CREDIT CARD FRAUD DETECTION USING SPECIFIC MACHINE LEARNING METHODS

BY

S.Akshaya

(16PCS001)

Under the guidance of

Dr.(Mrs) G.Padmavathi M.Sc., M.Phil.,Ph.D

Professor Department of Computer Science

A Project Report Submitted to

Avinashilingam Institute for Home Science and Higher Education for Women

(Deemed to be University)

Coimbatore-641043

In Partial fulfillment of the requirements for the Approval of

Master's Degree in Computer Science

April-2018

CREDIT CARD FRAUD DETECTION USING SPECIFIC MACHINE LEARNING METHODS

BY

S. AKSHAYA

(16PCS001)

A Project Report submitted to

Avinashilingam Institute for Home Science and Higher Education for Women

(Deemed to be University)

Coimbatore-641043

In Partial fulfillment of the requirements for the

Master's Degree in Computer Science

April-2018

Signature of the Supervisor **Signature of the Head of the Department**

Viva Voce examination held on _____

Signature of the External Examiner

ACKNOWLEDGEMENT

I would like to express my sincere thanks to God Almighty, for his constant love and grace that he showered upon me.

I would like to express my deep sense of reverential gratitude and sincere thanks to **Shri, Dr. P. R. Krishnakumar, Chancellor,** Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for his support and encouragement during the course of my study.

I owe my great deal of gratitude to **Dr. Premavathy Vijayan, Vice Chancellor**, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for extending all resources that facilitated the conduct of the present work.

I express my gratitude to **Dr. S. Kowsalya, Registrar**, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for providing all facilities necessary for the work.

I also thankful to **Dr. A. Parvathi, Dean Faculty of Science,** Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for granting the facility required.

I wish to place my deep sense of gratitude to **Dr. V. Radha, Professor and Head, Department of Computer Science,** for providing all the facilities required to complete the project.

I heartily thank my esteemed project guided by **Dr.G.Padmavathi Professor Department of Computer Science**, for imparting the tremendous assistance and well-timed support for triumph of my project.

I express my honorable thanks to my project coordinator **Dr. G. Sudhamathy, Assistant Professor, Department of Computer Science,** for her kind advice and knowledgeable suggestions which helped me to complete my project successfully.

I would like to extend my hearty thanks to one and all who helped me directly or indirectly for successful completion of my project.

Finally, I take pride to thank my beloved parents, my family members and my friends without whose support, encouragement and kind blessings .I would not have succeeded in my Endeavour.

CHAPTER I

INTRODUCTION

1. Introduction

The first chapter provides an introduction which is related to this project "Credit Card Fraud detection".it starts with cybercrime in general and then moves to fraud, credit card fraud and it types, detection methods since these are associated with project's main contribution.

1.1 Cyber Crime

Cyber Crime is an illegal activity which uses computer as an object of the crime either as a tool or a targeted victim. Cyber Crimes can be categorized into two ways computer as a tool and computer as a target.

There are so many varieties of crime which can be committed through Internet regularly; one way of using the computer act as a tool to commit crime when individual is the main target of cybercrime examples of such crimes include Cyber terrorism, IPR (intellectual Property Violations) violations, EFT (Electronic Fund Transfer) fraud, credit card frauds etc, while others are directed to the computer. Examples of such crimes are Denial of Service(DOS) attack, Hacking, Virus attacks etc.

1.2 Fraud

Frauds have evolved in 1994, and it has evolved in a faster rate since the beginning of E-commerce websites. Alexopoulos et al. (2007) defines fraud as "the intentional act perpetrated to achieve gain through illicit ways". Fraud can be inflicted everywhere including insurance companies, financial institutions, corporations etc., Figure 1 shows the different types of frauds.

1.2.1 Financial Fraud

The term financial fraud can be defined as an intentional act of deception involving in financial transactions for monetary gain. Credit card /Debit card fraud is one type of financial frauds. In this project Credit card /Debit card fraud are considered the same and several times credit card fraud is used throughout the rightup.

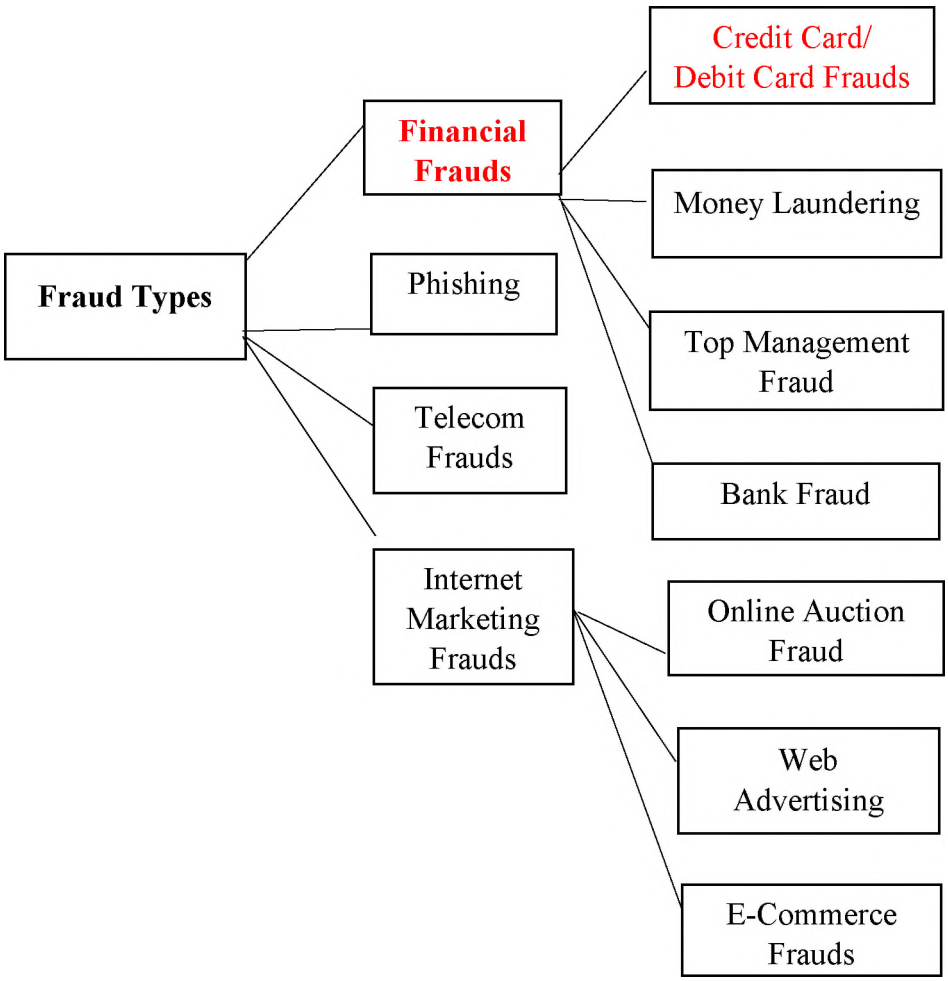


Figure 1: Taxonomy of Frauds

1.2.2 Credit Card Fraud

Credit card fraud is broad term used to represent the thefts where cardholders account including their personal information has been compromised which result in unauthorized transactions and result in loss of money for individual and financial institutions as well.

Credit card fraud is considered to be a one of a major threat to business establishment today. As large number of business are entirely depended on electronic data and computer network for their daily operations, all important documents and financial information is stored online. This can leave financial institutions and individuals exposed to privacy violations, data theft and so on., Cybercrimes have grown in frequency and great losses are on the rise. According to Laleh et.al. (2009) credit card fraud can be committed either through offline or online. These two modes are listed and are discussed below.

1.2.2.1 Offline Fraud

Offline Fraud occurs when card is physically stolen or lost at call center or any other place. Offline fraud was popular in early 2000's, more precisely cardholders tend to realize card has been lost or stolen and report to their respective bank before fraudster attempts to make illegitimate transactions using the card. As soon as the complaint has been registered in the bank, the latter will block the card so the fraudster can't use the card. Financial loss may occur if the cardholder does not realize the lost of their card lose.

1.2.2.2 Online Fraud

Online fraud is committed through Internet, phone, shopping, web, or in absence of card holder. The majority of credit card fraud occurs online or over the phone to make purchases which is also known as CNP or card-not-present transactions. This type of activity accounts for 45% of all fraudulent card usage.

Online fraud is also called as virtual card theft. This type of fraud can be more dangerous as the fraudsters can hold the information for a longer period of time before they use it so there is no way the card holder to know in advance that their information has been compromised. Therefore, online fraud can be only predicted after one or more illegal transactions might have taken place.

Credit card frauds have resulted in huge financial losses as the fraudulent transactions have been large value transactions and 47% of global credit card frauds occur in United States.

1.3 Types of Credit Card Frauds

Credit card frauds can be categorised into two types namely Traditional Techniques and Modern Techniques. Figure 2 represents the most common credit card frauds which are discussed in this section.

1.3.1 Traditional Techniques

Traditional Techniques are the oldest form of techniques used by fraudsters in late 2000s.Traditional Techniques are further classified into six types namely,

i. Application Fraud
ii. Assumed Identity
iii. Lost/stolen

iv. Identity Theft

v. Account Takeover

vi. Mail Non-Receipt Card Fraud

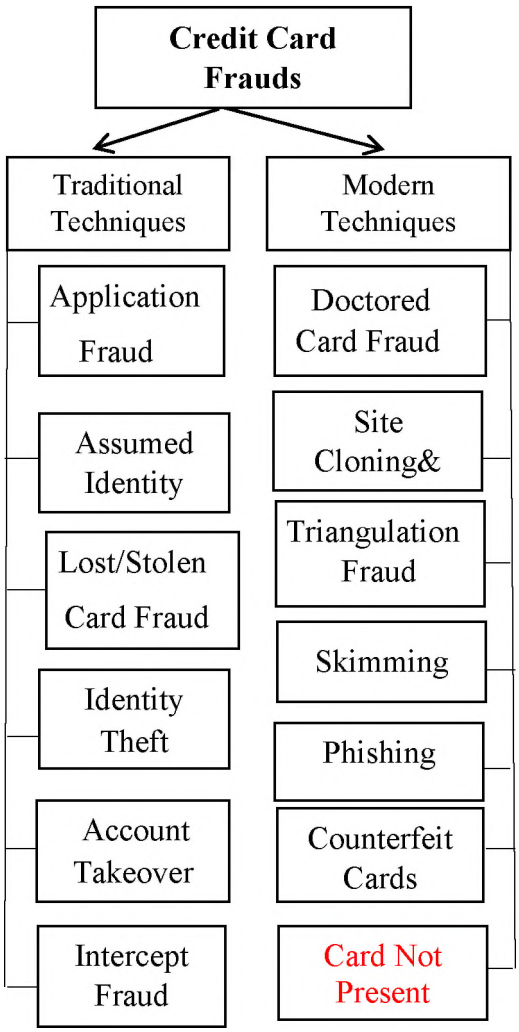


Figure 2: Types of Credit Card Frauds

1.3.1 .1 Application Fraud

Application fraud refers to fraud committed by submitting a new credit application with fraudulent details to a credit provider. Fraudsters normally, collect the personal and financial data of innocent users from the identity documents, bank statements, pay slips to commit the fraud.

1.3.1.2 Assumed Identity

Assumed Identity is the traditional form of Credit Card Fraud. Assumed identity fraud, a criminal will use a fake name and temporary address to claim credit card. It comes

under Application fraud. Fraudster will look for an individual who have moved recently so that electoral register will be expired. To confirm new address of the customer banks often check the electoral register. To commit this type of fraud the individual may also rent an apartment under false name or befriend an elderly person in order to give access to untraceable address to safeguard them..

1.3.1.3 Lost/Stolen

Lost/stolen card happens when the card is physically stolen either through theft or if the user has lost it. The criminals make unauthorized transactions of a credit card as a result of it being lost or stolen.

1.3.1.4 Identity Theft

Obtaining personal or financial information of another person for the purpose of assuming that person's name to engage in fraudulent activities.

1.3.1.5 Account Takeover

An account takeover can happen when a fraudster makes unauthorized transactions by gaining control over genuine user account.

1.3.1.6 Mail Non-Receipt Card Fraud

This type of fraud is also known as intercept fraud. In this case, the criminal will intercept the card and then register the card and use it to make purchases and more.

1.3.2 Modern Techniques

Modern Techniques are further classified into seven types namely,

i. Doctored Cards
ii. Site Cloning
iii. Triangulation Fraud
iv. Skimming
v. Phishing
vi. Counterfeit
vii. Card Not Present

1.3.2.1 Doctored Cards

A doctored card is a card whereby a strong magnet has erased its metallic stripe. Criminals do this and then manage to change the details on the card itself so that they match those of valid cards. Naturally, this card won't work when a criminal tries to pay for something. A doctored card is a simplest way for the fraudster can go about tampering with an existing card

1.3.2.2 Site Cloning

Criminals set up a website that identically looks like legitimate website with a slightly different address. Fraudsters deceive customers by creating a cloned website to get orders from customers and the customer completes their order by providing their credit card number. As a result, fraudster who obtained the customer's credit card information can use their information to make purchases at various ecommerce websites or actual shops.

1.3.2.3 Triangulation Fraud

Triangulation fraud occurs through online shopping website where fraudster provides goods at high discount rate and ship the product before payment. Fraudster site appear to be a genuine auction or traditional sales site. For accessing the website, initially customers have to provide their information including their name, address and credit card details to the site. Once the customer has ordered goods from their website fraudsters use another stolen credit card details to order goods from the legitimate site and supply goods to the customer who have ordered products from their site. Fraudsters initiate this process so that internet company to accumulate large amount of goods by using stolen credit card numbers. Authorities won't have enough time to catch the criminals as fraudsters close the website within a few weeks and operate a new site. It is a complex trail of fraudulent activity.

1.3.2.4 Skimming

Skimming is also referred as Electronic or manual credit card imprints Patidar et al. (2011) define Skimming as the "Fraudster skims information that is placed on the magnetic strip of the card and they used to encode a fake card to complete fraudulent transactions and they use a special purpose device called skimmers to capture the information of the credit card that are enclosed into their magnetic strip. Using this stolen

card information criminals credit counterfeit cards in order to use them for shopping or they supply the card information at various online shops. In earlier days, skimmer devices have also been used in ATM machines. In addition to that to record the PIN code of the cardholder during ATM transactions micro cameras have been used

1.3.2.5 Phishing

Fraudster sends an email to a user falsely posing as a legitimate enterprise in an attempt to deceive the user for obtaining private information of the user. Phishing email , generally direct the user to visit the website to update user personal information including their name, address, credit card details etc., and some spam emails might also include links to fraudulent websites which again deceit the victims into revealing their personal information.

1.3.2.6 Counterfeit

Legitimate cards are duplicated, which are then used for fraudulent activities. Counterfeit cards are used to make illegal transactions and purchases.

1.3.2.7 Card Not Present (CNP)

CNP fraud takes place when the customer makes online purchases. This type of fraud is committed when cardholders account number, name, card expiry date and billing address is known to a fraudster. Online phishing results in more number of CNP frauds (Unauthorized access of credit card information for fraudulent activities over the Internet, mail or phone). Merchants have to bear the loss when CNP fraud occurs. It will result in significant impact on merchant's profit, particularly for retail establishments which retain only small profit margin. Online shoplifting is a type of CNP fraud which is also termed as friendly fraud. Online shoplifting occurs through chargeback process. A consumer purchase goods using credit card, receives the goods and later the consumer claims that goods have not been received. As a result, Credit Card Company forces the merchant to refund customer's purchase.

1.3.2.8 False Merchant Sites

According to Patidar et.al. (2011,) these websites offer services at low price than rival sites. The site requests the customers to provide their personal information including their credit card details to access the content of the site. Most of these websites

claim to be free but ask for credit card information to verify individuals age. These sites try to collect as much credit card numbers as possible. These sites are usually a big part of criminal network that uses the details it collects to sell valid credit card details to other fraudsters.

1.4 Credit Card Fraud Statistics

Credit card companies have loss close to $50 billion dollars per year because of fraud. In mid-2000's two biggest credit card frauds took place where a gang of international fraudsters managed to steal the details of over 32,000 credit cards. Stolen credit card information was used to clone credit card and scam for at least 17 million over a period of several years. Another biggest fraud to date was committed by gang of criminals from New York where they have stolen up to $200 million. As per Federal Trade Commission, identity theft is the fastest growing white-collar crime in the United States. It is increasing on an average nearly 40 percent per year for the past several years. The Privacy Rights Clearinghouse estimates that each year 700,000 people in the United States are victims of identity theft. The Federal Trade Commission estimates that 10 million people are victimized by credit card theft each year. As per the Australia financial institutions data, card fraud has increased by 3.1% closely to $538.2 million. Card-not-present fraud increased 10%, now accounting for 82% of all fraud on Australian cards. As per the Indian Computer Emergency Response Team (CERT-IN), it was reported that, at least one cyber-attack for every 10 mts in 2017. According to Nilson Report, card fraud losses may rose to US$ 21 billion in 2015 and it is expected to reach US$31 billion by 2020 in India and all over 27,482 cases of cybercrimes have been reported across the world.

Figure 3: Statistics of Card Fraud Projection World Wide:2010-2020

Figure 3 shows the statistics of Card Fraud Projection World Wide from 2010-2020.As credit card fraud is seen to be high in number, it has resulted in huge financial loss. Moreover, 47% of global credit card frauds occur in United States. The majority of credit card fraud occurs through online as there is rapid growth of Internet-based purchases to be seen and consumer also make purchases over the phone which result in CNP or card-not-present transactions. This type of activity accounts for 45% of all fraudulent card usage. According to the statistics report from Australia financial institutions, card fraud increased by 3.1% to $538.2 million: card-not-present fraud increased 10%, now accounting for 82% of all fraud on Australian cards.

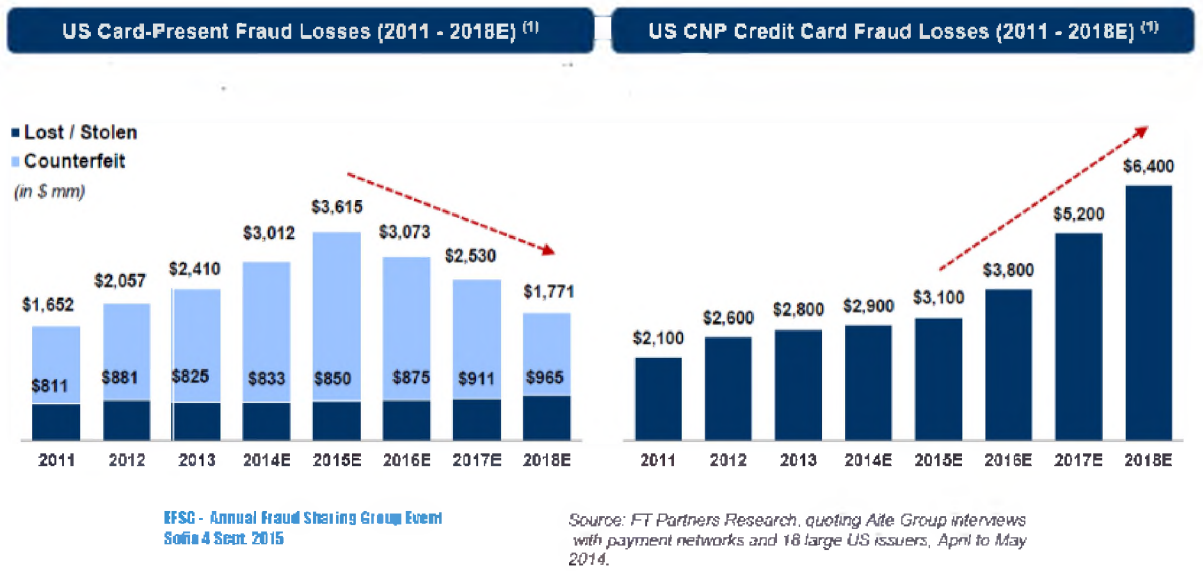


Figure 4: Comparison of card present and card not present Fraud losses in 2011-18 statistics

The above Figure 4 represents the comparison of Card Present and Card Not Present, fraud cases shows that Card Not Present frauds are high in number compared to Card Present Frauds.

1.5 Motivation

Credit card fraud has become a major challenge in recent times which result in huge financial loses for banking industry, financial institutions and credit card holders as well. As credit card frauds are on the rise, it affects millions of people each year. According to the report from Barclays major credit world's credit card frauds happen in U.S and most of the frauds occur through online, CNP (Card-Not-Present) fraud is most vulnerable than all forms of credit card fraud so it is imperative to detect these types of frauds as there is significant rise of credit card frauds globally.

1.6 Objective of the Project

The objective of the project is to develop an application for credit card fraud detection especially for Card Not Present category and compare the performance of specific machine learning methods based on the standard metrics for Credit Card Fraud Detection.

1.7 Overview of the Project

An application is developed using 'R' package to compare the performance of certain machine learning methods like K-Nearest Neighbor (KNN) and Support Vector Machine(SVM) for credit card fraud detection especially CNP category which is the most vulnerable types of credit card fraud. clustering is used for exploratory data analysis to find hidden patterns or grouping in data. Two validation meaures namely Internal and Stability measures are done to find optimal scores of clusters. UCI bench mark dataset is taken for experiment.

1.8 Organisation of the Report

- **Chapter 1** discusses about the Cyber Crime and fraud in general and then about to Credit Card Frauds, it types and some statistics particularly.
- **Chapter 2** discusses about the different Credit Card Fraud detection methods, comparison of certain methods and its literature review.
- **Chapter 3** discusses about the Machine Learning Techniques and their Classification.
- **Chapter 4** discusses the proposed approach of the project.
- **Chapter 5** discusses, presents the results and finally chapter 6 presents the conclusion and future scope.

Conclusion

Credit card fraud have risen exponentially over the past few years and unfortunately, fraud is one of the main challenges consumers have to deal with in their credit card ratings. As credit card is used for making purchases also, it is widely used in ecommerce website for online payment of purchases. Anyone who knows the details of card can make fraudulent transactions. Currently, card holder comes to know only after the fraudulent transaction is carried out. Credit card frauds often bear a recognizable pattern. With the help of machine learning, monitoring system can be trained to gather and analyze abnormal patterns predict accuracy of fraudulent transactions based on performance metrics.

CHAPTER II

BACKGROUND STUDY AND COMPARISON

This chapter discusses about various credit card fraud detection methods, its comparison and literature review.

2.1 Credit Card Fraud Detection Methods

Different credit card fraud detection methods have been discussed here, as fraud detection methods are used to monitor behavior of credit card users to detect undesirable behavior. To identify different credit card fraud effectively, it is necessary the understand the methods involved in detecting credit card frauds.

