**Credit Card Fraud Detection**

### Submitted By

| **Name** | **Student ID** |
| --- | --- |
| Muid Hasan Maruf | 0242220005101237 |

**MINI LAB PROJECT REPORT**

This Report Presented in Partial Fulfillment of the course **CSE-222. Object Oriented Programming II Lab**



### DAFFODIL INTERNATIONAL UNIVERSITY

**Dhaka, Bangladesh**

##### December 9, 2024

## DECLARATION

We hereby declare that this lab project has been done by us under the supervision of Mr. Abdul Muntakim**,** Lecturer**,** Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere as lab projects.

##### Submitted To:



**Mr. Abdul Muntakim**

Lecturer

Department of Computer Science and Engineering Daffodil International University

##### Submitted by

| Muid Hasan Maruf  Section: 63\_A2  ID: 0242220005101237 | |
| --- | --- |

# Table of Contents

**Declaration** [**I**](#_heading=h.30j0zll)

1. **Introduction** [**1**](#_heading=h.2et92p0)
   1. Introduction [1](#_heading=h.3dy6vkm)
   2. Motivation [2](#_heading=h.4d34og8)
   3. Objectives [2](#_heading=h.17dp8vu)
   4. Feasibility Study [2](#_heading=h.26in1rg)
   5. Gap Analysis [4](#_heading=h.35nkun2)
   6. Project Outcome [4](#_heading=h.44sinio)
2. **Proposed Methodology/Architecture** [**5**](#_heading=h.z337ya)
   1. Requirement Analysis & Design Specification [5](#_heading=h.1y810tw)
      1. Overview [5](#_heading=h.2xcytpi)
      2. Proposed Methodology/ System Design [6](#_heading=h.3whwml4)
      3. UI Design [7](#_heading=h.qsh70q)
   2. Overall Project Plan [7](#_heading=h.1pxezwc)
3. **Implementation and Results** [**8**](#_heading=h.2p2csry)
   1. Implementation [8](#_heading=h.3o7alnk)
   2. Performance Analysis [9](#_heading=h.ihv636)
      1. Machine learning algorithms [9](#_heading=h.1hmsyys)
   3. Results and Discussion [11](#_heading=h.2grqrue)-18
4. **Engineering Standards and Mapping** [**19**](#_heading=h.3fwokq0)
   1. Impact on Society, Environment and Sustainability [19](#_heading=h.4f1mdlm)
      1. Impact on Life [19](#_heading=h.19c6y18)
      2. Impact on Society & Environment [20](#_heading=h.28h4qwu)
      3. Ethical Aspects [20](#_heading=h.37m2jsg)
      4. Sustainability Plan [20](#_heading=h.46r0co2)
5. **Conclusion 21**
   1. Summary 21
   2. Limitation 21
   3. Future Work 22

**References 23**

**Chapter 1**

# Introduction

chapter-1 will provide an overview of credit card fraud detection, the challenges in detecting fraud, and the objective of developing a machine learning model to improve accuracy and real-time fraud detection, while discussing the project’s significance and scope.

### Introduction

Financial fraudulent activity is an ever-growing threat in this modern era, where business, industries, organizations and government are highly dependable on internet technology. And with rapidly developing economy globalization, credit card transactions became popular for commercial transactions in both online and offline. According to the Global Payment report 2020, 24.2 percent in eCommerce and 20.9 percent in Point of Sale (POS) payments were made by credit card in 2019. Thus, the huge transactional services are often targeted by cyber criminals to perform fraudulent activity using credit card services. [[1]](#_heading=h.sqyw64) The unauthorized usages of cards and unusual transactions are defined as credit card fraud. Namely there are three three types of credit card frauds in general. Conventional frauds, which are performed by stilling, faking or counterfeiting. Online frauds are often committed by the fake or false merchant sites. And Merchant related frauds involved with merchant collusion and triangulation. [[2]](#_heading=h.3cqmetx) By using traditional methods of manual detection can be time consuming, inefficient and costly for detecting the fraudulent transactions. Credit card fraud detection is the process of identifying transactions whether it’s legitimate (genuine) or fraudulent. To counter cyber-criminal activities, data mining and machine learning approaches are vastly used. Therefore, these techniques can be applied for credit card fraud detection methods. This project focuses on detecting fraudulent transactions in credit card data using machine learning. Fraud detection is essential to safeguard financial institutions and their clients, given the rise in cybercrime and financial fraud. The project uses a Kaggle dataset of anonymized credit card transactions.

Chapter 1. Introduction 1.2. Motivation

### Motivation

The increasing frequency of fraudulent activities in the digital economy poses significant challenges to financial security. Building accurate fraud detection systems is crucial for minimizing losses and improving transaction security.

