DETECTING CREDIT CARD FRAUD TRANSACTIONS USING MACHINE LEARNING

Project Report

Submitted in partial fulfillment of the requirements for the award of degree

Of

Bachelor of Technology

In

COMPUTER SCIENCE AND ENGINEERING

By

G YASASWINI	(17K61A05D6)
K MANJUSHA	(17K61A0548)
T SAI AVINASH	(17K61A05H2)

Under the esteemed guidance of
MR. S. JAGAN MOHAN, Assistant Professor

Department of Computer Science and Engineering
SASI INSTITUTE OF TECHNOLOGY & ENGINEERING
(Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada and SBTET-Hyderabad, Accredited by NAAC with 'A' Grade and NBA, Ranked as "A" Grade by Govt. of A.P., Recognised by UGC 2(f) & 12(B))
Kadakatla, Tadepalligudem– 534 101

ACADEMIC YEAR 2020-2021

VISION AND MISSION OF INSTITUTE

VISION

Compete as a premier institute for professional education by creating technocrats who can address the society's needs through inventions and innovations.

MISSION

1. Partake in the national growth of the technological, industrial arena with societal responsibilities.
2. Provide an environment that promotes productive research.
3. Meet stakeholder's expectations through continued and sustained quality improvements.

VISION AND MISSION OF DEPARTMENT

VISION

Build a learning environment that enhances creativity, social awareness and leadership skills for total personality development.

MISSION

The Computer Science and Engineering department's consistent effort is to provide the learner with an exposure to emerging technologies by providing hands-on experience making them creative and research-oriented professionals with values, Leadership qualities and zeal to serve the society

PROGRAM OUTCOMES (POs)

Students in the Computer Science and Engineering program should, at the time of their graduation be in possession of:

PO1.Engineering Knowledge: Apply knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

PO2.Problem Analysis: Identify, formulate, research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.

PO3.Design/ Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal and environmental considerations.

PO4. Conduct investigations of complex problems using research-based knowledge and research methods including design of experiments, analysis and interpretation of data and synthesis of information to provide valid conclusions.

PO5.Modern Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering and IT tools including prediction and modelling to complex engineering activities with an under- standing of the limitations.

PO6.The Engineer and Society: Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice.

PO7.Environment and Sustainability: Understand the impact of professional engineering solutions in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.

PO8.Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.

PO9. Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.

PO10.Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations and give and receive clear instructions.

PO11.Life-long Learning: Recognize the need for and have the preparation and ability to engage in independent and life- long learning in the broadest context of technological change.

PO12.Project Management and Finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1.Mobile & Web Application Development: Ability to develop mobile & web applications using J2EE, Android and J2ME.

PSO2.Cloud Services: To deploy virtualized and cloud-based services in the organization.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates will be able to analyze, design and develop advanced computer applications to provide solutions to the complex problems.

PEO2: Graduates are well trained, confident, research oriented and industry ready professionals who are intellectual, ethical and socially committed.

PEO3: Graduates will have the technical, communication skills and character that will prepare them for technical and leadership roles.

COURSE OUTCOMES (COs)

CO1. Develop problem formation and design skills for engineering and real world problems.
CO2. Collect and Generate ideas through literature surveys on current research areas which help to analyze and present to impart knowledge in different fields.
CO3. Impart knowledge on software & hardware to meet industry perspective needs and standards.
CO4. Create interest to carry out research on innovative ideas as a lifelong learning.
CO5. Ability to work with a team, and enrich presentation and communication skills.
CO6. Create a platform that makes students employable

EXPECTED OUTCOMES

PROGRAM OUTCOMES (POs)

PO1: Engineering Knowledge

PO2: Problem Analysis

PO3: Design/Development of Solutions

PO4: Conduct investigation of complex problems PO5: Modern Tool Usage

PO6: The Engineer and Society

PO7: Environment and Sustainability

PO8: Ethics

PO9: Individual Team Work

PO10: Communication

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: **Mobile & Web Application Development**

SASI INSTITUTE OF TECHNOLOGY & ENGINEERING

(Approved by AICTE, New Delhi, Affiliated to JNTUK, Kakinada and
SBTET-Hyderabad Accredited by NAAC with 'A' Grade and NBA)
Kadakatla, TADEPALLIGUDEM – 534 101, WG-Dst, A P, Ph: 08818
244989 **Department of Computer Science and Engineering**

CERTIFICATE

*This is to certify that the project work entitled "**Detecting Credit Card Fraud Transactions Using Machine Learning**" is being submitted by **G. Yasaswini (17K61A05D6), K. Manjusha (17K61A0548), T. Sai Avinash (17K61A05H2)** in partial fulfilment for the award of the degree of **BACHELOR OF TECHNOLOGY**, in **Computer Science and Engineering** to Jawaharlal Nehru Technological University, Kakinada during the academic year 2020 to 2021 is a record of bonafide work carried out by them under my/our guidance and supervision. The results presented in this thesis have been verified and are found to be satisfactory. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any other degree or diploma.*

Project Supervisor

Mr. S. Jagan Mohan

Assistant Professor

Dept of CSE

Head of Department
Dr.M.V.S.S. Nagendranath
Associate Professor & HOD of CSE

External Examiner

DECLARATION BY THE CANDIDATES

We **G. Yasaswini (17K61A05D6), K. Manjusha (17K61A0548), T. Sai Avinash (17K61A05H2),** hereby declare the project report entitled "**Detecting Credit Card Fraud Transactions using Machine Learning**" under esteemed supervision of **Mr. S. Jagan Mohan**, is submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering. This is a record of work carried out by us and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other University or Institute for the award of any other degree or diploma.

G. Yasaswini(17K61A05D6)
K. Manjusha(17K61A0548)
T. Sai Avinash(17K61A05H2)

ABSTRACT

This project is mainly focused on Credit card fraud detection in the real-world. Nowadays Credit card frauds are increasing drastically compared to earlier days. Fraudsters are using fake identity and various technologies to trap the users and get the money out of them. Therefore, it is very essential to provide a solution for this problem. In this proposed system, we have designed a model to detect the fraud in credit card transactions. To detect the fraud, we have to identify the past transactions in order to check the behavior and patterns of the users. So we are going to apply a machine learning algorithm to detect the fraud. Because ML is a powerful technology that is easy to implement and understand the patterns. In ML, Random Forest Algorithm is one of the classification and regression algorithms that provides very fast and efficient results. So we are going to use this classifier to detect fraudulent transactions.

ACKNOWLEDGEMENT

We pay obeisance to the dynamic Chairman of Sasi Educational Society, **Mr. Sri Burugupalli Venu Gopala Krishna**, for his inspiring presence, which has always been the principal driving force behind all over endeavors.

First of all, we would like to thank **Mr. Meka. Narendra Krishna**, Vice-Chairman, Sasi Institute of Technology & Engineering, for his everlasting support.

We would like to thank **Dr. K. Bhanu Prasad**, Director, Sasi Institute of Technology & Engineering, for his support.

We would like to thank **Dr.J. Annand Chadu Lal**, Principal, Sasi Institute of Technology & Engineering, for his support.

We would like to thank **Dr.M.V.S.S. Nagendranath**, Associate Professor and Head of the Department, Computer Science and Engineering, for providing us with invaluable feedback on our work, which allowed us to constantly improve.

We are grateful to our project coordinators **Dr. M. Kandan** and thanks to all teaching and non teaching staff members who contributed for the successful completion of our project work.

We would like to thank our Guide **Mr. S. Jagan Mohan**, Asst Professor, Sasi Institute of Technology & Engineering for her constant encouragement, monitoring and guidance throughout the submission of the project. She motivated us whenever we encountered an obstacle along the way.

With gratitude,

G. Yasaswini(17K61A05D6)

K. Manjusha(17K61A0548)

T. Sai Avinash(17K61A05H2)

CONTENTS

		Page No
ABSTRACT		**viii**
LIST OF FIGURES		**ix**
LIST OF TABLES		**x**
CHAPTER I: INTRODUCTION		**1-6**
1.1	Preamble	1
1.2	Problem Statement	2
1.3	Objective of the project	3
1.4	Methodology	3
1.5	Significance of the work	4
1.6	Organization of the report	5
1.7	Summary	6
CHAPTER II: LITERATURE SURVEY		**7-16**
2.1	Literature Review	7
2.2	Comparison table of Literature Review	14
2.3	Summary	16
CHAPTER III: SOFTWARE REQUIREMENTS		**17-20**
3.1	Tools used	17
3.2	Dataset	19
3.3	Summary	20
CHAPTER IV: SYSTEM DESIGN		**21-36**
4.1	Architecture Diagram	21
4.2	Use Case Diagram	22
4.3	Class Diagram	23
4.4	Behaviour Diagram	24
4.5	Implementation Diagram	30
4.6	Data Flow Diagram	33
4.7	Summary	34

CHAPTER V: IMPLEMENTATION **35-44**

5.1 Proposed Methodology 35

5.2 Proposed Methodology Module 37

5.3 Data Collection 37

5.4 Data Preprocessing 38

5.5 Feature Selection 39

5.6 Splitting the dataset 41

5.7 Training & Testing the model 42

5.8 Evaluating the model 43

5.9 Summary 44

CHAPTER VI: EXPERIMENTAL RESULTS **45-46**

6.1 Results 45

CHAPTER VII: CONCLUSION AND FUTURE ENHANCEMENT **47**

APPENDIX A **50-53**

APPENDIX B **54-57**

APPENDIX C **58**

APPENDIX D **59-62**

LIST OF FIGURES

FIG NO	FIG NAME NO	PAGE
Fig 1.1	Machine Learning Technology	1
Fig 1.2	Architecture of Fraud Detection	2
Fig 4.1	Architecture diagram	21
Fig 4.2	Use case diagram	22
Fig 4.3	Class diagram	23
Fig 4.4	Sequence diagram	24
Fig 4.5	Collaboration diagram	25
Fig 4.6	State diagram	27
Fig 4.7	Activity diagram	28
Fig 4.8	Component diagram	30
Fig 4.9	Deployment diagram	31
Fig 4.10	Data flow diagram	32
Fig 5.1	Pie chart of dataset	36
Fig 5.2	Top 5 rows of dataset	36
Fig 5.3	describe() for dataset	37
Fig 5.4	Pre-processing the dataset	39
Fig 5.5	Dataset splitting	41
Fig 6.1	Pie chart showing the fraud and valid transactions	45
Fig 6.2	Heatmap showing the visual representation of whole dataset	46

LIST OF TABLES

TABLE NO	TABLE NAME	PAGENO
Table 2.1	Comparison of Literature Review	14

CHAPTER I

INTRODUCTION

1.1 PREAMBLE

In this chapter we are going to introduce our project domain, it's architecture, problem statement, aim, methodology we are going to include in our project and what we are going to do in our project.

1.1.1 Machine Learning

Machine learning is a powerful technology that provides the machine the ability to automatically learn and improve without being explicitly programmed. This means machines will be trained according to the human's needs. This is one of the powerful technologies that everyone can easily use. This means that everyone should be able to learn, play and create great applications by using this technology. It provides smart alternatives to analyze the huge volumes of data. It focuses on development of computer programs that can access the data and use it to learn for themselves. By developing fast and efficient algorithms for real-time processing the data, this technology provides more accurate results and analysis. That's why we are going to use these algorithms in our project for detecting frauds in credit card transactions to get more accurate results.

Figure 1.1 Machine Learning Technology

1.1.2 Architecture

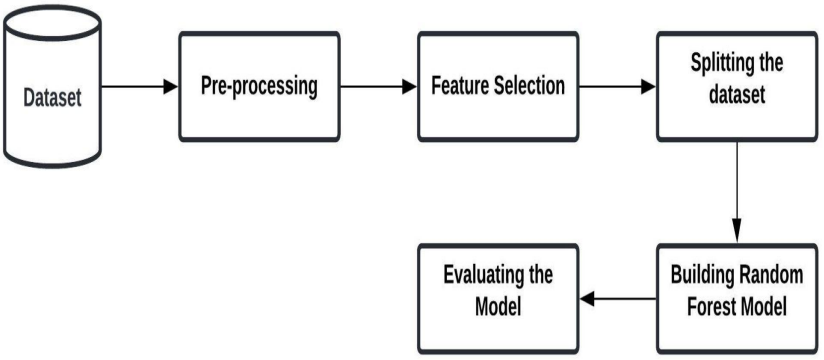


Figure 1.2 Architecture of Fraud Detection

1.1.3 Detection Analysis in Machine Learning

Detection Analysis is used to identify anomalous data from huge data. It is nothing but analyzing the outliers from the same set of data. For detecting frauds, Machine Learning technology provides efficient algorithms such as Linear Regression, Logistic Regression, Random Forest, Naive Bayes, K-Nearest Neighbor and Neural Networks are classification algorithms that can be used for detecting fraudulent transactions.

It is the identification of rare items, events or observations which looks different from the majority of the data. Anomalous data can lead to some problems or rare events such as credit card fraud, bank fraud, medical problems, structural defects etc. So we are going to detect the frauds from non fraudulent transactions to reduce the financial loss occurring for industries or maybe for customers.

1.2 PROBLEM STATEMENT

Credit card frauds are increasing heavily because of fraud financial loss is increasing drastically. Every year due to fraud, Billions of amounts are lost. To analyse the fraud there is a lack of research. The existing systems faced some challenges by detecting fraudulent transactions in credit cards. Those are:

- ➢ Enormous data is processed everyday and the model build must be fast enough to respond to the scam in time
- ➢ Imbalanced data i.e., most of the transactions are not fraudulent which makes it really difficult to find the fraudulent ones
- ➢ Data availability as the data is mostly private
- ➢ Misclassified data can be another major issue, as not every fraudulent transactions is caught and reported
- ➢ Adaptive techniques can be used by the scammers

These are the challenges that have occured in the existing systems. So we are going to overcome these challenges by using efficient Machine Learning techniques.

1.3 OBJECTIVES OF THE PROJECT

- ➢ To identify the suspicious events and report them to an analyst while letting normal transactions be automatically processed.
- ➢ To reduce losses due to payment fraud for both merchants and issuing banks and increase revenue opportunities for merchants.
- ➢ To implement Machine Learning algorithms for detecting frauds in credit card transactions.
- ➢ To detect fraudulent activities the credit card fraud detection system will be introduced.

1.4 METHODOLOGY

In our daily lives credit cards are widely used all over the world. Therefore credit card fraud is increasing every day. Data scientists are trying to get the best solution to detect this type of fraud. Since credit cards contain sensitive data, credit card fraud not only affects the cardholders but also affects the banks, governments and types of the financial departments, resulting in high financial losses. To overcome this loss and avoid this situation, many machine learning algorithms can be used to detect fraudulent transactions. Algorithms such as logistic regression, naive bayes, random forest, K nearest neighbor and neural networks are classification algorithms that can be used to detect fraudulent transactions.

