

Research on Machine Learning Techniques for Credit Card Fraud Detection System

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology in Computer Science and Engineering with Specialization in Internet of Things

by

Siddhartha Soni (20BCT0080)

Rohan Verma (20BCT0311)

Rakshit Gupta (20BCE0824)

Under the guidance of

Prof. K. Ragavan

SCOPE

VIT, Vellore.



May, 2024

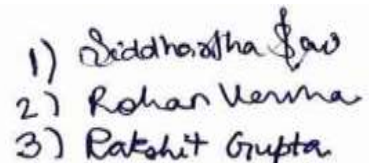
DECLARATION

We hereby declare that the thesis entitled "Research on Machine Learning Techniques for Credit Card Fraud Detection System" submitted by us, for the award of the degree of *Bachelor of Technology in Computer Science and Engineering* VIT is a record of bonafide work carried out by us under the supervision of Prof. Ragavan K.

We further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore

Date : 08.05.24



1) Siddhartha Saw
2) Rohan Verma
3) Rakshit Gupta

Signature of the Candidates

CERTIFICATE


This is to certify that the thesis entitled “Research on Machine Learning Techniques for Credit Card Fraud Detection System” submitted by Siddhartha Soni, Rohan Verma and Rakshit Gupta, 20BCT0080, 20BCT0311 and 20BCE0824, School of Computer Science and Engineering, VIT, for the award of the degree of *Bachelor of Technology in Computer Science and Engineering* is a record of bonafide work carried out by them under my supervision during the period, 03.01.2024 to 8.05.2024, as per the VIT code of academic and research ethics.


The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.


Place : Vellore

Date : 08.05.2024


Signature of the Guide


Internal Examiner


External Examiner


Head of the Department
(Prof. Umadevi KS)
B. TECH - CSE

Head of the Department
(Dr. SHARMILA BANU K)
B. TECH - IOT

ACKNOWLEDGEMENTS

It is our pleasure to express with a deep sense of gratitude to my thesis guide **Dr. K. Ragavan.**, Assistant Professor Senior Grade 1 in School of Computer Science Engineering, Vellore Institute of Technology, for his constant guidance, continual encouragement, and understanding; more than all, he taught us patience in my endeavour. Our association with him is not confined to academics only, but it is a great opportunity on our part to work with an intellectual and expert in the field of Machine Learning.

We would like to express my heartfelt gratitude to **Dr. G Viswanathan**, Chancellor; **Mr. Sankar Viswanathan**, Vice President; **Dr. Sekar Viswanathan**, Vice President; **Dr. G V Selvam**, Vice President; **Dr. V. S. Kanchana Bhaaskaran**, Vice Chancellor, **Dr. Partha Sharathi Mallick**, Pro-Vice Chancellor; **Dr. T. Jayabarathi**, Registrar and **Dr. K. Ramesh Babu**, Dean of School of Computer Science Engineering, for providing us with an enriching environment to work in and for their inspirational guidance throughout the tenure of the course.

We wish to express our sincere gratitude to **Dr. K. Sharmila Banu**, Associate Professor Senior and Head of Internet of Things and **Prof. Umadevi KS**, Associate Professor Senior and Head of Computer Science and Engineering for their support and encouragement for the completion of this project work. In a jubilant mood, I express ingeniously my whole-hearted thanks the Project Coordinators, all teaching staff and members working as limbs of our university for their not-self-centred enthusiasm coupled with timely encouragements showered on us with zeal, which prompted the acquirement of the requisite knowledge to finalize our course study successfully.

Siddhartha Soni

Rohan Verma

Rakshit Gupta

Executive Summary

The goal of the Credit Card Fraud Detection System (CCFDS) project is to provide a reliable and effective system for detecting and stopping credit card fraud. The goal is to quickly identify and stop fraudulent activity by using cutting-edge machine learning algorithms. In order to increase the accuracy and precision of fraud detection, this research uses algorithms like Random Forest and Logistic Regression to forecast fraudulent transactions.

There is a notable imbalance in the statistics, as only 492 out of 284,807 transactions are deemed fraudulent. The project comprises multiple crucial phases, such as feature engineering, model training, and data preprocessing. To improve the efficacy of the system, hybrid model techniques and anomaly detection are applied.

The system's architecture addresses technical, financial, and social viability while emphasizing both functional and non-functional requirements. Crucial Considerations include data security, privacy, legal compliance, and scalability.

The project employs a methodical approach that includes extensive testing, assessment, and optimization to guarantee dependability and efficiency. The ultimate objective is to build a strong system for detecting credit card fraud that can adjust to changing fraud trends, guaranteeing improved security for financial transactions while reducing false positives and offering a smooth user experience.

Chapter No.	CONTENTS	Page No.
	Acknowledgement	i.
	Executive Summary	ii.
	Table of Contents	iii.
	List of Figures	v.
	Abbreviations	vi.
	Symbols and Notations	vii.
1	INTRODUCTION	1
	1.1 Objectives	1
	1.2 Motivation	2
	1.3 Background	4
2	PROJECT DESCRIPTION AND GOALS	5
	2.1 Survey on Existing System	5
	2.2 Research Gap	6
	2.3 Problem Statement	7
3	TECHNICAL SPECIFICATION	8
	3.1 Requirements	9
	3.1.1 Functional	9
	3.1.2 Non-Functional	10
	3.2 Feasibility Study	11
	3.2.1 Technical Feasibility	11
	3.2.2 Economic Feasibility	12
	3.2.3 Social Feasibility	13
	3.3 System Specification	14
	3.3.1 Hardware Specification	14
	3.3.2 Software Specification	15
	3.3.3 Standards and Policies	16
4	DESIGN APPROACH AND DETAILS	17
	4.1 System Architecture	17
	4.2 Design	20
	4.2.1 Data Flow Diagram	20

4.2.2	Use Case Diagram	22
4.2.3	Class Diagram	24
4.2.4	Sequence Diagram	25
5	SCHEDULE, TASKS AND MILESTONES	29
5.1	Gantt Chart	29
5.2	Module Description	30
5.2.1	Module - 1	31
5.2.2	Module - 2	32
5.2.3	Module – 3	33
5.2.4	Module – 4	34
5.2.5	Module - 5	
5.3	Testing	37
5.3.1	Unit Testing	37
5.3.2	Integration Testing	39
6	PROJECT DEMONSTRATION	40
7	RESULT & DISCUSSION	50
8	SUMMARY	54
9	REFERENCES	55
	APPENDIX A – SAMPLE CODE	57

List of Figures

Figure No.	Title	Page No.
4.1	System Design	19
4.2	Data Information	19
4.3	Data Flow Diagram	20
4.4	Use Case Diagram	22
4.5	Class Diagram	24
4.6	Sequence Diagram	25
5.1	Gantt Chart	29
5.2	Data Information	30
5.3	Feature Importance	32
5.4	Outliers V/S no Outliers	33
5.5	Random Forest Classifier	34
5.6	Logistic Regression Classifier	35
5.7	Hybrid Model	36
5.8	Dataset Description	37
5.9	Unit Testing - 2	37
5.10	Unit Testing - 3	38
5.11	Integration Model Testing	39
6.1	Fraud V/S Non Fraud Transactions	40
6.2	Fraud Transactions with respect to Time	41
6.3	Correlation Plot	42
6.4	Feature Importance	43
6.5	Confusion Matrix of Random Forest	45
6.6	Confusion Matrix of Logistic Regression	47
7.1	Random Forest Model Evaluation	50
7.2	Logistic Regression Model Evaluation	51
7.3	Hybrid Classifier Model Evaluation	52
7.4	ROC Score Comparison	53

List of Abbreviations

CCFDS	Credit Card Fraud Detection System
PCI DSS	Payment Card Industry Data Security Standard
RF	Random Forest
LR	Logistic Regression
RAM	Random Access Memory
ROM	Read Only Memory
ML	Machine Learning
AI	Artificial Intelligence
EDA	Exploratory Data Analysis

Symbols and Notations

σ	Sigma
Z	linear combination of features
b	bias term

1. INTRODUCTION

1.1 OBJECTIVE

The core objective of this project revolves around crafting, refining, and validating a Credit Card Fraud Detection System (CCFDS) employing machine learning techniques to swiftly identify and thwart fraudulent transactions occurring in real-time. This endeavor entails a comprehensive exploration of various machine learning algorithms such as Random Forest and Logistic Regression, meticulously assessing their efficacy through rigorous testing on collected data. Moreover, a key aspect of this project involves amalgamating the predictive capabilities of both classifiers to enhance the system's accuracy in discerning fraudulent activities. Concurrently, efforts are dedicated to portraying the data visually, providing insightful representations that aid in comprehending patterns and trends within the dataset. Looking forward, the ultimate vision is to deploy this system to operate seamlessly with real-time data, bolstering its effectiveness in safeguarding against fraudulent transactions with swift and precise detection capabilities.

This project's overarching objective is to improve the Credit Card Fraud Detection System (CCFDS) through a variety of means. It does this by utilizing machine learning algorithms' adaptability to address the complex problem of fraud detection in financial transactions. In order to identify each algorithm's unique advantages and disadvantages and eventually combine them to provide a more reliable and accurate fraud detection system, the project will carefully examine and test techniques like Random Forest and Logistic Regression. Moreover, the research also explores data visualization, using graphical displays to clarify complex trends and irregularities in the dataset, allowing for a better understanding of fraudulent activity. With an eye toward the future, the initiative plans to integrate the system with real-time data feeds.

1.2 MOTIVATION

Consumers and organizations are seriously threatened by credit card fraud, which is the unlawful use of credit card information for cash withdrawals or other fraudulent activities. 3,432 instances of credit and debit card fraud were reported in India in 2021 alone, indicating a growing issue brought on by phishing, data breaches, and other criminal activity. Consumers may suffer significant consequences, including money loss, identity theft, and a decline in confidence in electronic payment systems. Businesses may also experience chargebacks and harm to their brand. Credit card fraud detection systems (CCFDS) are an example of an advanced solution that is required since traditional fraud detection approaches are frequently insufficient to counter these evolving threats. Using machine learning (ML) methods, CCFDS can examine vast amounts of transaction data in order to identify tendencies and anomalies, making it possible to identify and stop fraudulent activity in real time. This strategy not only lowers financial loss and improves security, but it also increases consumer confidence in digital payments. The goal of CCFDS implementation is to make the financial system safer by lessening the effects of fraud and preserving the ease of credit card transactions.

In order to combat the increasing sophistication of fraud schemes, credit card fraud detection systems (CCFDS) must be implemented. Financial institutions and companies need to use increasingly sophisticated technology to keep ahead of fraudsters who are getting better at taking advantage of weaknesses in credit card systems. In order to detect fraudulent transactions more quickly and accurately, machine learning (ML) and artificial intelligence (AI) are essential components of CCFDS. These systems have the ability to learn from fresh data continually, adjusting to evolving fraud tendencies and lowering false positives—all of which contribute to preserving a favorable client experience

1.3 BACKGROUND

Understanding the historical history and the technical developments that have influenced the present method of identifying and preventing credit card fraud is necessary to comprehend the background of a credit card fraud detection system (CCFDS). Since the creation of credit cards, there has been a recurring issue with credit card fraud as thieves always come up with new ways to take advantage of holes in payment systems. Robust detection systems are becoming increasingly necessary since e-commerce, digital transactions, and online shopping have raised the risk of credit card fraud.

When credit card fraud detection first started out, most techniques were rule-based and relied on predetermined sets of rules and heuristics to spot suspicious transactions. These regulations may involve looking for transactions from odd geographic areas, transactions with abnormally large quantities, or transactions that happen quickly one after the other. Although they were somewhat successful, these rule-based systems had several drawbacks. They required regular maintenance as fraud trends changed, frequently produced high false positive rates, and were unable to adjust to new fraud patterns.

The issue of credit card fraud has become more severe in the twenty-first century due to the quick development of e-commerce and digital payments, with thieves using ever-more-advanced methods. More sophisticated detection systems were necessary as data breaches, phishing scams, and the use of malware to steal credit card information increased in frequency. As a result of this change, machine learning (ML) and artificial intelligence (AI) approaches gained traction. Unlike old methods, these techniques could evaluate massive amounts of transaction data and identify intricate patterns.

2 PROJECT DESCRIPTION AND GOALS

2.1 SURVEY ON EXISTING SYSTEM

According to recent research, credit card theft can be successfully identified using deep learning techniques, namely a 20-layer Convolutional Neural Network (CNN) by authors - Fawaz Khaled, Alarfaj , Iqra Malik, Hikmat Ullah Khan , Naif Almusallam , Muhammad -Ramzan , And MuzamilAhmed [1]. This shows that sophisticated algorithms might perform better at detecting fraud, and scientists are looking into more sophisticated techniques to increase accuracy.

The IEEE study from 2023 emphasizes how critical adaptability is to machine learning models used in banking fraud detection by authors Maram Alamri and Mourad Ykhlef [2]. It draws attention to the necessity of generalization across various datasets and data reliance in order to build a robust system, emphasizing the requirement for dependable and flexible solutions.

A 2022 study proposes a hybrid method of detecting credit card fraud that combines deep neural networks and CatBoost by authors - Nghia Nguyen, Truc Duong, Tram Chau, Van-Ho Nguyen, Trang Trinh, Duy Tran And Thanh Ho [3]. This increases accuracy but complicates understanding the inner workings of the models and necessitates enormous amounts of data for training.

For the purpose of detecting credit card fraud, researchers have improved the robustness and diversity of base classifiers in their Adaboost based approach by authors - Wang Ning, Siliang Chen, Songyi Lei², And Xiongbiao Liao [4]. Nevertheless, this intricate method can pose difficulties in execution and necessitate substantial resources.

A 2023 study addresses uneven credit card data in fraud detection by presenting a hybrid technique that combines undersampling and oversampling by authors - Maram Alamri And Mourad Ykhlef. This method presents BCBSMOTE, a novel strategy intended to balance skewed datasets—a prevalent problem in the detection of credit card fraud. The goal of BCBSMOTE's data distribution balancing strategy is to increase the precision of fraudulent transaction identification. Although this technique has the potential to increase detection rates, there are certain hazards involved [5].

A 2023 study explores a machine learning-based method for detecting online payment fraud by authors - Abdulwahab Ali Almazroi 1 And Nasir Ayub. [6]. The method aims to improve security by detecting fraudulent activity instantly. However, its efficacy may be limited due to data reliance and computational resources. The study underscores the challenges in developing reliable, efficient, and scalable online payment fraud detection models.

A new framework for credit card fraud detection with the goal of enhancing detection accuracy by tackling feature selection and data imbalance by authors - Ayoub Mniai , Mouna Tarik , And Khalid Jebari Ayoub Mniai , Mouna Tarik , And Khalid Jebari [7]. In order to detect fraudulent transactions, the framework focuses on skewed data distributions and the selection of pertinent attributes. The study does, however, provide a warning on its generalizability because its results can vary depending on the dataset. In spite of this, the framework which emphasizes flexible methods and careful feature selection represents a substantial breakthrough in fraud detection.

The goal of a 2022 IEEE study is to achieve high accuracy and fewer false positives when examining the potential of deep learning for credit card fraud detection by authors - Priyanshu Katiyara, Lakshay Sachabb, Rithik Chhabrac, Vishal Pandeyd , Dr. Hoor Fatima [8]. The study emphasizes the value of thorough preprocessing and high-quality data, but it also shows how the credit card industry may be able to increase the accuracy of fraud detection.

A customizable framework for synthesizing unbalanced data using a duo-GAN structure is proposed in an IEEE research from 2021 by authors - Francisco Ferreira , Nuno Lourenço, Bruno Cabral [9]. This approach addresses data imbalance in domains such as credit card fraud detection. In order to enhance the representation of underrepresented classes and lessen the influence of data imbalance on model training, the framework creates fake samples.