1. Detect Financial Fraud:
   * Prevent financial losses and protect businesses and customers from fraud.
2. Enhance Security:
   * Enable real-time detection to block fraudulent transactions before they are completed.
3. Improve Customer Trust:
   * Safeguard customers’ personal and financial data, maintaining trust and confidence.
4. Utilize Machine Learning:
   * Apply machine learning techniques to analyze patterns and detect fraud effectively in large datasets.
5. Feature Engineering:
   * Leverage domain-specific knowledge (e.g., transaction hour, scaled amounts) to improve model performance.

### Objectives

* + - Find an appropriate agent architecture for the fraud detection system based on accuracy.
    - Develop and incorporate a framework in the available database to identify fraudulent transactions where the number of cases of fraud is very limited compared to legal cases.
    - Determine which algorithm gives a better result based on accuracy, Recall and Precision
    - Build machine learning models to identify fraudulent transactions.
    - Address challenges such as class imbalance and high dimensionality.
    - Compare model performance using evaluation metrics like accuracy and precision.

### Feasibility Study

With the rapid increase in digital transactions, credit card fraud has become a critical issue, necessitating the development of sophisticated detection systems. This paper reviews various machine learning (ML) and deep learning (DL) techniques applied to credit card fraud detection, including Logistic Regression, Support Vector Machine (SVM), Nave Bayes (NB), Decision Trees (DT), and Random Forest (RF), which are commonly used due to their simplicity and effectiveness in binary classification tasks [[3][4].](#_heading=h.4bvk7pj) Ensemble methods,

Chapter 1. Introduction 1.4. Feasibility Study

particularly RF, enhance accuracy by combining multiple classifiers and are often paired with data balancing techniques such as the Synthetic Minority Oversampling Technique (SMOTE) and Adaptive Synthetic Sampling (ADASYN) to address the class imbalance problem inherent in fraud detection datasets [[5][6].](#_heading=h.1664s55) Deep learning models such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) networks, and Autoencoders are increasingly explored for their ability to identify complex and evolving fraud patterns within high-dimensional transaction data [[7][8].](#_heading=h.25b2l0r) Recent studies also explore hybrid models that integrate ML and DL approaches, along with probabilistic and rule-based filtering techniques, to dynamically assess transaction risk and adapt to emerging fraud behaviors [[7][9].](#_heading=h.kgcv8k) These adaptive mechanisms improve real-time detection and enable fraud detection systems to respond to evolving fraud patterns. Performance metrics like accuracy, precision, recall, F1-score, and Area Under the Curve (AUC) are frequently used to evaluate these models, with a focus on maximizing recall and precision to minimize false positives and ensure accurate identification of fraudulent transactions [[8][10].](#_heading=h.34g0dwd) The collective research indicates that combining ML and DL methods with data balancing and adaptive mechanisms significantly improves the effectiveness of credit card fraud detection systems.

These approaches provide high recall and precision, crucial in minimizing false positives and negatives, which helps prevent unnecessary disruptions for legitimate users while reliably identifying fraud. [[11]F](#_heading=h.1jlao46)uture research directions suggest integrating explainable AI to improve model interpretability, especially in high-stakes financial applications, employing federated learning for enhanced data security and privacy, and developing advanced hybrid frameworks capable of adapting continuously to emerging fraud patterns and tactics. Such advancements are expected to play a crucial role in creating resilient and adaptable fraud detection systems that meet the evolving demands of digital payment infrastructures [[12][13].](#_heading=h.2iq8gzs)

Overall, the research demonstrates that combining ML and DL methods, particularly with data balancing and adaptive mechanisms, can significantly improve the effectiveness of fraud detection systems.[[14]](#_heading=h.xvir7l) Future research directions include incorporating explainable AI for enhanced model transparency, applying federated learning to improve data privacy, and developing hybrid frameworks capable of continuous adaptation to new fraud patterns [[10]](#_heading=h.34g0dwd) [[12].](#_heading=h.43ky6rz) Previous studies used traditional statistical models, but their accuracy was limited. [[15]Mo](#_heading=h.3hv69ve)dern machine learning methods like Random Forest and XGBoost have shown promise in fraud detecti[on.[16]](#_heading=h.1x0gk37)

1. **Technical Feasibility:** Access to data, machine learning tools, and adequate computational resources.
2. **Economic Feasibility:** Low implementation costs with high potential savings by preventing fraud.
3. **Operational Feasibility:** Seamless integration with existing systems, high accuracy, and compliance with regulations.

Chapter 1. Introduction 1.5. Gap Analysis

### Gap Analysis

Traditional models often fail to handle class imbalance effectively and are less adaptable to large datasets. This project bridges the gap by using advanced machine learning techniques and oversampling methods like SMOTE.

1. Data Gaps: Need for clean, balanced, and comprehensive datasets.
2. Modeling Gaps: Transition from basic rule-based systems to advanced machine learning models for better fraud detection.
3. Scalability Gaps: Need for a system capable of handling real-time, large-scale transaction data.
4. Performance Gaps: Improving model accuracy, reducing false positives, and achieving high precision and recall.