Comparative analysis is performed to find out which algorithm model performs best among them and provides the best solution. Data mining technology is mainly used to find patterns in the data. For these patterns machine learning algorithms can be used to train the model. Using these development techniques has improved the accuracy of detecting fraudulent transactions.

1.5 SIGNIFICANCE OF THE WORK

The following steps are the modules that we are going to perform step by step in our project.

Step 1: Data Collection

When we use machine learning,the first step is to understand your problem. According to the problem definition, collect data. For machine learning, you can create a dataset or use an existing dataset. There are many platforms that provide datasets to solve machine learning problems.

Step 2:Data Preprocessing

After collecting the data, it needs to be processed. If the data is not preprocessed or the original data is provided to the model, the expected results cannot be provided. Try to use the technology that can provide the best form of data to improve the accuracy of the model. If the dataset is skewed, try to balance it, perform feature selection, feature extraction and transfer learning.

Step 3: Split the dataset

After cleaning the data, divide the dataset. The data can be divided into train test ratio, train test verification ratio or use cross-validation. By splitting the data set, you can provide a training data set for training the model, and rest for evaluating the model. By doing this, we avoid overfitting of the model.

Step 4: Choosing a model

After preprocessing the data, we need to select a model based on our data set and perform the types of tasks that need to be performed, such as classification and

clustering. It is very important to choose the right model, otherwise the results will not be obtained.

Step 5: Evaluate the model

After training the model, predict the result on an invisible data set. If the predictor provides the expected results, it can be said that the model is ready to classify the data. If the results are not satisfactory, please retrain the model and change the parameters, fine-tune them, and then continue to try until you get satisfactory results.

1.6 ORGANISATION OF THE WORK

CHAPTER I:INTRODUCTION

This chapter contains PROBLEM STATEMENTS, OBJECTIVE OF THE PROJECT, METHODOLOGY, SIGNIFICANCE OF THE WORK, SUMMARY.

CHAPTER II:LITERATURE SURVEY

This chapter contains INTRODUCTION, COMPARISON OF LITERATURE SURVEY and we need to explain all the IEEE papers, SUMMARY.

CHAPTER III: SYSTEM REQUIREMENTS

This chapter contains TOOLS USED, DATASET, SUMMARY.

CHAPTER IV: SYSTEM DESIGN

This chapter contains ARCHITECTURE DIAGRAM, USECASE DIAGRAM, CLASS DIAGRAM, BEHAVIOUR DIAGRAM, IMPLEMENTATION DIAGRAM, DATA FLOW DIAGRAM, SUMMARY.

CHAPTER V: IMPLEMENTATION

This chapter contains PROPOSED METHODOLOGY, PROPOSED METHODOLOGY MODULE, DATA COLLECTION, DATA PREPROCESSING, FEATURE SELECTION, SPLITTING THE DATASET, TRAINING & TESTING THE MODEL, EVALUATING THE MODEL, SUMMARY.

CHAPTER VI: EXPERIMENTAL RESULTS

This chapter contains RESULTS of the project

CHAPTER VII: CONCLUSION AND FUTURE ENHANCEMENT

This chapter contains CONCLUSION, FUTURE ENHANCEMENT of our project.

1.7 SUMMARY

In today's digital world, all the transactions are done through debit or via credit cards. Today, the use of cash has been reduced drastically compared to debit or credit card transactions. Due to the increase in cashless transactions, fraudulent transactions are also increasing rapidly. Therefore credit card fraud detection can be used here. Many machine learning algorithms are used. By comparing all the algorithms, we will discover which algorithm produces the best accuracy for the detection of the frauds.

CHAPTER II

LITERATURE SURVEY

PREAMBLE

In chapter 1, we have discussed our project domain and what we are going to do in this project. Based on chapter 1, we have surveyed about 20 papers related to our project. In this chapter, we will discuss those 20 papers briefly. From that we will get to know the limitations in the existing systems. And we will try to rectify those limitations in our project by considering efficient techniques.

2.1 LITERATURE REVIEW

In this section, we will briefly discuss all the papers one by one that we have surveyed related to our project.

2.1.1 Credit card fraud detection using AdaBoost and Majority Voting

Chu Kiong Loo is the author of this paper. In this paper[1], they have used AdaBoost and majority voting methods for detecting credit card frauds. To evaluate these methods efficiently, they have used two data sets. One is considered as a test data set, and for analysis another dataset is considered as a trained data set. Finally the experiment results that the majority voting method achieves best accuracy for detecting fraud transactions in huge volumes of credit card transaction data set.

2.1.2 Credit card fraud detection using Random Forest Algorithm

M. Suresh Kumar is the author of this paper. In this paper[2], they mainly focused on credit card fraud detection in the real-world. They said that fraud detection is purely based on fraudulent transactions. These types of activities can happen both online and offline. But now-a-days online frauds are increasing day-by-day. So to find the online fraud transactions, they have proposed a system that uses Random Forest Algorithm(RFA) and finds the fraud transactions with its accuracy. This algorithm is purely based on supervised learning where it uses decision trees for classification of dataset. After the classification, a confusion matrix is generated. Based on that classification, the performance of RFA is evaluated. And given the accuracy is about 90%.

2.1.3 Credit card fraud detection using Machine Learning Methodology

Hamzah Ali Shukur and **Sefer Kurnaz** are the authors of this paper. In this paper[3], they say that 'Although several criminal activities have occured in commercial activities, fraudulent ecart activities are one of the most common disrupted activities for online customers.' Data processing techniques are used to examine patterns and characteristics of suspicious and non-suspect transactions that support normalization and abnormal knowledge. In contrast, machine learning techniques are used to predict suspicious and non-suspect transactions mechanically by victim classifier. This paper discusses classification based mainly on supervision. When using normalization and principal element analysis to preprocess the dataset, compared with the results obtained before the preprocessing of the data set, the accuracy of all the classifiers reached 95.0%.

2.1.4 Credit card fraud detection using Machine Learning Algorithms

Pratibha K is the author of this paper. The author in this paper[4] says that the credit card frauds are increasing day-by-day and also some banking companies and the company's that are providing services are also facing problems. So, In this paper they have tried to build a model to best predict fraud and non-fraud transactions by using machine learning algorithms and neural networks. Their main purpose is to predict fraud and reduce fraud. They built a complex machine learning model by using statistics, calculus and linear algebra for predicting the fraud. They have achieved 98% accuracy by using ANN.

2.1.5 Machine Learning for credit card fraud detection system

Lakshmi S V S S is the author of this paper. In this paper[5], they have investigated the performance of logistic regression, decision tree and random forest for credit card fraud detection. They have collected a dataset. And the author has been done oversampling to balance the dataset. The mentioned three techniques were applied and work is implemented in R language. The performance is evaluated based on sensitivity, specificity, accuracy and error rate. The results show that the random forest classifier performs best compared to logistic regression and decision tree with an accuracy of 95.5%.

2.1.6 A review on credit card fraud detection using Machine Learning

Suresh K Shirgave and **Chetan J. Awati** are the authors of this paper. In this paper[6], they have considered machine learning because it is one of the most successful techniques to identify frauds. They have reviewed different fraud detection techniques and compared them by using performance measures like specificity, accuracy and precision. They have also proposed a FDS which uses RFA. Their proposed system increased detection of fraud in credit cards. In their proposed system, they have also used the learning to rank approach to rank the alert and also effectively addresses the problem depth in fraud detection.

2.1.7 Credit card fraud detection using Machine Learning techniques: A comparative analysis

John O. Awoyemi and **Adebayo O. Adethunmbi** are the authors of this paper. In this paper[7], they have investigated the performance of naive bayes, knn and logistic regression on highly skewed credit card fraud data. They have applied these three techniques on the preprocessed data. They have implemented their work in Python. They have evaluated the performance based on accuracy, sensitivity, specificity, precision, Matthews correlation coefficient and balanced classification rate. Their results show that knn performs better than the other techniques and achieved 97% accuracy.

2.1.8 Credit card fraud detection using Naive Bayesian and C4.5 Decision Tree classifiers

Admel Husejinovic is the author of this paper. In this paper[8], they have used naive bayes and C4.5 decision tree classifiers for predicting the fraud cases in credit card transactions. They have evaluated the performance based on precision, recall, accuracy and PRC rates. As the PRC rates are obtained between 0.99 and 1.00 expresses that those algorithms are quite good at detecting binary class 0 transactions in the dataset. This results in the best PRC class rate 1 being obtained with the C4.5 decision tree. They have predicted 92% of fraud transactions correctly by using the C4.5 decision tree classifier.

2.1.9 Fraudulent detection in credit card transactions using Radial Basis Function Kernel Method based Support Vector Machine

Shrikant Kokate and **Dr. C. M. Sheela Rani** is the author of this paper. In this paper[9], they have proposed a support vector machine Kernel functions model. For analysing the frauds, they have considered four kernel functions like linear, sigmoid, polynomial and RBF kernel functions. By using the SVM kernel functions, they have measured the non linear performance. They have evaluated the performance evaluation based on accuracy, sensitivity, specificity etc. Their result shows all kernels give better accuracy, Sensitivity and specificity than existing classifiers. Hence the new proposed model sigmoid-RBF is developed based on the RBF uses a composite of sigmoid functions in place of the Gaussian function as the basis function. That proposed sigmoid- RBF function gives best performance over the other kernel functions. The accuracy level is 96% which is effectively better than other SVM kernel functions.

2.1.10 Credit card fraud detection using Machine Learning and Data Science

S P Maniraj is the author of this paper. In this paper[10], they identified some problems in credit card fraud and solved them by using data science and machine learning. This problem includes modeling the past credit card transactions using data that have proven to be fraudulent. Then, use this model to identify whether the new transaction is fraudulent. Their goal is to detect 100% of fraudulent transactions while minimizing misclassifications. In this process, they mainly focus on analysing and preprocessing the data set, and deploying a variety of anomaly detection algorithms on the credit card transaction data after PCA conversation, such as local outlier factor and isolation forest algorithm.

2.1.11 Using Variational Auto Encoding in Credit Card Fraud Detection

Huang Tingfei, Cheng Guangquan and Huang Kuihua are the authors of this paper. In this paper[11], they proposed an oversampling method based on variational automatic coding(VAE) and they combined this method with the classic deep learning techniques, to solve the problem. This method is used to generate a large number of different cases from the minority group in the imbalanced data set, and then used to train the classification network. This proposed method is tested on the open credit card fraud dataset which contains the transactions. The experimental

results show that the performance of VAE method is better than the synthetic minority oversampling techniques and traditional deep neural network methods. These results show that novel sampling methods based on VAE can be efficiently applied to imbalanced classification problems.

2.1.12 Credit Card Fraud Detection Using Hidden Markov Model

Mandeep Singh, Sunny Kumar and Tushant Garg are the authors of this paper. In this paper[12], they observed the behaviour of credit card transactions using a Hidden Markov Model[HMM] and it shows how it detects frauds. Initially the HMM was trained with the normal behaviour of the transactions. If any of the transactions was not accepted by the trained HMM with a high enough probability, then it is declared as a fraudulent transaction. They are trying to ensure that no genuine transactions are rejected by the model. It is also going to explain how the HMM can detect whether an incoming transaction is genuine or not.

2.1.13 Enhancing the Credit Card Fraud Detection Through Ensemble Techniques

Aisha Barahim,Amal Alharji and Norah Alasaibia are the authors of this paper. In this paper[13], they investigated the performance of the several individual different classifiers and the combination of the classifiers using the ensemble techniques for the credit card fraud detection.Initially, the three well-known classifiers have been applied. The classifiers are Decision Tree, Naive Bayes and SVM. Then the ensemble learning module has been applied using the boosting technique. In this paper data set used is an open source credit card transaction data set. The performance of these classification techniques is evaluated based on the accuracy, ROC value, F-measure, specificity and the sensitivity. The experimental results show that Boosting with Decision Tree performs better than other techniques.

2.1.14 Adaptive Model for Credit Card Fraud Detection

Imane Sadgali, Nawal Sael and Faouzia Benabbou are the authors of this paper. In this paper[14], they proposed an adaptive method for the credit card fraud detection, which takes advantage of the performance of a technology with high accuracy and takes into account the type of transaction and the customer's personal data. This method is a multi-level framework which includes the bank security,

customer information and the information about the transaction itself. This framework uses different detection algorithms to improve the accuracy.

2.1.15 A Comparative Study on Credit Card Fraud Detection

Arpit Jain, Narendra Kr Sharma, Anshul Aggarwal and Love Aggarwal are the authors of this paper. In this paper[15], they have introduced some ideas of frauds associated with the credit cards or the other online cards. They tried to find a solution so that fraud gets caught even before the transaction takes place or happens. They have applied various supervised algorithms on the data set. The algorithms used in the paper are KNN, Logistic Regression, Random Forest, Support Vector Machine, Decision Tree and Artificial neural network. They have found random forest performs better than all the other algorithms in all the conditions.

2.1.16 A Machine Learning Approach for Credit Card Fraud Detection

Mohammad Gandhi Babu,Pravin Kshirsagar and Boyini Mamatha are the authors of this paper. In this paper[16], they have proposed some algorithms like naive bayes, delivery regression j48 and adaboost. The delivery regression and adaboost algorithms carry out higher in fraud detection. They have collected a data set. Data set from a mastercard transactions are accrued from kaggle and it consists of an entire of two, eighty four, 808 credit card transactions of a european financial institution data set.

2.1.17 Credit Card Fraud Detection using Machine Learning

M.Thirunavukkarasu, Achutha Nimisha and Adusumilli Jyothsna are the authors of this paper. In this paper[17], they have proposed a project to detect the fraud activity in credit card transactions. This system provides an important feature to detect illegal and illicit transactions. To come up with a solution they proposed an algorithm called random forest algorithm. By using the random forest algorithm frauds can be detected easily. The performance is based on the accuracy, sensitivity, specificity and precision. The results indicate the best accuracy with 98.6% by using the random forest algorithm.

2.1.18 Credit Card Fraud Detection Using Machine Learning

Rahul Powar, Rohan Dawkhar and Pratichi are the authors of this paper. In this paper[18], they have proposed an algorithm called KNN technique to detect the frauds. Machine learning techniques are widely used to detect the frauds. KNN is one of the best classifier algorithms that have been used in credit card fraud detection. KNN is a supervised learning algorithm. The performance of these techniques is evaluated based on the accuracy, sensitivity, precision and recall.

2.1.19 Credit Card Fraud Detection based on Machine Learning Algorithms

Heta Naik and Prashasti Kanikar are the authors of this paper. In this paper[19], they have proposed some algorithms like Naive Bayes, Logistic regression, J48 and AdaBoost. Through the comparative analysis it is concluded that Logistic regression and AdaBoost algorithms perform better in credit card fraud detection. By using these algorithms frauds can be detected.

