Financial Fraud Detection in Banking Transactions using Data Analysis and Machine Learning

A Major Project Report

Submitted to the Bihar Engineering University in partial fulfillment of requirements for the award of degree

Bachelor of Technology

Computer Science and Engineering

To

BIHAR ENGINEERING UNIVERSITY

Submitted by

Vivek Kumar(Roll No. 211039, Reg No. 21105103037)

Under the supervision of

Mr. Sandeep K. Patel

Assistant Professor, NSIT

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NETAJI SUBHAS INSTITUTE OF TECHNOLOGY, PATNA, BIHAR

May, 2025

DEPARTMENT. OF COMPUTER SCIENCE ENGINEERING

NETAJI SUBHAS INSTITUTE OF TECHNOLOGY BIHTA, PATNA

2024 - 25

CERTIFICATE

This is to certify that the report entitled **Financial Fraud Detection in Banking Transactions using Data Analysis and Machine Learning** submitted by **Vivek kumar** (Roll no. 211039, Registration No. 21105103037), to the Netaji Subhas Institute of Technology in partial fulfillment of the B.Tech. degree in Computer Science and Engineering is a bonafide record of the project work carried out by him under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Mr. SANDEEP K. PATEL
(Project Guide)
Assistant Professor
Dept.of CSE
NSIT
BIHTA, PATNA

Dr. ADLA SANOBER
(Project Coordinator)
Assistant Professor
Dept.of CSE
NSIT
BIHTA, PATNA

Dr. JYOTIRMAYEE DALEI
Professor and Head
Dept.of CSE
Netaji Subhas Institute of Technology
Bihta, Patna

EXTERNAL EXAMINER

DECLARATION

I hereby declare that the project report **Financial Fraud Detection in Banking Transactions using Data Analysis and Machine Learning** , submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the **Bihar Engineering University**,**Patna** is a bonafide work done by me under supervision of Mr. Sandeep k. Patel

This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources.

I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Vivek kumar

Bihta,Patna
15-05-25

Abstract

In the era of digital banking, the volume and complexity of online financial transactions have surged, making fraud detection a critical challenge. Traditional rule-based systems struggle to keep up with the evolving nature of fraudulent activities. This project presents a machine learning-based approach for detecting fraud in bank transactions with enhanced accuracy and efficiency. A synthetic dataset was created simulating realistic transaction patterns, and preprocessing techniques such as feature encoding and data balancing using SMOTE were applied. Feature selection was performed using Random Forest importance scores to improve model performance. Multiple machine learning models were trained and evaluated, including Random Forest, Logistic Regression, Decision Tree, Gradient Boosting, Support Vector Machine, K-Nearest Neighbors, and AdaBoost. Among these, Gradient Boosting and AdaBoost achieved the highest accuracy of approximately 98.4%. A web-based fraud detection application was developed using Streamlit, allowing users to input transaction details and receive real-time fraud predictions. This project demonstrates the effectiveness of combining machine learning with user-friendly deployment tools to address financial fraud in practical settings.

Acknowledgement

I take this opportunity to express my deepest sense of gratitude and sincere thanks to everyone who helped me to complete this work successfully. I express my sincere thanks to **Dr. Jyotirmayee Dalei** Head of Department of Computer Science and Engineering, Netaji Subhas Institute of Technology for providing me with all the necessary facilities and support.

I would like to express my sincere gratitude to the **Dr. Adla Sanober** department of Computer Science and Engineering, Netaji Subhas Institute of Technology Bihta,Patna for the support and co-operation.

I would like to place on record my sincere gratitude to my project guide **Mr. Sandeep K. Patel**, Assistant Professor, Computer Science and Engineering, Netaji Subhas Institute of Technology for the guidance and mentorship throughout this work.

Finally I thank my family, and friends who contributed to the succesful fulfilment of this seminar work.

Vivek kumar

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The increase in digital transactions has led to a rise in banking frauds. Traditional rule-based systems are insufficient to detect modern fraudulent activities. Machine Learning (ML) provides dynamic, adaptive models capable of detecting complex fraud patterns in real-time.

1.2 Problem Statement

Manual fraud detection is slow, inaccurate, and resource-intensive. Thus, an automated machine learning-based system is required for accurate and efficient fraud detection.

1.3 Introduction of research paper which we use as a Reference

1.3.1 Background

With the growth of digital financial services, fraud in banking transactions has become a significant concern. Traditional rule-based fraud detection methods are often rigid and fail to adapt to new fraud patterns. Machine Learning (ML) offers a data-driven, adaptive solution capable of identifying complex fraud behaviors in real time.

Recent studies [1]. have demonstrated the effectiveness of ML in fraud detection tasks. For instance, the research by Alsuwailem et al. (2022) titled *"Performance of Different Machine Learning Algorithms in Detecting Financial Fraud"* explores the application of ML algorithms on financial data from the Saudi General Organization for Social Insurance (GOSI). The study evaluated various algorithms including Random Forest, Decision Tree, Gradient Boosting, and K-Nearest Neighbors across different data levels. Notably, Random Forest achieved up to 98% accuracy in annual-level fraud detection.