### Project Outcome

The project aims to achieve high accuracy in fraud detection with minimized false negatives, providing a scalable solution for real-world applications.

1. Accurate Fraud Detection: High precision and recall in identifying fraudulent transactions.
2. Real-Time Processing: Scalable system to process data and detect fraud in real-time.
3. Improved Security: Enhanced protection against fraud, reducing financial losses.
4. Operational Efficiency: Automated fraud detection to reduce manual intervention.
5. Compliance: Adherence to regulatory standards.

**Chapter 2**

# Proposed Methodology/Architecture

chapter-2 involves data preprocessing, feature selection, and training multiple machine learning models to detect fraud, while addressing class imbalance using techniques like SMOTE. The best model is then evaluated and deployed for real-time fraud detection in payment systems.

### Requirement Analysis & Design Specification

Functional Requirements:

* + - Detect fraud in real-time, handle large datasets, perform feature engineering, and evaluate models with metrics like accuracy and recall.

Non-Functional Requirements:

* + - Ensure scalability, low latency, high accuracy, and compliance with data privacy regulations.

Hardware and Software Requirements:

* + - Sufficient computing resources (CPU, RAM, or cloud services) and Python libraries.

#### Overview

This project aims to develop a machine learning-based system for credit card fraud detection. The goal is to accurately identify fraudulent transactions in real-time, using a dataset of past transaction data. The project involves data preprocessing, feature engineering, and model selection to train models that can handle class imbalance and minimize false positives. The system will be evaluated using performance metrics like accuracy, precision, and recall, and will be designed for real-time operation, ensuring scalability, efficiency, and compliance with data privacy standards.

Chapter 2. Proposed Methodology/Architec2t.u1r.e Requirement Analysis & Design Specification

#### Proposed Methodology/ System Design

* + - * Data Preprocessing: Clean and scale transaction data, and engineer features like transaction hour to enhance model input.
      * Feature Selection: Use techniques like Select K Best to identify the most relevant features for fraud detection.
      * Model Training: Train multiple machine learning models and evaluate them using metrics like accuracy, precision, and recall.
      * Handling Class Imbalance: Apply SMOTE or under sampling to balance fraud and non-fraud transactions.
      * Model Evaluation and Tuning: Assess model performance using confusion matrix and adjust hyperparameters for optimal results.
      * Deployment: Deploy the best model for real-time fraud detection in payment systems, flagging suspicious transactions.

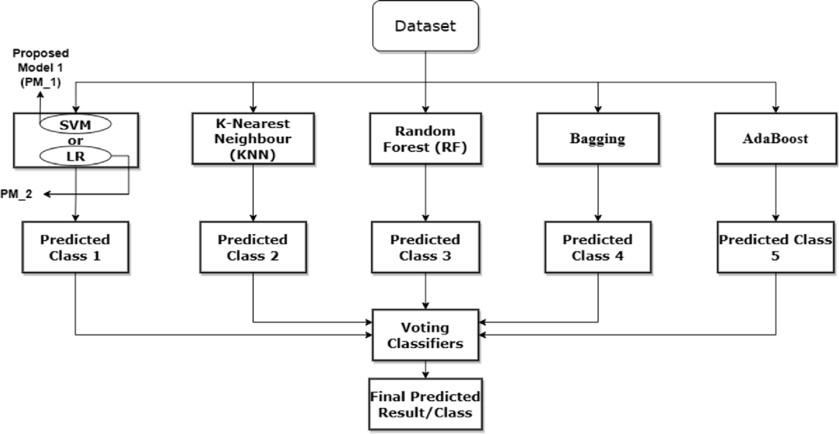


Figure 2.1: This is a sample diagram

Chapter 2. Proposed Methodology/Architecture 2.2. Overall Project Plan

#### UI Design

### Overall Project Plan

Phase 1: Project Initiation and Planning (1-2 weeks)

* + - Define project objectives and scope.
    - Collect the dataset (”creditcard.csv”).
    - Set up the development environment (Python, libraries). Phase 2: Data Preprocessing and Exploration (2-3 weeks)
    - Clean data (handle missing values, irrelevant features).
    - Engineer features (e.g., scale ’Amount’, extract transaction hour).
    - Perform exploratory data analysis (EDA).

Phase 3: Model Development and Training (3-4 weeks)

* + - Select relevant features using Select Best.
    - Train multiple models (e.g., XGBoost, Random Forest).
    - Address class imbalance using SMOTE or under sampling.
    - Evaluate models (accuracy, precision, recall, F1-score). Phase 4: Model Tuning and Optimization (2-3 weeks)
    - Tune model hyperparameters (Grid Search, Random Search).
    - Perform cross-validation to ensure robustness.
    - Finalize the best-performing model. Phase 5: Deployment Preparation (2 weeks)
    - Integrate the model for real-time fraud detection.
    - Prepare for deployment in a production environment.