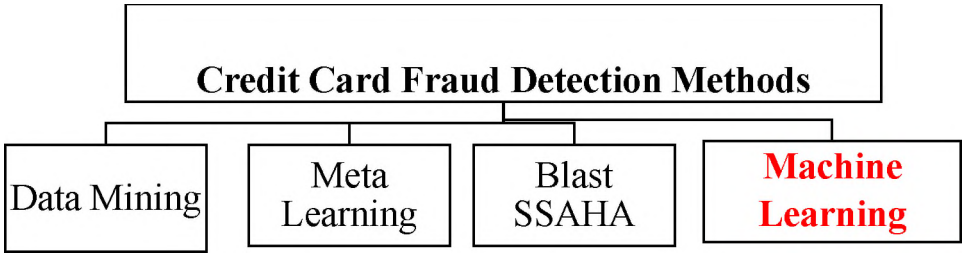


Figure 5: Credit Card Fraud Detection Methods

2.1.1 Data Mining

Data mining techniques are very efficient in detecting fraud because of its efficiency in discovering or recognizing unusual or unknown patterns in a collected dataset. Data mining is simply a technology that allows the discovery of patterns from a dataset. It is also known as knowledge discovery database (KDD). Data is collected from different sources and make predictions based on the patterns discovered.

2.1.2 Meta Learning

Meta Learning strategy aims to filter legitimate transaction from fraudulent ones by applying combiner strategy. To train multiple base classifiers combiner strategy uses attributes and correct classification in credit card transactions to identify fraudulent transactions. Four main stages in meta learning process to detect fraud: first, base classifiers are established using a training dataset that consist of 50% of fraudulent transactions and randomly choosing equal number of legitimate transactions. Using validation dataset base classifiers are applied to generate base predictions. Validation set consist of all transactions. To produce meta classifier, meta algorithm is applied to this combined dataset. For forward looking predictions meta classifier is applied to testing data.

2.1.3 BLAST-SSAHA

In BLAST-SSAHA are efficient sequence alignment algorithms used for credit card fraud detection for analysing the spending behaviour of customers. The hybridization of BLAST and SSAHA algorithm is also known as BLAH-FDS algorithm. BLAH-FDS is a two sequence alignment algorithm, where profile analyser (PA) is used to find similarity of incoming transactions on credit card based on legitimate user past spending behavior. The unusual transactions which represents past fraudulent behavior are passed to the deviation analyser (DA) by the profile analyser(PA). The final result about the nature of the transaction is based on the observations by these two analysers.

2.1.3.1 Fusion Approach using Dempster-shafer theory and Bayesian learning

In FDS of Dempster-Shafer theory and Bayesian learning is a hybrid approach for credit card fraud detection which evaluates current as well as past behavior. FDS system consists of four main components, namely dempster Shafer rule-based filter transaction history database and Bayesian learning. In rule-based component, suspicious levels of incoming transactions which deviate from actual pattern are considered to be a fraudulent transaction. By combining multiple evidences, dempster Shafter's theory computes initial belief to combine its belief to apply this theory. Depending upon the initial belief the transactions is classified as suspicious or unsuspicious. Once the transaction is found to be suspicious, the belief is further extended or weakened depending upon its similarity with genuine or fraudulent transaction history using Bayesian learning.

2.1.4 Machine Learning

Machine Learning is a statistical based model, main idea behind this technique, is discovering pattern in a collection of data instances in an automated manner to detect fraudulent transactions. Accuracy of the fraud detection problem is more important than correct classification. The following technique are used to detect fraudulent transactions first, transactions are filtered by checking efficient conditions (example: sufficient balance) and then scored by predictive model. Predictive model assign score based on the risk of the fraud (high or low) assign score for each transactions and those with high risk, alert is generated based on high risk.

2.2 Comparison of Various Credit Card Fraud Detection Methods

In order to Compare a various credit card fraud mechanisms. Comparison table was prepared. All the techniques described in the table (1) have its own strength and its weakness. Survey of Comparison table enables to build a hybrid approach for developing some effective algorithms for providing better results for the existing approaches.

Table 1: Comparison of Various Credit Card Fraud Detection Methods

Methods	Advantages	Limitations	Observations
Data Mining	It helps in analysing and Predicting large amount of Data. Detect frauds by comparing number of id's and number of attempts to gain access in E-commerce Service	Accuracy is medium Compared to other methods.	It is an important part of knowledge discovery process that can analyze an enormous set of data and get hidden and useful knowledge.
Meta Learning	Meta Learning Algorithm create optimal solutions for handling uncertainty in complex domains and recognize patterns in Fraudulent Transactions.	Performance estimation may be unreliable because of Natural limitations of estimating the true performance of the dataset.	Two methods of Combiner Algorithms. Used to detect Fraudulent Transactions.
BLAST-SSAHA	Accuracy is high and Processing Speed is fast	Cannot detect duplicate Transactions or cloned Credit card frauds.	Based of Analysers (profile and deviation analyser) Conclusion is drawn and final decision is taken
Machine Learning	Accuracy is high compared to Other methods	Processing speed Is considerably Very low	Intrusion Detection in Many database applications Ecommerce.

Table 2 Literature Review on Credit Card Fraud Detection Techniques

Table 2 Discuss about the references of Literature review of Credit Card Fraud Detection Techniques

Authors	Year	Techniques	Observations
Srivastava et.al	2008	Data Mining	The fraud is detected by analysing Spending behaviour of the user.
Joseph pun, yuri Lawryshyn	2012	Meta Learning	Improvement in Catch fraud than Neural Network
Vatsa et.al	2014	Blast -SSAHA	The fraud is detected using a gam theoretic method. In this model th Communication between the Attacker and the FDS will be as a Multistage game played between two players where in trying to exploit.
Bolton, R. & Hand, D.	2001	Machine Learning	Unsupervised Profiling Methods for Fraud Detection. Credit Scoring and Credit Control.

Conclusion

Due to rapid growth of ecommerce websites, credit card fraud has become profuse in number in recent years so it is vital to detect the fraud as soon as possible to reduce huge amount of financial loss in shopping websites as well as financial institutions. Fraud detection methods will be essential to prevent the user accounts from fraudsters. There are many methods available to detect credit card frauds.in In this section four existing technique of credit card fraud detection methods have been briefly discussed and these four detection methods have been compared. It is concluded that Machine learning is considered to be most popular detection method as it is accuracy rate is high in predicting fraudulent methods compared to other methods.

CHAPTER III

MACHINE LEARNING METHODS

This project applies specific machine learning methods for credit card fraud detection. Therefore, this chapter discusses machine learning methods and their important classification.

3.1 Introduction

In 1997, Tom Mitchell Carnegie Mellon University defined **Machine Learning** as, "A computer program is said to learn from experience (E) with respect to some task (T) and some performance measure (P), if its performance on (T), as measured by (P), improves with experience (E)." Machine learning is one of a branch in computer science which enables the system to learn automatically and also study the design of the algorithm to work on it. It is considered to be a type of Artificial Intelligence (AI) which enables the software application to predict accurate outcomes without explicitly programming the same. The basis of machine learning is to receive input data and predict outcomes based on analysis. Machine learning tasks helps to find predictive outcomes or predict modeling based on the learning which is obtained through available data, experience or instructions. It is often categorized into two types namely Supervised Learning and Unsupervised Learning. However, a third category is also available. Supervised learning requires input data and desired output to furnish accurate predictions during training data. Once the training is completed the algorithm will be applied based on the learning experience to the new data. In unsupervised learning, desired outcomes need not to be trained with the algorithm, instead deep learning is used to draw conclusions. Unsupervised learning is mostly used for complex tasks.

3.2 Classification of Machine Learning Methods

Machine learning methods are broadly classified into three types namely, supervised learning, unsupervised learning and reinforcement learning. Supervised learning methods are further divided into two types namely, Linear and Non-Linear methods. Similarly, Unsupervised learning methods are categorised into three types Clustering, Expectation Maximization, Self Organising Map (SOM) .Figure 7 shows the broad classification of Machine Learning methods.

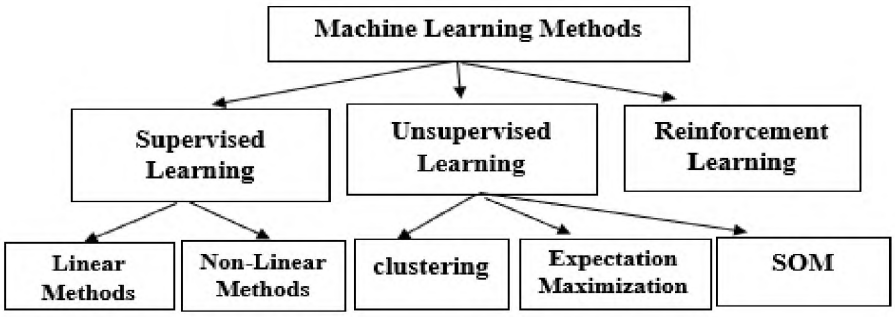


Figure 7: Classifying Machine Learning Methods

3.2.1 Supervised Machine Learning: The program is "trained" on a pre-defined set of "training examples", which then facilitate its ability to reach an accurate conclusion when given new data.

Supervised learning is categorized into two types namely,

- Linear Methods
- Non-Linear Methods

3.3.1.1 Linear Methods

A linear classifier uses object characteristics to identify which class or (group) it belongs to, this is achieved by making a classification decision based on linear combination of the characteristics.

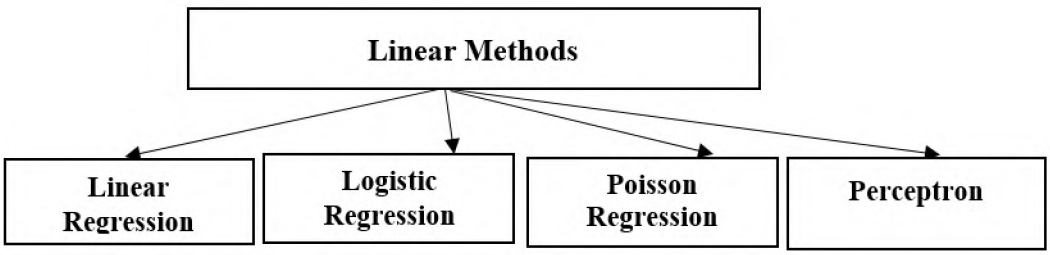


Figure 8: Types of Linear Methods

Linear Regression

A linear Regression is a statistical method that allows to summarize about single independent variable which is use to predict the value of dependent variable. There are two types of Linear Regression: Simple Regression and Multiple Regression.

Simple Regression is used to predict the value of the dependent variable by using single Independent Variable.

Multiple Regression uses two or more independent variables to predict the values of dependent variable.

By analyzing the relationship between two variables, a linear equation is plotted to observe data. To determine the strength of the relationship between the two variables, scatter plot can be used. A linear equation of form is used $Y = a + bX$, where X is the explanatory variable and Y is the dependent variable. The slope of the line is b, and a is the intercept (the value of y when $x = 0$).

Logistic Regression

Logistic is also a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable in which there are two or more possible outcomes. The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest and a set of independent variables. It is initially converted to linear format then converted back to original probability after the analysis. Odds ratio is the ratio of probability of success, to the probability of failure.

$$\text{Odd Ratio} = \frac{P}{1-P}$$ this varies from 0 to ∞

The general mathematical equation for logistic regression is:

$$Y=1/(1+e^{-(a+b_1 x_1+b_2x_2+b_3x_3+\ldots..+b_nx_n)})$$

In this equation y is the responsible variable, $a, b_1, b_2 \ldots\ldots b_n$ are the coefficients are $x_1, x_2, \ldots\ldots x_n$ are the predictor variables. The function used to create the logistic regression model is the glm() function.

Poisson Regression

Poisson Regression involves regression models in which the response variable is in form of count and not fractional numbers. The poisson regression is represented by equation given below

$$\log(y) = a+b_1+x_1+b_2+x_2+b_nx_n\ldots.$$

In this equation y is the response variable, a, b_1, b_2,....b_n are the coefficient and x_1,x_2,.....x_n are the predictor variables. The function used to create the Poisson regression model is the glm () function. glm() is one of the function in R package.

Perceptron

Perceptron is a supervised learning method of binary classifiers where an input represented by set of vector of numbers belong to a specific class. Perceptron make predictions based on linear predictor function using feature vector by combining set of weights. The method allows the processing elements in the training set one at a time based on online learning. As this algorithm is used for learning a binary classifier, it maps an input(x) to an output value f(x) which is a single binary value.

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

Where w is a vector of real valued weight, w.x is the dot product $\sum_{i=1}^{m} w_i x_i$, where 'm' is the number of inputs to the perceptron and b is the bias. The bias does not depend on any input value. The value f(x) is used to classify x either as a positive or negative instance (0 or 1). If b is negative, then the weighted inputs must produce a positive value greater than |b| to push the neuron classifier over the 0 threshold. If the learning set is not linearly separable, the algorithm will not terminate. Due to this all vectors will not be classified properly.

3.2.1.2 Non-Linear Methods

When data is non-linearly separable it can be non-linearly used so it can represent large class of functions. Non-linear will provide more accurate predictions than linear classifier. Non-linear is classified into single and combined. Figure 9 represent, the different types of Non-Linear Methods.

Non-Linear are categorised into two ways single model and hybrid model. Single model is classified into 8 types and hybrid model is classified into 3 types which is explained below

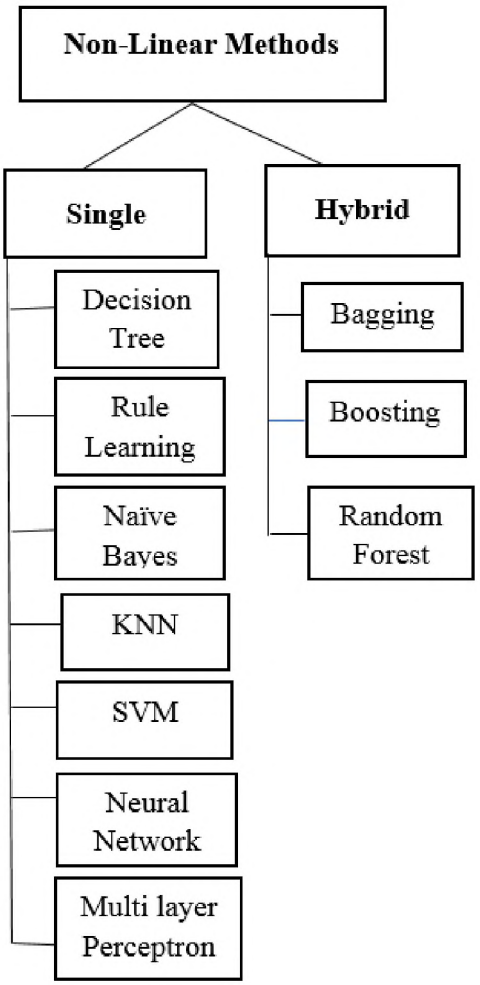


Figure 9: Types of Non-Linear Methods

Single Model

Single model has seven methods which is represented in Figure 9 are explained below.,

Decision tree

Decision tree is a supervised learning approach where internal nodes in the tree represents the observation of the item's target value. It is one of the predictive modelling approach used in machine learning. In tree models, target variable takes discrete set of values called classification trees. In this tree, leaves and tree structure represent class labels and branches represent conjunctions of features that leads to those class labels. In decision trees, target variables can take continuous values which is typically referred as regression trees. Using decision analysis, a decision tree can be used to explicitly represent decisions and decision making. Based on the several input variables the goal is to create

the models that predict the value of target variable. A tree can be learned by splitting the source into subset based on the attributes value. The process is repeated on each derived subset which is called recursive partitioning.

Decision tree Types

Decision tree is mainly of two types namely, Classification tree analysis is used to predict outcome to which class the data belongs. Regression tree analysis produce predicted outcomes as a real number.The term classification and regression tree(CART) analysis was first referred by Brieman et al. trees used for regression and classification both have same similarities but there are some differences as well such as the procedure used to determine where to split.

Rule Based Learning

Rule based learning encompass any machine learning method that learns, identifies or evolve rules to manipulate, store or apply. The main characteristic of rule-based machine learning is the utilization and identification of a set of relational rules that collectively represent the knowledge captured by the system. Rule based machine learning applies some form of learning to identify useful rules for each contextual knowledge. Rule based machining learning approaches include learning classifier systems, artificial immune systems and association rule learning.

Naïve Bayes

Naïve Bayes classification is named after Thomas Bayes who proposed the Bayes Theorem. It is supervised learning algorithm which can be used for binary or multi-class classification problems. Based on hypothesis it allows to capture uncertainty about the model by determining probabilities of the outcomes in a principled way. Naïve Bayes classification is used as a probabilistic learning method.it is mostly used to spam filtering (filter spam emails) and classify text documents.

Bayes theorem

Machine learning is often focused in selecting best hypothesis (h) in a given data(d). By knowing the prior knowledge of the problem most probable hypothesis can be selected. Using prior knowledge Bayes Theorem can calculate the probability of the hypothesis. Bayes Theorem is stated as:

$$P(h|d) = (P(d|h) * P(h)) / P(d)$$

Where

- **P(h|d)** is the probability of hypothesis (h)given the data (d). This is called the posterior probability.
- **P(d|h)** is the probability of data(d)given that the hypothesis h was true.
- **P(h)** is the probability of hypothesis (h) being true (regardless of the data). This is called the prior probability of(h).
- **P(d)** is the probability of the data (regardless of the hypothesis).

  Posterior probability can be calculated for number of different hypotheses, maximum probable hypothesis is called maximum a posteriori (MAP) hypothesis.

It can be written as:

$$MAP(h) = max(P(h|d))$$

or

$$MAP(h) = max((P(d|h) * P(h)) / P(d))$$

$$MAP(h) = max(P(d|h) * P(h))$$

P(d) is use to calculate the probability and it is only use to normalize.

It is called naïve Bayes because to make calculation of the probabilities tractable, the probabilities hypothesis is simplified. Instead of calculating the values of each attribute value, they are assumed to be conditionally independent for given target value and calculated as P(d1|h)*P(d2|H).

Naïve Bayes Models Representation

Naïve Bayes representation is through probabilities which includes class probabilities and conditional probabilities

Class Probabilities: It refers to probabilities of each class in a training dataset. Class probabilities is the frequency of instances which can be calculated by dividing each class by total number of instances.

ConditionalProbabilities:

In conditional probability each input value given a class value. Conditional Probabilities denotes the frequency of each attribute value in a given class value which can be calculated by dividing the frequency of instances with that class value.

K-Nearest Neighbor Classifier

The k-nearest neighbour is an instance-based learning which carries classification based on similarity measure. Euclidean distance is used in KNN classifier. In dataset every data point is calculated using the Euclidean distance between the input point and the current point.is calculated. Distance is sorted in an increasing order. The lowest distance of the data point is selected for the input data point. For input point the classifier returns the majority of the class as a classification.

Predictions using KNN

Predictions for new instance(x) is done through searching the entire training set by finding K most similar instances(neighbors) and output variables are summarized for those K instances. This might be a mean output variable for regression and mode (most common value) for classification. Euclidean distance measure is used to determine which of the K instance in the training dataset are most similar to a new input. Euclidean distance is calculated as the square root of the sum of squared differences between the new point(x) and the existing point (xi) across all the input attributes j.

EuclideanDistance(x, xi) = sqrt(sum ((xj – xij) ^2))

Some of the popular distance measure include:

Hamming Distance: Distance between binary vectors are calculated.

Manhattan Distance: Manhattan Distance is also called as City Block Distance which is used to calculate the distance between the real vectors using the sum of their absolute difference.

Minkowski Distance: It is the generalization of Manhattan and Euclidean distance.

There are many other distance measures that can be used, such as Jaccard, Mahalanobis, Tanimoto and Cosine distance. Based on the properties of data distance metric can be

choosed. Euclidean is considered to be a good distance measure to use if the input variables are similar in type.

Computational Complexity of KNN may increase with the size of the training dataset. KNN can be made as stochastic by taking a sample from the training dataset where k-most similar instances are calculated.

Neural Network

Neural network is an information-processing machine and can be viewed as analogous to human nervous system. Just like human nervous system, which is made up of interconnected neurons, a neural network is made up of interconnected information processing units. The information processing units do not work in a linear manner. In fact, neural network draws its strength from parallel processing of information, which allows it to deal with non-linearity. Neural network becomes handy to infer meaning and detect patterns from complex data sets. A neural network is a model characterized by an activation function, which is used by interconnected information processing units to transform input into output. A neural network has always been compared to human nervous system. Information in passed through interconnected units analogous to information passage through neurons in humans. The first layer of the neural network receives the raw input, processes it and passes the processed information to the hidden layers. The hidden layer passes the information to the last layer, which produces the output. In fact, neural network draws its strength from parallel processing of information, which allows it to deal with non-linearity. Neural network becomes handy to infer meaning and detect patterns from complex data sets. Artificial Neural Network (ANN) is used to classify an input into a group from a set of predefined groups (for example, fraudulent and non-fraudulent transactions). In the case of ANN predictor, the ANN returns the prediction of a certain input data. If the results present a great difference with the prediction it will mean that this sample should be deeply investigated. The advantage of neural network is that it is adaptive in nature. It learns from the information provided, i.e. trains itself from the data, which has a known outcome and optimizes its weights for a better prediction in situations with unknown outcome.

Support Vector Machine (SVM)

The Support Vector Machine (SVM) is suitable for binary classification technique such as credit card fraud detection which uses only binary classes namely, legitimate and fraudulent class. SVM requires large training dataset to achieve maximum accuracy. SVM method constructs hyperplane which separate the data into two classes i.e., Positive and Negative. SVM is based on two main properties kernel representation and margin optimization. Kernel can be used to learn complex regions. This basic technique finds the smallest hypersphere in the kernel space which contains all training instances, and then determines which side the test instance lies in the hypersphere. The hyperplane with maximum margin is the one which gives the greatest separation between the classes. The instances which is nearest to the maximum hyper plane is the support vectors. The fraud is detected is based on test instances, if the test instances lies within the learned region, it is considered as legitimate; else it is declared as fraudulent one.

Multi-Layer Perceptron

A multilayer perceptron (MLP) is a feed forward Neural network which utilizes supervised learning techniques. This technique applies back propagation method for training. It is distinguished from a linear perceptron by using multiple layers and non-linear activation. A standard perceptron is calculated using continuous function

$$x7: \rightarrow f_{step}(w_0 + \mathrm{h}w, x\mathrm{i})$$

A multilayer perceptron is a finite acyclic graph. Neurons of i^{th} layer have input features for neurons $(i+1)^{th}$ layer where nodes are neurons with logistic function. MLP is typically applied to input patterns using n dimensions, where each dimension has n input neurons. Figure 10 shows the three layers of multi-layer perceptron.

