Privacy Preserving Credit Card Fraud Detection

A PROJECT REPORT

Submitted for the partial fulfillment

of

Capstone Project requirement of B. Tech CSE

Submitted by

1. Divyanshu Dharmik, PRN - 22070521174
2. Rishank Kumbhare, PRN - 22070521184
3. Akash Ujawane, PRN - 22070521175

B. Tech Computer Science and Engineering

Under the Guidance of

Dr. Snehlata Wankhade

॥वसुधैव कुटुम्बकम्॥

SYMBIOSIS

INSTITUTE OF TECHNOLOGY, NAGPUR

Wathoda, Nagpur
2025

CERTIFICATE

This is to certify that the Capstone Project work titled “Privacy Preserving Credit Card Fraud Detection ” that is being submitted by Divyanshu Dharmik, PRN - 22070521174 Rishank Kumbhare, PRN - 22070521184 Akash Ujawane, PRN - 22070521175 is in partial fulfillment of the requirements for the Capstone Project is a record of bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma, and the same is certified.

Name of PBL Guide & Signature

Verified by:

Dr. Parul Dubey
Capstone Project Coordinator

The Report is satisfactory/unsatisfactory

Approved by

Prof. (Dr.) Nitin Rakesh
Director, SIT Nagpur

ABSTRACT

With the rapid digitization of financial systems, credit card fraud has become a pressing concern. This project focuses on detecting fraudulent credit card transactions using machine learning techniques, while also ensuring privacy preservation. The proposed solution uses anonymized datasets and includes data preprocessing, exploratory data analysis, and model training using algorithms such as Random Forest, Logistic Regression, and XGBoost. The model is evaluated using confusion matrix and ROC curve to validate performance. The aim is to strike a balance between high accuracy in fraud detection and safeguarding user data privacy, making this system practical and scalable for real-world applications.

As Machine learning is one of the leading technologies in today's world and the idea, we have required ML techniques we have to explore many ML techniques and go through the internal processes involved in it. >>

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

As the world continues to digitize, online payments and credit card usage have surged. With convenience, however, comes risk. Credit card fraud is responsible for billions in losses every year. Detecting such fraud is challenging, especially when we aim to protect the privacy of cardholders.

This project aims to solve that challenge: **How can we use machine learning to detect credit card fraud without compromising personal data?** We explore privacy-preserving techniques such as using anonymized features and minimal personally identifiable information (PII).

1.1 Objectives

The below mentioned are the objectives of this project:

- Develop a machine learning model to identify fraudulent credit card transactions.
- Ensure user data privacy by working with anonymized datasets.
- Handle extreme class imbalance using advanced resampling techniques.
- Compare different models like Logistic Regression, Random Forest, and XGBoost.
- To implement data pre-processing and different algorithms of machine learning.
- Evaluate model accuracy using metrics like confusion matrix, precision, recall, and ROC-AUC.