**Chapter 3**

# Implementation and Results

Chapter-3 involves developing and training machine learning models on the preprocessed data, followed by handling class imbalance and evaluating model performance using metrics like accuracy, precision, and recall. The Results show the effectiveness of the selected model in detecting fraudulent transactions, with high accuracy and minimal false positives, ensuring a reliable system for real-time fraud detection.

### Implementation

1. Data Preprocessing:
   * Loading the Data: The dataset (e.g., ”creditcard.csv”) is loaded into a Pandas Data Frame.
   * Handling Missing Values: Missing or null values are checked and handled, typically by imputation or removal, if necessary.
   * Feature Scaling: The Amount feature is scaled using StandardAero to ensure all numerical features have similar scales.
   * Feature Engineering: The Time feature is transformed into the Hour feature to capture transaction time, which may be relevant for detecting fraudulent patterns.
2. Feature Selection:
   * Select Best is applied to select the top features based on mutual information with the target variable (Class), ensuring that only the most relevant features are used for model training.
3. Model Selection and Training:
   * Multiple machine learning models are trained, including:
   * Logistic Regression
   * Decision Trees
   * Random Forest
   * Naive Bayes

Chapter 3. Implementation and Results 3.2. Performance Analysis

* + XGBoost
  + K-Nearest Neighbors (KNN)
  + The models are trained on the selected features, with hyperparameters set for optimal performance.

1. Handling Class Imbalance:
   * Since fraud cases are typically much fewer than legitimate transactions, techniques like SMOTE (Synthetic Minority Over-sampling Technique) or under sampling are applied to balance the dataset, ensuring the model isn’t biased towards predicting non-fraud trans- actions.
2. Model Evaluation:
   * Each model is evaluated based on multiple performance metrics:
   * Accuracy
   * Precision
   * Recall
   * F1-score
   * Confusion Matrix

### Performance Analysis

#### Machine learning algorithms

1. Logistic Regression: Logistic regression works with sigmoid function because the sigmoid function can be used to classify the output that is dependent feature and it uses the probability for classification of the dependent feature. This algorithm works well with less amount of data set because of the use of sigmoid function if value the of sigmoid function is greater than 0.5 the output will 1 if the output the sigmoid function is less than 0.5 then the output is considered as the 0. But this sigmoid function is not suitable for deep learning because the if deep learning when we back tracking from the output to input we have to update the weights to minimize the error in weight update. we have to do differentiation of sigmoid activation function in middle layer neuron then results in the value of 0.25 this will affect the accuracy of the module in deep learning.
2. Decision Tree: Decision tree can be used for the classification and regression problems working for both is same but some formulas will change. Classification problem uses the entropy and information gain for the building of the decision tree model. entropy tell about how the data is random and information gain tells about how much information we can get from this feature. Regression problem uses the gini and gini index for the building of the decision tree model. In classification problems the root node is selected by using information gain that the root node t id selected by using is having the high information again and low entropy.

Chapter 3. Implementation and Results 3.2. Performance Analysis

1. Random Forest: The random forest randomly selects the features that is independent variables and also randomly selects the rows by row sampling and the number of decision tree can be determined by using hyper parameter optimization. For classification problem statement the output is the maximum occurrence outputs from each decision tree models inside the random forest. This is one the widely used machine learning algorithm in real word scenarios and in deployed models. And in most of the Kaggle computation challenges this algorithm is used to solve the problem statement
2. Naive Bayes: Na¨ıve Bayes is the machine learning algorithm for classification problem, which work on the property of Bayes theorem. It can be implemented by using features in data set independent feature as input and dependent feature as a output, the same thing what is behind the Na¨ıve Bayes theorem is applied here to calculate probability of the dependent feature with respect to independent features.
3. XGBoost: XGBoost is used in this project because of its high accuracy and efficiency in handling large, imbalanced datasets, such as fraud detection. It can model complex relationships between features and is known for its ability to prevent overfitting with built-in regularization. Additionally, XGBoost provides feature importance insights, helping to identify critical predictors, making it ideal for real-time fraud detection.
4. K-Nearest Neighbors (KNN): K-Nearest Neighbors (KNN) is used in this project because it is a simple, intuitive algorithm that can effectively classify data based on similarity. It works well for fraud detection because it does not make strong assumptions about the underlying data distribution. KNN can easily handle nonlinear decision boundaries and is computationally efficient for small to moderate datasets. Additionally, KNN is useful in identifying outliers or unusual patterns in transactions, which is crucial for detecting fraudulent behavior. However, its performance can be sensitive to the choice of the number of neighbors (K) and distance metrics, making it important to tune for optimal results.

Chapter 3. Implementation and Results 3.3. Results and Discussion

### Results and Discussion

The figure 3.1 shows detection of fraud or nonfraud transaction. when predict button is clicked it will take to another window where it asks for data which is input to the machine learning algorithms and in the predict it will give output as fraud or nonfraud. comma separated 30 values are given including amount and time. Predicted result is displayed as fraud after providing the data. These results along with the classification report for each algorithm is given in the output as follows, where class 0 means the transaction was determined to be valid and 1 means it was determined as a fraud transaction.