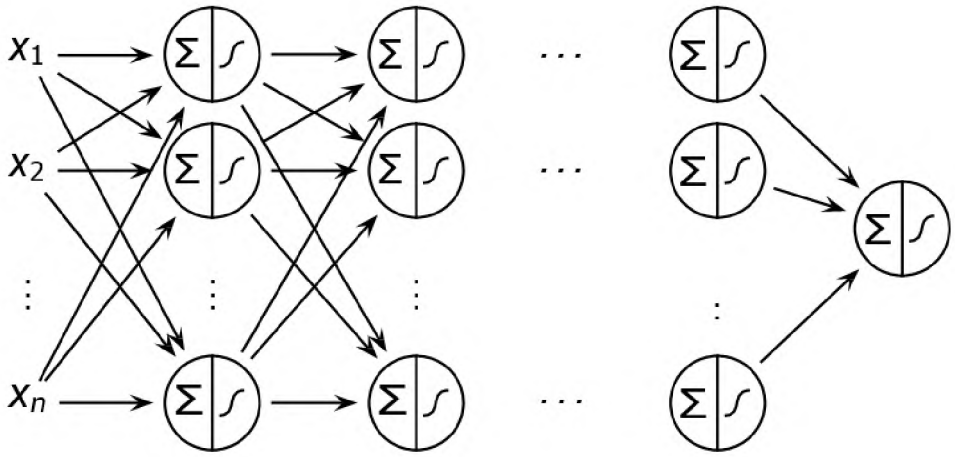


Figure 10: Multi-Layer Perceptron

The number of output neurons depends upon how the target values are described in the training patterns. A node that is neither input nor output is called hidden neurons. Since MLP is a acyclic graph all nodes can be organised in layers. MLP is a connection structure where all neurons of one layer are connected to all neurons of next layers which are numerated without any shortcut process. All neuron connections are weighted using a real number.

Succ(i) is the set of neuron(j) for which a connection(i)-> j exists.

Pred(i) is the set of neuron(j) for which a connection (j)-> i exists.

Hidden and output neurons have some variables i network input.

Algorithm (forward pass): for MLP

Require: Pattern x, MLP, Enumeration of all neurons in topological order

Ensure: calculate output of MLP

1: **for all** input neurons *I* **do**

2: set $a_i \leftarrow x_i$

3: **end for**

4: **for all** hidden and output neurons *I* in topological order **do**

5: set $net_i \leftarrow w_{i0} + \sum_{j \in Pred(i)} w_{ij} a_j$

6: set $a_i \leftarrow f_{log}(net_i)$

7: **end for**

8: **for all** output neurons I **do**

9: assemble a_i in output vector y

10: **end for**

11: **return** y

Machine Learning ensembling method helps in combining several models. This method provides better prediction model compared to a single model.

Hybrid Method

Hybrid methods are the combination of more than one models. The most common way of combining models are bagging, boosting and random forest. Figure 10 represents three types of hybrid methods which are explained below

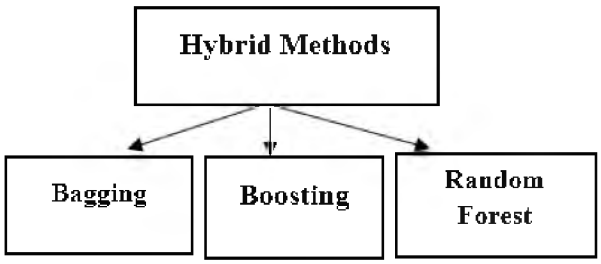


Figure 11: Three Types of Hybrid Methods

Bagging

Bagging cart is an ensemble technique which can handle classification and regression methods. This technique can be used as a variation reduction technique by randomization its construction procedure and then creating an ensemble out of it. It works by combining the classification of randomly generated training set to find final prediction. Bagging cart is considered to be a smoothing operation which improves the performance of classification and regression trees. In order to classify a new instance, it is done repeatedly to each tree using ensemble. Each class is chosen based on a voting procedure in a tree. The final prediction for new instance is gaining maximum values by the class.

Bagging when used with decision tree algorithm can be used in a highly imbalanced dataset and also weight the result of the tree and reducing variance of the dataset and the over fitting.

Boosting

Boosting has been used to convert weak learners to strong learners. The main process of boosting is to fit sequences of weak learners models that are only slightly better than the random guessing such as small decisions for the weighted versions of data. Boosting is not algorithmically constrained, most boosting algorithms consist of repetitive weak classifier with reference to a distribution system at last can be added to the strong classifier. When strong classifier is added they are typically weighted in some way that they relate to the weak learners accuracy. After a weak learner is added the examples that are classified correctly lose weight (some boosting algorithms decrease weight repeatedly, for eg., boost by majority) and the examples which are misclassified gain weight. Thus, future weak learners focus mostly on examples to check previously weak learners that are misclassified.

Random Forest

Random forest is an ensembling method for classification as well as regression technique, the higher the number of trees the higher will be the accuracy. As random forest is supervised learning method training is required where ensemble is built from a sample with a replacement. In random forest, algorithms are changed for the way the sub trees are learned in order to have less correlation for the resulting prediction. In CART, spilt points are selected by looking through all the variables values in order to find optimal spilt point. Random forest algorithm changes this procedure for the ease of learning algorithm to limit to the random sample of features for which to search. Parameters are specified based on the number of features that can be spilt point to the algorithm.

3.3 Unsupervised Learning

Unsupervised learning is a type of machine learning technique which is used to draw inferences from datasets of input data without labeled references. Figure 11 represents different types of unsupervised learning.

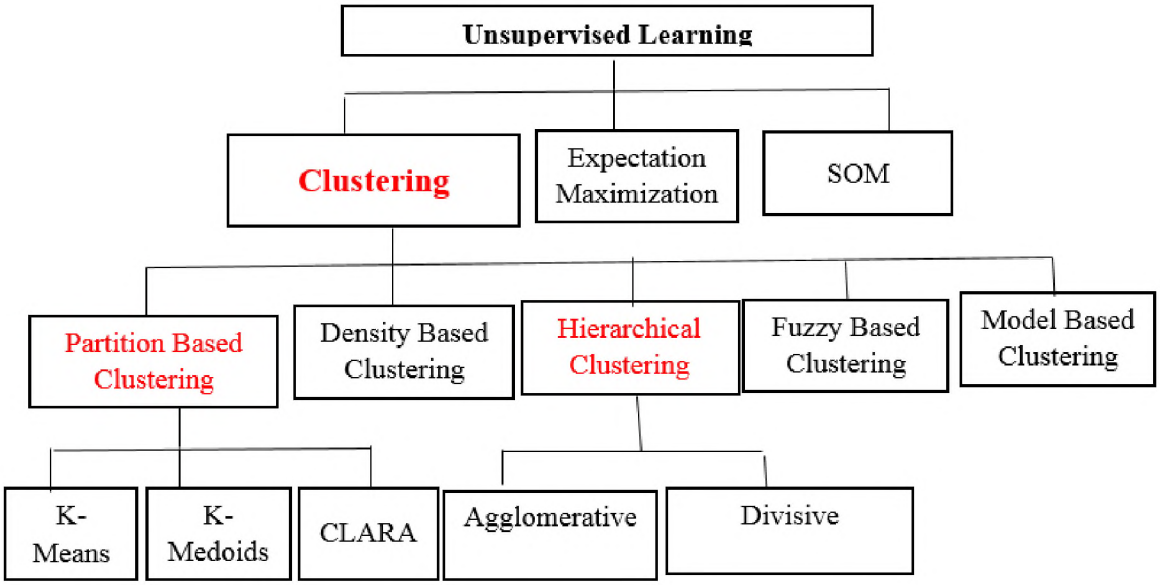


Figure 12: Types of Unsupervised learning

Clustering

The most common unsupervised learning method is cluster analysis, which is used for data analysis to find hidden patterns or grouping in data. Based on similarity, clusters are modeled which is measured using a Euclidean or Probabilistic distance metric.

Partitioning Based Algorithm

Partitioning algorithms are clustering approaches that split the datasets, containing n observations, into a set of k groups (i.e. clusters). The algorithms require the analyst to specify the number of clusters to be generated.

The most common partition algorithms are:

- K-means clustering
- K-medoids clustering
- CLARA Clustering

K-Means Clustering

The popular clustering algorithm that minimizes the clustering error is the K-means algorithm. K-means algorithm is first applied to an N-dimensional population for clustering them into *k* sets on the basis of a sample by MacQueen in 1967. The algorithm is based on the input parameter *k*. First of all, *k* centroid points are selected randomly. These *k* centroids are the means of *k* clusters. Then, each item in the dataset is assigned to a cluster

which is nearest to them. Means of all clusters are calculated again with new points added to them, until values of means do not change. The global k means clustering algorithm which constitutes a deterministic effective global clustering algorithm for the minimization of the clustering error that employs the k means algorithm as a local search procedure. The k means clustering algorithm progress in an incremental way to solve the clustering problem.

K-Medoids Clustering

K-medoids clustering or PAM (Partitioning Around Medoids) is defined by Kaufman & Rousseeuw, 1990, in which, each cluster is represented by one of the objects in the cluster. It is a "non-parametric" robust alternative to k-means clustering, less sensitive to outliers. Partitioning around medoids (PAM) is similar to K-means, but is considered more robust because it admits the use of other dissimilarities besides Euclidean distance.

Density Based Clustering

Density-Based spatial clustering with noise was first introduced by Ester et.al. (1996). In this clustering algorithm, given a set of points in some space it groups together points that are closely packed together marking as outliers points that lie alone in low-density regions. The function dbscan () in the package fpc is used for the density-based clustering. Unlike k-means, number of cluster need not be specified as a parameter for DBSCAN. It assumes the number of clusters based on the data and it discovers cluster based on arbitrary shape. Approximate local density ε-neighborhood is fundamental in DBSCAN clustering. The algorithm has two set of parameters:

- ε:minPts: To define a cluster minimum number of data points are required in a neighbourhood.
- DBSCAN categorise data points into three types based on using these two parameters.
- Core Points: A data point (p) is referred as core point if Nbhd (p,ε) which contains at least minpts, |Nbhd(p,ε)|>=minPts.
- Border Points: A data point (*q) is referred as border point if Nbhd (q, ε)
- Outlier: A data point *o* is an outlier if it is neither a border point nor core point.

Hierarchical clustering

A hierarchical clustering is often displayed graphically using a tree-like diagram called a dendrogram, which displays both the cluster-sub cluster relationship. Hierarchical clustering is an agglomerative clustering algorithm that yields a dendrogram which can be cut at a chosen height to produce the desired number of clusters (Kaufman and Rousseeuw 1990). Each observation is initially placed in its own cluster, and the clusters are successively joined together in order of their "closeness". The closeness of any two clusters is determined by a dissimilarity matrix, and can be based on a variety of agglomeration methods.

Hierarchical clustering can be subdivided into two types namely, Agglomerative and Divisive.

Agglomerative hierarchical clustering algorithms start with each object being a separate cluster itself, and successively merge groups according to a distance measure. The clustering may stop when all objects are in a single group or at any other point the user wants. These methods generally follow a greedy-like bottom-up merging.

Divisive hierarchical clustering algorithms follow the opposite strategy. They start with one group of all objects and successively split groups into smaller ones, until each object falls in one cluster, or as desired. Divisive approaches divide the data objects in disjoint groups at every step, and follow the same pattern until all objects fall into a separate cluster. This is similar to the approach followed by divide-and-conquer algorithms.

Fuzzy Based Clustering

In fuzzy based clustering each data point belongs to more than one cluster. Based on the similarity between items data points are clustered, clustered items sets should be as similar as possible to each other and dissimilar items to be as much same in other groups. It is much easier to create fuzzy boundaries than to settle one data points in more than one cluster. To find optimal location for any data points fuzzy clustering uses least square method. Optimal location may reside in a probability space between two or more clusters. Fuzzy based clustering algorithms are categorised into two areas: Classical Fuzzy clustering and shape based fuzzy clustering.

Classical Fuzzy Clustering Algorithms

Classical Fuzzy algorithms are classified into four types namely,

- Fuzzy C-Means Algorithm
- Gutafson-Kessel Algorithm
- Gath-Geva Algorithm
- Shaped -based Fuzzy Clustering Algorithm

Fuzzy C-Means Algorithm(FCM)

FCM is identical to k-means algorithm. Using a membership function the data points can theoretically belong to all groups. Membership function has two grades between 0 and 1, where 0 refers to the data point which is farthest possible point from cluster's center and 1 refers to data point which is closest to the center. FCM subtypes include, Possibilistic C-Means(PCM), Possibilistic Fuzzy C-Means (PFCM) and Fuzzy Possibilistic C-Means(FPCM).

Gustafson-Kessel (GK)Algorithm

GK algorithm has elliptical shaped clusters while C-means assumes clusters are spherical shaped. Gk associates a data point with a cluster and matrix as well.

Gath-Geva Algorithm

It is similar to FCM cluster, in this clustering, clusters can be of any shape. Gath Geva algorithm is also called as Gaussian Mixture decomposition.

Shape-based fuzzy clustering algorithms

Shaped based fuzzy clustering algorithms are divided into three types namely, Circular Shaped, Elliptical Shaped and Generic Shaped.

Circular Shaped: This algorithm is incorporated in Fuzzy C-means so it is called CS-FCM.data points are in circular shape in CS-FCM.

Elliptical Shaped: This algorithm is used in GK algorithm that constraints points to elliptical shapes.

Generic Shaped: This algorithm allows clusters of any shape.

Model Based Clustering

The Basic idea behind Model based Clustering is that in a mixture of two or more components, sample observations arise from a distribution. Each mixture has an associated probability model or "weight" which is described by a density function. Any probability model can be adopted for the components but typically it is assumed as p-variate normal distributions. Probability model will be a mixture of multivariate normal distributions. If components is assumed as a joint distribution then the mixture model would be of p-dimensional observations.

$$f(x|\pi, \mu, \Sigma) = \prod_{i=1}^{n} \sum_{k=1}^{G} \pi_k f_k(x_i|\mu_k, \Sigma_k)$$

$$= \text{const.} \prod_{i=1}^{n} \sum_{k=1}^{G} \pi_k |\Sigma_k|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(x_i - \mu_k)^j \Sigma_k^{-1}(x_i - \mu_k)\},$$

Where π_k is the probability that belongs to xi of K^{th} component ($0 < \pi_k < 1$, $\sum_k \pi_k = 1$).For modelling Σk, clusters in the mixture model are centered at μk with ellipsoidal shape. Attribute Σk determine the geometric feature of k component to parameterize each Σk in the mixture model as $\Sigma_k = \lambda_k D_k A_k D_k'$, Where D_k represents the orthogonal matrix of eigen vectors of Σk' A_k denotes diagonal matrix which is proportional to eigen values of Σk and λk denotes scalar. Model based clustering can be implemented using Mclust function in R. To find possible parameterization of the Covariance matrix as it represents three letters: E for equal, V for Variable and I for Coordinate axes. Three identifiers are designated to three letters namely, first identifier refers to volume, the second identifies shape and the third to orientation.

Expectation Maximization

Expectation Maximization is an algorithm used for likelihood estimation in models with hidden variables. It involves iteratively computing expectations of terms in the log-likelihood function under the current posterior, and then solving for the maximum likelihood parameters. Common applications include fitting mixture models, learning Bayes net parameters with latent data, and learning hidden Markov models. The best hypothesis for the distributional parameters is the maximum likelihood hypothesis – the

one that maximizes the probability that this data comes from K distributions, each with a mean m_k and variance sigma. $_K$ Normal distribution process is done. In a single modal normal distribution this hypothesis *h* is estimated directly from the data as:

$$\text{estimated } m = m\sim = \text{sum}(x_i)/N$$

The two steps of the EM algorithm are:

- E-step: perform probabilistic assignments of each data point to some class based on the current hypothesis *h* for the distributional class parameters;
- M-step: update the hypothesis *h* for the distributional class parameters based on the new data assignments.
- During the E-step we are calculating the expected value of cluster assignments are calculated.
- During the M-step, a new maximum likelihood for the hypothesis is calculated.

Self Organising Map (SOM)

The Self -Organising Map(SOM)belongs to the category of competitive learning networks and it is one of the most popular neural network models. SOM is based on unsupervised learning where there will be no human intervention required and only input data will be known. SOM is used to detect features related to the problem, thus it is also being called Self Organising Feature Map. To map units SOM uses topology preserving mapping from the high dimensional space. Mapping neurons or units is usually form a two-dimensional lattice. Points which are near to each other in the input space are mapped to nearby map neurons in the SOM. Thus, SOM can be used for cluster analysis for high dimensional data. Generalization is one of the important feature of SOM as it helps the network to recognize or characterize input data it has never encountered before. The SOM is a two-dimensional array of neurons: $\mathbf{M} = \{\mathbf{m}_1, ..., \mathbf{m}_{pxq}\}$.Based on neighborhood relation the neurons are connected to adjacent neurons. Distance between mapping units is defined according to topology relations. Neurons which are adjacent to belong to the neighborhood N_c of the Neuron $\mathbf{m}_c$.

Reinforcement Learning

Reinforcement Learning is a type of machine learning methods that allows software agents and machines to automatically determine the ideal behaviour within a

specific context in order to maximize its performance. Basic reinforcement is modeled as a Markov decision process: a set of environment and agent states, S; a set of actions, $P_a(s, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$ A, of the agent; is the probability of transition from state s to state s' under action a. $R_a(s, s')$ is the immediate reward after transition from s to rules that describe what the agent observes. Rules are often stochastic in nature. The observation typically involves the scalar, immediate reward associated with the last transition. In many works, the agent is assumed to observe the current environmental state (based on full observability). If not, the agent has partial observability. Sometimes the set of actions available to the agent is restricted (a zero balance cannot be reduced). Two elements make reinforcement learning powerful: the use of samples to optimize performance and the use of function approximation to deal with large environments. Reinforcement learning can be used in large environments in the following situations: A model of the environment is known, but an analytic solution is not available; Only a simulation model of the environment is given (the subject of simulation-based optimization). The only way to collect information about the environment is to interact with it.

Conclusion

This section briefly discussed various machine learning methods. All these techniques can be used for data analytics and visualization process to obtain optimal results. The next section briefly discusses the methodology used in this project.

CHAPTER IV

PROPOSED METHODOLOGY

A machine learning method can learn on data without depending on any rule-based programming, which provide a framework for studying the problem to gain knowledge, to make predictions from a set of data. Therefore Machine learning methods are found to be more effective in detecting subtle or non-intuitive patterns to identify fraudulent transactions in financial applications .Machine Learning Methods use complex algorithms to analyse patterns in a dataset. As fraudulent transactions tends to look like legitimate ones, it is necessary to evaluate the performance of fraud detection techniques to predict the accuracy, sensitivity and specificity are used to find the best algorithm in detecting fraud. Classification and clustering methods are used in this proposed approach. Classification is defined as a process of finding a set of models to differentiate data concepts for the purpose of being able to use the model to predict the class objects whose class label is unknown. Classification algorithms uses supervised learning model as it is used to produce the knowledge of already classified data and finds predictive pattern, The two types of classification algorithms used in this project are KNN, SVM.They are explained below: Comparative result shows that KNN is better than SVM. The most common unsupervised learning method is cluster analysis, which is used for exploratory data analysis to find hidden patterns or grouping in data. The clusters are modeled using a measure of similarity which is defined upon metrics such as Euclidean or probabilistic distance. Two clustering algorithms are compared using two clustering validation measures Internal and Stability measures. The results shows that hierarchical and PAM clustering provide better performance and the performance prediction is based on cluster validation measures. Figure 13 shows the machine learning methods used in this project. Therefore two different methods are derived based on classification and based on clustering.

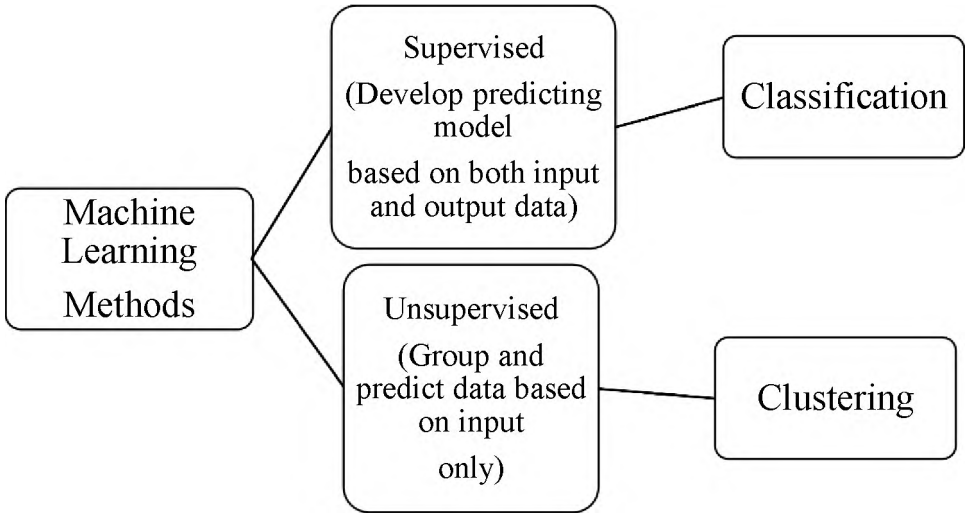


Figure 13: Machine Learning Methods used in this Project

4.1 Methodology Based on Classification

Classification and Clustering are the two Approaches used in this proposed work. Figure (12) is an Approach (1) for classification and methodology of classification approach is explained below, after comes the methodology for clustering approach(2).

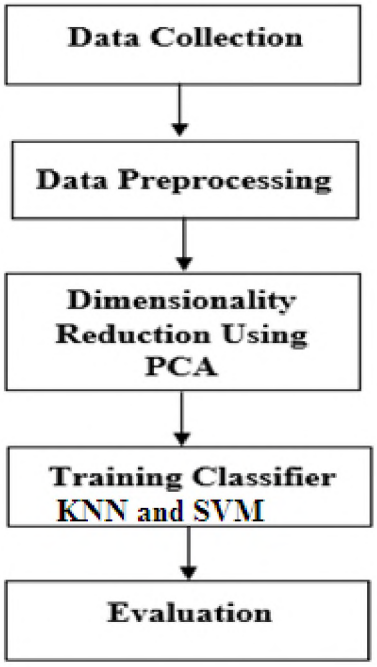


Figure 14: Methodology of Classification Approach (1)

4.1.1 Data Collection

The Dataset taken for study is the benchmark dataset. The world line and the Machine Learning Group of Université Libre de Bruxelles(ULB) database contains credit card transactions of European card holders over a two-day collection period in September 2013.This project uses 'R' Package for all the steps.

