Credit Card Fraud Detection System

A major project report submitted in partial fulfilment of the requirement for the award of a degree of

Bachelor of Technology

in

Computer Science & Engineering / Information Technology

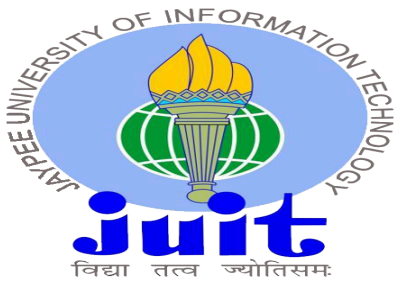
Submitted by

Monisha Surana (201291)

Animesh Singh (201349)

Under the guidance & supervision of

Dr. Rakesh Kanji



Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology,

Waknaghat, Solan - 173234 (India)

CERTIFICATE

This is to certify that the work which is being presented in the project report titled **"Credit Card Fraud Detection System"** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering / Information Technology** submitted to the Department of Computer Science And Engineering, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by **Monisha Surana(201291)** and **Animesh Singh(201349)** during the period from August 2023 to May 2024 under the supervision of **Dr. Rakesh Kanji,**(Assistant Professor(SG), Department of Computer Science and Engineering, Jaypee University of Information Technology).

Monisha S 13/5/24

(Student Signature with Date)

Student Name: Monisha Surana

Roll No.: 201291

13/5/24

(Student Signature with Date)

Student Name: Animesh Singh

Roll No.: 201349

The above statement made is correct to the best of our knowledge.

Rakesh Kanji 13/5/24

(Supervisor Signature with Date)

Supervisor Name: Dr. Rakesh Kanji

Designation: Assistant Professor (SG)

Department: CSE/IT

Dated: 13 May 2024

DECLARATION

We at this moment declare that the work presented in this report entitled **'Credit Card Fraud Detection System'** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering / Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of our work carried out over a period from August 2023 to May 2024 under the supervision of **Dr Rakesh Kanji** (Assistant Professor(SG), Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for any other degree or diploma award.

Monisha S 13/5/24

(Student Signature with Date)

Student Name: Monisha Surana

Roll No.: 201291

13/5/24

(Student Signature with Date)

Student Name: Animesh Singh

Roll No.: 201349

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Rakesh Kanji 13/5/24

(Supervisor Signature with Date)

Supervisor Name: Dr. Rakesh Kanji

Designation: Assistant Professor (SG)

Department: CSE/IT

Dated: 13 May 2024

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing to make it possible to complete the project work successfully.

We are grateful and wish out profound indebtedness to Supervisor **Dr Rakesh Kanji, Assistant Professor(SG)**, Department of CSE Jaypee University of Information Technology, Waknaghat. Deep Knowledge & keen interest of our supervisor in the field of "**Research Area**" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, We want to thank the various staff individuals, both educating and non-instructing, who have developed their convenient help and facilitated our undertaking.

Finally, We must acknowledge with due respect the constant support and patience of our parents.

Monisha Surana
(201291)

Animesh Singh
(201349)

TABLE OF CONTENT

Chapter-5 Results and Evaluation

Chapter-6 Conclusions and Future Scope

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Credit card fraud is a major concern for financial institutions, which put forth significant effort to thwart cybercriminals and swindle artists of all sizes. Historically, credit card fraud was easier to identify because it primarily involved criminals swiping credit cards and making unauthorised purchases at ATMs or retail locations. However, as electronic commerce has proliferated, fraud has grown increasingly sophisticated and complex.

Banks must continually patch vulnerabilities in their payment-transaction systems to ensure that purchasers are legitimate cardholders and thereby prevent deception. In addition, they employ digital tools and alter their fraud detection systems continuously in response to evolving techniques and new data.

A digital platform designed to prevent fraudulent transactions, the Fraud Reporting Exchange (FRX), was recently introduced by the Australian Banking Association (ABA). The ABA, in conjunction with the nation's foremost financial institutions, has identified bank transfers as the prevailing method of payment utilised in fraudulent activities. A 'near real-time reporting' system enabled by the FRX enables banks to rapidly communicate information regarding fraudulent transactions in order to thwart money transfers to criminals.

Credit fraud detection has become predominantly automated and digitised in recent years, due to the exponential growth of data and the surge in electronic payment transactions. When individual instances of credit card fraud are detected, the majority of contemporary solutions utilise machine learning (ML) and artificial intelligence (AI) to power data analysis, predictive modelling, decision-making, fraud alerts, and remediation.

CHAPTER-1

INTRODUCTION

1.1 General Introduction

Credit cards are already prevalent in today's digital banking environment, allowing for fast transactions and providing consumers with never-before-seen ease. On the other hand, the increasing prevalence of this fraud is an enormous challenge for businesses, consumers, and financial institutions. As the financial ecosystem is shaped by technology advancements, fraudsters utilise advanced techniques to take advantage of weaknesses, hence solutions that are both resilient and flexible must be developed. This "Credit Card Fraud Detection System" is a strategic solution to the urgent problem, using cutting-edge technologies to strengthen the barriers against unauthorised financial activity.

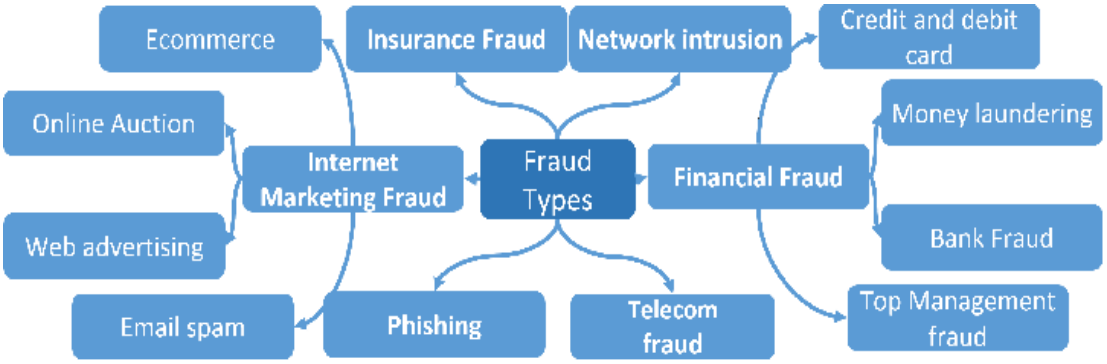


Fig. 1. Taxonomy for Frauds [10]

Credit card fraud can happen if someone physically steals your card or virtually hacks your account, and it can be a significant headache to resolve. If you're a victim of fraud, you may incur unauthorised charges that can result in hefty bills. And if your credit card balance increases drastically, you may risk harm to your credit score.

Thankfully, there are actions you can take now to ward off fraud and identify potential unauthorised use of your card early. The key to safeguarding your credit card information from fraudsters is to remain proactive and on top of your accounts.

One of the most widespread types of fraud crimes in the United States in 2021 was credit card fraud, the responsible federal institution according to FTC records, indexed about 390,000 cases [25]. Nevertheless, these plain and simple figures give only an obvious idea of its complexity.

In December 2022, the payments industry watchdog Nilson Report predicted that over the next 10 years, card fraud losses in the United States will reach $165.1 billion, hitting all age groups and all states. Card-not-present fraud, which includes online, over-the-phone, and mail-order transactions, is the only type of theft that is predicted to cost the United States $5.72 billion in damages in 2022, according to Insider Intelligence.

The most straightforward technique to detect this form of fraud is to examine the spending trends on each card and identify any deviations from the "usual" spending patterns.

The most basic step for credit card fraud protection is to keep the credit cards in a place which is not easily accessible to others. First, make sure that a new credit card kit/envelope is not tampered with, and sign on the back of the card as soon as you receive it.

Always keep the credit card protected in a compact wallet which will make it tough for snatchers or pickpockets. After every purchase, one must never forget to put the card away as soon as possible because thieves can store a digital imprint of the credit card through photos using cell phone cameras. It is also advisable to validate the possession of the credit card in your wallet from time to time, even if you have not used it in a while.

An essential part of this project is real-time monitoring, which makes it possible to check transactions instantly as they happen. It is impossible to overestimate the

significance of real-time fraud detection; prompt discovery enables prompt intervention, stopping illegal activity and reducing possible financial damages. The system is positioned as a proactive and responsive tool in the ongoing fight against this type of fraud because of its real-time transaction processing capabilities.

Any fraud detection system's effectiveness is largely dependent on the user experience. Taking this into account, the project has a user-friendly interface created with Streamlit. In addition to enabling smooth communication, this interface gives end users the ability to take an active role in the process. Docker containerization is used by our system to guarantee that the system is extensible, versatile and the deployment is effective.

At the centre of financial security and technology innovation is this system. With the help of machine learning, real-time monitoring, and a user-friendly interface, the project aims to tackle the credit card fraud problem. It is a proactive step towards creating a more secure digital payment environment, instilling confidence in users and fortifying the foundations of electronic financial transactions.

1.2 Problem Statement

1.2.1 Problem Definition

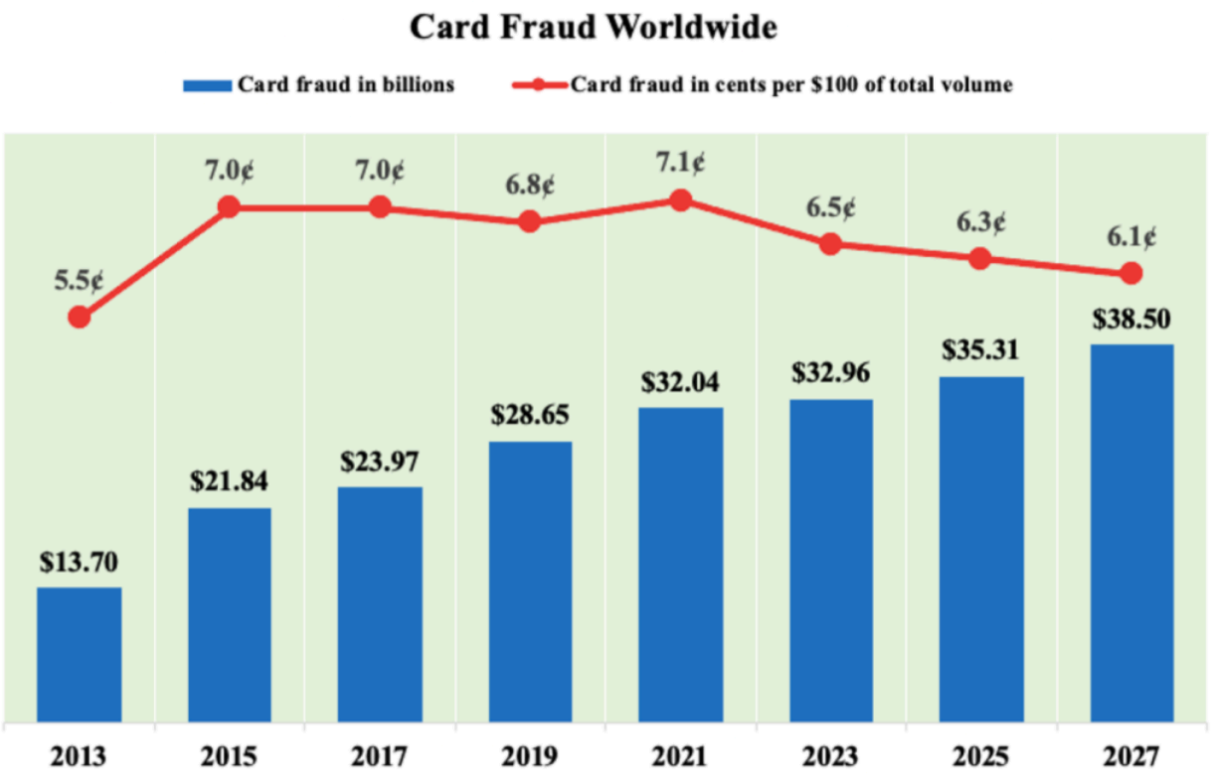


Fig. 2. Card fraud worldwide from 2010 to 2027 [13].

The frequency of this fraud poses a huge danger to financial security, resulting in substantial economic losses and decreased faith in digital transactions. The major purpose of this project is the detection of credit card fraudulent transactions since it's necessary to figure out the fraudulent transactions so that customers don't get charged for the purchase of things that they didn't buy. The detection of credit card fraudulent transactions will be performed with multiple ML techniques then a comparison will be made between the outcomes and results of each technique to find the best and most suited model for the detection of credit card transactions that are fraudulent, graphs and numbers will be provided as well.

1.2.2 Problem Analysis

As a continuous threat to financial transactions, with increasingly complex tactics developed by scammers to evade standard security systems. The key problem consists in precisely identifying fraudulent transactions amidst a massive volume of genuine transactions, ensuring that customers are protected from unauthorised charges and financial institutions can reduce risks effectively. Traditional rule-based systems for fraud detection often fall short in recognising subtle patterns and new fraud trends, underscoring the need for advanced machine learning approaches.

To tackle the issue, this Streamlit App [23] makes use of a machine learning model that is provided as an API via the FastAPI framework. The examination of the situation includes multiple aspects:

- **Scope:** The extent of this fraud is a complex issue that includes a variety of tactics such as identity theft, unauthorised transactions, and card information theft. We deployed machine learning techniques to detect credit card fraud transactions using real datasets. We use these techniques to develop classification utilising machine learning approaches. We uncovered critical criteria that contribute to greater accuracy in detecting credit card mall fraudulent transactions. Also, a user interface is supplied to enable smooth front-end functionality.

- **Need for Real-time Detection:** Real-time detection is necessary because of the seriousness of theft. Conventional approaches frequently fail to quickly respond to new challenges. A successful approach necessitates ongoing observation and prompt detection of questionable transactions. Real-time fraud detection enables organisations to block fraudulent transactions before they are completed, preventing financial losses and lowering the possibility of chargebacks.

- **Utilising Machine Learning for Detection:** Detecting Fraudulent Transactions using Machine Learning: The program makes use of a machine learning model that is meant to identify fraudulent transactions by taking into account important factors including transaction hours, type, amount, and balance fluctuations. The model can differentiate between authentic and fraudulent transactions because of its capacity to learn complex patterns. Machine learning algorithms can spot odd patterns or departures from usual patterns in transactional data. By “training” on past data, the algorithms learn to detect genuine transactions and flag questionable activities that may signal fraud.

- **Frontend Interface with Streamlit:** Users can engage with the model through Streamlit's user-friendly frontend interface. It focuses on the user experience by providing a user-friendly platform for entering transaction information and getting immediate feedback on the likelihood of fraud if it has been conducted.

- **Docker Containerization:** The machine learning model, FastAPI, and Streamlit are all combined and contained within a container because of the use of Docker. This makes deployment easier and reduces compatibility problems by ensuring portability, scalability, and consistency across many environments. Each container operates separately, which indicates that it will not interfere with other containers on the same machine or server. This is crucial to guarantee the security and stability of applications in shared environments.