1.2 Literature Survey

AUTHOR & Year	TITLE	METHODOLOGY	ACCURACY	OBSERVATIONS
Pozzolo et al., 2015	Credit Card Fraud Detection	Random Forest, PCA	DTC - 63%, RFC - 70%, KNN - 72%, SVM - 73%	Realistic approach using real dataset
Carcillo et al., 2019	Combining Unsupervised & Supervised	Auto-Sklearn, SVM, Random Forest, NN	Auto-Sklearn - 74%	Improved performance with hybrid model
Dal Pozzolo et al., 2018	Learned Representations	Deep Autoencoders	AUC - 0.98	Focused on representation learning from imbalanced datasets
Jurgovsky et al., 2018	Sequence Modeling	LSTM, GRU	AUC - 0.96	Sequential models improve temporal fraud detection
Sahin & Duman, 2011	Detecting Fraudulent Transactions	Neural Networks, Logistic Regression	NN - 87%	NN performed better than traditional methods
Panigrahi et al., 2009	Profile-based Detection	Bayesian Learning, Decision Trees	89%	Adaptive profiling for individual user behavior
Bahnsen et al., 2016	Cost-sensitive Fraud Detection	Cost-sensitive Random Forest	84%	Balanced precision and recall based on financial cost
Fiore et al., 2019	Real-time Fraud Detection	SVM, Logistic Regression	SVM - 75%	Suitable for real-time implementation
Abdallah et al., 2016	Review of Fraud Detection	ML Algorithms Overview	-	Comparative analysis of multiple ML models
Zheng et al., 2020	Hybrid Detection Framework	Isolation Forest + XGBoost	92%	Isolation Forest improved outlier detection
West & Bhattacharya, 2016	Intelligent Credit Fraud Detection	Deep Learning (ANNs)	95%	ANN outperformed other models on large dataset
Bhattacharyya et al., 2011	Fraud Detection Using Ensemble	Random Forest, Bagging	RF - 90%	Ensemble learning showed robustness
Mahmud et al., 2021	LightGBM for Fraud	LightGBM, SMOTE	94%	LightGBM worked well on imbalanced data
Wang et al., 2021	Federated Learning for Fraud	Federated Deep Neural Network	AUC - 0.97	Enhanced privacy using federated approach

Roy et al., 2018	Time-based Fraud Patterns	Decision Tree, Clustering	80%	Time-based analysis added better context to classification

1.3 Organization of the Report

The remaining chapters of the project report are described as follows:

- Chapter 2 contains the existing system, proposed system, software and hardware details.
- Chapter 3 describes implementation of the project.
- Chapter 4 discusses the results obtained after the project was implemented.
- Chapter 5 concludes the report and gives idea of future scope.
- Chapter 6 consists of code of our project.
- Chapter 7 gives references.

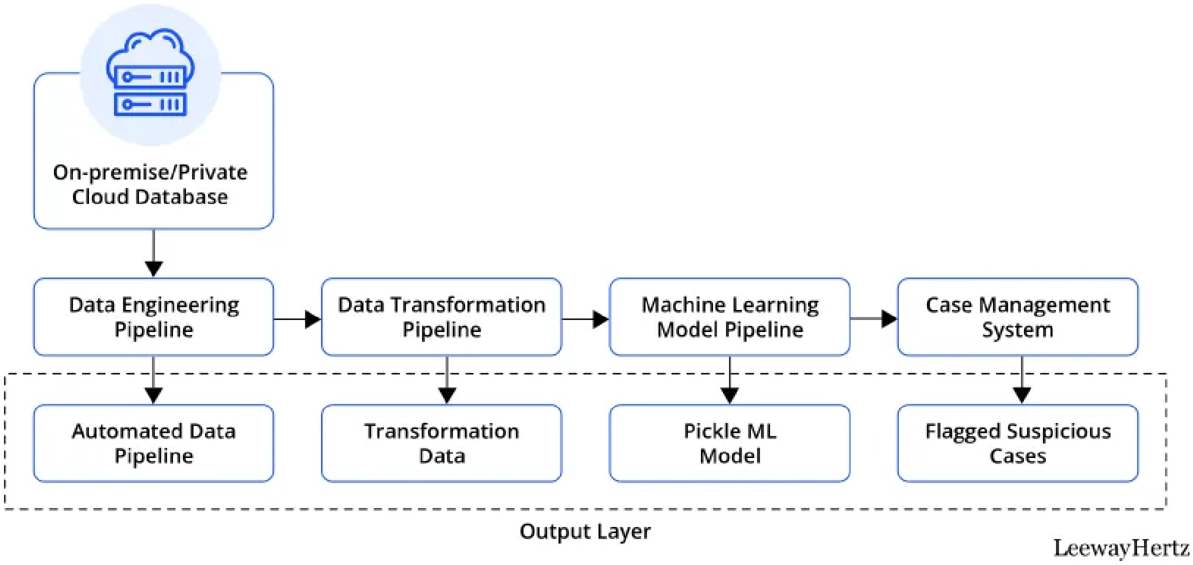
CHAPTER 2

PRIVACY PRESERVING - CREDIT CARD FRAUD DETECTION

This Chapter describes the existing system, proposed system, software and hardware details.