4.1.2 Data Preprocessing

In Data Preprocessing the raw data is converted into usable format by reducing noise and removing null values. Data preprocessing mainly concludes with data cleaning, integration and transformation. By this way incorrect and irrelevant data can be removed and useful dataset can be extracted.

4.1.3 Dimensionality Reduction Using Principal Component Analysis

When the input data to an algorithm is too large to be processed and it is suspected to be redundant, then the input data will be transformed into a reduced representation set of features (also named as feature vector). Transforming the input data into the set of features is called feature extraction. If the features extracted are carefully chosen it is expected that the features set will extract the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full-size input. As a result of Principal Component Analysis transformation, the dataset is of numerical variables. To maintain the confidentiality of sensitive information this transformation is applied. Time variable is the first transaction in the dataset that contains the second elapsed between each transaction. The Amount variable is the transaction Amount as this feature can be used as an example-dependant cost-sensitive learning. Amount and Time variables which are not transformed by PCA. Class variable indicate whether the transactions are fraudulent or not. In the training stage, the processed data are fed into a classifier algorithm and then the experiments are evaluated using three Performance metrics. The processed dataset contains only numerical input variable which is a result of PCA.This feature selection transformation results 28 principal components. Performance comparison of the classifiers is evaluated based on accuracy, specificity and sensitivity.

4.1.4 Training Classifiers

Three classification algorithms namely KNN, SVM and Neural Network are trained to build a model for prediction to derive the results. Before train **()** method, trainControl **()** method is used to control the computational nuances of the train **()** method for all three classifiers. The "number" parameter holds the number of resampling iterations. The "repeats" parameter contains the complete sets of folds to compute for repeated cross-validation. This train Control **()** methods returns a list which is passed to the train **()** method.

First KNN (k-Nearest Neighbor) is trained using knn () function which is used to train a model. The knn () function identifies the k-nearest neighbors using Euclidean distance where k is a user-specified number. Before Train and Test data split, data is distributed randomly. In R, sample () method is used which helps to randomize all the records of data frame. In SVM, Caret package provides train **()** method for training for the data.

4.1.5 Evaluation

The Performance of the algorithm are based on three metrics namely Accuracy, Sensitivity and Specificity. While building a predictive model it is necessary to evaluate the capability of the model of the unseen data. This can be done by estimating accuracy using data that are not used to train the model such as a test set or by using cross validation. Accuracy can be estimated using repeated K-fold Cross estimation. Repeated k-fold estimation is defined as the process of splitting the data into K-folds which is repeated for a number of times. The final model accuracy is taken as the mean for number of repeats. It is the extent to which a test measures what it is supposed to measure; It is the accuracy of the test. Validity is measured by sensitivity and specificity. Sensitivity is the ability of a test to correctly classify a transaction as fraud. It is the Probability of being test positive when fraudulent transactions are present. Specificity is the ability of a test to correctly classify the transaction as Non-Fraud. Probability of being test negative when fraudulent transactions are not present.

Positive Prediction Value (PPV) represents how many of test positives are true positives; and if this number is greater than the threshold value 0.5, it is considered to be fraud, if the test is correctly classifying the fraudulent transactions then it results in True Positive.

Negative Predictive Value (NPP) represents the one that does not detect the condition when the condition is absent. NPP denotes True negative. When the threshold is below 0.5 value it is represented as Non- Fraudulent Transactions. Here, Probability of (normal transaction where fraud is not present which result in test is negative)

Two Classifiers used in this proposed work namely K-Nearest Neighbor and Support Vector Machine.

4.1.6 K-Nearest Neighbor(KNN)

A k-nearest neighbor (KNN) is an approach to data classification that estimates how likely a data point is to a member of one group or other, depending on what group the data points are nearest to it.

KNN Classifier

The k-nearest neighbor is an instance-based learning which carries out its classification based on a similarity measure, like Euclidean, Mahanttan or Minkowski distance functions. The first two distance measures work well with continuous variables while the third suits categorical variables.

The Euclidean distance measure is used in this experiment for the KNN classifier. The Euclidean distance using the equation.For every data point in the dataset, the Euclidean distance between an input data point and current point is calculated. These distances are sorted in increasing order and k items with lowest distances to the input data point are selected.

The majority of class among these items are found and the classifier returns the majority class as the classification for the input point. Parameter tuning fork is carried out for k = 1, 3, 5, 7, 9, 11 and k = 3 showed optimal performance. Thus, value of k = 3 is used in the KNN Classifier. When KNN is used for classification, the output can be calculated as class with high frequency from k-most similar instances. Each instance vote for a class and the class with most vote is taken as prediction. Probability of class can be calculated as the normal frequency that belongs to each class in set of k instance for each new instance.

$$P(class=0) = count(class=0) / (count(class=0)+count(class=1)$$

KNN Algorithm

- Take the instance x to be classified
- Find k nearest neighbors of x in the training data.
- Determine the class c of the majority of the instances among the k nearest neighbors.
- Return the class c as the classification of x.
- In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression:

The steps are as follows,

- In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer)
- If k = 1, then the object is simply assigned to the class of that single nearest neighbor. Classification can be used to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones.
- The training examples are vectors in a multidimensional feature space, each with a class label.
- In training phase, the algorithm consists only of storing the feature vectors and class labels of the training samples.
- In Classification, k is a user-defined constant, and an unlabeled vector (a query or test point) is classified by assigning the label which is most frequent among the k training samples nearest to that query point.
- A commonly used distance metric for continuous variables is Euclidean distance. The training examples are vectors in a multidimensional feature space, each with a class label.
- The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples.
- In classification accuracy of KNN can be improved significantly if the distance metric is learned with specialized algorithms such as Large Margin Nearest Neighbor or Neighborhood components analysis.

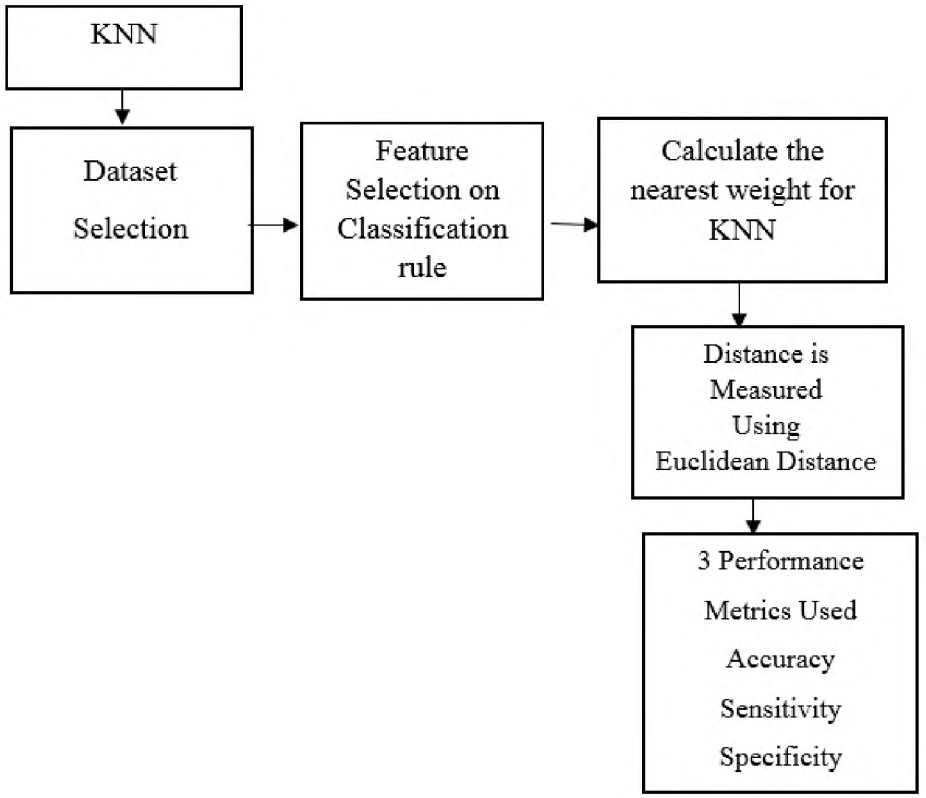


Figure 15: KNN Fraud Detection Process

4.1.7 Support Vector Machine(SVM) Classifier

Support Vector Machine (SVM) was developed from the theory of Structural Risk Minimization. SVM is used in binary classification to classify into two categories; fraudulent or non-fraudulent. The decision function of SVM is shown in equation:$f(x)$=sgn $(x.w)+b$....*eqn (1)*

Where x, is an input vector which contain weight value and value b remains constant. Decision boundary of SVM is represented in eqn(1). In Training phase SVM has to learn the parameter value of w and b which are derived by maximizing the margin of separation between two classes. Based on the margin maximization criterion is used between two classes. The distance between the two hyperplane is to find a hyper plane H: y= w.x+ b = 0 ..eqn is used and the equation of two hyper plane are as follows H1: y = w.x+ b= +1 and H2: y = w.x + b = -1. In SVM, hyper plane is used to best separate the points in the input variable space based on their class, either 0 or 1. Using threshold two classes are separated resulting H and two margin boundaries are H1 and H2. As there are two hyper planes, 2 margin is obtained, resulting 2/||w||, where ||w|| represents the norm of

the vector w. Misclassification error is reduced by introducing a slack variable. The optimization problem for the calculation of w and b can be defined by the eqn (2),

$$Min\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \quad \ldots \text{eqn(2)}$$

eqn (2), the complexity of SVM is reduced by minimizing the $\|w\|^2$ and misclassification errors are reduced by minimizing the slack variable. C is a regularization parameter which denotes the classification error and it is the tradeoff between two classes

$$L(w,b,\xi,\alpha,\beta) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\{y[wx+b]-1+\xi_i\} - \sum_{i=1}^{n}\beta_i\xi_i \quad \text{(eqn:3)}$$

eqn (3), gives the solution for the optimization problem which is reduced by minimizing w, b & ξ and maximizing α & β. It is better to solve the problem by introducing the dual formation in eq (4).By substituting the eqn(4) the problem is transformed with dual formation

$$\max\left\{\sum_{i=1}^{n}\alpha - \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \langle x_i, y_j\rangle\right\}$$

eqn (4) this equation is maximised under the constraints,

$$\sum_{i=1}^{n}\alpha_i y_i = 0 \text{ and } 0 \le \alpha_i \le C \text{ for } i = 1,2,\ldots,n. \quad \text{(eqn:5)}$$

The Kuhh Tucker Condition which is also known as karush -Kuhh Conditions is applied to eqn(4) to eqn(5). Kuhh tucker conditions is used for theconstraint optimization problem. Mathematical optimization is a first-order necessary conditions for a solution in nonlinear programming to be optimal, provided that some regularity conditions are satisfied. Allowing inequality constraints, the KKT approach to nonlinear programming generalizes the method of Lagrange multipliers, which allows only equality constraints. The system of equations and inequalities corresponding to the KKT conditions is usually not solved directly, except in the few special cases where a closed-form solution can be derived analytically. In SVM, the decision boundary is determined by the eqn. $f(x)$=Ns

$i = 1$ (eq:6)

where x is the input vector, (x, x_i) represents inner product, b is the bias term and Ns is the number of support vectors. In this proposed work, kernel function is used to linearly separate data and all support vectors lay on the margin. Kernel function provides a bridge between linearity to non-linearity which can be expressed in terms of dot product. It defines the similarity or distance measure between new data and support vectors. Kernel function is determined by the equation $\langle x_i, x_j \rangle \rightarrow k(x_i, x_j)$

Input vectors are mapped into a higher dimensional feature space by using a kernel function.

Linear Kernel function

Linear kernel function is the simplest of all kernel function $k(x_i, x_j) = x_i, x_j . \langle x, y \rangle$ is a given inner product plus an optional constant c.

To calculate SVM data should be correctly classified. The following calculations are required to calculate SVM.

[a] If $Y_i = +1; wx_i + b \geq 1$

[b] If $Y_i = -1; wx_i + b \leq 1$

[c] For all i; $y_i (w_i + b) \geq 1$

The goal of the SVM model is to predict the target value of data instances in the testing set with a set of given attributes. SVM can handle high-dimensional input data, only fewer learning parameters are required so it does not suffer from local minima.

Linear Kernel Function works based on two basic principles. They are,

- To define a maximize margin for optimal hyper plane.
- Data is mapped to high dimensional space where it is easier to with linear decision surface: to map data implicitly to this space the problem has to be reformulated.

SVM Process

Usually a dataset is divided into two categories training set and testing set. First step is to train a set of data .In testing set data points which are used in the training set are excluded. Training set can be defined as the set of examples used for learning to fit the

parameters of classifier. In SVM, training set is used to find the optimal support vectors. In validation set, set of examples were used to tune the parameters of the classifier. To find the optimal number of support vectors or to assess the performance of a fully trained classifier validation is used. After choosing the final model, estimate error rate is checked (FP rate or TP) rate. Next step is to scale feature vectors, in this process feature vectors are transformed into numeric vectors as the dataset contain various string like 0x<100 conversion. SVM can work only with numeric inputs so it is necessary to scale these features.Cross validation is also referred as model validation technique which is used for assessing how the result of the statistical analysis that will generalize to an independent dataset. This is mainly used for prediction and to estimate how accurately a predictive model will perform. In prediction process, a model is given which is usually a set of known data where the model is tested against the testing dataset. The goal of cross validation is to define a dataset to test the model in training dataset. Feature data(input) is fed into SVM model for training and testing, after training SVM model is used for predicting data. Matrix of training data corresponds to an observation and each column corresponds to a feature or variable. For performance assessment, test dataset with much lower fraud rate (0.5%) than in training datasets are used with different levels of under sampling. This process provides an indication of performance that are expected when models are applied for fraud detection where fraudulent transaction are typically low.

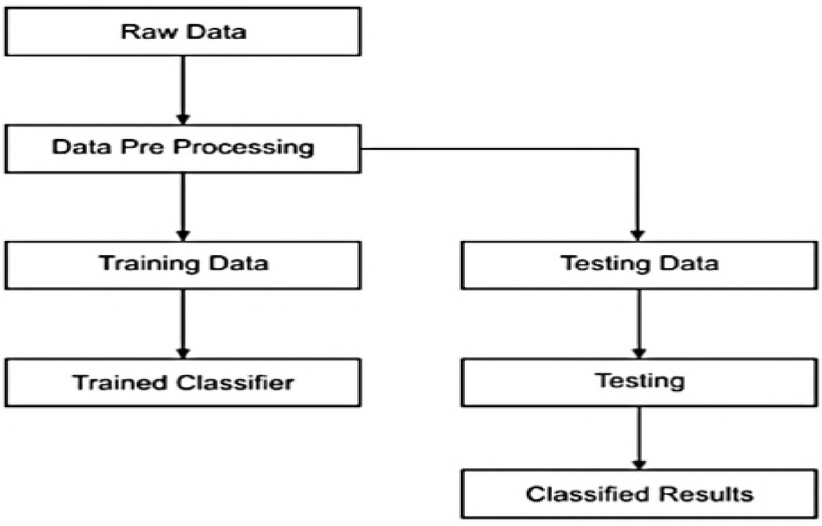


Figure 14: Flow Diagram of SVM process

The next approach is based on Clustering

4.2 Methodology Based on Clustering Approach

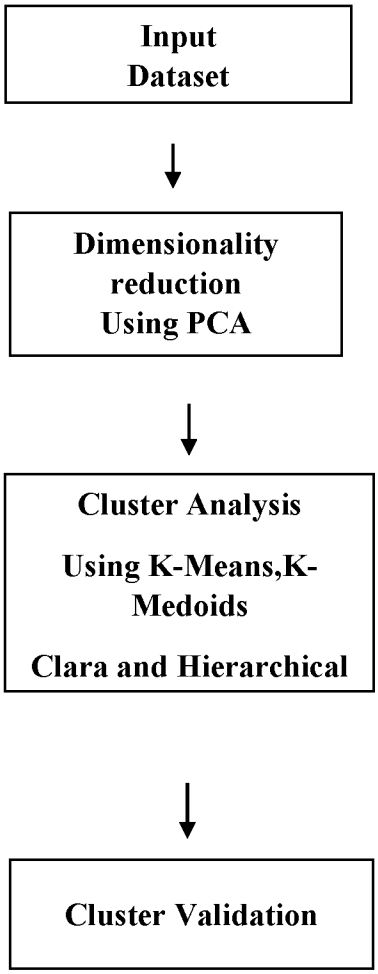


Figure 15: Methodology Second Based on Clustering Approach (2)

Credit Card fraud dataset is taken as input. By using principal Component Analysis (PCA) high dimensional data are often transformed into low dimensional data (jolliffe 2002). The main basis of PCA based dimension reduction is PCA picks up the dimensions with large variances. PCA is good at picking up linear relationships between features in the data. The first dimension, also called the first principal component (PC), reflects the majority of variation in our data. Next step is the cluster analysis. The term cluster analysis (first used by Tryon, 1939) encompasses a number of different algorithms and methods for grouping objects of similar kind into respective categories. In other words, cluster analysis is an exploratory data analysis tool which aims at sorting different objects into groups in a way that the degree of association between two objects is maximal if they belong to the same group and minimal otherwise. Cluster analysis can be used to discover

structures in data without providing an explanation/interpretation. Two clustering techniques are used in this proposed approach namely Partition Algorithms and Hierarchical clustering. Partition Algorithm are classified into three types namely K-means, k-Medoids and CLARA. Partition algorithm is breaking the dataset into groups based on similarity.The process of Cluster validation evaluates the goodness of cluster algorithm results. Using ClValid package internal measure validation is used in credit card fraud dataset. Based on the cluster quality the results are obtained for the two clustering algorithms, and optimal score is determined.

4.2.1 Clustering Techniques

The Clustering technique is used to place data elements into related groups without advance knowledge of the group description. The clustering technique groups the data instances in to subsets in such a manner that similar instances are grouped together while different instances belong to different groups. The objective of this research work is to study the performance of three clustering algorithms such as Hierarchical clustering, PAM clustering and K-Means clustering algorithm on Credit Card Fraud Dataset. The performance of the clustering algorithms dataset are evaluated in terms of internal validation measures.

K-means clustering

K-means clustering is a type of unsupervised learning, which can be used for unlabeled data (i.e., data which is not defined as categories or groups). The main goal of this algorithm is to find the number of groups in the data and the number of groups are represented by the variable K. The algorithm uses iteration method to assign each data point to one of the K group which is based on the features provided. Based on feature similarity data points are clustered. The result of k-means clustering is based on two process

- To label new data the centroids of the k clusters can be used.
- Based on labelling each data point is assigned to a single cluster.

  Clustering helps to find and analyse the groups that have formed organically.

K-means Clustering Algorithm

- Input: N examples $\{x_1, \ldots, x_N\}$ ($x_n \in R^D$); the number of partitions K
- Initialize: K cluster centers $\mu_1, \ldots, \mu_K$. Several initialization options:
- Randomly initialized anywhere in R^D
- Choose any K examples as the cluster centers

Iterate:

- Assign each of example x_n to its closest cluster center
- $C_k = \{n : k = \text{argmin} ||x_n - \mu_k||^2\}$
- k
- (C_k is the set of examples closest to μ_k)
- Recompute the new cluster centers μ_k (mean/centroid of the set C_k
- Repeat while not converged
- Possible convergence criteria: cluster centers do not change anymore

K-Medoids Clustering

The k-medoids algorithm is a partitioning clustering algorithm related to k-Means algorithm. Both the k-means and k-medoids algorithms are partitional (breaking the dataset into groups) and both attempt to minimize the distance between the points labelled to be in a cluster and a point designated as the centre of that cluster. K-Medoids chooses the data point as centres and works with a generalization of the Manhattan Norm to define distance between data points. This method clusters the dataset of n objects into K clusters. The silhouette is a useful tool for determining the number of clusters K. R has the cluster package to do K-medoids clustering where pam () and Pamk () are the functions used in k-medoids. K-means and K-medoids clustering produce almost same result. The only difference is that in K-means the cluster is represented by the cluster centre and in K-medoids the cluster is represented by the object close to the centre. But in the presence of outliers, k-medoids is more robust than k-means clustering. Partitioning Around Medoids (PAM) is a classic algorithm applied in K-medoids clustering Compared to the k-means approach, the function pam has the following features:

(a) it also accepts a dissimilarity matrix;

(b) it is more robust because it minimizes a sum of dissimilarities instead of a sum of squared Euclidean distances;

(c) it provides a novel graphical display, the silhouette plot which also allows to select the number of clusters

K-Medoids Algorithm

The pam-algorithm is based on the search for k representative objects or medoids among the observations of the dataset. These observations should represent the structure of the data. After finding a set of k medoids, k clusters are constructed by assigning each observation to the nearest medoid. The goal is to find k representative objects which minimize the sum of the dissimilarities of the observations to their closest representative object. The algorithm first looks for a good initial set of medoids (this is called the BUILD phase). Then it finds a local minimum for the objective function, that is, a solution such that there is no single switch of an observation with a medoid that will decrease the objective (this is called the SWAP phase).

Build phase:

- Choose k entities to become the medoids, or in case these entities were provided use them as the medoids;
- Calculate the dissimilarity matrix if it was not informed;
- Assign every entity to its closest medoid;

Swap phase:

- For each cluster search if any of the entities of the cluster lower the average dissimilarity coefficient, if it does select the entity that lowers this coefficient the most as the medoid for this cluster;
- If at least one medoid has changed go to (3), else end the algorithm.

Clara Clustering

CLARA (Clustering Large Applications, (Kaufman and Rousseeuw 1990)) is an extension to k-medoids methods to deal with data containing a large number of objects (more than several thousand observations) in order to reduce computing time and RAM

storage problem. This is achieved using the sampling approach. Instead of finding medoids for the entire data set, CLARA considers a small sample of the data with fixed size (*sample size*) and applies the PAM algorithm to generate an optimal set of medoids for the sample. The quality of resulting medoids is measured by the average dissimilarity between every object in the entire data set and the medoid of its cluster, defined as the cost function. CLARA repeats the sampling and clustering processes a pre-specified number of times in order to minimize the sampling bias. The final clustering results correspond to the set of medoids with the minimal cost

Clara Algorithm

- Split randomly the data sets in multiple subsets with fixed size.
- Compute PAM algorithm on each subset and choose the corresponding k representative objects (medoids). Assign each observation of the entire data set to the closest medoid.
- Calculate the mean (or the sum) of the dissimilarities of the observations to their closest medoid. This is used as a measure of the goodness of the clustering.
- Retain the sub-dataset for which the mean (or sum) is minimal. A further analysis is carried out on the final partition.

Hierarchical Clustering

Hierarchical clustering is a method of cluster analysis which tends to build hierarchy of clusters. Partition can be visualized as a tree structure in dendrogram format. In hierarchical Clustering it is possible to view different levels of granularities. The hierarchical clustering can be done using the function hclust () in the fpc package. Each observation is initially placed in its own cluster, and the clusters are successively joined together in order of their "closeness".