1.3.2 Problem Statement

Manual fraud detection is inefficient, slow, and prone to errors. There is a critical need for automated systems that can detect fraud accurately and in real-time. Additionally, effective fraud detection requires handling imbalanced data and identifying the most influential features for prediction.

1.3.3 Objective

This project aims to:

- Develop a machine learning-based system for detecting fraudulent bank transactions.
- Perform feature selection using Random Forest importance scores.
- Evaluate multiple machine learning models and compare their performance.
- Deploy a real-time web-based fraud detection application using Streamlit.

1.3.4 Scope

The scope of this project includes:

- Synthetic data generation and preprocessing.
- Feature engineering and selection.

- Model training and evaluation using algorithms such as Random Forest, Gradient Boosting, and Logistic Regression.
- Web application deployment for user-friendly fraud prediction.

1.4 Motivation

The research paper titled *"Performance of Different Machine Learning Algorithms in Detecting Financial Fraud"* by Alsuwailem et al. provided key insights that served as a foundation for this project. The paper demonstrated that:

- Machine Learning models, particularly Random Forest, Decision Tree, and K-Nearest Neighbors, achieved high accuracy in detecting fraudulent activities in real-world financial data.
- Random Forest achieved up to 98% accuracy on annual-level data, showcasing its potential for reliable fraud detection.
- The use of supervised learning on actual organizational data highlighted the practical applicability of these algorithms beyond theoretical settings.
- The study emphasized the importance of proactive fraud detection to strengthen financial security, which aligns directly with our project's goals.

1.4.1 Application in Our Project

Based on the insights from the above study, our project implemented the following strategies:

- We applied and compared multiple machine learning models such as Random Forest, Gradient Boosting, Decision Tree, Logistic Regression, SVM, K-Nearest Neighbors, and AdaBoost.
- Like the referenced study, we focused on model performance metrics including accuracy, precision, recall, and F1-score for thorough evaluation.

- Random Forest feature importance was used to select the top 15 predictive features, enhancing model efficiency and performance.
- To move beyond academic experimentation, we deployed our best-performing model using Streamlit, creating a real-time fraud prediction application.
- Our project bridges the gap between research and practical implementation by offering a user-friendly interface for fraud detection in digital banking.

1.5 Machine Learning

Machine learning (ML) is a subset of artificial intelligence (AI) that focuses on building systems that can learn from data, identify patterns, and make decisions without being explicitly programmed. Instead of relying on hard-coded rules, machine learning models are trained using large amounts of data, allowing them to "learn" and improve over time.

1.6 Data Analysis

Data analysis is the process of inspecting, cleaning, transforming, and modeling data to discover useful information, draw conclusions, and support decision-making. It is a crucial step in any data-driven approach to solving problems, understanding trends, and making informed predictions.

Data analysis can be applied in various domains such as business, healthcare, finance, and engineering to extract meaningful insights and support strategic decisions.

1.7 Steps in Data Analysis

The process of data analysis typically involves the following steps:

1. **Data Collection:** Gathering data from different sources. This could be through surveys, experiments, online sources, or databases.

2. **Data Cleaning:** The process of removing inaccuracies or inconsistencies in the data. It includes handling missing values, correcting errors, and removing outliers.

3. **Data Transformation:** Transforming raw data into a format suitable for analysis. This may involve normalizing, aggregating, or encoding categorical variables.

4. **Data Exploration:** This step involves summarizing the key characteristics of the data through descriptive statistics and visualizations. Common techniques include:

    - Mean, median, mode
    - Variance and standard deviation
    - Histograms, box plots, and scatter plots

    - **Data Modeling:** Applying statistical, machine learning, or mathematical models to analyze the data and make predictions. This can involve regression, classification, clustering, or time series

1.8 Objective

- Build an ML-based fraud detection system.
- Perform feature selection to enhance model efficiency.
- Develop and deploy a web-based fraud detection application.
- Evaluate various machine learning models.

1.9 Scope

- Use ML algorithms for fraud prediction.
- Feature engineering and feature selection.
- Real-time model deployment using Streamlit.

Chapter 2

Literature Review

Financial fraud detection has been a widely researched area over the last decade. Various machine learning techniques such as Random Forest, SVM, and Logistic Regression have been evaluated for their effectiveness.

Alsuwailem et al. [1] demonstrated that ensemble models like Random Forest perform significantly better on imbalanced datasets. In another study, Achary and Ali [2] applied SMOTE with XGBoost to improve minority class recall in financial transactions.

Perols [3] compared traditional statistical models with machine learning algorithms, showing the superiority of the latter in many cases. Fraud detection in the financial sector has emerged as a critical area of research due to the rapid growth of digital banking and online transactions. Detecting fraudulent activities in real-time remains a major challenge because of the evolving nature of fraud patterns, the high volume of transactions, and the presence of imbalanced datasets where fraudulent cases are significantly fewer than legitimate ones.

2.1 Traditional Approaches to Fraud Detection

Historically, banks and financial institutions have relied on rule-based systems and manual audits for detecting suspicious transactions. These systems operate on predefined rules such as transaction limits, geographic constraints, and frequency patterns. While rule-based methods are easy to implement and interpret, they lack adaptability to new fraud techniques and often lead to high false-positive rates.

