

FRAUD DETECTION IN FINANCIAL TRANSACTIONS

Real Time Project report

Submitted in partial fulfilment of the requirement for the award of the Degree of

Bachelor of Technology (B. Tech)

in

COMPUTER SCIENCE AND ENGINEERING (AI&ML)

By

D.AKSHAYA

22AG1A6617

Under the Esteemed Guidance of

Mrs K. Swetha Sailaja

Assistant Professor



Department of Computer Science and Engineering (AI&ML)

ACE ENGINEERING COLLEGE

An Autonomous Institution

NBA Accredited B.Tech Courses, Accorded NAAC “A” Grade

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad, Telangana)

Ankushapur(V), Ghatkesar(M), Medchal- Malkajgiri Dist - 501 301.

JULY 2024.



ACE

ENGINEERING COLLEGE

AN AUTONOMOUS INSTITUTION

Website: www.aceec.ac.in E-mail: info@aceec.ac.in

COMPUTER SCIENCE AND ENGINEERING (AI&ML)

CERTIFICATE

This is to certify that the Real Time Project work entitled “**FRAUD DETECTION IN FINANCIAL TRANSACTIONS**” is being submitted by **D. AKSHAYA (22AG1A6617)**, in partial fulfilment for the award of Degree of **BACHELOR OF TECHNOLOGY** in **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI&ML)** to the Jawaharlal Nehru Technological University, Hyderabad is a record of Bonafide work carried out by them under our guidance and supervision.

The results embodied in this project have not been submitted by the student to any other University or Institution for the award of any Degree or Diploma.

Internal Guide

Mrs K. Swetha Sailaja
Assistant Professor

Head of the Department

Dr S. Kavitha
Associate Professor and
Head Dept. of CSE(AI&ML)

ACKNOWLEDGEMENT

We would like to express our gratitude to all the people behind the screen who have helped us transform an idea into a real time application.

We would like to express our heart-felt gratitude to our parents without whom we would not have been privileged to achieve and fulfil our dreams.

A special thanks to our General Secretary, **Prof. Y. V. Gopala Krishna Murthy**, for having founded such an esteemed institution. We are also grateful to our beloved principal, **Dr. B. L. RAJU** for permitting us to carry out this project.

We profoundly thank **Dr. S. Kavitha**, Associate Professor and Head of the Department of Computer Science and Engineering (AI&ML), who has been an excellent guide and also a great source of inspiration to our work.

We extremely thank **Mrs. J Bhargavi**, Assistant Professor, Project coordinator, who helped us in all the way in fulfilling of all aspects in completion of our Real Time Project work.

We are very thankful to our internal guide, **Mrs K. Swetha Sailaja**, Assistant Professor who has been an excellent and also given continuous support for the Completion of our project work.

The satisfaction and euphoria that accompany the successful completion of the task would be great, but incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success. In this context, we would like to thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased our task.

D.AKSHAYA 22AG1A6617

ABSTRACT

Credit card fraud events take place frequently and then result in huge financial losses. The number of online transactions has grown in large quantities and online credit card transactions holds a huge share of these transactions. Therefore, banks and financial institutions offer credit card fraud detection applications much value and demand. Fraudulent transactions can occur in various ways and can be put into different categories. This paper focuses on four main fraud occasions in real-world transactions. Each fraud is addressed using a series of machine learning models and the best method is selected via an evaluation. This evaluation provides a comprehensive guide to selecting an optimal algorithm with respect to the type of the frauds and we illustrate the evaluation with an appropriate performance measure. Another major key area that we address in our project is real-time credit card fraud detection. For this, we take the use of predictive analytics done by the implemented machine learning models and an API module to decide if a particular transaction is genuine or fraudulent.

INDEX

CONTENTS	PAGE NO
1. INTRODUCTION	1
1.1 Overview of Fraud in Financial Transactions	1
1.2 Importance of Fraud Detection	1
1.3 Scope and Structure	2
2. LITERATURE SURVEY	3
2.1 Overview of Machine Learning in Fraud Detection	3
2.2 Supervised Learning Techniques	3
2.3 Handling Imbalanced Data	4
3. SYSTEM REQUIREMENTS	5
3.1 Hardware Requirements	5
3.2 Software Requirements	5
4. SYSTEM ARCHITECTURE	7
5. SYSTEM DESIGN	10
5.1 Flow chart	10
5.2 UML Diagrams	11

5.2.1 Class Diagram	11
5.2.2 Use Case Diagram	12
5.2.3 Sequence Diagram	13
5.2.4 Activity Diagram	14
6. IMPLEMENTATION	15
7. TESTING	17
7.1 Testing Methodologies	17
7.2 Model Evaluation	17
7.3 Testing Procedures	17
7.4 Testing Tools	18
8. OUTPUT SCREENS	19
9. CONCLUSION	20
9.1 Future Enhancement	20
10. REFERENCES	21
11. APPENDICES	22

LIST OF FIGURES

Fig. No.	Figure Name	Page No.
4.1	Workflow for credit card data analysis	7
5.1	Flow Chart for credit card data analysis	10
5.2.1	Class Diagram	11
5.2.2	Use Case Diagram	12
5.2.3	Sequence Diagram	13
5.2.4	Activity Diagram	14
8.1	Accuracy Score	19
8.2	Accuracy of Training and Testing Data	19

LIST OF TABLES

Tab. No.	Table Name	Page No.
6.1.	Data Set	15

LIST OF NOTATIONS / ABBREVIATIONS**Abbreviation****Full Form**

EDA	-	Exploratory Data Analysis
ML	-	Machine Learning
SMOTE	-	Synthetic Minority Over-sampling Technique
RAM	-	Random Access Memory
SSD	-	Solid State Drive
AWS	-	Amazon Web Services
ANN	-	Artificial Neural Network
IEEE	-	Institute of Electrical and Electronics Engineers
ICCSI	-	International Conference on Cyber-Physical Social Intelligence
ITM	-	Institute of Technology Management

CHAPTER 1

INTRODUCTION

Credit card fraud events take place frequently and then result in huge financial losses. The number of online transactions has grown in large quantities and online credit card transactions holds a huge share of these transactions. Therefore, banks and financial institutions offer credit card fraud detection applications much value and demand. Fraudulent transactions can occur in various ways and can be put into different categories. This paper focuses on four main fraud occasions in real-world transactions. Each fraud is addressed using a series of machine learning models and the best method is selected via an evaluation. This evaluation provides a comprehensive guide to selecting an optimal algorithm with respect to the type of the frauds and we illustrate the evaluation with an appropriate performance measure. Another major key area that we address in our project is real-time credit card fraud detection. For this, we take the use of predictive analytics done by the implemented machine learning models and an API module to decide if a particular transaction is genuine or fraudulent.