In order to solve data imbalance in domains such as credit card fraud detection, a 2021 IEEE study suggests a flexible framework using a duo-GAN structure to synthesize unbalanced data [10] by authors - Mariachiara Mecati ,, Marco Torchiano , Antonio Vetrò, And Juan Carlos De Martin. In order to reduce the influence of data imbalance on model training.

An IEEE study from 2019 called ObjectiveCost-Sensitive Support Vector Machine (OCS-SVM) by authors - Shuang Yu, Xiongfei Li, Xiaoli Zhang And Hancheng Wang enhances classification tasks' accuracy and dependability, particularly when misclassification costs fluctuate [11].

Using the European Cardholders 2013 dataset, a 2023 IEEE paper introduces a unique approach to fraud detection using Generative Adversarial Networks (GANs) by authors - Fahdah A. Almarshad , Ghada Abdalaziz Gashgari , And Abdullah I. A. Alzahrani [12]. GANs increase the accuracy of detecting credit card fraud.

The 2020 IEEE paper examines a number of cybercrime detection methods, highlighting how they have evolved to meet new threats. While recognizing the necessity for flexibility and adaptation by authors Wadha Abdullah AlKhater, Somaya AlMaadeed, Abdulghani Ali Ahmed [13].

In [14], 20 different datasets from the UCI repository were used to compare two categorization models: Random Forest and Decision Tree (J48). The datasets have 148–20,000 instances each, varying in size. The comparison concentrated on important classification metrics, such as the proportion of instances that were correctly and wrongly identified, as well as F-Measure, Precision, Accuracy, and Recall.

Educational Research utilizes logistic regression techniques to test research hypotheses, providing a comprehensive evaluation of outcomes and assumptions, with recommendations for minimal observation-to-predictor ratio and reporting formats [15].

2.2 RESEARCH GAP

- **Insufficient Consideration of Unbalanced Datasets:** The datasets pertaining to credit card fraud are naturally unbalanced, containing a comparatively small proportion of fraudulent transactions in contrast to authentic ones. The difficulties presented by unbalanced data and the tactics needed to reduce bias and enhance model performance in such situations are frequently ignored by the research that already exists.
- **Limited Generalization Across Diverse Datasets:** Some research only concentrate on certain datasets or geographical areas, which restricts the applicability of their conclusions in more general settings. To guarantee the models' relevance in a variety of real-world situations, studies assessing the effectiveness of credit card fraud detection models across various datasets and geographical areas are required.
- **Advanced methods that potentially enhance model performance,** such anomaly detection and feature transformation, are frequently ignored in the literature.
- **The important step of performing exploratory data analysis (EDA)** to learn more about the properties and distribution of data pertinent to credit card fraud detection is frequently missed by existing literature surveys. This omission restricts our ability to comprehend the underlying patterns and abnormalities in the dataset, which could result in subpar model construction and assessment.

2.3 PROBLEM STATEMENT

Credit card fraud is a serious problem that affects financial institutions as well as consumers. Conventional rule-based detection systems frequently are unable to keep up with the rapid advancements in fraud tactics, which results in an increase in false positives and fraudulent activity that goes undiscovered. The precision and agility needed to effectively counter sophisticated fraud schemes are lacking in current technologies. Furthermore, the growth of internet commerce has opened up new channels for scammers to take advantage of security flaws and carry out fraudulent transactions without being detected. This urgent problem necessitates creative solutions that can quickly spot and stop fraudulent activity, supporting efforts to prevent and detect fraud in order to protect consumer confidence and financial integrity in the digital era. It is becoming more and more necessary to create sophisticated fraud detection systems that take advantage of these obstacles with machine learning, anomaly detection, enhance accuracy and responsiveness in identifying fraudulent transactions. Such systems have the potential to revolutionize fraud prevention strategies, providing proactive protection against emerging threats while minimizing false positives and ensuring a seamless and secure transaction experience for consumers.

To combat credit card fraud, parties such as financial institutions, regulatory organizations, and law enforcement agencies must work together in addition to advancing technology. Stakeholders can increase their defenses against fraudulent activity by creating standardized standards for fraud detection and prevention and encouraging information sharing. Additionally, educating customers about typical fraud schemes and the best ways to protect their financial information will give them the confidence to take preventative action to protect themselves against fraud. By use of inventive technology, cooperative endeavors, and consumer education, the hazards associated with credit card fraud.

3 TECHNICAL SPECIFICATION

3.1 REQUIREMENTS

The Credit Card Fraud Detection System (CCFDS) project depends on having clearly specified criteria that cover both functional and non-functional features. This section lists the fundamental attributes and features that the system has to have in order to reliably, securely, and efficiently identify and stop credit card fraud.

3.1.1 FUNCTIONAL

- **Dataset:** The dataset, containing 284,807 transactions over two days, includes 492 frauds out of 284,807. It's heavily skewed, with frauds accounting for 0.172% of transactions. The dataset's major components are V1, V2, ... V28, with 'Time' and 'Amount' remaining unchanged. Confidentiality concerns prevent disclosure.
- **Algorithm Selection:** A range of machine learning techniques, such as logistic regression, decision trees, random forests, etc., should be available in the system for the purpose of detecting credit card fraud.
- **Data Cleaning:** In order to handle missing values, outliers, and inconsistent data in the credit card transaction dataset, the system needs to execute data cleaning.
- **Feature Engineering:** It should support feature engineering techniques to extract relevant features from the preprocessed data, enhancing model performance.

3.1.2 NON – FUNCTIONAL

- **Response Time:** The system should respond to user requests promptly, with transaction processing and model inference completed within acceptable timeframes.
- **Throughput:** It should be able to process a high number of credit card transactions quickly and effectively without experiencing any performance deterioration.

- **Availability:** To guarantee continuous operation, the system must have a high degree of availability, with little downtime and service interruptions.
- **Fault Tolerance:** It should have systems in place to recover gracefully from faults and preserve data integrity, making it resistant to system failures.
- **Compute Resources:** Access to computational resources such as CPUs or GPUs for model training and inference.
- **Anonymization:** To safeguard the security and privacy of cardholders, personally identifiable information (PII) from credit card transactions should be anonymized or pseudonymized.
- **Legal requirements:** Adhere to all applicable legal and regulatory obligations pertaining to financial transactions, data protection, and credit card fraud detection.

3.2 FEASIBILITY STUDY

The Credit Card Fraud Detection System (CCFDS) feasibility study assesses the project's practicality and viability from a number of angles, including social, economic, and technical ones.

3.2.1 Technical Feasibility

Technical viability evaluates whether the project can be carried out successfully in terms of technology. This assessment entails looking at things like:

- **Resource Availability:** To determine if the dataset, software and system is available for the project.
- **Compatibility with Current Systems:** Determine whether there will be major conflicts or interruptions if the planned project integrates with or coexists with current hardware, software, and systems.

- **Compatibility with Current System:** Determine whether there will be major conflicts or interruptions if the project integrates with or coexists with current hardware, software, and system.
- **Scalability and Performance:** Determine the suggested solution can accommodate growing user loads, data volumes, and processing demands while keeping performance levels appropriate.
- **Technical dangers and Challenges:** Recognize any potential dangers, restrictions, or difficulties that could occur in the course of developing and implementing a project.

3.2.2 Economic Feasibility

- The expenses associated with purchasing servers, storage devices, and other infrastructure components will differ according to the needs and preferences of the company. Advanced CPU and GPU capabilities found in high-performance servers can range in price from several thousand to tens of thousands of dollars per unit.
- **Operational Costs:** Estimating the ongoing operational costs of maintaining and running the CCFDS.
- **Fraud Losses Reduction:** Calculating the possible decrease in fraud losses brought about by the CCFDS's application.
- Scalability without appreciable extra expenses should be a feature of a well-designed system, enabling a positive return on investment as benefits mount and costs are deducted over time.
- Depending on the organization's business and location, costs related to guaranteeing compliance with regulatory standards, such as audit fees, legal counsel, and implementation of compliance procedures, might differ significantly.

3.2.3 Social Feasibility

- **User Acceptance:** Credit card holders, financial institutions, and regulatory agencies must all accept and cooperate with the Credit Card Fraud Detection System (CCFDS) for it to be successfully implemented. To guarantee a user-friendly and intuitive system interface, it is crucial to interact with these stakeholders at every stage of the development process, responding to their worries, getting their input, and applying user-centric design concepts.
- **Privacy Concerns:** Because CCFDS handles sensitive financial data, privacy and data protection issues are brought up. To protect the integrity and confidentiality of transaction data, strong security mechanisms including encryption, access limits, and data anonymization strategies must be put in place. To gain the trust of consumers and adhere to privacy standards, it is imperative to maintain transparency regarding data collecting, processing, and storage procedures.
- **Ethical Aspects:** The creation and management of CCFDS are heavily influenced by ethical factors. The system's fraud detection algorithms and decision-making procedures must abide by moral precepts including justice, accountability, and transparency.
- **Legal and Regulatory Compliance:** CCFDS is required to abide by all applicable laws, rules, and industry guidelines that control data protection and financial activities. This entails abiding by rules like the General Data Protection Regulation (GDPR), the Payment Card Industry Data Security Standard (PCI DSS).

3.3 System Specification

The Credit Card Fraud Detection System's (CCFDS) hardware and software specifications are essential for guaranteeing the system's dependability, scalability, and performance. For the Credit Card Fraud Detection System (CCFDS) to function reliably, scalable, and efficiently, it is essential that the hardware and software specifications fulfill strict requirements. To effectively manage the computing demands of the system, careful selection of hardware components is necessary. Large transaction datasets can be processed

quickly by multi-core CPUs with fast clock speeds, and efficient operation during demanding data processing activities is ensured by enough RAM.

3.3.1 Hardware Specification :

- In order to maintain ideal operating temperatures and avoid thermal throttling, which can impair system performance, efficient cooling solutions are essential. Examples of these solutions include liquid cooling systems, fans, and CPU and GPU heatsinks.
- Efficient system operation and rapid data access are contingent upon optimal storage setup.
- The capacity to carry out computational tasks successfully and efficiently. Performance is critical for CCFDS tasks including inference, model training, and data preprocessing.
- Kaggle offers CPU and GPU instances on machines with different performance capacities. Because of their capacity for parallel processing, GPU instances are especially useful for jobs involving deep learning models, as they can greatly shorten training periods.

3.3.2 Software Specification:

- The main programming language utilized to construct the CCFDS project on Kaggle is Python. Python is a great choice for data science project because it provides a large selection of libraries and tools for data processing, analysis, and machine learning.
- Building machine learning model and carrying out data analysis activities are common uses for Python because to its ease of use, readability, and vast ecosystem of third-party libraries.
- Python's core libraries for data processing and numerical computing are called Pandas and NumPy, respectively. They offer the fundamental data structures and operations needed to effectively handle big datasets, carry out data preparation operations, and extract features from transaction data.

- For the purpose of detecting fraud, machine learning models like Random Forest, Logistic Regression, and Gradient Boosting Classifier are trained and assessed using Scikit-learn.
- In the CCFDS project, visualization is a crucial component of data processing and model evaluation. Python packages like Matplotlib, Seaborn, and Plotly can be used to create interactive visualizations, charts, and instructive plots that allow users to investigate features correlations, data distributions, and metrics related to model evaluation.

3.3.3 Standards and Policies:

- **Data Security Standards:** Ensuring sensitive credit card data is protected and confidence is upheld by adhering to data security standards like the Payment Card Industry Data Security Standard (PCI DSS). Implementing encryption, safe network setups, access limits, and routine security audits are all part of adhering to regulations.
- **Privacy rules:** Protecting people's right to privacy requires the establishment of explicit privacy rules and procedures. These policies should specify procedures for gathering consent and addressing the rights of data subjects, as well as methods for collecting, storing, processing, and exchanging data.
- **Fraud Detection Algorithms and Models:** Accountability, justice, and transparency are necessary for the development and application of strong fraud detection algorithms and models. To ensure accuracy and efficacy, model performance must be validated and monitored on a regular basis.
- **Regulatory Compliance:** It is critical to adhere to all applicable laws, rules, and industry standards pertaining to data security, fraud prevention, and financial transactions. It is imperative for organizations to be updated about modifications to regulatory requirements and to guarantee continuous adherence to relevant laws and regulations.
- **Incident Response and Reporting:** Effective handling of security incidents and breaches depends on having incident response protocols and reporting systems in place.

4 DESIGN APPROACH AND DETAILS

4.1 System Architecture:

- **Data collection :** The dataset, containing 284,807 transactions over two days, includes 492 frauds out of 284,807. It's heavily skewed, with frauds accounting for 0.172% of transactions. The dataset's major components are V1, V2, ... V28, with 'Time' and 'Amount' remaining unchanged. Confidentiality concerns prevent disclosure. The dataset has been collected and analysed during a research collaboration of Worldline and the Machine Learning Group.
- **Preprocessing:** To guarantee data quality and get it ready for model training, the gathered dataset is preprocessed. Managing outliers, inconsistent data, and missing numbers are all part of this procedure. The integrity and dependability of the dataset are improved by applying techniques like imputation, outlier detection, and data cleansing.
- **Feature Engineering:** From the raw transaction data, feature engineering techniques are used to extract pertinent information and generate new features. This could entail encoding categorical variables, creating new features based on domain knowledge, or altering already-existing features.
- **Data Visualization:** The system does a thorough data visualization prior to preprocessing in order to obtain an understanding of the distribution, relationships, and trends within the dataset. Potential trends and anomalies can be found with the aid of visualizations like correlation matrices, scatter plots, and histograms.
- **Outlier Detection:** After being carefully examined to determine whether they are true anomalies or data errors, outliers are used to influence further preprocessing procedures.
- **Algorithm Selection:** In order to detect credit card fraud, the system chooses and trains machine learning algorithms - logistic regression and random forests. Create a separate model by combining both the models.

- **Data Splitting:** A training set and a testing set are created from the preprocessed dataset. The testing set provides unseen data for assessing model performance, whereas the training set trains the machine learning algorithms.
- **Training Process:** To identify trends and connections between features and fraudulent transactions, the chosen algorithms are trained using the labeled training data. The algorithms modify their internal conditions during training in order to maximize efficiency and reduce prediction mistakes.
- **Performance Metrics:** A variety of performance metrics, like as accuracy, precision, recall, and F1-score, are used to assess trained models against the testing dataset. These measurements shed light on how well the models categorize transactions and spot fraudulent activity.
- **Optimization and Adjustment:** To enhance the models' performance, modifications may be made in light of the evaluation's findings. This could entail experimenting with different algorithms, modifying decision thresholds, or fine-tuning model parameters. The objective is to reduce false positives and false negatives, maximize fraud detection rates, and optimize model performance.