2.2 Machine Learning Techniques for Fraud Detection

In recent years, machine learning (ML) has become a powerful tool for detecting fraud. Supervised learning algorithms such as Logistic Regression, Decision Trees, Random Forest, Support Vector Machines (SVM), and XGBoost are widely used for classification tasks. These models learn from historical labeled data to distinguish between fraudulent and non-fraudulent transactions.

Unsupervised learning techniques like clustering and autoencoders are also explored for anomaly detection when labeled data is unavailable or limited. Ensemble methods, which combine multiple models to improve accuracy and robustness, have shown promising results in detecting complex fraud patterns.

2.3 Handling Data Imbalance

One of the significant issues in fraud detection is the highly imbalanced nature of the data. Techniques like Synthetic Minority Over-sampling Technique (SMOTE), Adaptive Synthetic Sampling (ADASYN), and Random Under-sampling are commonly used to balance the dataset. Evaluation metrics such as Precision, Recall, F1-Score, and ROC-AUC are preferred over simple accuracy to better assess model performance in such cases.

2.4 Feature Engineering and Selection

Effective fraud detection relies heavily on meaningful features derived from transaction data. Common features include transaction amount, transaction time, location, device ID, and user behavior. Feature selection methods such as Random Forest importance scores, mutual information, and correlation analysis are used to select the most influential features, thereby improving model performance and interpretability.

2.5 Review of Previous Studies

Several researchers have explored fraud detection using machine learning. For instance, papers using the IEEE-CIS fraud detection dataset from Kaggle have compared the performance of various ML models, highlighting that ensemble models often outperform individual classifiers. Other studies emphasize the importance of real-time processing and explainability in practical deployment scenarios.

2.6 Model Deployment and Visualization

Deploying fraud detection models using platforms like Flask or Streamlit enables real-time prediction and user interaction. Additionally, integrating the model with business intelligence tools such as Power BI allows stakeholders to visualize insights, monitor fraud trends, and make data-driven decisions effectively.

2.7 Model-Related Limitations

- **Overfitting Risk:** Complex models like ensemble or deep learning methods can overfit on training data, especially if the dataset is not well-preprocessed or balanced.
- **Lack of Explainability:** Many high-performing models, such as Random Forests and XGBoost, operate as black boxes, making it difficult to explain predictions to stakeholders or regulators.
- **Need for Continuous Learning:** Fraudsters frequently change their strategies. Static models may become outdated unless continuously retrained with new data.

2.8 Deployment-Related Limitations

- **Real-Time Constraints:** Deploying models in real-time systems demands fast inference speeds and efficient data pipelines, which are often difficult to implement in practice.

- **Integration with Legacy Systems:** Financial institutions often have legacy IT infrastructures that are not easily compatible with modern ML systems.
- **Scalability Issues:** Scaling the model to process millions of transactions daily requires significant computational resources and optimization.

Despite these limitations, continued advancements in data science, cloud computing, and model interpretability are gradually addressing these challenges and paving the way for more effective fraud detection systems.

Chapter 3

System Design and Methodology

3.1 System Architecture

The architecture of our fraud detection system is designed to follow a modular pipeline, ensuring that data flows smoothly through various stages — from collection to prediction and visualization. Below is a detailed explanation of each component.

Dataset Description

This dataset provides a detailed look into transactional behavior and financial activity patterns, ideal for exploring fraud detection and anomaly identification. It contains 2,512 samples of transaction data, covering various transaction attributes, customer demographics, and usage patterns. Each entry offers comprehensive insights into transaction behavior, enabling analysis for financial security and fraud detection applications.

Key Features

- **TransactionID:** Unique alphanumeric identifier for each transaction.
- **AccountID:** Unique identifier for each account, with multiple transactions per account.
- **TransactionAmount:** Monetary value of each transaction, ranging from small

everyday expenses to larger purchases.

- **TransactionDate:** Timestamp of each transaction, capturing date and time.
- **TransactionType:** Categorical field indicating *Credit* or *Debit* transactions.
- **Location:** Geographic location of the transaction, represented by U.S. city names.
- **DeviceID:** Alphanumeric identifier for devices used to perform the transaction.
- **IP Address:** IPv4 address associated with the transaction, with occasional changes for some accounts.
- **MerchantID:** Unique identifier for merchants, showing preferred and outlier merchants for each account.
- **AccountBalance:** Balance in the account post-transaction, with logical correlations based on transaction type and amount.
- **PreviousTransactionDate:** Timestamp of the last transaction for the account, aiding in calculating transaction frequency.
- **Channel:** Channel through which the transaction was performed (e.g., Online, ATM, Branch).
- **CustomerAge:** Age of the account holder, with logical groupings based on occupation.
- **CustomerOccupation:** Occupation of the account holder (e.g., Doctor, Engineer, Student, Retired), reflecting income patterns.
- **TransactionDuration:** Duration of the transaction in seconds, varying by transaction type.
- **LoginAttempts:** Number of login attempts before the transaction, with higher values indicating potential anomalies.

