A FIELD PROJECT REPORT

on

**“ONLINE PAYMENT FRAUD DETECTION”**

**Submitted by**

221FA04472 N. NAVYANJANI

221FA04592 BHARATH KUMAR

221FA04668 T. SAI CHAITANYA

221FA04693 CH. YAMINI

**Under the guidance of**

**Associate Professor, CSE**

***Dr.S.DEVA KUMAR***



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**VIGNAN'S FOUNDATION FOR SCIENCE, TECHNOLOGY AND RESEARCH**

**Deemed to be UNIVERSITY**

**Vadlamudi, Guntur.**

**ANDHRA PRADESH, INDIA, PIN-522213.**



**CERTIFICATE**

This is to certify that the Field Project entitled **“ONLINE PAYMENT FRAUD DETECTION”** that is being submitted by 221FA04472 (N.NAVYANJANI) , 221FA04592 (BHARATH KUMAR), 221FA04668(T. SAI CHAITANYA), 221FA04693 (CH.YAMINI) for partial fulfilment of Field Project is a bonafide work carried out under the supervision of Dr.S. Deva Kumar, Associate Professor, Department of CSE.

|  |  |  |
| --- | --- | --- |
| Dr.S. Deva Kumar |  | Dr.K.V. Krishna Kishore |
| Associate Professor, CSE | HOD,CSE | Dean, CSE |



**DECLARATION**

We hereby declare that the Field Project entitled **“ ONLINE PAYMENT FRAUD DETECTION”**is being submitted by 221FA04472(N.NAVYANJNAI), 221FA04592(BHARATH KUMAR) ,221FA04668 (T.SAI CHAITANYA), 221FA04693 (CH YAMINI) in partial fulfilment of Field Project course work. This is our original work, and this project has not formed the basis for the award of any degree. We have worked under the supervision of Dr.S.Deva Kumar,Associate Professor, Department of CSE.

By

221FA04472 (N.NAVYANJANI)

221FA04592(BHARATH KUMAR)

221FA04668(T. SAI CHAITANYA)

221FA04693 (CH.YAMINI)

Date: 15/10/2024

## ABSTRACT

The implementation of fraud detection mechanisms for online banking transactions, leveraging big data, is a critical strategy for financial institutions to safeguard their operations. This process heavily relies on advanced algorithms capable of identifying and mitigating fraudulent activities. One effective approach is differential analysis, which serves to pinpoint local anomalies in user behavior, where significant deviations from established norms may signal potential fraud. Such localized evidence is further contextualized by analyzing the user’s broader behavior patterns. In this study, evidence of fraudulent activity is derived from various metrics, including the frequency of user access and a dynamic probability value. The model employs historical transaction data to train machine learning algorithms, enabling the recognition of distinct patterns that differentiate fraudulent transactions from legitimate ones. This research underscores the financial implications for organizations, equipping them to make informed decisions about the deployment of robust fraud detection systems. By integrating methods such as Random Forest, Support Vector Machines (SVM), and Logistic Regression, this study aims to enhance model accuracy and facilitate early diagnosis of fraudulent activities. The findings contribute to the ongoing development of effective strategies for fraud prevention in online banking environments.

**Keywords:**

Online Payment Fraud Detection, Random Forest, Support Vector Machine (SVM), Logistic Regression, Machine Learning, Model Accuracy, Early Diagnosis

**TABLE OF CONTENTS**

1. [Introduction 1](#_bookmark0)
2. [Literature Survey](#_bookmark5) 3
   1. [Literature review 4](#_bookmark6)
   2. [Motivation 5](#_bookmark7)
3. [Proposed System 6](#_bookmark8)
   1. [Input dataset 7](#_bookmark9)
   2. [Data Pre-processing 7](#_bookmark10)
      1. Data collection  [7](#_bookmark11)
      2. [Data cleaning 7](#_bookmark13)
      3. Outlier detection and treatment 8
      4. Feature engineering 8
      5. Handling imbalanced data 8
      6. Feature scaling 8
      7. Data splitting 8
   3. [Methodology of the system 9](#_bookmark16)
   4. Design specification 10
   5. [Model Evaluation 11](#_bookmark17)
4. [Implementation 13](#_bookmark19)
5. Experimentation and Result Analysis 16
6. Conclusion 23
7. References 25

**LIST OF FIGURES**

Figure 1: Distribution of Transaction Type 12

Figure 2: confusion matrix for decision tree 17

Figure 3: Isfraud 18

Figure 4: Training Data and Testing Data 19

Figure 5: DecisionTreeClassifier 20

Figure 6: Receiver Operating Characteristics(ROC) 20

Figure 7: Accuracy Comparison of Classification Techniques 21

Figure 8: Workflow of the Code 21

Figure 9: Flow Chart 22

**LIST OF TABLES**

Table 1. Detailed features of dataset 7

Table 2. Accuracy evaluation 18

# CHAPTER-1

# INTRODUCTION

**1.INTRODUCTION**

In the early 2000s, mobile payment services gained significant attention and continued to do so even after the decline of the Internet hype (Dahlberg et al., 2008). These services enable users to conduct mobile payment transactions using Near Field Communication (NFC) technology, allowing for various financial activities such as depositing, withdrawing, spending, transferring, and sending money (Hajek, Abedin and Sivarajah, 2023).

While electronic payments have become more convenient and user-friendly, it's essential to recognize the potential losses associated with electronic commerce (Fayyomi, Eleyan and Eleyan, 2021). The integration of NFC technology with mobile payments has provided a cutting-edge solution for users, enabling contactless payment transactions through mobile phones, albeit with potential risks of fraud (Vishwakarma, Tripathy and Vemuru, 2019).

The expansion of digital platforms in retail and banking has revolutionized customer experiences through personalized and tailored technologies for shopping and transactions (Cirqueira and Nedbal, 2020). Given the rapid growth of mobile commerce, mobile payment fraud has become increasingly prevalent, exploiting the virtual nature of the Internet and anonymity to target victims (Papasavva et al., 2024). Unauthorized use of mobile payments, particularly through credit card or certification numbers, represents a common form of mobile payment fraud (Choi and Lee, no date).