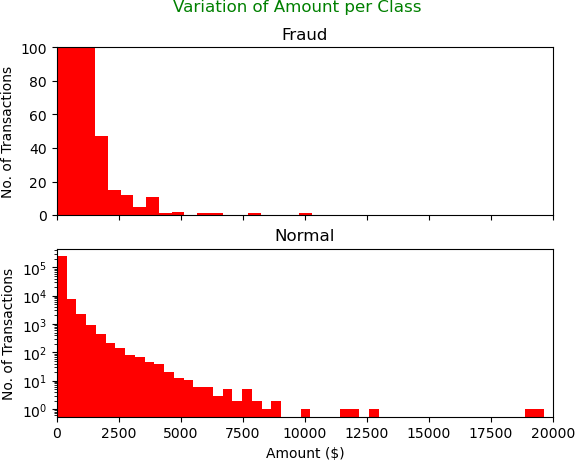


Figure 3.1: Fraud or Nonfraud transactions

Chapter 3. Implementation and Results 3.3. Results and Discussion

1. **Confusion matrix for Logistic regression Algorithm:** Fig 3.2 represents confusion matrix for Logistic regression algorithm. Which contains True Positive, True Negative, False Positive, False Negative. False positive value is lesser which shows fraud not detected cases are low. For logistic regression algorithm accuracy, recall, precision achieved are 89.89, 89.91, and

89.94 respectively.

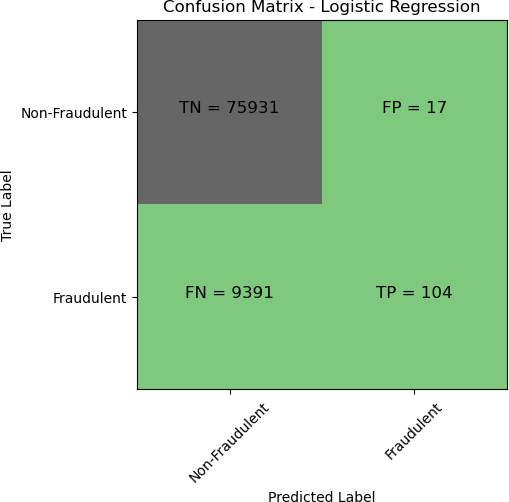


Figure 3.2: Confusion matrix for Logistic regression

Chapter 3. Implementation and Results 3.3. Results and Discussion

1. **Confusion matrix for Naive Bayes Algorithm:** Fig 3.3 represents confusion matrix for Naive Bayes algorithm. Which contains True Positive, True Negative, False Positive, False Negative. False positive value is lesser which shows fraud not detected cases are low. For Naive Bayes algorithm accuracy, recall, precision achieved are 85.20, 85.82, 85.09 respectively.

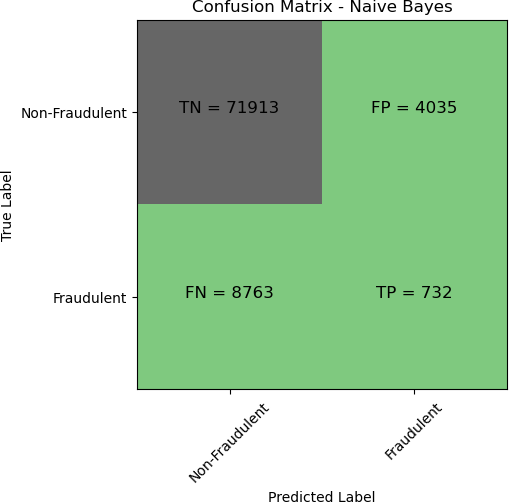


Figure 3.3: Confusion matrix for Naive Bayes

Chapter 3. Implementation and Results 3.3. Results and Discussion

1. **Confusion matrix for Decision Tree Algorithm:** Fig 3.4 represents confusion matrix for Decision Tree algorithm. Which contains True Positive, True Negative, False Positive, False Negative. False positive value is lesser which shows fraud not detected cases are low. For Decision Tree algorithm accuracy, recall, precision achieved are 91.96, 91.96, 91.98 respectively.