1.3 Objectives

- **Enhanced Detection Accuracy:** Create and deploy a machine learning model that can recognise complex patterns and abnormalities in transaction data to effectively identify fraudulent credit card transactions, reducing false positives and negatives.
- **Enable Real-Time Processing:** Set up a system that can process credit card transactions instantly so that possible fraud can be quickly identified. The aim is to furnish users and financial institutions with instantaneous feedback, facilitating expeditious measures to avert illicit activities.
- **Less manual work required:** Improved accuracy leads to less work for analysts. Even in a small bank, people are unable to manually check every transaction, says AltexSoft data science competence head Alexander Konduforov. "ML-driven systems filter out, roughly speaking, 99.9 percent of normal patterns leaving only 0.1 percent of events to be verified by experts."
- **Create a User-Friendly Interface:** Utilising Streamlit, provides an intuitive and user-friendly interface that helps users to interact with the system with ease. Features like input areas for transaction details and a place to show the results of real-time fraud detection are included in this.
- **Ensure Scalability with Docker:** To package the machine learning model, FastAPI framework, Streamlit interface, and dependencies into a container, use Docker containerization. Each container operates separately, which indicates that it will not interfere with other containers on the same machine or server. This is crucial to guarantee the security and stability of applications in shared environments.

1.4 Significance and Motivation of the Project Work

Beyond the domains of technology, the Credit Card Fraud Detection System project is significant because it tackles an increasingly important issue in the financial sector. Credit card fraud has spread like wildfire, affecting people, companies, and financial institutions all over the world as a result of the increase in digital transactions. To strengthen the security of electronic payments through the use of cutting-edge machine-learning algorithms, this project is extremely important. The project's creation of a real-time detection system that can quickly identify and stop fraudulent transactions helps to reduce financial losses and protect the integrity of digital transactions.

Due to the accessible interface created with Streamlit, users can actively engage in the fraud detection process. Moreover, the project's scalability are improved by using Docker for containerization, which allows it to be adjusted to a variety of computer settings. Since electronic transactions are still a necessary component of modern life, the effective use of this credit card fraud detection system not only shields people and organisations from financial loss but also promotes confidence in the dependability and security of online payment methods. The initiative is significant because it has the potential to reduce fraud, improve financial security, and aid in the development of a more secure and reliable digital financial environment.

Motivation of the Project:

- **Mitigating Financial Losses:** By providing a strong solution to protect financial assets, the project aims to reduce the major financial losses that people and companies have gone through as a result of credit card theft and fraud.
- **Adapt to Evolving Fraud Tactics:** Driven by the ongoing evolution of fraudulent strategies, the project aims to utilise cutting-edge machine learning techniques to keep ahead of quick and proactive defence mechanisms employed by fraudsters.
- **Real-Time Fraud Detection and Intervention:** The goal is to develop a system

that can identify fraudulent transactions in real-time, take prompt action to reduce their impact, and send out alerts when fraud is detected.

- **Maintaining the reputation of businesses and organisations:** By accommodating such systems in an organisation, they will be able to display commitment and will be able to build trust among the customers.
- **Building User Trust and Confidence:** The project's motivation is to increase user trust and confidence in digital transactions by empowering people with an intuitive interface. Driven by the aim of optimising the user experience, the system facilitates user participation in the fraud detection process, hence cultivating a feeling of confidence in electronic payments.
- **Reduction of operational cost:** Detecting and preventing fraud in the early stages reduces costs incurred by the company associated with investigating and resolving fraudulent transactions.

1.5 Organisation

This project report is divided as follows:

Chapter 1

The project is introduced in brief in this chapter. It covers the difficulties brought forth by financial transaction fraud in a concise and short manner. The project's goals and objectives are described; they include improving detection accuracy, facilitating real-time processing etc.

Chapter 2

Relevant studies that research and give surveys and insight on various ML algorithms used for the system are stated in the chapter. It explores ongoing research and understanding on the system. It also provides a thorough analysis of traditional and modern approaches to fraud detection, highlighting the function of machine learning. By reviewing various research papers along with their strengths and weaknesses we gather insights on how to go about in our research of the project.

Chapter 3

This chapter focuses on system development and goes over important and crucial points. It delves into feature engineering in order to improve data quality. The way API will be working with our project. The architecture of the project has a strong design that ensures reactivity. The technology stack, which includes FastAPI and Docker, python working together to produce a dynamic and secure fraud detection system.

Chapter 4

This chapter describes the overall design work that has been completed as well as how we've tracked it at every stage. It provides information on the work completed in different areas as well as results at different situations. It offers details on the model that we developed with the help of several modules and packages of Python.

Chapter 5

The results of this system are provided in the results and evaluation chapter. Performance metrics for the machine learning model, namely using random forest and logistic regression, in addition to algorithms like XGBoost, are detailed, including accuracy, precision, recall, and F1-score. The discussion of user comments on the Streamlit interface illuminates the user experience. Additionally, the outcomes of the system's deployment via Docker are emphasised, highlighting the benefits and possible difficulties that may have arisen throughout the process.

Chapter 6

An overview of significant discoveries and accomplishments is given in the concluding chapter. Future enhancements are included. It sums up how well the system accomplished its goals. Difficulties faced during the project are recognised, providing a fair and unbiased viewpoint providing us ideas for future developments. The chapter concludes with a discussion of the project's future scope and recommendations for improvements, new features, and directions for future study.

CHAPTER-2

LITERATURE SURVEY

2.1. Overview of Relevant Literature

1. Vaishnavi Nath Dornadula

- Created a new approach to streaming transaction data fraud detection.
- overcome difficulties with datasets containing card fraud by machine learning methods.
- Customers were grouped according to transactions to identify patterns in behaviour.
- Used several classifiers to forecast fraud on client groups.

Method Used:

- Clustering method to group cardholders based on transaction amount.
- Sliding-Window method to aggregate transactions and extract behavioural patterns.

Limitations:

- Accuracy and precision are not ideal parameters for model evaluation.
- Concept drift and data imbalance are challenges in fraud detection.

2. A. Mahajan, V. S. Baghel and R. Jayaraman

- Created a new approach to streaming transaction data fraud detection.
- overcome difficulties with datasets containing card fraud by machine learning methods.
- Customers were grouped according to transactions to identify patterns in behaviour.

- Used several classifiers to forecast fraud on client groups.

Method Used:

- Clustering method to group cardholders based on transaction amount.
- Sliding-Window method to aggregate transactions and extract behavioural patterns.

Limitations:

- Accuracy and precision are not ideal parameters for model evaluation.
- Concept drift and data imbalance are challenges in fraud detection.

3. W. -F. Yu and N. Wang

- The paper focuses on fraud detection using outlier mining methods based on distance sum.
- Increased accuracy using anomaly detection techniques to find odd transaction patterns.
- Can be used when the abnormalities are few.

Method Used:

- Outlier mining method used to detect anomalous transactions.
- Distance sum is calculated to quantify the abnormality according to their feature distance.

Limitations:

- Accuracy is mostly reliable on the quality and consistency of the input data.
- The interpretability of the outlier method can create challenges.

4. Baker Al Smadi, Manki Min

- The paper focuses on reviewing existing techniques for fraud detection, aiming to compare strengths and weaknesses.
- Many ML algorithms and statistical methods which are commonly used are evaluated.
- Emphasises on the need for continuous improvement.

5. Sangeeta Mittal, Shivani Tyagi

- The paper focuses on reviewing existing techniques for fraud detection to find the best one.
- Assesses the performance of Logistic Regression, Decision Tree, Random Forest, Support Vector Machine.
- Random Forest performs superiorly as compared to other algorithms.

6. John Richard D. Kho, Larry A. Vea

- The paper focuses on fraud detection using transaction behaviour analysis to identify suspicious patterns.
- Focuses on transactional frequency,amount of time, time of day.

Method Used:

- Feature extraction is done from transaction data such as frequency, time.
- Uses a clustering algorithm to group the transactions.

Limitations:

- Handling imbalanced datasets where fraud is rare compared to real ones.

7. Y. Sahin, E. Duman

- In this study, the main focus lies on classification models based on Artificial Neural Networks (ANN) and Logistic Regression (LR).

- One of the first studies to compare the performance of ANN and LR.
- The ANN and LR classifier models are developed in this work with the pertinent modelling techniques integrated into IBM SPSS Clementine 12.

Method Used:

- LR is used for binary classification to differentiate between legitimate and illegitimate transactions.
- The study uses a sigmoid function to calculate the probability that the transaction is fraud or not.

TABLE II. TRAINING AND TEST SET SIZES FOR EACH SAMPLE

Samples		# of Records	
		Training Set	Test Set
1F-to-1N	Normal	681	292
	Fraud	681	292
1F-to-4N	Normal	2723	1168
	Fraud	681	292
1F-to-9N	Normal	6130	2626
	Fraud	681	292

Fig. 3. Results [7].

8. Thanh Thi Nguyen, Hammad Tahir, Mohamed Abdelrazek

- In this study, authors proposed a DL-based method to detect credit card fraud.
- The research was created using CNNs to extract spatial features.
- The DL models are trained on labelled data.

Method Used:

- The model is trained on labelled data and LSTMs are employed to

model sequential patterns.

Limitations:

- The major limitation of the paper is that the data required is quite large in volume.
- They are usually considered black-box models which lack transparency.

9. Saurabh C. Dubey, Ketan S. Mundhe, Aditya A. Kadam

- ANN along with backpropagation were used in this study.
- ANN trained with backpropagation showed promising results.

Method Used:

- Backpropagation supervised learning technique is used to train ANNs by adjusting the data's weights and biases.

Limitations:

- Neural Networks can be attacked, where intruders manipulate input data.
- They are usually considered black-box models which lack transparency.

V. PROPOSED SYSTEM

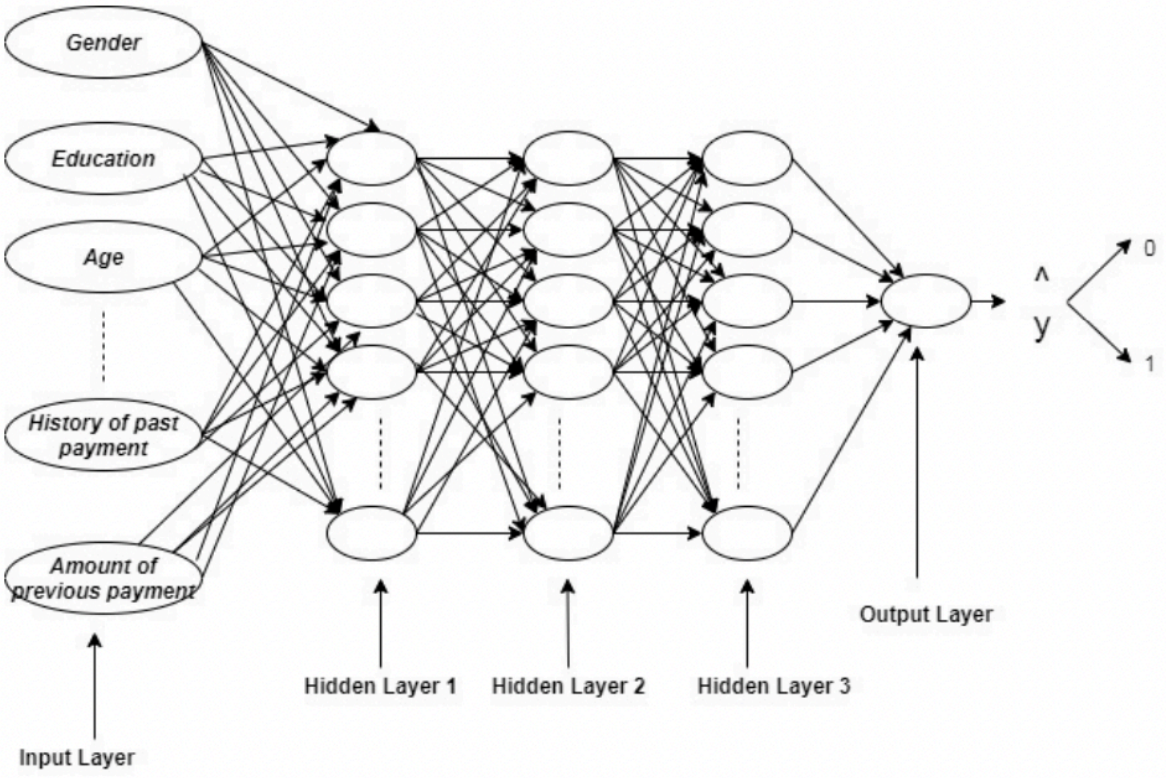


Fig. 4. Architecture of ANN[9].

Sno.	Author Name	Algorithm/Dataset used	Results
1	[1] Vaishnavi Nath Dornadula	Random Forest Logistic Regression One class SVM	One class SVM works better.
2	[2] Anish Mahajan	Logistic Regression	The logistic regression model was able to achieve an accuracy of 94% in detecting fraud.
3	[3] Wen-Fang Yu	Outlier mining algorithm based on distance sum	Outlier mining can detect fraud better than anomaly detection based on clustering.
4	[4] Baker Al Smadi	Neural Networks Decision Tree Hidden Markov Model	Fraud detection techniques aim to detect and prevent fraudulent transactions.

5	[5] Shivani Tyagi	Andrea Dal Pozzolo	Unsupervised algorithms handle the dataset skewness in better ways. Their performance metrics are better.
6	[6] John Richard D. Kho	Random Tree J48	J48 has lower accuracy than random trees.
7	[7] Y. Sahin	Artificial Neural Network Logistic Regression	Artificial Neural Network classifiers are better than Logistic Regression.
8	[8] Aya Abd El Naby	Multi-layer Perceptron Convolution Neural Network	The OSCNN model gives 0.989 accuracy after applying MLP and SMOTE together.
9	[9] Saurabh C. Dubey	Artificial Neural Networks Backpropagation	Using ANN and BackPropagation technique, which gives an accuracy rate of 99.6%.

Table 2: Literature Survey

2.2. Key Gaps in the literature

- The performance of the proposed classifier suffers in terms of response time.
- Outlier Mining only works when anomalies are far less than normal data.
- The Logistic Regression Model is expensive and has low accuracy. It is not efficient for huge datasets.
- Random Tree accuracy decreases in the succeeding dataset variations.
- Cost-based predictions are not taken care of in ANN and logistic regression models.
- Imbalanced dataset can be handled by one-class classifiers like SVM.
- Normal MLP algorithm of Deep Learning gives just 0.88 accuracy.

CHAPTER-3

SYSTEM DEVELOPMENT

3.1 Requirements and Analysis

3.1.1 Requirements

- Security Card Analysis: The main task is to identify the potential business potential.
- Instant processing: The system must work immediately to prevent fraud.

3.1.2 Non-Functional Requirements

- Accuracy: Ensure high accuracy in fraud detection to reduce false positives and negatives.
- Scalability: Make sure the system can handle the growing business.
- Latency: Keep processing time low for in-flight detection.
- Security: Protect customers' sensitive information and secure structure to prevent attacks.