Hierarchical clustering is of two types Agglomerative and divisive. Agglomerative Hierarchical clustering is a bottom up approach. Agglomerative is also known as Agnes (Agglomerative nesting). In this each observation starts with its own cluster, and pairs of cluster are merged into one cluster and this moves up the hierarchy. The closeness of any two clusters is determined by a dissimilarity matrix, and can be based on a variety of agglomeration methods. Divisive clustering is a top down approach. To cover each cluster

rect. hclust () function is used. Divisive clustering is also known as DIANA (Divisive Analysis). In order to decide which clusters should be combined (for agglomerative), or where a cluster should be split (for divisive), a measure of dissimilarity between sets of observations is required. In most methods of hierarchical clustering, cluster similarity is achieved by use of an appropriate metric (a measure of distance between pairs of observations), and a linkage criterion which specifies the dissimilarity of sets as a function of the pairwise distances of observations in the sets.

Hierarchical Clustering Algorithm

Given a set of N items to be clustered, and an N*N distance (or similarity) matrix, the basic process of hierarchical clustering (defined by S.C. Johnson in 1967) is the following:

- Start by assigning each item to a cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters the same as the distances (similarities) between the items they contain.
- Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less.
- Compute distances (similarities) between the new cluster and each of the old clusters.
- Repeat steps 2 and 3 until all items are clustered into a single cluster of size N. (*)

4.3 Cluster validation

The term **cluster validation** is used to design the procedure of evaluating the goodness of clustering algorithm results. This is important to avoid finding patterns in a random data, as well as, in the situation where two clustering algorithms have to be compared. Cluster validation is concerned with the quality of clusters generated by an algorithm for clustering of data.

Clustering validation includes three main tasks:

- Cluster tendency
- Cluster evaluation
- Cluster stability

Clustering tendency assesses whether the data can be clusterable, that is, whether the data contains any inherent grouping structure. This should be checked before applying clustering analysis

Clustering evaluation assesses the goodness or quality of the clustering.

Clustering stability seeks to understand the sensitivity of the clustering result to various algorithmic parameters, for example, the number of clusters.

The aim of this part is to:

- describe the different methods for clustering validation
- compare the quality of clustering results obtained with different clustering algorithms
- provide R package for validating clustering results

Clustering validation measures in clValid package

- Internal validation measures
- Stability validation measures

CLVALID is technique for evaluating unsupervised clustering techniques. CLVALID uses the R package "clValid" to compare the relative properties of two different clustering methods across a several different numbers of clusters. This module aims to help choose a method that is most compact, well-separated, connected, and stable. It also makes use of bio conductor annotation packages to biologically validate the results. In this propose work only internal measures have been used to check the quality of the clustering.

4.3.1 Internal Validation: This validation evaluates the quality of the clustering based solely on the dataset and the clustering partition. This assessment is demonstrated by the measures Connectivity, Silhouette Width and Dunn Index, which were chosen to elucidate the compactness, connectedness and separation of the cluster partitions. It uses intrinsic information in the data to assess the quality of the clustering. The **internal measures** included in **clValid** package are:

➢ **Connectivity**

This measure reflects the extent to which items that are placed in the same cluster are also considered their nearest neighbors in the data space - or, in other words, the degree of connectedness of the clusters.

$$\text{Conn(C)} = \sum_{i=1}^{N} \sum_{j=1}^{L} \text{Xi, nni(j)}$$

where L is a parameter that determines the number of neighbors that contribute to the connectivity measure. The connectivity has a value between zero and 1 and should be minimized.

➢ **Average Silhouette width**

This index defines compactness based on the pairwise distances between all elements in the cluster, and separation based on pairwise distances between all points in the cluster and all points in the closest other cluster. Silhouette function is used to asses the optimal number of clusters and the values as close to (+) 1 as possible are most desirable

$$S(i) = \frac{\text{bi} - \text{ai}}{\max(\text{bi, ai})}$$

Where,

$$\text{ai} = \frac{1}{\text{n(C(i))}} \sum_{j \in C(i)} \text{dist(i,j)}$$

$$\text{bi} = \min \text{ck} \in C \backslash C(i) \sum_{j \in \text{Ck}} \frac{\text{dist(i,j)}}{\text{n(Ck)}}$$

where a_i is the average distance between i and all other observations in the same cluster, and b_i is the average distance between i and the observations in the nearest neighboring cluster",

$$a_i = \frac{1}{n(C(i))} \sum_{j \in C(i)} dist(i,j), \quad b_i = \min_{C_k \in \mathcal{C} \backslash C(i)} \sum_{j \in C_k} \frac{dist(i,j)}{n(C_k)},$$

where C(i) is the cluster containing observation i, dist(i; j) is the distance (e.g. Euclidean, Manhattan) between observations i and j, and n(C) is the cardinality of cluster C. The silhouette width thus lies in the interval [1; 1], and should be maximized.

- **Dunn index**

Dunn Index represents the ratio of the smallest distance between observations not in the same cluster to the largest intra-cluster distance. As you can imagine, the nominator should be maximized, therefore the index should be maximized.

$$D(\mathcal{C}) = \frac{\min_{C_k, C_l \in \mathcal{C}, C_k \neq C_l} \left(\min_{i \in C_k, j \in C_l} dist(i,j) \right)}{\max_{C_m \in \mathcal{C}} diam(C_m)}$$

where diam(C_m) is the maximum distance between observations in cluster C_m. The Dunn index has a value between zero and 1, and should be maximized.

4.3.2 Stability Measures

Stability Validation: This validation uses an iterative approach of removing one column (or row) from the dataset and comparing the results. It is a special version of internal validation. It evaluates the consistency of a clustering result by comparing it with the clusters obtained after each column is removed, one at a time.

The **cluster stability measures** include:

- The **average proportion of non-overlap** (APN)

The APN measures the average proportion of observations not placed in the same cluster but by clustering based on the full data and clustering based on the data with a single column removed.

$$\text{APN(C)} = \frac{1}{\text{MN}} \sum_{i=1}^{N} \sum_{l=1}^{M} \left(1 - \frac{\text{n}(C^{i,l} \cap C^{i,0})}{\text{n}(C^{i,0})} \right)$$

Let $C^{i;0}$ represent the cluster containing observation i using the original clustering (based on all available data), and $C^{i;}$ represent the cluster containing observation i where the clustering is based on the dataset with column removed. Then, the APN measure is defined as The APN is in the interval [0; 1], with values close to zero corresponding with highly consistent clustering results.

- The **average distance** (AD)

The AD measures the average distance between observations placed in the same cluster under both cases (full dataset and removal of one column).

$$\mathrm{AD(C)} = \frac{1}{\mathrm{MN}} \sum_{i=1}^{N} \sum_{l=1}^{M} \frac{1}{\mathrm{n}(C^{i,0}) n(C^{i,l})}$$

- The **average distance between means** (ADM)

The ADM measures the average distance between cluster centers for observations placed in the same cluster under both cases.

$$ADM(C) = \frac{1}{\mathrm{MN}} \sum_{i=1}^{N} \sum_{l=1}^{M} dist(x'c^{i,l}, x'c^{i,0})$$

where $x_{Ci;0}$ is the mean of the observations in the cluster which contain observation i, when clustering is based on the full data, and $x_{Ci;`}$ is similarly de ned. Currently, ADM only uses the Euclidean distance. It also has a value between zero and 1, and again smaller values are prefered.

- The **Figure Of Merit** (FOM)

The FOM measures the average intra-cluster variance of the deleted column, where the clustering is based on the remaining (undeleted) columns. It also has a value between zero and 1, and again smaller values are preferred.

$$\mathrm{FOM(l, C)} = \sqrt{\frac{1}{\mathrm{N}} \sum_{k=1}^{K} \sum_{i \in \mathrm{Ck(l)}} dist(\mathrm{xi, l, x'k(l)})}$$

where $x_{i;`}$ is the value of the ith observation in the `th column, and $x_{Ck(`)}$ is the average of cluster $C_k(`)$. Currently, the only distance available for FOM is Euclidean. The FOM is multiplied by an adjustment factor n to alleviate the tendency to decrease as the number of clusters increases. The nal score is averaged over all the removed columns, and has a value between zero and 1, with smaller values equaling better performance.

Conclusion

This section briefly discussed about the proposed methodology of this project and results of the proposed work are explained in next chapter in detailed manner.

Chapter VI

EXPERIMENTAL RESULTS AND DISCUSSION

6.1 Experimental Results and Discussion

This Chapter discusses about the experimental results of the proposed approach.

The experiments are performed in R package. The R Language is a project developed at Bell Laboratories and available for free under the GNU license. The R language programming environment is built around a standard command-line interface. The R programming language is an open source scripting language for predictive analytics and data visualization. UCI machine repository dataset is taken for experiment which is represented in Annexure A.

The results are discussed below

6.1.1 Results

Two classifiers namely KNN, SVM are compared based on three performance metrics i.e., Accuracy, Sensitivity and Specificity. Table 3: represents Performance Comparison of Classifiers.

Accuracy

Accuracy is perhaps the most intuitive performance measure. It is simply the ratio of correctly predicted observations. Classification accuracy is the ratio of correct predictions to total predictions made.

classification accuracy = correct predictions / total predictions

Sensitivity

Sensitivity is also referred as true positive rate. It's the ratio of correctly predicted positive events.

$$Sensitivity = (Number\ of\ True\ Positives) \div (Number\ of\ False\ Negatives)$$

Specificity

Specificity is also called the true negative rate measures the proportion of negatives that are correctly identified.

$$Specificity = (Number\ of\ True\ Negatives) \div (Number\ of\ False\ Positives)$$

Table 3: Performance Comparison of Classifiers

Performance Metrics	Classifiers	
	k-Nearest Neighbor	Support Vector Machine
Accuracy	0.9937	0.9874
Sensitivity	1.0000	1.0000
Specificity	0.7143	0.4286

Table 4: Clustering Internal Measures

Clustering Method	Measure	Cluster Sizes				
		2	3	4	5	6
Hierarchical	**Connectivity**	2.9290	5.8579	8.7869	11.28	13.5865
	Dunn	0.5267	0.6674	0.6005	0.3619	0.3789
	Silhouette	0.7991	0.7693	0.6890	0.6458	0.6041
k-means	**Connectivity**	6.8742	9.8032	69.47	72.40	156.17
	Dunn	0.2256	0.2441	0.0673	0.0924	0.0773
	Silhouette	0.6506	0.6516	0.2868	0.2871	0.2935
Pam	**Connectivity**	504.67	504.55	594.68	732.21	677.37
	Dunn	0.0124	0.0165	0.0229	0.0177	0.0177
	Silhouette	0.0473	0.0599	0.0615	0.0396	0.0510
Clara	**Connectivity**	512.73	511.30	587.20	510.98	634.82
	Dunn	0.0188	0.0226	0.0118	0.0191	0.0213
	Silhouette	0.0520	0.0606	0.0442	0.064	0.0587

Table 4 represents the output of clustering internal measures of Hierarchical, K-Means, Pam, Clara Method based on their clustering sizes.

Table 5: Optimal Scores of Clustering in Internal Validation

Internal Validation	Score	Method	Clusters
Connectivity	2.9290	Hierarchical	2
Dunn	0.6674	Hierarchical	3
silhouette	0.7991	Hierarchical	2

Table 5 represents the output of optimal Scores of Clusters. Here, Hierarchical clustering with two clusters performs the best in each case.(Connectivity & Silhouette) The validation measures can also be displayed graphically using the plot() method. Plots for individual measures can be requested using the measures argument. A legend is also included with each plot. The default location of the legend is the top right corner of each plot, this can be changed using the legend Loc argument. Here, all three plots into a single gure are combined and so suppress the legends in each individual plot. Instead of viewing all the validation measures via the summary () method, optimal values can be viewed by using the optimal Scores () method. The optimal Score output is represented in graph plot in Annexures M-P.

Table 6: Stability Measures of Clustering

Clustering Method	Measure	Cluster Sizes				
		2	3	4	5	6
Hierarchical	**APN**	0.001	0.001	0.010	0.010	0.023
	AD	6.477	6.427	6.334	6.298	6.204
	ADM	0.010	0.020	0.100	0.096	0.393
	FOM	0.888	0.876	0.874	0.863	0.855
k-means	**APN**	0.001	0.001	0.010	0.010	0.023
	AD	6.477	6.427	6.334	6.298	6.204
	ADM	0.010	0.020	0.100	0.096	0.393
	FOM	0.888	0.876	0.874	0.863	0.855
Pam	**APN**	0.226	0.210	0.201	0.218	0.172

	AD	6.811	6.374	6.233	6.162	5.954
	ADM	0.653	0.578	0.575	0.688	0.490
	FOM	0.996	0.882	0.877	0.875	0.864
Clara	**APN**	0.162	0.189	0.176	0.398	0.208
	AD	6.802	6.372	6.231	6.331	6.123
	ADM	0.524	0.508	0.591	1.156	0.692
	FOM	0.997	0.884	0.878	0.875	0.872

Table 6 Represents the output of stability measures of four clustering methods based on four measures namely, APN, AD, ADM and FOM .

Table 7: Optimal Scores of Stability Measures

Intern Validation	**Score**	**Method**	**Clusters**
APN	0.0000	Hierarchical	2
AD	5.9540	Pam	6
ADM	0.0000	Hierarchical	2
FOM	0.82525	Hierarchical	6

Table 7 represents the optimal scores of Stability Measures. Here, Hierarchical Clustering with two clusters and six cluster performs well in three cases (APN, ADM, FOM). The Validation measures can also be graphically displayed using the plot () method which is represented in Annexure Q-T

CHAPTER VI

CONCLUSION AND FUTURE ENHANCEMENT

Conclusion

The performance of classification and clustering algorithms on credit card fraud dataset is taken up for experimentation. Two Classification algorithms namely K-Nearest Neighbor (KNN) and Support Vector Machine(SVM) are used compare the performance of two classifiers using three performance metrics namely., Accuracy, Sensitivity and Specificity. Four clustering algorithms such as k-means clustering algorithm, PAM clustering algorithm, Clara algorithm, and hierarchical clustering algorithm are applied on the credit card dataset. The performance is analyzed based on the internal validation measures and Stability measures. The results show that hierarchical clustering is the best algorithm compared to other two algorithms in terms of validation measures and in stability measures PAM and Hierarchical clustering

Future Enhancement

The research work can be done using a hybrid model by using supervised learning technique. As it provides optimal results than single based model. In terms of Clustering There are large numbers of dataset present and many other cluster analysis are present. So, the future work will be based on other clustering algorithms that can be applied on the data set and the best techniques can be identified based on cluster quality.

REFERENCES

Journal References

- Dheepa, V, Dhanapal, R, "Behavior Based Credit Card Fraud Detection Using Support Vector Machines", ICTACT Journal on Soft Computing, Volume 2, Issue 4, July 2012.
- John O. Awoyemi, Adebayo O. Adetunmbi, Samuel A. Oluwadare. "Credit Card Fraud Detection Using Machine Learmimg Techniques", Institute of Electrical and Electronics Engineers, 2017.
- Sai Nidhi,E , Snigdha,D "Credit Card Fraud Detection Analysis Using R", International Journal of Innovations & Advancement in Computer Science, Volume 6, October 2017.
- Suganya, R. & Shanthi, R. Fuzzy C-Means Algorithm, A Review. International Journal of Scientific and Research Publications, Volume 2, Issue 11, November 2012.
- Vaishali, "Fraud Detection in Credit Card by Clustering Approach", International Journal of Computer Applications, Volume 98, July 2014.
- Yokesh Narekar,M & Sushil Kumar Chavan, " A Review on Credit Card Fraud Detection Using BLAST- SSAHA Method", International Journal of Advance Research in Computer and Communication Engineering, Volume 4, Issue 11, November 2015.

Web References

- https://en.wikipedia.org/wiki/Linear_classifier
- https://www.techopedia.com/definition/32066/k-nearest-neighbor-k-nn
- https://machinelearningmastery.com/k-nearest-neighbors-for-machine-learning/
- https://eight2late.wordpress.com/2015/11/06/a-gentle-introduction-to-naive-bayes-classification-using-r/
- https://www.csie.ntu.edu.tw/~cjlin/papers/kernel-check/kcheck.pdf
- http://www.statsoft.com/Textbook/Support-Vector-Machines
- http://r-statistics.co/Linear-Regression.html
- https://businesstech.co.za/news/banking/172249/the-biggest-types-of-credit-card-fraud-in-south-africa/
- http://www.fraudpractice.com/fl-fraudhist.html
- http://www.india.com/news/india/27482-cases-of-cybercrimes-reported-in-2017-one-attack-in-india-every-10-minutes-2341055/

Codings of Clustering Algorithm

```
head(dataset)

str(dataset)

summary(dataset)

sus_dataset<-scale(dataset)

head(sus_dataset)

dist_data<-dist(sus_dataset, method = 'euclidean')

hdata<-hclust(dist_data)

plot(hdata)

abline(h=3.75, lty=2)

plot(hdata)

rect.hclust(hdata,k=25,border="blue")

set.seed(123)

kus_dataset<-kmeans(sus_dataset, centers = 5, nstart = 50)

plot(x=sus_dataset[,1], y=sus_dataset[,2], col=kus_dataset$cluster)

points(kus_dataset$centers, pch=3, cex=2)

library(cluster)

clusplot(sus_dataset, kus_dataset$cluster, color = T, labels = 5, main = 'Cluster Plot')

plot(silhouette(kus_dataset$cluster, dist = dist_data), col=2:5)

PAMus_dataset<-pam(sus_dataset, 4)

clusplot(sus_dataset, PAMus_dataset$clustering, color = T, main = 'Cluster Plot')

 kmax<-10
```

```
WSSus_dataset<-sapply(1:kmax, function(k) kmeans(sus_dataset, centers = k, nstart = 10)$tot.withinss)

plot(1:kmax, WSSus_dataset, type = 'b', xlab = 'k', ylab = 'Total wss')

abline(v=4, lty=2)

data <-dataset

pc.cr <- princomp(dataset, cor=TRUE)

pc.cr

summary(pc.cr)

loadings(pc.cr)

plot(pc.cr)

biplot(pc.cr)

plot(dataset, pch="+",main="Scatterplotmatrix")

fit <- kmeans(dataset, 6)

library(cluster)

clusplot(dataset, fit$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)

library(factoextra)

fviz_cluster(pam.res, frame.type = "t",frame.alpha = 0, frame.level = 0.7)

fviz_cluster(pam.res, geom = "point")

fviz_cluster(pam.res) + scale_color_brewer(palette = "Set2")+ scale_fill_brewer(palette = "Set2")
+ theme_minimal()

res.dist <- get_dist(dataset, stand = TRUE, method = "pearson")

fviz_dist(res.dist, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))

res <- hcut(dataset, k = 4, stand = TRUE)
```

```
fviz_dend(res, rect = TRUE, cex = 0.5,k_colors = c("#00AFBB","#2E9FDF", "#E7B800", "#FC4E07"))


my_data <- scale(dataset[, -5])
get_clust_tendency(my_data, n = 50,gradient = list(low = "steelblue",  high = "white"))
my_data <- scale(dataset)
library("factoextra")
fviz_nbclust(my_data, kmeans, method = "gap_stat")
factoextra::fviz_nbclust(res.nbclust) + theme_minimal()
my_data <- scale(dataset[, -5])
library("factoextra")
res.hc <- eclust(my_data, "hclust", k = 3, graph = FALSE)
fviz_dend(res.hc, rect = TRUE, show_labels = FALSE)
fviz_silhouette(res.hc)
sil <- res.hc$silinfo$widths[, 1:3]
neg_sil_index <- which(sil[, 'sil_width'] < 0)
sil[neg_sil_index, , drop = FALSE]
my_data <- scale(dataset)
fviz_nbclust(df, pam, method = "wss") +
geom_vline(xintercept = 3, linetype = 2)
pam.res <- pam(df, 3)
print(pam.res)
dd<- cbind(dataset, cluster = pam.res$cluster)
```

```
head(dd, n = 3)
pam.res$medoids


head(pam.res$clustering)
fviz_cluster(pam.res,
             palette = T,
             main = 'PAM Cluster Plot')
require(cluster)
fviz_nbclust(dd, pam, method = "silhouette")+
labs(subtitle = "PAM Silhouette method")
library(cluster)
library(factoextra)
fviz_nbclust(df, clara, method = "wss")+
geom_vline(xintercept = 3, linetype = 2)
clara.res <- clara(df, 3, samples = 50, pamLike = TRUE)
print(clara.res)
dd<- cbind(df, cluster = clara.res$cluster)
head(dd, n = 4)
clara.res$medoids
head(clara.res$clustering, 10)
fviz_cluster(clara.res,
             palette = T,
             main = 'CLARA cluster plot')
```

```
require(cluster)
fviz_nbclust(dd, clara, method = "silhouette")+
labs(subtitle = "CLARA Silhouette method")
library("clValid")
intern <- clValid(my_data, nClust = 2:6,  clMethods = c("hierarchical","kmeans","pam",”clara”), validation = "internal")
summary(intern)
optimalScores(intern)
plot(intern)
 op <- par(no.readonly=TRUE)
 par(mfrow=c(2,2),mar=c(4,4,3,1))
plot(intern, legend=FALSE)
plot(nClusters(intern),measures(intern,"Dunn")[,,1],type="n",axes=F,
       + xlab="",ylab="")
legend("center", clusterMethods(intern), col=1:9, lty=1:9, pch=paste(1:9))
par(op)
stab<-clValid
(my_data,2:6,clMethods=c("hierarchical","kmeans","pam",”clara”),validation="stability")
summary(stab)
optimalScores(stab)
plot(stab)
op <- par(no.readonly = TRUE)
par(mfrow = c(2, 2), mar = c(4, 4, 3, 1))
```

```
plot(intern, legend = FALSE)

plot(nClusters(intern), measures(intern, "Dunn")[, , 1], type = "n",axes = F, xlab = "", ylab = "")

legend("center", clusterMethods(intern), col = 1:9, lty = 1:9,pch = paste(1:9))

par(op)
```