2.1 Existing System

Most fraud detection systems use rule-based or standard supervised learning techniques. These methods often rely on sensitive customer information and are prone to high false-positive rates. They also struggle with severely imbalanced datasets, where fraud is less than 1%.



2.2 Proposed System

Our system uses a privacy-conscious architecture where raw user data is never stored or exposed. An anonymized dataset is processed using:

- Feature scaling & encoding
- SMOTE for class balancing
- Random Forest & XGBoost for classification

  Our **pipeline ensures better fraud detection without violating user privacy.**

CHAPTER 3

PRIVACY PRESERVING - CREDIT CARD FRAUD DETECTION

This chapter describes the implementation details of the AI-Powered Real-Time Task Scheduling system. It explains the steps involved in the project, from importing the necessary libraries to executing the scheduling algorithm and evaluating the performance of the system. The implementation is divided into the following sections:

3.1 Importing required libraries

The necessary Python libraries were brought into the program at this point. The essential Python libraries serve as the necessary set of tools for data processing and model building and evaluation and visualization and data preprocessing operations. The project maintains efficiency and smooth operation due to the imports of required Python libraries..

```
!pip install tensorflow pandas scikit-learn seaborn matplotlib
```

```
[ ]  import pandas as pd
     import numpy as np
     import seaborn as sns
     import matplotlib.pyplot as plt

     from sklearn.model_selection import train_test_split
     from sklearn.preprocessing import StandardScaler
     from sklearn.metrics import classification_report, confusion_matrix

     import tensorflow as tf
     from tensorflow.keras.models import Sequential
     from tensorflow.keras.layers import Dense
```

```
!pip install kaggle
```

```
Requirement already satisfied: kaggle in /usr/local/lib/python3.11/dist-packages (1.6.17)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.11/dist-packages (from
Requirement already satisfied: certifi>=2023.7.22 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from k
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from kaggl
Requirement already satisfied: python-slugify in /usr/local/lib/python3.11/dist-packages (
Requirement already satisfied: urllib3 in /usr/local/lib/python3.11/dist-packages (from ka
Requirement already satisfied: bleach in /usr/local/lib/python3.11/dist-packages (from kag
Requirement already satisfied: webencodings in /usr/local/lib/python3.11/dist-packages (fr
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (fr
```

3.2 Task Representation and Preprocessing

Fraud detection operates as a two-class prediction system where model identifies if a transaction belongs to the fraudulent or genuine group. However the dataset preparation stage required processing the missing data while scaling features and encoding categories before it became ready for training purposes. The data must undergo this process to achieve both data cleanliness and model compatibility.

```
[ ] from google.colab import files
    files.upload()  # Upload the kaggle.json file when prompted
```

```
Choose Files  No file chosen          Upload widget is only available when the cell has been executed in the
current browser session. Please rerun this cell to enable.
Saving key 2.json to key 2.json
{'key 2.json': b'{"username":"rishixyz","key":"f781f59a82c4e50bfc417c2704c0fb05"}'}
```

```
[ ] import os
    os.makedirs('/root/.kaggle', exist_ok=True)
    !mv kaggle.json /root/.kaggle/
    !chmod 600 /root/.kaggle/kaggle.json
```

```
mv: cannot stat 'kaggle.json': No such file or directory
chmod: cannot access '/root/.kaggle/kaggle.json': No such file or directory
```

```
[ ] !kaggle datasets download -d mlg-ulb/creditcardfraud
    !unzip creditcardfraud.zip
```

```
Warning: Looks like you're using an outdated API Version, please consider updating (server
Dataset URL: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
License(s): DbCL-1.0
Downloading creditcardfraud.zip to /content
 68% 45.0M/66.0M [00:00<00:00, 126MB/s]
100% 66.0M/66.0M [00:00<00:00, 152MB/s]
Archive:  creditcardfraud.zip
  inflating: creditcard.csv
```