3.2 Project design and architecture

3.2.1 System architecture

- Microservice architecture: Use microservices for modular and scalable development.
- Contents: including preliminary data, model design, model training, time estimation and other models.

3.2.2 Technology Stack

- Machine learning algorithms: Choose an appropriate framework

for building models (**Scikit-learn**)[22].

- Streamlit: Create interactive web interfaces to view and monitor data. Streamlit is an open-source Python framework for **building machine learning and data science websites**. Using Streamlit we can instantly create web applications and distribute them easily. Streamlit allows you to write applications just like Python code. Streamlit turns files into shared web applications in minutes. You don't need any experience because everything is done in pure Python. Streamlit provides seamless interaction between coding and displaying results in a web application. Scikit-learn, Keras, PyTorch, SymPy (latex), NumPy, pandas, Matplotlib etc. It is compatible with major Python libraries such as.

- FastAPI: Build fast, efficient APIs for instantaneous predictions. FastAPI is a web framework for creating RESTful APIs in Python. FastAPI is based on Pydantic and input instructions for data validation, parsing, removal, and automatic generation of OpenAPI files. It supports asynchronous programming and can work with Uvicorn and Gunicorn. FastAPI is a modern and fast (productive) framework for creating APIs using Python 3.8, based on the standard Python programming language.

  Key features are:

  - Fast: Very efficient using NodeJS and Go (thanks to Starlette and Pydantic). It is one of the fastest Python frameworks available.
  - Fast Coding: Create features approximately 200% to 300% faster. * Fewer errors: Reduce human (developer) errors by approximately 40%. *
  - Intuition: Supports powerful editor. It's been done everywhere. Reduce debugging time.
  - Simple: Designed to be easy to use and learn. Reduce time spent reading information.

- Summary: Reduce code duplication. Each parameter declaration has various functions. Fewer errors.
- Robust: Get the developer code. Contains automatic interactive information.
- Standards-based: The API is based on and fully compatible with the open standards OpenAPI (formerly Swagger) and JSON Schema.

- Docker: Store applications for easy deployment and scalability. Docker is an open-source container platform. It allows us to package applications into containers. This is a complete process that combines the application source code with the operating system (OS) libraries and dependencies required to run the code in the environment. What is Docker-Compose? Docker Compose is a tool designed to define and share multiple containers. Compose allows you to create YAML archives to define your services and translate everything up and down with a single command.

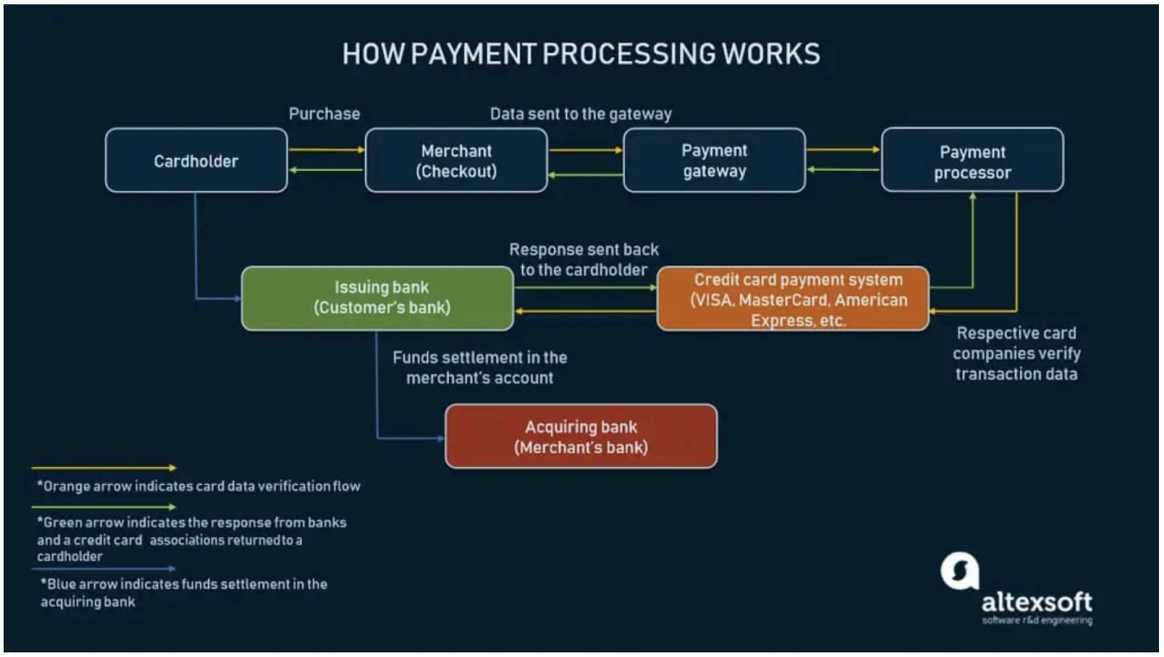


Fig. 5 Process of payment[9].

3.3 Data preparation

Attribute name	Description
Transaction id	Identification number of a transaction
Cardholder id	Unique Identification number given to the cardholder
Amount	Amount transferred or credited in a particular transaction by the customer
Time	Details like time and date, to identify when the transaction was made
Label	To specify whether the transaction is genuine or fraudulent

Table 2: Raw attributes of card transactions

3.3.1 Data collection

- There is no publicly available information regarding financial services, especially in the case of mobile money transfers. Financial information is very important to many researchers, especially those of us researching to detect fraud. Part of the problem is that financial markets are private, which means no public information is available. Synthetic data created using a simulator called PaySim was used as data to create the models used in this project. PaySim uses data collected from personal data to create synthetic data similar to traditional transactions and inject bad behaviour after evaluating the effectiveness of the method of fraud investigation.

3.3.2 Data Analysis

```
df = pd.read_csv("/content/drive/MyDrive/ML-Credit-Card-Fraud/creditcard.csv")
df.head()
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0

5 rows × 31 columns

```
df.describe()
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	
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	2.848070e+05	2.848070e+05	2.848070e
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	...	1.654067e-16	-3.568593e-16	2.578648e-16	4.473266e
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	...	7.345240e-01	7.257016e-01	6.244603e-01	6.056471e
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	...	-3.483038e+01	-1.093314e+01	-4.480774e+01	-2.836627e
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01	-7.682956e-01	-5.540759e-01	-2.086297e-01	-6.430976e-01	...	-2.283949e-01	-5.423504e-01	-1.618463e-01	-3.545861e
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02	-2.741871e-01	4.010308e-02	2.235804e-02	-5.142873e-02	...	-2.945017e-02	6.781943e-03	-1.119293e-02	4.097606e
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01	3.985649e-01	5.704361e-01	3.273459e-01	5.971390e-01	...	1.863772e-01	5.285536e-01	1.476421e-01	4.395266e
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01	7.330163e+01	1.205895e+02	2.000721e+01	1.559499e+01	...	2.720284e+01	1.050309e+01	2.252841e+01	4.584549e

8 rows × 31 columns

Fig. 6: Data analysis of dataset

3.3.3 Data preprocessing

- The data is also inconsistent. Only 0.17% of transactions are considered fraudulent. When working with arbitrary data, you can do several things:
    - Synthetic Minor Oversampling Technique (SMOTE)
    - Undersampling

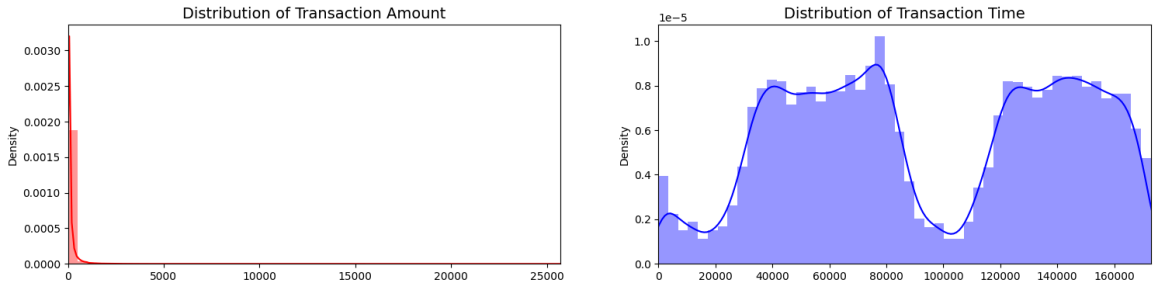


Fig. 7: Data Distribution

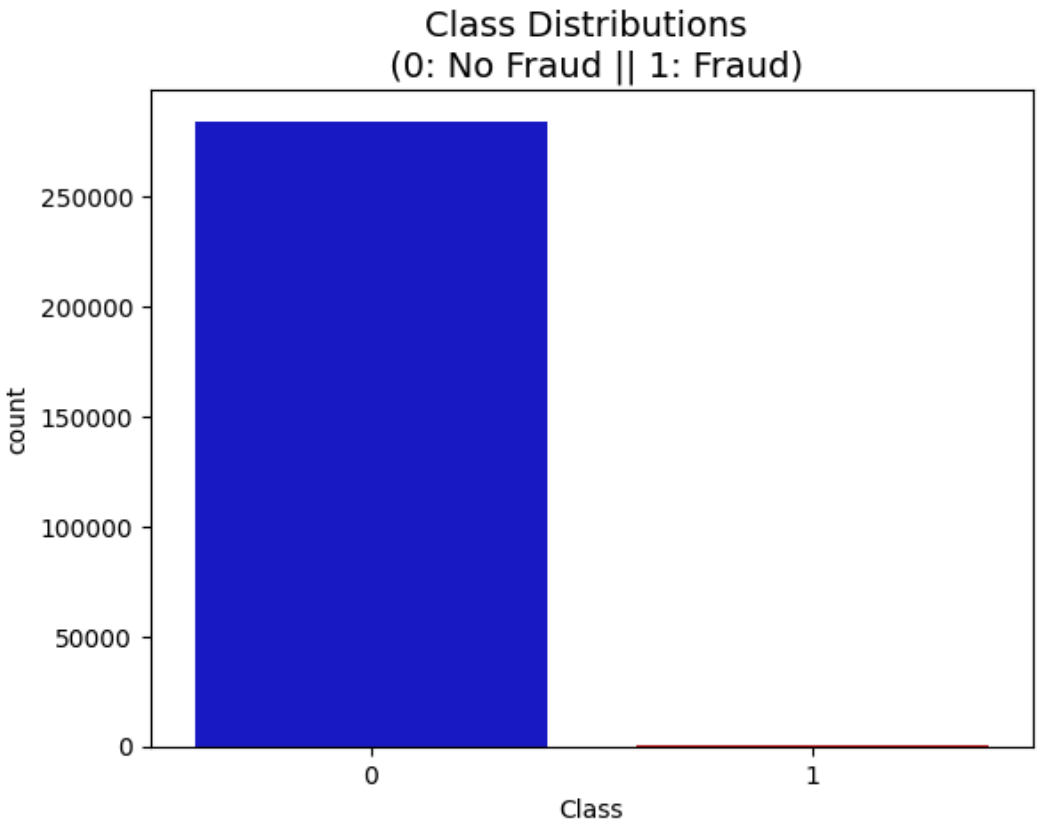


Fig. 8: Class Distribution

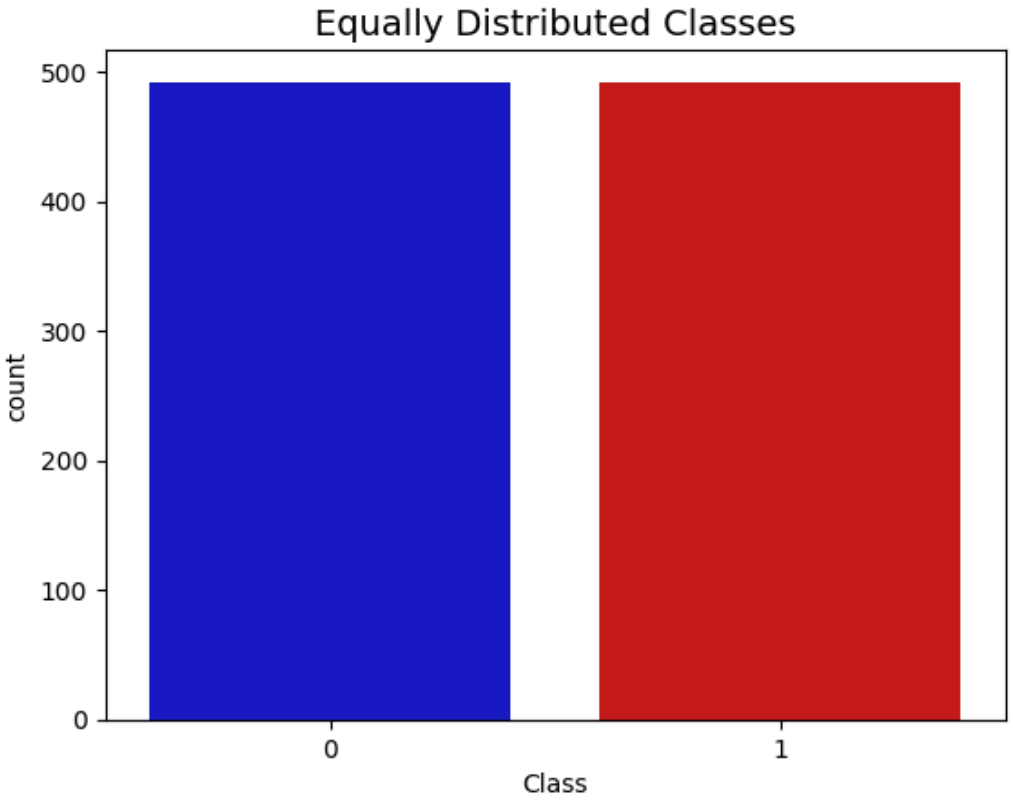


Fig. 9: Class Distribution

3.3.4 Correlation Matrices

- Correlation matrices are the essence of understanding our data. We want to know if there are features that influence heavily on whether a specific transaction is a fraud. However, we must use the correct data frame (subsample) for us to see which features have a high positive or negative correlation concerning fraudulent transactions.
- Summary and Explanation:
    - Negative Correlations: V17, V14, V12 and V10 are negatively correlated. Notice how the lower these values are, the more likely the result will be a fraudulent transaction.
    - Positive Correlations: V2, V4, V11, and V19 are positively correlated. Notice how the higher these values are, the more likely the result will be a fraudulent transaction.
    - BoxPlots: We will use boxplots to have a better understanding of the distribution of these features in fraudulent and non-fraudulent transactions.