1.1 Overview of Fraud in Financial Transactions

Fraudulent activities in financial transactions have become a significant threat to both individuals and financial institutions. These activities encompass a wide range of illicit behaviours, including unauthorized access to accounts, identity theft, account takeovers, and various forms of transaction fraud. The increasing reliance on digital banking and online transactions has provided new opportunities for fraudsters to exploit vulnerabilities in financial systems. The financial losses due to fraud are substantial, with billions of dollars lost annually worldwide. Beyond the financial implications, fraud can severely damage the reputation and trustworthiness of financial institutions, leading to long-term negative consequences.

1.2 Importance of Fraud Detection

Given the pervasive nature of financial fraud, effective detection and prevention strategies are essential. Fraud detection is critical to protect customers' assets, safeguard the integrity of financial systems, and ensure compliance with regulatory requirements. Early detection of fraudulent activities can prevent significant financial losses and reduce the risk of legal repercussions. Real-time fraud detection systems are particularly important, as they can identify

suspicious activities and block fraudulent transactions before they are completed. This proactive approach helps maintain customer trust and supports the overall stability of the financial sector.

1.3 Scope and Structure

The document is organized into several key sections to provide a logical and comprehensive flow of information. The initial sections offer background information on financial fraud and traditional detection methods, setting the stage for the introduction of machine learning approaches. Following this, a detailed overview of logistic regression is provided, explaining why it is a suitable choice for fraud detection. Subsequent sections delve into the practical aspects of building a fraud detection system, including data collection, preprocessing, exploratory data analysis (EDA), and model training. Techniques for handling imbalanced data and improving model performance are discussed to ensure the robustness and accuracy of the detection system. The final sections cover the implementation and deployment of the model in a real-world setting, including integration with financial systems and considerations for scalability and maintenance. Real-world case studies and lessons learned from past implementations are presented to provide practical insights and best practices. Ethical considerations and future directions are also addressed, emphasizing the importance of responsible use and continuous improvement in fraud detection techniques.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview of Machine Learning in Fraud Detection

In the fraud detection project, Machine Learning (ML) plays a crucial role in identifying potentially fraudulent financial transactions. Historical transaction data is used to train ML models, enabling them to recognize patterns and anomalies associated with fraud. Various features such as transaction amount, frequency, location, and user behaviour are analysed to create a robust feature set. Logistic Regression, a widely used binary classification algorithm, is implemented to predict the likelihood of a transaction being fraudulent. The model is continuously trained and validated to improve its accuracy and adaptability to new fraud patterns. Real-time prediction capabilities allow for immediate detection and response, minimizing financial losses. ML enhances the detection process by automating and scaling the analysis, providing a more efficient and effective fraud detection mechanism compared to traditional rule-based systems.

2.2 Supervised Learning Techniques

Supervised learning involves training models on labelled datasets where the outcome (fraudulent or legitimate) is known. Several studies have explored the application of supervised learning techniques to fraud detection:

- **Logistic Regression:** As a linear model, logistic regression is widely used for binary classification problems. It is favoured for its simplicity, interpretability, and effectiveness. Delamare et al. (2009) highlighted the use of logistic regression in credit card fraud detection, emphasizing its ability to handle large datasets and provide probabilistic outputs for decision-making.
- **Decision Trees and Random Forests:** These algorithms offer high accuracy and are easy to interpret. For instance, Whit row et al. (2009) demonstrated the effectiveness of decision trees in detecting fraud by analysing transaction patterns and customer behaviour.
- **Neural Networks:** Although more complex, neural networks can capture intricate patterns in data. A study by Dorronsoro et al. (1997) applied neural networks to credit

card fraud detection, achieving promising results by learning from historical transaction data.

2.3 Handling Imbalanced Data

Fraudulent transactions are rare compared to legitimate ones, leading to imbalanced datasets that challenge ML models:

- **Oversampling and Under Sampling:** Techniques like SMOTE (Synthetic Minority Over-sampling Technique) and under sampling balance the dataset by generating synthetic samples or reducing the number of majority class samples. Chawla et al. (2002) introduced SMOTE, which has been widely adopted in fraud detection to address class imbalance.
- **Cost-Sensitive Learning:** Modifying the learning algorithm to consider the cost of misclassification can improve performance on imbalanced datasets. Elkan (2001) discussed cost-sensitive approaches, highlighting their effectiveness in fraud detection scenarios where the cost of false negatives is particularly high.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 Hardware Requirements

1. Computing Power:

- **CPU:** A multi-core processor with sufficient computational power to handle large datasets and perform model training efficiently.
- **RAM:** At least 8GB of RAM, preferably more for handling large-scale data processing and model training.

2. Storage:

- **Hard Drive:** A fast and reliable storage device with sufficient capacity (at least 500GB) to store transaction data, feature sets, and trained models.
- **SSD:** SSDs are recommended for faster data retrieval and processing speeds, especially during real-time transaction monitoring.

3. Scalability:

- Ensure the hardware infrastructure supports scalability to accommodate increasing transaction volumes and data growth over time.
- Cloud-based solutions (e.g., AWS, Azure) can provide scalable computing resources and storage options as needed.

3.2 Software Requirements

1. Operating System:

- Support for both Linux and Windows environments, depending on the organization's preferences and existing infrastructure.

2. Programming Languages:

- **Python:** Required for data preprocessing, feature engineering, model development, and deployment.

3. Development Frameworks and Libraries:

- **Machine Learning Libraries:**
 - **scikit-learn:** For implementing logistic regression models, handling data preprocessing, and model evaluation.
- **Pandas, NumPy:** For data manipulation and numerical computations.
- **Matplotlib, Seaborn:** For data visualization to explore fraud patterns and model performance.