2.1.20 Credit Card Fraud Detection in internet using K-nearest Neighbor Algorithm

C.Sudha and T.Nirmal Raj are the authors of this paper. In this paper[20], they have used a K-nearest neighbor algorithm. The KNN algorithm is an evolutionary search and optimization technology that imitates natural evolution to find the optimal solution for a problem. This method proves with the highest accuracy in the deduction of the fraudulent transactions and minimizing the number of false alarms. If this algorithm is applied to the bank credit card fraud detection detection system the possibility of the fraudulent transactions can be predicted shortly after the credit card transactions.

Table 2.1 Comparison of Literature Review

S.No	Author and Year	Proposed Algorithm	Merits	Demerits/ Future implementation
1.	Chu kiong loo et al.2018	Adaboost and majority voting	Perfect MCC score of 1 has been achieved	Noisy data is not completely removed
2.	M Suresh Kumar et al.2019	Random forest	Accuracy is about 90-95%	Imbalanced data
3.	Hamzah Ali Shukur et al.2019	Logistic regression, KNN	Logistic regression given the best accuracy	Misclassifications
4.	Pratibha K et al.2020	ANN	Accuracy is about 97%	Will use AI to tune the parameters used in this approach
5.	Lakshmi S V S S et al.2018	Logistic regression, decision tree and random forest	Random forest achieves the best accuracy about 95%	Overfitting
6.	Suresh K shirgave et al.2019	SVM, decision tree, random forest	Random forest achieves 96% accuracy	There is a delay in ranking alert
7.	John O. Awoyemi et al.2017	Naive bayes, KNN, Logistic regression	Naive bayes achieves 97.9% accuracy	Try to do research on meta-classifiers to get more accurate results
8.	Admel Husejinovic et al.2020	C4.5 decision tree, Naive bayes	Possess best PRC rate and 92% of frauds are predicted correctly	Quite imbalanced
9.	Shrikant Kokate et al.2020	RBF kernel method based SVM	Accuracy is about 96%	Multi-classifiers and meta-learning can be considered in future
10.	S P Maniraj et al.2019	Local outlier factor and Isolation forest	Accuracy is about 98%	Try to improve this model by adding different algorithms

11.	Huang Tingfei et al.2020	Oversampling method based on VAE	Easy to sample latent space for good data generation and interpolation.	If data is used as an image then generated images are blurry.
12.	Mandeep Singh et al.2019	Hidden Markov Model	It helps to obtain a high fraud coverage combined with a low false alarm rate and fast detection.	High, expensive,Low accuracy,not scalable to large size of data sets.
13.	Aisha Barahim et al.2019	Decision Tree,naive Bayes,SVM	High flexibility,easy to implement,easy to display and to understand.	Requirements to check each condition one by one.
14.	Imane Sadgali et al.2020	Adaptive method	Efficient training	Low accuracy,not scalable to large size of data sets
15.	Arpit Jain et al.2020	KNN,Logistic regression,Random Forest,Decision Tree and Artificial Neural Network.	High speed in detection,portability,high accuracy	High processing time for large neural networks,poor explanation capability.
16.	Mohammad Gandhi et al.2020	Naive bayes,J48,deliver regression and adaboost.	Easy to implement,training period less	All predictors are independent,faces zero frequency problem
17.	Thirunavukkarasu et al.2021	Random forest algorithm	High accuracy	Computationally intensive for large data sets.
18.	Rahul Powar et al.2020	KNN	Very easy to implement for multi-class problem	Outlier sensitivity
19.	Heta Naik	Naive	Efficient	High data

	et al.2019	Bayes,Logistic regression,J48 and adaboost	computation,Model flexibility	maintenance,cant handle missing data
20.	C.Sudha et al.2017	KNN	Can be used for both classification and regression	Missing value treatment

2.2 SUMMARY

In this chapter, we have briefly discussed all the papers that we have surveyed. And gained knowledge about our ongoing project. We have keenly observed the limitations of the existing systems and we will try to overcome those limitations by applying efficient algorithms using different preprocessing techniques.

CHAPTER III

SYSTEM REQUIREMENTS

PREAMBLE

In this chapter, we are going to discuss the requirements of the system in order to develop the proposed methodology. By this, we will get to know the different tools we are going to use, the dataset we are going to consider, software and hardware requirements. All come together will achieve the successful implementation.

3.1 TOOLS USED

In order to build a system that will detect fraudulent transactions based on the past data. We require some software and hardware requirements to develop a successful fraud detection system.

- ➢ Software Requirements
  - ○ Python 3.8
  - ○ Google Colab
- ➢ Hardware Requirements
  - ○ Windows Operating system
  - ○ At Least 2GB of RAM is required

3.1.1 Google Colab

Colab is a free version of the jupyter notebook environment running wholly in the cloud. This allows us to train machine learning and deep learning models on CPUs, GPUs and TPUs. It has several features that helps us to edit documents the same way with the google docs. This supports many popular and high-level machine learning libraries which can be easily loaded in the notebook. The most popular AI frameworks were made by Google called TensorFlow and also a development tool called Collaboratory. Collaboratory is also known as Google Colab or Collab.

We can be able to share the code as well with viewing, editing and reading as well. It is a comfortable platform for any ML project related work that can be done very easily and it can be useful for students as well. It is very easy to learn about the tools provided by the Google Colab. It is free of cost. Anyone can easily go through it without any payment. It provides for the code

as well. We can able to save the code in google drive also that means it can be accessible in offline mode.

Features of Google Colab

- ➢ Write and execute code in python.
- ➢ Free Cloud service with free GPU and TPU.
- ➢ Import data from google drive.
- ➢ Import external datasets.

3.1.2 Python

Python is a general-purpose interpreted, interactive, object-oriented and high-level programming language. It was created by Guido van Rossum and released in 1991. Like Perl, python source code is also available under the GNN General Public License(GPL).

Python is a high-level, interpreted, interactive and object-oriented scripting language. It is designed to be highly readable. It uses english keywords frequently whereas other languages use punctuations and it has fewer syntactic constructions than other languages.

It is one of the powerful languages which is easy to learn and code. Anyone having minimum knowledge in english can easily code with python language. It is a user-friendly, robust and portable programming language. It doesn't need any separate compiler like any other language. While running the code, it will compile itself without need of any compiler.

Python is a must for students and working professionals to become a great software engineer specially when they are working in Web Development Domain. The key advantages of learning Python are:

Python is Interpreted: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to Perl and PHP.

Python is Interactive: You can actually sit at a python prompt and interact with the interpreter directly to write your programs.

Python is Object-Oriented: Python Supports Object-Oriented style or technique of programming that encapsulates code within objects.

Python is a Beginner's Language: Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Characteristics of Python:

- ➢ It supports functional and structured programming as well as OOP.
- ➢ It can be used as a scripting language or can be compiled to byte-code for building large applications.
- ➢ It provides very high-level dynamic data types and supports dynamic type checking.
- ➢ It supports automatic garbage collection.
- ➢ It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

3.2 DATASET

The dataset we are going to use is available at kaggle website. It consists of 284,807 credit card transactions made by the credit card owners in September 2013 in Europe. Out of 284,807 transactions in the dataset, 492 were fraud transactions. The dataset includes 31 features. The first 28 features i.e., V1 to V28 are the principal components obtained using principal component analysis (PCA). The basic reason is to maintain privacy. Time and Amount are the two features that are not transformed using PCA. And the last feature is Class, which is a classification variable. It consists of two values 0,1. 0 represents the normal transaction whereas 1 represents the fraud transaction.

It is well known that the performance of various machine learning methods decreases when the analyzed dataset is unbalanced. So we are using a Random forest

classifier for better performance. Because it uses Bagging technique, for splitting the dataset into subsets for building various decision trees. To obtain more accurate results, a cross validation procedure is used to train and test the model in a subset of the dataset. Then the average of all the noted metrics is calculated over the dataset.

3.3 SUMMARY

In this Chapter, We gathered all the information about our project requirements. And we have discussed briefly about the tools we are going to use, the dataset we are going to consider and finally mentioned software requirements and minimum hardware requirements we require to implement the model.

CHAPTER IV

SYSTEM DESIGN

PREAMBLE

In this chapter, we are going to discuss the various UML diagrams in our project's view. It will define the components, modules, interfaces, and data for a system to satisfy specified requirements in the previous chapter. This chapter helps to understand the type of input data and how the output can be produced. By this, we will get to know how our proposed system will look like and what are the modules required to implement the system.

4.1 ARCHITECTURE DIAGRAM

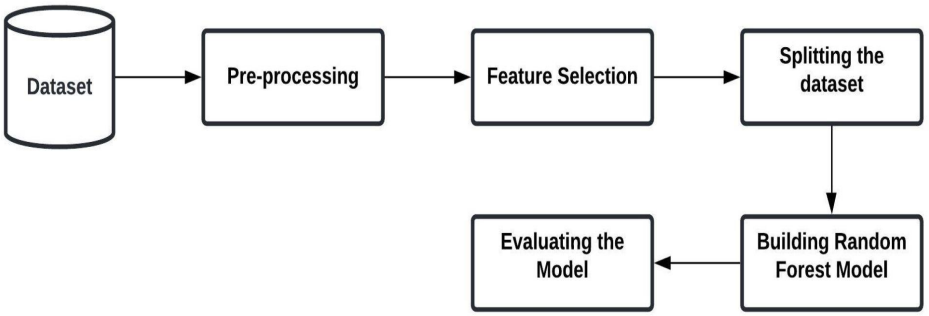


Fig 4.1 Architecture diagram

An architecture diagram is the conceptual model that defines the structure, behavior and more views of a system. In our architecture diagram, we have represented the steps in the implementation of our project in order to know how the system will look after successful implementation. This is useful for the visual representation of our project. By this, we will get to know the outlook of the system architecture. The architecture diagram defines the techniques, algorithms, modules, steps we have to follow inorder to implement the system.

In other words, an architectural diagram is a diagram of a system that is used to abstract the overall outline of the software system and the relationships, constraints and boundaries between components. It is important as it provides an overall view of the physical deployment of the software system and its evolution roadmap.

4.2 USE CASE DIAGRAM

Fig 4.2 Use case diagram

Use case diagrams are used to represent the dynamic behavior of the system. It encapsulates the functionality of the system by merging use cases, actors, and their relationships. It models the tasks, services and functions required by the system/subsystem of the application. It describes the advanced functions of the system and also tells the user how to deal with the system.

Elements of the use case diagram are:

Actor:

An actor describes an entity or entities that perform certain roles in a given system. The different roles represented by the participants are the actual business roles of the users in a given system. Participants in the use case diagram interact with the use case.

Use case:

Use cases describe the interactions between participants and IT systems during the execution of the business process. A use case represents a part of the function of an IT system and enables users to access this function.

4.3 CLASS DIAGRAM

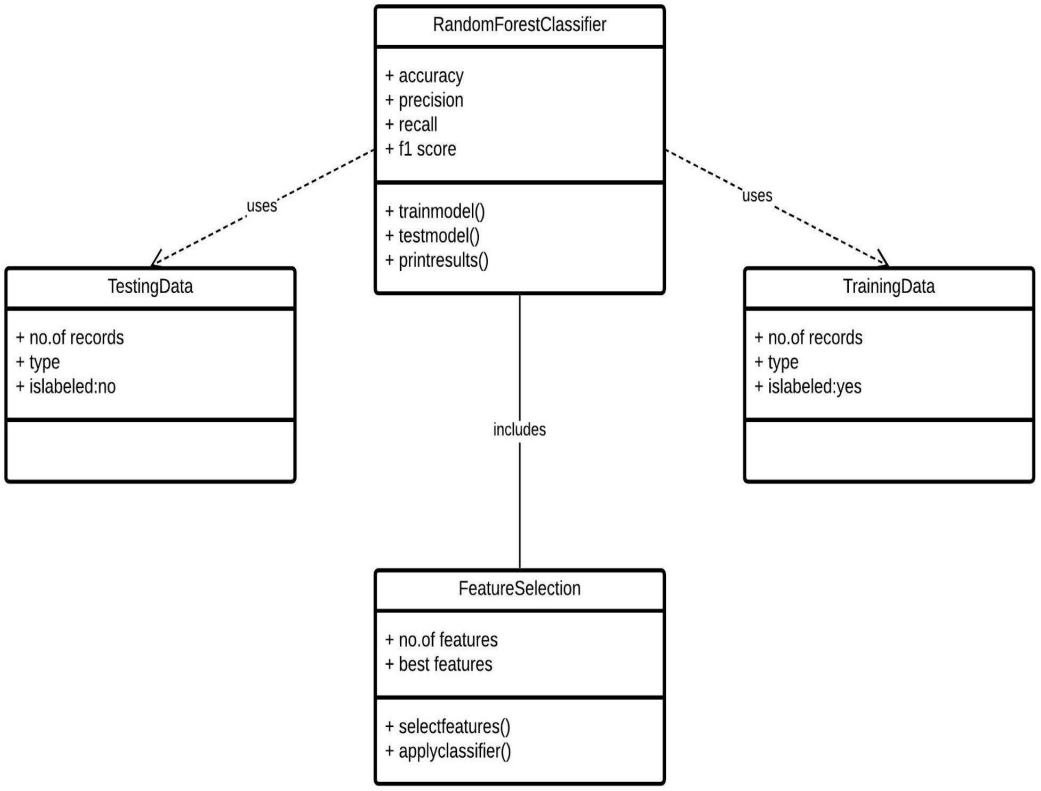


Fig 4.3 Class diagram

Classes in UML diagrams are the blueprint for creating objects or sets of objects. Class defines what an object can do. It is a template for creating various objects and implementing their behaviour in the system. A class in UML is represented by a rectangle that includes rows with the class names, attributes and operations. Class diagrams illustrate the data models of even very complex information systems. The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a structural diagram.

The purpose of the class diagram is to model the static view of the application. Class diagrams can be directly mapped with the object oriented languages.