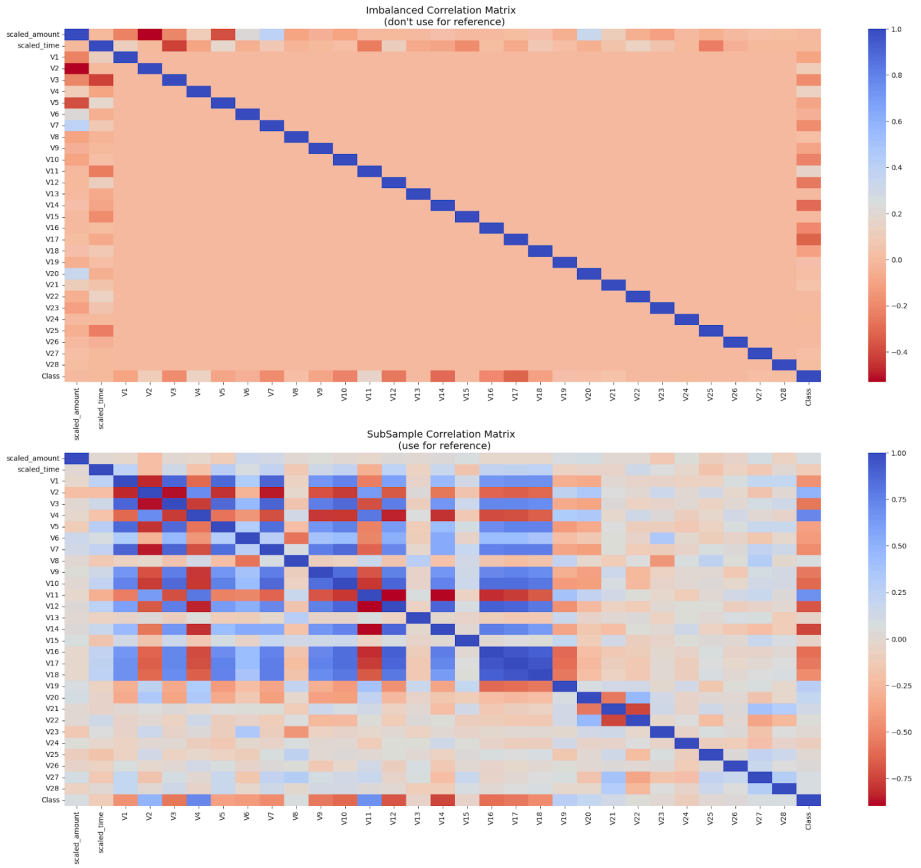


Fig. 10: Correlation Matrices

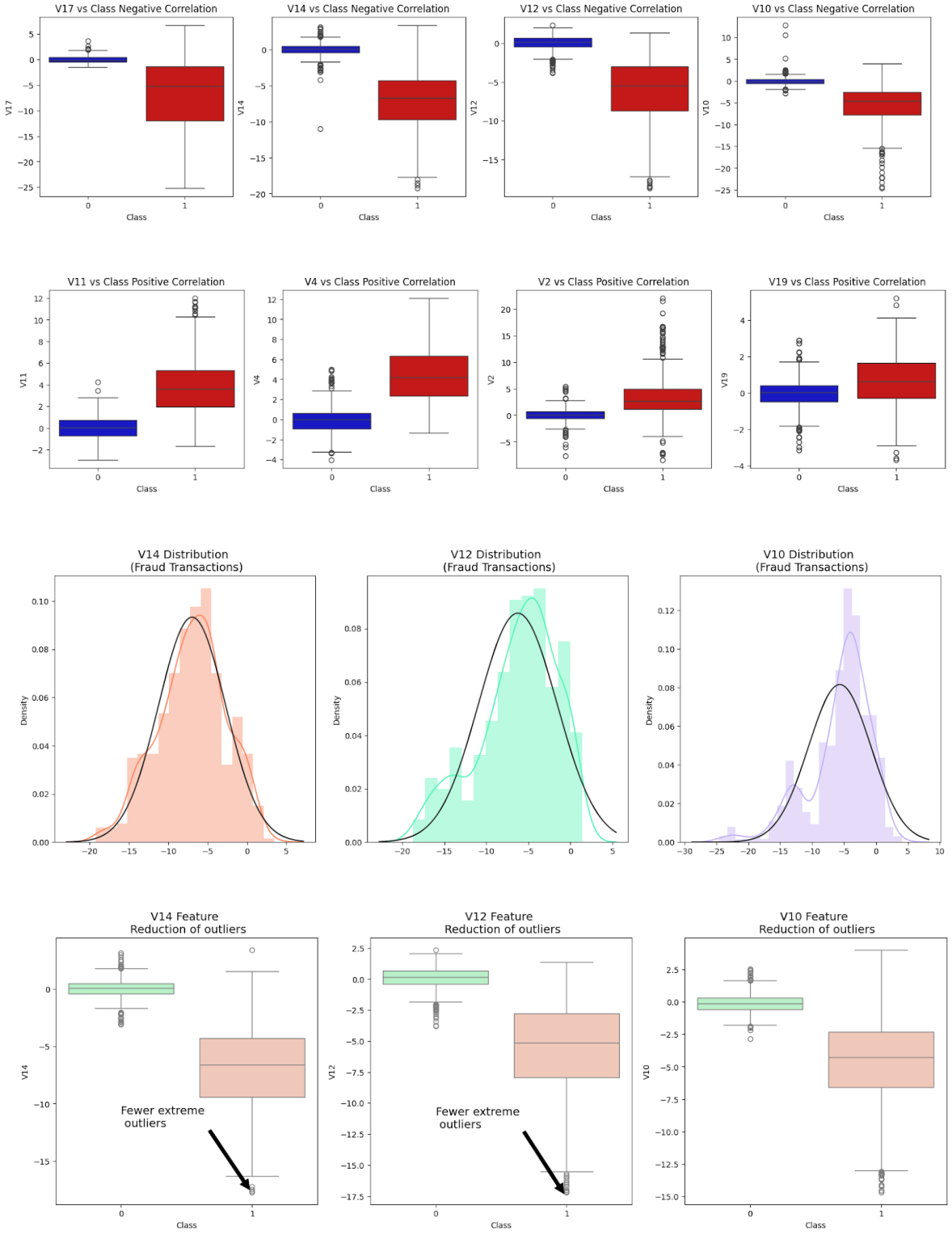


Fig. 11: Positive and Negative Correlation.

Dimensionality Reduction and Clustering: Understanding t-SNE: To understand this algorithm you have to understand the following terms: Euclidean Distance Conditional Probability Normal and T-Distribution Plots

3.4 Implementation

3.4.1 Algorithms

```
classifiers = {
    "LogisiticRegression": LogisticRegression(),
    "Support Vector Classifier": SVC(),
    "DecisionTreeClassifier": DecisionTreeClassifier(),
    "RandomForestClassifier": RandomForestClassifier()
}

from sklearn.model_selection import cross_val_score

for key, classifier in classifiers.items():
    classifier.fit(X_train, y_train)
    training_score = cross_val_score(classifier, X_train, y_train, cv=5)
    print("Classifiers: ", classifier.__class__.__name__, "Has a training score of",
round(training_score.mean(), 2) * 100, "% accuracy score")
```

3.4.2 Model Training

```
from sklearn.model_selection import GridSearchCV

# Logistic Regression
log_reg_params = {"penalty": ['l1', 'l2'], 'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000]}

grid_log_reg = GridSearchCV(LogisticRegression(), log_reg_params)
grid_log_reg.fit(X_train, y_train)
# We automatically get the logistic regression with the best parameters.
log_reg = grid_log_reg.best_estimator_

# Random Forest Classifier
rf_params = {'n_estimators': [100, 200, 300], 'max_depth': [2, 4, 6, 8], 'min_samples_leaf': [5, 10,
15]}
grid_rf = GridSearchCV(RandomForestClassifier(), rf_params)
grid_rf.fit(X_train, y_train)
# RF best estimator
rf_clf = grid_rf.best_estimator_

# Support Vector Classifier
svc_params = {'C': [0.5, 0.7, 0.9, 1], 'kernel': ['rbf', 'poly', 'sigmoid', 'linear']}
grid_svc = GridSearchCV(SVC(), svc_params)
grid_svc.fit(X_train, y_train)

# SVC best estimator
svc = grid_svc.best_estimator_

# DecisionTree Classifier
tree_params = {"criterion": ["gini", "entropy"], "max_depth": list(range(2,4,1)),
               "min_samples_leaf": list(range(5,7,1))}
grid_tree = GridSearchCV(DecisionTreeClassifier(), tree_params)
grid_tree.fit(X_train, y_train)

# tree best estimator
tree_clf = grid_tree.best_estimator_
```

3.4.3 Streamlit Dashboard

Streamlit Dashboard

```python
import streamlit as st
import json
import requests as re

st.sidebar.header('Input Features of The Transaction')

sender_name = st.sidebar.text_input("""Input Sender ID""")
receiver_name = st.sidebar.text_input("""Input Receiver ID""")
step = st.sidebar.slider("""Number of Hours it took the Transaction to complete: """)
types = st.sidebar.subheader(f"""
                Enter Type of Transfer Made:\n\n\n\n
                0 for 'Cash In' Transaction\n
                1 for 'Cash Out' Transaction\n
                2 for 'Debit' Transaction\n
                3 for 'Payment' Transaction\n
                4 for 'Transfer' Transaction\n""")
types = st.sidebar.selectbox("",(0,1,2,3,4))
x = ''
if types == 0:
    x = 'Cash in'
if types == 1:
    x = 'Cash Out'
if types == 2:
    x = 'Debit'
if types == 3:
    x = 'Payment'
if types == 4:
    x =  'Transfer'
```

3.4.4 FastAPI for Real-time Prediction

FAST API

```python
from fastapi import FastAPI, File, Query, UploadFile, HTTPException, Form
from fastapi.responses import FileResponse, PlainTextResponse
import uvicorn
import joblib
import numpy as np
from pydantic import BaseModel


app = FastAPI(
    title="Credit Card Fraud Detection API",
    description="""An API that utilises a Machine Learning model that detects if a credit card """
    """transaction is fraudulent or not based on the following features: hours, amount, transaction type etc.""",
    version="1.0.0",
    debug=True
)

model = joblib.load('credit_fraud.pkl')

@app.get("/", response_class=PlainTextResponse)
async def running():
  note = """
Credit Card Fraud Detection API 🙌
Note: add "/docs" to the URL to get the Swagger UI Docs or "/redoc"
  """
  return note

favicon_path = 'favicon.png'
@app.get('/favicon.png', include_in_schema=False)
async def favicon():
    return FileResponse(favicon_path)

class fraudDetection(BaseModel):
    step:int
    types:int
    amount:float
    oldbalanceorig:float
    newbalanceorig:float
    oldbalancedest:float
    newbalancedest:float
    isflaggedfraud:float
```

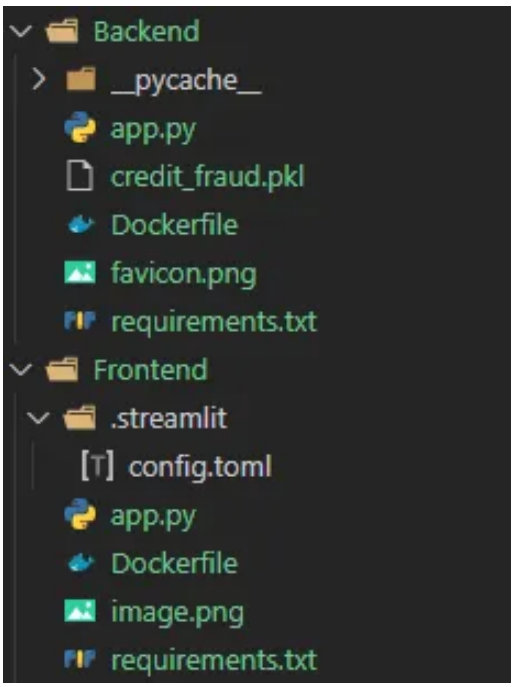
3.4.5 DockerFile

Dockerfile

```
WORKDIR /app

# Copy your application code and model file
COPY . /app

# Install dependencies
RUN pip install --no-cache-dir scikit-learn==1.2.2 fastapi uvicorn[standard]

# Expose the port your FastAPI application will run on
EXPOSE 8000

# Command to run the FastAPI application
CMD ["uvicorn", "app:app", "--host", "0.0.0.0", "--port", "8000"]
```

Dockerfile

```
# Use an official Python runtime as a parent image
FROM python:3.8

# Set the working directory to /app
WORKDIR /app

# Copy the current directory contents into the container at /app
COPY . /app

# Install any needed packages specified in requirements.txt
RUN pip install --no-cache-dir -r requirements.txt

# Make port 8501 available to the world outside this container
EXPOSE 8501

# Run app.py when the container launches
CMD ["streamlit", "run", "app.py"]
```



Project Structure

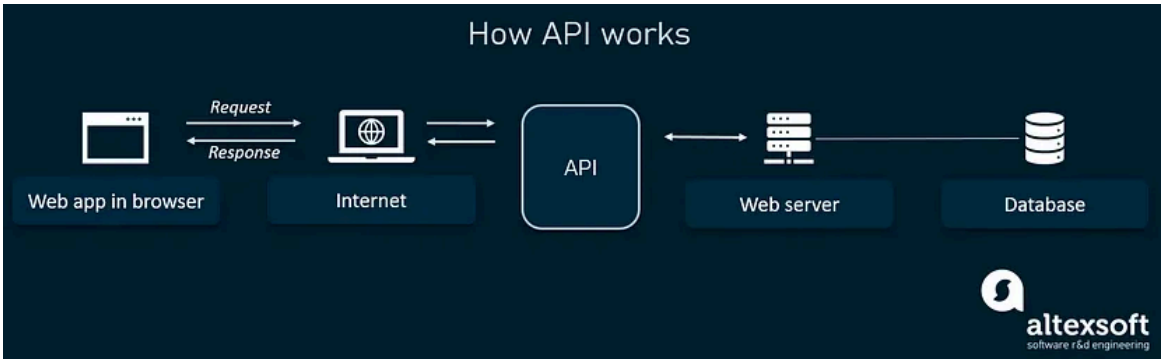


Fig. 12: Working of API

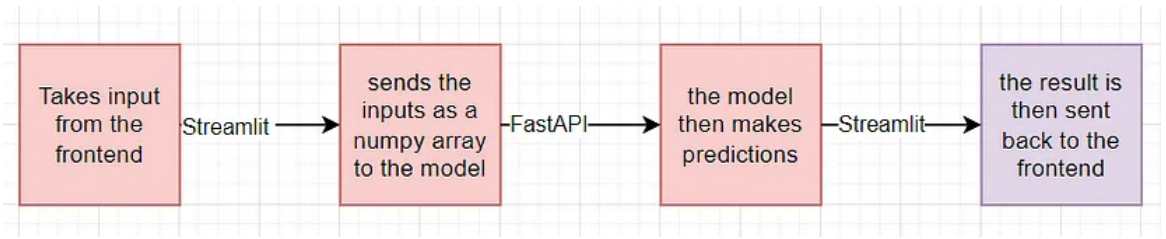


Fig. 13: Data Transfer between API and Streamlit Application

3.5 Key Challenges

3.5.1 Unbalanced dataset

Solution: Use strategies such as oversampling the minority or undersampling most classes to balance the dataset.

3.5.2 Features highly responsible for detection

Challenge: Understanding features that highly contribute towards classification.