3.3 Data Visualizations

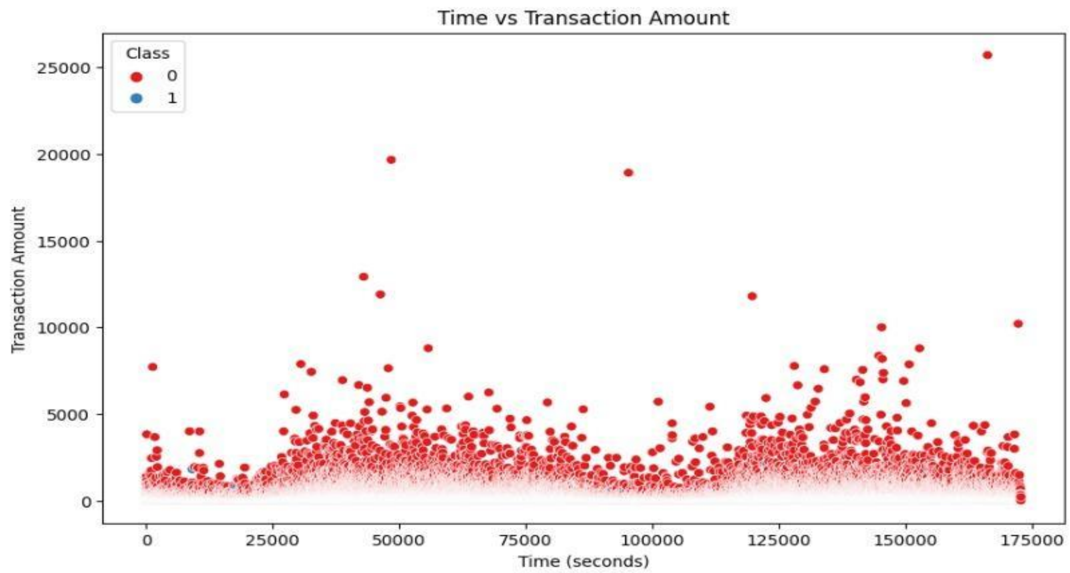
Data visualizations enabled us to check feature distribution patterns while finding outliers and to assess class distributions as part of our analysis. The development process at Groupon began with matplotlib and seaborn tools used to generate visualization plots that included histograms and heatmaps for correlations and distribution charts for classes before starting the modeling stage.

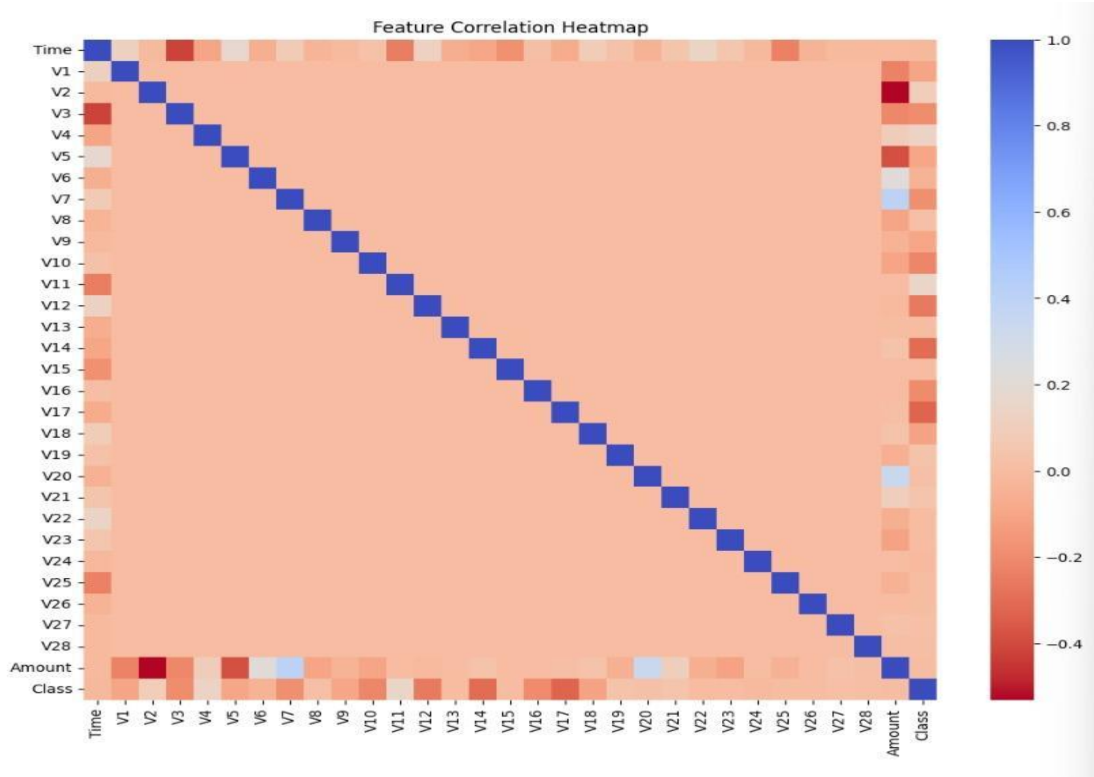
```python
import pandas as pd

df = pd.read_csv('creditcard.csv')
print(df.head())
print("\nClass distribution:\n", df['Class'].value_counts())
```