CHAPTER 4

SYSTEM ARCHITECTURE

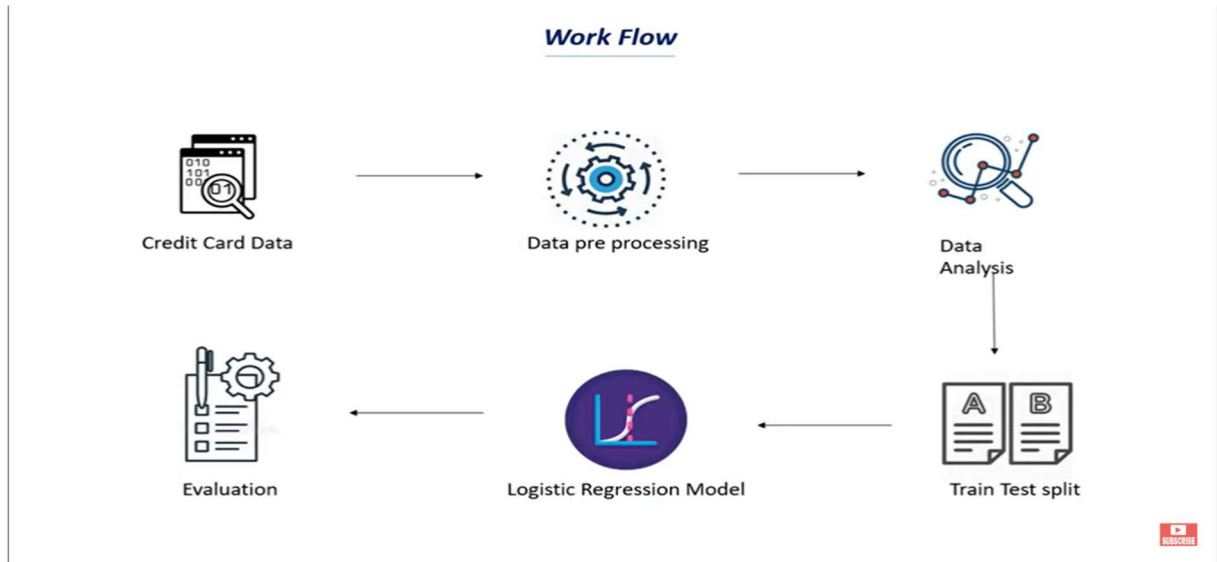


Fig. 4.1 workflow for credit card data analysis

MODULE-1: Credit Card Data

In this step, we gather and collect the credit card transaction data. This dataset typically includes various attributes such as transaction amount, transaction date, merchant details, and potentially some user-specific information. The data might be sourced from a financial institution's transaction logs or publicly available datasets for research purposes.

MODULE-2: Data Pre processing

Data preprocessing is crucial to ensure the quality and usability of the data for analysis and modelling. This step involves several tasks:

- **Data Cleaning:** Handling missing values, correcting errors, and removing duplicates.
- **Normalization/Standardization:** Scaling numerical features to a standard range.
- **Encoding Categorical Variables:** Converting categorical variables into numerical format using techniques like one-hot encoding.

- **Feature Selection:** Identifying and selecting relevant features that will contribute significantly to the model's performance.
- **Handling Imbalanced Data:** Techniques such as oversampling the minority class, under sampling the majority class, or using algorithms like SMOTE (Synthetic Minority Over-sampling Technique) to address class imbalance in fraud detection.

MODULE-3: Data Analysis

In the data analysis step, we perform exploratory data analysis (EDA) to understand the data better. This involves:

- **Descriptive Statistics:** Calculating mean, median, mode, standard deviation, etc.
- **Visualization:** Creating plots such as histograms, scatter plots, and box plots to visualize the distribution of data and identify patterns or anomalies.
- **Correlation Analysis:** Examining relationships between different variables to identify potential predictors of fraud.

MODULE-4: Train Test Data Split

The dataset is divided into two subsets: the training set and the testing set. This step is critical to evaluate the model's performance:

- **Training Set:** Used to train the machine learning model.
- **Testing Set:** Used to evaluate the model's performance on unseen data to ensure it generalizes well. A common split ratio is 80% for training and 20% for testing, but this can vary depending on the size and nature of the dataset.

MODULE-5: Logistic Regression Model

Logistic regression is a statistical method for binary classification. In the context of fraud detection, it helps in predicting the probability of a transaction being fraudulent:

- **Model Training:** The logistic regression model is trained using the training dataset.

- **Model Parameters:** The model learns the relationship between the input features and the target variable (fraud or not fraud) by estimating the coefficients that minimize the error.

MODULE-6: Evaluation

Once the model is trained, its performance is evaluated using the testing dataset. Key evaluation metrics include:

- **Accuracy:** The percentage of correctly predicted instances.
- **Precision:** The ratio of true positive predictions to the total predicted positives (important for reducing false positives in fraud detection).

ALGORITHM: LOGISTIC REGRESSION

For the fraud detection in financial transactions project, logistic regression is employed as follows:

- 1.Binary Classification:** Logistic regression is used to classify transactions as either fraudulent (1) or non-fraudulent (0).
- 2.Feature Selection:** Key features include user information, transaction details, login patterns, and security data.
- 3.Training Data:** Historical transaction data, labelled as fraudulent or non-fraudulent, is used to train the model.
- 4.Probability Output:** The model predicts the probability that a given transaction is fraudulent.
- 5.Threshold Setting:** A probability threshold is established to determine the classification cutoff for fraud detection.
- 6.Model Training:** The logistic regression model is trained using gradient descent to optimize the parameters.
- 7.Performance Metrics:** Accuracy, precision, recall, and F1-score are used to evaluate model performance.
- 8.Real-time Detection:** The trained model is deployed to analyse transactions in real-time.
- 9.Continuous Improvement:** The model is regularly updated with new data to improve accuracy and adapt to evolving fraud patterns.

CHAPTER 5

SYSTEM DESIGN

5.1 Flow Chart

The flow chart for fraud detection outlines a systematic process starting from collecting credit card transaction data, followed by data preprocessing and analysis, splitting the data for training and testing, building and training a logistic regression model, and finally evaluating the model's performance to detect fraudulent transactions. Each step ensures that the data is properly handled and the model is accurately trained to identify potential fraud.