Solution: Use of Correlation Matrices with subsample.

3.5.3 Model Interpretation

Challenge: Understand and explain the decision model.

Solution: Use standard interpretation (e.g. SHAP values) and provide transparency.

3.5.4 Privacy Policy

Competition: Manage Customer Information.

Solution: Use data anonymisation techniques and ensure compliance with privacy laws.

3.5.5 Model Maintenance

Challenge: Keep the model in sync with the fraud model change.

Solution: Have continuous learning, and constantly introduce new products into the model.

The credit card authentication development is summarised here. ML, Streamlit, Docker, and FastAPI for the process (requirements, design), and problems faced during implementation and at the construction stage.

CHAPTER-4 TESTING

4.1 Testing Strategy

4.1.1 Unit Testing

- Objective: Authenticate different materials, functioning and models.
- Tools: Employ an open-source testing framework such as unittest or pytest for the Python framework.
- Implementation: Develop tests and train for preliminary data, prototype design, and model training.

4.1.2 Integration Testing

- Objective: Ensure they complement each other well.
- Tools: Use test frameworks, ML models and API testing tools
- Implementation: Ensure that Streamlit, FastAPI, and the machine learning model for testing are smoothly interacting.

4.1.3 End-to-End Testing

- Objective: Validate the entire system's functionality, from data input to real-time fraud prediction.
- Tools: Use automated testing frameworks like Selenium or custom scripts.
- Implementation: Simulate user interactions with the web interface, entering transaction data and verifying the model's predictions.

4.1.4 Performance Testing

- Objective: Evaluate the system's responsiveness and resource usage under different loads.
- Tools: Utilise tools like Apache JMeter or locust.io.
- Implementation: Measure response times for real-time predictions and assess the application's scalability.

4.1.5 Security Testing

- Objective: Identify and address potential security vulnerabilities.
- Tools: Perform security audits using tools like Invicti and Acunetix.
- Implementation: Ensure that the system is resistant to common security threats, such as SQL injection, cross-site scripting or BYOD.

4.2 Test Cases and Outcomes

4.2.1 Unit Test Cases

- Test Case 1: Data Preprocessing

  Input: Raw transaction data

  Expected Outcome: Processed data without missing values or anomaly detection.

- Test Case 2: Model Training

  Input: Training data

  Expected Outcome: Trained model with acceptable accuracy and Precision.

4.2.2 Integration Test Cases

- Test Case 3: Streamlit and FastAPI Integration

  Input: Simulated user interactions with the model

  Expected Outcome: Web interface and API seamlessly working together.

4.2.3 End-to-End Test Cases

- Test Case 4: Real-time Prediction

  Input: Simulated transactions

  Expected Outcome: The model provides accurate fraud predictions in a real-time manner.

4.2.4 Performance Test Cases

- Test Case 5: Response Time Under Load

  Input: Concurrent user interactions

  Expected Outcome: The application maintains acceptable response times under varying loads.

4.2.5 Security Test Cases

- Test Case 6: Input Validation

  Input: Malicious input data

  Expected Outcome: The system rejects malicious input and maintains integrity.

4.2.6 Outcomes

- Successful Outcomes: All tests pass without errors, indicating the system's reliability and correctness.
- Issues Detected: Document any bugs, performance bottlenecks, or security vulnerabilities, along with proposed solutions.
- Performance Metrics: Record response times and resource usage metrics under different test scenarios.

Regularly conduct automated testing as part of the continuous integration and deployment (CI/CD) [21] pipeline to ensure the ongoing reliability of the system and better performance. Adjust test cases as the system evolves and new features are added.

CHAPTER-5

RESULTS AND EVALUATION

5.1 Analysis of system developed

5.1.1 Logistic Regression

Logistic Regression[19] is a statistical technique for estimating the likelihood of binary events. It estimates the probability of a relationship between a dependent variable and one or more independent variables using a logistic function, making it useful for classification tasks in machine learning.

To predict the outcome of a transaction, we will divide our dataset into testing and training sets and use the Logistic Regression function from the Python package.

The figure represents the different scores of LR against varying training sizes.

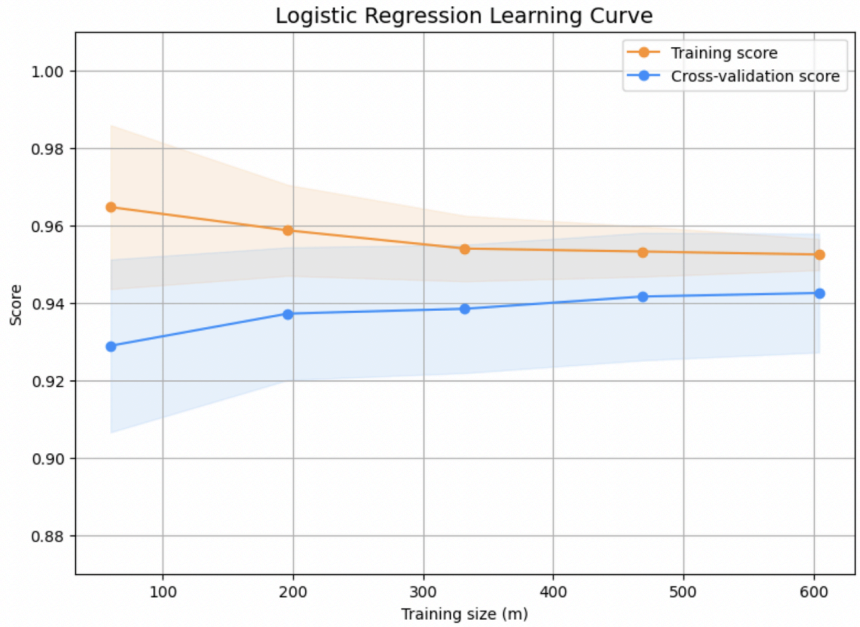


Fig. 14: Logistic Regression Learning Curve

5.1.2 Random Forest Classifier

A collective learning approach called a Random Forest Classifier[20] builds a large number of decision trees during training and outputs the mode of the classes for classification issues. By combining predictions from several trees, it reduces overfitting and increases accuracy, making it reliable and efficient for a range of machine-learning applications.

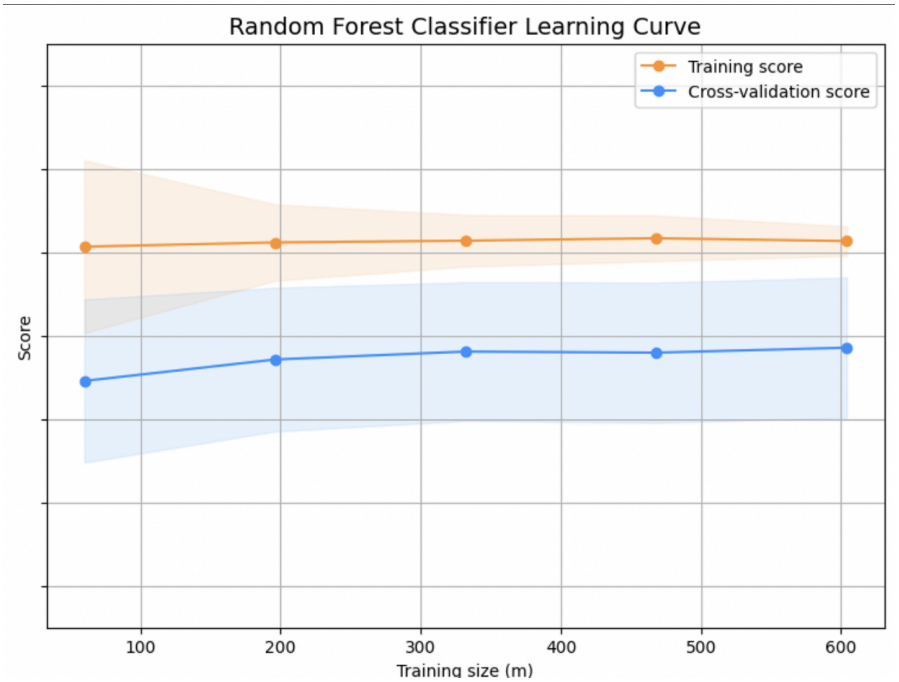


Fig. 15: Random Forest Classifier learning curve

5.1.3 Support Vector Classifier

The support vector machine (SVM) is an algorithm for machine learning that solves complex classification, regression, and outlier detection problems using supervised learning models. It accomplishes this by executing optimal data transformations, which establish boundaries between data points according to predefined classes, labels, or outputs.

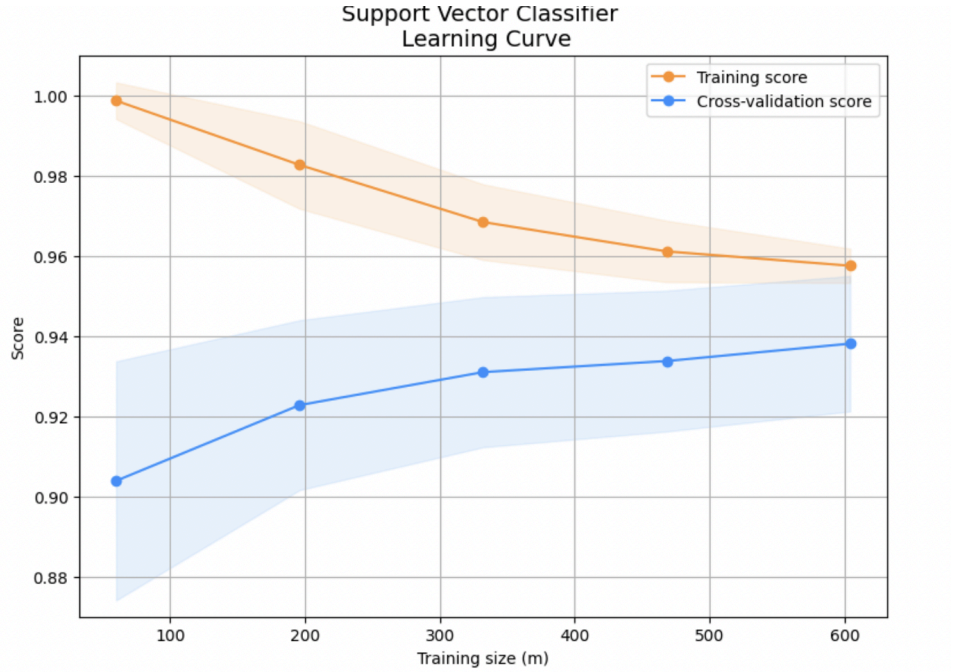


Fig. 16: SVM Classifier learning curve

5.1.4 Decision Tree Classifier

A Decision Tree is a type of Supervised learning algorithm that is capable of solving both classification and regression issues. However, it is primarily used for tackling classification problems. The classifier is organised in a tree structure, with core nodes representing dataset attributes, branches representing decision rules, and each leaf node representing an outcome.

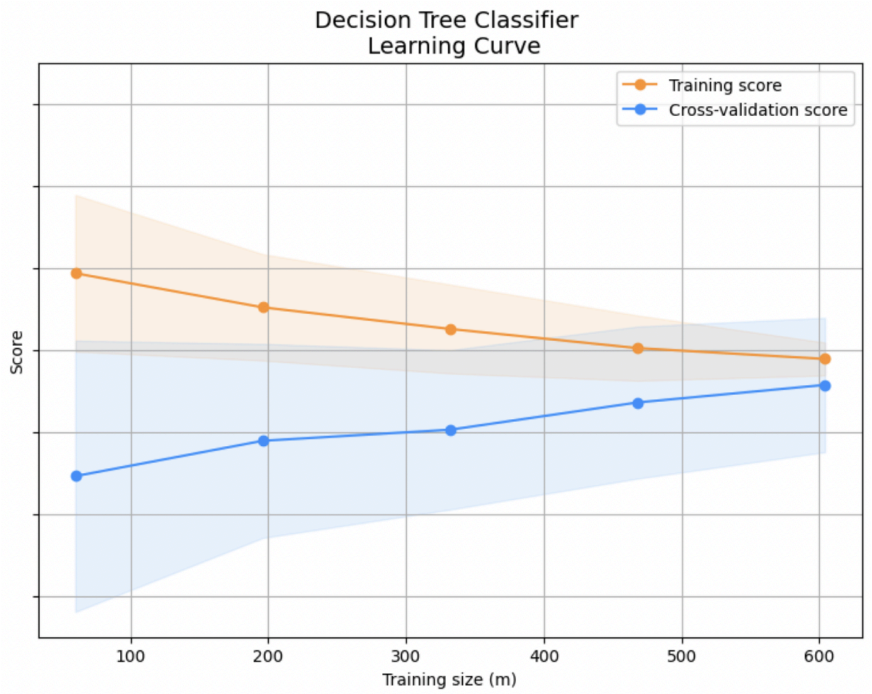


Fig. 17: Decision tree Classifier Learning Curve

5.2 ROC Curves of Models

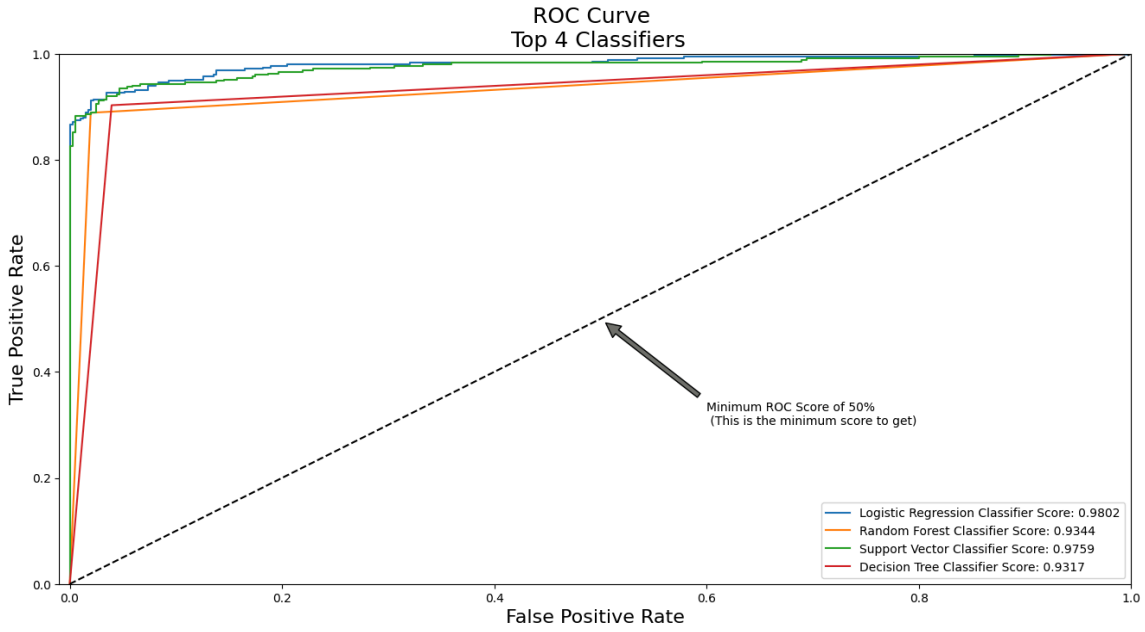


Fig. 18: ROC Curve of classifiers

As we can see the best outcome is from Logistic Regression which is 0.9802. Therefore, we will choose this model to

continue the detection.