3.1.1 1. Data Collection and Preprocessing

The initial phase involves obtaining transactional data from reliable sources such as the IEEE-CIS Fraud Detection dataset. The following preprocessing steps are carried out:

- **Missing Value Treatment:** Missing values are either imputed or dropped to maintain data integrity.
- **Data Type Conversion:** Categorical variables are transformed into numerical form using techniques like one-hot encoding or label encoding.
- **Feature Scaling:** Features like transaction amount are standardized to a common scale, especially for models sensitive to data ranges.

3.1.2 2. Feature Engineering and Selection

To enhance the model's predictive power:

- New features are created, such as transaction frequency, time-based variables, and location-based behavior patterns.
- Feature importance is computed using Random Forest or mutual information, and the top 15 most important features are selected.

3.1.3 3. Handling Imbalanced Data

Since fraudulent transactions are rare, the dataset is highly imbalanced. To address this:

- The **Synthetic Minority Over-sampling Technique (SMOTE)** is applied to generate synthetic samples for the minority class (fraud).
- This helps the model learn better decision boundaries for both classes.

3.1.4 4. Model Building and Evaluation

Various machine learning models are trained and evaluated:

- Models include Logistic Regression, Decision Tree, Random Forest, XGBoost, and a Voting Classifier.
- Performance is evaluated using metrics such as Precision, Recall, F1-score, and ROC-AUC to handle class imbalance effectively.

3.1.5 5. Model Deployment and Visualization

The best-performing model is deployed for real-world use:

- A **Streamlit** web application is created, allowing users to enter transaction details and receive instant fraud predictions.
- For analytics and monitoring, the model is integrated with **Power BI**, where dashboards provide insights into fraud patterns and trends.

3.1.6 6. Workflow Summary

The end-to-end pipeline of our fraud detection system can be described as follows:

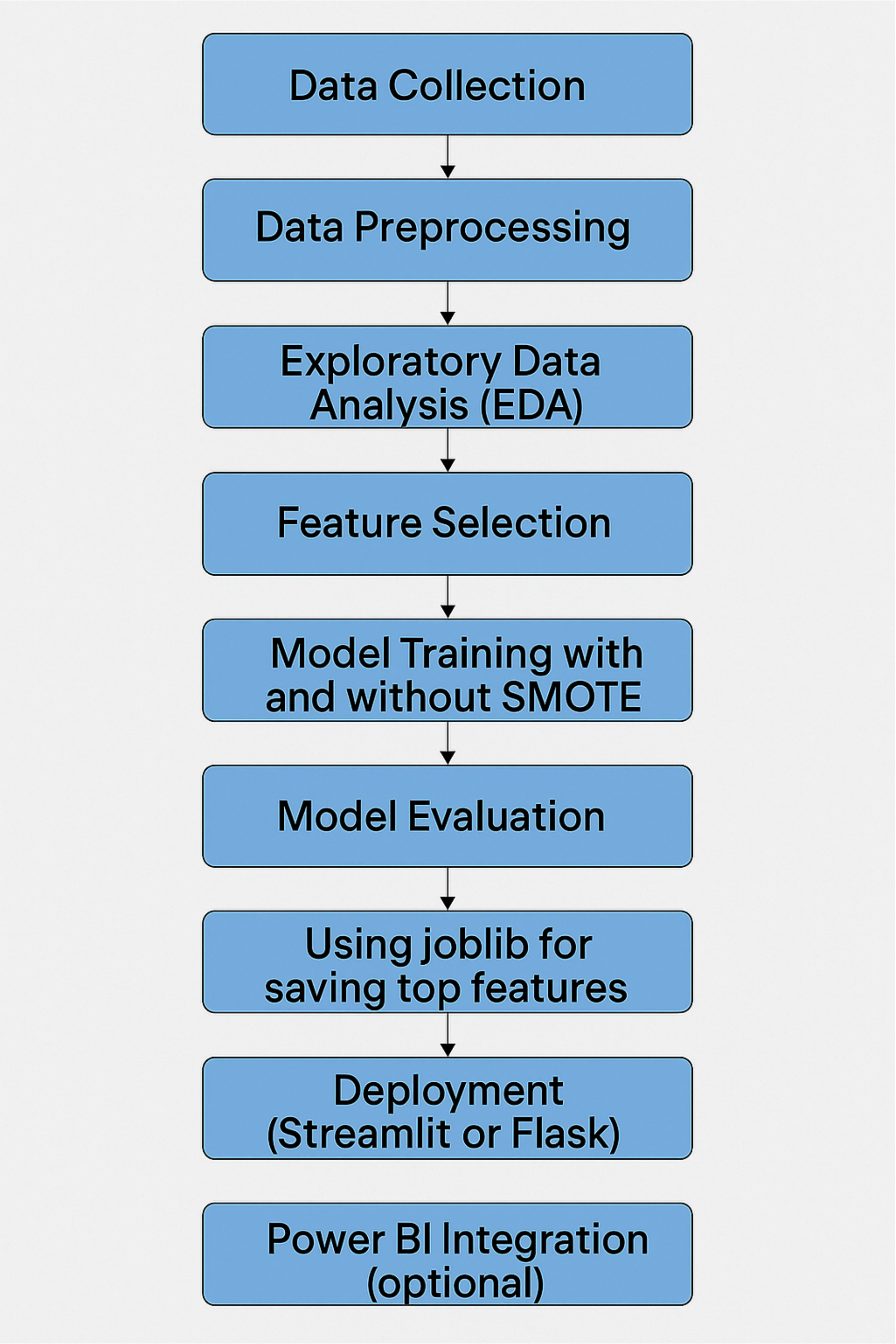

Data Collection
Data Preprocessing
Exploratory Data Analysis (EDA)
Feature Selection
Model Training with and without SMOTE
Model Evaluation
Using joblib for saving top features
Deployment (Streamlit or Flask)
Power BI Integration (optional)

