Intelligent Fraud Prevention in Credit Card Transactions

By

Md. Shakil Hossain (Exam Roll: 230362, Class Roll: 1061)

A Thesis proposal submitted to the
Institute of Information Technology
in partial fulfilment of the requirements for the degree of
Master of Science in Information and Communication Technology

To
Rashed Mazumder
Associate Professor
Institute of Information Technology
Jahangirnagar University

Institute of Information Technology
Jahangirnagar University
Savar, Dhaka-1342
February, 2025

DECLARATION

This proposal is submitted to the **Institute of Information Technology**, Jahangirnagar University, Savar, Dhaka in partial fulfillment of the requirements for having the M.S. degree in ICT. This is also needed to certify that the project work is under the Masters 1st Semester course of "**MICT-5199: Project Work-1**".

I hereby declare that this project is based on the results found by myself. Materials of work found by other researchers are mentioned by reference. This project, neither in whole nor in part, has been previously submitted for any degree.

Md. Shakil Hossain
Class Roll: 1061
Exam Roll: 230362

CERTIFICATE

The project titled "**Intelligent Fraud Prevention in Credit Card Transactions**" submitted by Md.Shakil Hossain-1061, Session: 2022-2023, has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Master of Science in Information Technology in February 2025.

Rashed Mazumder
Supervisor

Accepted and approved in partial fulfillment of the requirement for the degree Master of Science in Information Technology.

BOARD OF EXAMINERS

Dr. Risala Tasin Khan — Chairman
Professor, IIT, JU — Masters exam Committee

Dr. Shamim Al Mamun — Member
Professor, IIT, JU — Masters exam Committee

Rashed Mazumder — Member
Associate Professor, IIT, JU — Masters exam Committee

Prof.Dr. Sajjad Waheed — External Member

ACKNOWLEDGEMENTS

First of all we would like to thank the Almighty. We are also pleased to have the opportunity to express our heartfelt thanks and gratitude to those who contributed greatly to this report.

This project is performed under the supervision of **Rashed Mazumder**, Associate Professor, Institute of Information Technology (IIT), Jahangirnagar University, Savar, Dhaka. During the work, he has supplied us with a number of materials related to the present investigation. Without his help, kind support and generous time span he has given, we could not have performed the project work successfully in due time. First and foremost, we wish to acknowledge our profound and sincere gratitude to him for his guidance, valuable suggestions, encouragement, and cordial cooperation.

We express our thanks to the other faculty members of IIT who have helped us directly or indirectly by providing their valuable support in completing this work. We express our gratitude to all other sources from which we have found help. We are indebted to those who have helped us directly or indirectly in completing this work. We would like to thank all the staff of IIT, Jahangirnagar University, and our friends who have helped us by giving their encouragement and cooperation throughout the work.

ABSTRACT

Credit card fraud is a significant problem, with billions of dollars lost each year. Machine learning can be used to detect credit card fraud by identifying patterns that are indicative of fraudulent transactions. Credit card fraud refers to the physical loss of a credit card or the loss of sensitive credit card information. Many machine-learning algorithms can be used for detection. This project proposes to develop a machine-learning model to detect credit card fraud. The model will be trained on a dataset of historical credit card transactions and evaluated on a holdout dataset of unseen transactions.

Keywords: Credit Card Fraud Detection, Fraud Detection, Fraudulent Transactions, K- Nearest Neighbors, Support Vector Machine, Logistic Regression, Decision Tree.

LIST OF ABBREVIATIONS

IIT	Institute of Information Technology
JU	Jahangirnagar University
LR	Logistic Regression
KNN	K-Nearest Neighbors
DT	Decision Tree
SVM	Support Vector Machine
GNB	Gaussian Naive Bayes
RF	Random Forest
XGBoost	Extreme Gradient Boosting
AURPC	Areas under the precision-reall curve
AUROC	Areas under the receiver operating characteristic curve
KYRBS	Korean Youth Risk Behavior Survey.
AUC	Accuracy