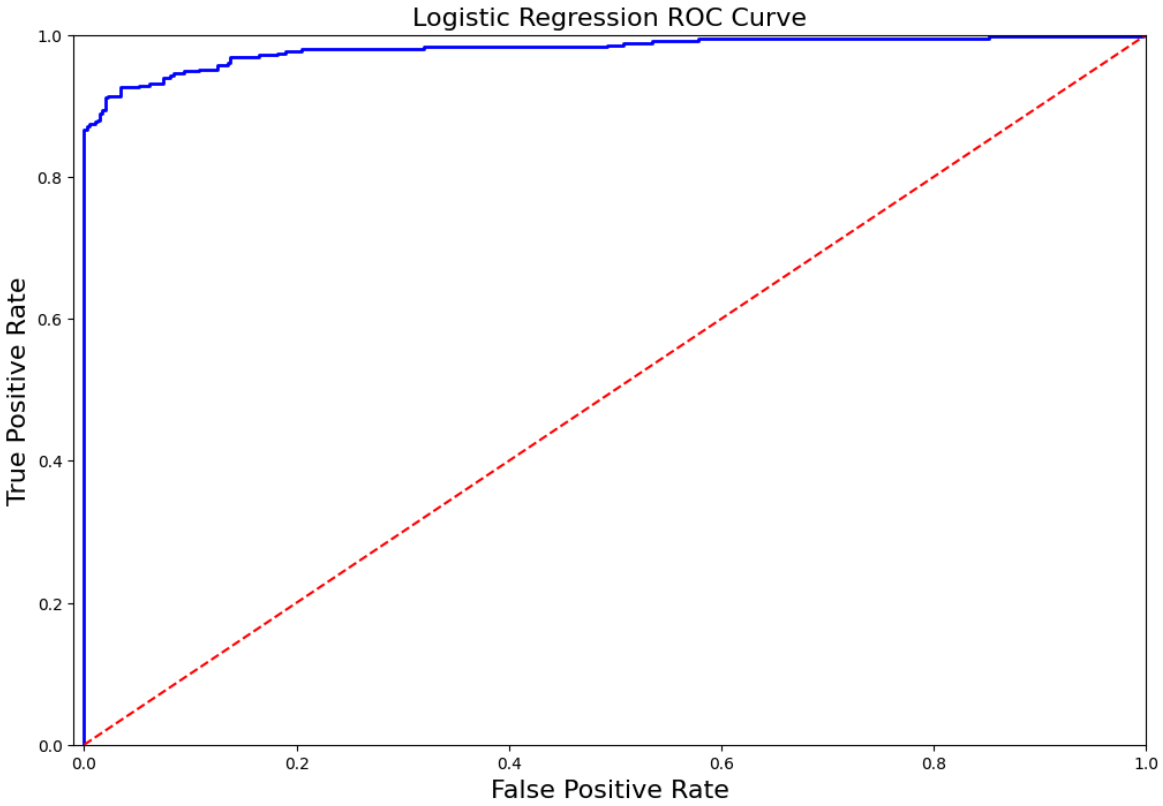


Fig. 19: ROC Curve of LR

This is the ROC curve showing the outcome of logistics regression.

5.3 Confusion matrix of the various ML models

Formulas:

Accuracy: $\frac{TP+TN}{TP+TN+FN+FP}$

Precision: $\frac{TP}{TP+FN}$

MCC: $\frac{TN*TP-FP*FN}{\sqrt{(TP+FN)(TP+FP)(TN+FP)(TN+FN)}}$

TP - True Positive

TN - True Negative

FP - False Positive

FN - False Negative

1. Logistic Regression

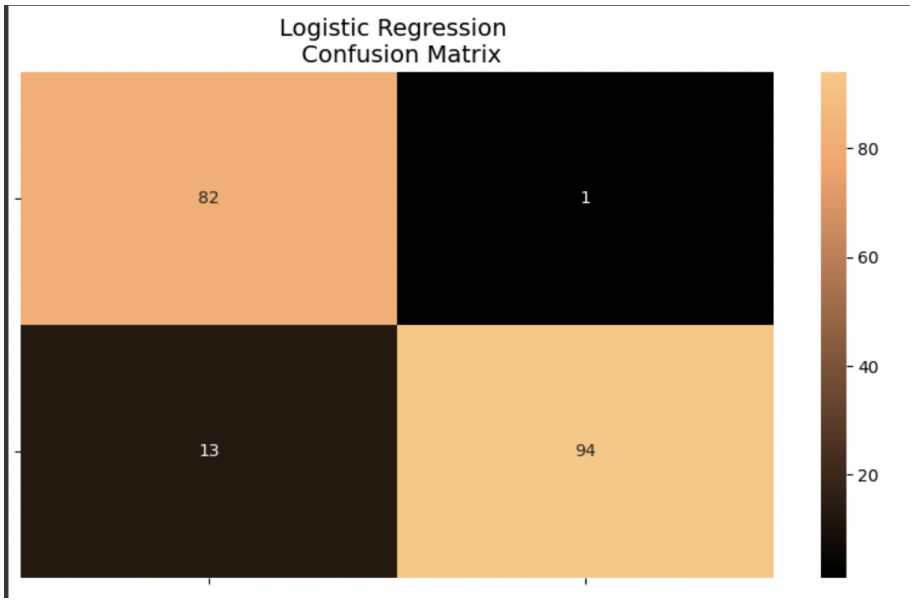


Fig. 20: Logistic Regression Confusion Matrix

2. Random Forest Classification

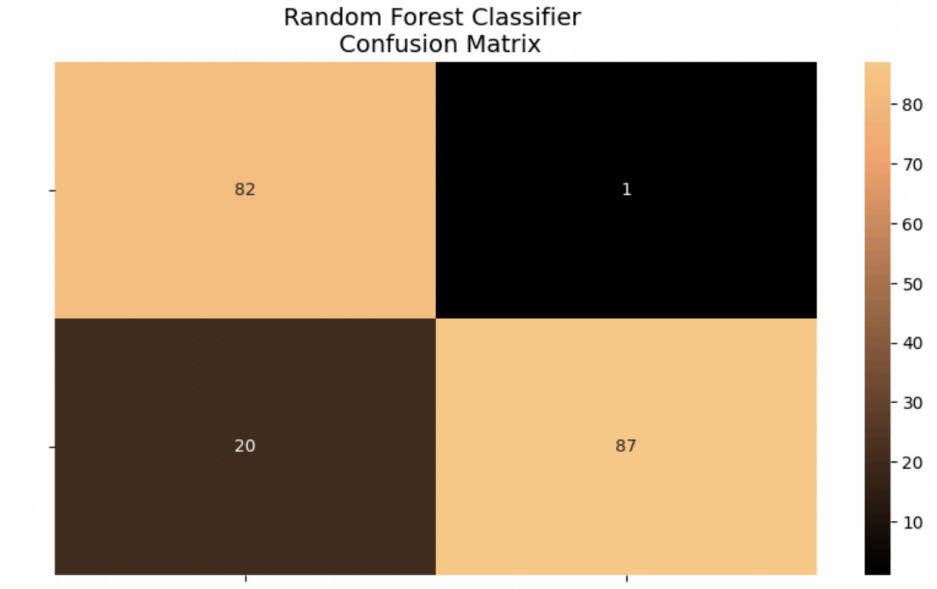


Fig. 21: Random Forest Classifier Confusion Matrix

3. Support Vector Machine

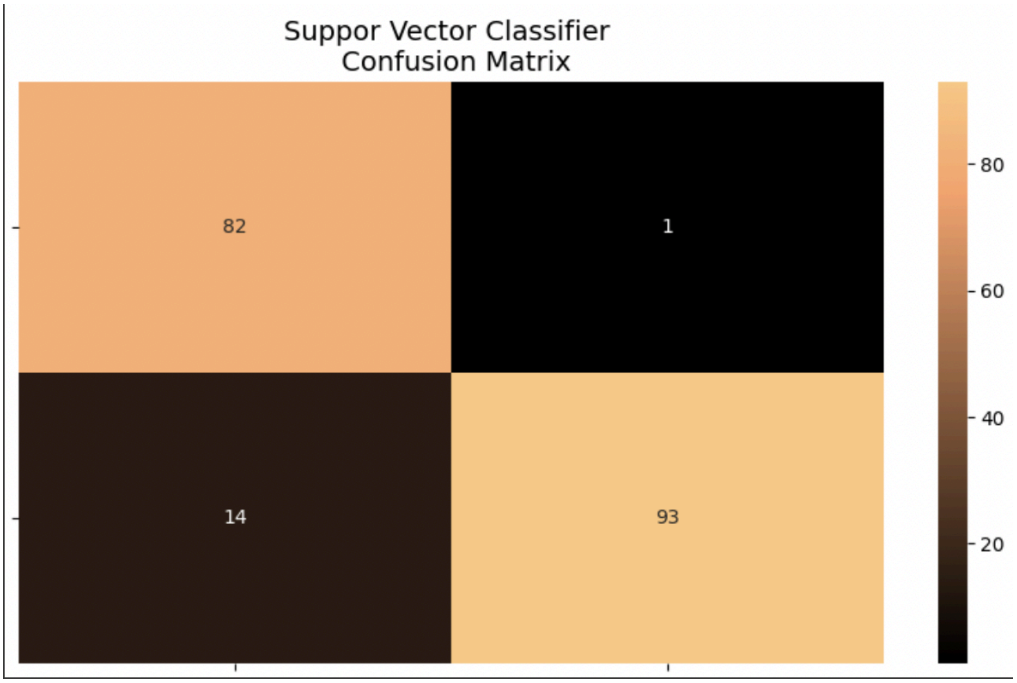


Fig. 22: SVM Confusion Matrix

4. Decision Tree Classification

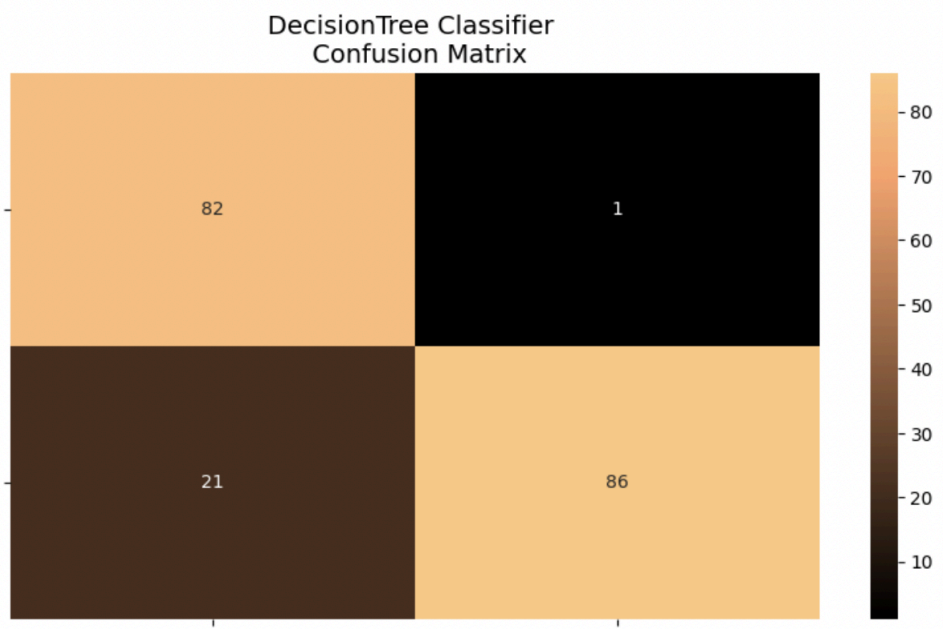


Fig. 23: Decision Tree Confusion Matrix

5.4 Comparison

In this section, we will implement a simple Neural Network (with one hidden layer) to see which of the two logistic regressions models we implemented in the (undersample or oversample(SMOTE)) has a better accuracy for detecting fraud and non-fraud transactions.

- **Main Goal:** to explore how our simple neural network behaves in both the random undersample and oversample data frames and see whether they can accurately predict both non-fraud and fraud cases. Why not only focus on fraud? Imagine you were a cardholder and after you purchased an item your card gets blocked because the bank's algorithm thought your purchase was a fraud. That's why we shouldn't emphasise only detecting fraud cases but we should also emphasise correctly categorising non-fraud transactions.

The Confusion Matrix:

- **Upper Left Square:** The amount correctly classified by our model of no fraud transactions.
- **Upper Right Square:** The amount of incorrectly classified transactions as fraud cases, but the actual label is no fraud.
- **Lower Left Square:** The amount of incorrectly classified transactions as no fraud cases, but the actual label is fraud.
- **Lower Right Square:** The amount correctly classified by our model of fraud transactions.

Keras || Random UnderSampling:

- **Dataset:** In this final phase of testing we will fit this model in both the random undersampled subset and oversampled dataset (SMOTE) to predict the final result using the original data frame testing data.
- **Neural Network Structure:** As stated previously, this will be a

simple model composed of one input layer (where the number of nodes equals the number of features) plus a bias node, one hidden layer with 32 nodes and one output node composed of two possible results 0 or 1 (No fraud or fraud). Other characteristics: The learning rate will be 0.001, the optimizer we will use is the Adam optimizer, the activation function that is used in this scenario is "Relu" and for the final outputs we will use sparse categorical cross-entropy, which gives the probability whether an instance case is no fraud or fraud (The prediction will pick the highest probability between the two.)