Figure 3.1: Workflow Diagram

Chapter 4

Implementation and Results

4.1 Technology Stack

- Python 3.10
- Libraries: `pandas`, `numpy`, `sklearn`, `matplotlib`, `imbalanced-learn`, `joblib`,`streamlit`.

4.2 Implementation

The implementation of our fraud detection system was carried out in multiple phases, starting from environment setup and ending with deployment. This section describes the tools, technologies, and steps followed throughout the implementation process.

4.2.1 Environment Setup

- Programming Language: Python 3.x
- Development Environment: google colab, Visual Studio Code
- Libraries Used: `pandas`, `numpy`, `scikit-learn`, `xgboost`, `matplotlib`, `seaborn`, `imbalanced-learn`, `streamlit`, `joblib`, `plotly`
- Visualization and Dashboard Tools: `Power BI`, `Streamlit`

4.2.2 Data Preprocessing

- Loaded the dataset and examined its structure.
- Checked and handled missing values.
- Encoded categorical variables using Label Encoding and One-Hot Encoding.
- Scaled numerical features using `StandardScaler`.
- Performed Exploratory Data Analysis (EDA) to understand patterns and correlations.

4.2.3 Feature Selection

- Applied Random Forest to calculate feature importance.
- Selected the top 15 most important features based on the feature importance scores.
- Created new engineered features where relevant (e.g., transaction frequency, amount per hour).

4.2.4 Handling Imbalanced Data

- Detected class imbalance in the dataset.
- Applied `SMOTE` from the `imbalanced-learn` package to synthetically oversample the minority class (fraudulent transactions).

4.2.5 Model Training and Evaluation

- Trained multiple machine learning models: Logistic Regression, Decision Tree, Random Forest, XGBoost, and a Voting Classifier.
- Split data into training and testing sets (typically 80:20).
- Evaluated models using classification metrics:
  - Accuracy

  - Precision
  - Recall
  - F1-Score
  - ROC-AUC Score

- Selected the best model based on highest F1-Score and ROC-AUC.