Essential elements of UML class diagram are:

1. Class Name
2. Attributes
3. Operations

4.4 BEHAVIOUR DIAGRAM

4.4.1 Sequence Diagram

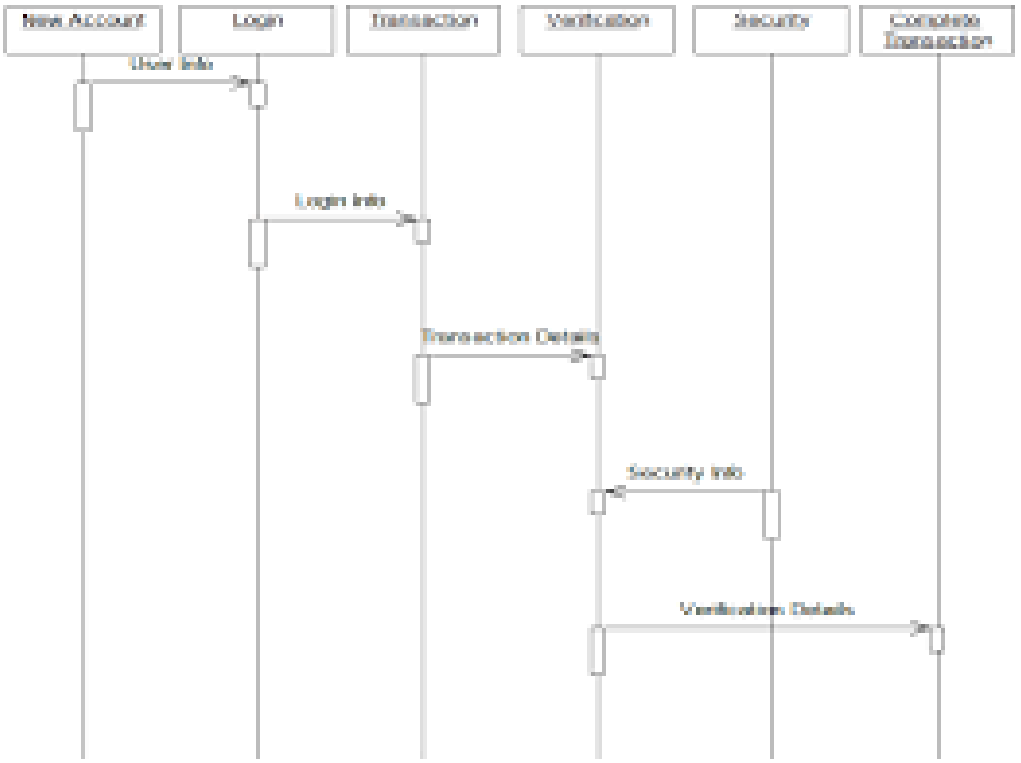


Fig 4.4 Sequence diagram

A sequence diagram is a type of interaction diagram because it describes how and in which order a group of objects work together. Sequence diagram represents the flow of messages in the system. Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. Sequence diagrams are also known as event diagrams, timing diagrams or event scenarios.

Benefits of sequence diagrams are:

- ➢ Represents the details of a UML use case.

- ➢ Model the logic of a complex process, function, or operation.
- ➢ Understand how objects and collaborations interact with each other to complete a process.
- ➢ Plan and understand the detailed features of existing or future scenarios.

4.4.2 Collaboration Diagram

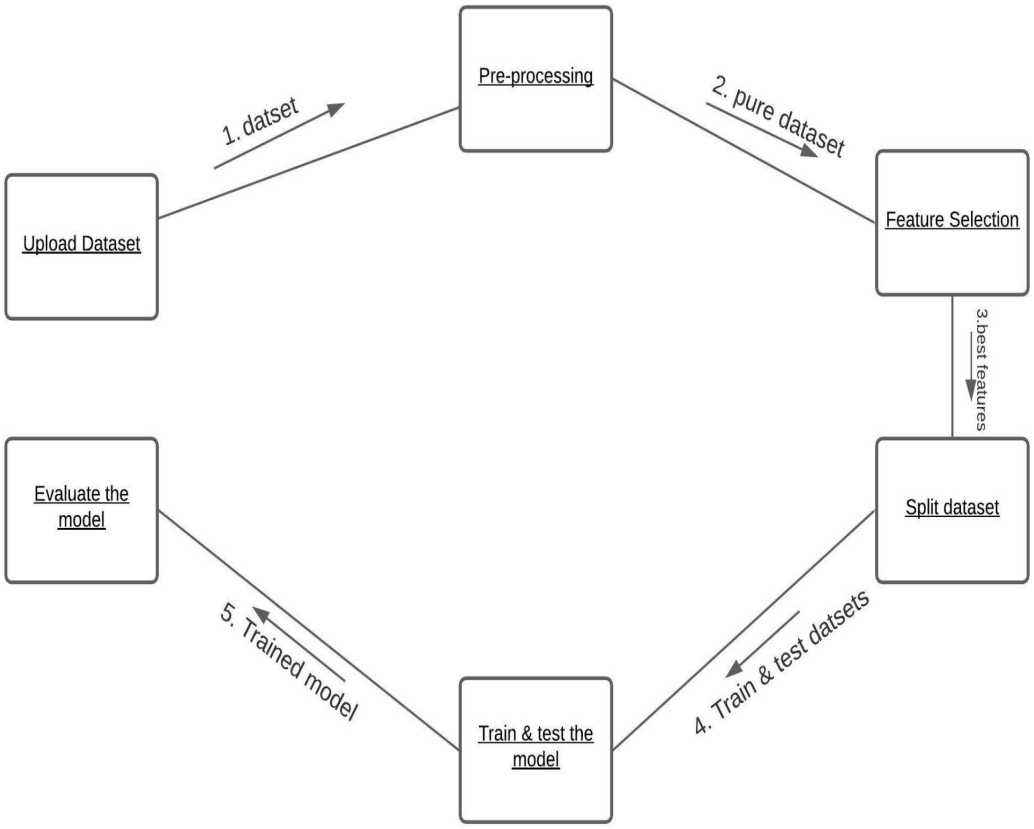


Fig 4.5 Collaboration diagram

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

Components of a collaboration diagram are:

Objects:

The representation of an object is done by an object symbol with its name and class underlined, separated by a colon.

Actors:

In the collaboration diagram, the actor plays the main role as it invokes the interaction. Each actor has its respective role and name. In this, one actor initiates the use case.

Links:

The link is an instance of association, which associates the objects and actors. It portrays a relationship between the objects through which the messages are sent. It is represented by a solid line. The link helps an object to connect with or navigate to another object, such that the message flows are attached to links.

Messages:

It is a communication between objects which carries information and includes a sequence number, so that the activity may take place. It is represented by a labeled arrow, which is placed near a link. The messages are sent from the sender to the receiver, and the direction must be navigable in that particular direction. The receiver must understand the message.

4.4.3 State Diagram

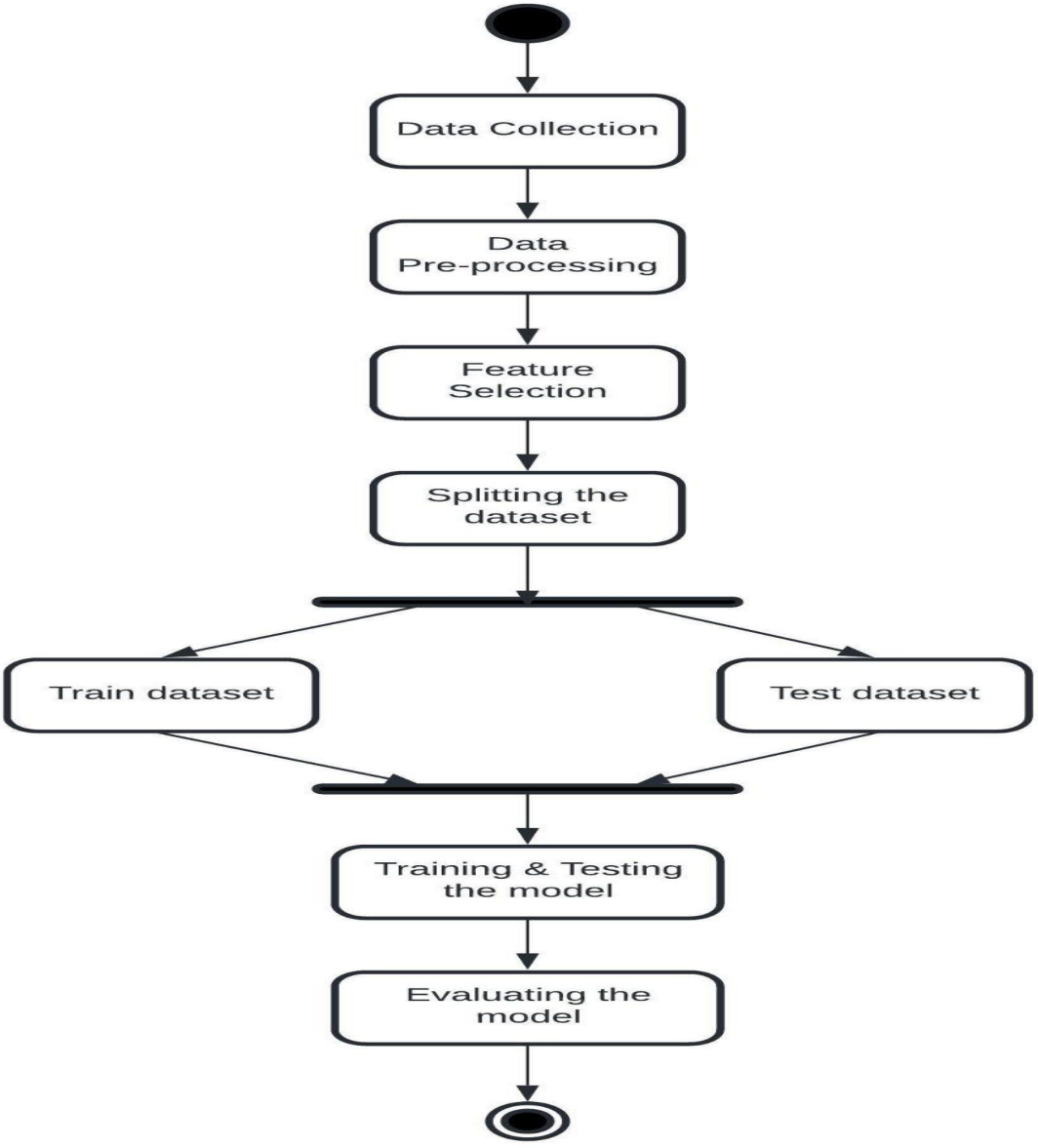


Fig 4.6 State diagram

A state diagram sometimes known as state machine diagram, statechart diagram or a state transition diagram, is a type of behavioural diagram that shows transitions between various objects. It is used to represent the condition of the system or part of the system at finite instances of time. It visualises a sequence of states that an object can assume in its lifecycle. It is used to describe the behavior of a system, subsystem, component or class.

Initial state

Black filled circle represents the initial state of a system.

Transition

Solid arrow represents the transition or change of state.

State

Rounded rectangle represents the state of an object of a class at an instant of time.

Final state

Filled circle within a circle represents the final state of the state diagram.

4.4.4 Activity Diagram

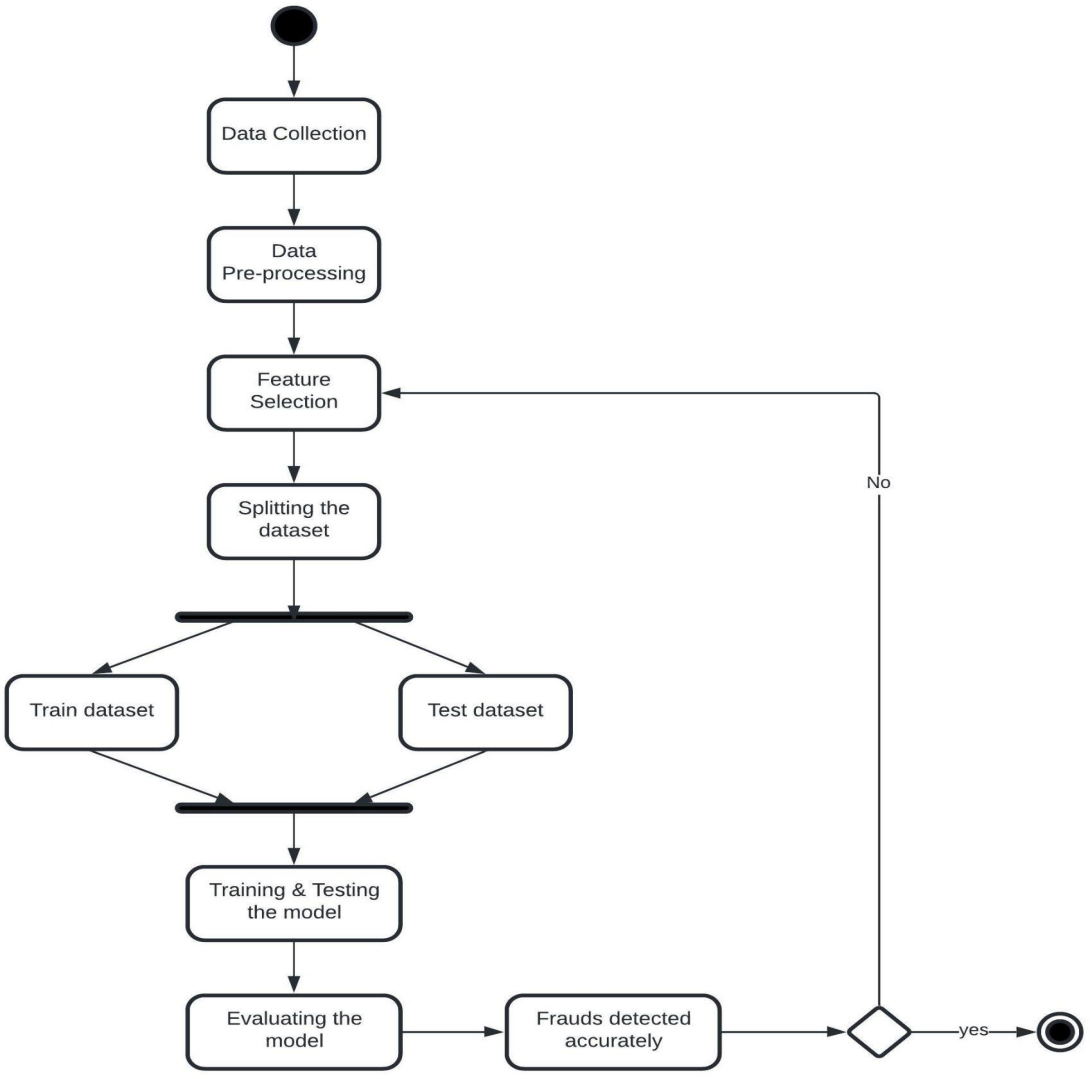


Fig 4.7 Activity diagram

An activity diagram is another important behavioral diagram in UML. It describes the dynamic behavior of the system. It is essentially an advanced version of flowchart that models the flow from one activity to another activity.

Initial state

A starting state before an activity takes place is observed by using the initial state and it is represented as a filled circle.

Action or Activity state

Any action or event that takes place is represented using an activity. It is represented as a rounded rectangle.

Action flow

Action flow is also represented as paths. These are used to show the transition from one activity to another activity.