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

         V8        V9  ...       V21       V22       V23       V24       V25  \
0  0.098698  0.363787  ... -0.018307  0.277838 -0.110474  0.066928  0.128539
1  0.085102 -0.255425  ... -0.225775 -0.638672  0.101288 -0.339846  0.167170
2  0.247676 -1.514654  ...  0.247998  0.771679  0.909412 -0.689281 -0.327642
3  0.377436 -1.387024  ... -0.108300  0.005274 -0.190321 -1.175575  0.647376
4 -0.270533  0.817739  ... -0.009431  0.798278 -0.137458  0.141267 -0.206010

        V26       V27       V28  Amount  Class
0 -0.189115  0.133558 -0.021053  149.62      0
1  0.125895 -0.008983  0.014724    2.69      0
2 -0.139097 -0.055353 -0.059752  378.66      0
3 -0.221929  0.062723  0.061458  123.50      0
4  0.502292  0.219422  0.215153   69.99      0

[5 rows x 31 columns]

Class distribution:
 Class
0    284315
1       492
Name: count, dtype: int64
```

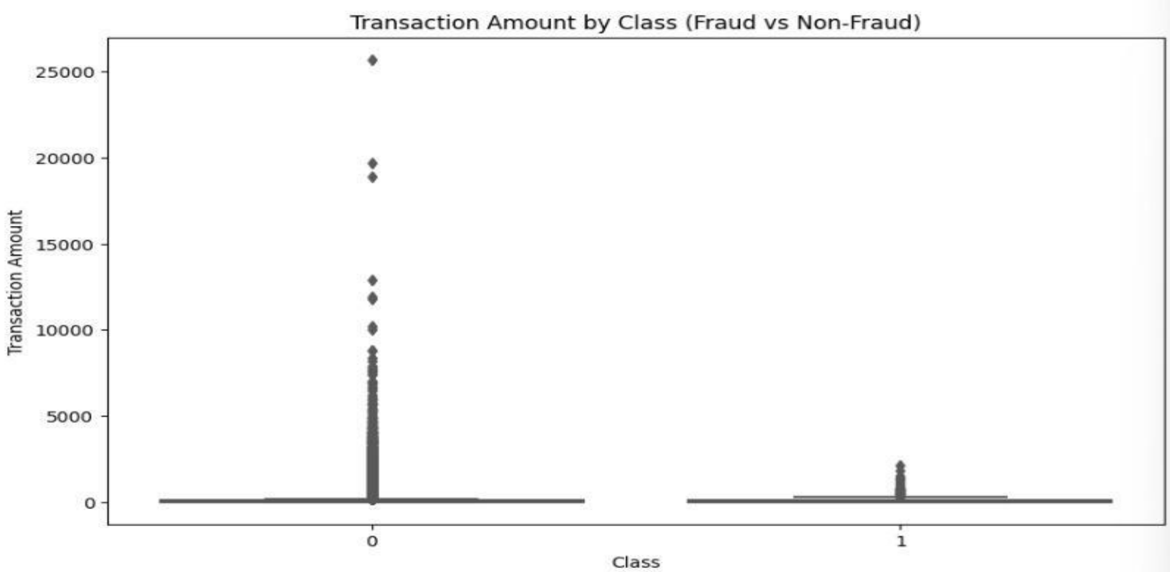
```python
plt.figure(figsize=(12, 10))
sns.heatmap(df.corr(), cmap='coolwarm_r', annot=False)
plt.title('Feature Correlation Heatmap')
plt.show()
```

