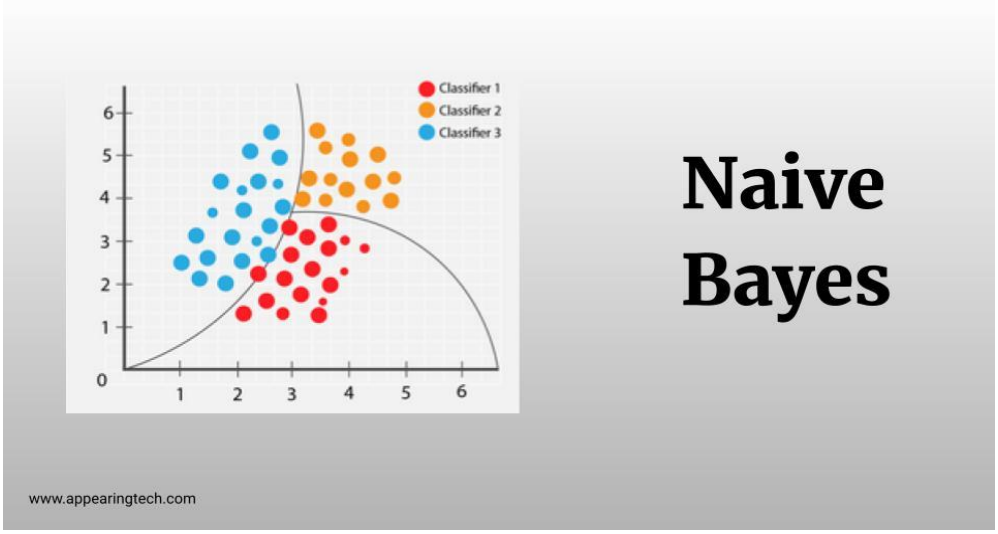



**Fig 4.3.5 Naive Bayes Classifier Algorithm**

A classifier is a machine learning model segregating different objects on the basis of certain features of variables. The Bayes Rule provides the formula for the probability of Y given X. But, in real-world problems, you typically have multiple X variables. When the features are independence, we can extend the Bayes Rule to what is called Naive Bayes. It is called 'Naive' because of the naive assumption that the X's are independent of each other. Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

### 4.3.5 Passive Aggressive Algorithm

Passive-Aggressive algorithms are generally used for large-scale learning. This algorithm has been a passive for a correct classification outcome, and turns aggressive in the event of a miscalculation, updating and adjusting. There are two parts are Passive and Aggressive. The Passive is nothing but if the prediction is correct, keep the model and do not make any changes. i.e., the data in the example is not enough to cause any changes in the model. And the other part is Aggressive is a if the prediction is incorrect, make changes to the model. i.e., some change to the model may correct it.Thus remains passive for correct predictions and responds aggressively to incorrect predictions.

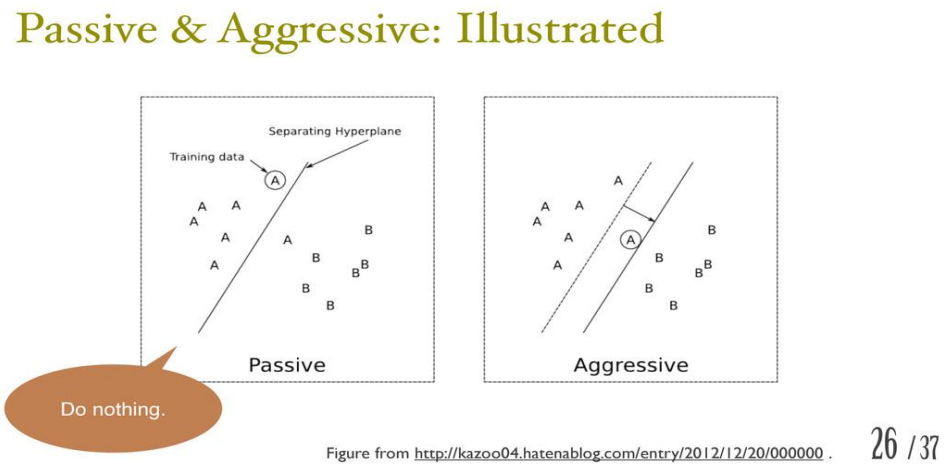



**Fig 4.3.6 Passive Aggressive Algorithm**

Passive-Aggressive algorithms are generally used for large-scale learning. It is one of the few ‘online-learning algorithms‘. In online machine learning algorithms, the input data comes in sequential order and the machine learning model is updated step-by-step, as opposed to batch learning, where the entire training dataset is used at once. This is very useful in situations where there is a huge amount of data and it is computationally infeasible to train the entire dataset because of the sheer size of the data.

Passive-Aggressive algorithms are somewhat similar to a Perceptron model, in the sense that they do not require a learning rate. However, they do include a regularization parameter. Passive-aggressive classification is one of the available incremental learning algorithms and it is very simple to implement, since it has a closed-form update rule.

## 4.4 PERFORMANCE MEASURE USING ACCURACY

The performances measure of different machine learning algorithms are calculated as accuracy. The accuracy can be defined as the ratio of the number of correctly classified cases to the total of cases under evaluation.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

Where, TP = True positives, TN = True Negatives

FP = False positives, FN = False Negatives

## 4.5 COMPARING OF ALL ALGORITHMS

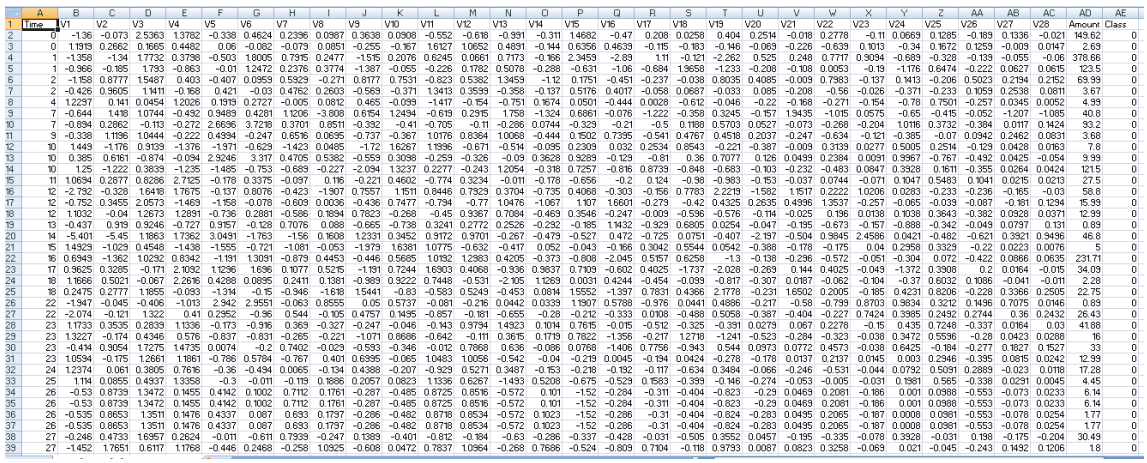
These Compares of all machine learning algorithms like Random Forest, Logistic Regression, Naive Bayes Classifier, Passive Aggressive and Decision Tree Algorithm. And choose the one as the best algorithm depends on best accurate results.

# 5. IMPLEMENTATION

The implementation of the project “Credit card fraud detection using machine learning”, starts with data collection. We collect the data as ‘Creditcard’ dataset which are stored as csv file. The dataset are collected from the website Kaggle which maintains the transactions made by credit cards in September 2013 by European cardholders with 2,84,807 transactions.

## 5.1 DATA COLLECTION

The data can be collected as ‘Creditcard’ dataset which contains all transactions. These CreditCard dataset of csv file are shown in figure 5.1



| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | -1.36 | -0.073 | 2.5363 | 1.3782 | -0.338 | 0.4624 | 0.2396 | 0.0987 | 0.3638 | 0.0908 | -0.552 | -0.618 | -0.991 | -0.311 | 1.4682 | -0.47 | 0.208 | 0.0258 | 0.404 | 0.2514 | -0.018 | 0.2778 | -0.11 | 0.0669 | 0.1285 | -0.189 | 0.1336 | -0.021 | 149.62 | 0 |
| 0 | 1.1919 | 0.2662 | 0.1665 | 0.4482 | 0.06 | -0.082 | -0.079 | 0.0851 | -0.255 | -0.167 | 1.6127 | 1.0652 | 0.4891 | -0.144 | 0.6356 | 0.4639 | -0.115 | -0.183 | -0.146 | -0.069 | -0.226 | -0.639 | 0.1013 | -0.34 | 0.1672 | 0.1259 | -0.009 | 0.0147 | 2.69 | 0 |
| 1 | -1.358 | -1.34 | 1.7732 | 0.3798 | -0.503 | 1.8005 | 0.7915 | 0.2477 | -1.515 | 0.2076 | 0.6245 | 0.0661 | 0.7173 | -0.166 | 2.3459 | -2.89 | 1.11 | -0.121 | -2.262 | 0.525 | 0.248 | 0.7717 | 0.9094 | -0.689 | -0.328 | -0.139 | -0.055 | -0.06 | 378.66 | 0 |
| 1 | -0.966 | -0.185 | 1.793 | -0.863 | -0.01 | 1.2472 | 0.2376 | 0.3774 | -1.387 | -0.055 | -0.226 | 0.1782 | 0.5078 | -0.288 | -0.631 | -1.06 | -0.684 | 1.9658 | -1.233 | -0.208 | -0.108 | 0.0053 | -0.19 | -1.176 | 0.6474 | -0.222 | 0.0627 | 0.0615 | 123.5 | 0 |
| 2 | -1.158 | 0.8777 | 1.5487 | 0.403 | -0.407 | 0.0959 | 0.5929 | -0.271 | 0.8177 | 0.7531 | -0.823 | 0.5382 | 1.3459 | -1.12 | 0.1751 | -0.451 | -0.237 | -0.038 | 0.8035 | 0.4085 | -0.009 | 0.7983 | -0.137 | 0.1413 | -0.206 | 0.5023 | 0.2194 | 0.2152 | 69.99 | 0 |
| 2 | -0.426 | 0.9605 | 1.1411 | -0.168 | 0.421 | -0.03 | 0.4762 | 0.2603 | -0.569 | -0.371 | 1.3413 | 0.3599 | -0.358 | -0.137 | 0.5176 | 0.4017 | -0.058 | 0.0687 | -0.033 | 0.085 | -0.208 | -0.56 | -0.026 | -0.371 | -0.233 | 0.1059 | 0.2538 | 0.0811 | 3.67 | 0 |
| 4 | 1.2297 | 0.141 | 0.0454 | 1.2026 | 0.1919 | 0.2727 | -0.005 | 0.0812 | 0.465 | -0.099 | -1.417 | -0.154 | -0.751 | 0.1674 | 0.0501 | -0.444 | 0.0028 | -0.612 | -0.046 | -0.22 | -0.168 | -0.271 | -0.154 | -0.78 | 0.7501 | -0.257 | 0.0345 | 0.0052 | 4.99 | 0 |
| 7 | -0.644 | 1.418 | 1.0744 | -0.492 | 0.9483 | 0.4281 | 1.1206 | -3.808 | 0.6154 | 1.2494 | -0.619 | 0.2915 | 1.758 | -1.324 | 0.6861 | -0.076 | -1.222 | -0.358 | 0.3245 | -0.157 | 1.9435 | -1.015 | 0.0575 | -0.65 | -0.415 | -0.052 | -1.207 | -1.085 | 40.8 | 0 |
| 7 | -0.894 | 0.2862 | -0.113 | -0.272 | 2.6696 | 3.7218 | 0.3701 | 0.8511 | -0.392 | -0.41 | -0.705 | -0.11 | -0.286 | 0.0744 | -0.329 | -0.21 | -0.5 | 0.1188 | 0.5703 | 0.0527 | -0.073 | -0.268 | -0.204 | 1.0116 | 0.3732 | -0.384 | 0.0117 | 0.1424 | 93.2 | 0 |
| 9 | -0.338 | 1.1196 | 1.0444 | -0.222 | 0.4994 | -0.247 | 0.6536 | 0.0595 | -0.737 | -0.367 | 1.0776 | 0.8284 | 1.0068 | -0.444 | 0.1502 | 0.7395 | -0.541 | 0.4767 | 0.4518 | 0.2037 | -0.247 | -0.634 | -0.121 | -0.385 | -0.07 | 0.0942 | 0.2462 | 0.0831 | 3.68 | 0 |
| 10 | 1.449 | -1.176 | 0.9139 | -1.376 | -1.971 | -0.629 | -1.423 | 0.0485 | -1.72 | 1.6267 | 1.1996 | -0.671 | -0.514 | -0.095 | 0.2309 | 0.032 | 0.2534 | 0.8543 | -0.221 | -0.387 | -0.009 | 0.3139 | 0.0277 | 0.5005 | 0.2514 | -0.129 | 0.0428 | 0.0163 | 7.8 | 0 |
| 10 | 0.385 | 0.6161 | -0.874 | -0.094 | 2.9246 | 3.317 | 0.4705 | 0.5382 | -0.559 | 0.3098 | -0.259 | -0.326 | -0.09 | 0.3628 | 0.9289 | -0.129 | -0.81 | 0.36 | 0.7077 | 0.126 | 0.0493 | 0.2384 | 0.0091 | 0.9967 | -0.767 | -0.492 | 0.0425 | -0.054 | 9.99 | 0 |
| 10 | 1.25 | -1.222 | 0.3839 | -1.235 | -1.485 | -0.753 | -0.689 | -0.227 | -2.094 | 1.3237 | 0.2277 | -0.243 | 1.2054 | -0.316 | 0.7257 | -0.816 | 0.8739 | -0.848 | -0.683 | -0.103 | -0.232 | -0.483 | 0.0847 | 0.3928 | 0.1611 | -0.355 | 0.0264 | 0.0424 | 121.5 | 0 |
| 11 | 1.0694 | 0.2877 | 0.8286 | 2.7125 | -0.178 | 0.3375 | -0.097 | 0.116 | -0.221 | 0.4602 | -0.774 | 0.3234 | -0.011 | -0.178 | -0.656 | -0.2 | 0.124 | -0.98 | -0.983 | -0.153 | -0.037 | 0.0744 | -0.071 | 0.1047 | 0.5483 | 0.1041 | 0.0215 | 0.0213 | 27.5 | 0 |
| 12 | -2.792 | -0.328 | 1.6418 | 1.7675 | -0.137 | 0.8076 | -0.423 | -1.907 | 0.7557 | 1.1511 | 0.8466 | 0.7329 | 0.3704 | -0.735 | 0.4068 | -0.303 | -0.156 | 0.7783 | 2.2219 | -1.582 | 1.1517 | 0.2222 | 1.0206 | 0.0293 | -0.233 | -0.236 | -0.185 | -0.03 | 58.8 | 0 |
| 12 | -0.752 | 0.3455 | 2.0573 | -1.469 | -1.158 | -0.078 | -0.609 | 0.0036 | -0.436 | 0.7477 | -0.794 | -0.77 | 1.0476 | -1.067 | 1.107 | 1.6601 | -0.279 | -0.42 | 0.4325 | 0.2635 | 0.4996 | 1.3537 | -0.257 | -0.065 | -0.039 | -0.087 | -0.181 | 0.1294 | 15.99 | 0 |
| 12 | 1.1032 | -0.04 | 1.2673 | 1.2891 | -0.736 | 0.2881 | -0.586 | 0.1894 | 0.7823 | -0.268 | -0.45 | 0.9367 | 0.7084 | -0.469 | 0.3546 | -0.247 | -0.009 | -0.596 | -0.576 | -0.114 | -0.025 | 0.196 | 0.0118 | 0.1038 | 0.3643 | -0.382 | 0.0928 | 0.0371 | 12.99 | 0 |
| 13 | -0.437 | 0.919 | 0.9246 | -0.727 | 0.9157 | -0.128 | 0.7076 | 0.088 | -0.665 | -0.738 | 0.3241 | 0.2772 | 0.2526 | -0.232 | -0.185 | 1.1432 | -0.929 | 0.6805 | 0.0254 | -0.047 | -0.195 | -0.673 | -0.157 | -0.888 | -0.342 | -0.049 | 0.0797 | 0.131 | 0.89 | 0 |
| 14 | -5.401 | -5.45 | 1.1663 | 1.7362 | 3.0491 | -1.763 | -1.56 | 0.1608 | 1.2331 | 0.3452 | 0.9172 | 0.9701 | -0.267 | -0.479 | -0.527 | 0.472 | -0.725 | 0.0751 | -0.407 | -2.197 | -0.504 | 0.9845 | 2.4586 | 0.0421 | -0.482 | -0.621 | 0.3521 | 0.9496 | 46.8 | 0 |