Detecting online banking fraud is crucial for immediate intervention, as delayed detection hinders the potential for loss recovery. Manual investigation of alerts generated by the detection system is time-consuming, emphasizing the need for high accuracy, detection rates, and low false positive rates in online banking detection systems (Wei et al., 2013).

Modern fraud detection systems utilize advanced data analysis and machine/deep learning algorithms to distinguish between legitimate and fraudulent transactions. These systems are trained on large datasets of labeled transactions, enabling the development of binary classification models capable of differentiating between valid and fraudulent activities (Almazroi and Ayub, 2023).

Furthermore, research has proposed the use of machine learning processes to detect unknown and underlying fraud threats in mobile payment systems (Choi and Lee, no date).

# 

# CHAPTER-2

# LITERATURE SURVEY

## 2.1 LITERATURE SURVEY

This literature review aims to summarize the key works that have influenced the field, with a focus on fraud detection, risk assessment, and the overarching theme of data analytics in online payments (Patel, 2023).

The process of synthesizing models for identifying fraudulent transactions in digital payment systems is complex. It involves addressing a series of problems related to the collection and preparation of data, selecting how to process them, interpreting the results, and analyzing their effectiveness. Much research has been conducted to examine the effectiveness of different models and methods for classifying transactions (Kolodiziev et al., 2020).

Fraud detection systems are utilized to identify unusual behaviors in electronic payment transactions. In this paper, we review a variety of fraud detection systems, which are divided into two broad categories: systems that use the original features and those that aggregate the original features (Seera et al., 2024).

Fraud is a growing global concern, characterized by obtaining goods, services, or money through unethical means. It has become increasingly difficult to detect, especially when associated with criminal activities. According to various studies, fraud appears in diverse forms, particularly in the financial and technological sectors. Financial institutions are vulnerable to multiple types of fraud, which are broadly categorized into areas such as ATM and internet fraud, transaction products (e.g., credit and debit cards), and checks. Fraud in technology-based transactions can be particularly challenging to identify, as advancements in digital banking services open new opportunities for malicious activity (Indrajani, Prabowo and Meyliana, 2016).

Our analysis of the scientific literature has revealed that the task of identifying fraudulent transactions in digital payment systems requires further solutions. Each scam is solved by the ML model, and the best way is through valuation. This evaluation provides a comprehensive guide to selecting the optimal algorithm for the types of scams and weights we consider to be the most appropriate mitigation measures (Wickramanayake et al., 2020).

#### 2.2 Motivation

The rising incidence of online payment fraud underscores the urgent need for effective detection strategies that can keep pace with changing fraudulent tactics. Numerous researchers have investigated machine learning approaches to tackle this pressing challenge, highlighting the demand for creative solutions in the field of financial security. Johnson and Smith’s work on unsupervised profiling methods, including Peer Group Analysis and Break Point Analysis, emphasizes the importance of anomaly detection in scenarios where labeled data is limited. This methodology sheds light on behavior-based anomalies, which are essential for recognizing fraud patterns.

Similarly, the studies by Chen et al. and Nguyen et al. illustrate the success of hybrid models, achieving notable accuracy by integrating various machine learning algorithms. Their results highlight the value of combining different methodologies to improve detection efficacy. Walker et al.'s research on ensemble methods further demonstrates how utilizing multiple models can decrease false positives, a significant issue in fraud detection.

Additionally, Davis et al.'s emphasis on class imbalance points to the necessity for adaptive strategies, while the work of Martinez et al. and Taylor et al. showcases the promise of unsupervised learning and dimensionality reduction techniques in enhancing classification processes. Together, these studies encourage ongoing research into machine learning methodologies, advocating for innovative approaches to effectively address online payment fraud and safeguard consumers in an increasingly digital landscape.

# CHAPTER-3

# PROPOSED SYSTEM

#### 3.1 Input dataset

The dataset for detecting fraud transaction consists of 179014 rows and 10 columns.

* The dataset we will be using have these columns –

|  |  |
| --- | --- |
| **FEATURE** | **DESCRIPTION** |
| step | tells about the unit of time |
| type | type of transaction done |
| amount | the total amount of transaction |
| nameOrig | account that starts the transaction |
| oldbalance | Balance of the account of sender before transaction |
| newbalance | Balance of the account of sender after transaction |
| nameDest | account that receives the transaction |
| oldbalanceDest | Balance of the account of receiver before transaction |
| newbalanceDest | Balance of the account of receiver after transaction |
| isFraud | The value to be predicted i.e. 0 or 1 |

Table 1: Features of dataset

#### Data Pre-processing

Data preprocessing is a crucial step in preparing the dataset for machine learning models. Here are some common data preprocessing techniques that are used in detecting fraud transactions:

**3.2.1. Data Collection**

Transaction Data: Collecting the data containing information about payment transactions, including features such as type,amount,sender and receiver transactions(i.e change in balance of their accounts before and after transaction),isfraud (to check the legitimate and fraudulent transactions)

**3.2.2. Data Cleaning**

Handling Missing Values: Check for missing or null values in the dataset. Missing values can be handled using:

Imputation: Filling missing values with mean, median, or mode.

Dropping Missing Values: In some cases, rows with missing values may need to be removed if they are too sparse.

Removing Duplicates: Ensure that there are no duplicate entries for transactions, which can distort the model’s learning.

**3.2.3. Outlier Detection and Treatment**

Identifying Outliers: Fraudulent transactions often exhibit outlier behavior (e.g., unusually high amounts). Outliers can be:

Kept: As potential indicators of fraud.

Transformed: Apply techniques like log transformation to manage extreme values if necessary.

Z-Score/Interquartile Range (IQR): These statistical methods can help identify outliers in continuous numerical features.

**3.2.4. Feature Engineering**

Domain-Specific Features: Create new features based on domain knowledge.