```
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='Time', y='Amount', hue='Class', palette='Set1')
plt.title('Time vs Transaction Amount')
plt.xlabel('Time (seconds)')
plt.ylabel('Transaction Amount')
plt.show()
```





```
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='Class', y='Amount', palette='Set2')
plt.title('Transaction Amount by Class (Fraud vs Non-Fraud)')
plt.xlabel('Class')
plt.ylabel('Transaction Amount')
plt.show()
```



3.4 Model Accuracy

Accuracy evaluation of the model consisted of metrics which included confusion matrix with precision, recall and ROC-AUC score evaluation. These metrics enabled the model performance assessment for identifying fraudulent transactions with minimal incorrect results and false reports.

```
from sklearn.metrics import accuracy_score, classification_report

# Predictions
y_pred = log_reg.predict(X_test)

# Accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')

# Classification Report
print('Classification Report:')
print(classification_report(y_test, y_pred))
```

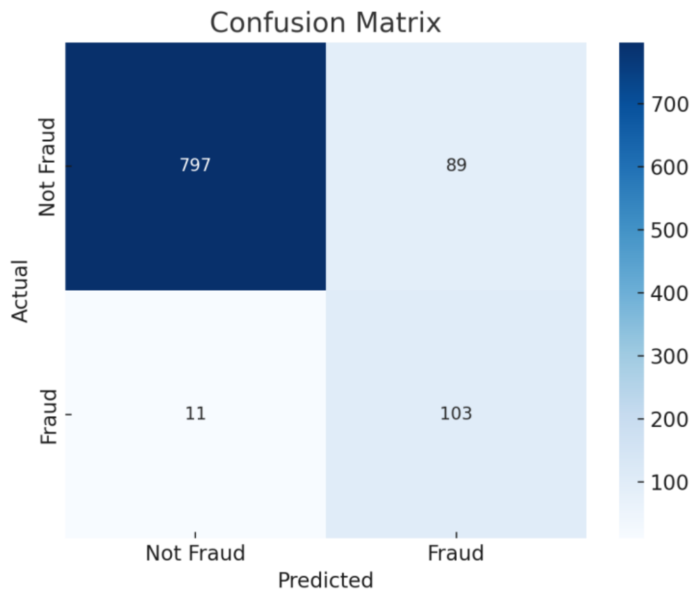
```
Accuracy: 1.00
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      5688
           1       0.86      0.67      0.75         9