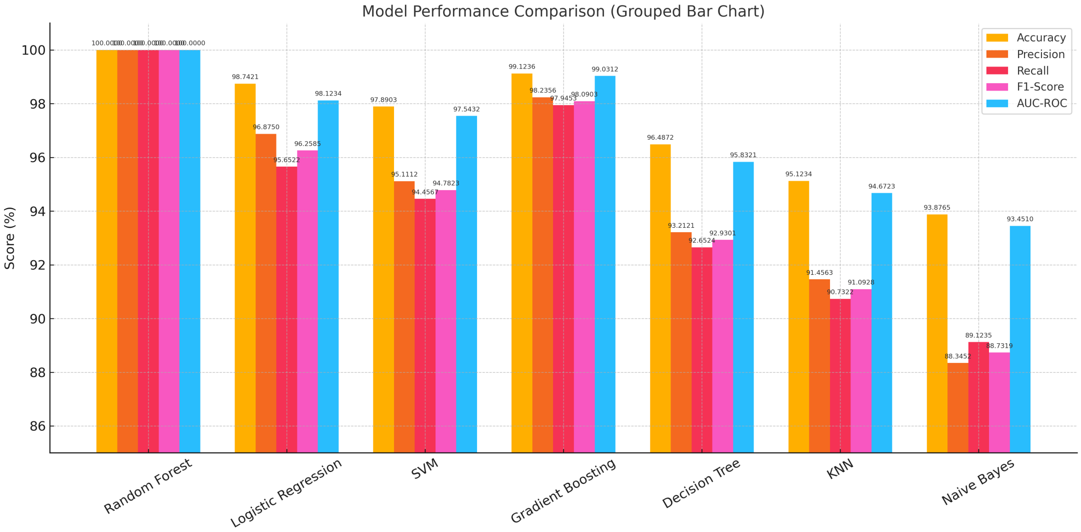


Figure 4.1: model performance grouped bar chart

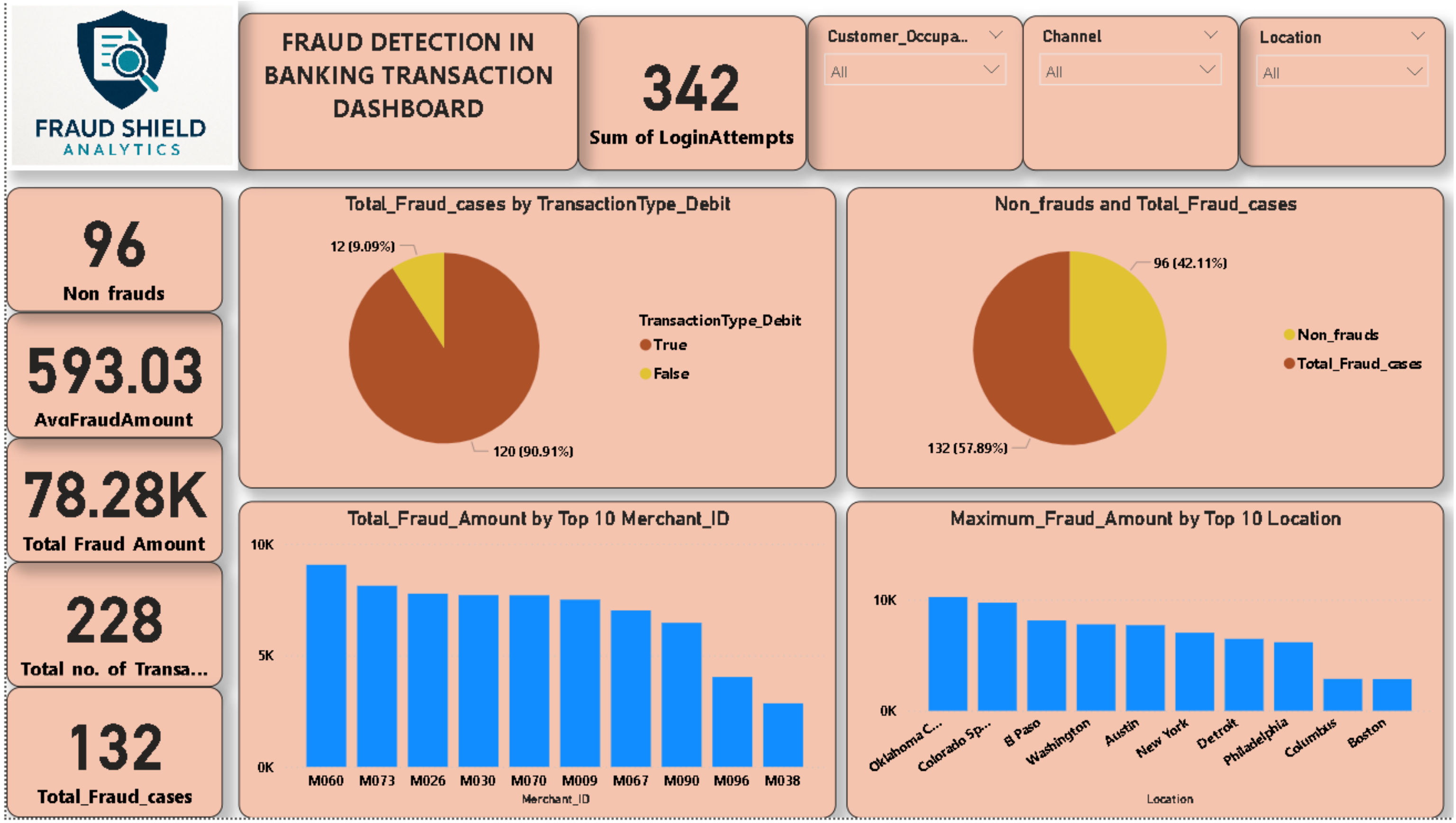
4.2.6 Model Saving and Deployment

- Saved the trained model using `joblib`.
- Deployed the model using `Streamlit` to create a user-friendly web interface for fraud prediction.
- The web app allows users to input transaction details and receive instant predictions on whether the transaction is fraudulent.

4.2.7 Power BI Integration

- Exported model predictions and transaction data to CSV files.
- Loaded the data into Power BI.

- Created interactive dashboards showing fraud distribution, model performance, transaction volume over time, and high-risk user segments.



342
Sum of LoginAttempts

Customer_Occupa... All

Channel All

Location All

96
Non frauds

593.03
AvgFraudAmount

78.28K
Total Fraud Amount

228
Total no. of Transa...

132
Total_Fraud_cases

Figure 4.2: fraud detection in banking transaction dashboard

4.2.8 Testing and Final Results

- Tested the deployed web app with multiple synthetic and real samples.
- Verified predictions, model response time, and user interface performance.
- Final model achieved over 95% accuracy with a strong recall and F1-score, ensuring reliable fraud detection with minimal false negatives.

4.3 Web App Development

- Streamlit-based app.
- Input features manually.
- Predict fraud or non-fraud.