Codings of Classification Algorithm

```
options(warn=1) library(FactoMineR)

 library(factoextra)

 library(caret)

library(e1071)

dataset <- read.csv('/home/maghes/PROJECTs/R/fraud/changes.csv', header = TRUE, sep = ',')
dim(dataset)

summary(dataset) preprocessParams <- preProcess(dataset[,1:31], method=c("scale")) pcaData
<- PCA(dataset, scale.unit = TRUE, ncp = 5, graph = TRUE) print(pcaData) fviz_eig(pcaData,
addlabels = TRUE, ylim = c(0, 50)) var <- get_pca_var(pcaData) var fviz_pca_var(pcaData,
col.var = "black") fviz_pca_var(pcaData, col.var = "cos2", gradient.cols = c("#00AFBB",
"#E7B800", "#FC4E07"),    repel = TRUE # Avoid text overlapping) intrain <-
createDataPartition(y = pcaData$svd$U, p= 0.7, list = FALSE) training <- dataset[intrain,]
testing <-dataset [-intrain,] training <-training[rowSums(is.na(training)) == 0,] testing <-
testing[rowSums(is.na(testing)) == 0,] dim(training); dim(testing);

anyNA(dataset) training[["Class"]] = factor(training[["Class"]]) trctrl <-
trainControl(method = "repeatedcv", number = 10, repeats = 3)
set.seed(3333) knn_fit <- train(Class ~., data = training, method = "knn",

                trControl=trctrl,
preProcess = c("center", "pca"),
tuneLength = 10)

 knn_fit test_knn <- predict(knn_fit, newdata = testing) test_knn test_knn[test_knn < 0.5] = 0;
test_knn[test_knn > 0.5] = 1; knnmatrix <- confusionMatrix(test_knn, testing$Class) knnmatrix
model_svm <- svm(Class ~. , training) trctrl <- trainControl(method = "repeatedcv", number =
10, repeats = 3) set.seed(3233) svm_Linear <- train(V31 ~., data = training, method =
"svmLinear",
```

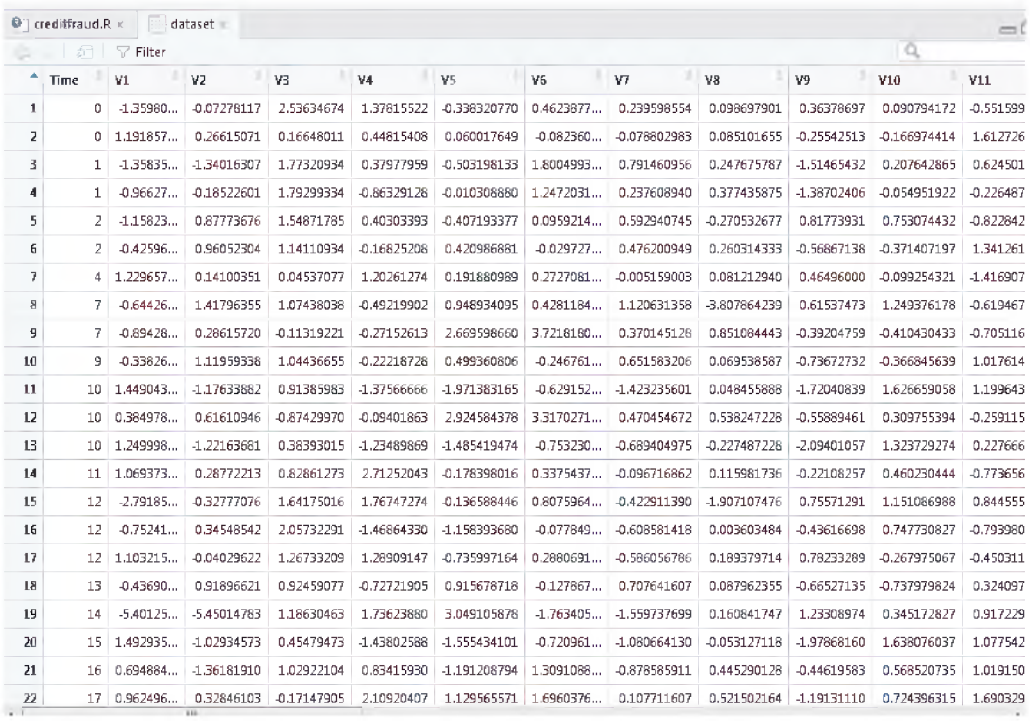
```
                trControl=trctrl,
preProcess = c("center", "scale"),
tuneLength = 10)


test_pred <- predict(model_svm,testing) test_pred test_pred[test_pred < 0.5] = 0;
test_pred[test_pred > 0.5] = 1; svmmatrix <- confusionMatrix(test_pred, testing$Class)
svmmatrix svmAccury  <- svmmatrix$overall knnAccurcy  <- knnmatrix$overall accurcy <-
c(svmAccurcy[1],knnAccurcy[1],svmAccurcy[2],knnAccurcy[2]) barplot(accurcy,
main="Accurcy Compare", horiz=FALSE,col=c("darkblue","red"), names.arg=c("KNN
Accurcy", "SVM Accurcy","KNN Kappa", "SVM Kappa")) svmClass  <- svmmatrix$byClass
knnClass  <- knnmatrix$byClass
knnCompare                                                                    <-

c(knnClass[1],knnClass[2],knnClass[3],knnClass[4],knnClass[5],knnClass[6],knnClass[7],kn
nClass[8],knnClass[9],knnClass[10],knnClass[11]) svmCompare <- c(svmClass [1],svmClass
[2],svmClass [3],svmClass [4],svmClass

[5],svmClass [6],svmClass [7],svmClass [8],svmClass [9],svmClass [10],svmClass [11])


barplot(knnCompare, main="Accurcy Compare", horiz=FALSE,
names.arg=c("Sensitivity","Specificity","Pos Pred Value","Neg Pred

Value","Recall","F1","Prevalence",   "Detection      Rate","Detection
Prevalence","Balanced Accuracy"))

knnCompare                                                                    <-

c(knnClass[1],knnClass[2],knnClass[3],knnClass[4],knnClass[5],knnClass[6],knnClass[7],kn
nClass[8],knnClass[9],knnClass[10],knnClass[11]) svmCompare <- c(svmClass [1],svmClass
[2],svmClass [3],svmClass [4],svmClass

[5],svmClass [6],svmClass [7],svmClass [8],svmClass [9],svmClass [10],svmClass[11])  reshape
data into long format library(reshape2) library(ggplot2)

barplot(matrix(knnCompare,svmCompare), beside=T,  col=c("aquamarine3","coral"),
```

```
names.arg=c("Sensitivity","Specificity","Pos Pred Value","Neg Pred
Value","Recall","F1","Prevalence", "Detection Rate","Detection Prevalence","Balanced
Accuracy"))
```

Annexure

SCREENSHOTS

Annexure A: Dataset



creditfraud.R × | dataset ×

Filter

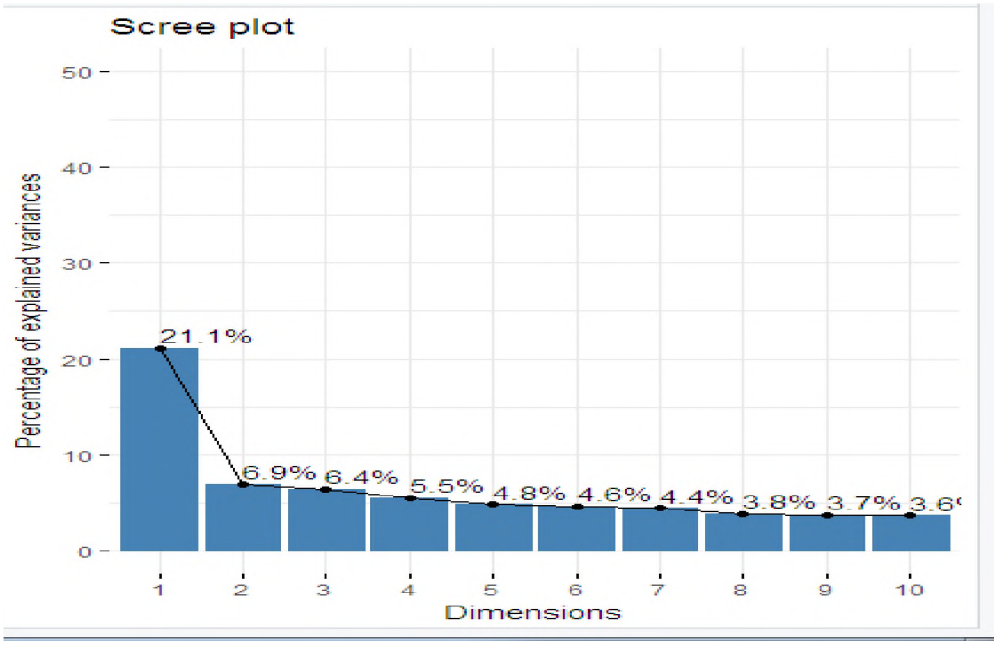
	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11
1	0	-1.35980...	-0.07278117	2.53634674	1.37815522	-0.338320770	0.4623877...	0.239598554	0.098697901	0.36378697	0.090794172	-0.551599
2	0	1.191857...	0.26615071	0.16648011	0.44815408	0.060017649	-0.082360...	-0.078802983	0.085101655	-0.25542513	-0.166974414	1.612726
3	1	-1.35835...	-1.34016307	1.77320934	0.37977959	-0.503198133	1.8004993...	0.791460956	0.247675787	-1.51465432	0.207642865	0.624501
4	1	-0.96627...	-0.18522601	1.79299334	-0.86329128	-0.010308880	1.2472031...	0.237608940	0.377435875	-1.38702406	-0.054951922	-0.226487
5	2	-1.15823...	0.87773676	1.54871785	0.40303393	-0.407193377	0.0959214...	0.592940745	-0.270532677	0.81773931	0.753074432	-0.822842
6	2	-0.42596...	0.96052304	1.14110934	-0.16825208	0.420986881	-0.029727...	0.476200949	0.260314333	-0.56867138	-0.371407197	1.341261
7	4	1.229657...	0.14100351	0.04537077	1.20261274	0.191880989	0.2727081...	-0.005159003	0.081212940	0.46496000	-0.099254321	-1.416907
8	7	-0.64426...	1.41796355	1.07438038	-0.49219902	0.948934095	0.4281184...	1.120631358	-3.807864239	0.61537473	1.249376178	-0.619467
9	7	-0.89428...	0.28615720	-0.11319221	-0.27152613	2.669598660	3.7218180...	0.370145128	0.851084443	-0.39204759	-0.410430433	-0.705116
10	9	-0.33826...	1.11959338	1.04436655	-0.22218726	0.499360806	-0.246761...	0.651583206	0.069539587	-0.73672732	-0.366845639	1.017614
11	10	1.449043...	-1.17633882	0.91385983	-1.37566666	-1.971383165	-0.629152...	-1.423235601	0.048455888	-1.72040839	1.626659058	1.199643
12	10	0.384978...	0.61610946	-0.87429970	-0.09401863	2.924584378	3.3170271...	0.470454672	0.538247228	-0.55889461	0.309755394	-0.259115
13	10	1.249998...	-1.22163681	0.38393015	-1.23489869	-1.485419474	-0.753230...	-0.689404975	-0.227487226	-2.09401057	1.323729274	0.227666
14	11	1.069373...	0.28772213	0.82861273	2.71252043	-0.178398016	0.3375437...	-0.096716862	0.115981736	-0.22108257	0.460230444	-0.773656
15	12	-2.79185...	-0.32777076	1.64175016	1.76747274	-0.136588446	0.8075964...	-0.422911390	-1.907107476	0.75571291	1.151086988	0.844555
16	12	-0.75241...	0.34548542	2.05732291	-1.46864330	-1.158393680	-0.077849...	-0.608581418	0.003603484	-0.43616698	0.747730827	-0.793980
17	12	1.103215...	-0.04029622	1.26733209	1.28909147	-0.735997164	0.2880691...	-0.586056786	0.189379714	0.78233289	-0.267975067	-0.450311
18	13	-0.43690...	0.91896621	0.92459077	-0.72721905	0.915678718	-0.127867...	0.707641607	0.087962355	-0.66527135	-0.737979824	0.324097
19	14	-5.40125...	-5.45014783	1.18630463	1.73623880	3.049105878	-1.763405...	-1.559737699	0.160841747	1.23308974	0.345172827	0.917229
20	15	1.492935...	-1.02934573	0.45479473	-1.43802588	-1.555434101	-0.720961...	-1.080664130	-0.053127118	-1.97868160	1.638076037	1.077542
21	16	0.694884...	-1.36181910	1.02922104	0.83415930	-1.191208794	1.3091088...	-0.878585911	0.445290128	-0.44619583	0.568520735	1.019150
22	17	0.962496...	0.32846103	-0.17147905	2.10920407	1.129565571	1.6960376...	0.107711607	0.521502164	-1.19131110	0.724396315	1.690329

creditfraud.R × dataset ×

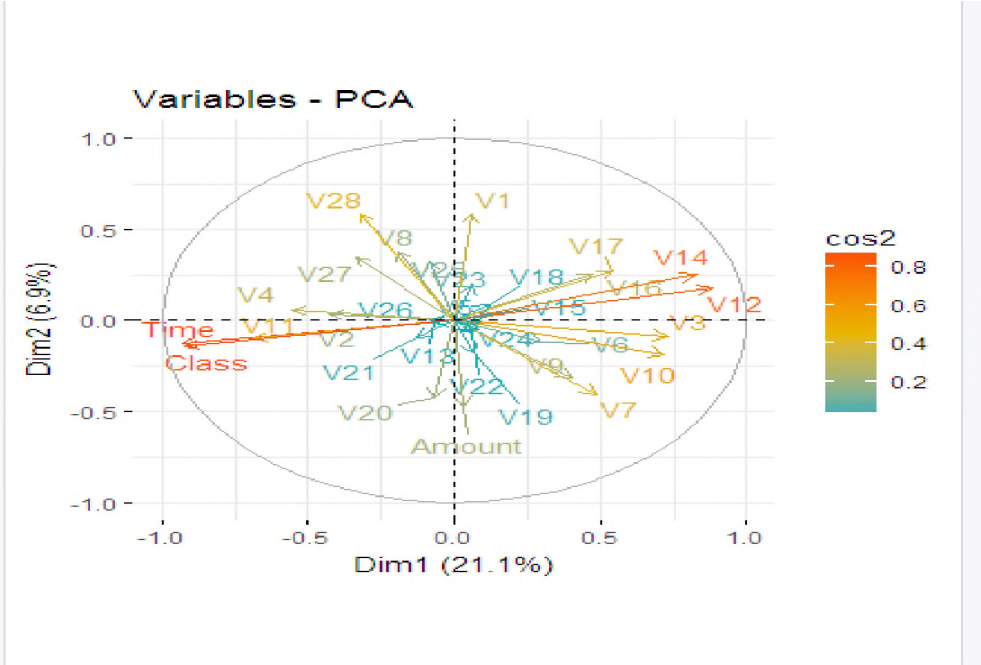
Filter

	V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
03992960	0.251412098	-0.018306778	0.277837576	-0.110473910	0.066928075	0.128539358	-0.189114...	0.133558377	-0.021053053	149.62	0
45783041	-0.069083135	-0.225775248	-0.638671953	0.101288021	-0.339846476	0.167170404	0.1258945...	-0.008983099	0.014724169	2.69	0
61857095	0.524979725	0.247998153	0.771679402	0.909412262	-0.689280956	-0.327641834	-0.139096...	-0.055352794	-0.059751841	378.66	0
32621970	-0.208037781	-0.108300452	0.005273597	-0.190320519	-1.175575332	0.647376035	-0.221928...	0.062722849	0.061457629	123.50	0
03486925	0.408542360	-0.009430697	0.798278495	-0.137458080	0.141266984	-0.206009588	0.5022922...	0.219422230	0.215153147	69.99	0
33193788	0.084968672	-0.208253515	-0.559824796	-0.026397668	-0.371426583	-0.232793817	0.1059147...	0.253844225	0.081080257	3.67	0
45575045	-0.219632553	-0.167716266	-0.270709726	-0.154103787	-0.780055415	0.750136936	-0.257236...	0.034507430	0.005167769	4.99	0
24504731	-0.156741852	1.943465340	-1.015454710	0.057503530	-0.649709006	-0.415266566	-0.051634...	-1.206921081	-1.085339188	40.80	0
70328167	0.052735669	-0.073425100	-0.268091632	-0.204232670	1.011591802	0.373204680	-0.384157...	0.011747356	0.142404330	93.20	0
51772964	0.203711455	-0.246913937	-0.633752642	-0.120794084	-0.385049925	-0.069733046	0.0941988...	0.246219305	0.083075649	3.68	0
21365414	-0.387226474	-0.009301897	0.313894411	0.027740158	0.500512287	0.251367359	-0.129477...	0.042849871	0.016253262	7.80	0
07663826	0.125991576	0.049923686	0.238421512	0.009129869	0.996710210	-0.767314827	-0.492208...	0.042472442	-0.054337388	9.99	0
83192626	-0.102755942	-0.231809239	-0.483285330	0.084667691	0.392830885	0.161134554	-0.354990...	0.026415549	0.042422089	121.50	0
82916082	-0.153197231	-0.036875532	0.074412403	-0.071407433	0.104743753	0.548264725	0.1040941...	0.021491058	0.021293311	27.50	0
21868014	-1.582122044	1.151663048	0.222181966	1.020586204	0.028316651	-0.232746324	-0.235557...	-0.164777512	-0.030153637	58.80	0
32535349	0.263450864	0.499624955	1.353650486	-0.256573280	-0.065083708	-0.039124354	-0.087086...	-0.180997500	0.129394059	15.99	0
75681622	-0.113910177	-0.024612006	0.196001953	0.013801654	0.103758331	0.364297541	-0.382260...	0.092809187	0.037050517	12.99	0
25436462	-0.047021282	-0.194795824	-0.672637997	-0.156857514	-0.888386321	-0.342413219	-0.049026...	0.079692399	0.131023789	0.89	0
06866573	-2.136848025	-0.503600329	0.384459786	2.458588576	0.042118897	-0.481630824	-0.621272...	0.392053290	0.949594246	46.80	0
54229515	-0.387910173	-0.177649846	-0.175073809	0.040002219	0.295813863	0.332930599	-0.220384...	0.022298436	0.007602256	5.00	0
00408169	-0.138333940	-0.295582932	-0.571955007	-0.050880701	-0.304214501	0.072001006	-0.422234...	0.086553398	0.063498649	231.71	0
27612322	-0.269320967	0.143997423	0.402491661	-0.048508221	-1.371866295	0.390813885	0.1999636...	0.016370643	-0.014605328	34.09	0

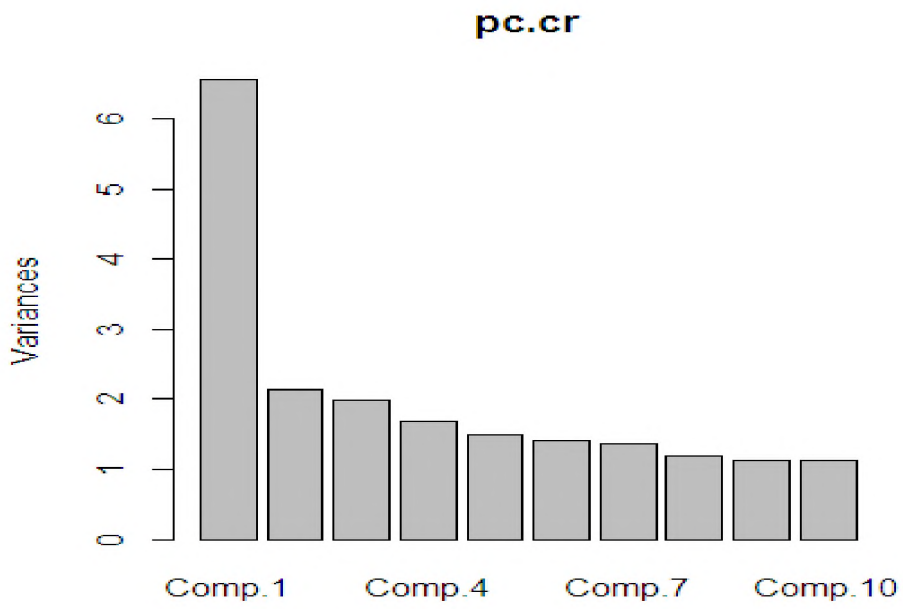
Annexure B: Principal Component Analysis(1) Plot



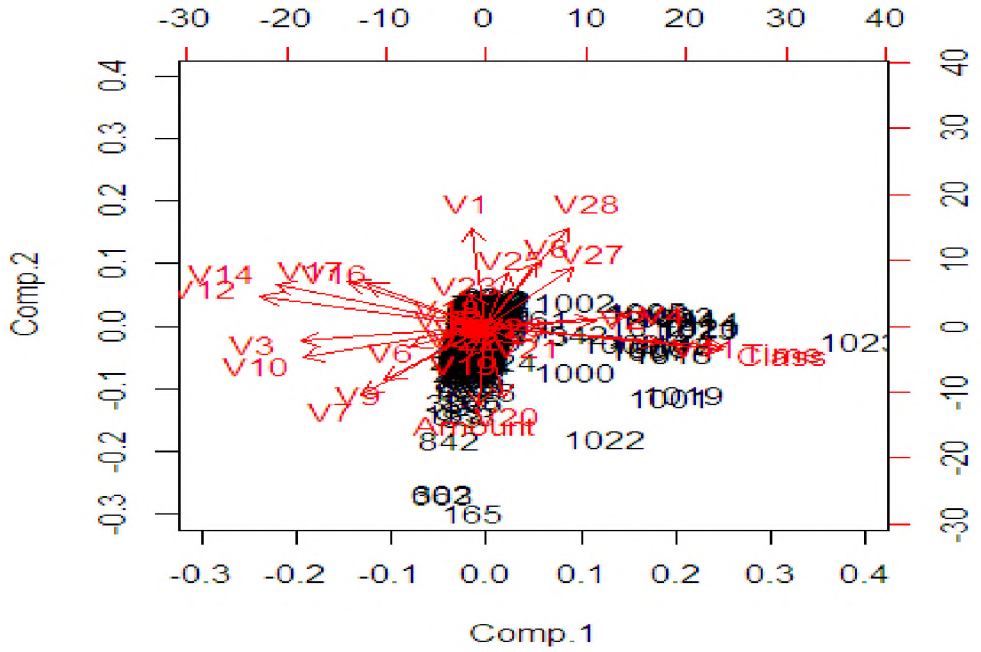
Annexure C: Biplot(1)