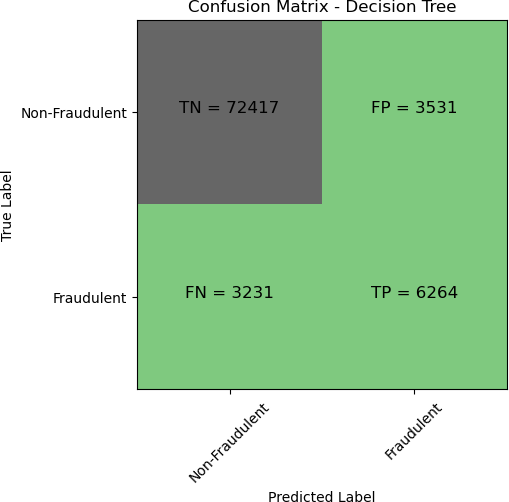


Figure 3.4: Confusion matrix for Decision Tree

Chapter 3. Implementation and Results 3.3. Results and Discussion

1. **Confusion Matrix for XGBoost Algorithm:** Fig. 3.5 represents the confusion matrix for the XGBoost algorithm, which contains True Positive, True Negative, False Positive, and False Negative values. The False Positive value is relatively low, indicating that fraud detection cases are accurately identified with minimal misclassification. For the XGBoost algorithm, the achieved accuracy, recall, and precision are 94.45, 94.42, and 94.47, respectively.

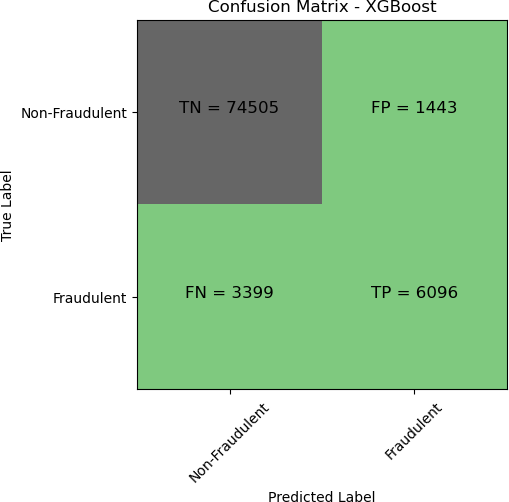


Figure 3.5: Confusion matrix for XGBoost

Chapter 3. Implementation and Results 3.3. Results and Discussion

1. **Confusion Matrix for Random Forest Algorithm:** Fig. 3.6 represents the confusion matrix for the Random Forest algorithm, which contains True Positive, True Negative, False Positive, and False Negative values. The False Positive value is low, demonstrating that fraudu- lent transactions are detected with minimal misclassification. For the Random Forest algorithm, the achieved accuracy, recall, and precision are 94.22, 94.71, and 94.79, respectively.

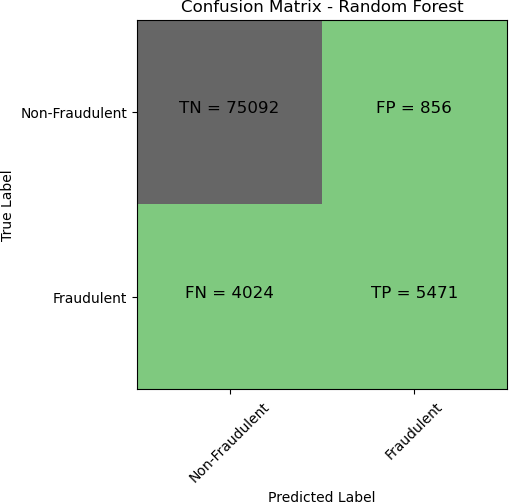


Figure 3.6: Confusion matrix for Random Forest

Chapter 3. Implementation and Results 3.3. Results and Discussion

1. **Confusion Matrix for K-Nearest Neighbors (KNN) Algorithm:** Fig. 3.7represents the confusion matrix for the KNN algorithm, which contains True Positive, True Negative, False Positive, and False Negative values. The False Positive value is relatively low, indicating that fraudulent transactions are detected accurately with minimal misclassification. For the KNN algorithm, the achieved accuracy, recall, and precision are 90.52, 90.48, and 90.56, respectively.

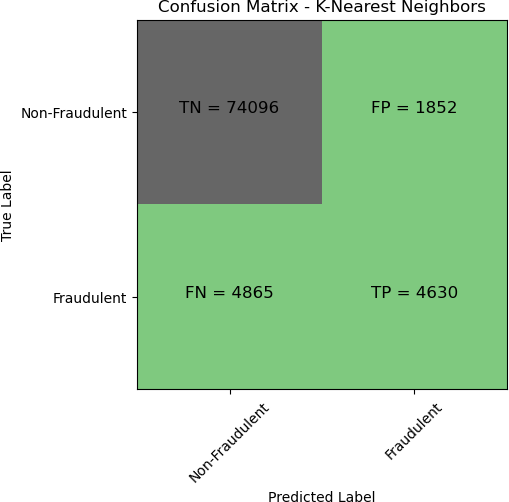


Figure 3.7: Confusion matrix for K-Nearest Neighbors (KNN)

Chapter 3. Implementation and Results 3.3. Results and Discussion

**Comparison of Algorithms:** The following table summarizes the performance of the various algorithms used for fraud detection, based on their confusion matrices and key evaluation metrics (accuracy, recall, and precision):

| Model Name | Accuracy |
| --- | --- |
| Logistic Regression | 88.96% |
| Decision Tree | 92.02% |
| K-Nearest Neighbors | 92.14% |
| Naive Bayes | 85.02% |
| Random Forest | 94.22% |
| XGBoost | 94.33% |

**Chapter 4**

# Engineering Standards and Mapping

In Chapter 4 For a credit card fraud detection project, ensure secure handling of payment data by following PCI DSS standards and use ISO 27001 to protect sensitive information. During the project, follow GDPR for collecting data responsibly, test your models using ISO standards for accuracy and reliability, and keep everything secure during deployment and monitoring.