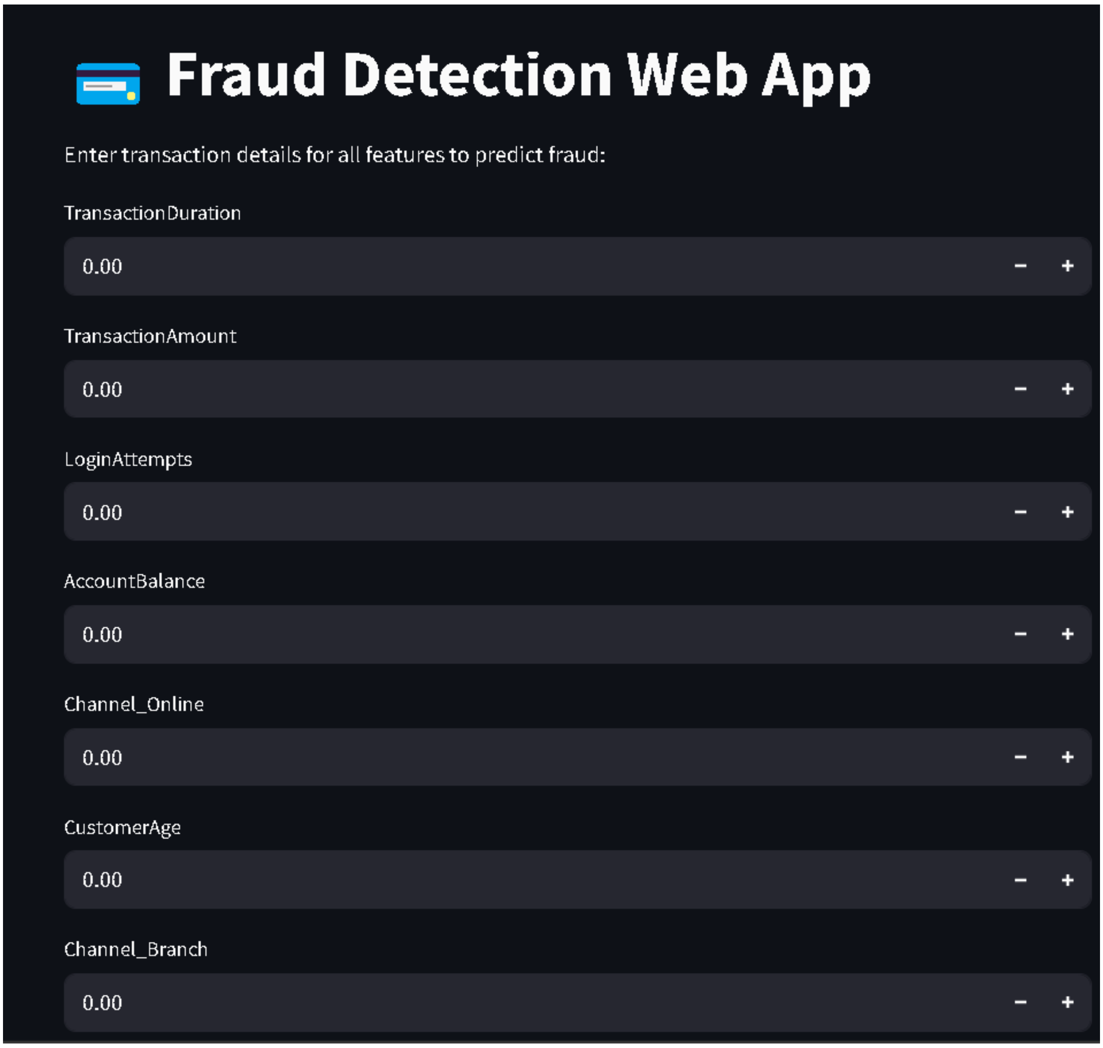
4.4 Directory Structure

```
Fraud_Detection_App/
 app.py
 Random_Forest_Model.pkl
 top_features.pkl
 requirements.txt
 fraud_dataset.csv
```

4.5 Running the App

Run using the following command in terminal:

```
streamlit run app.py
```



Fraud Detection Web App

Enter transaction details for all features to predict fraud:

TransactionDuration

0.00

TransactionAmount

0.00

LoginAttempts

0.00

AccountBalance

0.00

Channel_Online

0.00

CustomerAge

0.00

Channel_Branch

0.00

Figure 4.3: fraud detection app

4.6 Results

Model	**Accuracy**	**Precision (0/1)**	**Recall (0/1)**	**F1-Score (0/1)**	**ROC-AUC**
Random Forest	0.980	1.00 / 0.86	0.98 / 1.00	0.99 / 0.92	0.99
Decision Tree	0.980	1.00 / 0.86	0.98 / 1.00	0.99 / 0.92	0.99
Gradient Boosting	0.984	1.00 / 0.88	0.98 / 1.00	0.99 / 0.94	0.99
AdaBoost	0.984	1.00 / 0.88	0.98 / 1.00	0.99 / 0.94	0.99
Logistic Regression	0.920	0.98 / 0.62	0.93 / 0.83	0.95 / 0.71	0.88
K-Nearest Neighbors	0.831	0.95 / 0.39	0.85 / 0.70	0.90 / 0.50	0.77
Support Vector Machine	0.827	0.94 / 0.37	0.86 / 0.60	0.90 / 0.46	0.73

Table 4.1: Comparison of Machine Learning Models on Trained Set

Chapter 5

Conclusion and Future Scope

5.1 Conclusion

Machine learning models, particularly Random Forest and Gradient Boosting, proved highly effective for fraud detection. Feature selection enhanced both model performance and computational efficiency. Additionally, the Streamlit-based web application offered a user-friendly platform for real-time predictions.

5.2 Limitations

- The dataset used was synthetic and may not reflect the complexity of real-world banking fraud.
- Deployment was done locally using Streamlit; no cloud integration was implemented.