LIST OF FIGURES

Figure

LIST OF TABLES

Table

TABLE OF CONTENTS

CHAPTER I

Introduction

1.1 Overview

With the increase of people using credit cards in their daily lives, credit card companies should take special care of the security and safety of the customers. According to (Credit card statistics 2021), the number of people using credit cards worldwide was 2.8 billion in 2019; also, 70those users own a single card. Reports of Credit card fraud in the U.S. rose by 44.7in 2020. There are two kinds of credit card fraud, and the first is having a credit card account opened under your name by an identity thief. Reports of this fraudulent behaviour increased 48to 2020. The second type is when an identity thief uses an existing account you created, usually by stealing the information on the credit card. Reports on this type of Fraud increased 9to 2020 (Daly, 2021). Those statistics caught We's attention as the numbers have increased drastically and rapidly throughout the years, which motivated We to resolve the issue analytically by using different machine learning methods to detect fraudulent credit card transactions within numerous transactions.

1.2 Problem Statement

Credit card frauds are increasing heavily because of fraud financial loss is increasing drastically. Every year due to fraud Billions of amounts lost. To analyze the fraud there is lack of research. Many machine learning algorithms are implemented to detect real world credit card fraud.[1]

1.3 Motivation

The motivation behind this project stems from the increasing prevalence of credit card fraud, which has become a significant concern globally. With the rise in credit card usage, the number of fraudulent transactions has also surged, leading to substantial financial losses for both individuals and financial institutions. According to the report, the number of credit card users worldwide was 2.8 billion in 2019, and reports of credit card fraud in the U.S. rose by 44.7

The primary motivation for this project is to leverage machine learning techniques to develop a robust model capable of accurately identifying fraudulent credit card transactions. By analyzing historical transaction data, the project aims to identify patterns and anomalies that are indicative of fraud. The goal is to compare various machine learning algorithms, such as K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Logistic Regression, and Decision Trees, to determine which model performs best in detecting fraudulent transactions.

The project is driven by the desire to enhance the security of credit card transactions, thereby protecting consumers from unauthorized charges and reducing the financial burden on credit card companies. By developing an effective fraud detection system, the project aims to contribute to the ongoing efforts to combat credit card fraud, ultimately leading to increased customer satisfaction and trust in financial systems. The motivation is not only to address the current challenges but also to explore future improvements and applications of machine learning in fraud detection, such as integrating additional data sources like telecom data to enhance the accuracy and reliability of the detection models.[2]

1.4 Objective

1. **Fraud Detection:** To create a machine learning model capable of accurately identifying fraudulent transactions within a dataset of credit card transactions. This involves distinguishing between legitimate and fraudulent activities to prevent unauthorized charges. [3]

2. **Comparison of Machine Learning Algorithms:** To implement and compare the performance of various machine learning algorithms, including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Logistic Regression, and Decision Trees, in detecting credit card fraud. The goal is to determine which algorithm provides the highest accuracy and reliability in identifying

fraudulent transactions.[4]

3. **Data Analysis and Preprocessing:** To analyze and preprocess the dataset, ensuring it is suitable for training and testing the machine learning models. This includes handling missing data, normalizing features, and transforming data as necessary.[5]

4. **Model Evaluation:** To evaluate the performance of each machine learning model using metrics such as accuracy, precision, recall, and the confusion matrix. The evaluation will help identify the most effective model for fraud detection.[6]

5. **Optimization:** To optimize the selected models by tuning hyperparameters and improving their performance in detecting fraudulent transactions while minimizing false positives and false negatives.[7]

6. **Deployment:** To prepare the best-performing model for potential deployment in real-world scenarios, where financial institutions can use it to detect and prevent credit card fraud in real time.[8]

7. **Future Improvements:** To explore potential future enhancements, such as integrating additional data sources (e.g., telecom data) to improve the model's accuracy and reliability, and to adapt the model for different datasets and fraud detection scenarios.[9]

1.5 Research Question

What machine learning model is most suited for detecting fraudulent credit card transactions?

1.6 Research Outline

Chapter 1: This chapter will introduce the problem of intelligent fraud prevention in credit card transactions.

Chapter 2: This chapter will review the related literature on credit card fraud detection and various supervised machine learning algorithms.

Chapter 3: This chapter will describe the methodology.

Chapter 4: This chapter will describe Expected Results.

Chapter 5: This chapter will describe the Project Plan.