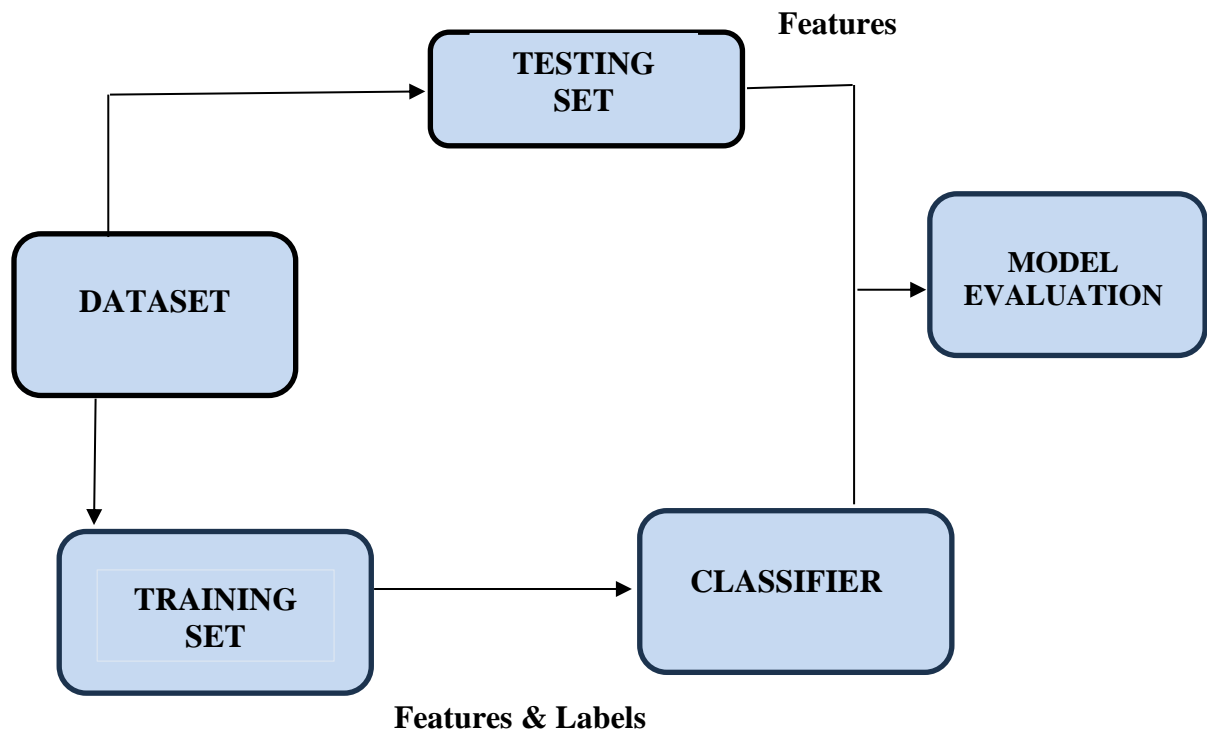


Figure 4.1 : System Design

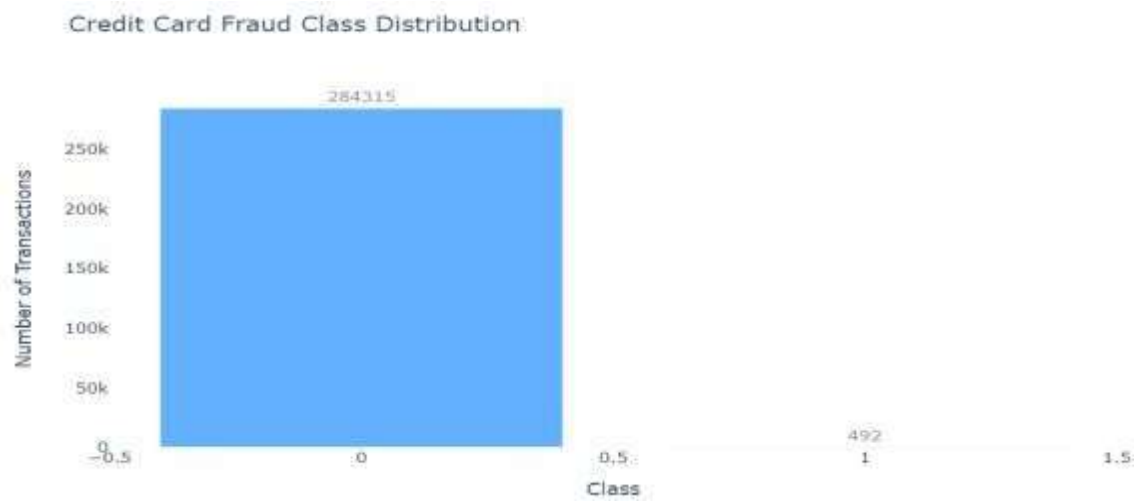


Figure 4.2 : Data Information

4.2 DESIGN

4.2.1 Data Flow Diagram

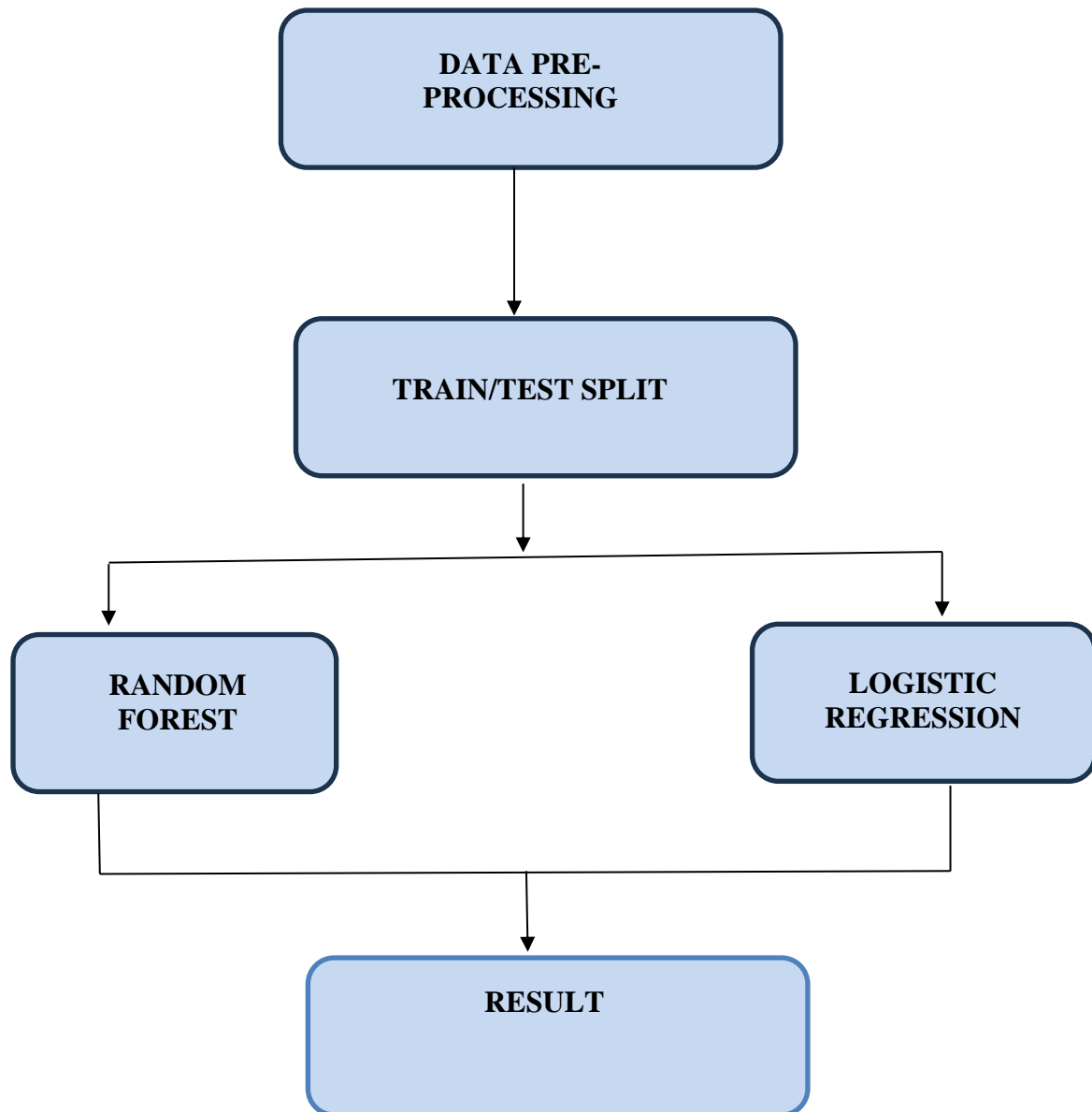


Figure 4.3 : Data Flow Diagram

The dataset comprising credit card transaction records flows into the system at the start of the process, as shown in the Credit Card Fraud Detection System Data Flow Diagram as shown in figure 4.3. Next, an 80:20 ratio is used to divide the dataset into training and test datasets. The training dataset is then fed into two distinct branches, Random Forest (RF) Classifier and Logistic Regression Classifier, each of which uses a different classification algorithm. After training its corresponding model, each branch calculates metrics like precision, recall, and F1-score to assess the model's performance on the test dataset. The hybrid model's performance is then evaluated by comparing its Receiver Operating Characteristic (ROC) score to that of the individual models after the predictions from the two models have been integrated. This data flow demonstrates the methodical procedure for instruction, assessment, and comparing different classification models within the Credit Card Fraud Detection System to ensure optimal fraud detection performance.

The training data flows into distinct branches to train the Random Forest Classifier and Logistic Regression Classifier models after the first data preprocessing processes and dataset splitting. Using the assigned features and labels from the training dataset, each model is trained. The test dataset is used to evaluate the models' predicted performance after training. For both algorithms, metrics including precision, recall, and F1-score are computed to assess how well they identify fraudulent transactions.

Moreover, the predictions from both models are merged in order to take use of their respective strengths and investigate possible synergy. By combining these predictions, a hybrid model is created that seeks to improve fraud detection performance above that of any one model working alone. Next, the predictions made by the hybrid model are assessed, and the ROC score is contrasted with the results obtained from the Random Forest and Logistic Regression models separately.

4.2.2 USE CASE DIAGRAM

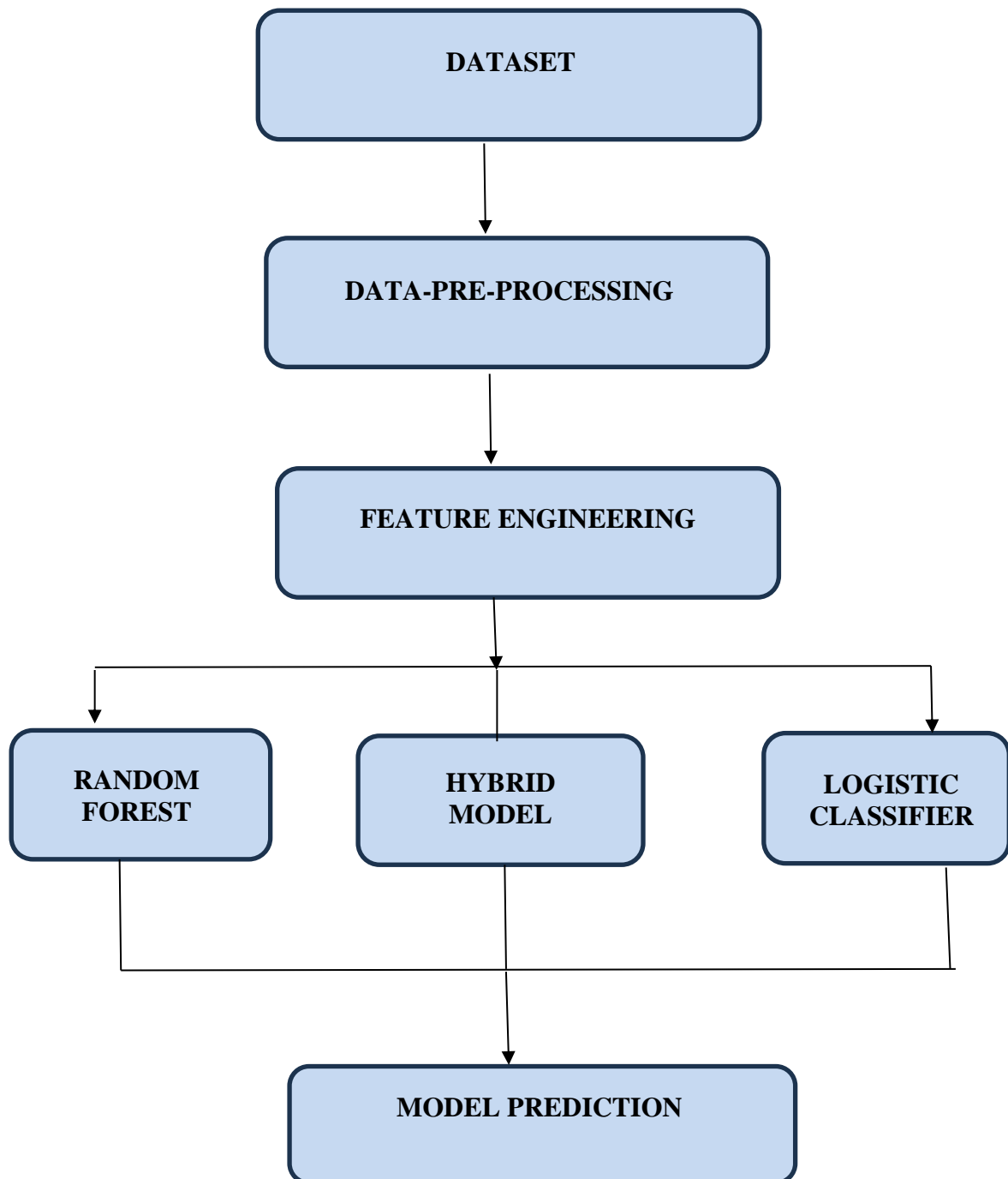


Figure 4.4 :Use Case Diagram

As shown in figure 4.4, one crucial use case for the Credit Card Fraud Detection System (CCFDS) is transaction analysis for fraud detection. In this case, a credit card holder starts a transaction, which causes transaction data to be sent to the CCFDS for assessment. The system carefully examines the data after it has been received, checking several transaction variables such as the amount, merchant, timestamp, and past trends. The CCFDS uses advanced algorithms to calculate a fraud detection score or classification that shows the probability of fraud in the transaction. The system marks a transaction as potentially fraudulent if the computed score is higher than a predetermined threshold. This allows the cardholder or appropriate authorities to look into the transaction further. On the other hand, if the score is less than the cutoff, the transaction is accepted, and authorization is granted. Subsequently, the credit card holder receives confirmation of the transaction status, either approved or flagged for potential fraud, ensuring transparency and security in financial transactions. This use case underscores the pivotal role of the CCFDS in pre-emptively identifying and mitigating fraudulent activities, safeguarding the integrity of credit card transactions and bolstering trust among stakeholders.

Instead of focusing on real-time transaction streams, the Credit Card Fraud Detection System (CCFDS) uses preprocessed data, with batch processing and retrospective analysis taking center stage. The main use case in this situation is the routine examination of past transaction data to spot fraudulent trends and reduce risks.

The system might go through a planned batch processing procedure, for example, in which it absorbs most of the transaction data that has collected over a given time frame, like weekly or daily intervals. After obtaining the pre-processed dataset, the CCFDS applies its fraud detection algorithms, looking at past trends and different aspects of transactions to identify possible fraud. After analysis is finished, the system produces reports or alerts that flag unusual or suspected transactions so fraud analysts or other investigators can look into them further.