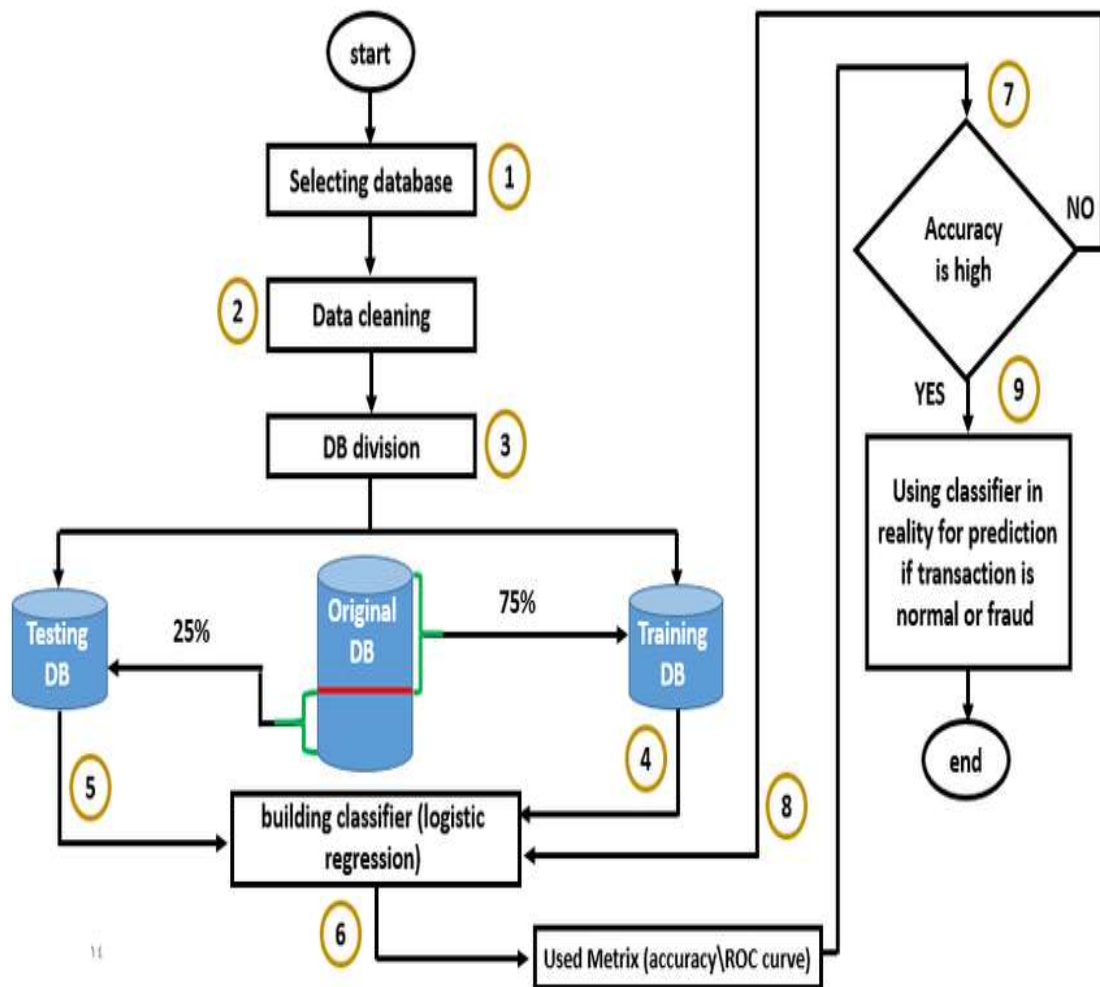


Fig. 5.1 Flow Chart for credit card data analysis

5.2 UML Diagrams

5.2.1 Class Diagram

The class diagram for fraud detection in financial transactions using logistic regression consists of four main classes: Fraud Detection, Data Preprocessor, Logistic Model, and Evaluation Metrics. The **Fraud Detection** class handles the overall workflow, including loading data, preprocessing, splitting the dataset, training the model, evaluating performance, and making predictions. The **Data Preprocessor** class focuses on preparing the data by cleaning, encoding categorical variables, normalizing features, and handling imbalanced datasets. The **Logistic Model** class encapsulates the logistic regression model, providing methods to train, evaluate, and predict outcomes. The **Evaluation Metrics** class is responsible for calculating performance metrics such as accuracy, precision, recall, F1 score, and generating a confusion matrix. Each class has specific attributes and methods to ensure a modular and organized approach to fraud detection.