Guards

A guard refers to a statement written next to a decision node on an arrow sometimes within square brackets.

Final state

The state which the system reaches when a particular process or activity ends is known as a Final state or End state. Filled circle within a circle represents the final state of the activity diagram.

4.5 IMPLEMENTATION DIAGRAM

4.5.1 Component Diagram

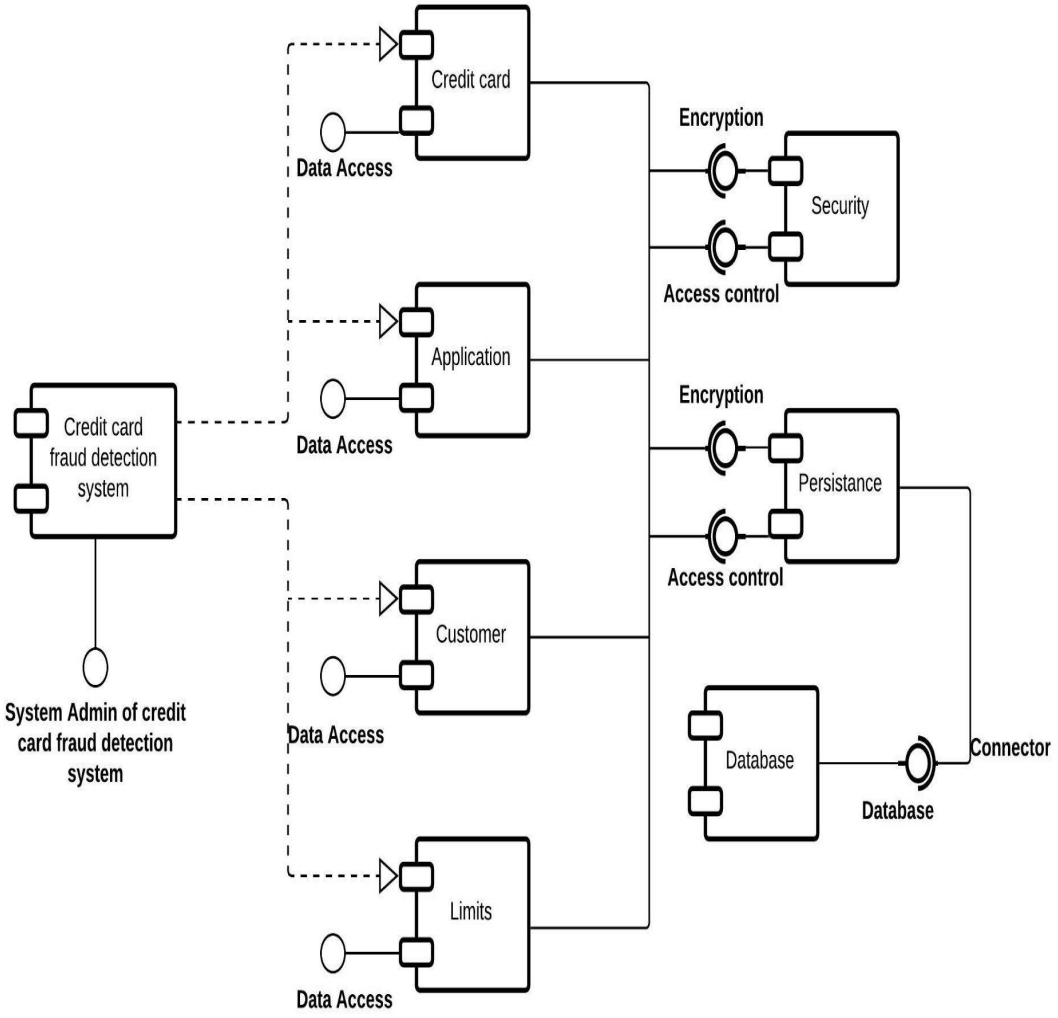


Fig 4.8 Component diagram

A component diagram is essentially a class diagram that focuses on a system's components that are often used to model the static implementation view of a system. The purpose of this diagram is to show the relationship between different components in a system. A component is a replaceable and executable piece of a system whose implementation details are hidden. It also requires an interface to carry out a function.

Port

A port is a feature of a classifier that specifies a distinct interaction point between the classifier and its environment. These are represented using a square along the edge of the system or a component. It is often used to expose required and provided interfaces of a component.

Interfaces

Interfaces in component diagrams show how components are wired together and interact with each other. An interface describes a group of operations required or provided by components. A full circle represents an interface **created or provided** by the component. A semi-circle represents a **required interface**.

4.5.2 Deployment Diagram

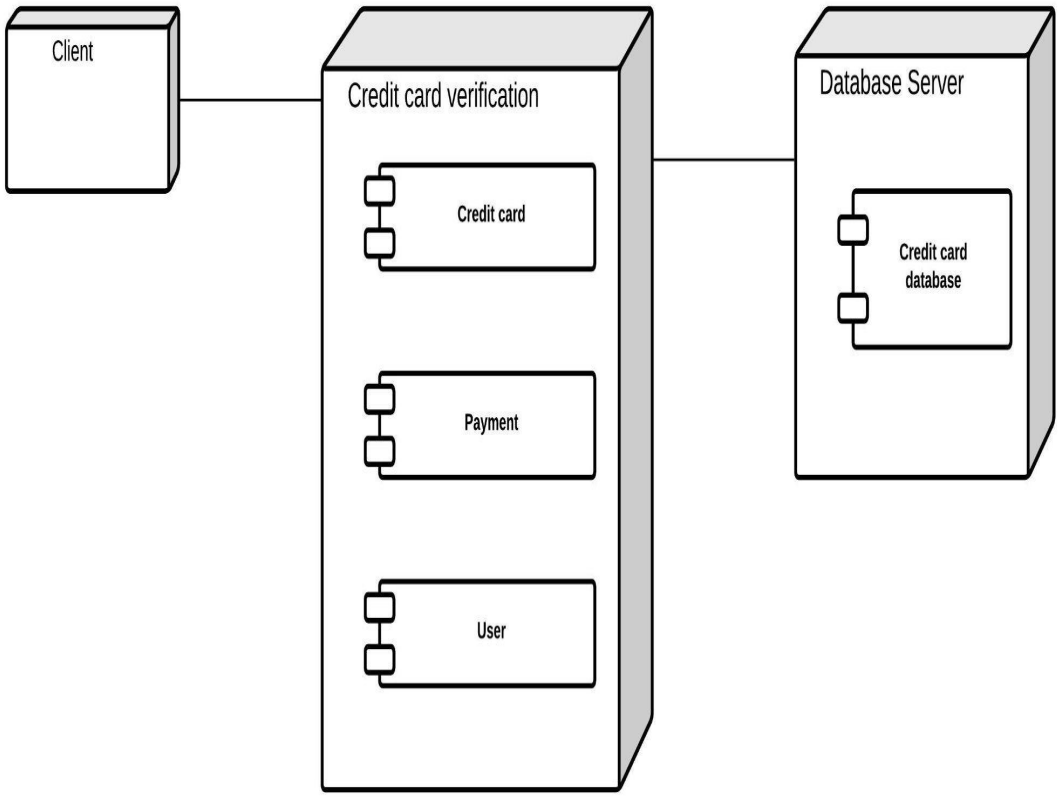


Fig 4.9 Deployment diagram

A deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them. It is a kind of structure diagram used in modelling the physical aspects of a system.

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

4.6 DATAFLOW DIAGRAM

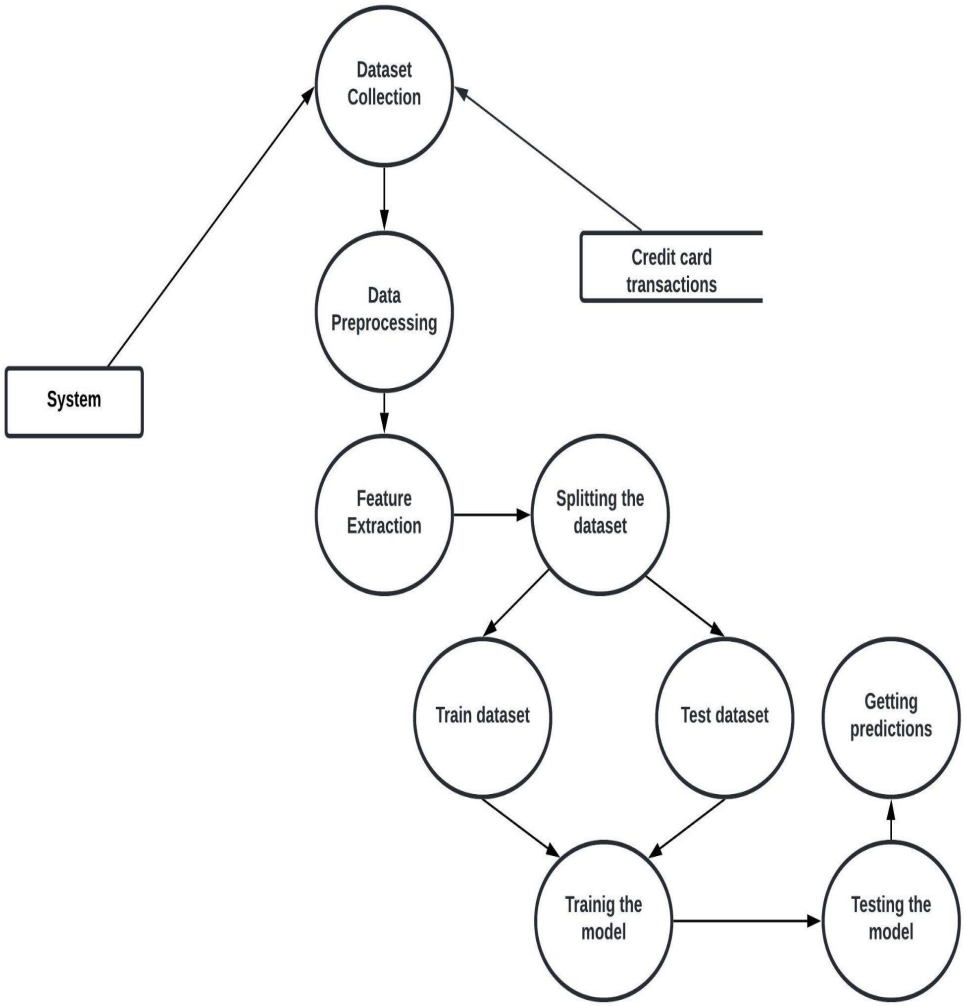


Fig 4.10 Data Flow diagram

A data flow diagram provides information about the outputs, inputs of each entity and the process itself. It has no control flow, there are no decision rules and no loops. It defines the flow of information for any process or a system. It shows how data enters the system, leaves the system, what changes the information and where data is stored. The DFD is also called a data flow graph or bubble chart.

Process

The process is a part of the system that transforms inputs to outputs. The symbol of a process is a circle, an oval, rectangle or a rectangle with rounded corners. The process is named in one word, a short sentence, or a phrase that is clearly to express its essence. Here we have four processes.

Data flow

Data flow shows the transfer of information from one part of the system to another. The symbol of the flow is the arrow. The flow should have a name that determines what information is being moved.

Warehouse

The warehouse or data store is used to store the data for later use. The symbol of the store is a side open rectangle as shown in the figure. Viewing the warehouse in DFD is independent of implementation. The flow from the warehouse usually represents the reading of the data stored in the warehouse, and the flow to the warehouse usually expresses data entry or updating.

Terminator

The terminator is an external entity that communicates with the system and stands outside of the system. The terminator may be another system with which the modeled system communicates.

4.7 SUMMARY

The Unified Modelling Language(UML) is a standard visual modelling language to document business processes. It is a general purpose, developmental, modelling language in the field of software engineering. In this chapter, we have discussed several UML diagrams belonging to our project. By that we came to know how our system will work. This chapter is considered to visualize the projects.

We have considered this chapter with the purpose of visually representing a system with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain or document information about the system. By this we will get to know the relationships among the various nodes and as well as actors and their roles in our project.

CHAPTER V

IMPLEMENTATION

PREAMBLE

In the last chapter, we get to know how the modules will be divided for the implementation of our project. So, In this chapter we will see how we can implement the methodology we have considered and what are the formulas we need to use in order to get the results effectively. We are briefly discussing about the methods that we are used to implement the system.

5.1 PROPOSED METHODOLOGY

Detection of Credit card frauds can be done in different ways using various strategies and different algorithms. In our project we are going to use a random forest algorithm for credit card fraud detection. Random forest algorithm is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both classification and regression models. It predicts the output with high accuracy, even for the large datasets it runs efficiently. It takes less training time as compared to other algorithms. It also prevents the overfitting issues whenever it occured. This algorithm provides more flexibility and it is easy to use. This methodology is expected to produce results with high accuracy, precision scores. Dataset plays a very important role in machine learning. The efficiency of the model depends on the type of the data that is used in the model. As our project includes the credit card details as it is a credit card fraud detection task, creating our own dataset is not possible. So the dataset used for the credit card fraud detection is from the kaggle.

Kaggle has provided the dataset that contains the transaction details. The dataset which is considered in our project contains 2,84,807 rows and 31 columns. Due to the confidentiality issue, the dataset consists of only the numerical values. Columns are labeled as V1, V2, …… , V28, Amount, Time, Class. For confidentiality, the real labels of the columns are not provided in the dataset and the whole dataset consists of numerical values as the original values. These are converted into PCA values except the value of amount and time. The ‘Class’ feature states

whether the instance of the dataset is a fraud transaction or a genuine transaction. Values for the class are either 0 or 1. 0 means genuine transaction/not fraud transaction and 1 means fraud transaction. There are a total 2,84,807 data instances which includes both fraud and genuine transactions.

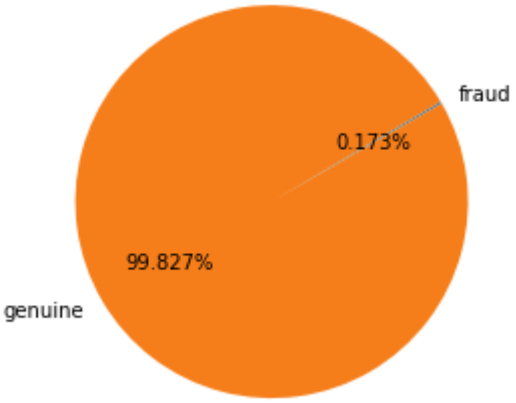


Fig 5.1 pie chart of dataset

From the above diagram we know that 99.827% of the dataset are genuine transactions and 0.173% of the transactions are fraudulent transactions.

	Time	V1	V2	V3	V4	V5	V6	V7	V8
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533

Fig 5.2 top five rows of dataset

The Fig 5.2 shows the top 5 rows of the dataset, it also shows few values of the features till V8. As seen in the figure all the values for the features are numerical values which PCA values due to confidentiality. Time feature in the dataset is time followed by V1 to V28 then Amount and finally Class.