4.2.3 CLASS DIAGRAM

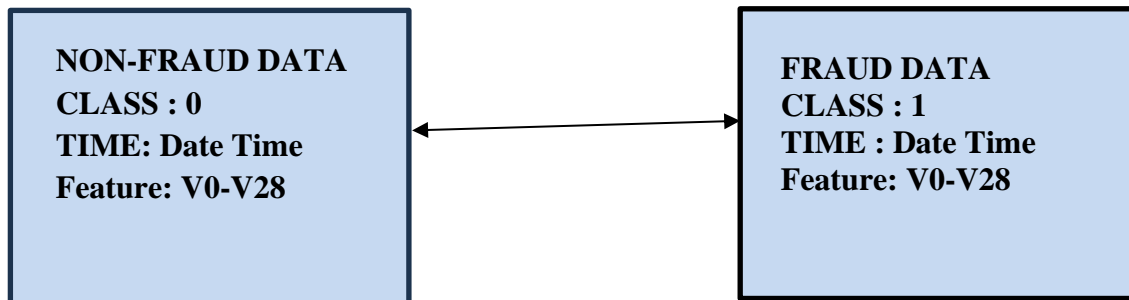


Figure 4.5 : Class Diagram

Two separate classes—non-fraudulent transactions (Class 0) and fraudulent transactions (Class 1)—are depicted in the Credit Card Fraud Detection System (CCFDS) class diagram as shown in figure 4.5. Transactional properties, such as time, amount, and 28 features (designated as V0-V28), are encapsulated in each class. The essential elements of credit card transactions are reflected in the shared properties of both classes. But the difference is in the class label: transactions classified as Class 0 are not fraudulent, while those classified as Class 1 are fraudulent. The data structure within CCFDS is represented in this class diagram in a clear and straightforward manner, making it easier to comprehend and apply fraud detection algorithms based on transactional attributes.

4.2.4 SEQUENCE DIAGRAM

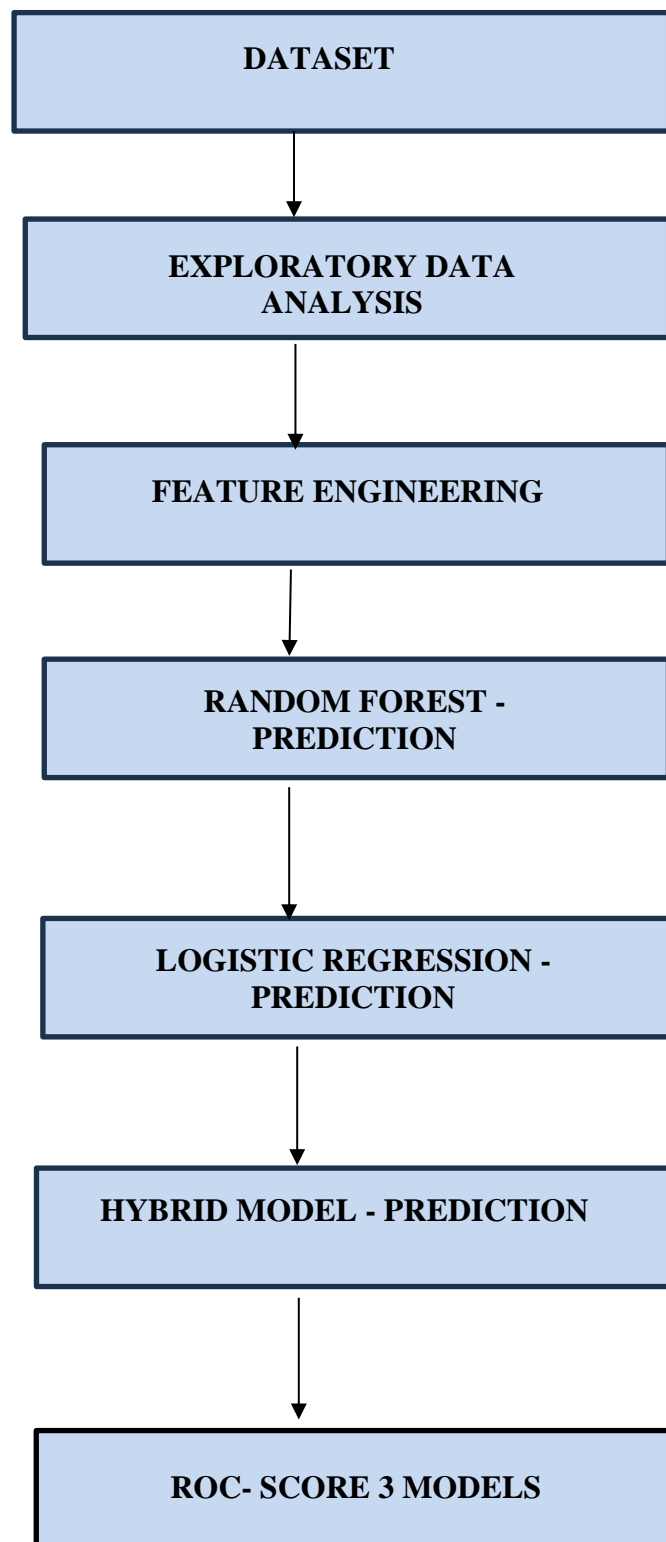


Figure 4.6 : Sequence Diagram

As shown in figure 4.6 in the sequence diagram above, the procedure starts with the dataset being given to the Credit Card Fraud Detection System (CCFDS). The system starts with exploratory data analysis (EDA) after obtaining the dataset, looking at different parts of the data to learn more about its distribution, trends, and possible anomalies. After EDA, the dataset is subjected to feature engineering, which selects, extracts, or modifies pertinent features in order to improve the models' predictive capacity.

Following feature engineering, the dataset is divided into training and testing sets. Random Forest (RF) and Logistic Regression (LR) classifiers are separately trained on the training set. These classifiers are used to forecast the test dataset and estimate the probability of fraud after they have been trained.

In parallel, the predictions from both the LR and RF classifiers are combined to create a hybrid model. After then, the predictions made by the hybrid model are assessed, and the model's overall performance in identifying fraudulent transactions is determined by computing the Receiver Operating Characteristic (ROC-score)

In order to improve the Credit Card Fraud Detection System's capacity to recognize and stop fraudulent activity in credit card transactions, a series of steps are presented here that demonstrate the methodical process of data processing, model training, prediction, and evaluation.

4.3 CONSTRAINTS, ALTERNATIVES AND TRADEOFFS

Constraints :

- **Data Availability and Quality:** To train dependable models, data must be precise, clean, and well-organized. Model training may be skewed by imbalanced data, which has fewer fraudulent transactions than non-fraudulent ones.
- **Rules and Adherence:** Regulations like the CCPA and GDPR, which restrict data usage and demand privacy protections, must be complied with by financial institutions. Constraints related to compliance may limit the kinds of data that can be shared or utilized.

- **Restrictions on Resources:** The amount of data analyzed and the sophistication of the models can be limited by processing power and storage capacity. Budget and staffing constraints may effect the scope of the project.
- **Real-time Requirements:** Quick and effective algorithms are required for certain fraud detection systems to function in real-time. In real-time settings, response time and latency are crucial.

Alternatives:

- **Time-based analysis :** A technique used in credit card fraud detection to spot odd or suspect activity over time is time-based analysis. In order to find anomalies such abrupt spikes in activity, changes in spending patterns, and temporal patterns, it entails looking at patterns and trends across time. This method aids in spotting questionable behavior that could indicate fraud, such as unexpected, frequent use of a card at night. Fraud detection systems are able to detect and react to suspicious activity more efficiently when they employ time-based analysis.
- **Behavioural Biometrics :** ne technique for detecting credit card fraud is behavioural biometrics, which looks for distinct patterns in user behaviour. This method compares observed behaviour to the cardholder's normal patterns based on individual features. For example, mouse movements, typing patterns, and gait analysis can all be used to spot fraudulent behaviour.
- **Text mining :** Advanced credit card fraud detection algorithms use a range of data sources and approaches to identify unusual or suspect activity. Time-based analysis examines transaction trends over predefined time periods.

Trade-offs :

- Tighter security protocols could lead to better fraud detection, but they might also degrade user experience.
- Setting a goal of fewer false positives may result in more false negatives, which could cover up fraud.
- Customers may experience inconvenience as fewer false negatives could result in more false positives.
- Longer training periods and greater computational resources are needed for complex models.

5 SCHEDULE, TASKS AND MILESTONES

5.1 GANTT CHART

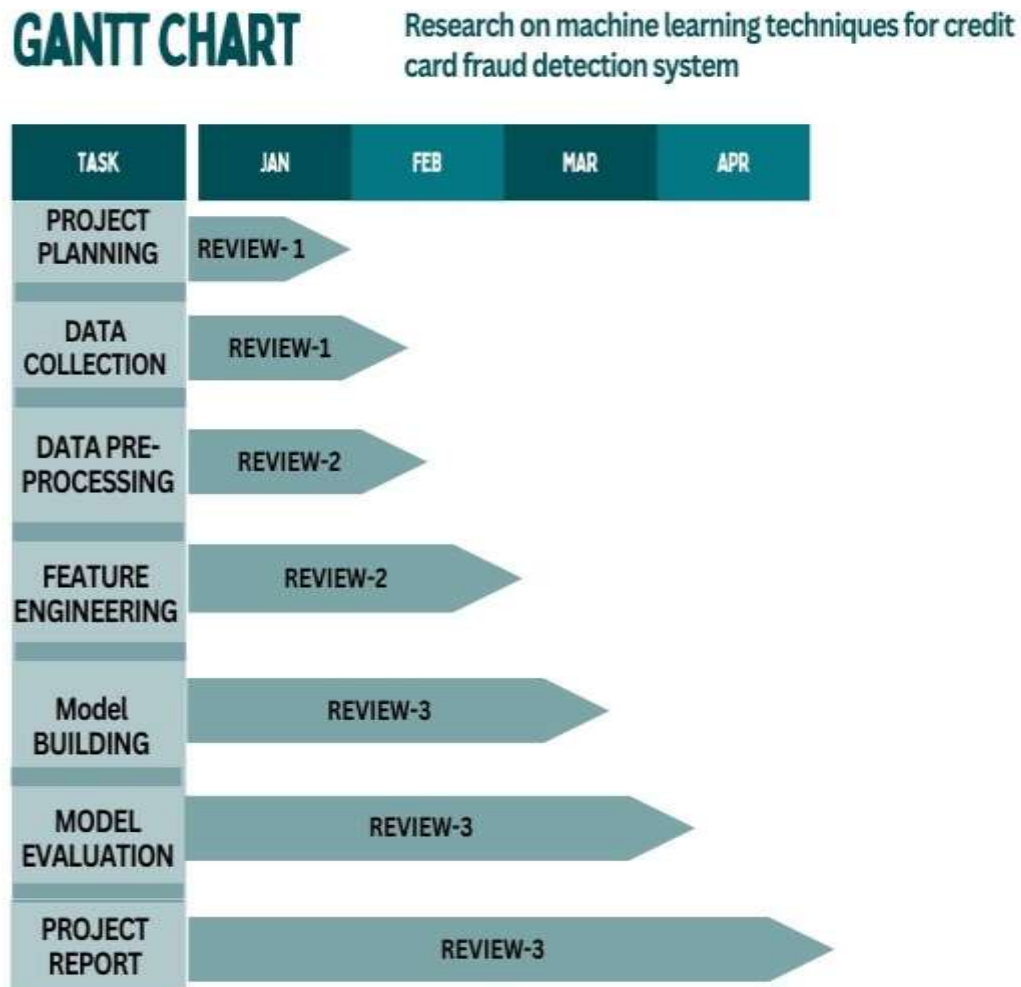


Figure 5.1 :Gantt Chart

5.2 Module Description

- **Data Collection:** Gather transactional data from various sources.
- **Data Preprocessing:** Clean, normalize, and transform raw data.
- **Feature Engineering:** Select, extract, and create relevant features.
- **Anomaly Detection:** Identify unusual patterns or outliers.
- **Machine Learning Models:** Develop and train fraud detection algorithms (combination of Random forest and Logistic Regression)

5.2.1 MODULE 1

The dataset, containing 284,807 transactions over two days, includes 492 frauds out of 284,807. It's heavily skewed, with frauds accounting for 0.172% of transactions. The dataset's major components are V1, V2, ... V28, with 'Time' and 'Amount' remaining unchanged as shown in figure 5.2 . Confidentiality concerns prevent disclosure. The dataset has been collected and analysed during a research collaboration of Worldline and the Machine Learning Group.



Figure 5.2 : Data Information

5.2.2 MODULE 2

In order to prepare the dataset for training machine learning models in the Credit Card Fraud Detection System (CCFDS), data preparation is essential. The data must be cleaned, transformed, and improved throughout a number of stages in order for it to be of high quality.

It is necessary to locate and handle missing data in order to identify credit card fraud. To ensure uniform scaling, features also need to be standardized or normalized. To boost predictive power, more features need to be designed, and the data needs to be split into training and testing sets. Missing data is located and handled with techniques like imputation or elimination. To appropriately scale characteristics, normalization or standardization techniques are applied, adding new features or changing ones that already exist. Finally, the dataset is split into training and testing sets to verify robustness and generalization. This enables the model's performance to be trained on a single piece.

At the data preprocessing stage for credit card fraud detection, several factors are carefully taken into account, including handling missing data, preserving feature uniformity through normalization or standardization, engineering features for improved predictive capabilities, and strategically splitting the dataset for effective model training and evaluation. Missing data is identified and managed using techniques such as imputation and exclusion. The next stage is to ensure that the attributes are on comparable scales by using standardization or normalization approaches. The accuracy of fraud detection can be increased by using feature engineering techniques to eliminate pertinent patterns and insights from the data. The dataset is ultimately split into subsets for testing and training, ensuring the integrity of the data by facilitating comprehensive model training on the one hand and objective evaluation on the other and the fraud detection system's trustworthiness.

5.2.3 MODULE 3

In feature engineering, pertinent features are carefully chosen, extracted, and created from credit card transaction data. The objective of this procedure is to identify meaningful trends and connections that can help with the precise identification of fraudulent activity. To distinguish between fraudulent and genuine transactions, elements like transaction amount, merchant type, transaction time, and prior transaction history, for instance, can be chosen and designed. By refining and improving these attributes, methods such as dimensionality reduction, transformation, and interaction term development enable machine learning algorithms to discern between fraudulent and non-fraudulent transactions with greater accuracy. In order to maximize model performance and enhance the system's capacity to reliably and precisely identify credit card fraud, feature engineering is essential to the CCFDS, as shown in figure 5.3.

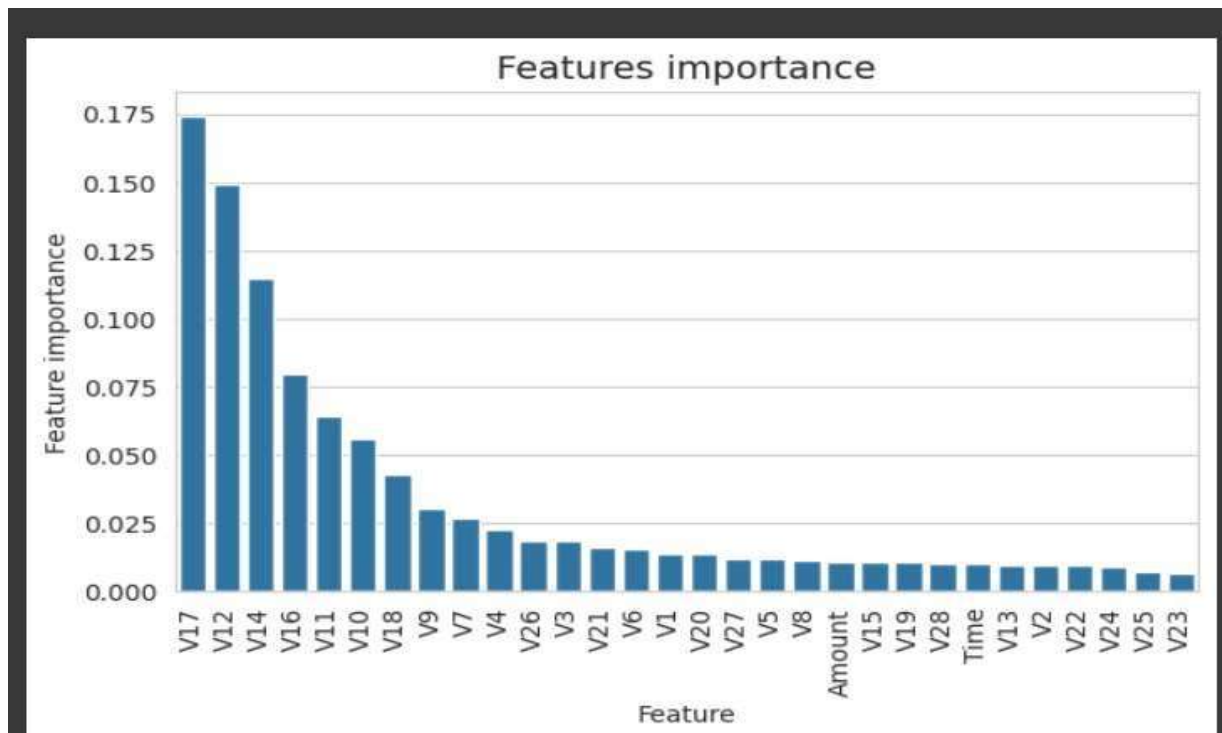


Figure 5.3 : Feature Importance

5.2.4 MODULE 4

Anomaly detection in CCFDS is the process of identifying strange or anomalous patterns in credit card transaction data that deviate from typical behaviour and may indicate fraudulent activity. Improving the system's ability to accurately detect fraudulent transactions requires this process. Anomalies might manifest in a variety of ways, such as unusual transaction volumes, peculiar time stamps, or peculiar spending patterns. Anomaly detection tools, including as statistical methodology, machine learning algorithms, and unsupervised learning approaches, are employed to effectively identify these anomalies, as shown in figure 5.3.