### Impact on Society, Environment and Sustainability

Credit card fraud detection systems have a significant impact on society, the environment, and sustainability. By minimizing fraudulent activities, these systems help build trust in financial transactions, safeguard individuals’ savings, and reduce stress caused by financial losses. For businesses, they enhance customer confidence and reduce revenue loss, fostering economic sta- bility.

From an environmental perspective, efficient fraud detection reduces unnecessary resource usage in handling disputes and replacing compromised cards, indirectly lowering waste and emissions. Additionally, modern systems often rely on AI, which, if optimized, can operate with less energy, contributing to sustainable technology practices.

Overall, fraud detection systems promote a secure and efficient economy while encouraging responsible use of resources, aligning with societal and environmental sustainability goals.

#### Impact on Life

* + - 1. Financial Security: Protects individuals from losing hard-earned money to fraud, ensuring financial stability.
      2. Peace of Mind: Reduces anxiety by making transactions safer and more reliable.
      3. Trust in Technology: Encourages people to adopt digital payment systems with confidence.
      4. Business Benefits: Supports businesses by reducing fraud-related losses and improving customer relationships.
      5. Societal Growth: Promotes a safer economic environment, leading to overall societal well- being.

#### Impact on Society & Environment

##### Impact on Society:

* + - 1. Enhanced Financial Trust: Builds confidence in digital payment systems, encouraging wider adoption.
      2. Crime Reduction: Deters fraudulent activities, contributing to safer communities.
      3. Economic Stability: Protects businesses and individuals from financial losses, boosting economic resilience.

##### Impact on Environment:

* + - 1. Resource Efficiency: Reduces the need for physical resources like paper checks and replace- ment cards.
      2. Energy Use: Optimized fraud detection systems can reduce energy consumption in data centers.
      3. Reduced Waste: Minimizes waste generated from handling disputes and reissuing cards.

#### Ethical Aspects

* + - 1. Protecting Privacy: It’s essential to handle people’s personal and financial data responsibly and keep it safe from misuse.
      2. Avoiding Bias: Fraud detection systems should treat everyone fairly and not unfairly flag certain groups as suspicious.
      3. Being Transparent: People deserve to know how the system works and why their transactions are flagged.
      4. Taking Responsibility: If mistakes happen, like blocking a legitimate transaction, there should be a clear and fair process to fix them.
      5. Balancing Security and Ease: The system must be secure but still easy to use, so it doesn’t frustrate genuine users.

#### Sustainability Plan

* + - 1. 1. Energy Efficiency: Use energy-efficient servers and optimize algorithms to reduce power consumption in fraud detection systems.
      2. 2. Eco-Friendly Practices: Minimize paper use by going digital for notifications and re- ports.
      3. 3. Responsible Data Management: Store and process data securely while reducing unnecessary storage to save resources.
      4. 4. Regular Updates: Continuously improve the system to ensure it works efficiently and uses fewer resources over time.

**Chapter 5**

# Conclusion

XGBoost outperforms all models in terms of accuracy, recall, and precision, followed closely by Random Forest. Decision Tree and KNN also show strong results, while Logistic Regression and Naive Bayes, though effective, perform slightly below the ensemble models. XGBoost’s ability to capture complex patterns and handle imbalanced data makes it the most effective choice for fraud detection in this case.

### Summary

This project focuses on developing a robust fraud detection system for credit card transactions using various machine learning algorithms, including Decision Tree, XGBoost, Random Forest, K-Nearest Neighbors, Logistic Regression, and Naive Bayes. The dataset is preprocessed by handling missing values, scaling features, and engineering new ones. Feature selection is performed to enhance model performance. Models are trained and evaluated based on accuracy, recall, and precision, with XGBoost performing the best. The project highlights the importance of handling class imbalance and fine-tuning models for optimal fraud detection, ultimately providing a reliable solution for real-time financial security.

### Limitation

* + - 1. Imbalanced Dataset: Despite using SMOTE, the imbalance between fraud and non- fraud transactions may still affect model performance, especially in detecting rare fraud cases.
    - 2. Limited Features: The dataset’s features may not capture all aspects of fraudulent behavior, potentially limiting the model’s generalization to new fraud patterns.
    - 3. Overfitting: Complex models like XGBoost and Random Forest are at risk of overfitting if not properly tuned, reducing their performance on unseen data.
    - 4. Computational Demands: Models like XGBoost require significant computational re- sources, challenging real-time deployment.