After viewing the dataset, the next step was to know how the data is distributed and visualize the data. The fig 5.2 mean, max, in values that can be obtained by describe() function.

	Time	V1	V2	V3	V4	V5	V6	V7	V8
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05
mean	94813.859575	3.919560e-15	5.688174e-16	-8.769071e-15	2.782312e-15	-1.552563e-15	2.010663e-15	-1.694249e-15	-1.927028e-16
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00	1.380247e+00	1.332271e+00	1.237094e+00	1.194353e+00
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00	-1.137433e+02	-2.616051e+01	-4.355724e+01	-7.321672e+01
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01	-7.682956e-01	-5.540759e-01	-2.086297e-01
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02	-2.741871e-01	4.010308e-02	2.235804e-02
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01	3.985649e-01	5.704361e-01	3.273459e-01
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01	7.330163e+01	1.205895e+02	2.000721e+01

Fig 5.3 describe() for dataset

We can say that the credit card fraud dataset is highly imbalanced as shown in fig 5.3 the number of genuine transactions is very high compared to the fraudulent transactions.

```
Genuine transactions: 284315
Fraudulent transactions: 492
```

5.2 PROPOSED METHODOLOGY MODULE

1. Data collection
2. Data Pre-processing
3. Feature Selection
4. Splitting the dataset
5. Training and testing the model
6. Evaluating the model

5.3 DATA COLLECTION

In this module we need to collect the dataset by importing a package in the python programming language. The dataset we have considered is available at kaggle. Collecting the data for training the random forest model is the basic step in the ML pipeline. The predictions made by the ML systems can only be as good as data on

which they have been trained. There are some problems that can arise while collecting the dataset. They are as follows:

- ➢ Inaccurate data
- ➢ Missing data
- ➢ Data Imbalance
- ➢ Data Bias

To avoid these problems, we have considered a pre-cleaned, freely available dataset. That means a clean, pre-existing, properly formulated dataset is considered here.

5.4 DATA PREPROCESSING

After collecting the data successfully, then we need to pre-process the data into a format so that the random forest classifier can use it for training. In our methodology, we are using a pre-processing technique called **Data Cleaning** is an automated technique in every ML model that will remove incorrectly added or classified data. In the data cleaning, we are going to handle the missing values. It doesn't really matter whether it is a regression, classification or any kind of problem, no model can handle NULL or NaN values on its own so we need to do that.

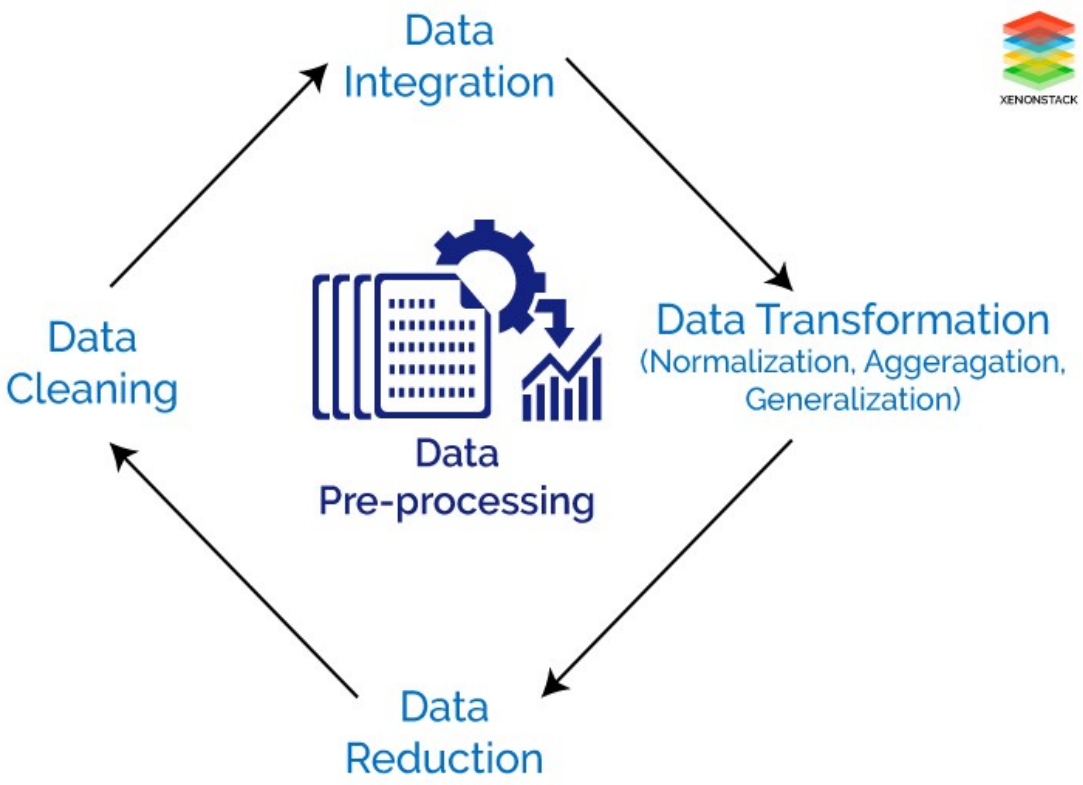


Fig 5.4 Pre-processing the data

First we need to check whether the dataset consists of missing values or not by using isna() method in python pandas library. From that we will get the number of missing values we are having in the dataset. Then we need to identify where the missing values exist in the dataset by using isna().sum() method in python. From that, it will return the column names along with the number of NaN values present in that column. In our dataset, which we have considered didn't have any null values.

5.5 FEATURE SELECTION

After preprocessing the data successfully, we need to select the best features for training the random forest model. Feature selection can be done using Random Forest comes under the category of embedded methods. It combines the qualities of both wrapper and filter methods. These are implemented by algorithms that have their own built-in feature selection methods. There are some benefits of using these embedded methods:

- ➢ Highly accurate
- ➢ Generalize better

➢ Interpretable

First of all, Random Forest builds 4-12 hundred decision trees. Each of them is built over a random extraction from the dataset and random extraction of the features. Not every tree uses all the features or observations this guarantees that the trees are de-correlated and therefore less prone to overfitting. At each node, the tree divides the dataset into two bags, each of them hosting observations that are more similar among themselves and different from the ones in the other bag. Then the importance of each feature is calculated. Before that we have to calculate the importance of each node in a particular bag.

For each decision tree, Scikit-learn calculates the importance of each node using Gini importance, assuming two child nodes (binary tree):

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)}$$

- ni_j - the importance of a node j
- W_j - the weighted number of samples reaching node j
- C_j - the impurity value of node j
- left(j) - the child node from left split on node j
- right(j) - the child node from right split on node j

The importance of each feature is then calculated as follows:

$$fi_i = (\sum_{j:\text{node j splits on feature i}} ni_j) / \sum_{k \in \text{all nodes}} ni_k$$

- fi_i - the importance of feature i
- ni_j - the importance of node j

These can be normalized to a value between 0 and 1 by dividing by the sum of all feature importance as:

$$normfi_i = fi_i / \sum_{j \in \text{all features}} fi_j$$

The final feature importance, at the Random Forest level, is it's average over all the trees. The sum of the features importance value on each trees is calculated and divided by the total number of trees:

$$RFfi_i = (\sum_{j \in \text{all trees}} normfi_{ij}) / T$$

- $RFfi_i$ - the importance of feature i calculated from all trees in the

Random Forest model

- $normfi_i$ - the normalized feature importance for i in tree j
- T - the total number of trees

In this way, the importance of each feature will be calculated to select the best features for training the model.

5.6 SPLITTING THE DATASET

After successful completion of selecting the best features to evaluate the Random Forest model, we need to split the dataset into train data and test data to train and test the performance of the Random Forest model. For that purpose, we are going to use the train-test split technique for evaluating the performance of the model. In this procedure, firstly we need to split the dataset into two subsets. The first subset is used for fitting the model and is referred to as the training dataset. The second subset is not used for training the model, instead the input element of the dataset is provided to the model, then predictions are made and compared to the expected values. From this we will get to know that the second part is referred to as the test dataset.

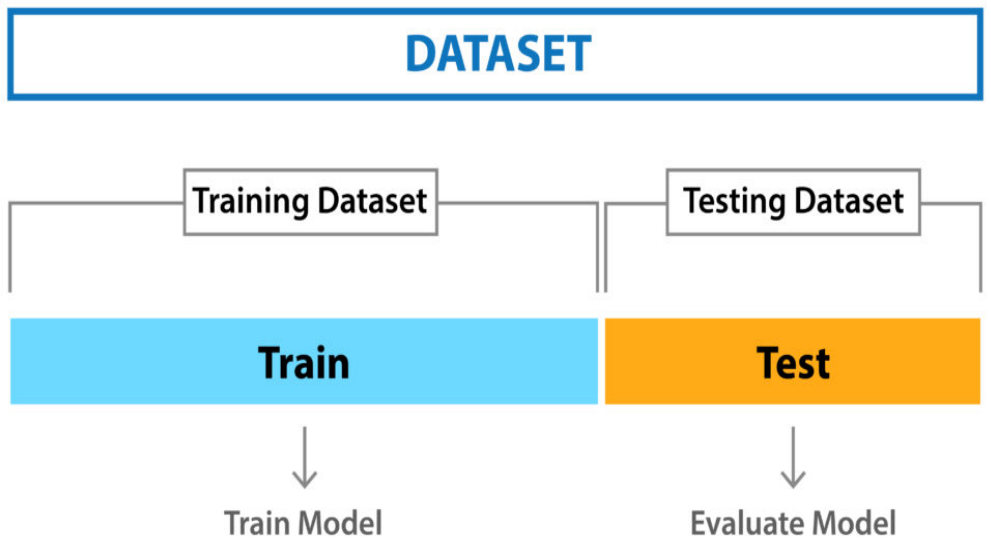


Fig 5.5 Dataset splitting

Train dataset is used to fit the Random Forest model. Whereas Test dataset is used to evaluate the fit Random Forest model. Here we split 80% of the dataset as training dataset and 20% of dataset as test dataset.

5.7 TRAINING AND TESTING THE MODEL

After splitting the data into training dataset and testing dataset, In this module we have to train the model on the dataset by using the random forest algorithm. This algorithm is used to combine the hundreds of decision trees and then trains each of the decision trees on a different sample of the observations. The final predictions of the random forest are made by averaging the predictions of each individual decision tree. The individual decision trees tend to overfit the training data but random forest can mitigate that issue by averaging the prediction results from different trees.

Once the training test is ready, we can import the RandomForestClassifier Class and fit the training set to our model. The class svc is assigned to the variable classifier. The criterion used here is “entropy”. The other criterion that can also be used is the “gini”. The classifier.fit() function is then used to train the model.

From sklearn.ensemble import RandomForestClassifier classifier =
RandomForestClassifier(n_estimators = 10, criterion = ’entropy’)
classifier.fit(X_train, Y_train)

Here n_estimators means The required number of trees in the random forest. Here we have taken the default value as 10.

Criterion means it is a function to analyze the accuracy of the split. Here we have taken entropy for the information gain.

After training the model we have to test the model on the dataset. The test set is a set of observations used to evaluate the performance of the model using some performance metric. No observations from the training set are included in the test set. Testing the model is nothing but measuring the accuracy of the model. Here we have used the train_test_split function to split 80% of our data into training dataset and remaining 20% of our data as to the test dataset. After training and testing the model we have to evaluate the model based on the best accuracy.

5.9 EVALUATING THE MODEL

After completion of the training and testing the model we have to evaluate the model based on the classification predictions. These model predictions are produced based on the class with highest probability, which is in a binary classification problem as we take 0 and 1. If 0 means genuine transaction and 1 means it is a fraud transaction. Based on the accuracy, recall, precision and F1-measure we evaluate the model.

Accuracy: It is one of the metrics for evaluating the classification models. Informally, it is the fraction of predictions our model got right. Whereas formally it is defined as the Number of correct predictions by Total number of predictions. Here we can say, Number of fraud and normal transactions correctly predicted by Actual number of fraud and normal transactions.

Accuracy = (TP+TN)/(TP+TN+FP+FN)

Recall: It is sometimes referred to as sensitivity, is the fraction of retrieved instances among all relevant instances. A perfect classifier has a recall equal to 1. Here we can say, it is the fraction of Number of transactions correctly predicted as relevant by the Total number of transactions incorrectly predicted as irrelevant.

Recall = TP/(TP+FN)

Precision: It is the fraction of relevant instances among the retrieved instances. Here, we can say, it measures the percentage of transactions flagged as fraud and are correctly classified.

Precision = TP/(TP+FP)

F1-score: It is also called F-score or F-measure. It is the weighted average of precision and recall. That means, it takes both false positives and false negatives. Actually, it is not as easy to understand as accuracy, but it is usually more than accuracy, especially if you have an uneven class distribution.

F1-measure = 2*(Precision*Recall)/(Precision+Recall)

When performing classification predictions, these four types of outcomes could occur.

True Positives: These are when we predict an observation belongs to a class and it actually belongs to that class.

True Negatives: These are when we predict an observation does not belong to a class and it actually does not belong to that class.

False Positives: These occur when we predict an observation belongs to a class when in reality it does not.

False Negatives: These occur when we predict an observation does not belong to a class when in fact it does.

5.10 SUMMARY

In the chapter, we have given a plan on how we are going to put that plan into action and we have discussed the modules we are going to have in our project. By this, the coding will be simplified and the equations we have considered in order to get the accurate results are also discussed here. At the end we have evaluated the performance of the model based on the accuracy obtained, recall, precision and f1 scores are calculated to evaluate the performance of the Random Forest Classifier. This chapter summarizes the critical steps involved to put our solution into practice. It is the step-by-step procedure we need to follow while getting the solution into practice.