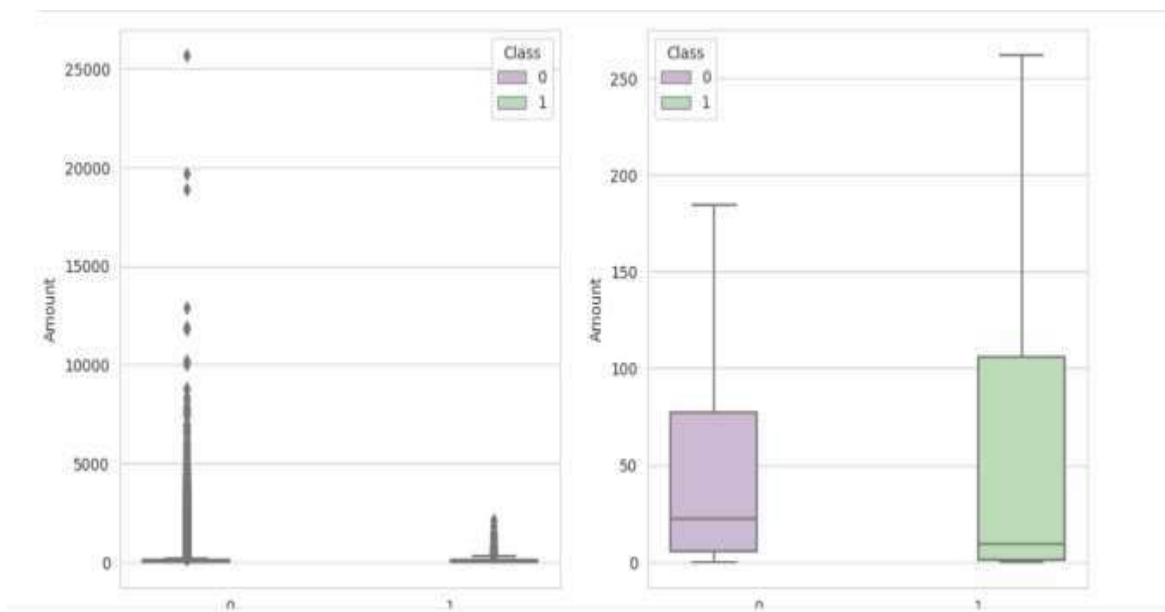


Figure 1.4 : Outliers V/S No Outliers

5.2.5 MODULE 5

Random Forest Classifier :

- Ensemble learning method using multiple decision trees.
- Bagging technique creates subsets of data by sampling with replacement.
- Random feature selection reduces correlation between trees.
- Decision trees built recursively, selecting best features for splits.
- Voting mechanism aggregates tree results for prediction.
- Reduces overfitting by averaging predictions and introducing randomness.
- Handles noisy data and outliers, suitable for high-dimensional data.
- Final prediction is based on the aggregated results from all decision trees.

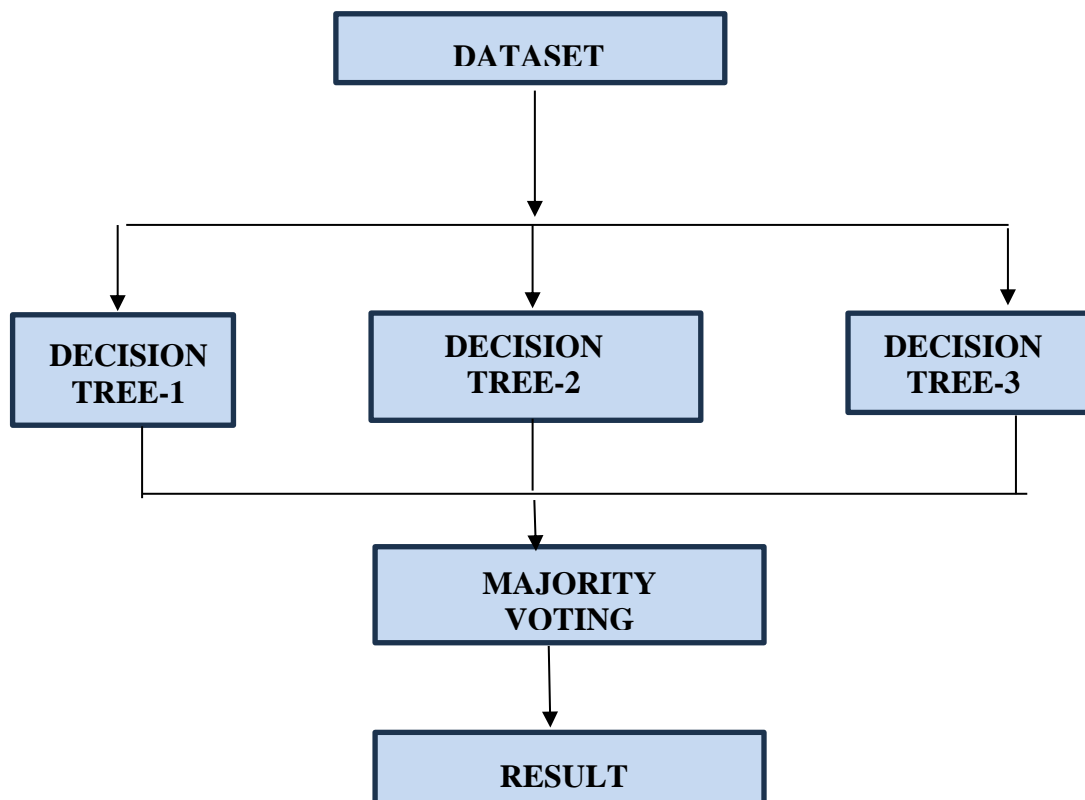


Figure 5.5 : Random Forest Classifier

Logistic Regression Classifier:

- Start: Begin with some initial guesses for the relationship between features and the target outcome.
- Adjust Parameters: Adjust the parameters (weights and bias) based on how wrong our predictions are compared to the actual outcomes.
- Predict Probability: Calculate the probability that each data point belongs to a certain class using a special function called the logistic function.
- Compare to Reality: Compare these predicted probabilities to the actual outcomes, seeing how well we're doing.
- Learn from Mistakes: Adjust the parameters again, trying to minimize the difference between predicted probabilities and actual outcomes.
- Repeat: Keep adjusting and improving our parameters until we're satisfied with our model's performance.

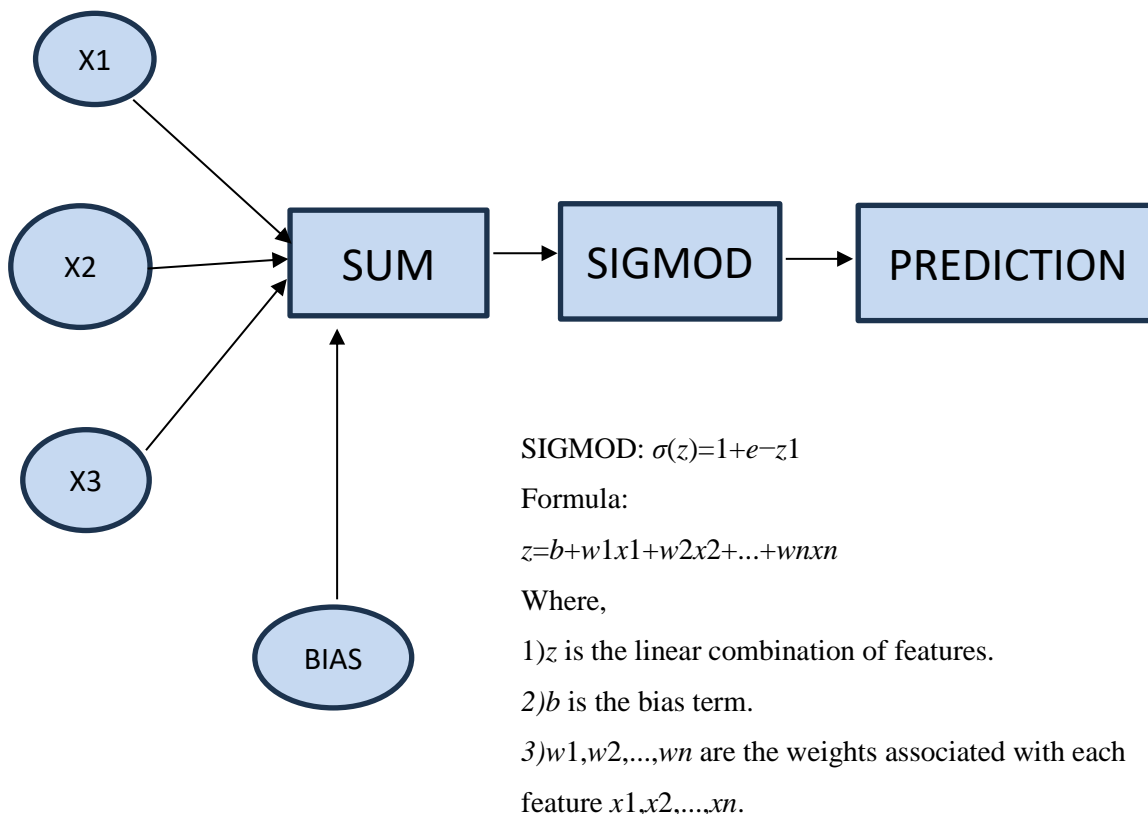


Figure 5.6 : Logistic Regression Classifier

Hybrid Model

A hybrid model that combines the Random Forest Classifier and the Logistic Regression Classifier in the Credit Card Fraud Detection System (CCFDS) provides a reliable technique of recognizing fraudulent transactions. This hybrid model improves overall performance and accuracy in detecting fraudulent activity by using the benefits of both approaches.

The Random Forest Classifier is a powerful tool used in the CCFDS framework, combining multiple decision trees to analyze transaction data. It is adept at handling high-dimensional data and identifying non-linear relationships between features and goal variables, making it an essential component for detecting fraudulent activities.

A helpful tool for modeling linear connections and generating probabilistic outcomes is the logistic regression classifier. When evaluating transaction fraud based on input features, it works especially well. The Logistic Regression Classifier in CCFDS offers insightful information on the importance of individual characteristics.

Enhancing probabilistic modeling and interpretability is a hybrid model that combines the Random Forest Classifier and Logistic Regression Classifier. The model trains on the same dataset, and the classifiers' predictions are integrated through ensemble techniques like voting and averaging. This combination improves the model's ability to detect fraud, as shown in figure 5.7.

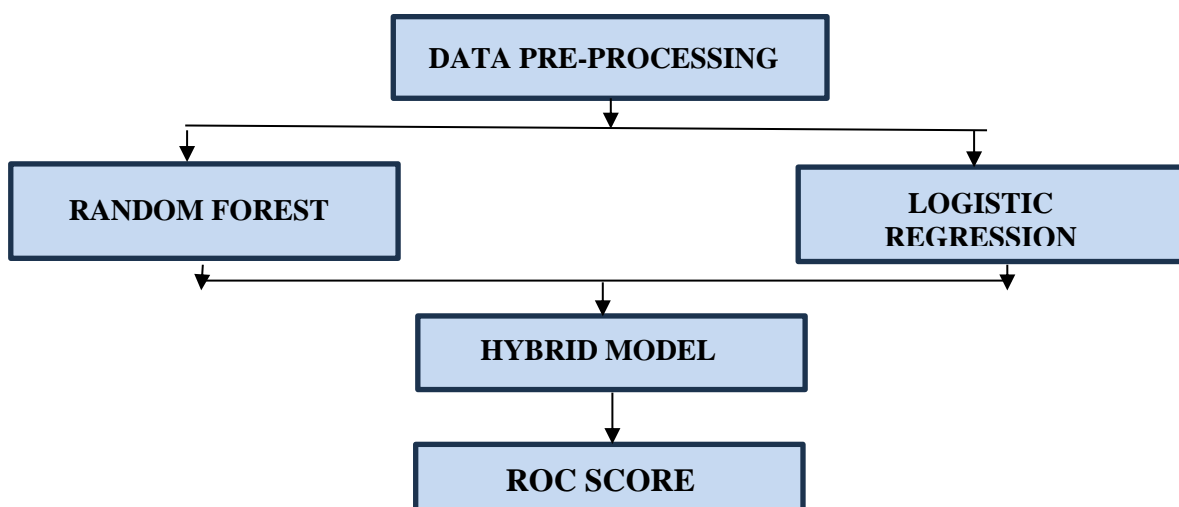


Figure 5.7 : Hybrid Model

5.3 TESTING

5.3.1 Unit Testing

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
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	2.848070e+05	2.848070e+05
mean	94813.859575	1.168375e-15	3.416908e-16	-1.379537e-15	2.074095e-15	9.604066e-16	1.487313e-15	-5.556467e-16	1.213481e-16	-2.406331e-15	2.239053e-15
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	1.098632e+00	1.088850e+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	-1.343407e+01	-2.458826e+01
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-8.915971e-01	-7.682956e-01	-5.540759e-01	-2.086297e-01	-6.430976e-01	-5.354257e-01
50%	84692.000000	1.810880e-02	6.548556e-02	1.798483e-01	-1.984653e-02	-5.433583e-02	-2.741871e-01	4.010308e-02	2.235804e-02	-5.142873e-02	-9.291738e-02
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.431413e-01	6.119264e-01	3.985649e-01	5.704361e-01	3.273459e-01	5.971390e-01	4.539234e-01
max	172792.000000	2.454930e+00	2.205773e+01	9.382550e+00	1.607534e+01	3.480167e+01	7.330163e+01	1.205895e+02	2.000721e+01	1.559499e+01	2.374514e+01

Figure 5.8 : Dataset Description

First five rows of the dataset:

	Time	V1		V2	V3		V4	V5		V6		V7	\
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599					
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803					
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461					
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609					
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941					
		V8	V9	V10	V11	V12	V13	V14	\				
0	0.098698	0.363787	0.090794	-0.551600	-0.617801	-0.991390	-0.311169						
1	0.085102	-0.255425	-0.166974	1.612727	1.065235	0.489095	-0.143772						
2	0.247676	-1.514654	0.207643	0.624501	0.066084	0.717293	-0.165946						
3	0.377436	-1.387024	-0.054952	-0.226487	0.178228	0.507757	-0.287924						
4	-0.270533	0.817739	0.753074	-0.822843	0.538196	1.345852	-1.119670						
		V15	V16	V17	V18	V19	V20	V21	\				
0	1.468177	-0.470401	0.207971	0.025791	0.403993	0.251412	-0.018307						
1	0.635558	0.463917	-0.114805	-0.183361	-0.145783	-0.069083	-0.225775						
2	2.345865	-2.890083	1.109969	-0.121359	-2.261857	0.524980	0.247998						
3	-0.631418	-1.059647	-0.684093	1.965775	-1.232622	-0.208038	-0.108300						
4	0.175121	-0.451449	-0.237033	-0.038195	0.803487	0.408542	-0.009431						
		V22	V23	V24	V25	V26	V27	V28	\				
0	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053						
1	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724						
2	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752						
3	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458						
4	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153						
Amount	Class												
0	140	60	0										

Figure 5.9: Unit Testing- 2

```

Missing values:
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
Class distribution:
Class
0      284315
1        492
Name: count, dtype: int64
All unit tests passed!