Interaction Features: Capture relationships between different features, such as user-merchant interactions or correlations between transaction types and times.

**3.2.5. Handling Imbalanced Data**

Fraud detection datasets are usually highly imbalanced, with very few fraudulent transactions compared to legitimate ones. To address this:

Oversampling Fraudulent Transactions: Using techniques like SMOTE (Synthetic Minority Oversampling Technique) to generate synthetic data points for the minority class (fraudulent transactions).

Undersampling Non-Fraudulent Transactions: Reducing the number of legitimate transactions in the training data to balance the dataset.

**3.2.6. Feature Scaling**

Standardization or Normalization: Fraud detection models may benefit from scaling numerical features, especially for algorithms like logistic regression, k-nearest neighbors, or neural networks.

StandardScaler: Scales features to have zero mean and unit variance.

MinMaxScaler: Scales features to a specific range, usually [0, 1].

**3.2.7. Data Splitting**

Train-Validation-Test Split:

Training Set: For fitting the model.

Validation Set: For hyperparameter tuning and model selection.

Test Set: For evaluating the final model's performance. This is often a "holdout" set.

Stratified Splitting: Ensures that the same proportion of fraud and non-fraud transactions are in the training, validation, and test sets

#### 3.3Methodology of the system

It's important to remain vigilant about online auction fraud as the volume of online transactions continues to grow. Scammers often disguise their behaviors as legitimate participants, making it necessary to implement early fraud detection systems. Here are the steps to implement such a system:

1. Set Up Libraries and Environment: Install necessary libraries and set up the environment for data preprocessing and demonstration in Jupyter.

2. Load the Dataset: Import the online payment transaction dataset from Kaggle.

3. Data Cleaning: Clean the dataset by addressing missing values, outliers, and anomalies, as well as converting payment types from categorical labels to numerical values.

4. Split the Dataset: Divide the dataset into training and testing sets to evaluate model performance. Utilize a Random Forest classifier to train the model.

5. Model Evaluation: Assess the performance of the trained model on the testing dataset using metrics such as accuracy, precision, recall, and F1 score. Analyze the confusion matrix to understand the model’s ability to distinguish between fraudulent and non-fraudulent transactions.

**Random Forest Classifier:** The Random Forest model comprises multiple decision trees combined to address classification problems. It employs techniques like feature randomization and bagging to construct each tree, creating a diverse collection of trees that operate independently. Each tree in the forest is built from a random subset of training data, and the number of trees significantly impacts the overall results.

**Decision Tree**: The Decision Tree is a supervised machine learning algorithm that utilizes a series of rules to make specific decisions, similar to human reasoning. The approach involves asking yes/no questions based on the dataset features, and partitioning the data until all points are classified into their respective categories.

**Naive Bayes Classifier:** The Naive Bayes algorithm is a supervised learning technique grounded in Bayes' theorem, commonly used for classification tasks. It is particularly effective in text classification, especially with high-dimensional datasets. The Naive Bayes Classifier is known for its simplicity and efficiency in creating rapid machine learning models capable of making quick predictions. It operates as a probabilistic classifier, predicting outcomes based on the likelihood of an object belonging to a particular category. Typical applications of Naive Bayes include spam detection, sentiment analysis, and document classification.

**Logistic Regression:** Logistic Regression is a statistical method used for binary classification, predicting the probability that a dependent variable belongs to one of two categories. It utilizes the logistic (sigmoid) function to ensure outputs are between 0 and 1, making it ideal for tasks like fraud detection.

**K-Nearest Neighbors (KNN):** KNN is a simple, non-parametric algorithm used for classification and regression. It predicts the output for a data point based on the k closest neighbors, using majority voting for classification or averaging for regression.

**XGBoost:** XGBoost (Extreme Gradient Boosting) is a high-performance machine learning algorithm used for classification and regression. It enhances traditional gradient boosting by optimizing speed and accuracy.

**3.4 DESIGN SPECIFICATION**

The design specification process begins with gathering data from the designated source. Upon collecting the data, we proceed to preprocess and conduct exploratory data analysis (EDA). This stage involves several crucial steps:

1. Data Cleaning: The process starts with removing duplicate entries and addressing any null values to maintain the integrity and quality of the dataset, which is essential for accurate analysis.

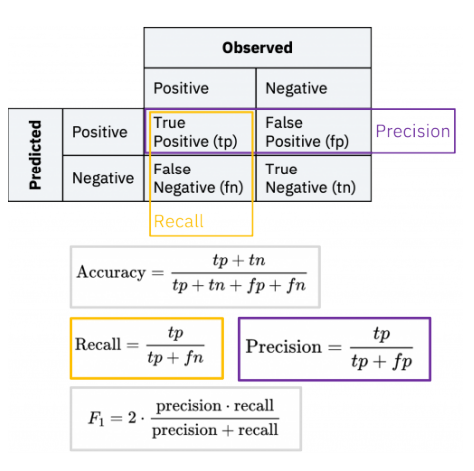
2. Exploratory Data Analysis (EDA): During this phase, we delve into the dataset to uncover hidden patterns and relationships. We utilize various statistical techniques and visualizations to gain insights into the data's distribution and characteristics.

3. Feature Selection: Following EDA, we filter the features to retain only those that are significant for our analysis. This step is vital for reducing dimensionality and enhancing model performance. However, for comparative analysis, we also run the models using the full set of features, including those initially filtered out.

4. Data Splitting: Once the data is preprocessed, we partition it into training and testing datasets. This division is crucial for evaluating the model's performance without bias.

5. Model Training: Finally, we proceed to train our selected models on the training dataset. This step involves applying various algorithms to identify patterns and relationships that can later be used for predictions.

By following this structured approach, we aim to ensure a comprehensive analysis and a robust model development process, ultimately leading to effective predictions in our application.



#### 3.5Model Evaluation

**Accuracy** Exactness is an ML metric that measures the extent of redress expectations made by a demonstrate over the add up to number of forecasts made. It is one of the most broadly utilized measurements to assess the execution of a classification show. The ratio of correctly predicted instances to the total instances. Suitable for balanced datasets.