CHAPTER VI

EXPERIMENTAL RESULTS

6.1 RESULTS

The results obtained by using Random Forest Classifier are:

1. **Below diagram is the visual representation of fraud and valid transactions**

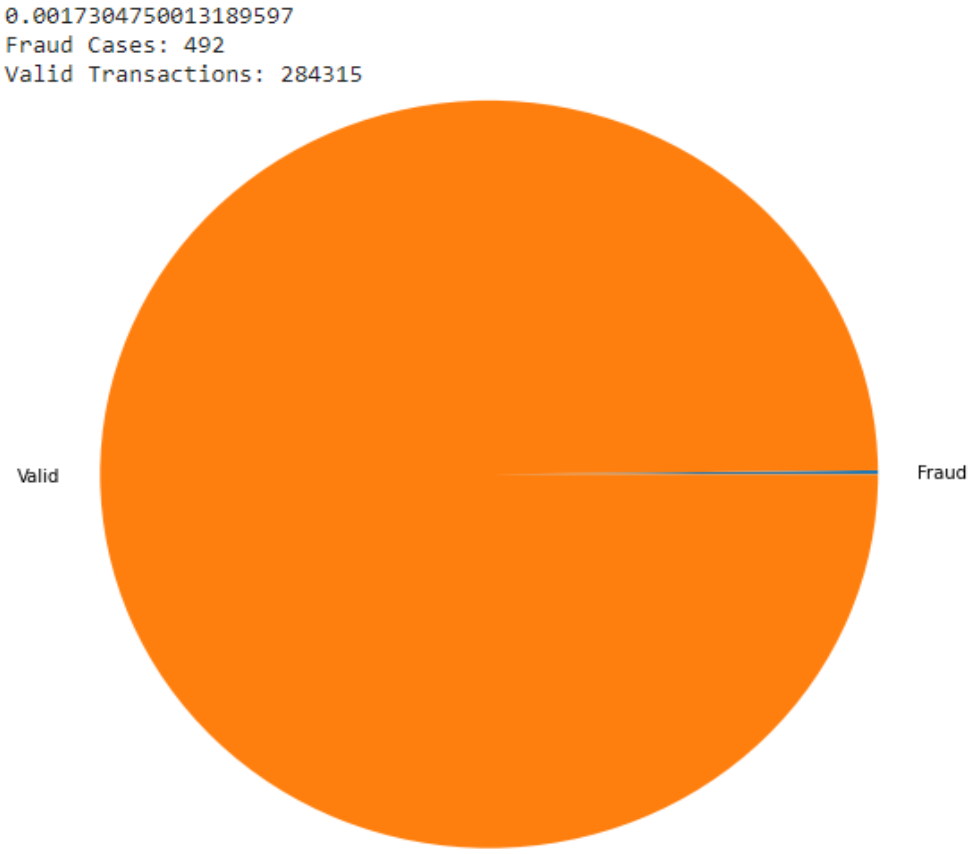


Fig 6.1 Pie chart showing the fraud and valid transactions

2. **Below diagram represents the visual representation of whole information about the dataset**

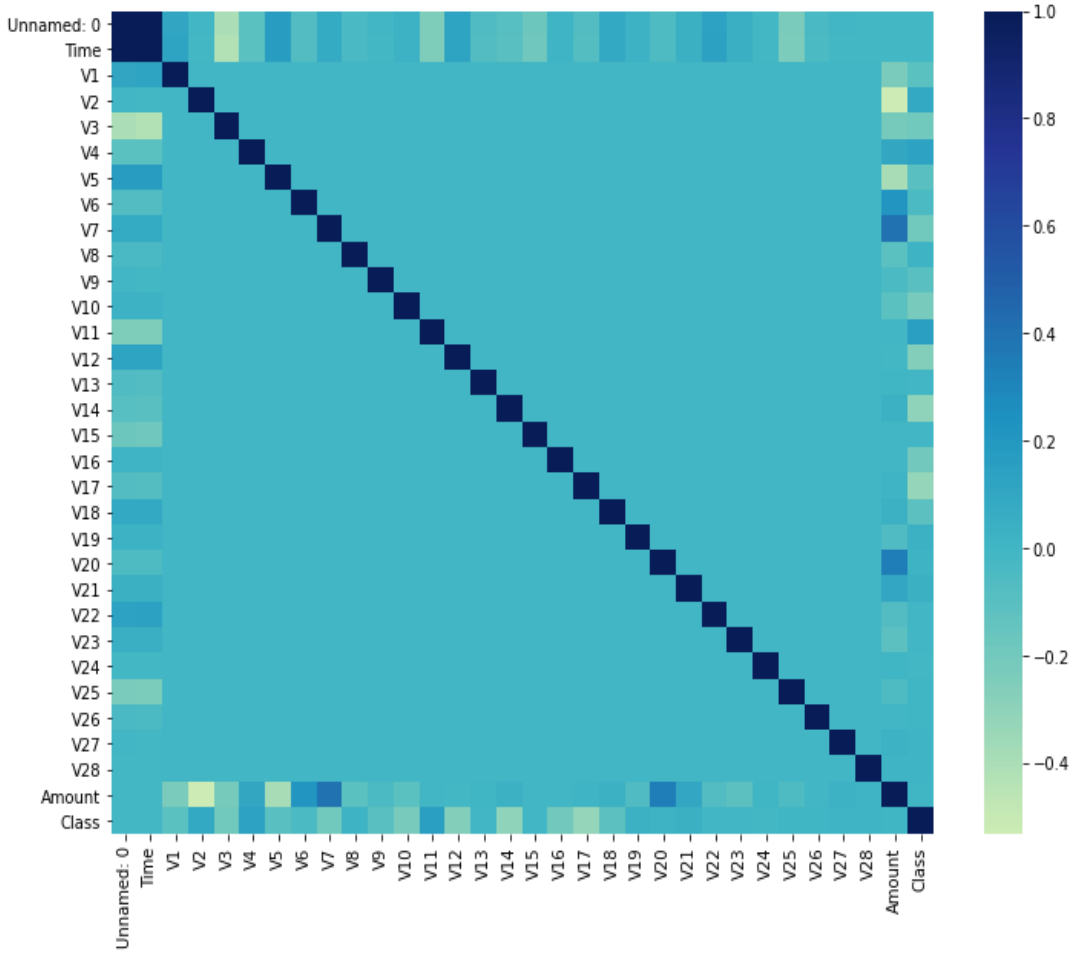


Fig 6.2 Heatmap showing the visual representation of whole dataset

CHAPTER VII

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

Here we have acquired the result of an accurate value of credit card fraud detection i.e., 0.9995786664794073 (99.95%) using a random forest algorithm. After feature selection and fine-tuning, the parameters and the performance of the model was increased upto 15-20%. In comparison to existing systems, this proposed system is applicable for larger datasets and provides more accurate results. In our project we estimate that the accuracy of our project will be 97% but after using the random forest algorithm it has given 99.95% accuracy, 91.57% precision, 81.72% recall and 86.36% F1-score. By our proposed algorithm i.e., Random Forest algorithm, it is concluded that it will provide better performance for larger training dataset but it is little bit slow while testing the model, it still suffers little.

7.2 FUTURE ENHANCEMENT

For future work, the efficiency of the model can be improved if the dataset is larger and balanced. If the original values of the dataset are known, then we can know how the data is correlated and which features are really important and train accordingly. In future different methods can be used to improve the results, more parameter tuning can be done. In addition, if we use online learning, it will enable rapid detection of fraud cases, potentially in real-time. This will help to detect and prevent fraudulent transactions before it happens. By this, we can reduce the number of losses incurred everyday in the financial sectors like banks, industries and big MNC companies also. In future we will try to add software applications to this by using different trending technologies like Deep Learning, Artificial Intelligence.

REFERENCES

- [1] Chu kiong loo: “Credit card fraud detection using AdaBoost and Majority Voting”, Proc. IEEE, Vol.06, pp.14277-14284, 2018.
- [2] M Suresh Kumar, V Soundarya, S Kavitha: “Credit card fraud detection using Random Forest algorithm”, Proc. IEEE, pp.149-153, 2019.
- [3] Hamzah Ali Shukur, Sefer Kurnaz: “Credit card fraud detection using Machine Learning methodology”, Proc. IJCSMC, Vol.08, No.03, pp.257-260, 2019.
- [4] Pratibha K, Varun Kumar K S, Vijaya Kumar V G: “Credit card fraud detection using Machine Learning algorithms”, Proc. IJERT, Vol.09, No.07, pp.1526-1530, 2020.
- [5] Lakshmi S V S S, Selvani Deepthi Kavila: “Machine Learning for credit card fraud detection system”, Proc. IJAER, Vol.13, No.24, pp.16819-16824, 2018.
- [6] Suresh K Shirgave, Chetan J. Awati, Rashmi More, Sonam S. Patil: “A review on Credit card fraud detection using Machine Learning”, Proc. IJSTR, Vol.08, No.10, 2019.
- [7] John O. Awoyemi, Adebayo O. Adetunmbi, Samuel A. Oluwadare: “Credit card fraud detection using Machine Learning Techniques: A comparative analysis”, Proc. IEEE, 2017.
- [8] Admel Husejinovic: “Credit card fraud detection using Naive Bayesian and C4.5 Decision tree classifiers”, Proc. PEN, Vol.08, No.01, pp.1-5, 2020.
- [9] Shrikant Kokate, Dr. C. M. Sheela Rani: “Fraudulent detection in credit card transactions using Radial Basis Function Kernel Method based on Support Vector Machine”, Proc. IJAST, Vol.29, No.12, pp.2557-2565, 2020.
- [10] S P Maniraj, Aditya Saini, Swarna Deep Sarkar, Shadab Ahmed: “Credit card fraud detection using Machine Learning and Data Science”, Proc. IJERT, Vol.08, No.09, pp.110-115, 2019.
- [11] Huang Tingfei, Cheng Guangquan and Huang Kuihua: “Using Variational Auto Encoding in Credit Card Fraud Detection”, Proc. IEEE, pp:149841-149853, Vol.8, 2020.
- [12] Mandeep Singh, Sunny Kumar and Tushant Garg: “Credit Card Fraud Detection Using Hidden Markov Model”, Proc. IJECS, pp.24878-24882, Vol.8, 2019.
- [13] Aisha Barahim,Amal Alharji and Norah Alasaibia: “Enhancing the Credit Card Fraud Detection Through Ensemble Techniques”, pp:1-8, Vol.16, 2019.

- [14] Imane Sadgali, Nawal Sael and Faouzia Benabbou: “Adaptive Model for Credit Card Fraud Detection”, pp.54-65, Vol.14, 2020.
- [15] Arpit Jain, Narendra Kr Sharma, Anshul Aggarwal and Love Aggarwal: “A Comparative Study on Credit Card Fraud Detection”, pp.54-65, Vol.14, 2020.
- [16] Mohammad Gandhi Babu,Pravin Kshirsagar and Boyini Mamatha: ”A Machine Learning Approach for Credit Card Fraud Detection”, pp.5237-5244, 2020.
- [17] M.Thirunavukkarasu, Achutha Nimisha and Adusumilli Jyothsna: “Credit Card Fraud Detection using Machine Learning”, pp.71-79, Vol.10, 2021.
- [18] Rahul Powar, Rohan Dawkhar and Pratichi: “Credit Card Fraud Detection Using Machine Learning”, pp.41-46, Vol.5, 2020.
- [19] Heta Naik and Prashasti Kanikar: “Credit Card Fraud Detection based on Machine Learning Algorithms”, pp.8-12, Vol.182, 2019.
- [20] C.Sudha and T.Nirmal Raj: “Credit Card Fraud Detection in internet using K-nearest Neighbor Algorithm”, pp.22-30, Vol.5, 2017.

APPENDIX A

SOURCE CODE

1. **Importing libraries required for implementation**

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split,cross_validate
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score
from sklearn.metrics import precision_score, recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import confusion_matrix
from sklearn.feature_selection import SelectKBest,f_classif
from google.colab import files
```

2. **Collecting the dataset**

```
uploaded = files.upload()
pd.read_csv("Creditcard.csv")
data=pd.read_csv("Creditcard.csv")
print(data.shape)
print(data.describe())
```

3. **Pre-processing the data**

```
data.isna().sum()
data.head()
fraud = data[data['Class'] == 1]
valid = data[data['Class'] == 0]
```

```
f=len(data[data['Class'] == 1])
v=len(data[data['Class'] == 0])
outlierFraction = len(fraud)/float(len(valid))
print(outlierFraction)
print('Fraud Cases: {}'.format(len(data[data['Class'] == 1])))
print('Valid Transactions: {}'.format(len(data[data['Class'] == 0])))
fig=plt.figure()
plt.pie([f,v],labels=['Fraud','Valid'],radius=2.5)
plt.show()
data[data['Class']==1]
print('Amount details of the fraudulent transaction')
fraud.Amount.describe()
print('Amount details of the valid transaction')
valid.Amount.describe()
fig = plt.figure(figsize = (12, 9))
sns.heatmap(data.corr(), cmap="YlGnBu",annot=False,center=0)
plt.show()
X = data.drop(['Class'], axis = 1)
Y = data["Class"]
print(X.shape)
print(Y.shape)
```

4. Selecting the best features and identifying the bad features

```
k_best=SelectKBest(f_classif,k=10)
k_best
k_best.fit(X,Y)
mask=k_best.get_support()
not_mask=np.logical_not(mask)
all_features=np.array(list(X))
best_features=all_features[mask]
bad_features=all_features[not_mask]
print("best_features=",best_features)
print("bad_features=",bad_features)
X.drop(bad_features,axis=1,inplace=True)
```

5. **Visual representation of best and bad features**

```
def plots(features, data):
    plt_index=0
    plt.figure(figsize=(10,10))
    plt.subplots_adjust(top=0.99,bottom=0.01, hspace=1.5, wspace=0.4)
    for feature in features:
        plt_index+=1
        feature_data=pd.concat([data[feature], data['Class']], axis=1)
        fraud=feature_data[data['Class']==1]
        genuine=feature_data[data['Class']==0]
        if len(genuine>10000):
            genuine=genuine[::100]
        plt.subplot(5,5,plt_index)
        sns.distplot(fraud[feature])
        sns.distplot(genuine[feature])
        plt.title(feature)
plots(bad_features,data)
plots(best_features,data)
```

6. **Splitting the dataset**

```
x_train, x_test, y_train, y_test = train_test_split(X,Y, test_size = 0.2, random_state = 12)
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

7. **Training and Testing the Random Forest Classifier**

```
rfc = RandomForestClassifier()
rfc.fit(x_train, y_train)
y_pred = rfc.predict(x_test)
```

8. Evaluating the performance of Random Forest Classifier

```
acc = accuracy_score(y_test, y_pred)
print("The accuracy is {}".format(acc))

prec = precision_score(y_test, y_pred)
print("The precision is {}".format(prec))

rec = recall_score(y_test, y_pred)
print("The recall is {}".format(rec))

f1 = f1_score(y_test, y_pred)
print("The F1-Score is {}".format(f1))
```

9. Generating Confusion matrix for fraud and normal transactions

```
LABELS = ['Normal', 'Fraud']
conf_matrix = confusion_matrix(y_test, y_pred)
plt.figure(figsize =(9,9))
sns.heatmap(conf_matrix, xticklabels = LABELS, yticklabels = LABELS, annot = True, fmt ="d");
plt.title("Confusion matrix")
plt.ylabel('True class')
plt.xlabel('Predicted class')
plt.show()
```