| 15 | 1.4923 | -1.029 | 0.4548 | -1.438 | -1.555 | -0.721 | -1.081 | -0.053 | -1.979 | 1.6381 | 1.0775 | -0.632 | -0.417 | 0.052 | -0.043 | -0.166 | 0.3942 | 0.5544 | 0.0542 | -0.388 | -0.178 | -0.175 | 0.04 | 0.2358 | 0.3329 | -0.22 | 0.0223 | 0.0076 | 5 | 0 |
| 16 | 0.6949 | -1.362 | 1.0232 | 0.8342 | -1.191 | 1.3091 | -0.879 | 0.4453 | -0.446 | 0.5685 | 1.0192 | 1.2983 | 0.4205 | -0.373 | -0.808 | -2.045 | 0.5157 | 0.6258 | -1.3 | -0.138 | -0.296 | -0.572 | -0.051 | -0.304 | 0.072 | -0.422 | 0.0866 | 0.0635 | 231.71 | 0 |
| 17 | 0.9625 | 0.3285 | -0.171 | 2.1092 | 1.1296 | 1.696 | 0.1077 | 0.5215 | -1.191 | 0.7244 | 1.6903 | 0.4068 | -0.336 | 0.9837 | 0.7109 | -0.602 | 0.4025 | -1.737 | -2.028 | -0.269 | 0.144 | 0.4025 | -0.049 | -1.372 | 0.3908 | 0.2 | 0.0164 | -0.015 | 34.09 | 0 |
| 18 | 1.1666 | 0.5021 | -0.067 | 2.2616 | 0.4288 | 0.0895 | 0.2411 | 0.1381 | -0.989 | 0.9222 | 0.7448 | -0.531 | -2.105 | 1.1269 | 0.0031 | 0.4244 | -0.454 | -0.099 | -0.817 | -0.307 | 0.0187 | -0.062 | -0.104 | -0.37 | 0.6032 | 0.1086 | -0.041 | -0.011 | 2.28 | 0 |
| 18 | 0.2475 | 0.2777 | 1.1855 | -0.093 | -1.314 | -0.15 | -0.946 | -1.618 | 1.5441 | -0.83 | -0.583 | 0.5249 | -0.453 | 0.0814 | 1.5552 | -1.397 | 0.7831 | 0.4366 | 2.1778 | -0.231 | 1.6502 | 0.2005 | -0.185 | 0.4231 | 0.8206 | -0.228 | 0.3366 | 0.2505 | 22.75 | 0 |
| 22 | -1.947 | -0.045 | -0.406 | -1.013 | 2.342 | 2.9551 | -0.063 | 0.8555 | 0.05 | 0.5737 | -0.081 | -0.216 | 0.0442 | 0.0339 | 1.1907 | 0.5788 | -0.376 | 0.0441 | 0.4886 | -0.217 | -0.58 | -0.799 | 0.8703 | 0.3634 | 0.3212 | 0.1496 | 0.7075 | 0.0146 | 0.89 | 0 |
| 22 | -2.074 | -0.121 | 1.322 | 0.41 | 0.2952 | -0.96 | 0.544 | -0.105 | 0.4757 | 0.1495 | -0.857 | -0.181 | -0.655 | -0.28 | -0.212 | -0.333 | 0.0108 | -0.488 | 0.5058 | -0.387 | -0.404 | -0.227 | 0.7424 | 0.3985 | 0.2492 | 0.2744 | 0.36 | 0.2432 | 26.43 | 0 |
| 23 | 1.1733 | 0.3535 | 0.2839 | 1.1336 | -0.173 | -0.916 | 0.369 | -0.327 | -0.247 | -0.046 | -0.143 | 0.9794 | 1.4323 | 0.1014 | 0.7615 | -0.015 | -0.512 | -0.325 | -0.391 | 0.0279 | 0.067 | 0.2278 | -0.15 | 0.435 | 0.7248 | -0.337 | 0.0164 | 0.03 | 41.88 | 0 |
| 23 | 1.3227 | -0.174 | 0.4346 | 0.576 | -0.837 | -0.831 | -0.265 | -0.221 | -1.071 | 0.8686 | -0.642 | -0.111 | 0.3615 | 0.1719 | 0.7822 | -1.356 | -0.217 | 1.2718 | -1.241 | -0.523 | -0.284 | -0.323 | -0.038 | 0.3472 | 0.5536 | -0.28 | 0.0423 | 0.0288 | 16 | 0 |
| 23 | -0.414 | 0.9054 | 1.7275 | 1.4735 | 0.0074 | -0.2 | 0.7402 | -0.029 | -0.593 | -0.346 | -0.012 | 0.7868 | 0.636 | -0.086 | 0.0768 | -1.406 | 0.7756 | -0.943 | 0.544 | 0.0373 | 0.0772 | 0.4573 | -0.038 | 0.6425 | -0.184 | -0.277 | 0.1827 | 0.1527 | 33 | 0 |
| 23 | 1.0594 | -0.175 | 1.2661 | 1.1861 | -0.786 | 0.5794 | -0.767 | 0.401 | 0.8395 | -0.065 | 1.0483 | 1.0056 | -0.542 | -0.04 | -0.219 | 0.0045 | -0.194 | 0.0424 | -0.278 | -0.178 | 0.0137 | 0.2137 | 0.0145 | 0.003 | 0.2346 | -0.395 | 0.0815 | 0.0242 | 12.99 | 0 |
| 24 | 1.2374 | 0.061 | 0.3805 | 0.7616 | -0.36 | -0.494 | 0.0065 | -0.134 | 0.4388 | -0.207 | -0.929 | 0.5271 | 0.3487 | -0.153 | -0.218 | -0.192 | -0.117 | -0.634 | 0.3484 | -0.066 | -0.246 | -0.531 | -0.044 | 0.0792 | 0.5091 | 0.2889 | -0.023 | 0.0118 | 17.28 | 0 |
| 25 | 1.114 | 0.0855 | 0.4937 | 1.3358 | -0.3 | -0.011 | -0.119 | 0.1886 | 0.2057 | 0.0823 | 1.1336 | 0.6267 | -1.493 | 0.5208 | -0.675 | -0.529 | 0.1583 | -0.399 | -0.146 | -0.274 | -0.053 | -0.005 | -0.031 | 0.1981 | 0.565 | -0.338 | 0.0291 | 0.0045 | 4.45 | 0 |
| 26 | -0.53 | 0.8739 | 1.3472 | 0.1455 | 0.4142 | 0.1002 | 0.7112 | 0.1761 | -0.287 | -0.485 | 0.8725 | 0.8516 | -0.572 | 0.101 | -1.52 | -0.284 | -0.311 | -0.404 | -0.823 | -0.29 | 0.0469 | 0.2081 | -0.186 | 0.001 | 0.0988 | -0.553 | -0.073 | 0.0233 | 6.14 | 0 |
| 26 | -0.53 | 0.8739 | 1.3472 | 0.1455 | 0.4142 | 0.1002 | 0.7112 | 0.1761 | -0.287 | -0.485 | 0.8725 | 0.8516 | -0.572 | 0.101 | -1.52 | -0.284 | -0.311 | -0.404 | -0.823 | -0.29 | 0.0469 | 0.2081 | -0.186 | 0.001 | 0.0988 | -0.553 | -0.073 | 0.0233 | 6.14 | 0 |
| 26 | -0.535 | 0.8653 | 1.3511 | 0.1476 | 0.4337 | 0.087 | 0.693 | 0.1797 | -0.286 | -0.482 | 0.8718 | 0.8534 | -0.572 | 0.1023 | -1.52 | -0.286 | -0.31 | -0.404 | -0.824 | -0.283 | 0.0495 | 0.2065 | -0.187 | 0.0008 | 0.0981 | -0.553 | -0.078 | 0.0254 | 1.77 | 0 |
| 26 | -0.535 | 0.8653 | 1.3511 | 0.1476 | 0.4337 | 0.087 | 0.693 | 0.1797 | -0.286 | -0.482 | 0.8718 | 0.8534 | -0.572 | 0.1023 | -1.52 | -0.286 | -0.31 | -0.404 | -0.824 | -0.283 | 0.0495 | 0.2065 | -0.187 | 0.0008 | 0.0981 | -0.553 | -0.078 | 0.0254 | 1.77 | 0 |
| 27 | -0.246 | 0.4733 | 1.6957 | 0.2624 | -0.011 | -0.611 | 0.7939 | -0.247 | 0.1389 | -0.401 | -0.812 | -0.184 | -0.63 | -0.286 | -0.337 | -0.428 | -0.031 | -0.505 | 0.3552 | 0.0457 | -0.195 | -0.335 | -0.078 | 0.3928 | -0.031 | 0.198 | -0.175 | -0.204 | 30.49 | 0 |
| 27 | -1.452 | 1.7651 | 0.6117 | 1.1768 | -0.446 | 0.2468 | -0.258 | 1.0925 | -0.608 | 0.0472 | 0.7837 | 1.0964 | -0.268 | 0.7686 | -0.524 | -0.809 | 0.7104 | -0.118 | 0.9793 | 0.0087 | 0.0823 | 0.3258 | -0.069 | 0.021 | -0.045 | -0.243 | 0.1492 | 0.1206 | 1.8 | 0 |