```

Figure 5.10 :Unit Testing- 3

5.3.2 INTEGRATION TESTING

```
Classification Report:
              precision    recall  f1-score   support

     0           0.93       1.00       0.96       56746
     1           1.00       0.92       0.96       56980

 accuracy              0.96       113726
 macro avg           0.96       0.96       0.96       113726
 weighted avg       0.96       0.96       0.96       113726
```

Figure 5.11 : Integration Model Testing

6 PROJECT DEMONSTRATION

Data Pre-processing

The first step in developing a thorough credit card fraud detection system is gathering a comprehensive dataset of credit card transactions that includes both fraudulent and true transactions. It takes a lot of data to train machine learning algorithms that can correctly identify fraud. To guarantee excellent data quality, the dataset is subjected to thorough preprocessing after it has been assembled. As shown in figure 6.1, to maintain consistency and comparability across the dataset, this method entails removing noise, such as outliers or inconsistencies, addressing missing values through imputation or row removal, and normalizing features.

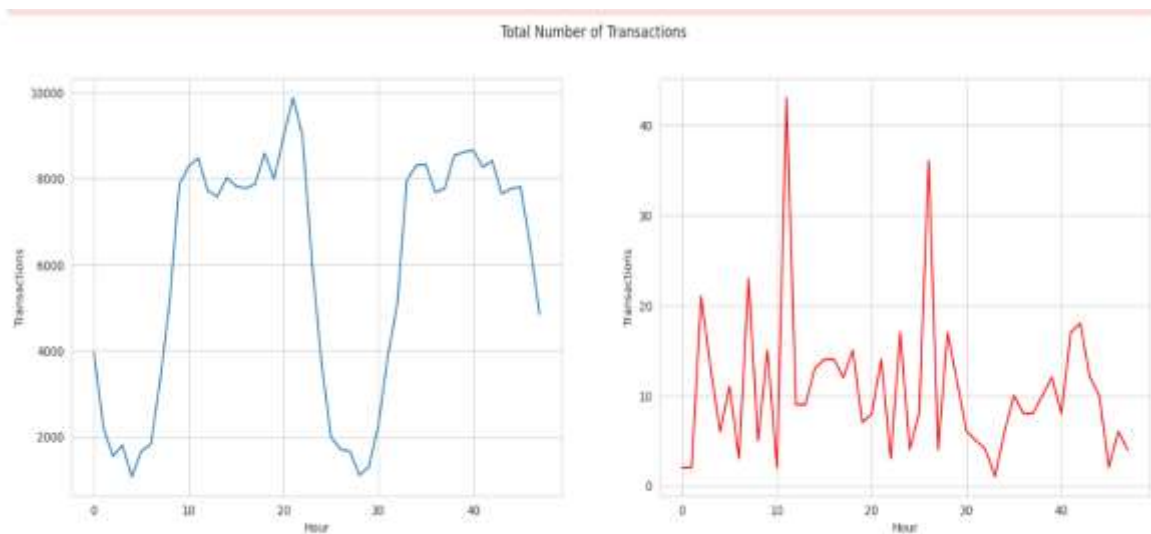


Figure 6.1 : FRAUD V/S NON- FRAUD TRANSACTIONS

The class 0 transactions are shown in the first graph above. After analyzing the first graph's transaction count, it can be concluded that fraudulent transactions tend to happen between 10 and 20 hours of the day, with a very irregular tendency. On the other hand, as the graph illustrates, the non-fraud data is very regular and changes over time.

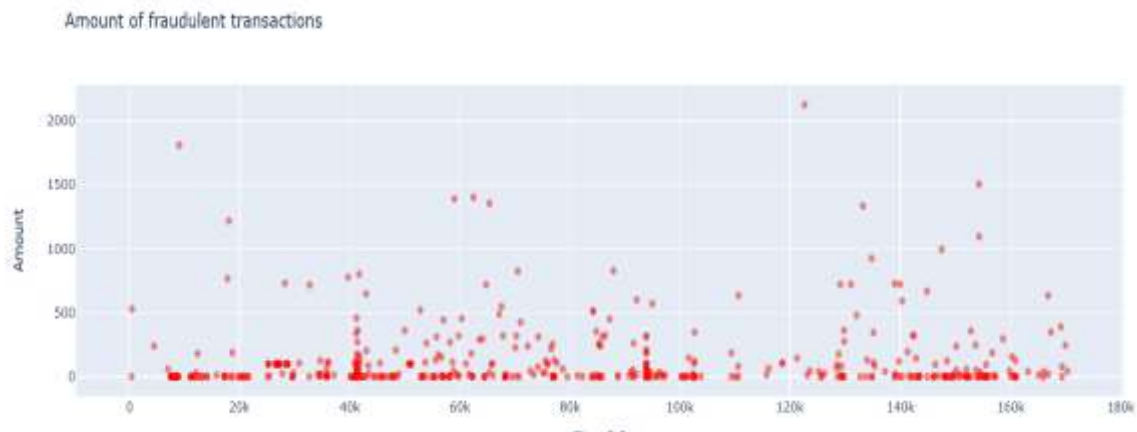


Figure 6.2 : Amount of Fraudulent transactions with respect to time

The volume of fraudulent transactions is shown in the plot above, and it is possible to determine when the fraud takes place most frequently. This may be useful in locating the weak point in the credit card security system. Important trends can be found by examining the distribution of fraudulent transactions.

For instance, if the plot suggests that fraud occurs more frequently on weekends or late at night, this may indicate that fraudsters choose these hours since it is easier to hide their activities and financial institutions may be less watchful during these periods.

Feature Engineering :

Developing machine learning models requires a critical phase called feature engineering, particularly when working with sensitive datasets like credit card transactions. The feature engineering process of the Credit Card Fraud Detection System (CCFDS) requires careful feature extraction and selection that preserves data confidentiality and privacy while enhancing model performance. This is because the system includes confidential features (V0-V28), as shown in figure 6.3.

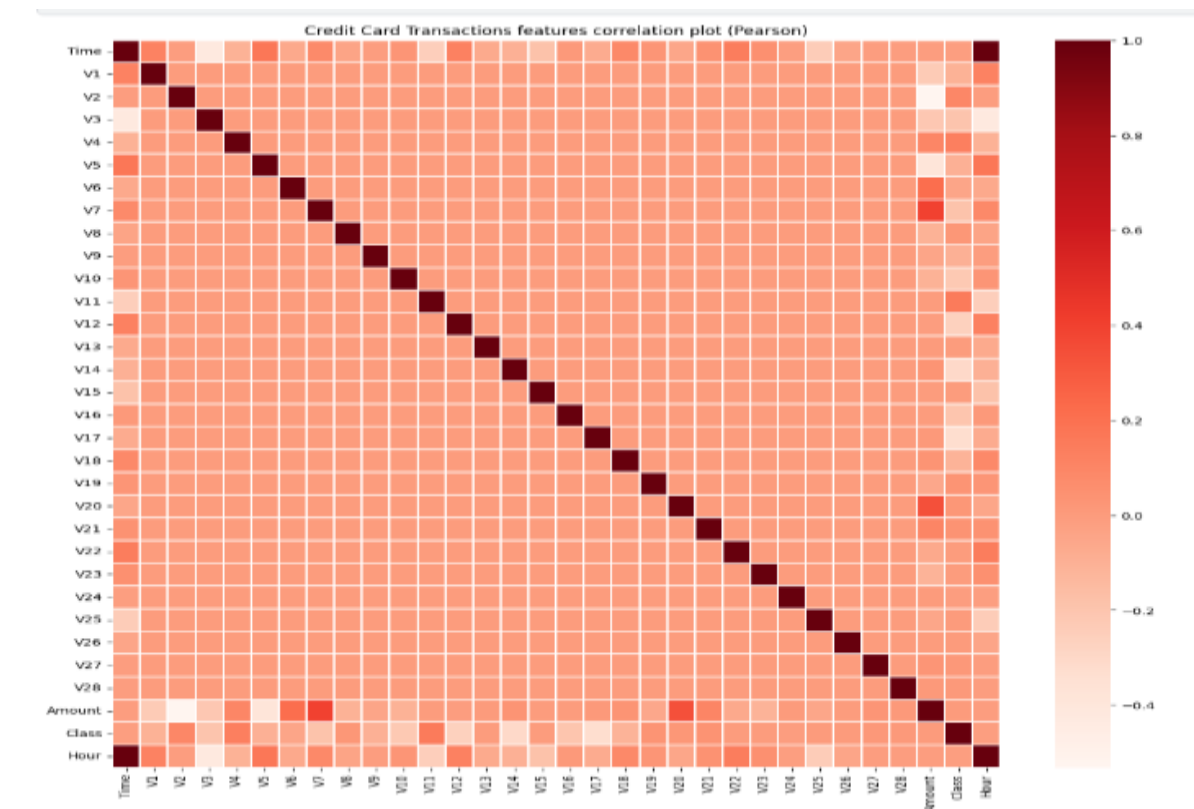


Figure 6.3 : Correlation Plot

Examining the links between the confidential features (V0-V28) and other features like "Amount" and "Time" is made easier with the help of the correlation graphic. The correlation plot's range is 0 to 1, with 1 denoting a perfect linear relationship and 0 denoting no correlation. Darker hues in the color scheme indicate stronger associations, and the usual range is white to maroon.

For instance, the graphic indicating a large correlation between V20 and 'Amount' implies a strong linear link between these two characteristics. This finding may allude to possible trends in financial transactions by showing that V20 increases as the value of 'Amount' does. In a similar vein, a strong correlation between V1 and "Time" denotes a relationship between these two characteristics, which may help identify fraudulent activity by exposing temporal trends

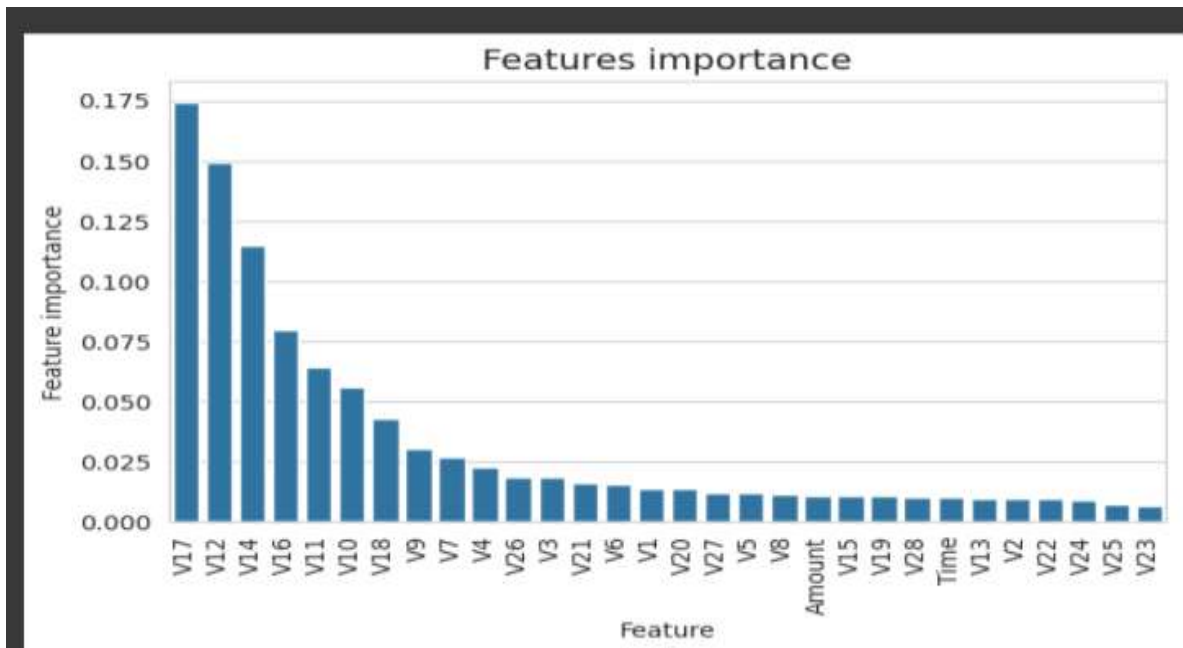


Figure 6.4 : Feature Importance

Using Random Forest to assess feature relevance is a viable method to identify which features are most helpful for detecting fraud without disclosing sensitive data, as features V0-V28 in the CCFDS are confidential. Which features are more predictive in spotting fraudulent transactions are indicated by feature significance, as shown in figure 6.4.

- V17: This attribute is of utmost significance, indicating that it is crucial in differentiating between transactions that are fraudulent and those that are not.
- V12 and V14: These variables come in right behind V17, which suggests they also make a big difference in the model's capacity to identify fraud.