5.3 Benifits of doing this project

This project on **fraud detection** and **data analysis** using tools like **Excel**, **Power BI**, **Python** and **machine learning** can significantly help in securing a job in the **data science** and **analytics** fields. Here's how:

5.4 Relevance to Data Science

- **Fraud detection** is a core problem in data science, particularly in **machine learning** and **predictive analytics**. My experience with fraud detection algorithms like **Random Forest**, **Logistic Regression**, and **Gradient Boosting** demonstrates me understanding of key machine learning techniques used in real-world applications.
- By **evaluating models** and selecting the best-performing models, I am showcasing essential skills in **model evaluation** and **hyperparameter tuning**, which are fundamental in data science.

5.5 Hands-On Experience

- Employers look for **hands-on experience**. My project covers several practical aspects that will be useful in a data science role:
  - **Data preprocessing and cleaning** with **Pandas**.
  - **Building and evaluating machine learning models**.
  - **Deploying models** using **Flask** or **Streamlit**.
  - **Data visualization** with **Power BI** for creating interactive reports and dashboards.
- These are the skills that hiring managers look for when seeking candidates for data analyst and data science positions.

5.6 Future Scope

While the current implementation of the fraud detection system demonstrates high accuracy and practical usability, there are several areas where the system can be further enhanced:

- **Real-time Transaction Monitoring:** Integrating the model with real-time banking systems can help detect fraudulent transactions instantly as they occur.

- **Incorporating Deep Learning:** Techniques such as LSTM or Autoencoders can be used to capture more complex fraud patterns, especially for time-series transaction data.

- **Integration with Big Data Platforms:** The system can be scaled using platforms like Apache Spark or Hadoop to handle large-scale transaction data in real-time.

- **Continuous Learning System:** Implementing online learning or incremental training can help the model adapt to new fraud patterns over time without needing a complete retraining.

- **Use of Advanced Feature Engineering:** Behavioral biometrics, location data, device fingerprinting, and clickstream analysis can be added to enrich the input features.

- **Explainability and Interpretability:** Integration of tools like SHAP or LIME can help explain model predictions, making the system more transparent to regulators and financial analysts.

- **Cross-Platform Integration:** The model can be embedded into mobile banking applications or integrated via APIs into existing core banking systems.

- **Adaptive Fraud Detection Strategies:** As fraud strategies evolve, the model can be updated regularly using feedback loops and anomaly detection algorithms to stay robust.

  These enhancements would not only make the system more powerful and scalable but also ensure adaptability to dynamic fraud behaviors in real-world banking environments.

References

[1] A. A. S. Alsuwailem, E. Salem, and A. K. J. Saudagar, “Performance of different machine learning algorithms in detecting financial fraud,” *Computational Economics*, vol. 62, no. 4, pp. 1631–1667, 2023. [Online]. Available: https://link.springer.com/article/10.1007/s10614-022-10314-x

[2] R. Achary and C. J. Shelke, “Fraud detection in banking transactions using machine learning,” in *2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*. IEEE, 2023, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/10091067

[3] J. Perols, “Financial statement fraud detection: An analysis of statistical and machine learning algorithms,” *Auditing: A Journal of Practice & Theory*, vol. 30, no. 2, pp. 19–50, 2011. [Online]. Available: https://publications.aaahq.org/ajpt/article-abstract/30/2/19/5673

[4] A. Ali, S. Abd Razak, S. H. Othman, T. A. E. Eisa, A. Al-Dhaqm, M. Nasser, T. Elhassan, H. Elshafie, and A. Saif, “Financial fraud detection based on machine learning: a systematic literature review,” *Applied Sciences*, vol. 12, no. 19, p. 9637, 2022.

[5] R. Zhang, Y. Cheng, L. Wang, N. Sang, and J. Xu, “Efficient bank fraud detection with machine learning,” *Journal of Computational Methods in Engineering Applications*, pp. 1–10, 2023. [Online]. Available: https://ojs.sgsci.org/journals/jcmea/article/view/194

[6] O. A. Bello, A. Folorunso, O. E. Ejiofor, F. Z. Budale, K. Adebayo, and O. A. Babatunde, “Machine learning approaches for enhancing fraud

prevention in financial transactions," *International Journal of Management Technology*, vol. 10, no. 1, pp. 85–108, 2023. [Online]. Available: https://www.researchgate.net/profile/Oluwabusayo-Bello/publication/381548533

[7] D. Kalbande, P. Prabhu, A. Gharat, and T. Rajabally, "A fraud detection system using machine learning," in *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT).* IEEE, 2021, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9580102/

[8] D. Njoku, V. Iwuchukwu, J. Jibiri, C. Ikwuazom, C. Ofoegbu, and F. Nwokoma, "Machine learning approach for fraud detection system in financial institution: A web-based application," *Machine Learning*, vol. 20, no. 4, pp. 01–12, 2024. [Online]. Available: https://www.researchgate.net/profile/Vitalis-Iwuchukwu/publication/380174951

[9] E. M. Al-Dahasi, R. K. Alsheikh, F. A. Khan, and G. Jeon, "Optimizing fraud detection in financial transactions with machine learning and imbalance mitigation," *Expert Systems*, vol. 42, no. 2, p. e13682, 2025. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.13682

[1–9].