**Fig 5.1 Credit card Data Collection**

In this dataset, there are 31 columns out of which 28 are named as v1-v28 to protect sensitive data. The other columns represent time, amount and class. Time shows the time gap between the first transaction and the following one. Amount is the amount of money transacted. Class 0 represents a valid transaction and 1 represents a fraudulent one.

## 5.2 IMPORTING LIBRARIES

The Libraries are imported to build the classification model of different machine learning algorithms. They are used to create GUI applications to build the model and accuracy as results.

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
import matplotlib.pyplot as plt
import numpy as np
from tkinter.filedialog import askopenfilename
import numpy as np
import pandas as pd
from sklearn import *
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import PassiveAggressiveClassifier
```

## 5.3 DATA UPLOADED

The creditcard dataset is uploaded to build the classification model by inserting the file.

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

## 5.4 SPLITTING THE DATA INTO TRAINING AND TESTING

The data can be splitting into training and testing into two parts with total datasize. The dataset are splits into 6:4, 7:3 and 8:2 ratios of training and testing of the data respectively.

```
def generateModel():
    global X, Y, X_train, X_test, y_train, y_test
    train = pd.read_csv(filename)
    X, Y, X_train, X_test, y_train, y_test = traintest(train)
    text.insert(END,"Train & Test Model Generated\n\n")
    text.insert(END,"Total Dataset Size : "+str(len(train))+"\n")
    text.insert(END,"Split Training Size : "+str(len(X_train))+"\n")
    text.insert(END,"Split Test Size : "+str(len(X_test))+"\n")
```

## 5.5 BUILDING THE CLASSIFICATION MODEL

The Classification Models as Logistic Regression, Random Forest, Decision Tree, Naive Bayes Classifier and Passive Aggressive Algorithms are different machine learning algorithms are built the frauds as calculate as Accuracies.

### 5.5.1 Classification model for Random Forest

The data is classified into train and test with 6:4, 7:3 and 8:2 ratios size respectively and is used to building the model.

Random Forest Algorithm code for calculating accuracy as follows:

```
def runRandomForest():
    headers =
["Time","V1","V2","V3","V4","V5","V6","V7","V8","V9","V10","V11","V1
2","V13","V14","V15","V16","V17","V18","V19","V20","V21","V22","V23
","V24","V25","V26","V27","V28","Amount","Class"]
    global random_acc
    global cls
    global X, Y, X_train, X_test, y_train, y_test
    cls =
RandomForestClassifier(n_estimators=50,max_depth=2,random_state=0,clas
s_weight='balanced')
    cls.fit(X_train, y_train)
    text.insert(END,"Prediction Results\n\n")
    prediction_data = prediction(X_test, cls)
    random_acc = cal_accuracy(y_test, prediction_data,'Random Forest
Accuracy')
```

### 5.5.2 Classification Model for Logistic Regression

The data is classified into train and test with 6:4, 7:3 and 8:2 ratios size respectively and is used to building the model.

Logistic Regression Algorithm accuracy code for calculating is as follows:

```
def runLogisticRegression():
    headers =
["Time","V1","V2","V3","V4","V5","V6","V7","V8","V9","V10","V11","V12",
"V13","V14","V15","V16","V17","V18","V19","V20","V21","V22","V23","V2
4","V25","V26","V27","V28","Amount","Class"]
    global logistic_acc
    global cls
    global X, Y, X_train, X_test, y_train, y_test
    cls =
LogisticRegression(solver='liblinear',max_iter=100,random_state=0,class_weigh
t='balanced')
    cls.fit(X_train, y_train)
    text.insert(END,"Prediction Results\n\n")
    prediction_data = prediction(X_test, cls)
    logistic_acc = cal_accuracy(y_test, prediction_data,'Logistic Regression
Accuracy')
```

### 5.5.3 Classification Model for Decision Tree Algorithm

The data is classified into train and test with 6:4, 7:3 and 8:2 ratios size respectively and is used to building the model.

Decision Tree Algorithm accuracy code for calculating is as follows:

```
def runDecisionTree():
    headers =
["Time","V1","V2","V3","V4","V5","V6","V7","V8","V9","V10","V11","V
12","V13","V14","V15","V16","V17","V18","V19","V20","V21","V22","V
23","V24","V25","V26","V27","V28","Amount","Class"]
    global decision_acc
    global cls
    global X, Y, X_train, X_test, y_train, y_test
    cls =
DecisionTreeClassifier(max_depth=2,random_state=0,class_weight='balance
d')
    cls.fit(X_train, y_train)
    text.insert(END,"Prediction Results\n\n")
    prediction_data = prediction(X_test, cls)
    decision_acc = cal_accuracy(y_test, prediction_data,'Decision tree
Accuracy')
```

### 5.5.4 Classification Model for Naive Bayes Classifier Algorithm

The data is classified into train and test with 6:4, 7:3 and 8:2 ratios size respectively and is used to building the model.

Naive Bayes Classifier Algorithm accuracy code for calculating is as follows:

```
def runGaussianNB():
    headers =
["Time","V1","V2","V3","V4","V5","V6","V7","V8","V9","V10","V11","
V12","V13","V14","V15","V16","V17","V18","V19","V20","V21","V22"
,"V23","V24","V25","V26","V27","V28","Amount","Class"]
    global naive_acc
    global cls
    global X, Y, X_train, X_test, y_train, y_test
    cls = GaussianNB()
    cls.fit(X_train, y_train)
    text.insert(END,"Prediction Results\n\n")
    prediction_data = prediction(X_test, cls)
    naive_acc = cal_accuracy(y_test, prediction_data,'GaussianNB
Accuracy')
```

### 5.5.5 Classification Model for Passive Aggressive Algorithm

The data is classified into train and test with 6:4, 7:3 and 8:2 ratios size respectively and is used to building the model.

Passive Aggressive Algorithm accuracy code for calculating is as follows:

```
def runPassiveAggressive():
    headers =
["Time","V1","V2","V3","V4","V5","V6","V7","V8","V9","V10","V11","
V12","V13","V14","V15","V16","V17","V18","V19","V20","V21","V22"
,"V23","V24","V25","V26","V27","V28","Amount","Class"]
    global passive_acc
    global cls
    global X, Y, X_train, X_test, y_train, y_test
    cls = PassiveAggressiveClassifier(C = 0.5, random_state = 5)
    cls.fit(X_train, y_train)
    text.insert(END,"Prediction Results\n\n")
    prediction_data = prediction(X_test, cls)
    passive_acc = cal_accuracy(y_test, prediction_data,'PassiveAggreessive
Accuracy')
```

### 5.6 Performance Measure of Accuracy

The performances measures of different machine algorithms like Random Forest, Logistic Regression, Decision Tree, Naive Bayes Classifier and Passive Aggressive Algorithm are calculate the Accuracy by classification Models.

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

### 5.7 Comparison of algorithms

The Comparison of different machine learning algorithms depends on accuracies results of the classification models to detect the credit card transaction. These Comparison graphs of different algorithms in all splits of 6:4, 7:3, and 8:2 ratios of bar graph as shown.