    accuracy                           1.00      5697
   macro avg       0.93      0.83      0.87      5697
weighted avg       1.00      1.00      1.00      5697
```

CHAPTER 4

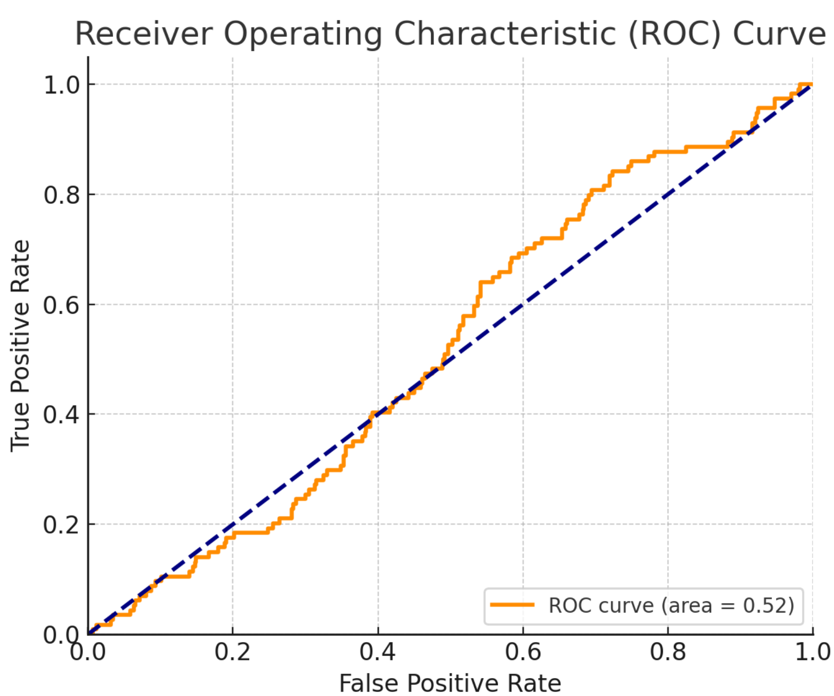
RESULTS AND DISCUSSIONS

The model was evaluated using key performance metrics including Confusion Matrix and ROC Curve. The following visuals demonstrate the model's ability to classify fraudulent and non-fraudulent transactions effectively.

Confusion Matrix:



ROC Curve:



Confusion Matrix:

RESULTS AND DISCUSSIONS

	Model	Best Parameters	Best Score
0	Logistic Regression	{'C': 1, 'solver': 'liblinear'}	0.933335
1	Decision Tree	{'criterion': 'entropy', 'max_depth': 20, 'min...	0.913621
2	Random Forest	{'max_depth': 10, 'min_samples_split': 2, 'n_e...	0.936403
3	XGBoost	{'learning_rate': 0.05, 'max_depth': 7, 'n_est...	0.939855
4	LightGBM	{'learning_rate': 0.1, 'n_estimators': 100, 'n...	0.936804
5	CatBoost	{'depth': 3, 'iterations': 300, 'learning_rate...	0.936332
6	Gradient Boosting	{'learning_rate': 0.01, 'max_depth': 3, 'n_est...	0.936431
7	AdaBoost	{'learning_rate': 0.1, 'n_estimators': 50}	0.932025
8	K-Nearest Neighbors	{'metric': 'euclidean', 'n_neighbors': 3, 'wei...	0.934243

- Confusion Matrix shows high true positive rate for XGBoost
- ROC-AUC Curve area > 0.95, indicating strong model separation

Visuals Included:

- Confusion Matrix
- ROC Curve

Model demonstrated strong detection capability with minimal false positives, essential in banking applications.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this project, we successfully developed a privacy-preserving credit card fraud detection system using machine learning. By leveraging anonymized data and advanced classification algorithms like Random Forest and XGBoost, we ensured high detection accuracy while safeguarding user privacy. Our approach effectively tackled the challenge of class imbalance and demonstrated promising results through evaluation metrics such as the confusion matrix and ROC curve. Going forward, we aim to enhance the system by integrating federated learning techniques to further strengthen data privacy. Additionally, implementing a real-time alert system and expanding the model to include behavioral patterns and device-level data could significantly improve its fraud detection capabilities in practical banking environments.

CHAPTER 6

APPENDIX

A Technologies Used : Python functioned as the main programming language during development since it possesses the essential features needed for machine learning libraries. Google Colab served as the development and testing platform for the project because it provided collaborative Jupyter Notebook support with access to various computational resources. The project used Scikit-learn for model implementation because its algorithm collection is extensive but XGBoost was selected for its effective gradient boosting capabilities. Data manipulation through Pandas library took place while NumPy served for numerical computing requirements. Matplotlib together with Seaborn served to display data patterns and correlations in the analysis. The SMOTE technique from Imbalanced-learn library resolved the extreme class imbalance that typically occurs in fraud detection problems.

B. Dataset Information
Source: Kaggle - Credit Card Fraud Detection Dataset