APPENDIX B

SCREENSHOTS

```
(284807, 32)
          Unnamed: 0           Time  ...         Amount          Class
count  284807.000000  284807.000000  ...  284807.000000  284807.000000
mean   142404.000000   94813.859575  ...      88.349619       0.001727
std     82216.843396   47488.145955  ...     250.120109       0.041527
min         1.000000       0.000000  ...       0.000000       0.000000
25%     71202.500000   54201.500000  ...       5.600000       0.000000
50%    142404.000000   84692.000000  ...      22.000000       0.000000
75%    213605.500000  139320.500000  ...      77.165000       0.000000
max    284807.000000  172792.000000  ...   25691.160000       1.000000

[8 rows x 32 columns]
```

DATASET SHAPE

```
data.isna().sum()

Unnamed: 0    0
Time          0
V1            0
V2            0
V3            0
V4            0
V5            0
V6            0
V7            0
V8            0
V9            0
V10           0
V11           0
V12           0
V13           0
V14           0
V15           0
V16           0
V17           0
V18           0
V19           0
V20           0
V21           0
V22           0
V23           0
V24           0
V25           0
V26           0
V27           0
V28           0
Amount        0
Class         0
dtype: int64
```

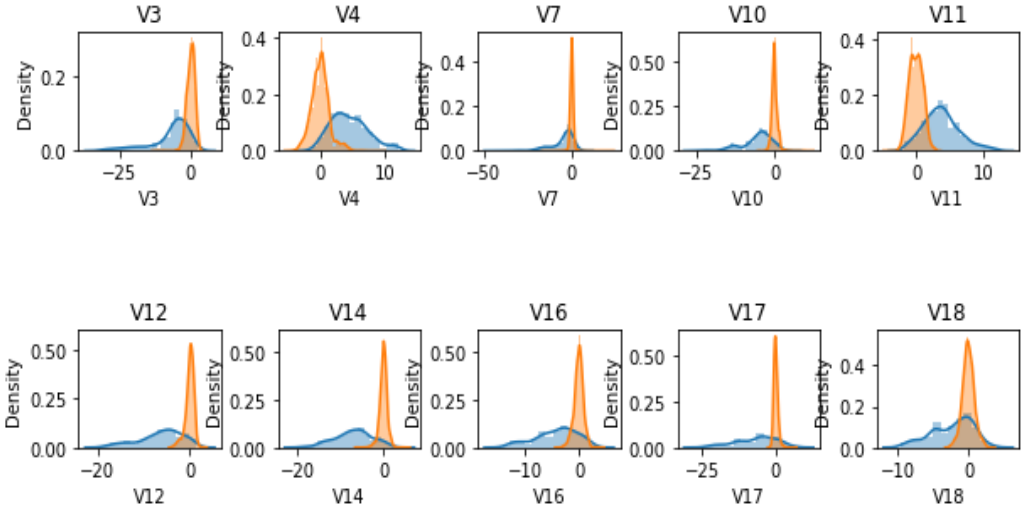
DISPLAYING THE COUNT OF NULL VALUES

```
Amount details of the fraudulent transaction
count     492.000000
mean      122.211321
std       256.683288
min         0.000000
25%         1.000000
50%         9.250000
75%       105.890000
max      2125.870000
Name: Amount, dtype: float64
```

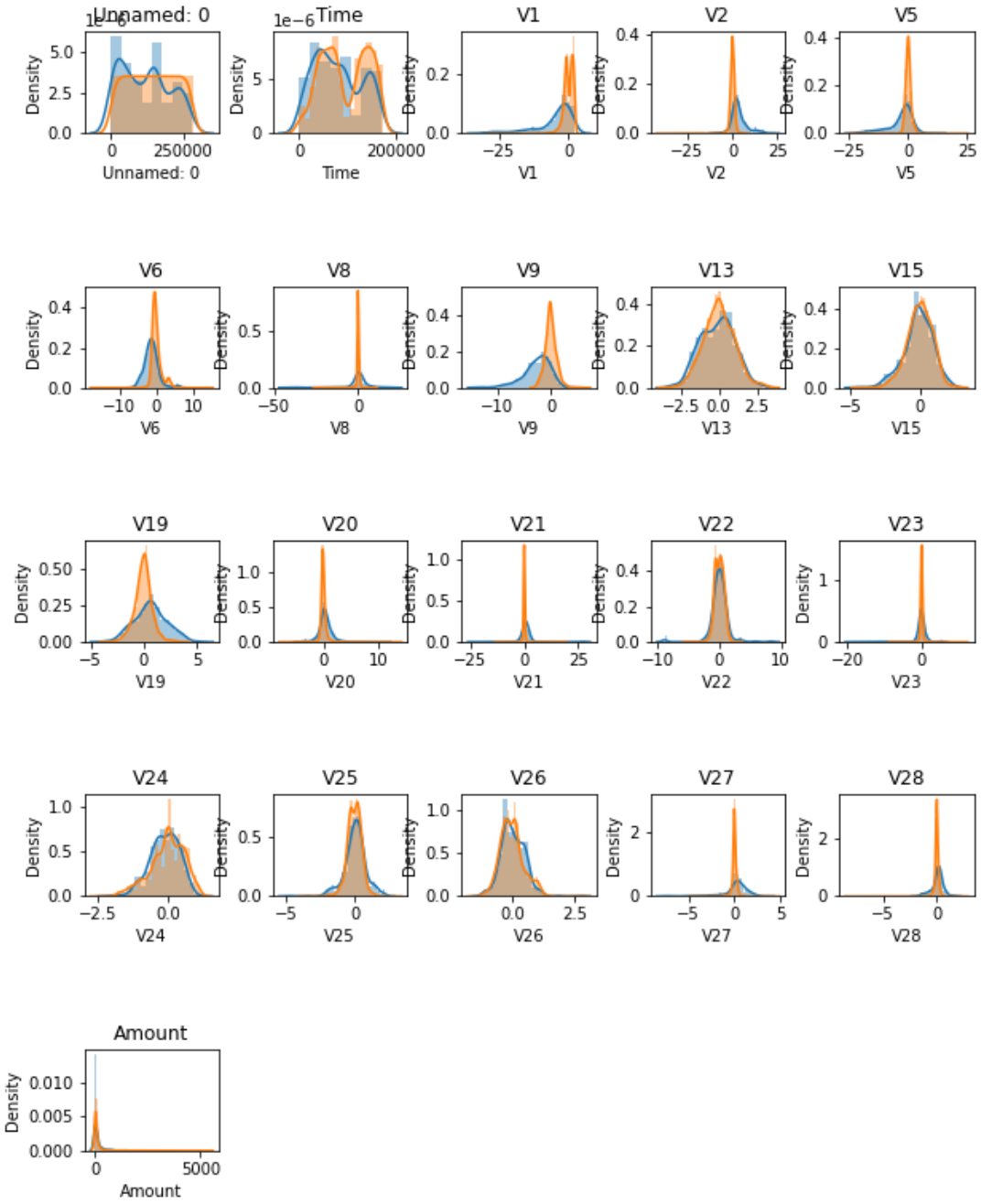
DETAILS OF FRAUD TRANSACTIONS

```
Amount details of the valid transaction
count    284315.000000
mean         88.291022
std         250.105092
min           0.000000
25%           5.650000
50%          22.000000
75%          77.050000
max       25691.160000
Name: Amount, dtype: float64
```

DETAILS OF VALID TRANSACTIONS



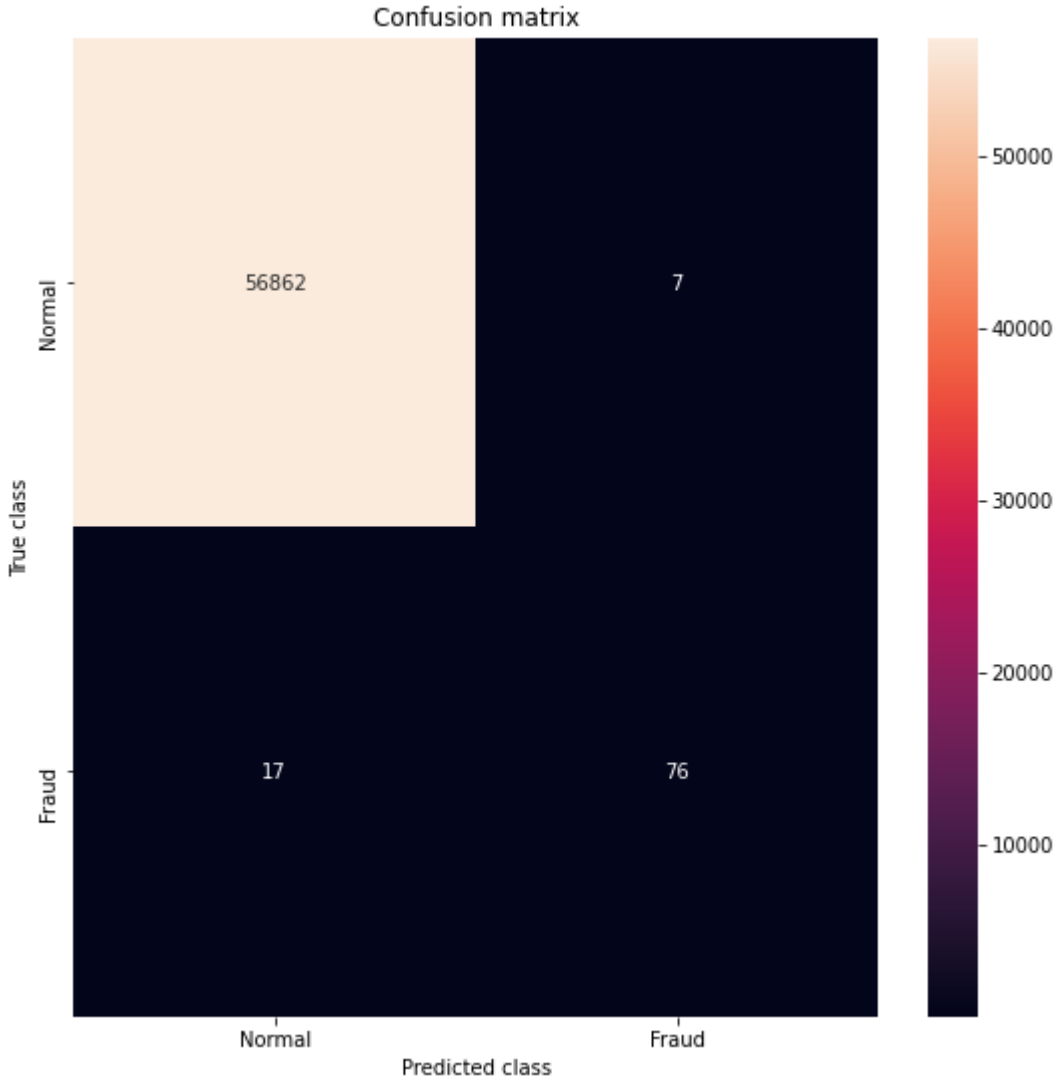
BEST FEATURES



BAD FEATURES

```
The accuracy is 0.9995786664794073
The precision is 0.9156626506024096
The recall is 0.8172043010752689
The F1-Score is 0.8636363636363636
```

RESULTS



CONFUSION MATRIX

APPENDIX C

STUDENT CONTRIBUTION

The below table represents the student contribution in this project

Tasks	G. Yasaswini (17K61A05D6)	K. Manjusha (17K61A0548)	T. Sai Avinash (17K61A05H2)
Literature Survey	✓	✓	✓
Problem Formulation	✓	✓	
Coding			✓
Documentation	✓	✓	✓
PPT Preparation	✓	✓	✓

APPENDIX D

POs, PSOs, PEOs and COs Relevance with Project

PROGRAM OUTCOMES (POs)

Pos	Program Outcomes	Relevance
PO1	**Engineering Knowledge:** Apply knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.	We have used several mathematical formulae that are related to our project work.
PO2	**Problem Analysis:** Identify, formulate, research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.	We have done literature surveys related to our project and have identified several problems in existing systems. Finally we solved two to three problems among them.
PO3	**Design/ Development of Solutions:** Design solutions for complex engineering problems and design system components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal and environmental considerations.	For the public safety we have provided a solution to the frauds occurring in credit card transactions to reduce financial loss.
PO4	**Conduct investigations of complex problems** using research based knowledge and research methods including design of experiments, analysis and interpretation of data and synthesis of information to provide valid conclusions.	
PO5	**Modern Tool Usage:** Create, select and apply appropriate techniques, resources and modern engineering and IT tools including prediction and modelling to complex engineering activities with an under- standing of the limitations.	Random Forest Algorithm in Machine Learning technology is used here.
PO6	**The Engineer and Society:** Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice.	
PO7	**Environment and Sustainability:** Understand the impact of professional engineering solutions	

	in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.	
PO8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.	
PO9	**Individual and Team Work:** Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.	We have worked together to complete our project intime. This team work helped us to complete our project before the targeted date.
PO10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations and give and receive clear instructions.	Communication is very important if we are working in a team. It helps us to get knowledge and learn several technical skills that other persons have.
PO11	**Life-long Learning:** Recognize the need for and have the preparation and ability to engage in independent and life- long learning in the broadest context of technological change.	
PO12	**Project Management and Finance:** Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	

PROGRAM SPECIFIC OUTCOME (PSOs)

PSOs	Program Specific Outcomes	Relevance
PSO1	**Mobile & Web Application Development:** Ability to develop mobile & web applications using J2EE Android and J2ME.	
PSO2	**Cloud Services:** To deploy virtualized and cloud based services in the organization.	

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEOs	Programme Educational Objectives	Relevance
PEO 1	Graduates will be able to analyze, design, and develop advanced computer applications to provide solutions to real world problems.	We have provided a solution for credit card frauds happening in real world scenarios.
PEO 2	Graduates are well trained, confident, research oriented and industry ready professionals who are intellectual, ethical and socially committed.	We are well trained, confident and industry ready to solve the problems that are frequently occurring in the world.
PEO 3	Graduates will have the technical, communication skills and character that will prepare them for technical and leadership roles.	We have enough technical, communication skills and we are well prepared to be responsible in our future roles whether it is a technical role or it may be leadership quality.

COURSE OUTCOME (COs)

COs	Course Outcome	Pos, PSOs and PEOs Mapped
CO1	Develop problem formation and design skills for engineering and real world problems	PO2, PEO1
CO2	Collect and Generate ideas through literature survey on current research areas which help to analyze and present to impart knowledge in different fields.	PO1, PO2, PO10
CO3	Import knowledge on software & hardware to meet industry perspective needs and standards.	PO3,PO5,PEO3
CO4	Create interest to carry out research on innovative ideas as a lifelong learning.	PO6, PO11, PEO2
CO5	Ability to work with a team, and enrich presentation and communication skills.	PO9, PEO3
CO6	Create a platform that makes students employable.	