Random – Forest Classifier Code Snippet:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report

# Load the dataset from Kaggle
data = pd.read_csv("/kaggle/input/fraud-detection/creditcard.csv")

# Assuming your target variable is 'Class' and features are all other columns
X = data.drop('Class', axis=1)
y = data['Class']

# Splitting data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Creating and training Random Forest classifier
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Making predictions
y_pred = model.predict(X_test)

# Evaluating the model
conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", conf_matrix)
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# Visualizing the confusion matrix
cm = pd.crosstab(y_test.values, y_pred, rownames=['Actual'], colnames=['Predicted'])

fig, ax1 = plt.subplots(ncols=1, figsize=(5, 5))
sns.heatmap(cm,
            xticklabels=['Not Fraud', 'Fraud'],
            yticklabels=['Not Fraud', 'Fraud'],
            annot=True,
            ax=ax1,
            linewidths=.2,
            linecolor="Darkblue",
            cmap="Blues")
plt.title('Confusion Matrix', fontsize=14)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

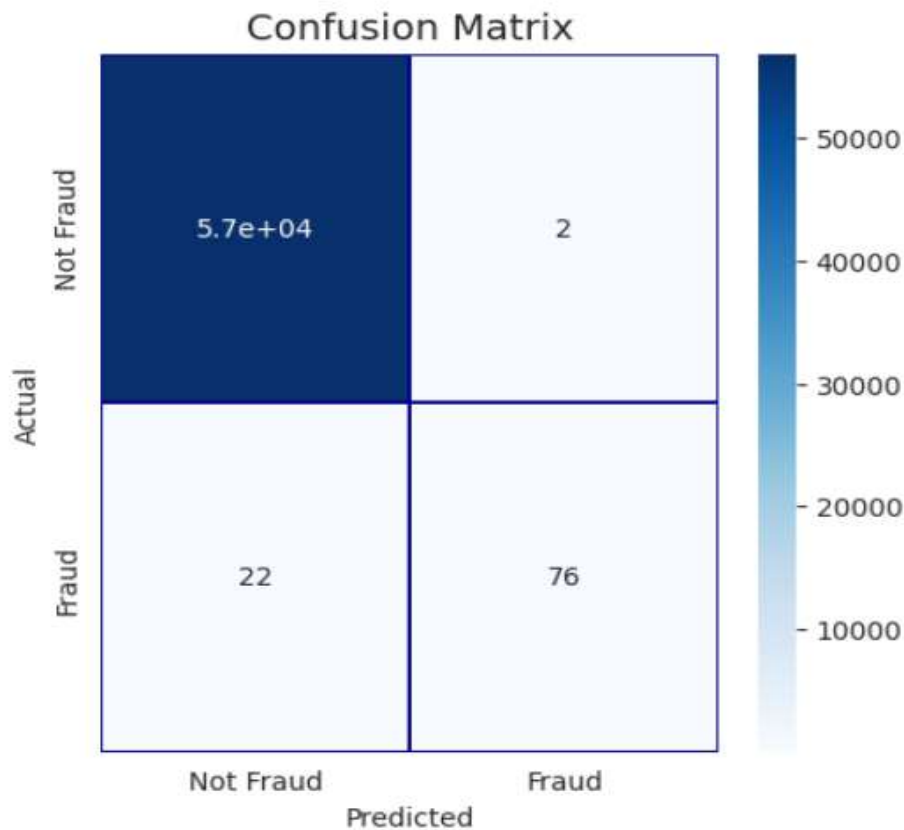


Figure 6.5 : Confusion Matrix of Random Forest Classifier

The efficacy and constraints of a Random Forest classifier for credit card fraud detection are shown by the confusion matrix. 57,000 of the assessed transactions were accurately predicted to be non-fraudulent, indicating the model's capacity to recognize lawful transactions. 22 fraudulent transactions, however, were mistakenly identified as non-fraudulent, suggesting possible fraudulent activity that the system missed. Customers' disruptions and operating expenses were minimized as there were only 2 false positive incidents. The model proved its ability to detect some fraudulent activity by successfully identifying 76 real fraudulent transactions, as shown in figure 6.5. False negatives indicate a trade-off between precision and recall in fraud detection and point to the need to enhance the model's memory. Taking care of this balance is essential to developing a strong and trustworthy system for detecting credit card fraud.

Logistic Regression Code Snippet:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, classification_report

# Load the dataset from Kaggle
data = pd.read_csv("/kaggle/input/fraud-detection/creditcard.csv")

# Assuming your target variable is 'Class' and features are all other columns
X = data.drop('Class', axis=1)
y = data['Class']

# Splitting data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Creating and training logistic regression model
model = LogisticRegression()
model.fit(X_train, y_train)

# Making predictions
y_pred = model.predict(X_test)

# Evaluating the model
conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", conf_matrix)
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# Visualizing the confusion matrix
cm = pd.crosstab(y_test.values, y_pred, rownames=['Actual'], colnames=['Predicted'])

fig, ax1 = plt.subplots(ncols=1, figsize=(5, 5))
sns.heatmap(cm,
            xticklabels=['Not Fraud', 'Fraud'],
            yticklabels=['Not Fraud', 'Fraud'],
            annot=True,
            ax=ax1,
            linewidths=.2,
            linecolor="Darkblue",
            cmap="Blues")
plt.title('Confusion Matrix', fontsize=14)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

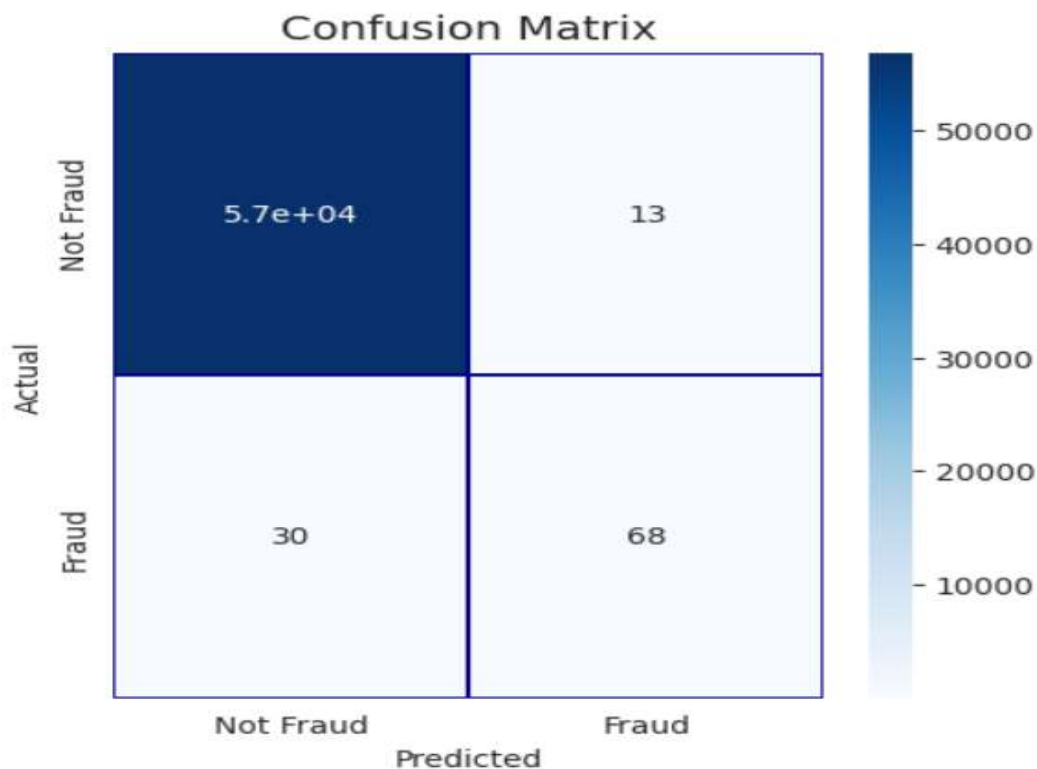


Figure 6.6 : Confusion Matrix of Logistic Regression

The confusion matrix of the logistic regression model shows how well it can identify credit card fraud. With True Negatives (TN) of about 57,000, the model has a high accuracy rate in identifying valid transactions. Nonetheless, 13 instances of False Positives (FP) result from non-fraudulent transactions that were mistakenly identified as fraudulent, creating cardholder annoyance and adding to the effort of fraud investigation teams. The model's capacity to identify fraud is demonstrated by its True Positives (TP) of 68, as shown in figure 6.6. The financial system is at risk, and the model's sensitivity to fraudulent patterns needs to be improved, as 30 False Negatives (FN) represent missed illicit transactions. The high proportion of True Positives and True Negatives in the model indicates a strong basis for fraud detection.

Hybrid Model Code Snippet

```
# Define the models
rf_model = RandomForestClassifier(n_jobs=4, random_state=2018)
lr_model = LogisticRegression()

# Train the models
rf_model.fit(train_df[predictors], train_df[target])
lr_model.fit(train_df[predictors], train_df[target])

# Get predicted probabilities
rf_probs = rf_model.predict_proba(test_df[predictors])
lr_probs = lr_model.predict_proba(test_df[predictors])

# Compute ROC AUC score for Random Forest
rf_roc_auc = roc_auc_score(test_df[target], rf_probs[:, 1])
print(" Score for Random Forest:", rf_roc_auc)

# Compute ROC AUC score for Logistic Regression
lr_roc_auc = roc_auc_score(test_df[target], lr_probs[:, 1])
print(" Score for Logistic Regression:", lr_roc_auc)

# Combine predictions
combined_probs = (rf_probs[:, 1] + lr_probs[:, 1]) / 2

# Compute ROC AUC score for combined predictions
combined_roc_auc = roc_auc_score(test_df[target], combined_probs)
print(" Score for Combined Predictions:", combined_roc_auc)
```

With a score of roughly 0.9436, the Random Forest model proved to be highly accurate in identifying transactions. The remarkable accuracy of this ensemble learning technique, which makes use of numerous decision trees, is attributed to its robustness and capacity to identify intricate links in the data.

Although slightly lower than Random Forest, the Logistic Regression model has good accuracy as well, with a score of roughly 0.9071. Because it is a more straightforward linear model, logistic regression works well in many situations but may have trouble with

more intricate patterns, which could account for the discrepancy in accuracy between the two models.

When the forecasts from the two models were combined, the result was an even higher score of almost 0.9497. By utilizing the advantages of both models, this strategy produces a fraud detection system that is more trustworthy and resilient. The combined forecasts imply that ensemble methods, which incorporate the results of several models, can provide better accuracy than single models.

7 RESULT AND DISCUSSION

➤ Random Forest Classifier Model Evaluation :

Classification Report:				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	56864
1	0.97	0.78	0.86	98
accuracy			1.00	56962
macro avg	0.99	0.89	0.93	56962
weighted avg	1.00	1.00	1.00	56962

Figure 7.1 :Random Forest Model Evaluation

The effectiveness of the Random Forest classifier in identifying credit card fraud is assessed in the classification report. Precision, recall, f1-score, support, overall accuracy, macro average, and weighted average are important metrics. High precision indicates the correctness of the model. Precision is defined as the percentage of true positive predictions among all predictions for a particular class. The recall function quantifies the percentage of actual positives that were true positive predictions out of all actual positives; it is highest for class "0" and lowest for class "1". A flawless score for class "0" and a strong score for class "1" characterize the F1-score, which strikes a balance between recall and precision trade-offs. Support shows how many instances of each class there are in the dataset; an imbalance in support draws attention to this. The total percentage of right answers is represented by accuracy forecasts, a value of 1.00 indicates that nearly every transaction was accurately identified by the algorithm. Recall, f1-score, and macro average precision are 0.89, 0.93, and 0.99, respectively, indicating the overall proficiency in both courses. The high percentage of accurate predictions for the non-fraudulent class is indicative of an excellent performance, as evidenced by the weighted average precision, recall, and f1-score , as shown in figure 7.1.

➤ **Logistic Regression Classifier Model Evaluation :**

Classification Report:				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	56864
1	0.84	0.69	0.76	98
accuracy			1.00	56962
macro avg	0.92	0.85	0.88	56962
weighted avg	1.00	1.00	1.00	56962

Figure 7.2 : Logistic Regression Model Evaluation

With an overall precision, recall, and f1-score of 1.00 for non-fraudulent transactions, the logistic regression model offers a high degree of accuracy in recognizing authentic transactions. It is not perfect at identifying fraudulent transactions, though. The model's precision for fraudulent transactions is 0.71, meaning that 71% of the anticipated fraud cases turned out to be fraudulent. With a recall of 0.69, 31% of real fraud cases were missed by the model. With a f1-score of 0.70, moderate performance is indicated. With a macro average of 0.85, both classes have a balanced point of view. It is clear that the model is good at detecting non-fraudulent transactions, but it has to get better at recalling fraudulent transactions. Increasing the model's sensitivity to fraud trends may result in a credit card fraud detection system that is more dependable and efficient, lowering false positives while preserving a reasonable quantity of false negatives, as shown in figure 7.2.

➤ **Hybrid Classifier Model Evaluation :**

The hybrid model performs well in identifying credit card fraud since it combines the capabilities of Random Forest and Logistic Regression classifiers. With a precision of 0.93 for legitimate transactions and 1.00 for fraudulent ones, it attains good recall and precision levels. The model strikes a balance between precision and recall, effectively capturing real transactions and recognizing the majority of fraudulent cases. The total accuracy of 0.96 shows stability in the accurate classification of big transactions. This model is a dependable option for detecting credit card fraud since it is strong and flexible enough to adjust to different fraudulent transaction patterns.

Classification Report:				
	precision	recall	f1-score	support
0	0.93	1.00	0.96	56746
1	1.00	0.92	0.96	56980
accuracy			0.96	113726
macro avg	0.96	0.96	0.96	113726
weighted avg	0.96	0.96	0.96	113726

Figure 7.3 : Hybrid Classifier Model Evaluation

➤ Model Comparison

Score for Random Forest: 0.9436380183602406
 Score for Logistic Regression: 0.9070838169603601
 Score for Combined Predictions: 0.9497411276423622

Figure 7.4 : ROC Score Comparison

With a ROC score of 0.9436, the Random Forest model is very capable of differentiating between fraudulent and non-fraudulent transactions. This implies that it can adjust to intricate data structures. The Logistic Regression model, on the other hand, has a marginally lower ROC score of 0.9071, indicating that it may not be as successful at capturing non-linear patterns. The benefit of utilizing both models is shown by the ROC score, which increases to 0.9497 when integrated in a hybrid manner. Because the hybrid model decreases false positives and false negatives and better distinguishes across classes, its higher ROC score points to a more reliable approach for credit card fraud detection. All things considered, integrating models into a hybrid method improves discrimination and accuracy, making it a useful tool for identifying credit card fraud, as shown figure 7.4.

8 SUMMARY

The goal of the Credit Card Fraud Detection System (CCFDS) project is to create a dependable and efficient method for spotting fraudulent credit card transactions. With the goal of minimizing false negatives and maximizing recall, this program uses machine learning to identify problematic patterns and eliminate false positives.

The project's dataset comprises a sizable number of credit card transactions, but the ratio of legitimate to fraudulent cases is noticeably off, making model training difficult. The project uses a variety of feature engineering and data preprocessing techniques, including feature scaling, controlling outliers, and oversampling the minority class to balance the data in order to address this. To generate a dataset that is useful for developing precise and trustworthy machine learning models, follow these steps.

Building a hybrid model that blends Random Forest and Logistic Regression classifiers is the project's methodology. Because of its ensemble learning structure, the Random Forest component delivers excellent accuracy and robustness, whereas the Logistic Regression component is more straightforward and effective. By combining the two models, the system can leverage their respective strengths and produce a more resilient solution. Confusion matrices and classification reports are utilized to analyze important performance parameters such as f1-score, precision, and recall for the hybrid model. The hybrid model's high accuracy is attributed to its accuracy in classifying transactions that are not fraudulent. The study does point out places for improvement, though, especially with regard to the recall for fraudulent transactions, which suggests some fraud cases were overlooked. This realization prompts changes to the model's parameters, class weights, and convergence techniques to increase the system's fraud detection capability without producing an unduly high number of false positives.