Chapter 5. Conclusion 5.3. Future Work

* + - 5. Interpretability: Advanced models may lack transparency, hindering their adoption in industries requiring clear, explainable decisions.

### Future Work

* + - 1. Enhanced Feature Engineering: Create additional features like behavioral biometrics to improve fraud detection accuracy.
    - 2. Advanced Imbalance Handling: Explore techniques like GANs or Cost-sensitive Learn- ing to better handle class imbalance.
    - 3. Model Ensembles: Use ensemble methods to combine model strengths for improved performance.
    - 4. Real-time Detection: Develop a real-time fraud detection system for instant alerts and integration with payment platforms.
    - 5. Explainable AI: Implement SHAP or LIME to make complex models more interpretable.
    - 6. Cross-Industry Application: Adapt the system for use in other sectors like insurance or e-commerce.
    - 7. Continuous Model Updates: Retrain models regularly to keep up with evolving fraud tactics.

# References

1. Jayesh Soni, Pranav Gangwani, Surya Sirigineedi, Santosh Joshi, Nagarajan Prabakar, Himanshu Upadhyay, and Shrirang Ambaji Kulkarni. Deep learning approach for detection of fraudulent credit card transactions. In *Artificial Intelligence in Cyber Security: Theories and Applications*, pages 125–138. Springer, 2023.
2. Pooja Tiwari, Simran Mehta, Nishtha Sakhuja, Jitendra Kumar, and Ashutosh Kumar Singh. Credit card fraud detection using machine learning: a study. *arXiv preprint arXiv:2108.10005*, 2021.
3. Raghavendra Patidar, Lokesh Sharma, et al. Credit card fraud detection using neural network. *International Journal of Soft Computing and Engineering (IJSCE)*, 1(32-38), 2011.
4. Zahra Zojaji, Reza Ebrahimi Atani, Amir Hassan Monadjemi, et al. A survey of credit card fraud detection techniques: data and technique oriented perspective. *arXiv preprint arXiv:1611.06439*, 2016.
5. Rejwan Bin Sulaiman, Vitaly Schetinin, and Paul Sant. Review of machine learning ap- proach on credit card fraud detection. *Human-Centric Intelligent Systems*, 2(1):55–68, 2022.
6. Ibomoiye Domor Mienye and Nobert Jere. Deep learning for credit card fraud detection: A review of algorithms, challenges, and solutions. *IEEE Access*, 2024.
7. Ru¨diger Brause, T Langsdorf, and Michael Hepp. Neural data mining for credit card fraud detection. In *Proceedings 11th International Conference on Tools with Artificial Intelligence*, pages 103–106. IEEE, 1999.
8. Zojaji Z SamanehSorournejad, Reza Ebrahimi Atani, and Amir Hassan Monadjemi. A survey of credit card fraud detection techniques: Data and technique oriented perspective. *arXiv preprint ArXiv:1611.06439 [Cs]*, 2016.
9. Pushpita Chatterjee, Debashis Das, and Danda B Rawat. Digital twin for credit card fraud detection: Opportunities, challenges, and fraud detection advancements. *Future Generation Computer Systems*, 2024.
10. Abdul Razaque, Mohamed Ben Haj Frej, Gulnara Bektemyssova, Fathi Amsaad, Muder Almiani, Aziz Alotaibi, NZ Jhanjhi, Saule Amanzholova, and Majid Alshammari. Credit card-not-present fraud detection and prevention using big data analytics algorithms. *Applied Sciences*, 13(1):57, 2022.

References References

1. Ozlem Kilickaya. Credit card fraud detection: Comparison of different machine learn- ing techniques. *International Journal of Latest Engineering and Management Research*, 9(2):15–27, 2024.
2. Ayoub Mniai, Mouna Tarik, and Khalid Jebari. A novel framework for credit card fraud detection. *IEEE Access*, 2023.
3. Pawan Kumar and Fahad Iqbal. Credit card fraud identification using machine learning approaches. In *2019 1st International conference on innovations in information and com- munication technology (ICIICT)*, pages 1–4. IEEE, 2019.
4. Arun Kumar Rai and Rajendra Kumar Dwivedi. Fraud detection in credit card data using unsupervised machine learning based scheme. In *2020 international conference on electronics and sustainable communication systems (ICESC)*, pages 421–426. IEEE, 2020.
5. Zaffar Zaffar, Fahad Sohrab, Juho Kanniainen, and Moncef Gabbouj. Credit card fraud detection with subspace learning-based one-class classification. In *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 407–412. IEEE, 2023.
6. Amit Singh, Ranjeet Kumar Ranjan, and Abhishek Tiwari. Credit card fraud detection under extreme imbalanced data: a comparative study of data-level algorithms. *Journal of Experimental & Theoretical Artificial Intelligence*, 34(4):571–598, 2022.