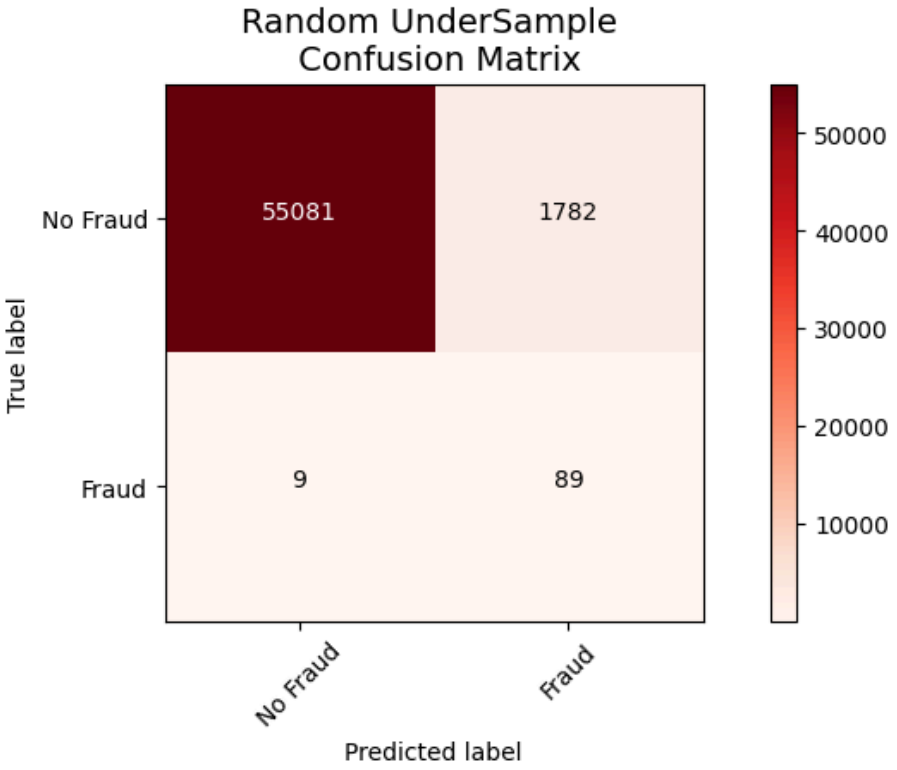


Fig. 24: Random Undersample confusion matrix

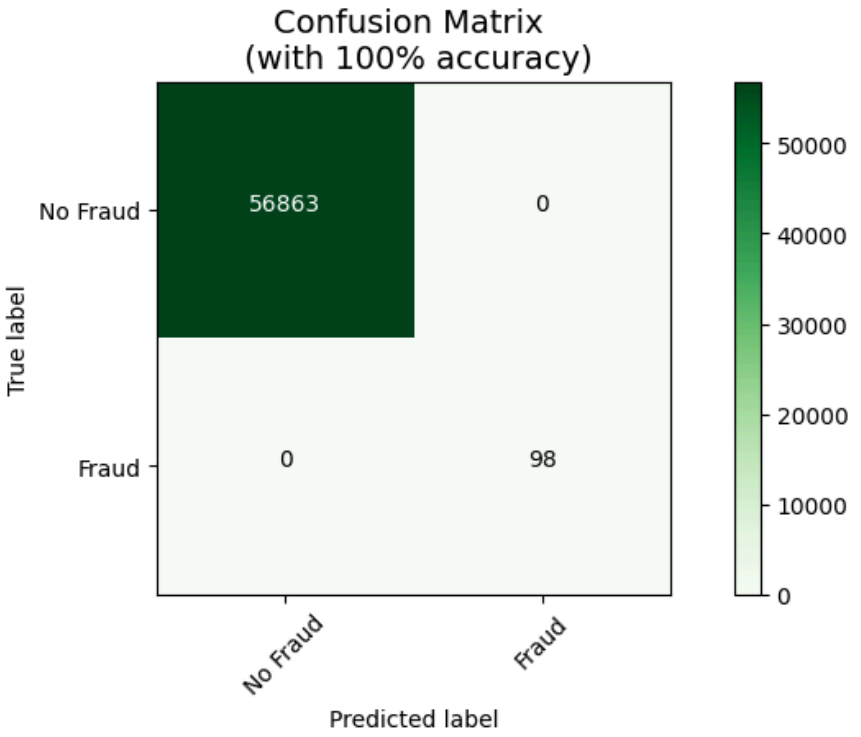


Fig. 25: Confusion matrix with 100% accuracy

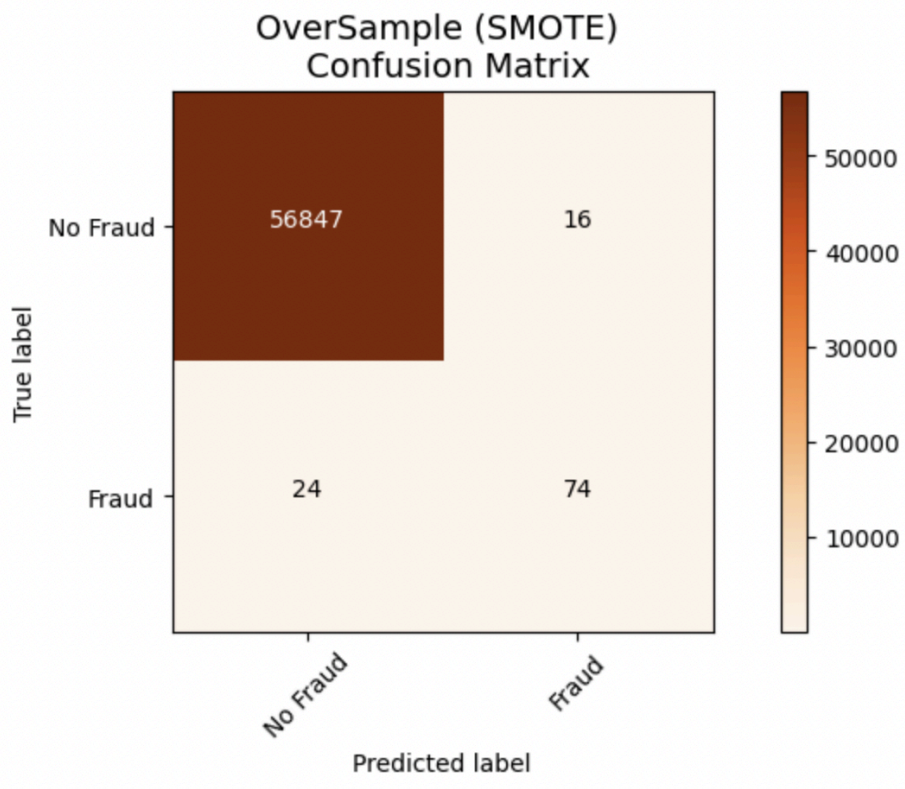


Fig. 26: Oversample(SMOTE) confusion matrix

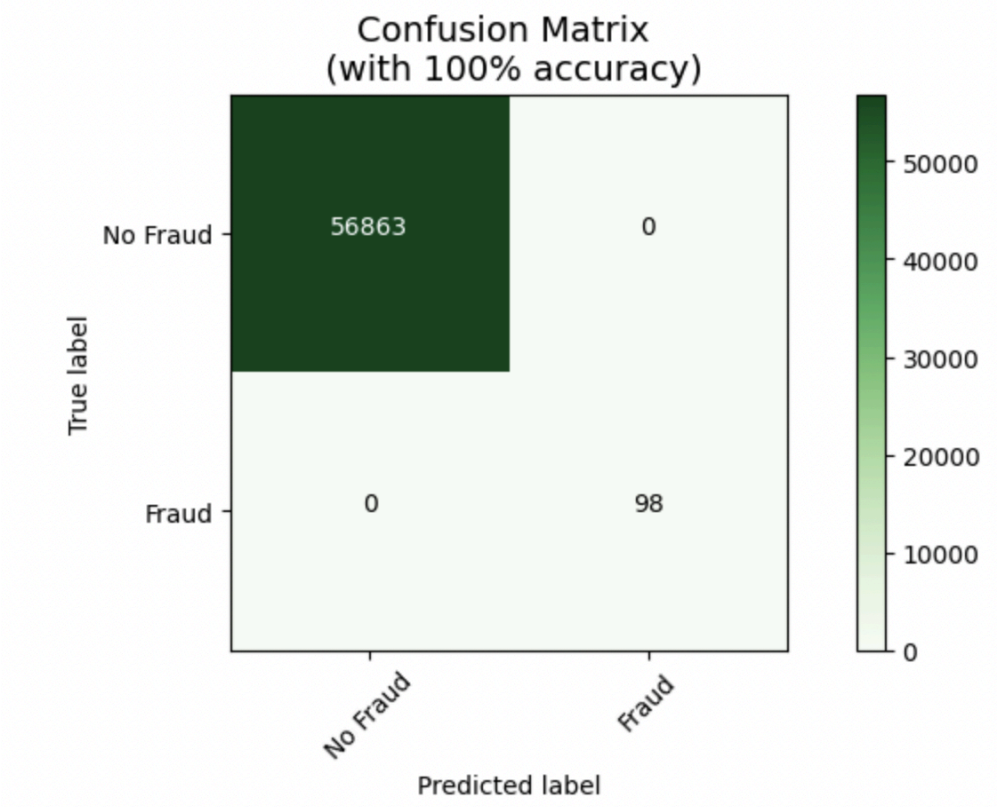


Fig. 27: Confusion matrix with 100% accuracy

Understanding SMOTE:

- Solving the Class Imbalance: SMOTE creates synthetic points from the minority class to reach an equal balance between the minority and majority class.
- Location of the synthetic points: SMOTE picks the distance between the closest neighbours of the minority class, in between these distances it creates synthetic points.
- Final Effect: More information is retained since we didn't have to delete any rows unlike in random undersampling.
- Accuracy || Time Tradeoff: Although SMOTE will likely be more accurate than random under-sampling, it will take more time to train since no rows are eliminated as previously stated.