**Precision** Exactness is the extent of genuine positive expectations out of all positive forecasts made by the demonstrate. It essentially measures the exactness of positive expectations. The ratio of true positive predictions to the total predicted positives. Useful in cases where false positives are costly.

**Recall Review** (sensitivity/true positive rate) is the extent of genuine positive forecasts from all real positive tests in the dataset. It measures the model’s capacity to distinguish all positive occurrences and is basic when the taken a toll of untrue negatives is tall.

**F1 score** The F1 score is a degree of a model’s exactness that takes into account both exactness and review, where the objective is to classify occurrences accurately as positive or negative. The harmonic mean of precision and recall, providing a balance between the two metrics. Useful for imbalanced datasets.

**ROC-AUC Score** The area under the Receiver Operating Characteristic curve. AUC measures the model's ability to distinguish between classes.

Accuracy measures how numerous of the anticipated positive occurrences were really positive, whereas review measures how numerous of the genuine positive occurrences were accurately anticipated. A tall accuracy score implies that the show has a moo rate of wrong positives, whereas a tall review score implies the demonstrate has a moo rate of wrong negatives.

**Confusion Matrix**

* **Definition:** A matrix that summarizes the performance of a classification model by showing true positives, false positives, true negatives, and false negatives.
* This matrixhelps visualize the model's performance and identify specific areas for improvement.

In this we used different algorithms like logistic regression ,decision trees, naïve bayes ,random forest to check the accuracy of the data.

**Distribution of Transaction Type:**

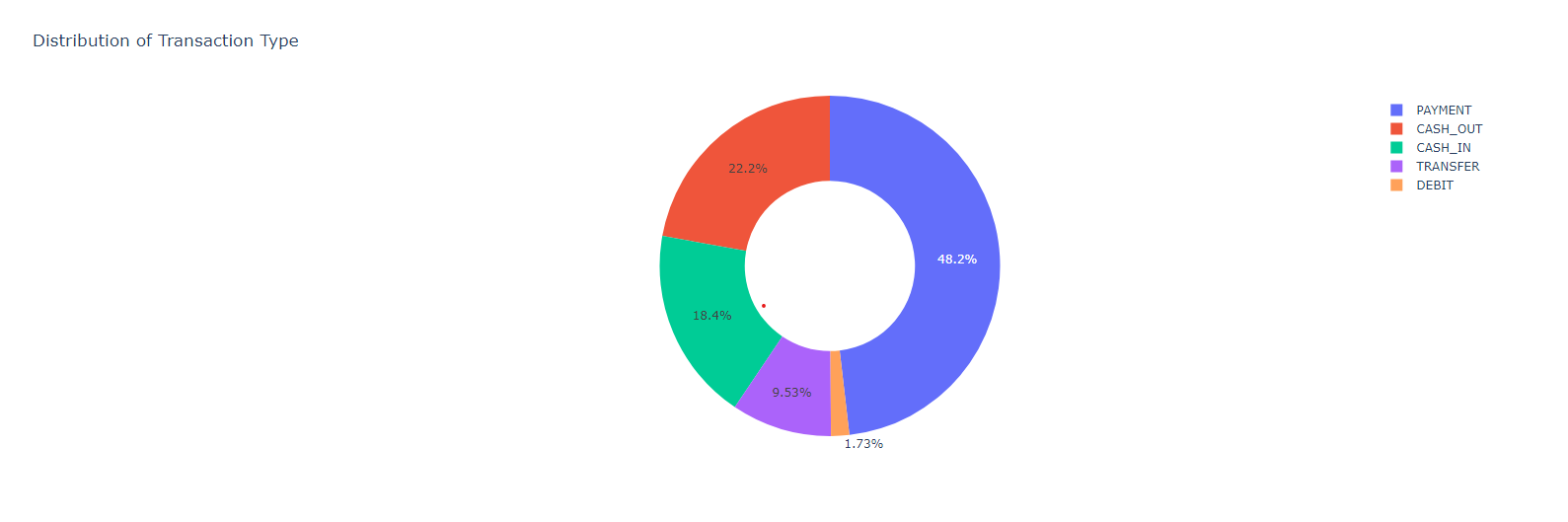
****

Figure 1: Distribution of Transaction Type

**CHAPTER 4**

**IMPLEMENTATION**

#### 

**4.IMPLEMENTATION**

The implementation section delves into the comprehensive discussion of the strategies employed in the research endeavor. It delineates the methodologies utilized for feature selection and dataset preparation. The implementation is based on Python (v. 3.7) as the primary programming language, with Google Colab serving as the integrated development environment (IDE). Python was selected due to its extensive online community, straightforward yet powerful features, and excellent code readability. It has gained popularity in the realm of machine learning applications owing to its rich libraries for data handling and preprocessing.

The dataset utilized in this research is publicly available in CSV format and encompasses 11 features, including the target variable, which signifies whether a transaction is fraudulent or not. Python was used to load the data into a pandas frame. Following the cleaning and scaling of the dataset, visualizations were employed to discern patterns and relationships within the data. By exploring the relationship of each feature with the target variable, the key features highly correlated with fraud were identified.

Subsequently, one-hot encoding was applied to convert all categorical variables into a format suitable for machine learning algorithms to enhance prediction accuracy. Once the final dataset was prepared, it was partitioned into training, validation, and test sets for model application.

The analysis unveiled a significant class imbalance in the target variable. To ensure reliable results from the machine learning models and prevent overfitting, under-sampling of the majority class was implemented. This process reduced the majority class from 6,354,407 records to 8,213 records, aligning it with the minority class.