```
def compGraph():
    data = {'Random Forest':random_acc , 'Logistic
Regression':logistic_acc, 'Decision Tree':decision_acc,
'Naive_Bayes':naive_acc, 'Passive Aggressive':passive_acc}
    algorithms = list(data.keys())
    accuracy = list(data.values())
    acc_fig = plt.figure(figsize = (10, 5))
    plt.bar(algorithms, accuracy, color = "blue", width = 0.2)
    plt.xlabel("\nMachine Learning Algorithms")
    plt.ylabel("Accuracies")
    plt.title("Comparison graph of different algorithms")
    plt.show()
```

# 6. RESULTS AND DISCUSSIONS

## 6.1 MAIN SCREEN

The Main Screen shows the Credit Card Fraud Detection using Machine learning Algorithms in GUI application. They have buttons to upload credit card dataset, Splits 6:4, Splits 7:3, Splits 8:2 , Run Random Forest, Logistic Regression, Decision Tree, Naive Bayes Classifier and Passive Aggressive Algorithms and Accuracy Comparison Graph and last Exit.


**Fig 6.1 Main Screen**

## 6.2 DATASET UPLOADED

Dataset is uploaded by inserting the 'Creditcard' dataset of csv file. And it displayed as loaded with filename.


**Fig 6.2 Dataset uploaded**

## 6.3 SPLITTING THE DATASET

The dataset can be splitted of total dataset size of transactions is 2,84,807. It can be splitted into two parts. It can split the dataset into 6:4, 7:3 and 8:2 ratios of training and testing of data.


**Fig 6.3.1 Split the dataset for 6:4**


**Fig 6.3.2 Split the dataset for 7:3**

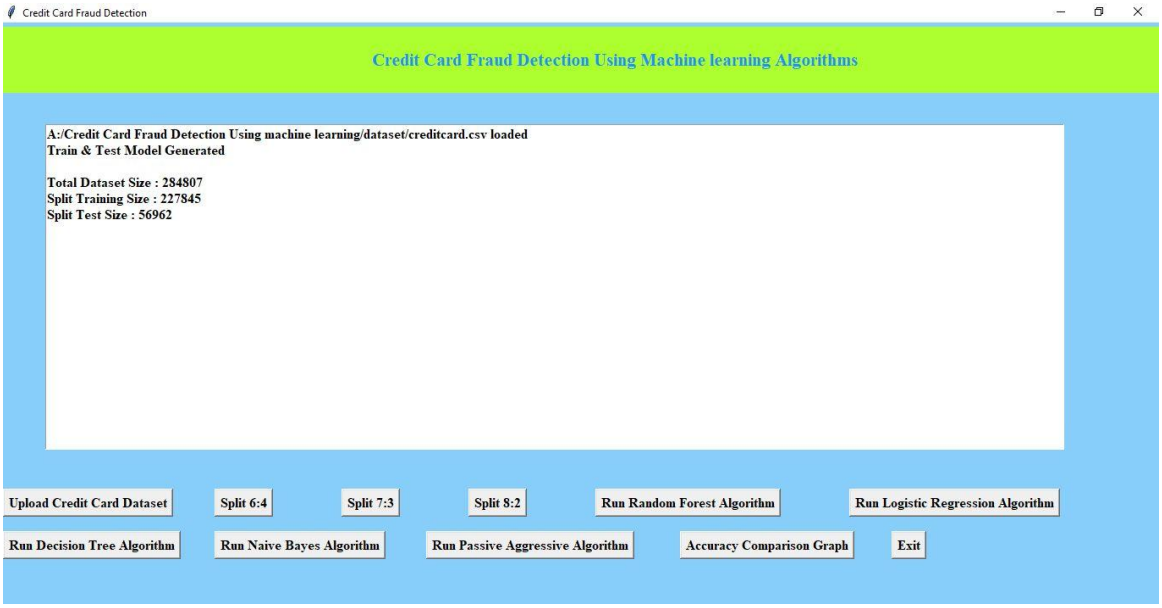



**Fig 6.3.3 Split the dataset for 8:2**

## 6.4 RANDOM FOREST ACCURACY

By clicking the Run Random Forest Algorithm button to generate the Random Forest model on train and test the data. The Prediction Results of this Random Forest generated 99.82%, 99.78% and 99.79% accuracy as shown below in fig 6.4.1, 6.4.2 and 6.4.3

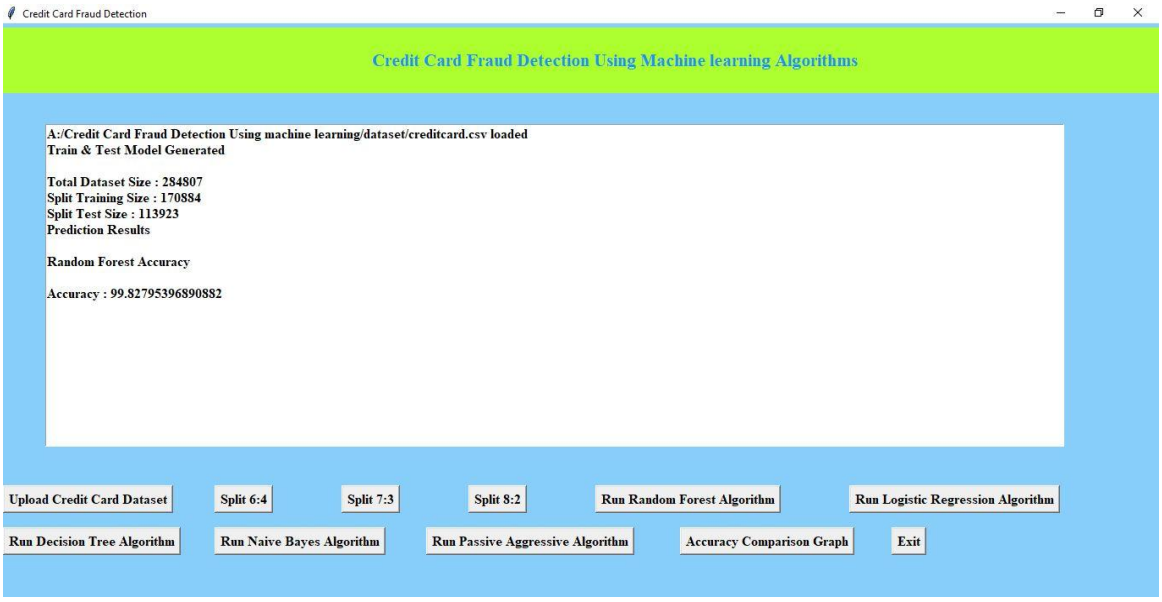



**Fig 6.4.1 Random Forest Accuracy for 6:4**

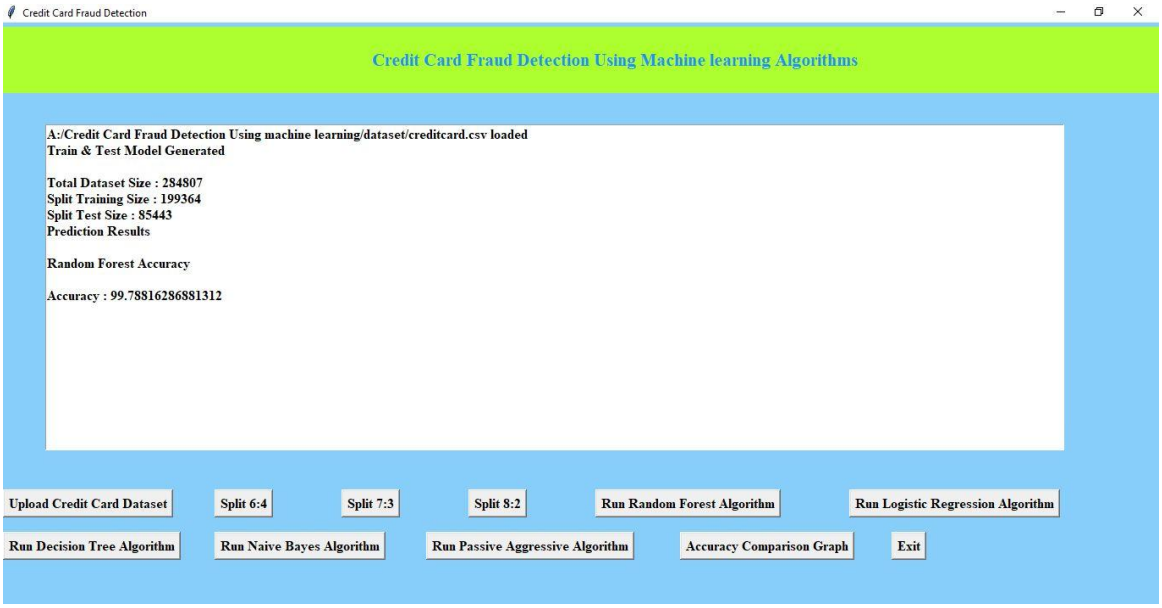



**Fig 6.4.2 Random Forest Accuracy for 7:3**


**Fig 6.4.3 Random Forest Accuracy for 8:2**

## 6.5 LOGISTIC REGRESSION ACCURACY

By clicking the Run Logistic Regression Algorithm button to generate the Logistic regression model on train and test the data. The Prediction Results of this Logistic Regression generated 97.61%, 97.78% and 97.71% accuracy as shown below in fig 6.5.1, 6.5.2 and 6.5.3


**Fig 6.5.1 Logistic Regression Accuracy for 6:4**

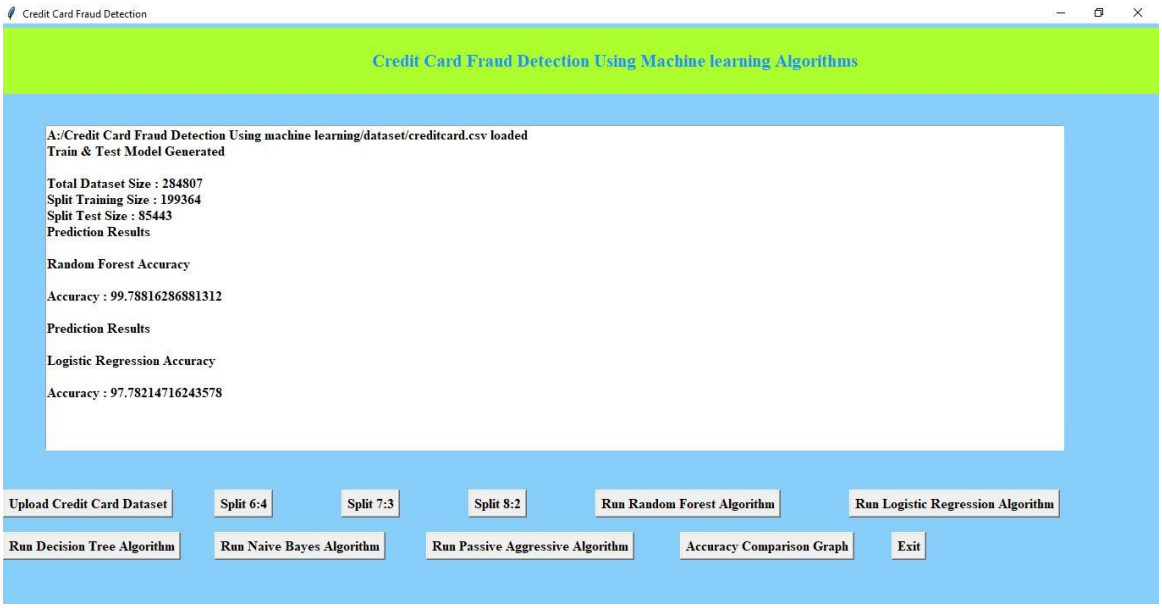



**Fig 6.5.2 Logistic Regression Accuracy for 7:3**

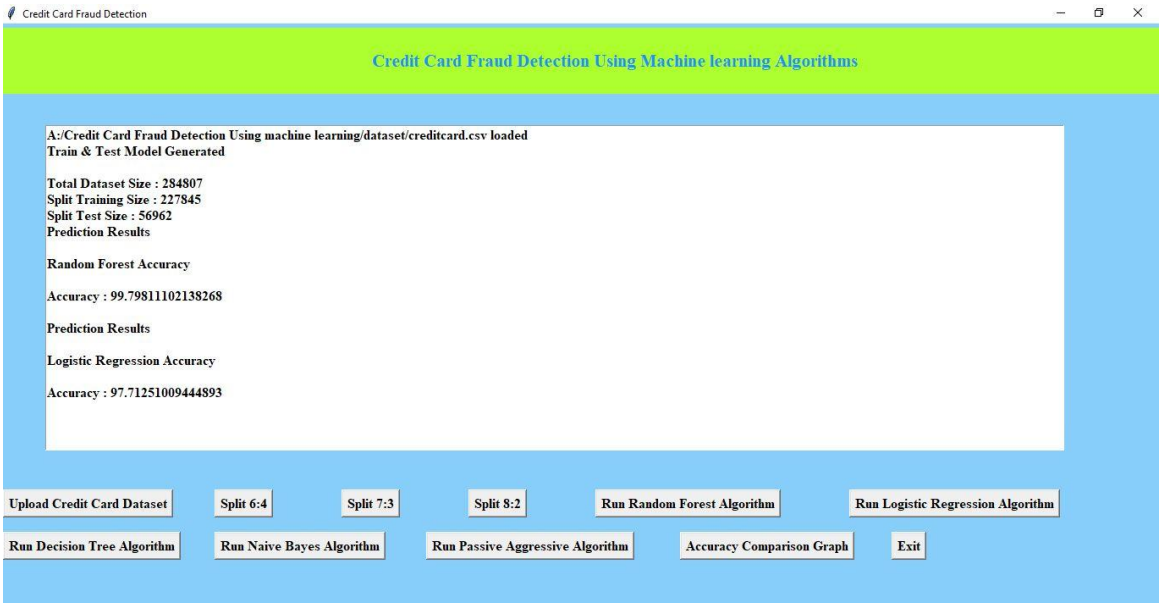



**Fig 6.5.3 Logistic Regression Accuracy for 8:2**

## 6.6 DECISION TREE ACCURACY

By clicking the Run Decision Tree Algorithm button to generate the decision tree model on train and test the data. The Prediction Results of this Decision Tree Algorithm generated 93.25%, 97.14% and 97.07% accuracy as shown below in fig 6.6.1, 6.6.2 and 6.6.3