9 References

- [1] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan and M. Ahmed, "Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms," in *IEEE Access*, vol. 10, pp. 39700-39715, 2022, doi: 10.1109/ACCESS.2022.3166891.
- [2] S. K. Hashemi, S. L. Mirtaheri and S. Greco, "Fraud Detection in Banking Data by Machine Learning Techniques," in *IEEE Access*, vol. 11, pp. 3034-3043, 2023, doi: 10.1109/ACCESS.2022.3232287.
- [3] N. Nguyen et al., "A Proposed Model for Card Fraud Detection Based on CatBoost and Deep Neural Network," in *IEEE Access*, vol. 10, pp. 96852-96861, 2022, doi: 10.1109/ACCESS.2022.3205416.
- [4] W. Ning, S. Chen, S. Lei and X. Liao, "AMWSPLAdaboost Credit Card Fraud Detection Method Based on Enhanced Base Classifier Diversity," in *IEEE Access*, vol. 11, pp. 66488-66496, 2023, doi: 10.1109/ACCESS.2023.3290957.
- [5] M. Alamri and M. Ykhlef, "Hybrid Undersampling and Oversampling for Handling Imbalanced Credit Card Data," in *IEEE Access*, vol. 12, pp. 14050-14060, 2024, doi: 10.1109/ACCESS.2024.3357091.
- [6] Almazroi, A. A. and Ayub, N., "Online Payment Fraud Detection Model Using Machine Learning Techniques", *IEEE Access*, vol. 11, pp. 137188–137203, 2023. doi:10.1109/ACCESS.2023.3339226.
- [7] A. Mniai, M. Tarik and K. Jebari, "A Novel Framework for Credit Card Fraud Detection," in *IEEE Access*, vol. 11, pp. 112776-112786, 2023, doi: 10.1109/ACCESS.2023.3323842
- [8] Zorion, Priyanshu Katiyar and Sachan, Lakshay and Chhabra, Rithik and Pandey, Vishal and Fatima, Dr. Hoor, Credit Card Financial Fraud Detection Using Deep Learning (November 10, 2023).
- [9] F. Ferreira, N. Lourenço, B. Cabral and J. P. Fernandes, "When Two are Better Than One: Synthesizing Heavily Unbalanced Data," in *IEEE Access*, vol. 9, pp. 150459-150469, 2021, doi: 10.1109/ACCESS.2021.3126656.
- [10] M. Mecati, M. Torchiano, A. Vetrò and J. C. d. Martin, "Measuring Imbalance on Intersectional Protected Attributes and on Target Variable to Forecast Unfair Classifications," in *IEEE Access*, vol. 11, pp. 26996-27011, 2023, doi: 10.1109/ACCESS.2023.3252370

- [11] S. Yu, X. Li, X. Zhang and H. Wang, "The OCS-SVM: An Objective-Cost-Sensitive SVM With Sample-Based Misclassification Cost Invariance," in *IEEE Access*, vol. 7, pp. 118931-118942, 2019, doi: 10.1109/ACCESS.2019.2933437
- [12] F. A. Almarshad, G. A. Gashgari and A. I. A. Alzahrani, "Generative Adversarial Networks-Based Novel Approach for Fraud Detection for the European Cardholders 2013 Dataset," in *IEEE Access*, vol. 11, pp. 107348-107368, 2023, doi: 10.1109/ACCESS.2023.3320072
- [13] W. A. Al-Khater, S. Al-Maadeed, A. A. Ahmed, A. S. Sadiq and M. K. Khan, "Comprehensive Review of Cybercrime Detection Techniques," in *IEEE Access*, vol. 8, pp. 137293-137311, 2020, doi: 10.1109/ACCESS.2020.3011259.
- [14] Ali, Jehad & Khan, Rehanullah & Ahmad, Nasir & Maqsood, Imran. (2012). Random Forests and Decision Trees. *International Journal of Computer Science Issues(IJCSI)*. 9.
- [15] Peng, Joanne & Lee, Kuk & Ingersoll, Gary. (2002). An Introduction to Logistic Regression Analysis and Reporting. *Journal of Educational Research - J EDUC RES*. 96. 3-14. 10.1080/00220670209598786.

APPENDIX A – SAMPLE CODE

[1] Dataset :

```
import pandas as pd
import plotly.graph_objs as go

# Calculate class frequencies
class_counts = data_df["Class"].value_counts()

# Create DataFrame
class_df = pd.DataFrame({'Class': class_counts.index, 'Frequency':
    class_counts.values})

# Create bar chart trace
trace = go.Bar(
    x=class_df['Class'],
    y=class_df['Frequency'],
    text=class_df['Frequency'],
    marker=dict(color="dodgerblue"), # Blue color for bars
    opacity=0.7, # Adjust transparency for better visualization
    textposition='outside', # Place frequency values outside bars for better readability
    hoverinfo='x+y', # Display class and frequency on hover
)

# Define layout
layout = go.Layout(
    title='Credit Card Fraud Class Distribution',
    xaxis=dict(title='Class', showticklabels=True),
    yaxis=dict(title='Number of Transactions'),
    hovermode='closest',
    plot_bgcolor='rgba(0,0,0,0)', # Set plot background color to transparent
    width=800,
    height=500,
    margin=dict(l=50, r=50, b=100, t=100), # Adjust margins for better visualization
)

# Create figure
fig = go.Figure(data=[trace], layout=layout)

# Display
fig.show()
```


[2] Maximum Amount Of Transactions

```
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(18,6))
s = sns.lineplot(ax = ax1, x="Hour", y="Max", data=df.loc[df.Class==0])
s = sns.lineplot(ax = ax2, x="Hour", y="Max", data=df.loc[df.Class==1], color="red")
plt.suptitle("Maximum Amount of Transactions")
plt.show();
```

```
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(18,6))
s = sns.lineplot(ax = ax1, x="Hour", y="Median", data=df.loc[df.Class==0])
s = sns.lineplot(ax = ax2, x="Hour", y="Median", data=df.loc[df.Class==1],
                  color="red")
plt.suptitle("Median Amount of Transactions")
plt.show();
```

```
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(18,6))
s = sns.lineplot(ax = ax1, x="Hour", y="Min", data=df.loc[df.Class==0])
s = sns.lineplot(ax = ax2, x="Hour", y="Min", data=df.loc[df.Class==1], color="red")
plt.suptitle("Minimum Amount of Transactions")
plt.show();
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(18,6))
s = sns.lineplot(ax = ax1, x="Hour", y="Min", data=df.loc[df.Class==0])
s = sns.lineplot(ax = ax2, x="Hour", y="Min", data=df.loc[df.Class==1], color="red")
plt.suptitle("Minimum Amount of Transactions")
plt.show();
```

[3] Outliers

```
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12,6))
s = sns.boxplot(ax = ax1, x="Class", y="Amount", hue="Class", data=data_df,
                palette="PRGn", showfliers=True)
s = sns.boxplot(ax = ax2, x="Class", y="Amount", hue="Class", data=data_df,
                palette="PRGn", showfliers=False)
plt.show();
```

[4] Heatmap

```
plt.figure(figsize = (14,14))
plt.title('Credit Card Transactions features correlation plot (Pearson)')
corr = data_df.corr()
sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns, linewidths=.1,
            cmap="Reds")
plt.show()
```

[5] Random Forest Classifier

```
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Making predictions
y_pred = model.predict(X_test)

# Evaluating the model
conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", conf_matrix)
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# Visualizing the confusion matrix
cm = pd.crosstab(y_test.values, y_pred, rownames=['Actual'], colnames=['Predicted'])

fig, ax1 = plt.subplots(ncols=1, figsize=(5, 5))
sns.heatmap(cm,
            xticklabels=['Not Fraud', 'Fraud'],
            yticklabels=['Not Fraud', 'Fraud'],
            annot=True,
            ax=ax1,
            linewidths=.2,
            linecolor="Darkblue",
            cmap="Blues")
plt.title('Confusion Matrix', fontsize=14)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

[6] Logistic Regression Classifier

```
model = LogisticRegression()
model.fit(X_train, y_train)

# Making predictions
y_pred = model.predict(X_test)

# Evaluating the model
conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", conf_matrix)
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# Visualizing the confusion matrix
cm = pd.crosstab(y_test.values, y_pred, rownames=['Actual'], colnames=['Predicted'])
```

```

fig, ax1 = plt.subplots(ncols=1, figsize=(5, 5))
sns.heatmap(cm,
             xticklabels=['Not Fraud', 'Fraud'],
             yticklabels=['Not Fraud', 'Fraud'],
             annot=True,
             ax=ax1,
             linewidths=.2,
             linecolor="Darkblue",
             cmap="Blues")
plt.title('Confusion Matrix', fontsize=14)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

```

[7] Feature Importance:

```

tmp = pd.DataFrame({'Feature': predictors, 'Feature importance':
                    clf.feature_importances_})
tmp = tmp.sort_values(by='Feature importance',ascending=False)
plt.figure(figsize = (7,4))
plt.title('Features importance',fontsize=14)
s = sns.barpplot(x='Feature',y='Feature importance',data=tmp)
s.set_xticklabels(s.get_xticklabels(),rotation=90)
plt.show()

```

[8] Hybrid Model

```

rf_model = RandomForestClassifier(n_jobs=4, random_state=2018)
lr_model = LogisticRegression()

# Train the models
rf_model.fit(train_df[predictors], train_df[target])
lr_model.fit(train_df[predictors], train_df[target])

# Get predicted probabilities
rf_probs = rf_model.predict_proba(test_df[predictors])
lr_probs = lr_model.predict_proba(test_df[predictors])

# Compute ROC AUC score for Random Forest
rf_roc_auc = roc_auc_score(test_df[target], rf_probs[:, 1])
print(" Score for Random Forest:", rf_roc_auc)

# Compute ROC AUC score for Logistic Regression
lr_roc_auc = roc_auc_score(test_df[target], lr_probs[:, 1])
print(" Score for Logistic Regression:", lr_roc_auc)

# Combine predictions
combined_probs = (rf_probs[:, 1] + lr_probs[:, 1]) / 2

```

```

# Compute ROC AUC score for combined predictions
combined_roc_auc = roc_auc_score(test_df[target], combined_probs)
print(" Score for Combined Predictions:", combined_roc_auc)

```

[9] Testing the Model

```

# Check for missing values
missing_values = data.isnull().sum()
print("Missing values:\n", missing_values) # Shows missing values in each column

# Get the distribution of the target variable
class_counts = data["Class"].value_counts() # 0: Non-fraud, 1: Fraud
print("Class distribution:\n", class_counts)

# Unit tests to validate the dataset and data quality
def test_data_loading(data):
    assert not data.empty, "Dataset should not be empty"
    expected_columns = ['Time', 'Amount', 'Class'] + [f'V{i}' for i in range(1, 29)] #
    Expected columns
    assert all(col in data.columns for col in expected_columns), "Missing expected
    columns"

def test_data_quality(data):
    assert missing_values.sum() == 0, "Dataset contains missing values"

def test_class_distribution(data):
    assert class_counts[1] > 0, "No fraudulent transactions found"
    assert class_counts[0] > 0, "There should be at least one non-fraudulent
    transaction"

# Run the unit tests
try:
    test_data_loading(data)
    test_data_quality(data)
    test_class_distribution(data)
    print("All unit tests passed!")
except AssertionError as e:
    print("Unit test failed:", e)

```



Research on machine learning techniques for credit card fraud detection system

Rohan Verma , Rakshit Gupta , Siddhartha Soni | Dr. K. Ragavan | SCOPE

Introduction

The introduction of the project emphasizes the importance of credit card fraud detection and the use of machine learning algorithms, such as Random Forest and Logistic Regression, to develop an effective fraud detection system. The aim is to compare the performance of these algorithms and create a hybrid model for improved accuracy in detecting credit card fraud.

Motivation

The motivation behind this project is to address the increasing threat of credit card fraud and develop a reliable system that can accurately detect and prevent fraudulent transactions, thereby safeguarding the financial interests of people.

SCOPE of the Project

The scope of the project is to design and develop a Credit Card Fraud Detection System (CCFDS) using machine learning techniques. This involves data collection, preprocessing, exploratory data analysis, algorithm selection, feature engineering, training and evaluating machine learning models, and integrating them into the CCFDS framework. The project also focuses on performance optimization, documentation, and reporting to enhance the system's effectiveness in detecting credit card fraud.

Methodology

- Gathered a dataset of credit card transactions and performed data preprocessing and feature engineering.
- Utilized machine learning algorithms such as Random Forest and Logistic Regression for fraud detection.
- Trained and evaluated the models, and created a hybrid model by combining their predictions.
- Compared the performance of the individual models and the hybrid model.

Adapted Methodology

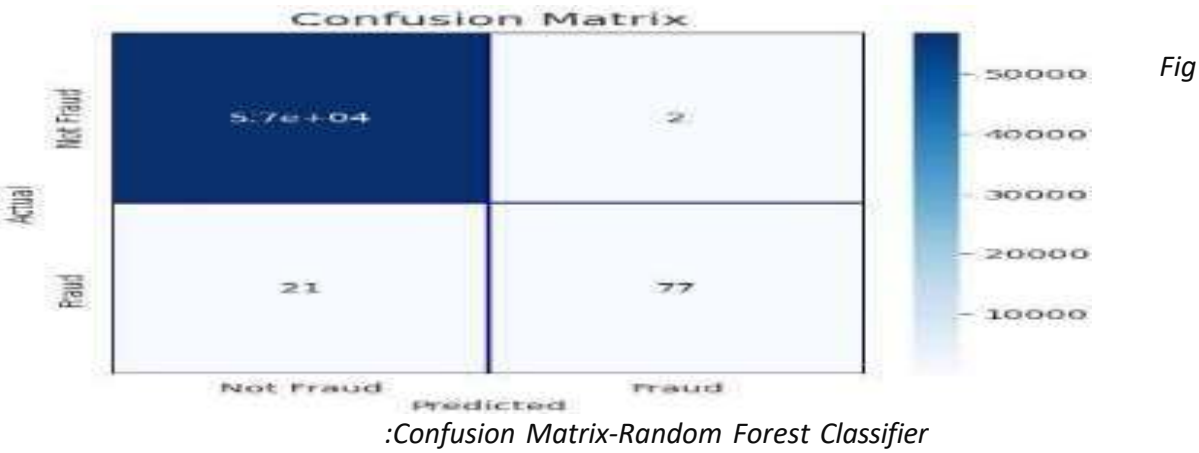
- **EDA:** Analyze transaction distribution for fraud patterns.
 - **Feature Engineering:** Create helpful features from transaction data.
 - **Model Selection:** Choose logistic regression, random forests for fraud.
 - **Training:** Split data, train models for performance assessment.
 - **Evaluation Metrics:** Precision, recall, F1-score, AUC-ROC for assessment.
- ### Modules Description
- **Data Collection:** Gather transactional data from various sources.
 - **Data Preprocessing:** Clean, normalize, and transform raw data.
 - **Feature Engineering:** Select, extract, and create relevant features.
 - **Anomaly Detection:** Identify unusual patterns or outliers.
 - **Machine Learning Models:** Develop and train fraud detection algorithms(combination of Random forest and Logistic Regression)

Results

- Random Forest achieved an accuracy of 99.95% with a precision of 91.89% and recall of 82.65%.
- Logistic Regression achieved an accuracy of 99.92% with a precision of 84.62% and recall of 68.37%.
- The Hybrid Model achieved an accuracy of 99.95% with a precision of 92.31% and recall of 82.65%.
- The Hybrid Model outperformed both individual models.

Classification Report:				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	56864
1	0.97	0.79	0.87	98
accuracy			1.00	56962
macro avg	0.99	0.89	0.93	56962
weighted avg	1.00	1.00	1.00	56962

Fig :Random Forest Classification report



Classification Report:				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	56864
1	0.71	0.69	0.70	98
accuracy			1.00	56962
macro avg	0.85	0.85	0.85	56962
weighted avg	1.00	1.00	1.00	56962

Fig :Logistic Regression Classification Report



Score for Random Forest: 0.9436380183602406
Score for Logistic Regression: 0.9070838169603601
Score for Combined Predictions: 0.9497411276423622

ROC scores

Conclusion

The Credit Card Fraud Detection System (CCFDS) successfully developed a system using machine learning techniques to detect and prevent credit card fraud. The hybrid model, combining Random Forest and Logistic Regression, demonstrated improved performance in detecting fraudulent transactions compared to individual models.

The project highlights the importance of data preprocessing, algorithm selection, and model evaluation in building an effective fraud detection system. The CCFDS has the potential to enhance the security and reliability of credit card transactions, protecting individuals and organizations from financial losses due to fraud.

References

[1] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan and M. Ahmed, "Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms .

[2] S. K. Hashemi, S. L. Mirtaheri and S. Greco, "Fraud Detection in Banking Data by Machine Learning Techniques .

Work- flow diagram