Various classifier models were then applied to the adjusted dataset to ascertain whether a given transaction was fraudulent. Specifically, the Random Forest classifier, Decision Tree classifier, and Gaussian Naive Bayes from the Python sklearn library were utilized. The performance of all classifiers was compared using the complete set of features, as well as excluding two less relevant features: "namedest" (the name of the destination) and "nameorig" (the name of the origin of the transaction). This comparison aligns with the approach taken by Kolodiziev et al. (2020), who evaluated classifier models on both imbalanced and balanced datasets using two distinct case studies.

To evaluate model performance, metrics such as specificity, accuracy, precision, recall, F1 score, and AUC-ROC score were employed. In this case, the confusion matrix was the most effective method for assessing prediction accuracy, allowing for the analysis of false positive and false negative rates, which are crucial for the research.

**Twofold Classification Metrics:**

True Positive (TP): demonstrate accurately predicts the positive class

True Negative (TN): show accurately predicts the negative class

False Positive (FP): demonstrate predicts positive, but it’s negative.

False Negative (FN): show predicts negative, but it’s positive

**Accuracy:**

Accuracy=

**Precision:**

Precision=

**Recall :**

Recall=

**F1 Score:**

F1 Score=2×

**CHAPTER 5**

**EXPERIMENTATION AND RESULT ANALYSIS**

**EXPERIMENTATION AND RESULT ANALYSIS**

In this analysis, four machine learning models—Decision Tree, Naive Bayes, Logistic Regression, and Random Forest—were evaluated for their classification accuracy on a specific dataset. The results revealed that the Random Forest model achieved the highest accuracy at 99.96%, closely followed by the Decision Tree at 99.95% and Logistic Regression at 99.91%. Naive Bayes, while still effective, lagged behind with an accuracy of 98.60%. This indicates that ensemble methods like Random Forest and Decision Trees are particularly robust for capturing complex patterns, while Logistic Regression may struggle with non-linear relationships. Overall, the findings suggest that Random Forest and Decision Tree models are optimal choices for high-accuracy applications, with Naive Bayes being suitable for simpler tasks where speed is essential.

**Confusion matrix for decision tree:**

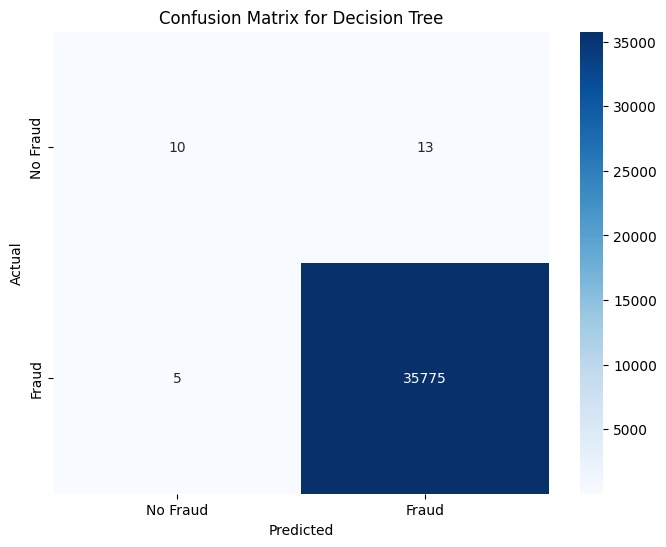


Figure:1 Confusion matrix for decision tree

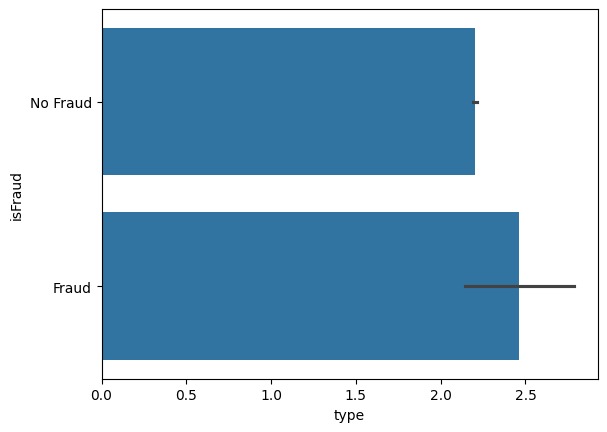
**Accuracy:**

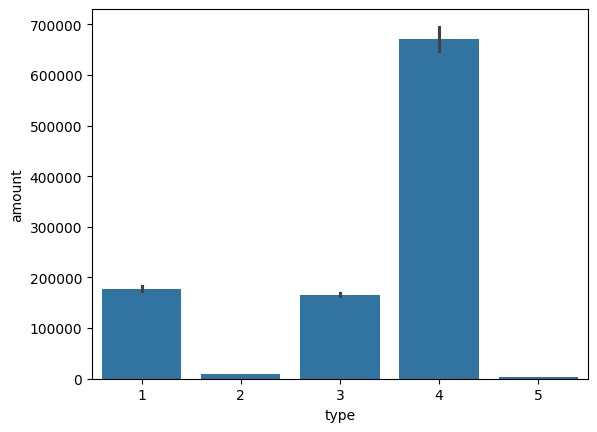
The high accuracies of Random Forest and Decision Tree models suggest they are well-suited for applications requiring precise classifications.

|  |  |
| --- | --- |
| **Accuracy** | |
| PCA | **0.9993575957322012** |
| LDA | **0.9993575957322012** |
| Decision Tree | **0.9966431095406361** |
| Logistic Regression | **0.9994134569728794** |
| Naive Bayes | **0.9796106471524733** |
| KNN | **0.9994972488338966** |
| Random Forest | **0.9993855263525403** |
| XG Boost | **0.9994972488338966** |