**Fig 6.6.1 Decision Tree Accuracy for 6:4**

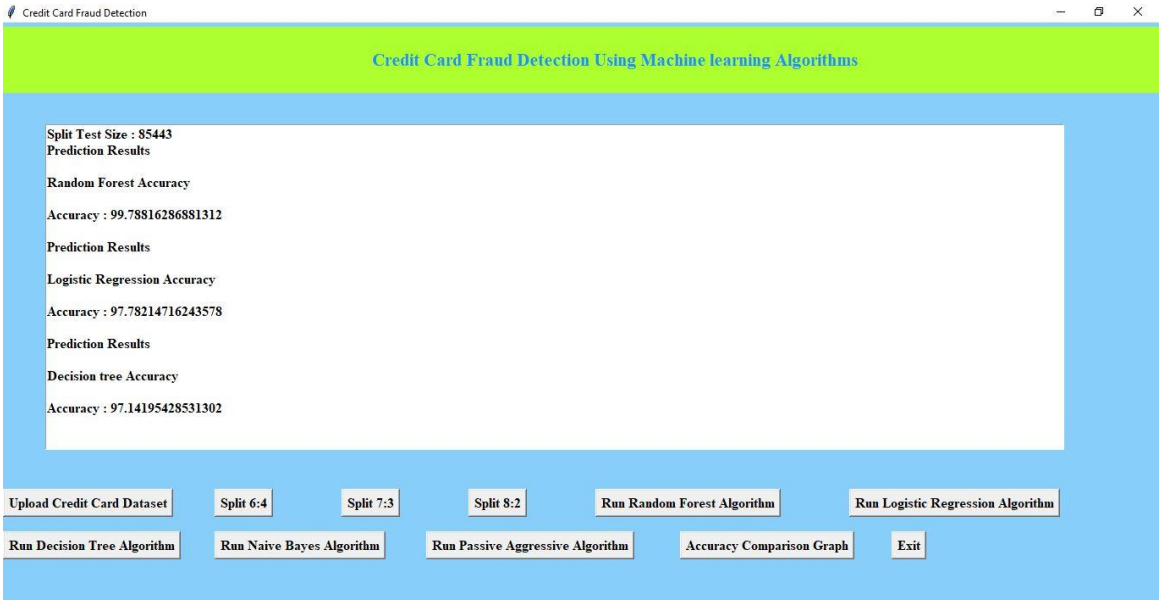



**Fig 6.6.2 Decision Tree Accuracy for 7:3**

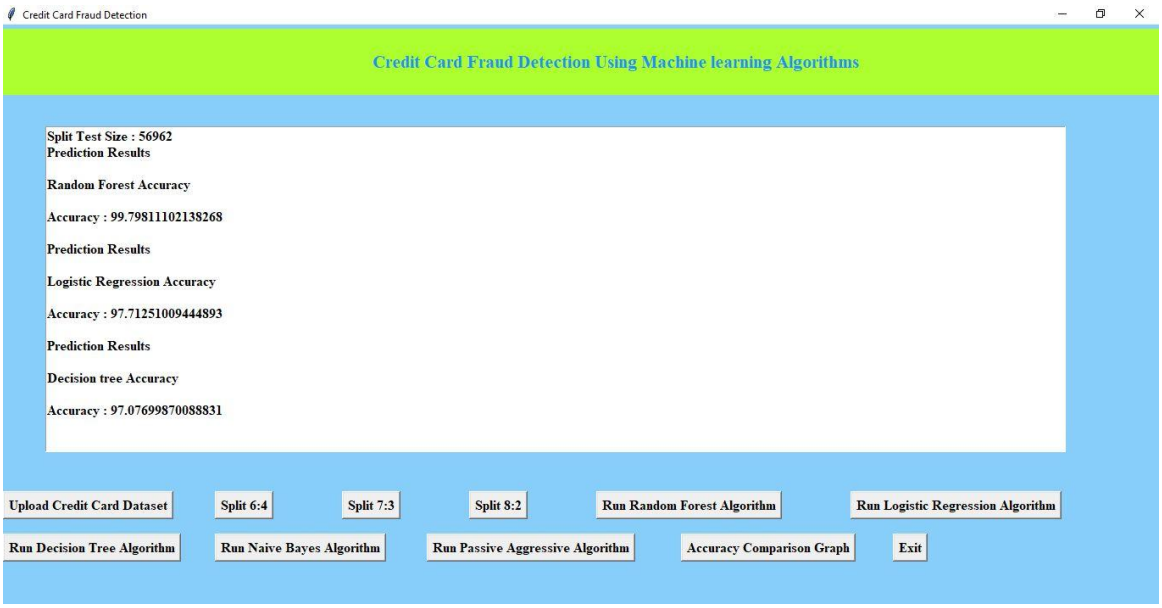



**Fig 6.6.3 Decision Tree Accuracy for 8:2**

## 6.7 NAIVE BAYES CLASSIFIER ACCURACY

By clicking the Run Naive Bayes Algorithm button to generate the Naive bayes classifier model on train and test the data. The Prediction Results of this Naive bayes Algorithm generated 99.32%, 99.32% and 99.29% accuracy as shown below in fig 6.7.1, 6.7.2 and 6.7.3


**Fig 6.7.1 Naive Bayes Classifier Accuracy for 6:4**

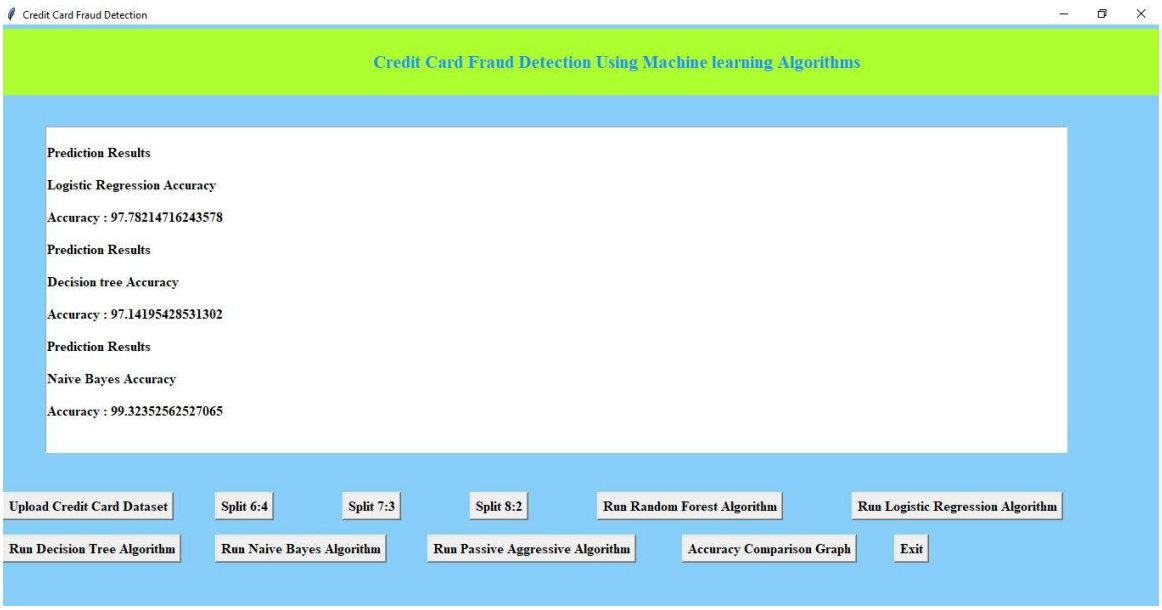



**Fig 6.7.2 Naive Bayes Classifier Accuracy for 7:3**

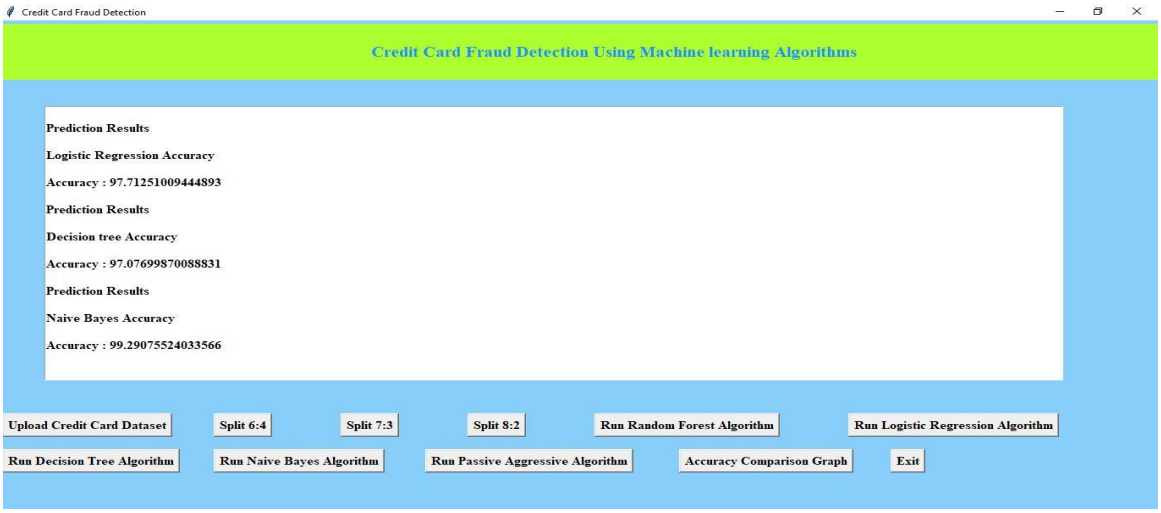



**Fig 6.7.3 Naive Bayes Classifier Accuracy for 8:2**

## 6.8 PASSIVE AGGRESSIVE ACCURACY

By clicking the Run Passive Aggressive Algorithm button to generate the passive aggressive model on train and test the data. The Prediction Results of this Passive Aggressive Algorithm generated 99.82%, 99.79% and 99.82% accuracy as shown below in fig 6.8.1, 6.8.2 and 6.8.3


**Fig 6.8.1 Passive Aggressive Accuracy for 6:4**

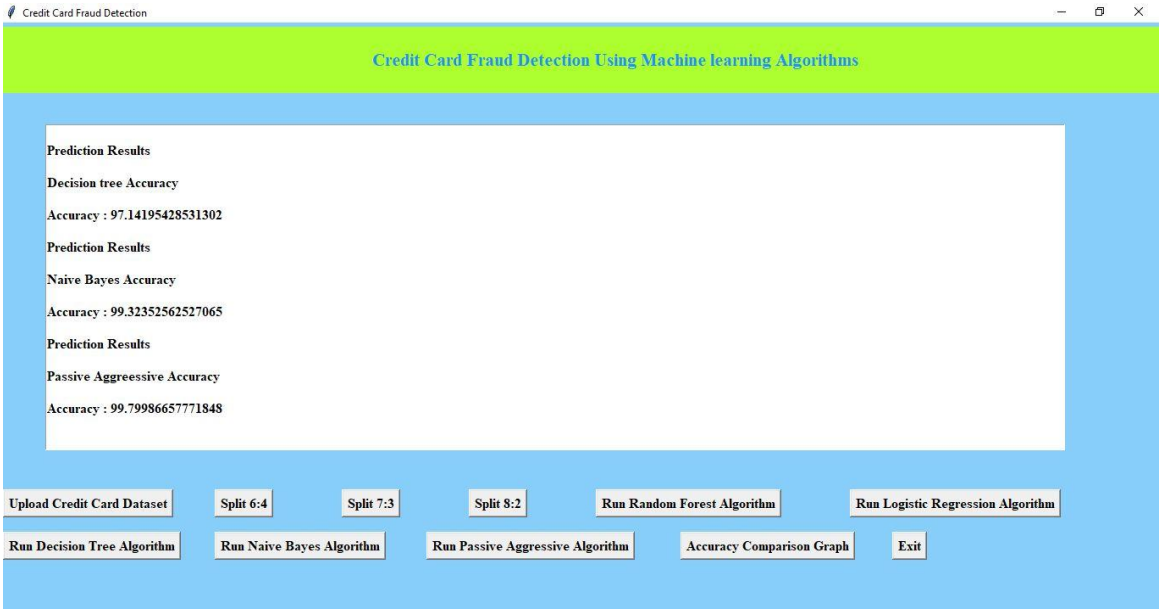



**Fig 6.8.2 Passive Aggressive Accuracy for 7:3**

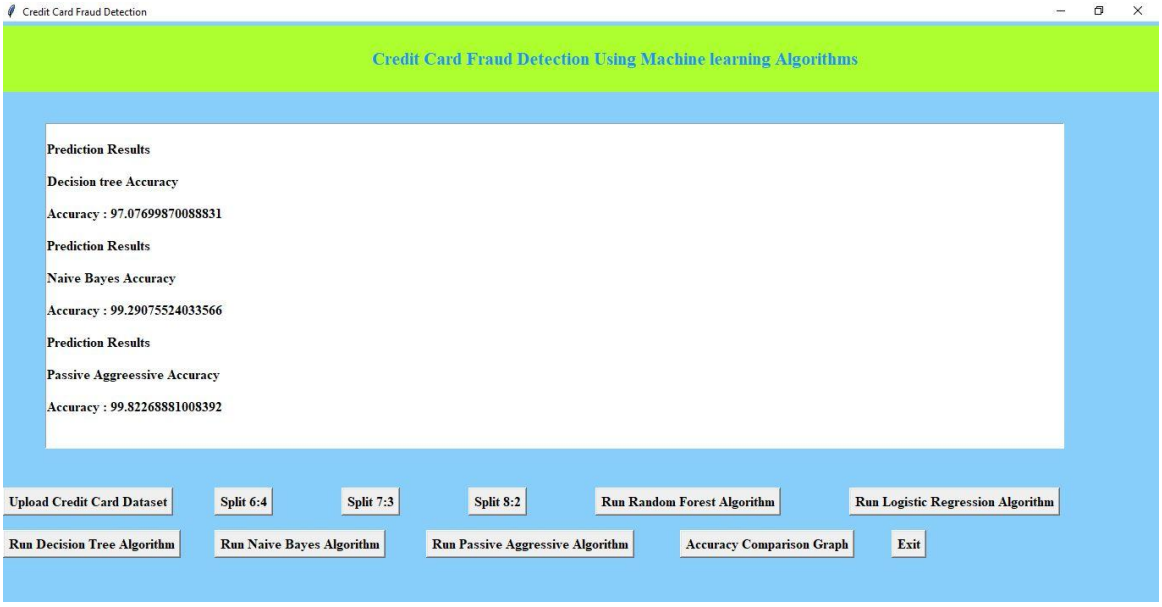



**Fig 6.8.2 Passive Aggressive Accuracy for 8:2**

## 6.9 ACCURACY COMPARISON

The Comparison of different machine learning algorithms such as Random Forest, Logistic Regression, Decision Tree, Naive Bayes and Passive Aggressive Algorithms of calculating accuracy results.

The comparison graph of different algorithm has been shown in Bar graph of Accuracies results of classification models of splits 6:4, 7:3 and 8:4 ratios of data.

The Passive Aggressive Algorithms shows as high and best accuracy results compared to all other machine learning algorithms.

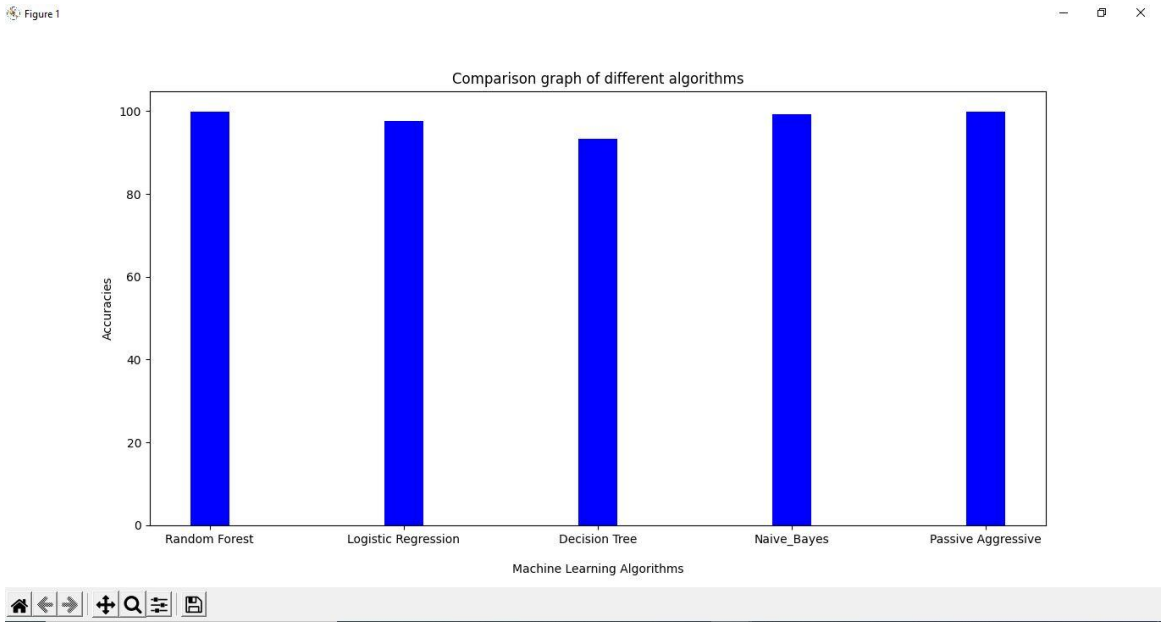



**Fig 6.9.1 Accuracy Comparison for 6:4**

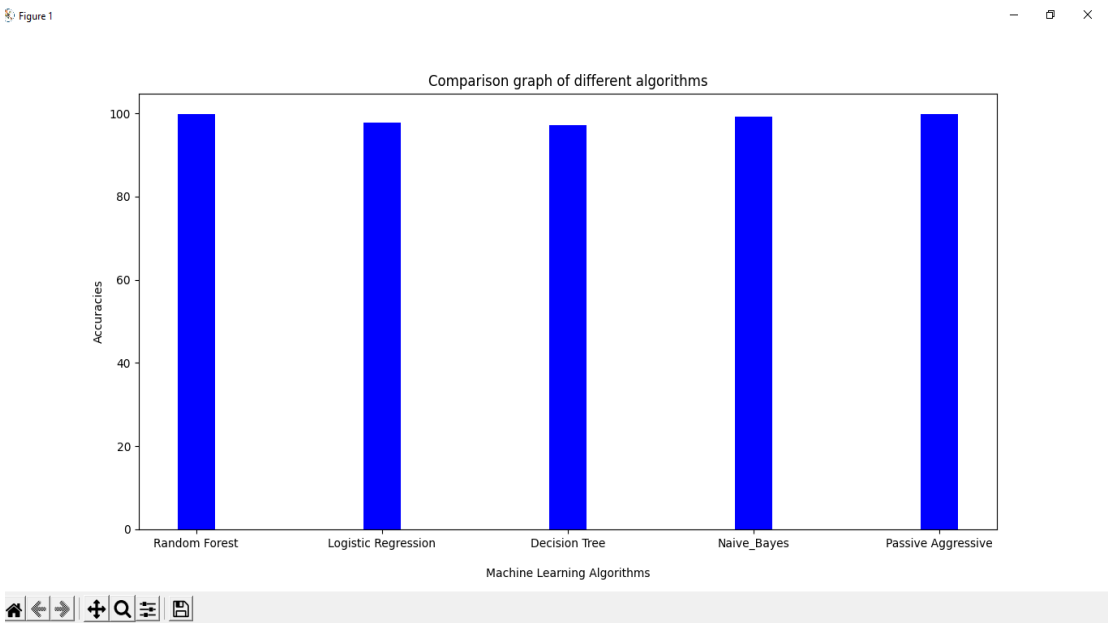



**Fig 6.9.2 Accuracy Comparison for 7:3**

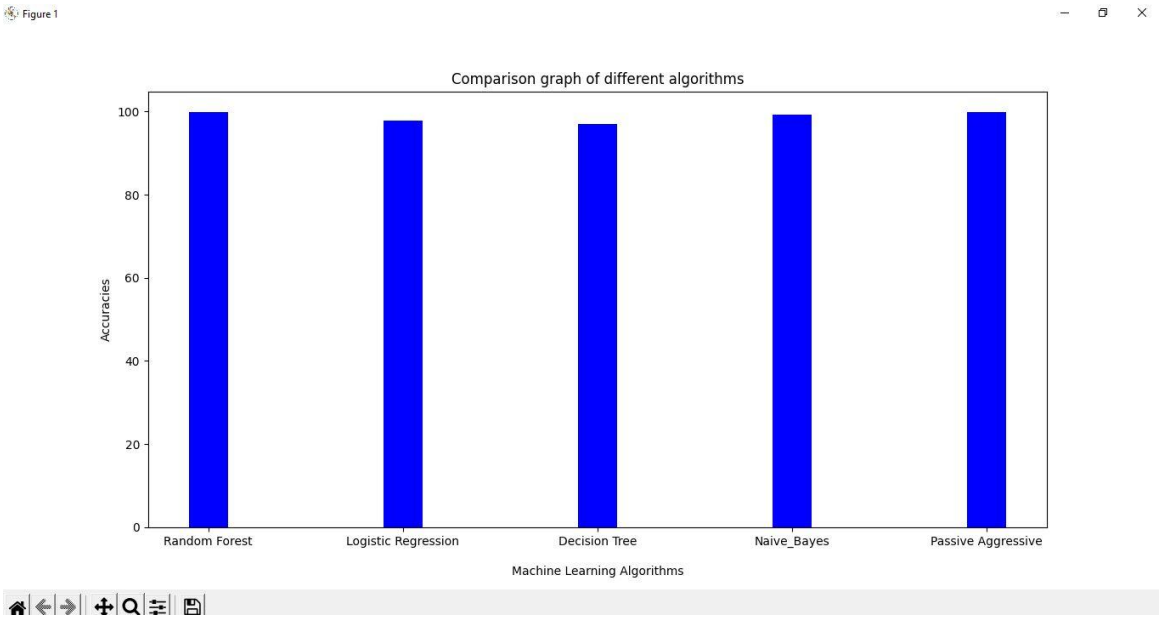



**Fig 6.9.3 Accuracy Comparison for 8:2**

By observing the graph, we conclude that for all splits 6:4, 7:3 and 8:2 ratios

- Passive Aggressive is best algorithm with 99.82%, 99.79% and 99.82% accuracy.
- Random Forest is classified with 99.82%, 99.78% and 99.79% accuracy.
- Logistic Regression is classified with 97.61%, 97.78% and 97.71% accuracy.
- Decision Tree is classified with 93.25%, 97.14% and 97.07%accuracy.
- Naive Bayes Classifier is classified with 99.32%, 99.32% and 99.29%accuracy.