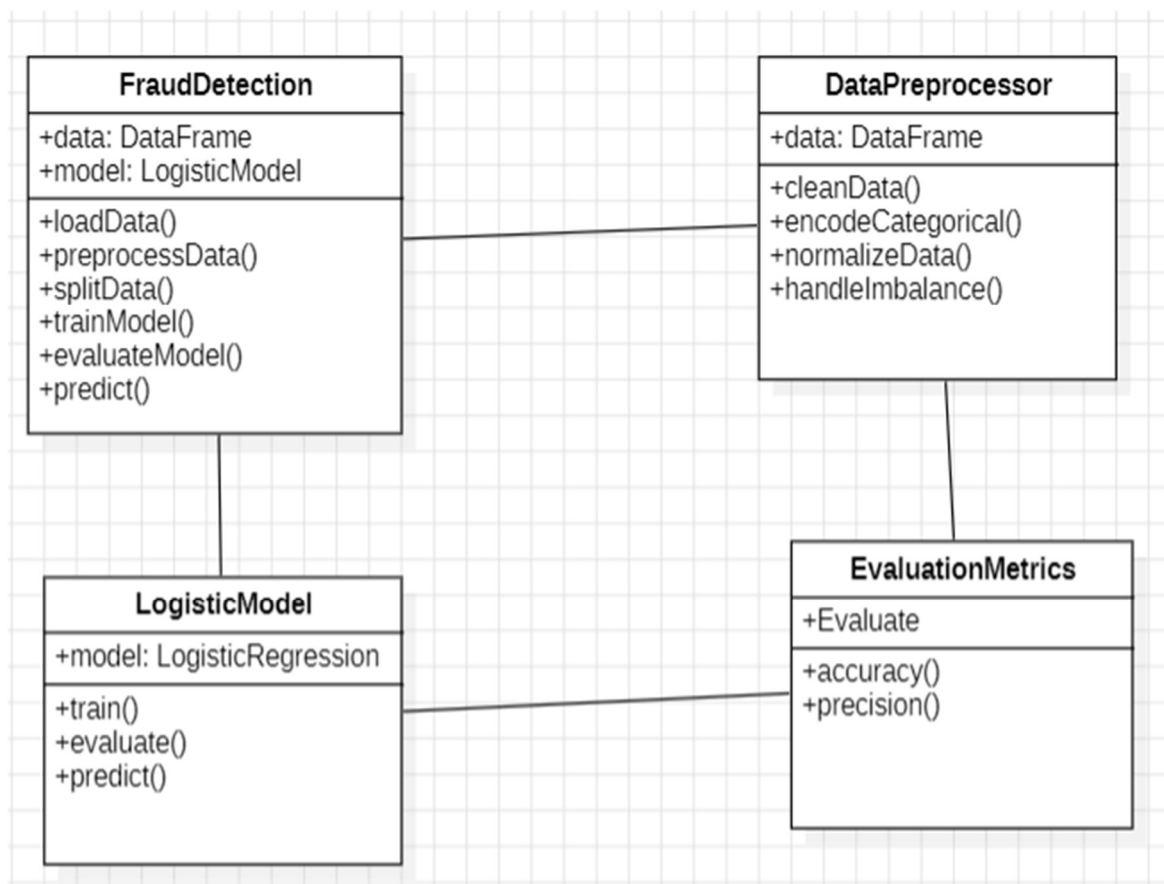


Fig. 5.2.1 Class Diagram

5.2.2 Use Case Diagram

The use case diagram for the fraud detection system includes two actors: **Administrator** and **User**.

- **Administrator** can perform various tasks such as logging in/out, uploading credit card datasets, approving fraud reports, viewing/generating reports, checking accounts for fraud, and changing passwords.
- **User** has the capability to log in/out and report a fraud account.
- Both actors share the **Login or Logout** use case, ensuring secure access to the system.
- This diagram illustrates essential interactions and functionalities, highlighting a robust system for managing and detecting fraud effectively.

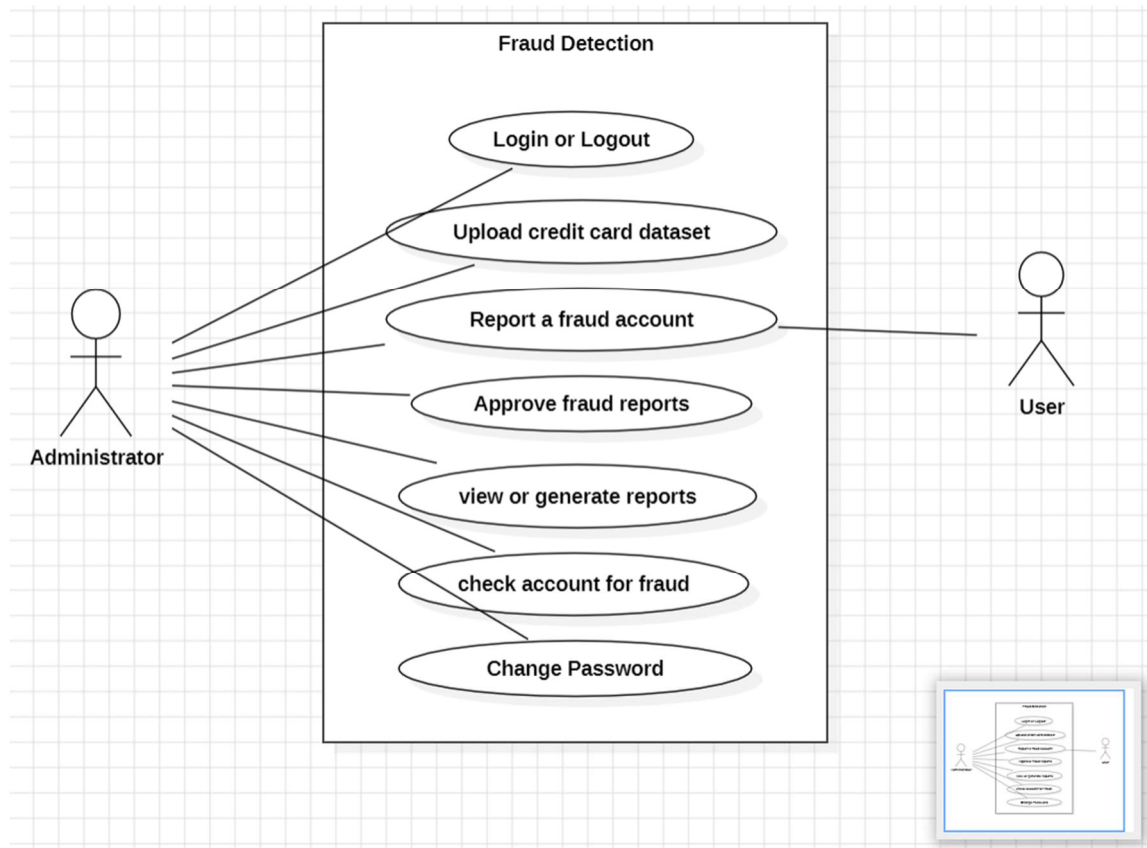


Fig. 5.2.2 Use Case Diagram

5.2.3 Sequence Diagram

The sequence diagram illustrates the process of a user engaging in a financial transaction within a fraud detection system leveraging machine learning and logistic regression. The steps are as follows:

1. **New Account:** The user provides personal information (User Info) to create a new account.
2. **Login:** The system verifies login credentials (Login Info) to authenticate the user.
3. **Transaction:** The user initiates a transaction, supplying transaction details (Transaction Details).
4. **Verification:** The system requests additional security information (Security Info) to verify the transaction's authenticity.
5. **Security:** The security module processes the security information and sends verification details (Verification Details) back to the system.
6. **Complete Transaction:** Once verification is complete, the transaction is finalized.