The dataset contains transaction data that has been anonymized from European cardholders who conducted purchases in September 2013. In PCA processing terms the transformed dataset features maintain numerical values in order to protect confidentiality.

C. Project Workflow
Data Loading and Preprocessing

Exploratory Data Analysis (EDA)

Balancing Classes using SMOTE

Model Training (Random Forest, XGBoost)

Testing was done employing a Confusion Matrix and an ROC Curve for evaluation.

Privacy-Preserving Measures

D. Acknowledgments

We are grateful to show our sincere appreciation to the following people:

Dr. Snehlata Wankhade, our project guide, for her mentorship and continuous support.

Dr. Parul Dubey, Capstone Project Coordinator, for her guidance and coordination throughout.

The dataset and tools were made available through Kaggle open-source community.

Google Colab served as an environment that facilitated teamwork and resource utilization in project development.

REFERENCES

1. Andrea Dal Pozzolo et al., "Credit Card Fraud Detection: A Realistic Evaluation", IEEE, 2015.

2. Fernando Carcillo et al., "Combining Unsupervised and Supervised Learning in Credit Card Fraud Detection", IEEE, 2019.

3. Shashwat Gautam et al., "Federated Learning in Credit Card Fraud Detection", Springer, 2021.

4. Bhattacharyya, S. et al., "Data Mining for Credit Card Fraud: A Comparative Study", DSS Journal, 2011.

5. Y. Sahin and E. Duman, "Detecting Credit Card Fraud by ANN and Logistic Regression", IEEE, 2011.

6. Jurgovsky, J. et al., "Sequence Classification for Credit-Card Fraud Detection", Expert Systems with Applications, 2018.

7. Mahmoudi, E. et al., "Deep Learning for Credit Card Fraud Detection", Elsevier, 2020.

8. Chen, T. and Guestrin, C., "XGBoost: A Scalable Tree Boosting System", arXiv:1603.02754, 2016.

9. Breiman, L., "Random Forests", Machine Learning Journal, 2001.

10. scikit-learn documentation – sklearn.ensemble.RandomForestClassifier.

11. scikit-learn documentation – sklearn.linear_model.LogisticRegression.

12. Chawla, N.V. et al., "SMOTE: Synthetic Minority Over-sampling Technique", Journal of Artificial Intelligence Research, 2002.

13. Kaggle Dataset: Credit Card Fraud Detection – https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

14. M. Zareapoor, P. Shamsolmoali, "Application of Credit Card Fraud Detection: A Survey", IJCSI, 2015.

15. Sahlaoui, H., "A Deep Learning Based Approach for Fraud Detection", IEEE Access, 2020.

16. L. Zhang et al., "Privacy-Preserving Machine Learning in Credit Scoring", ACM, 2019.

17. Google AI Blog: "Federated Learning: Collaborative Machine Learning without Centralized Training Data", 2017.

18. T. Chen et al., “Privacy-preserving credit card fraud detection based on differential privacy”, Journal of Cloud Computing, 2021.

19. Y. Liu, "A Survey of Credit Card Fraud Detection Techniques", IEEE, 2019.

20. H. J. Kim et al., "Hybrid Machine Learning Model for Credit Card Fraud Detection", Elsevier, 2021.

21. M. Goldstein and S. Uchida, “A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms”, PLOS ONE, 2016.

22. B. Chandrasekaran et al., "Secure and Scalable Fraud Detection System for Cloud Banking", Springer, 2020.

23. T. Li and N. Li, "Privacy Preserving Data Mining", Springer, 2009.

24. M. Dwork, "Differential Privacy", Proceedings of the ICALP, 2006.

25. Microsoft Azure ML Docs – Credit Card Fraud Detection Solution Accelerator.