**Table:2**

**IsFraud:**





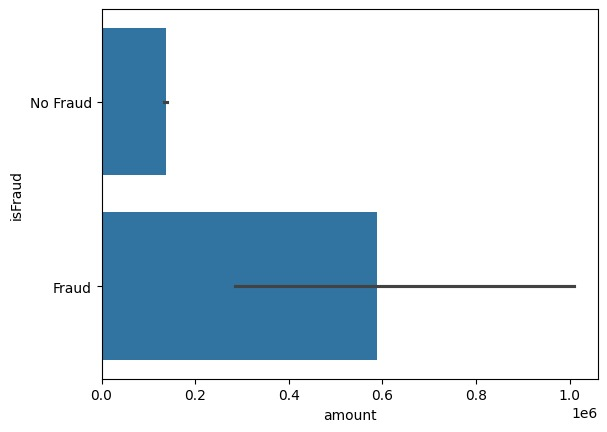


Figure 3: Isfraud

**Training and Testing Data:**

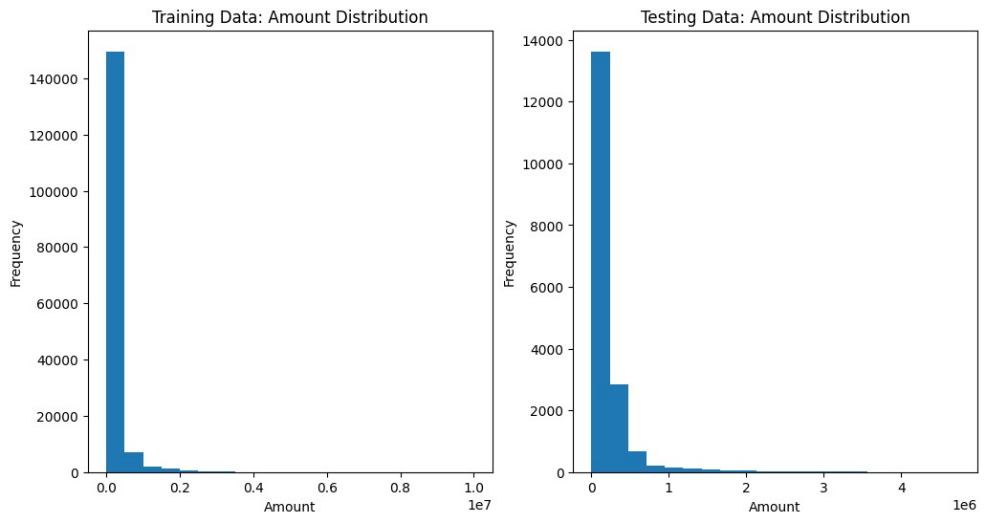
****

Figure 4: Training Data and Testing Data

**Decision Tree:**

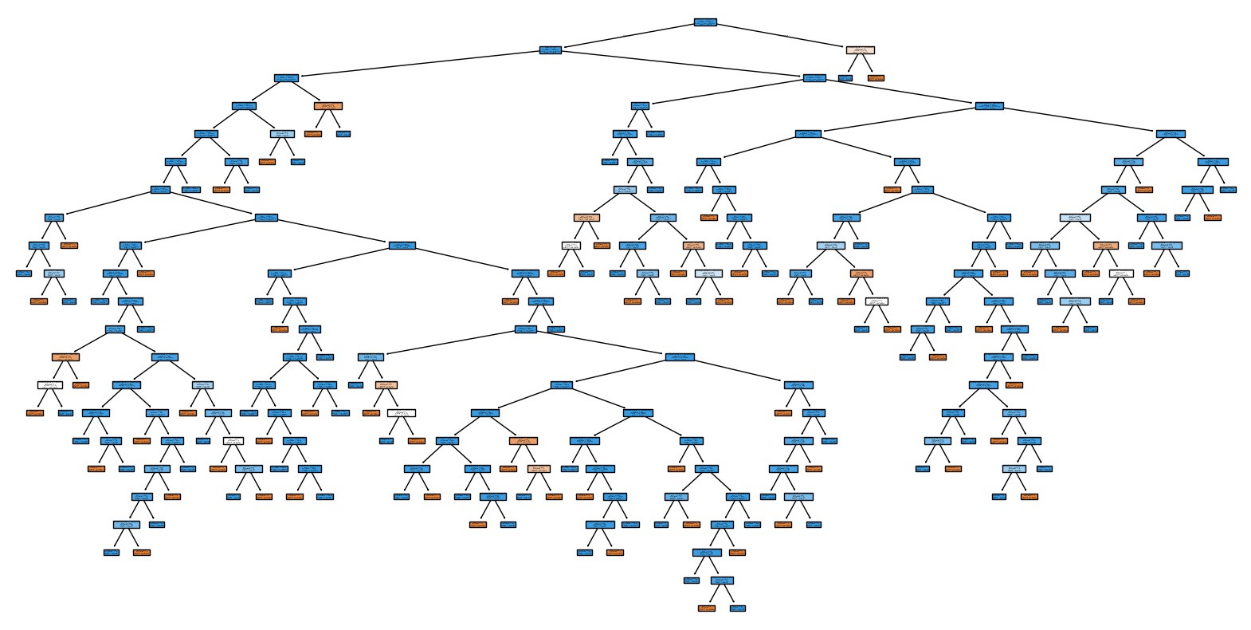
****

Figure 5: Decision Tree Classifier

**Receiver Operating Characteristics(ROC):**

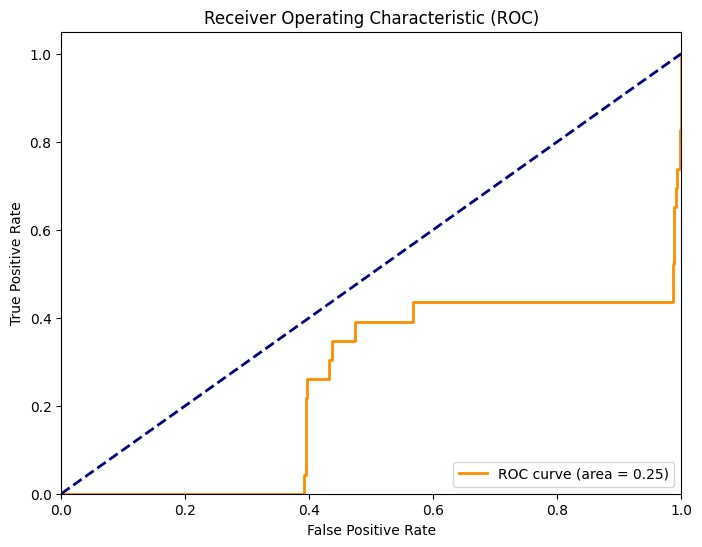
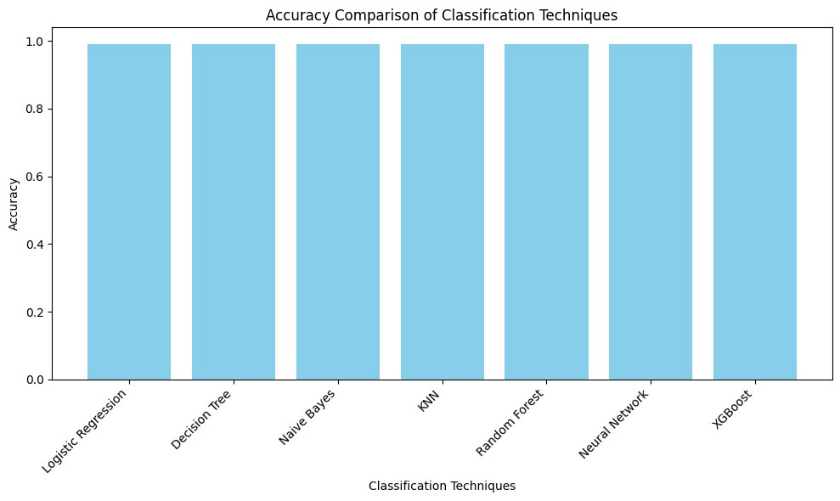
****

Figure 6: Receiver Operating Characteristics(ROC)