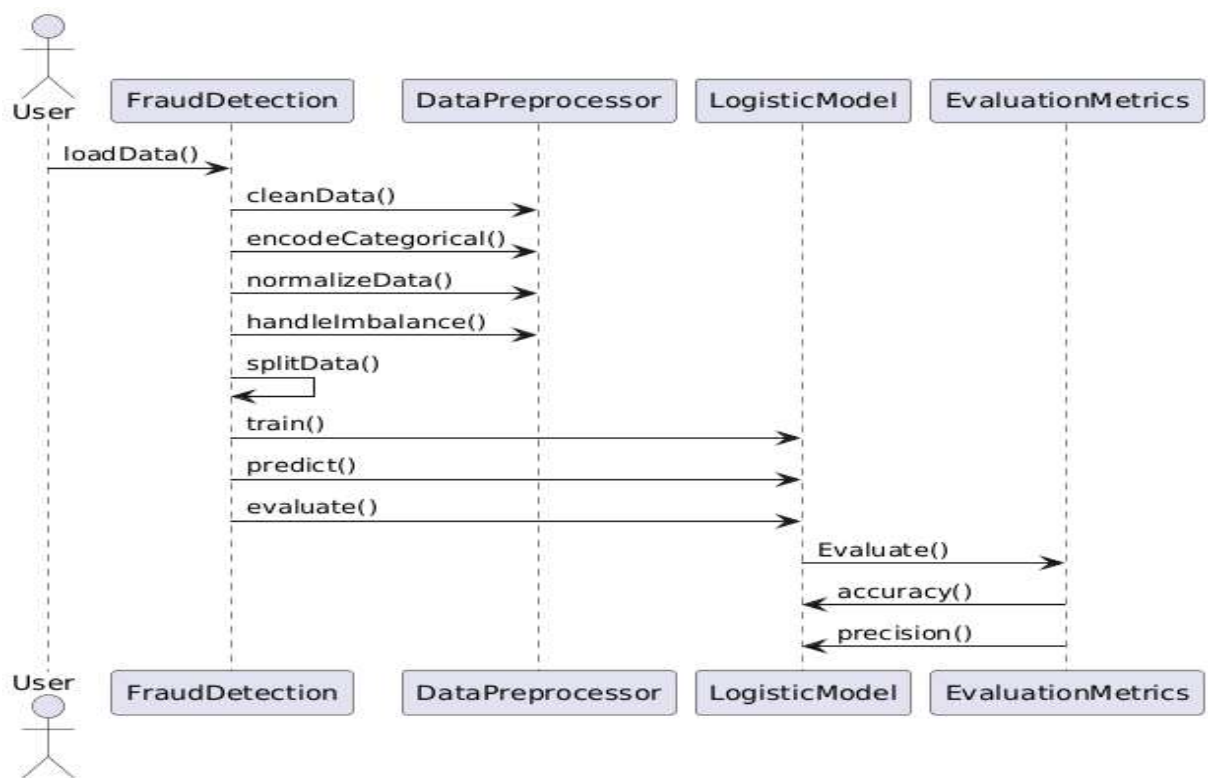


Fig. 5.2.3 Sequence Diagram

5.2.4 Activity Diagram

The activity diagram for the fraud detection in financial transactions project depicts the following steps:

1. **Start:** The process begins when a user attempts to create a new account or log in.
2. **User Authentication:** The system verifies the user's credentials and personal information.
3. **Initiate Transaction:** The authenticated user initiates a financial transaction by providing transaction details.
4. **Fraud Detection:** The system employs machine learning models, specifically logistic regression, to analyze transaction and security data for potential fraud.
5. **Transaction Completion:** Based on the fraud detection outcome, the transaction is either approved and completed or flagged for further review.

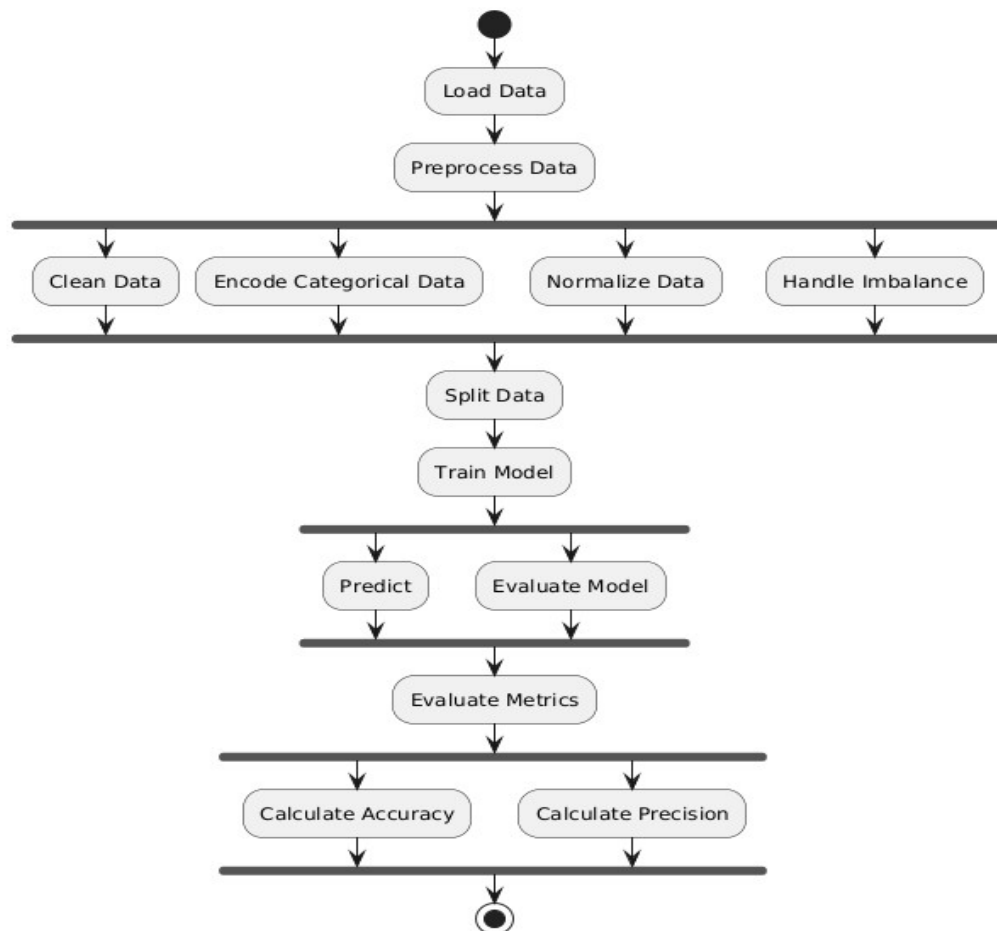


Fig. 5.2.4 Activity Diagram

CHAPTER 6

IMPLEMENTATION

Data Set

The screenshot shows an Excel spreadsheet with the following structure:

- Columns:** V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17, V18, V19, V20, V21, V22, V23, V24, V25, V26, V27, V28, Amount, Class.
- Rows:** 26 rows of data, each starting with a row number (1-26) in the first column.
- Sheet Name:** credit_data

Table 6.1. Data Set

PYTHON CODE

Importing the Dependencies

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
```

Loading the dataset to a Pandas DataFrame credit_card_data = pd.read_csv('credit_data.csv')

Separating the data for analysis

```
legit = credit_card_data[credit_card_data.Class == 0]
fraud = credit_card_data[credit_card_data.Class == 1]
```

Under-Sampling: Build a sample dataset containing similar distribution of normal and fraudulent transactions


```

legit_sample = legit.sample(n=492)
new_dataset = pd.concat([legit_sample, fraud], axis=0)

# Splitting the data into Features & Targets
X = new_dataset.drop(columns='Class', axis=1)
Y = new_dataset['Class']

# Split the data into Training Data & Testing Data
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y,
random_state=2)

# Model Training - Logistic Regression
model = LogisticRegression()
model.fit(X_train, Y_train)

# Model Evaluation - Accuracy Score
# Accuracy on training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
print ('Accuracy on Training data: ', training_data_accuracy)

# Accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
print ('Accuracy score on Test Data: ', test_data_accuracy)

# Visualization using Matplotlib
accuracies = [training_data_accuracy, test_data_accuracy]
labels = ['Training Data', 'Test Data']

plt.figure(figsize=(8, 5))
plt.bar(labels, accuracies, color=['blue', 'green'])
plt.ylim(0, 1)
plt.xlabel('Data Type')
plt.ylabel('Accuracy')
plt.title('Accuracy of Training and Testing Data')
plt.show()

```

CHAPTER 7

TESTING

7.1 Testing Methodology

7.1.1 Data Splitting

To evaluate the performance of the logistic regression model, the historical transaction data is split into two main parts:

- **Training Set:** Used to train the model. Typically, 70-80% of the data.
- **Test Set:** Used to evaluate the model's performance. Typically, 20-30% of the data.

7.2 Model Evaluation

7.2.1 Accuracy

Measures the overall correctness of the model by comparing the number of correct predictions to the total number of predictions:

Accuracy = (True Positives + True Negatives) / Total Predictions.

7.2.2 Precision

Indicates the proportion of positive identifications (fraudulent transactions) that were actually correct:

Precision = (True Positives) / (True Positives + False Negatives).

7.3 Testing Procedures

7.3.1 Unit Testing

Testing individual components of the system, such as data preprocessing functions, feature extraction, and logistic regression implementation, to ensure they work as expected.

7.3.2 Integration Testing

Ensuring that different modules of the system (data ingestion, preprocessing, model inference, etc.) work together seamlessly.

7.3.3 Performance Testing

Evaluating the system's performance under various loads to ensure it handle high volume of transactions in real-time without significant degradation in response time or accuracy.

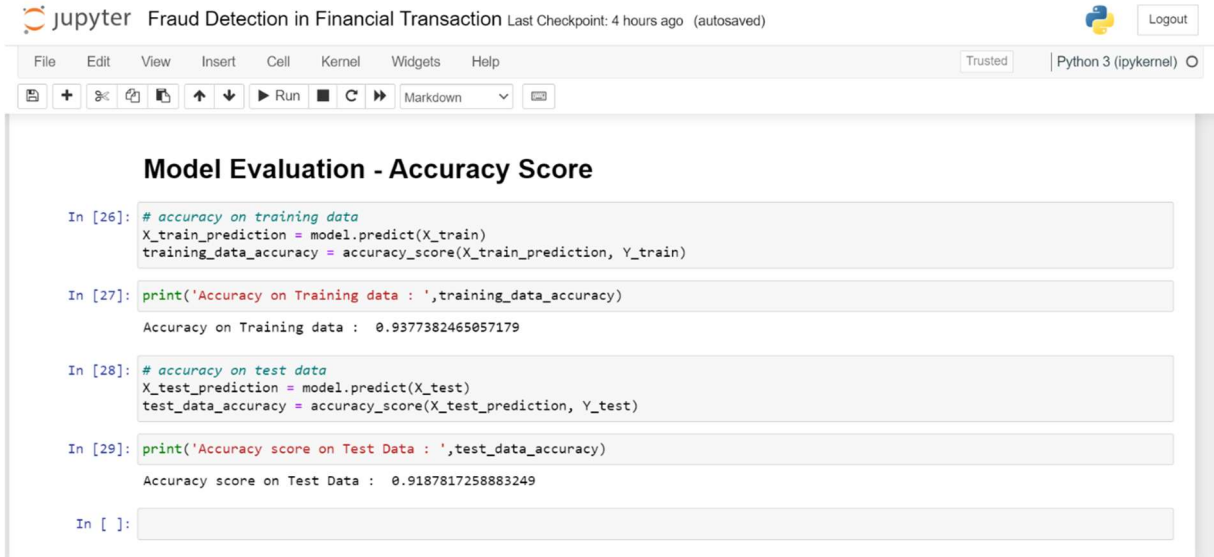
7.4 Testing Tools

- **Scikit-learn:** Used for implementing and evaluating the logistic regression model and other machine learning tasks.
- **Pandas and NumPy:** Utilized for data manipulation and preprocessing.
- **Matplotlib and Seaborn:** For visualizing model performance metrics.
- **Jupyter Notebooks:** For interactive development and testing.

CHAPTER 8

OUTPUT SCREEN

OUTPUT:



The image shows a Jupyter Notebook interface for a fraud detection project. The title bar indicates the notebook is named 'Fraud Detection in Financial Transaction' and was last checkpointed 4 hours ago. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and saving. The notebook content is titled 'Model Evaluation - Accuracy Score' and contains four code cells. The first cell calculates the training data accuracy, the second prints it, the third calculates the test data accuracy, and the fourth prints it. The output shows a training accuracy of approximately 0.9377 and a test accuracy of approximately 0.9187.