5.6 Deployment Results from Docker and Streamlit

- **FAST API**

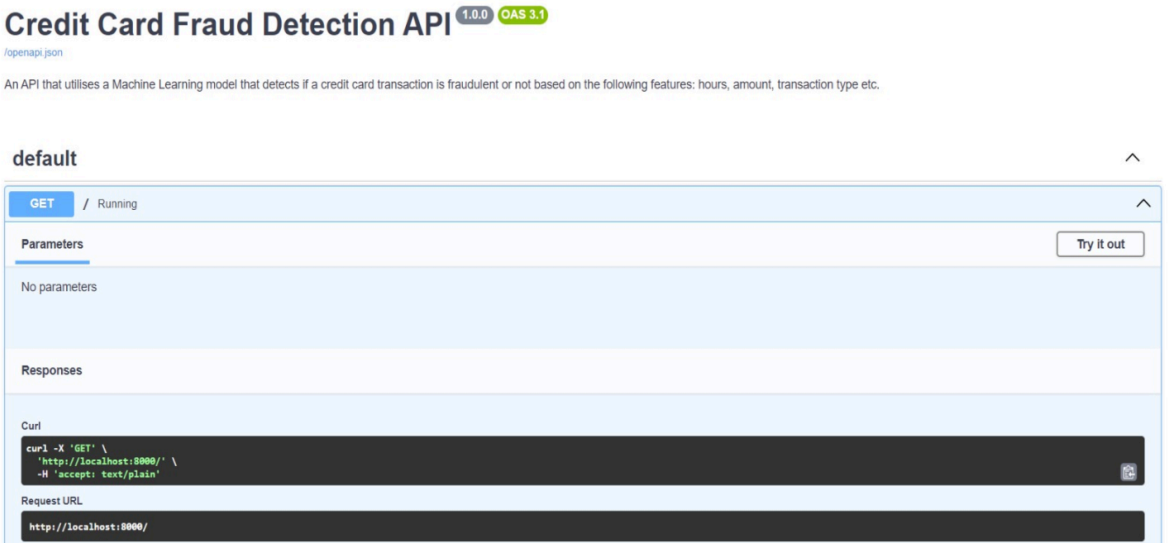


Fig. 28: FAST API 1.1

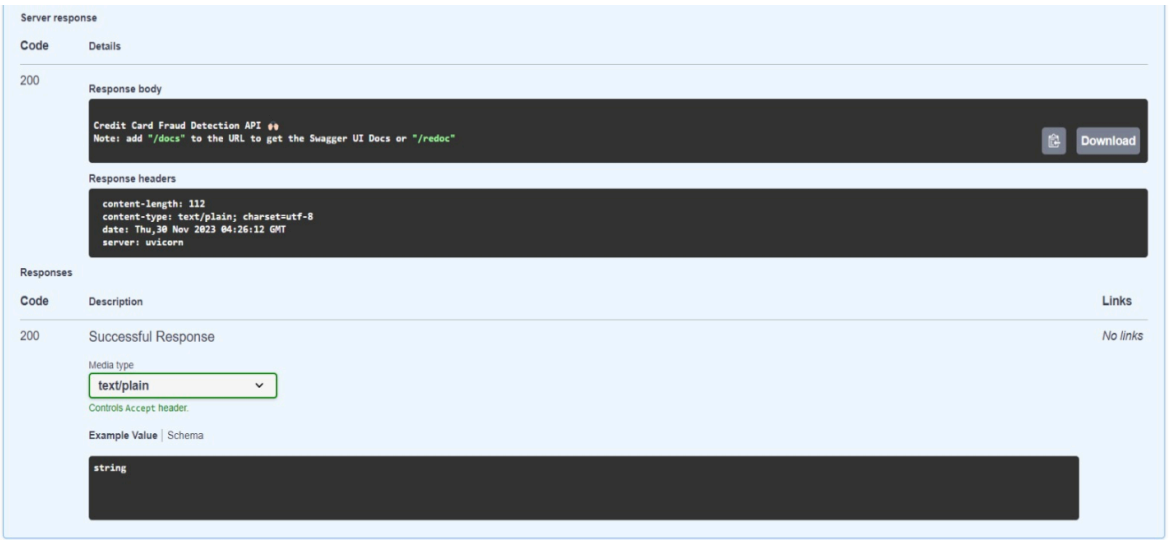


Fig. 29: FAST API 1.2

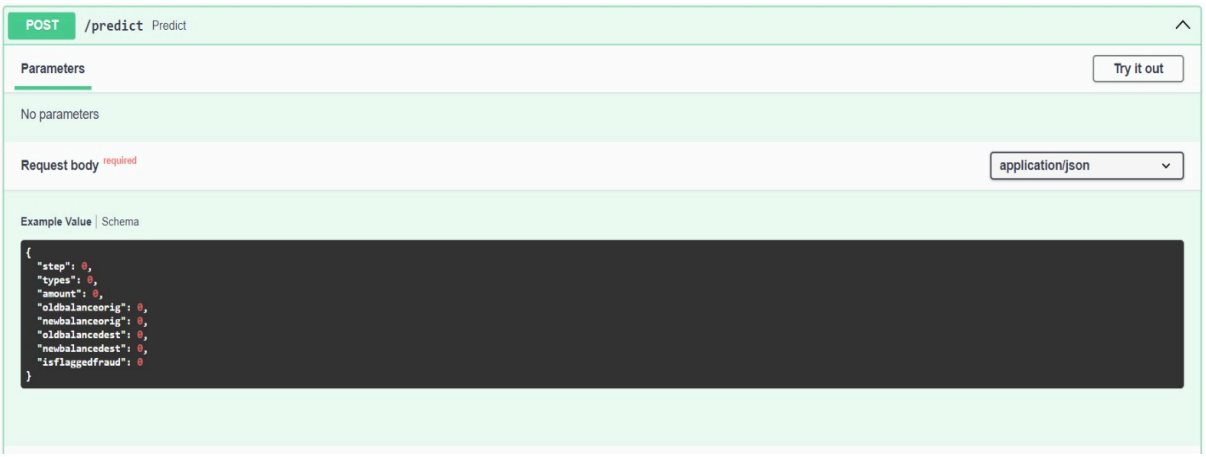


Fig. 30: FAST API 2.1

Fig. 31: FAST API 2.2

- **STREAMLIT**

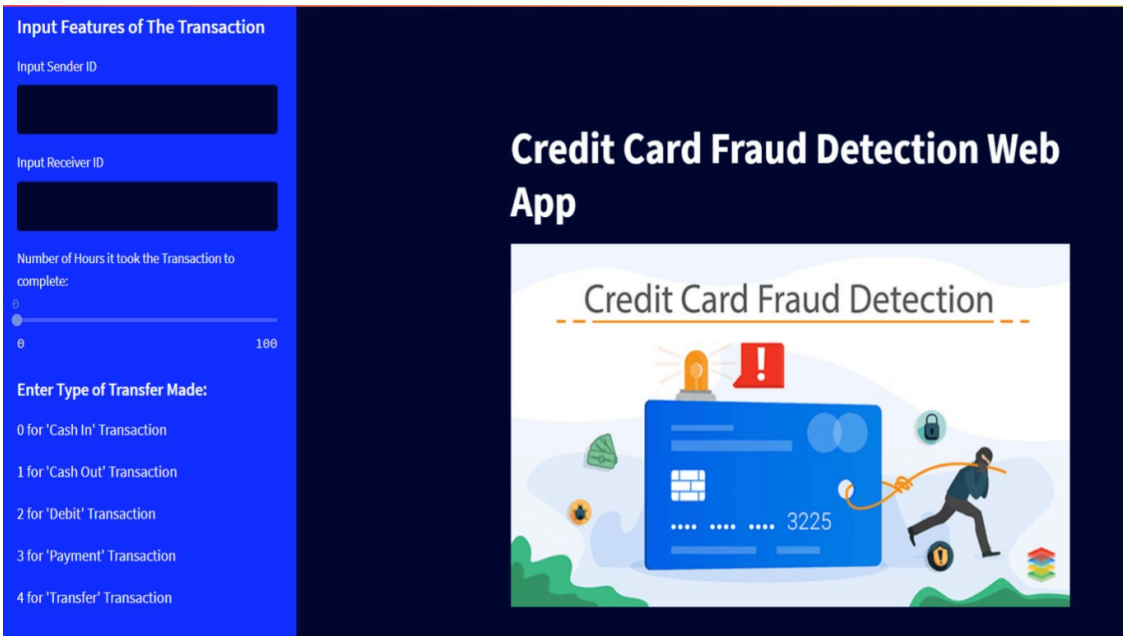


Fig. 32: STREAMLIT 1.1

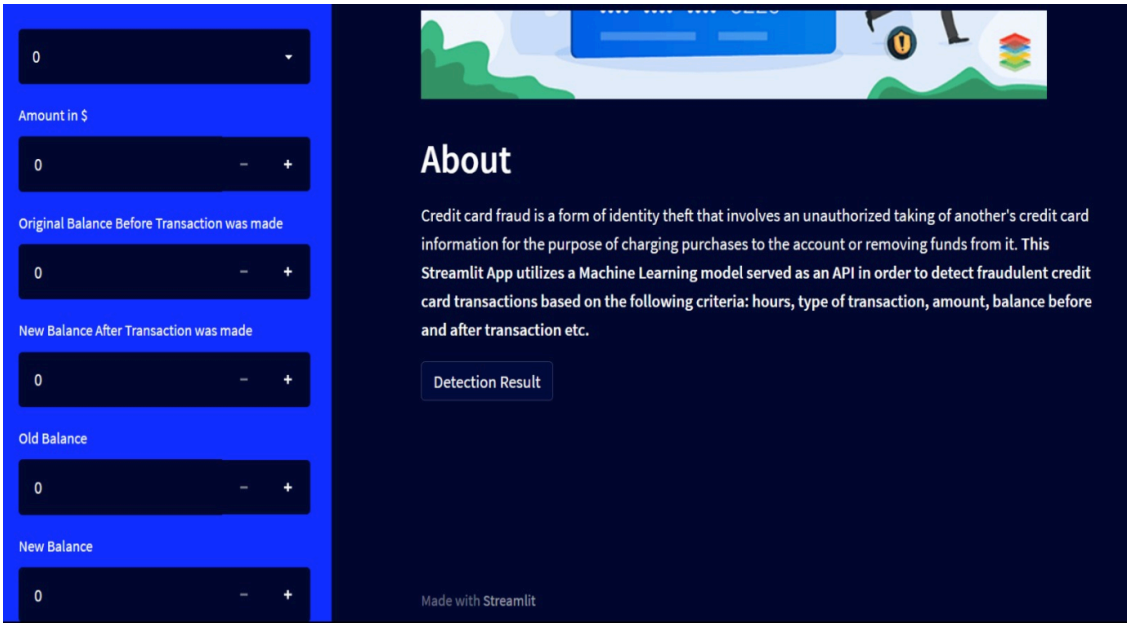


Fig. 33: STREAMLIT 1.2

CHAPTER-6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

- Implementing SMOTE on our imbalanced dataset helped us with the imbalance of our labels (more no fraud than fraud transactions). Nevertheless, I still have to state that sometimes the neural network on the oversampled dataset predicts fewer correct fraud transactions than our model using the undersampled dataset. However, remember that the removal of outliers was implemented only on the random undersample dataset and not on the oversampled one. Also, in our undersample data our model is unable to detect for a large number of cases non-fraud transactions correctly and instead, misclassified those non-fraud transactions as fraud cases. Imagine that people who were making regular purchases got their card blocked due to the reason that our model classified that transaction as a fraud transaction, this will be a huge disadvantage for the financial institution. The number of customer complaints and customer satisfaction will increase. The next step of this analysis will be to do an outlier removal on our oversample dataset and see if our accuracy in the test set improves.

- In this project, the development of a detection system using machine learning, Streamlit, Docker, and FastAPI has been successfully executed. The key findings, limitations, and contributions to the field are summarised as follows:

6.1.1 Key Findings

- Effective Fraud Detection: The implemented machine learning model, integrated into a user-friendly Streamlit interface and scalable FastAPI, demonstrated effective real-time fraud detection. Streamlined
- Development: The use of Streamlit simplified the development process, enabling data scientists to create interactive web applications without extensive web development knowledge.
- Containing with Docker: It helps in easy deployment of the application, with uniform performance in multiple environments.

6.1.2 Limitations

- Unbalanced data: Still at the present day tackling unbalanced data sets is a big issue which the researchers continuously work on and attempt to find a more efficient method of tackling class imbalance.
- Interpretation of standards: It is very important for compatibility and user trust when standards are good, make their interpretation.

6.1.3 Contributions to the Field

- Development ease: Our project ensures that it offers an easy development with the help of Streamlit, making it easier and faster to detect fraud activities.
- Containerized deployment: This is one of the reasons why using Docker makes deployment much easier. It makes sure that scaling can occur at the installation level and adoption among other environments.

6.2 Future Scope

6.2.1 Enhanced Model Performance

- Advanced Pattern Detection Techniques: Explore advanced machine learning algorithms such as recurrent neural networks (RNNs) or transformers to improve the model's ability to detect evolving patterns of activities. Utilize techniques like attention mechanisms or self-supervised learning to capture nuanced temporal dependencies and adapt to changing data dynamics effectively.
- Integration of Ensemble Models: Integrate ensemble learning methods, combining multiple models trained on different aspects or representations of the data, to enhance overall accuracy and reliability. Ensemble techniques like bagging, boosting, or stacking can help mitigate individual model biases and uncertainties, resulting in more robust predictions and insights.
- Hybrid Model Architectures: Develop hybrid model architectures that leverage the strengths of diverse models, such as combining deep learning with traditional statistical approaches or rule-based systems. This fusion of techniques allows for a holistic understanding of complex data patterns, improving the model's adaptability and performance across various scenarios.

6.2.2 Explainability and Interpretability

- Utilise and incorporate a range of interpretation techniques, such, as SHAP (Shapley Additive Explanation) to gain an understanding of the decision-making model.

6.2.3 Continuous Learning and Model Updating

- Dynamic dataset updates; Implement a method for the model to continuously learn from data and adjust to evolving patterns.
- Automate the retraining process; Establish streamlined workflows that periodically retrain your model ensuring its validity on time.

6.2.4 Regulatory Compliance and Security

- Privacy-preserving technologies: Explore and implement privacy-preserving technologies to improve the protection of sensitive customer information.
- Compliance: Stay informed of changes and ensure systems comply with data protection and security standards.

6.2.5 User Interface Enhancements

- User Feedback Integration: Capture user actions within the system to provide real-time performance evaluation feedback. Utilise metrics such as response times, error rates, and user interactions to gauge system performance. Incorporate this feedback loop into ongoing development cycles to prioritise optimizations and enhancements based on user needs and pain points. Streamlit
- Dashboard Enhancement: Enhance the Streamlit dashboard with intuitive visualisations to improve user understanding and engagement. Introduce interactive charts, graphs, and data summaries to convey complex information effectively. Incorporate user preferences and feedback to tailor visualisation choices and ensure relevance to their workflows, promoting better usability and satisfaction.
- Iterative Improvement Cycle: Establish an iterative improvement cycle driven by user feedback loops. Regularly solicit user input through surveys, usability testing, and direct communication channels to identify areas for dashboard enhancement. Continuously iterate on visualisation designs, features, and performance optimizations based on user insights, fostering a responsive and user-centric development process.

6.2.6 Performance Optimization

- Enhanced Scalability Testing: Expand scalability testing to cover various usage scenarios, including peak loads and stress testing, using appropriate tools. This broader approach helps identify potential bottlenecks under different conditions, enabling targeted optimization efforts.
- Distributed Caching Implementation: Introduce distributed caching mechanisms like Redis or Memcached to store frequently accessed data in memory. This reduces database load and improves response times for commonly requested information, enhancing system performance and scalability.
- Cache Invalidation and Optimization: Develop a robust cache invalidation strategy to maintain data consistency and accuracy. Implement techniques such as time-based expiration and event-driven invalidation to ensure timely updates while optimising cache key design to minimise cache misses and maximise utilisation. Regularly monitor and adjust caching parameters based on performance metrics to fine-tune system responsiveness.

In conclusion, while the current system demonstrates effective fraud detection, there are opportunities for further improvement and expansion. Proposed future work identifies ways to improve model performance, interpretability, compliance, user interface, and overall system optimization. By addressing these challenges, our fraud detection systems can remain responsive, secure, and user-friendly in the ever-evolving world of financial technology.

References

[1] Vaishnavi Nath Dornadula, S Geetha, “Credit Card Fraud Detection using Machine Learning Algorithms”,Procedia Computer Science,Volume 165,2019.

[2] A. Mahajan, V. S. Baghel and R. Jayaraman, "Credit Card Fraud Detection using Logistic Regression with Imbalanced Dataset," 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2023.

[3] W. -F. Yu and N. Wang, "Research on Credit Card Fraud Detection Model Based on Distance Sum," 2009 International Joint Conference on Artificial Intelligence, Hainan, China, 2009.

[4] B. Al Smadi and M. Min, "A Critical review of Credit Card Fraud Detection Techniques," 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 2020.

[5] S. Mittal and S. Tyagi, "Performance Evaluation of Machine Learning Algorithms for Credit Card Fraud Detection," 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2019.

[6] J. R. D. Kho and L. A. Vea, "Credit card fraud detection based on transaction behaviour," TENCON 2017 - 2017 IEEE Region 10 Conference, Penang, Malaysia, 2017.

[7] Y. Sahin and E. Duman, "Detecting credit card fraud by ANN and logistic regression," 2011 International Symposium on Innovations in Intelligent Systems and Applications, Istanbul, Turkey, 2011.

[8] A. A. El Naby, E. El-Din Hemdan and A. El-Sayed, "Deep Learning Approach for Credit Card Fraud Detection," 2021 International Conference on Electronic Engineering (ICEEM), Menouf, Egypt, 2021.

[9] Dubey, Saurabh C., Ketan S. Mundhe and Aditya Kadam. “Credit Card Fraud Detection using Artificial Neural Network and BackPropagation.” 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS) (2020).

[10] G. E. Melo-Acosta, F. Duitama-Muñoz and J. D. Arias-Londoño, "Fraud detection in big data using supervised and semi-supervised learning techniques," 2017 IEEE Colombian Conference on Communications and Computing (COLCOM), Cartagena, Colombia, 2017.

[11] Melo-Acosta, German E., et al. “Fraud Detection in Big Data Using Supervised and Semi-Supervised Learning Techniques.” 2017 IEEE Colombian Conference on Communications and Computing (COLCOM), 2017.

[12] Mohammed, Emad, and Behrouz Far. “Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study.” IEEE Annals of the History of Computing, IEEE, 1 July 2018,

[13] Nilsonreport.Online: https://nilsonreport.com/publication_newsletter_archive_issue.php?issue=1187

[14] Xuan, Shiyang, et al. “Random Forest for Credit Card Fraud Detection.” 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), 2018, doi:10.1109/icnsc.2018.8361343.

[15] Johnson Adeleke Adeyiga, Philip Gbounmi Toriola, Temitope Elizabeth Abioye(Ogunbiyi) et al. Fake News Detection Using a Logistic Regression Model

and Natural Language Processing Techniques, 14 July 2023.

[16] Pumsirirat, A. and Yan, L. (2018). Credit Card Fraud Detection using Deep Learning based on Auto-Encoder and Restricted Boltzmann Machine. International Journal of Advanced Computer Science and Applications, 9(1).

[17] Y. Li, X. Zhang, and D. Chen, “Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes,”

[18] Randhawa, Kuldeep, et al. “Credit Card Fraud Detection Using AdaBoost and Majority Voting.” IEEE Access, vol. 6, 2018.

[19] Johnson Adeleke Adeyiga, Philip Gbounmi Toriola, Temitope Elizabeth Abioye(Ogunbiyi) et al. Fake News Detection Using a Logistic Regression Model and Natural Language Processing Techniques, 14 July 2023.

[20] Nabila Farnaaz, M.A. Jabbar,"Random Forest Modeling for Network Intrusion Detection System", Procedia Computer Science, Volume 89, 2016.

[21] M. Shahin, M. Ali Babar and L. Zhu, "Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices," in IEEE Access, vol. 5.

[22] Scikit-learn : machine learning in Python [Online]. https:// scikit- learn. org/ stable/

[23] Streamlit: A faster way to build and share data apps [Online]. https://streamlit.io/

[24] Deploying Machine Learning models with Streamlit, FastAPI by Rihab[Online].

https://rihab-feki.medium.com/deploying-machine-learning-models-with-streamlit-fastapi-and-docker

[25] Bankrate: Credit Card Fraud [Online]. www.bankrate.com/finance/credit-cards/credit-card-fraud-statistics/

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 13 May 2024.

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report ✓ | Paper

Name: Manisha Susarla, Animesh Singh Department: CSE Enrolment No 201291, 201349

Contact No. 9760382077, 7488 093564 E-mail. 201291@juitsolan.in, 201349@juitsolan.in

Name of the Supervisor: Dr. Rakesh Kanji

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): CREDIT CARD FRAUD DETECTION SYSTEM USING ML

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages = 69
- Total No. of Preliminary pages = 54
- Total No. of pages accommodate bibliography/references = 3

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at 18 (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor)

Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature

Librarian

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com