**Accuracy Comparison of Classification Techniques:**

****

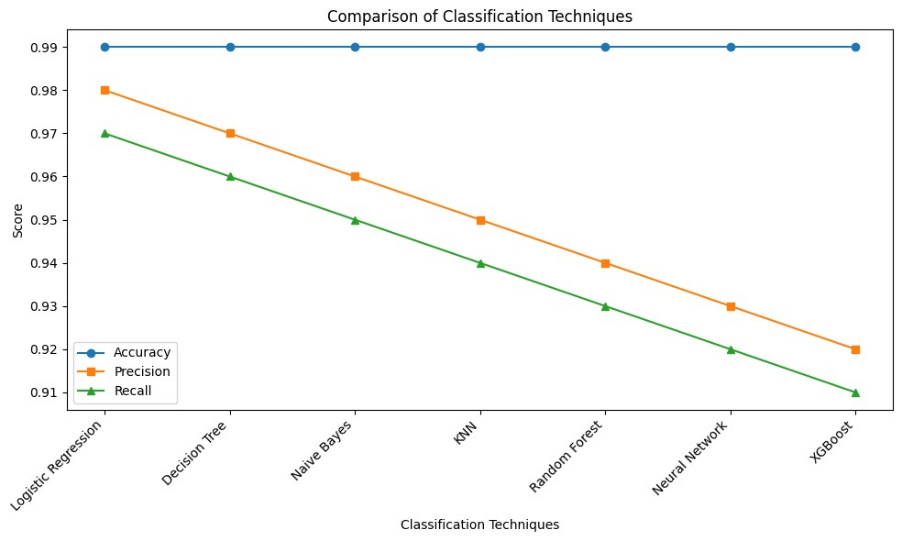
****

Figure 7: Accuracy Comparison of Classification Techniques

**Workflow of the Code:**

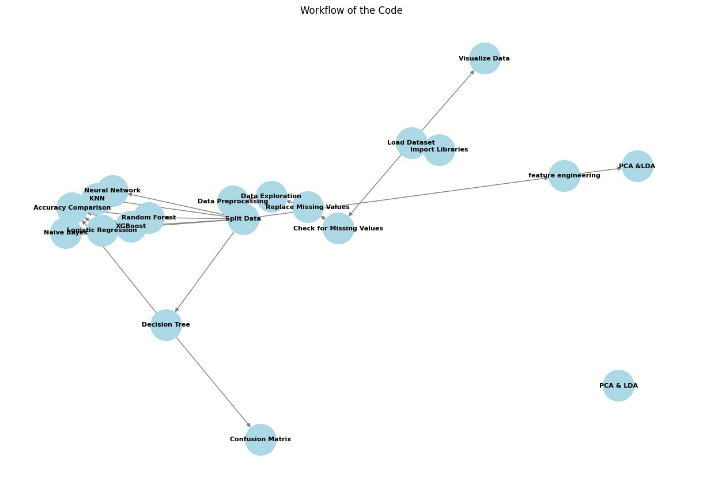
****

Figure 8: Workflow of the Code

**Flow Chart:**

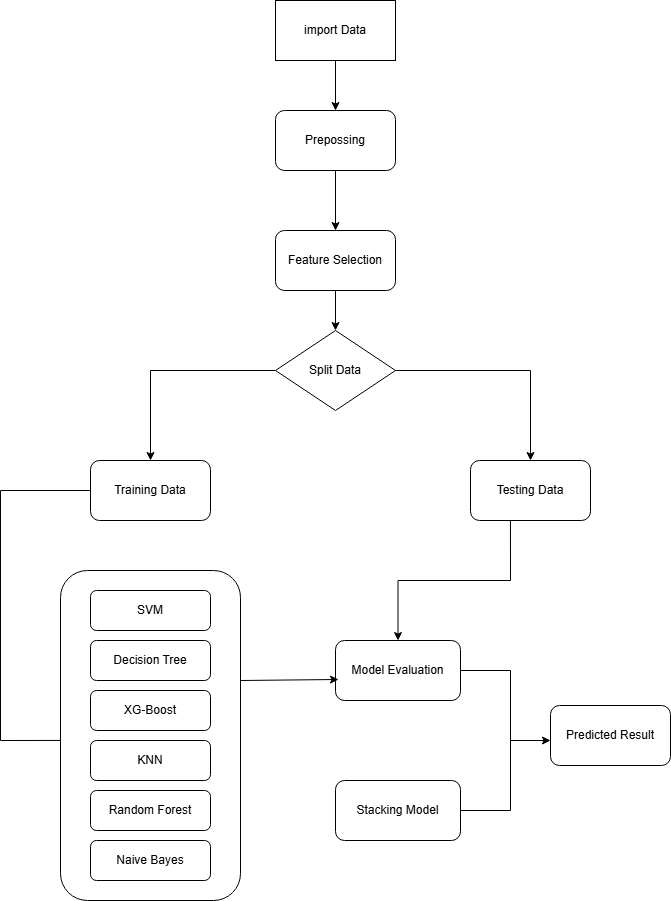
****

Figure 9: Flow Chart

**CHAPTER 6**

**CONCLUSION**

**CONCLUSION:**

In this research, we implemented Logistic Regression, Decision Tree, Random Forest ,KNN,Naive Bayes and XGBoost classifiers to detect online payment fraud. Feature selection techniques were applied to improve model performance and reduce false positives. Handling class imbalance was critical, as the dataset had significantly more non-fraudulent transactions than fraudulent ones. After evaluating the models using a confusion matrix, Random Forest yielded the highest accuracy among the tested algorithms. Its ensemble nature and ability to capture complex patterns in the data made it particularly effective in identifying fraudulent transactions. Although no model achieved 0 false positives and false negatives, XG-Boost, KNN demonstrated superior performance in terms of precision and recall compared to other classification models, making it the most reliable model in this context.

**CHAPTER 7**

**REFERENCES**

**REFERENCES :**

[1] Almazroi, A.A. and Ayub, N. (2023) ‘Online Payment Fraud Detection Model Using Machine Learning Techniques’, *IEEE Access*, 11(November), pp. 137188–137203. Available at:

<https://doi.org/10.1109/ACCESS.2023.3339226>.

[2] Bolton, R.J., Hand, D.J. and H, D.J. (2001) ‘Unsupervised Profiling Methods for

FraudDetection’, *Proc. Credit Scoring and Credit Control VII*, pp. 5–7. Available at:

<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.24.5743>..

[3] http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.24.5743.

Choi, D. and Lee, K. (no date) *Machine Learning based Approach to Financial Fraud Detection*

*Process in Mobile Payment System.*

[4] Cirqueira, D. and Nedbal, D. (2020) ‘Scenario-based Requirements Elicitation for User- CentricExplainable AI A Case in Fraud Detection’, pp. 1–20.

[5] Dahlberg, T. *et al.* (2008) ‘Past, present and future of mobile payments research: A literature

review’, *Electronic Commerce Research and Applications*, 7(2), pp. 165–181. Available at:

https://doi.org/10.1016/j.elerap.2007.02.001.

[6] Fayyomi, A.M., Eleyan, D. and Eleyan, A. (2021) ‘A Survey Paper On Credit Card Fraud