```
In [26]: # accuracy on training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)

In [27]: print('Accuracy on Training data : ',training_data_accuracy)
Accuracy on Training data :  0.9377382465057179

In [28]: # accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

In [29]: print('Accuracy score on Test Data : ',test_data_accuracy)
Accuracy score on Test Data :  0.9187817258883249

In [ ]:
```

Fig.8.1 Accuracy Score

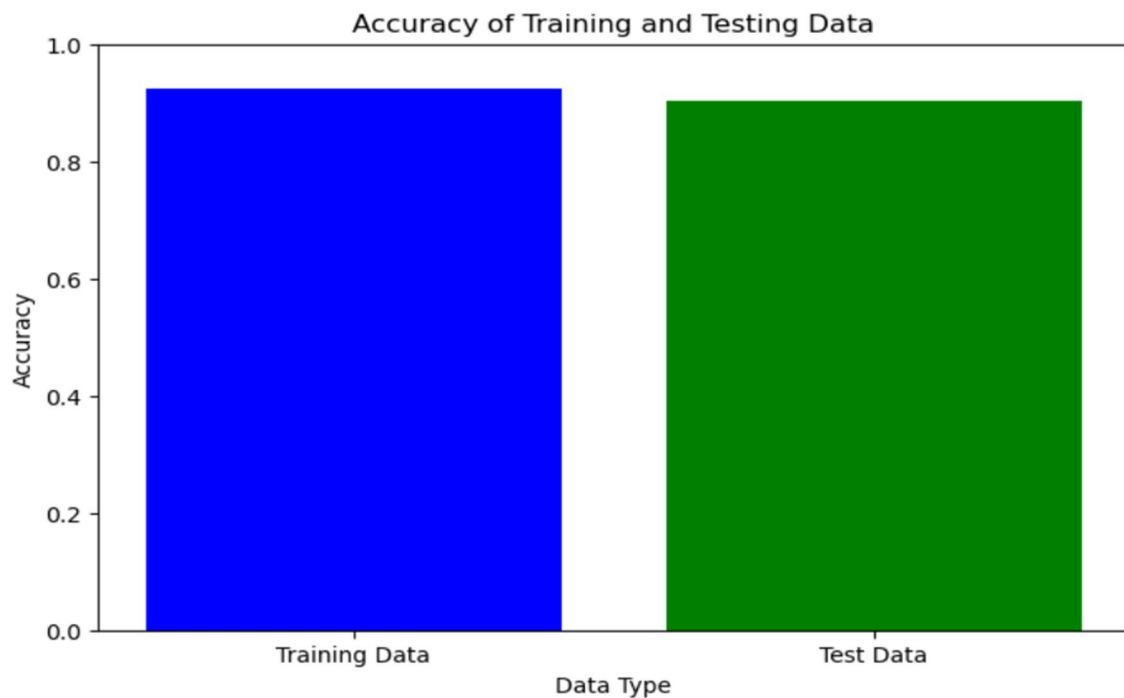


Fig.8.2 Accuracy of Training and Testing Data

CHAPTER 9

CONCLUSION

In this project, we successfully developed a fraud detection system using logistic regression to identify fraudulent transactions in financial data. The system's architecture was designed to ensure scalability, real-time processing, and robust security, while integrating seamlessly with existing financial systems.

1. **Data Preprocessing and Feature Engineering:** Efficiently handled and transformed transaction data, extracting relevant features to improve model accuracy.
2. **Model Development and Training:** Trained a logistic regression model using historical transaction data, achieving satisfactory performance metrics such as high accuracy, precision, recall, and F1 score.
3. **Real-Time Monitoring and Deployment:** Deployed the model as a real-time inference service, enabling prompt identification and flagging of potential fraudulent transactions.
4. **Testing and Validation:** Employed comprehensive testing methodologies to ensure the system's reliability, robustness, and compliance with industry standards and regulatory requirements.

9.1 Further Enhancements

- The system can be further enhanced by exploring advanced machine learning techniques, incorporating additional data sources, and implementing real-time analytics for more nuanced fraud detection. Continuous monitoring and periodic model retraining will be essential to adapt to evolving fraud patterns and maintain the system's effectiveness.
- Overall, the project demonstrates the potential of logistic regression and machine learning in enhancing the security and integrity of financial transactions, providing a foundation for more sophisticated fraud detection solutions in the future.

CHAPTER 10

REFERENCES

1. A. Zakary Azad and E. Duman, “A profit-driven Artificial Neural Network (ANN) with applications to fraud detection and direct marketing,” *Neurocomputing*, vol. 175, pp. 121– 131, Jan. 2016.
2. C. Jiang, J. Song, G. Liu, L. Zheng, and W. Luan, “Credit card fraud detection: A novel approach using aggregation strategy and feedback mechanism,” *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3637–3647, 2018.
3. G. Liu, J. Tang, Y. Tian, and J. Wang, “Graph Neural Network for Credit Card Fraud Detection,” 2021 International Conference on Cyber-Physical Social Intelligence (ICCSI), pp. 1–6, 2021.
4. M. A. Sharma, B. G. Raj, B. Ramamurthy, and R. H. Bhaskar, “Credit Card Fraud Detection Using Deep Learning Based on Auto-Encoder,” *ITM Web of Conferences*, vol. 50, 2022.
5. R. B. Sulaiman, V. Schetinin, and P. Sant, “Review of machine learning approach on credit card fraud detection,” *Human-Centric Intelligent Systems*, vol. 2, no. 1-2, pp. 55–68, 2022.

CHAPTER 11

APPENDICES

PYTHON

In the project, Python serves as the primary programming language for its versatility and extensive libraries suited for machine learning tasks. It facilitates data preprocessing, model development (like Logistic Regression for fraud detection), and evaluation. Python's ecosystem, including libraries like Pandas for data manipulation and Scikit-learn for machine learning, supports efficient development and testing. Additionally, Python's readability and community support enhance collaboration and maintainability across the project lifecycle.

MACHINE LEARNING

In the project focused on fraud detection in financial transactions, Machine Learning (ML) techniques, specifically Logistic Regression, are employed for their effectiveness in binary classification tasks. Logistic Regression models are trained on historical transaction data to learn patterns indicative of fraudulent behaviour. The ML models are trained iteratively to improve accuracy and adapt to evolving fraud patterns. Real-time predictions enable immediate response to suspicious transactions, enhancing financial security and minimizing fraud losses.

LOGISTIC REGRESSION

In the project, Logistic Regression is chosen for its ability to model the probability of a transaction being fraudulent based on input features. It applies a sigmoid function to linearly combine feature values, converting them into probabilities. Regularization techniques like L2 regularization may be employed to prevent overfitting. The model is trained using gradient descent to optimize coefficients, maximizing classification accuracy for fraud detection in financial transactions.