# 7 CONCLUSION AND FUTURE SCOPE

A Credit card fraud detection system is developed using machine learning algorithms which is used to classify the transactions as frauds and non-frauds. In this project, a classification model is built using the machine learning algorithms like Random Forest, Logistic regression, Decision tree, Naive Bayes Classifier and Passive Aggressive algorithms and the performance is measured using accuracy and the results are compared between all the algorithms and the best algorithm that detects the credit card fraud transactions is specified. In this project, it is observed that for training and testing data of 6:4, the Passive Aggressive algorithm and Random Forest has best classified the credit card frauds with 99.82% accuracy, for training and testing data of 7:3, the Passive Aggressive Algorithm has classified with 99.79% accuracy and for training and testing data of 8:2, the Passive Aggressive Algorithm has classified with 99.82% accuracy. Thus, by this we conclude that Passive Aggressive algorithm has best classified the credit card fraud transactions.

This project can be improved further by building the model using different advanced algorithms to detect frauds in credit card transaction in GUI application. It can be used in real-time applications where the frauds occur more frequently. It can be applied to bank applications and many other applications where the credit card transactions are used. So, that by applying the model we can predict and classify the transactions lively.

# BIBLIOGRAPHY


[1]. A. A. Taha and S. J. Malebary, "An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine," in IEEE Access, vol. 8, pp. 25579-25587, 2020.

[2]. S P Maniraj and Aditya Saini, Swarna Deep Sarkar and Shadab Ahmed. "Credit Card Fraud Detection using Machine Learning and Data Science," in International Journal of Engineering Research and Technology on Computer Science and Engineering, vol. 8, Issue 9, 2019.

[3]. S. Makki, Z. Assaghir, Y. Taher, R. Haque, M. Hacid and H. Zeineddine, "An Experimental Study With Imbalanced Classification Approaches for Credit Card Fraud Detection," in IEEE Access, vol. 7, pp. 93010-93022, 2019.

[4]. P. A. Dal, G. Boracchi, O. Caelen, C. Alippi and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy", IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 8, pp. 3784-3797, 2017.

[5]. C. Jiang, J. Song, G. Liu, L. Zheng and W. Luan, "Credit Card Fraud Detection: A Novel Approach Using Aggregation Strategy and Feedback Mechanism," in IEEE Internet of Things Journal, vol. 5, no. 5, pp. 3637-3647,2018.