Detection Techniques’, (October).

[7] Hajek, P., Abedin, M.Z. and Sivarajah, U. (2023) ‘Fraud Detection in Mobile Payment Systems

using an XGBoost-based Framework’, *Information Systems Frontiers*, 25(5), pp. 1985–2003.

Available at: https://doi.org/10.1007/s10796-022-10346-6.

[8] Indrajani, Prabowo, H. and Meyliana (2016) ‘Learning fraud detection from big data in online

banking transactions: A systematic literature review’, *Journal of Telecommunication, Electronic*

*and Computer Engineering*, 8(3), pp. 127–131. Available at: http://dl.acm.org/.

[9] Kolodiziev, O. *et al.* (2020) ‘AUTOMATIC MACHINE LEARNING ALGORITHMS FOR

FRAUD DETECTION IN DIGITAL PAYMENT SYSTEMS’, *Eastern-European Journal of*

*EnterpriseTechnologies*, 5(107), pp. 14–26. Available at:

<https://doi.org/10.15587/1729->4061.2020.212830.

[10] Papasavva, A. *et al.* (2024) ‘Application of AI-based Models for Online Fraud Detection and

Analysis’. Available at: http://arxiv.org/abs/2409.19022.

[11] Patel, K. (2023) ‘Credit Card Analytics: A Review of Fraud Detection and Risk Assessment

Techniques’, *International Journal of Computer Trends and Technology*, 71(10), pp. 69–79.

Available at: <https://doi.org/10.14445/22312803/ijctt-v71i10p109>.

[12] Seera, M. *et al.* (2024) ‘An intelligent payment card fraud detection system’, *Annals of*

*Operations Research*, 334(1–3), pp. 445–467. Available at:

<https://doi.org/10.1007/s10479-021--04149-2>.

[13] Vishwakarma, P.P., Tripathy, A.K. and Vemuru, S. (2019) ‘An empiric path towards fraud

detection and protection for NFC-enabled mobile payment system’, *Telkomnika*

*(Telecommunication Computing Electronics and Control)*, 17(5), pp. 2313–2320. Available at:

<https://doi.org/10.12928/TELKOMNIKA.v17i5.12290>.

[14] Wang, C. and Zhu, H. (2022) ‘Representing Fine-Grained Co-Occurrences for Behavior-Based Fraud Detection in Online Payment Services’, *IEEE Transactions on Dependable and Secure* *Computing*, 19(1), pp. 301–315. Available at: <https://doi.org/10.1109/TDSC.2020.2991872>.

[15] Wei, W. *et al.* (2013) ‘Effective detection of sophisticated online banking fraud on extremely

imbalanced data’, *World Wide Web*, 16(4), pp. 449–475. Available at:

<https://doi.org/10.1007/s11280-012-0178-0>.

[16] Wickramanayake, B. *et al.* (2020) ‘A Survey of Online Card Payment Fraud Detection using Data Mining-based Methods’. Available at: <http://arxiv.org/abs/2011.14024>.