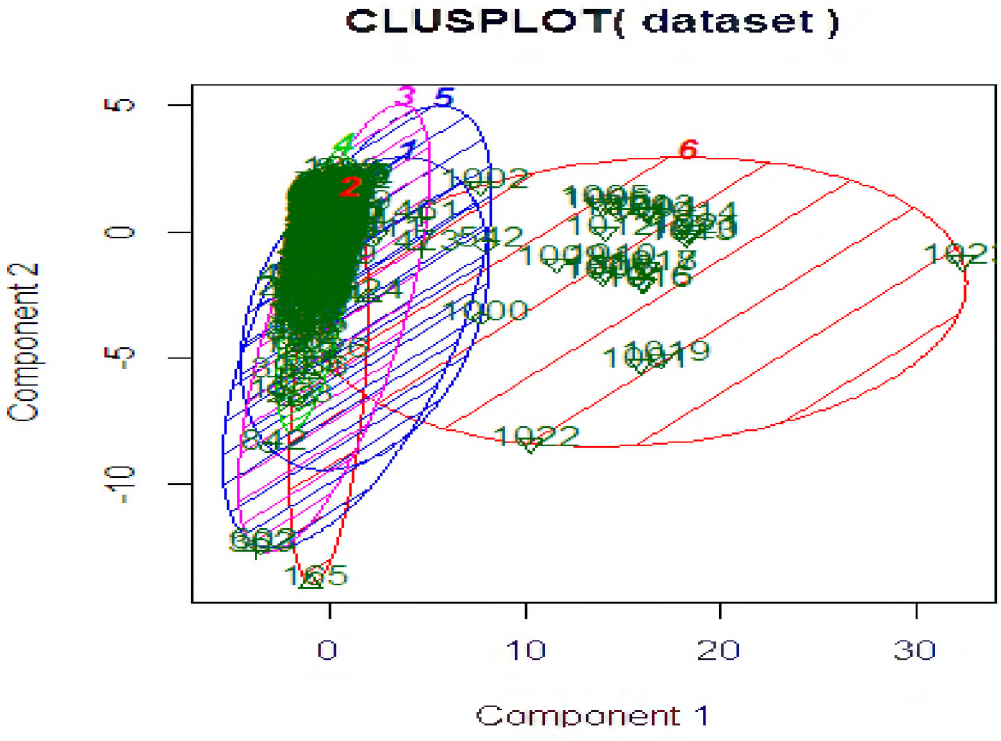
Annexure D: Principal Component Analysis(2)



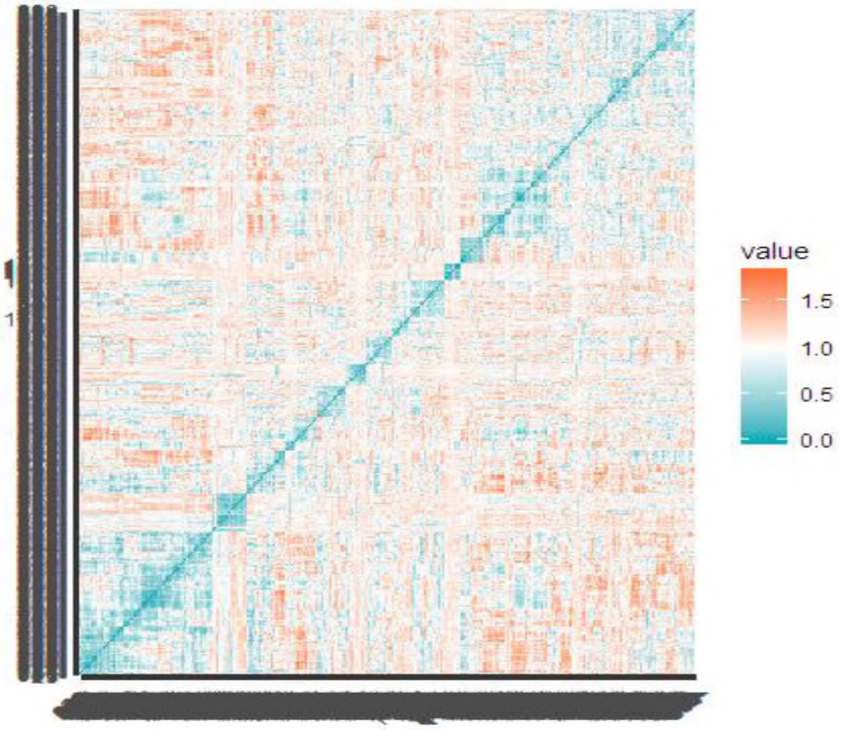
Annexure E: Biplot (2)



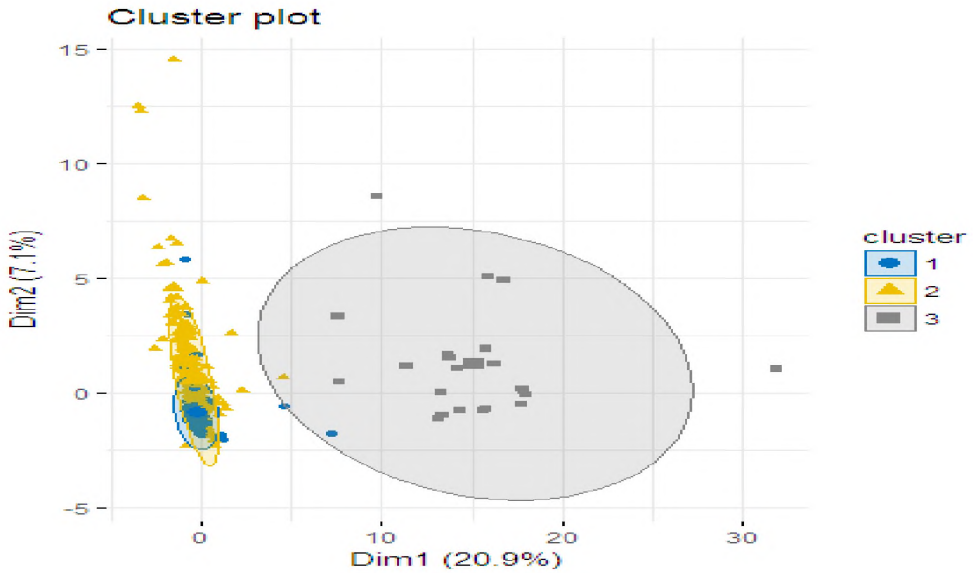
Annexure F: Dataset Cluster plot



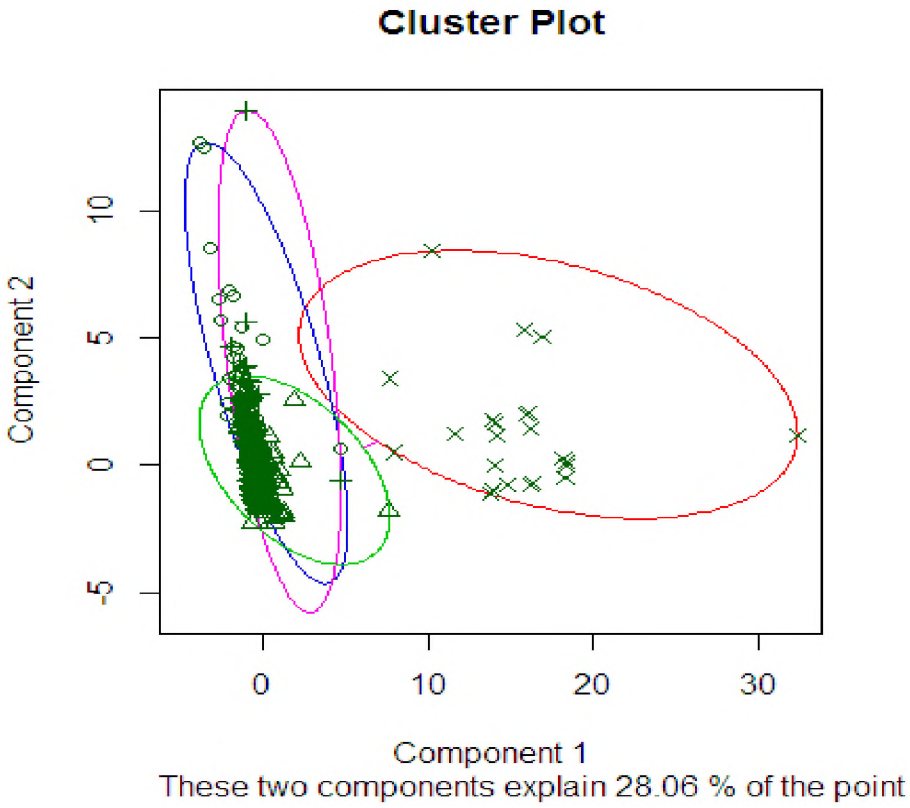
Annexure G: Heat Map



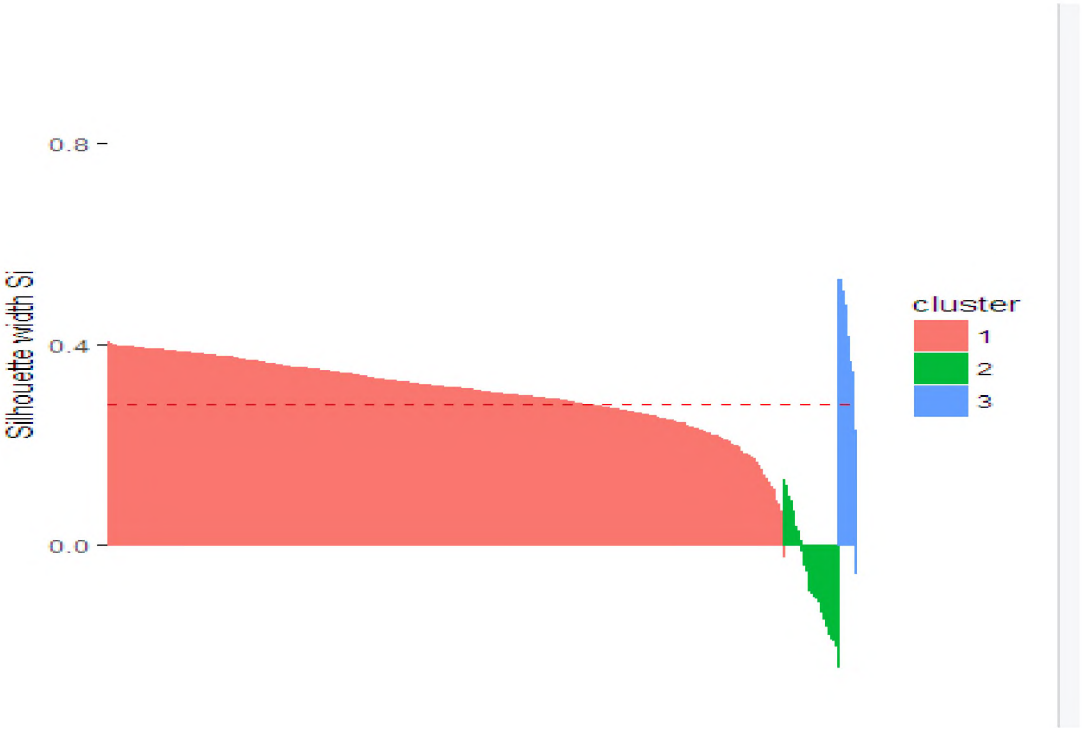
Annexure H: K-Means Plot



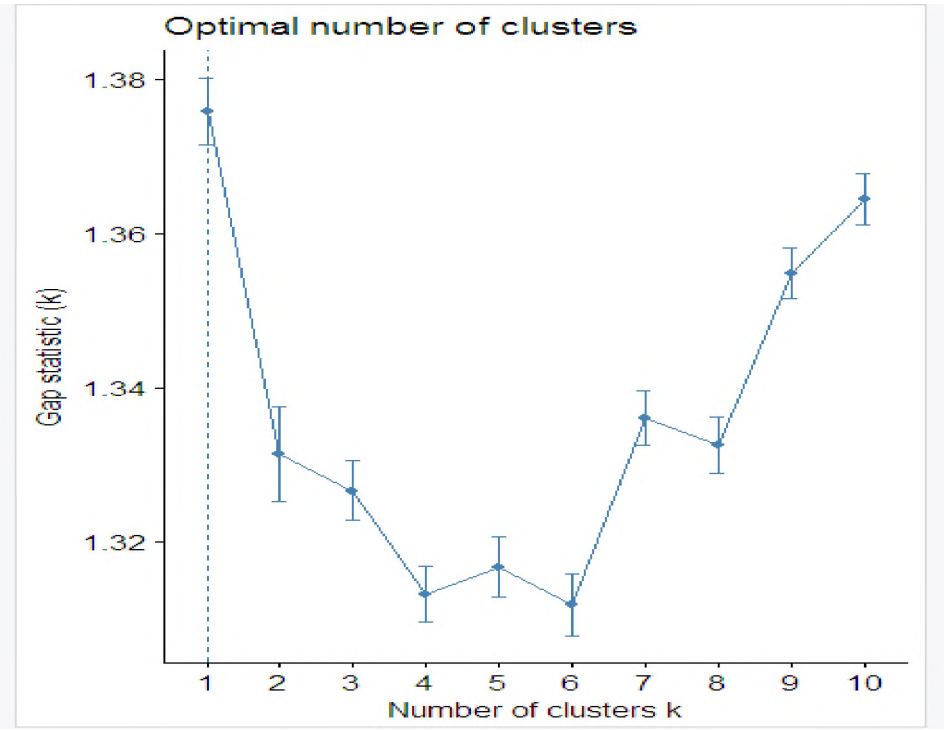
Annexure I: K-Medoid Cluster



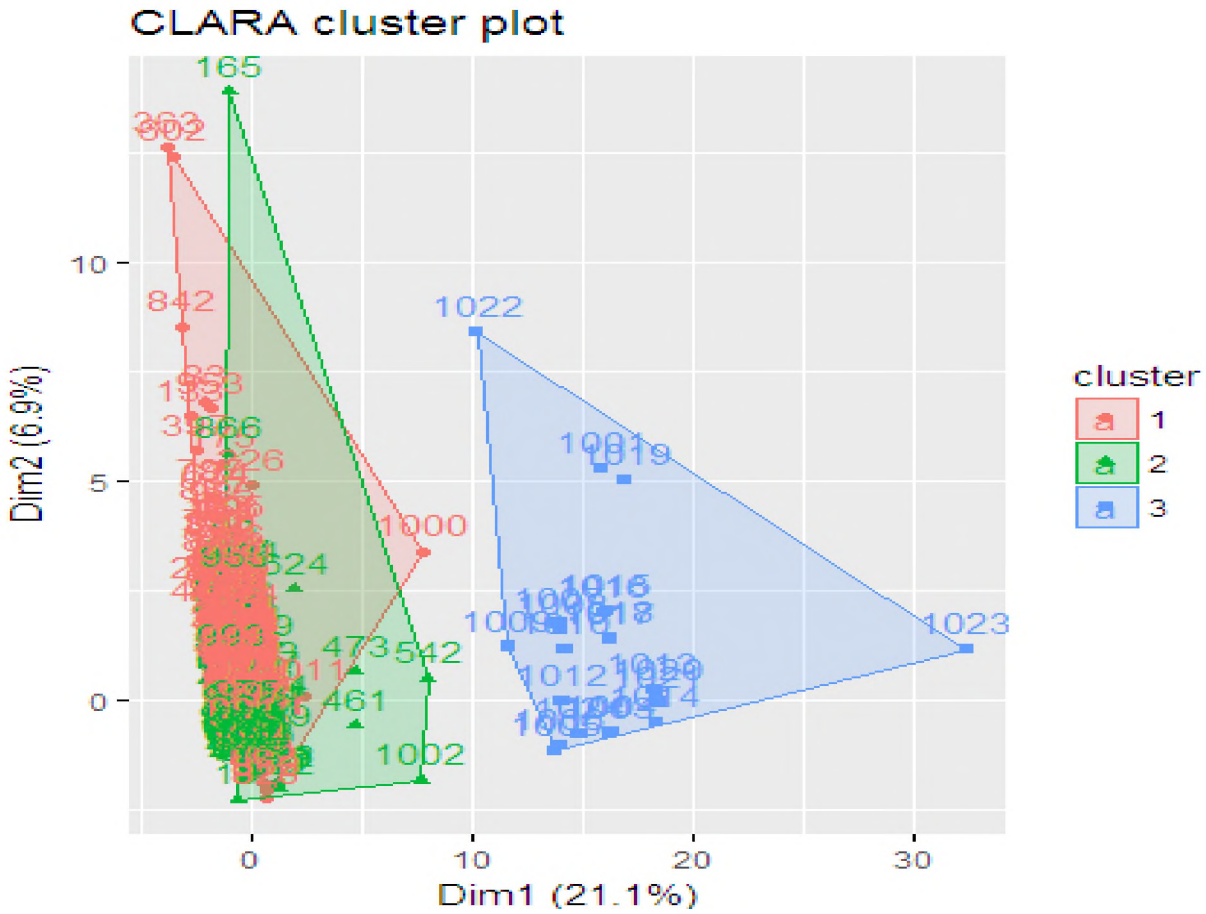
Annexure J: Average Silhouette Width



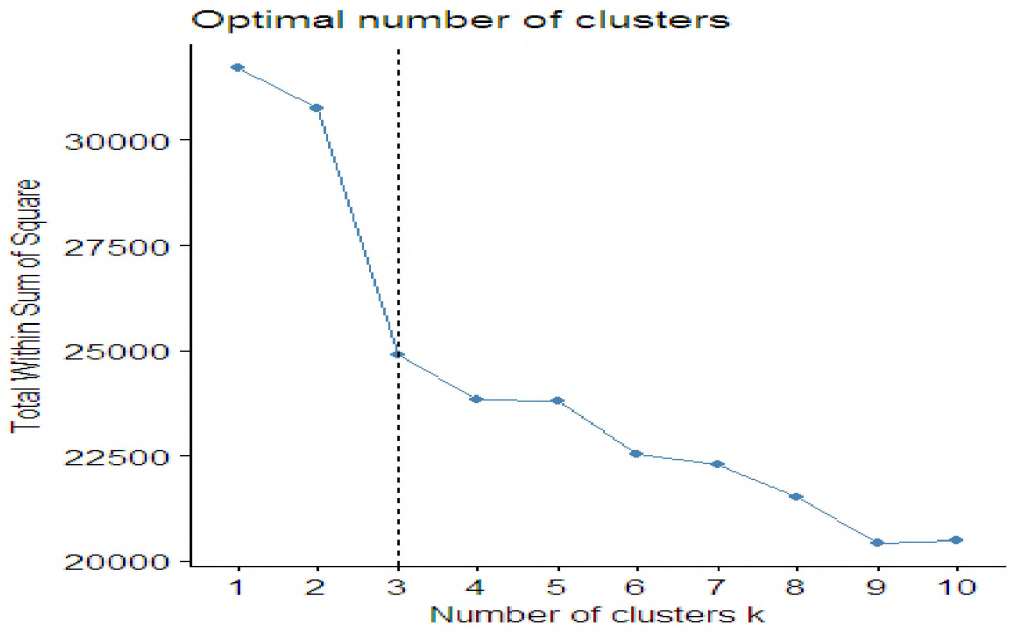
Annexure K: Gap Statistics plot (k-Means)