[6]. J. V. V. Sriram Sasank, G. R. Sahith, K. Abhinav and M. Belwal, "Credit Card Fraud Detection Using Various Classification and Sampling Techniques: A Comparative Study," 2019 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2019, pp. 1713-1718.

[7]. GJUS&T Hisar HCE, Sonepat , "Survey Paper on Credit Card Fraud Detection by Suman" , Research Scholar, published by International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3 Issue 3, March 2014.

[8]. CLIFTON PHUA1, VINCENT LEE1, KATE SMITH1 & ROSS GAYLER2, " A Comprehensive Survey of Data Mining-based Fraud Detection Research" published by School of Business Systems, Faculty of Information Technology, Monash University, Wellington Road, Clayton, Victoria 3800, Australia.

[9]. Massimiliano Zanin, Miguel Romance, Regino Criado, and SantiagoMoral, "Credit Card Fraud Detection through Parenclitic Network Analysis" published by Hindawi Complexity Volume 2018, Article ID 5764370, 9 pages.

[10]. David J.Wetson,David J.Hand,M Adams,Whitrow and Piotr Jusczak, "Plastic Card Fraud Detection using Peer Group Analysis" Springer, Issue 2008.

[11]. P.Richhariya and P.K. Singh, "Evaluating and emerging payment card fraud challenges and resolution," International Journal of Computer Applications, vol. 107. no. 14, pp.5-10, 2014

[12]. D.Sanchez, M.A. Vila, L. Cerda, and J.M. Serrano, "Association rules applied to credit card fraud detection", ScienceDirect,vol.36,pp.3630-3640, 2009.

[13]. A.E. Pasarica, "Card fraud detection using learning machines," The Bulletin of the Polytechnic Institute of Jassy, pp.29-45,2014.

[14]. F.N. Ogwueleka, "Data Mining Application in Credit Card fraud detection using self-organizing maps," Journal of Engineering Science and Technology, vol.6, no.3, pp.311-322, 2011.

[15]. V.Zaslavsky and A. Strizhak, "Credit card fraud detection using self-organizing maps," Information Security, vol. 18,pp. 48-63, 2006.

[16]. Z. Zojaji, R.E. Atani, and A. H. Monadjemi, "A survey of credit card fraud detection techniques: data and technique oriented perspective," Cryptography and Secuirty, 2016.

[17]. K.Ramakalyani and D. Umadevi, "Fraud Detection of Credit Card Payment System by Genetic Algorithm," Expert Systems with Applications. Vol.38.no.10, pp. 13057-13063, 2011.

[18]. V.R.Ganji and S.N.P. Mannem, "Credit card fraud detection using anti-k nearest neighbor algorithm", International Journal on Computer Science and Engineering(IJCSE), vol.4, no.06,pp. 1035-1039,2012.

[19]. Wen-Fang YU and Na Wang , “Research on Credit Card Fraud Detection Model Based on Distance Su ” published by International Joint Conference on Artificial Intelligence, 2009.

[20]. Ishu Trivedi, Monika, Mrigya, Mridushi , “Credit Card Fraud Detection” published by International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 1, January 2016.