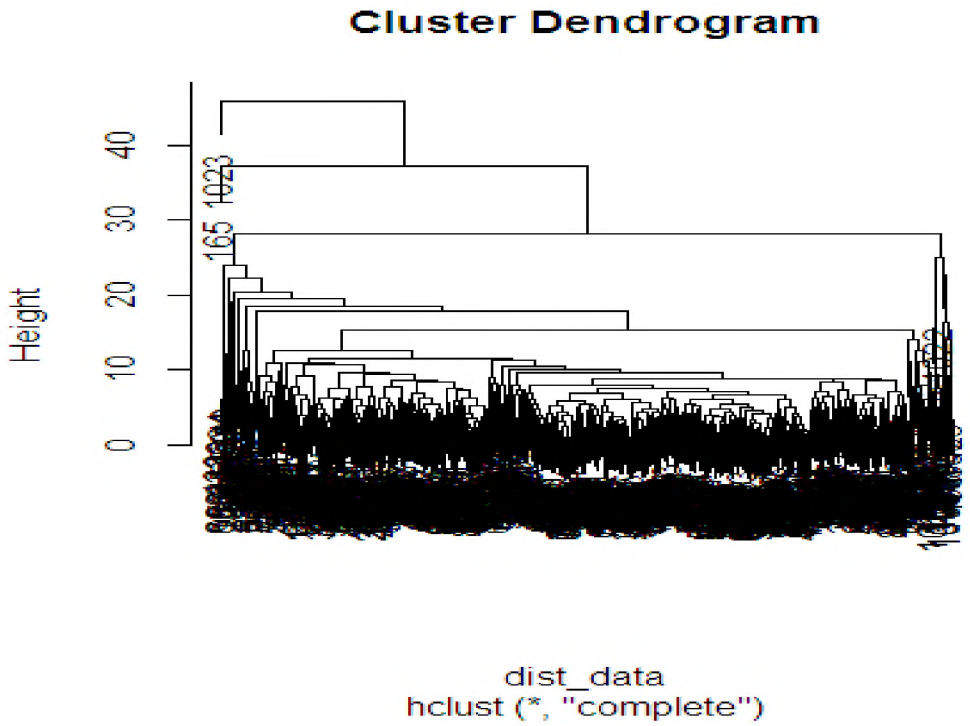
Annexure L: Clara Clustering Plot



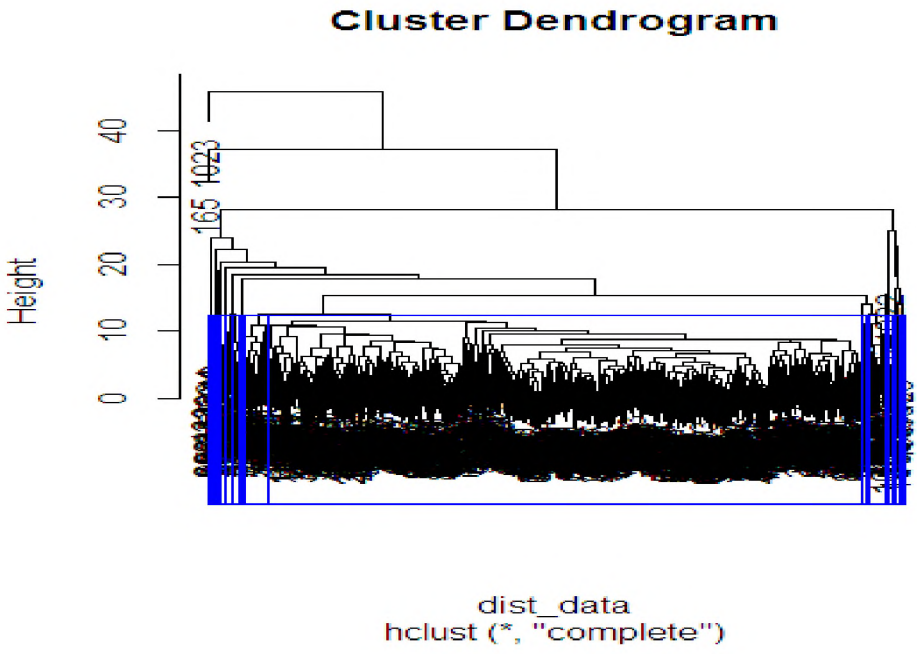
Annexure L: Clara optimal Cluster



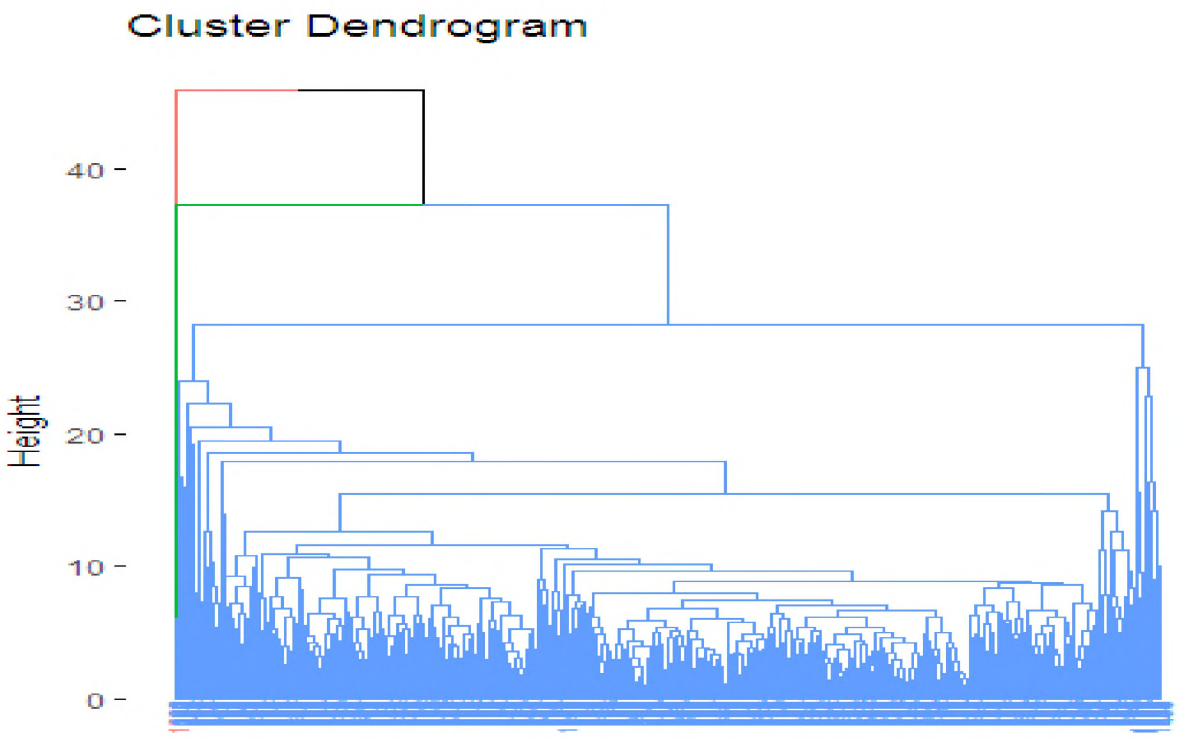
Annexure I: Hierarchical Clustering



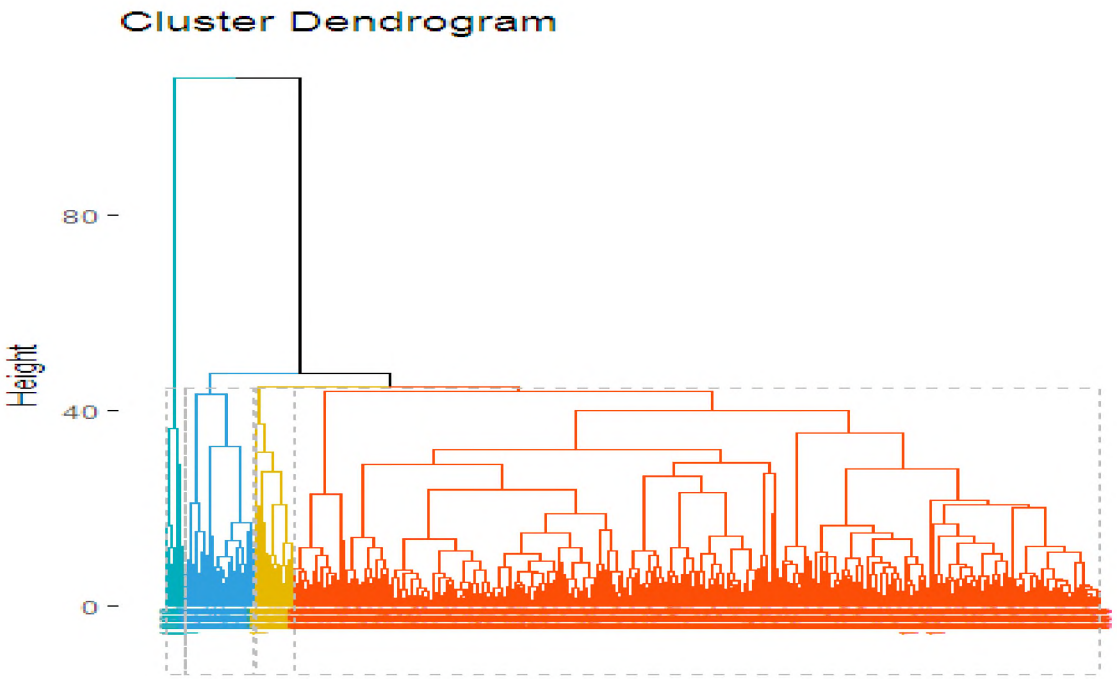
Annexure J: Rect Clust plot in Hierarchical Clustering



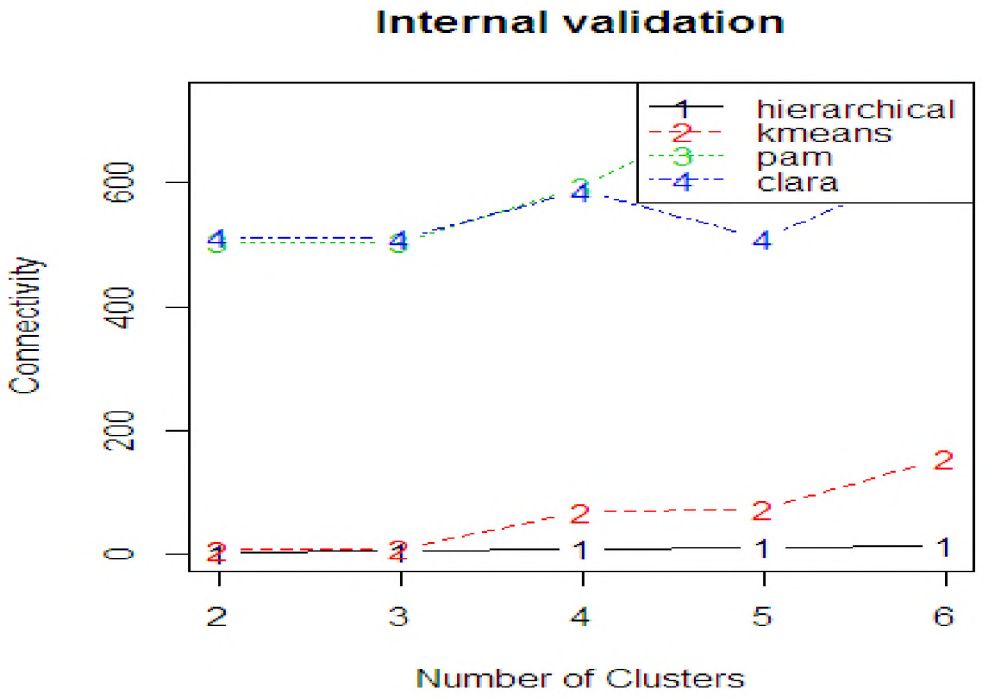
Annexure K : Hierachcial Cluster Color plot



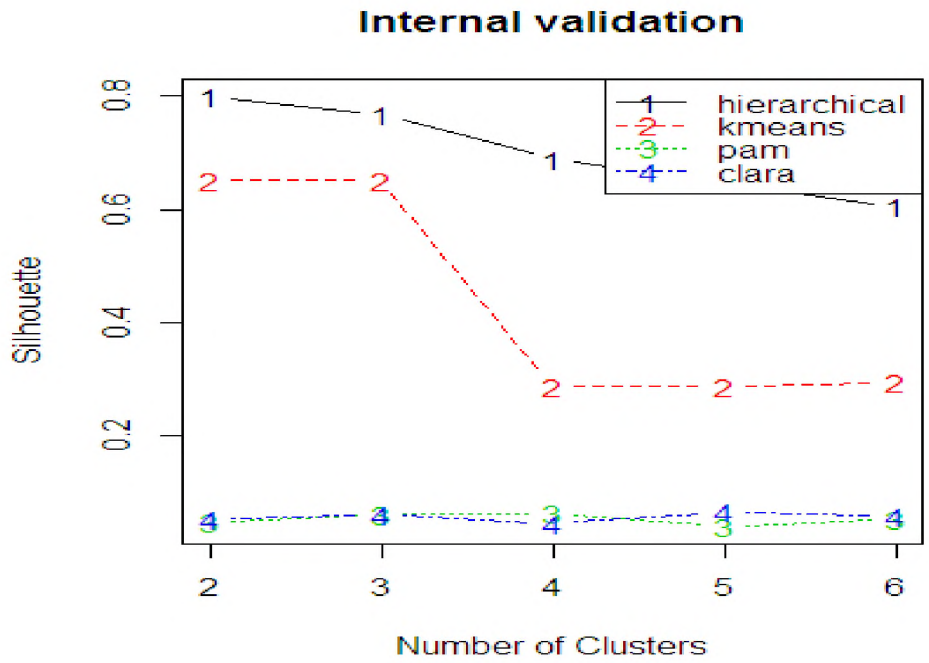
Annexure L: Cut tree dendrogram in Hierarchical Clustering



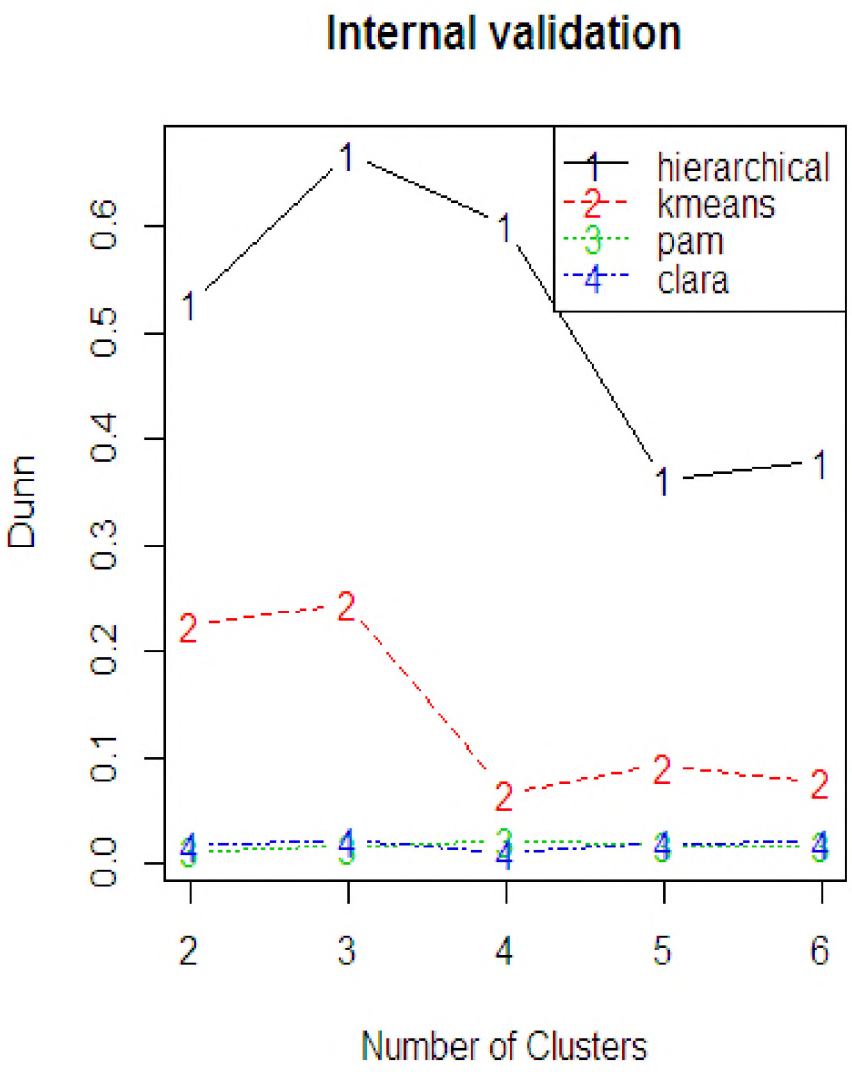
Annexure M: Connectivity Measure of Internal Validation



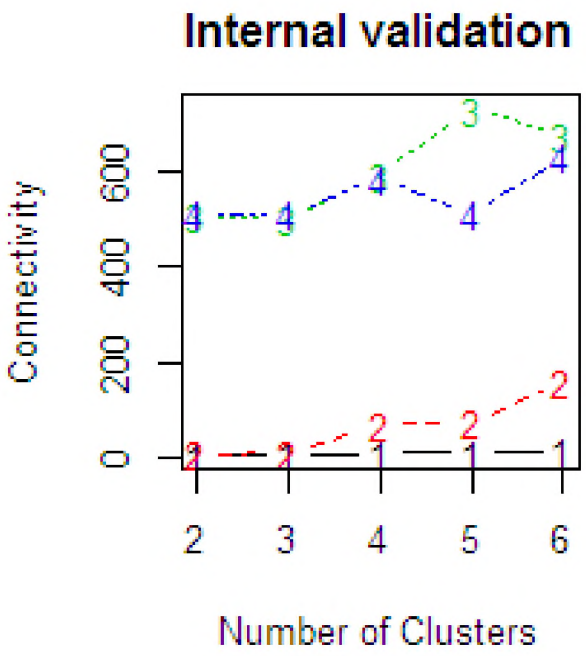
Annexure N: Silhouette Measure of Internal Validation

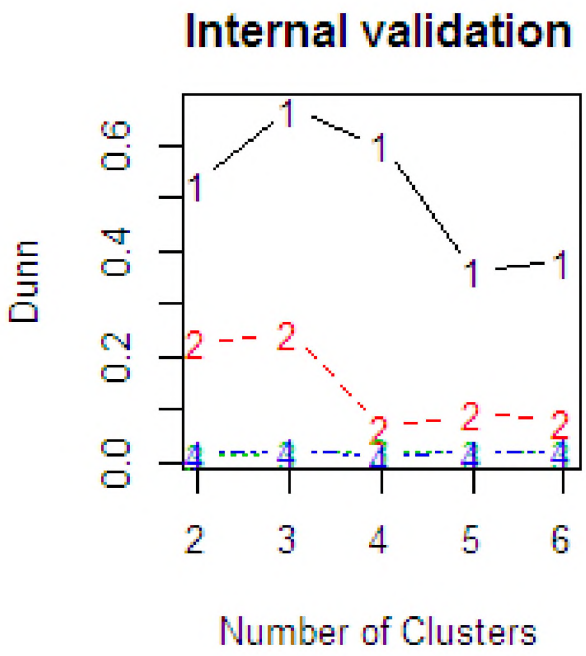


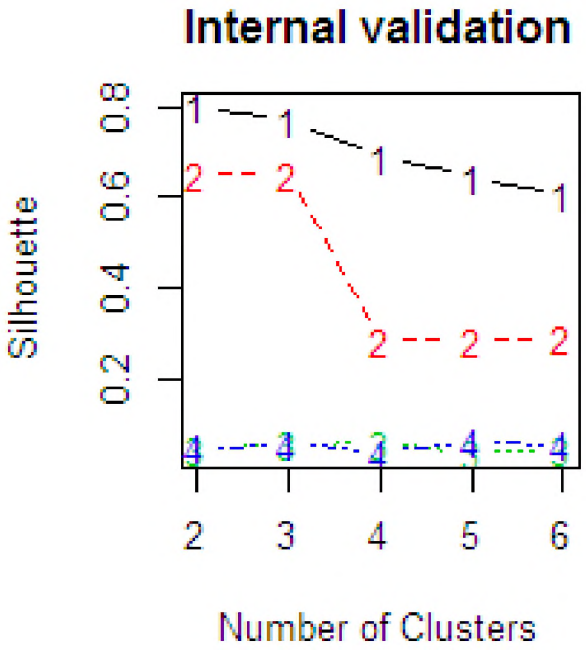
Annexure O: Dunn Measure of Internal Validation

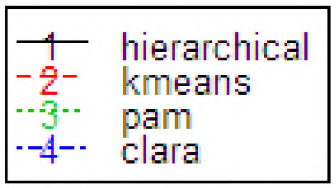


Annexure P: Overall Output of Internal Measures

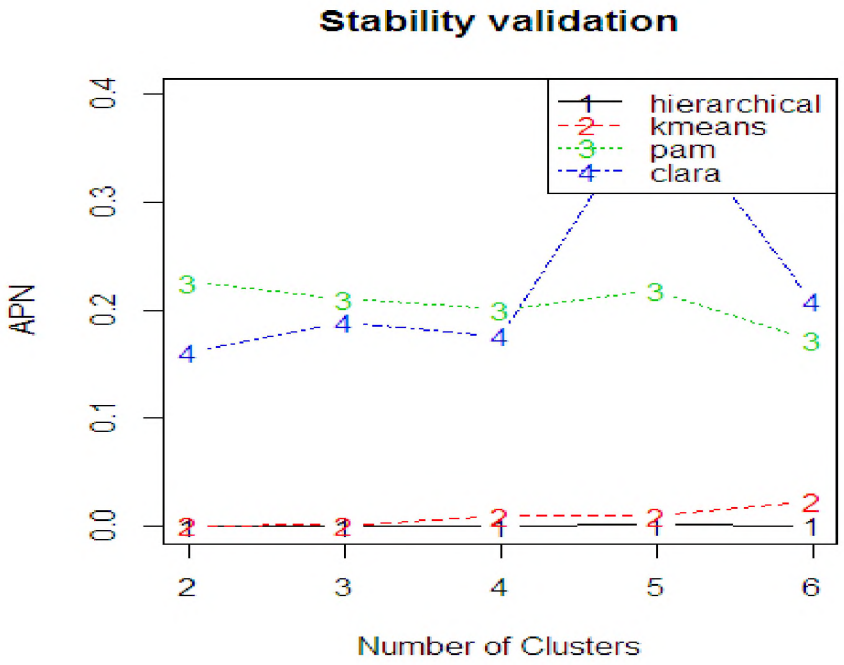




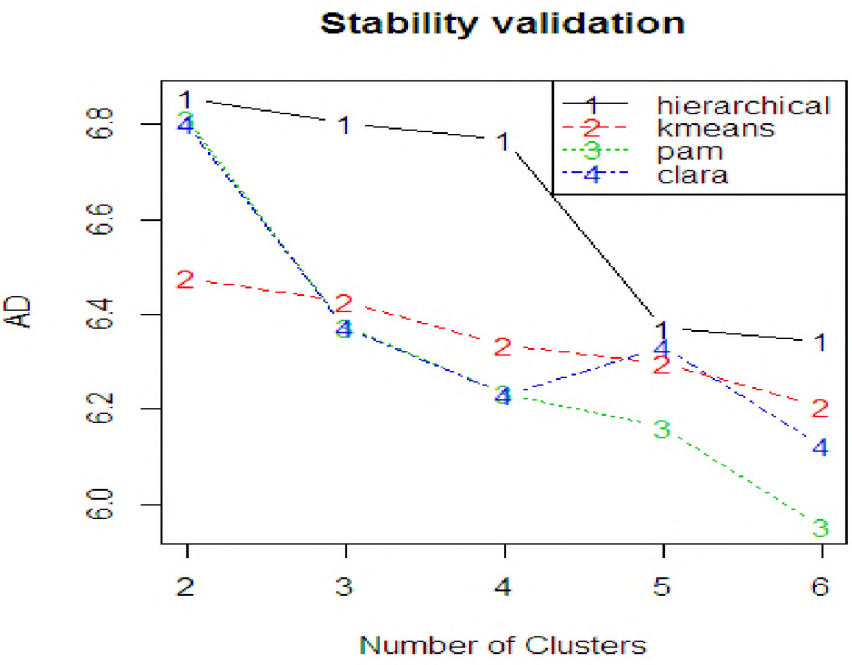




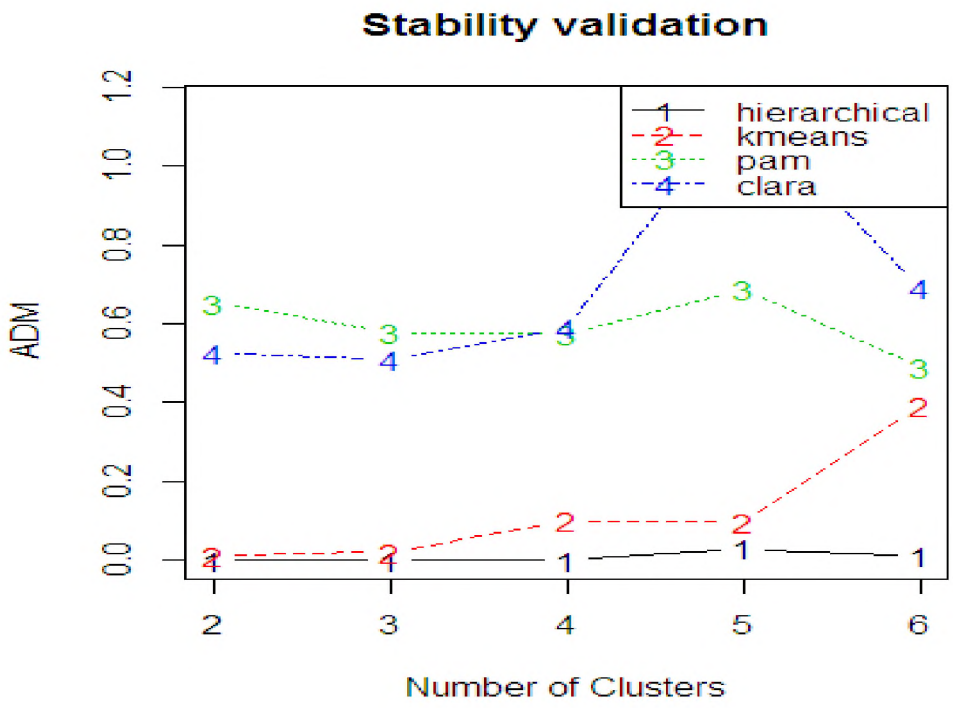
Annexure Q: Stability Measure of APN



Annexure R: Stability Measure of AD



Annexure S: Stability Measure of ADM



Annexure T: Stability Measure of FOM