CHAPTER II

Literature Review

2.1 Related Work

Zareapoor and his research team used multiple techniques to determine the best-performing model for detecting fraudulent transactions, which was established using the Accuracy of the model, the speed of detection and the cost. The models used were Neural Network, Bayesian Network, SVM, KNN and. The comparison table in theresearch paper showed that the Bayesian Network was high-speed in finding fraudulent transactions with high Accuracy. The N.N. performed well, as the detection was fast, with a medium accuracy. KNN's speed was good with medium Accuracy, andfinally, SVM scored one of the lower scores, as the speed was low, and the Accuracy wasmedium. As for the cost, All models built were expensive (Zareapoor et al., 2012).[10]

The model used by Alenzi and Aljehane to detect Fraud in credit cards was Logistic Regression. Their model scored 97.2% in Accuracy, 97% sensitivity and 2.8% Error Rate. A comparison was performed between their model and two other classifiers , which are3 They were voting Classifier and KNN. V.C. scored 90% in Accuracy, 88% sensitivity and 10% error rate, as for KNN where k = 1:10, the Accuracy of the model was 93%, the sensitivity 94%and 7% for the error rate (Alenzi Aljehane, 2020).[11]

Maniraj's team built a model to recognize if any new transaction is Fraud or non-fraud. Their goal was to get 100% in detecting fraudulent transactions andtry to minimize the incorrectly classified fraud instances. Their model has performed well as they got 99.7% of the fraudulent transactions (Maniraj et al., 2019).[12]

The classification approach used by Dheepa and Dhanapal was the behaviour-based classification approach, using a Support Vector Machine, where the behavioural patternsof the customers were analyzed to distinguish credit card fraud, such as the amount, date,time, place, and frequency of card usage. The Accuracy achieved by their approach wasmore than 80% (Dheepa Dhanapal, 2012).[13]

Mailini and Pushpa proposed using KNN and Outlier detection in identifying credit card fraud. After performing their model oversampled data, the authors found that the most suitable method for detecting and determining target instance anomaly is KNN, which showed that it is most suited to detecting Fraud with memory limitation. As for Outlier detection, the computation and memory required for credit card fraud detectionis much less in addition to working faster and better in large online datasets. However, their work and results showed that KNN was more accurate and efficient (Malini Pushpa,2017).[14]

Maes and his team proposed using Bayesian and Neural Networks to detect credit card fraud. Their results showed that Bayesian performance is 8% more effective in detecting Fraud than ANN, meaning BBN sometimes detects 8% more fraudulent transactions. In addition to the Learning times, ANN can go up to several hours,whereas BBN takes only 20 minutes (Maes et al., 2002).[15]

Jain's team used several ML techniques to distinguish credit card fraud; three of them are SVM, ANN and KNN. Then, to compare the outcome of each model, they calculated the true positive (T.P.), false Negative (F.N.), false positive (F.P.), and true negative (T.N.) generated. ANN scored 99.71% accuracy, 99.68% precision, and 0.12% false alarm rate.SVM accuracy is 94.65%, 85.45% for the precision, and 5.2% false alarm rate. Moreover, finally,the Accuracy of KNN is 97.15%, the precision is 96.84%, and the false alarm rate is 2.88% (Jain et al., 2019).[16]

Dighe and his team used KNN, Logistic Regression and Neural Networks, multilayer perceptron and Decision Tree in their work, then evaluated the results regarding numerous accuracy metrics. Of all the models created, the best performing one is KNN, which scored 99.13%, then in second place performing model at 96.40% and in last place is logistic Regression with 96.27% (Dighe et al., 2018).[17]

Sahin and Duman used four Support Vector Machine methods in detecting credit card fraud. (SVM) Support Vector Machine with RBF, Polynomial, Sigmoid, and Linear Kernel,all models scored 99.87% in the training model and 83.02% in the testing part of the model (Sahin Duman, 2011).[18]

2.2 Supervised Machine Learning Classification

This step describes supervised machine learning classifications: Logistic Regression, k-Nearest Neighbours, Random Forest, Support Vector Machine, Naive Bayes, Decision Tree, and Gradient Boosting Analysis.

2.2.1 Logistic Regression (LR)

A supervised machine learning approach for data categorization called logistic regression (LR) mines real-valued features from the input, multiplies each by weight, adds them all together, and then passes the total through a sigmoid function to generate a probability. A decision is concluded using a threshold. LR is a method for classifying our data set that squeezes the result of a linear equation between 0 and 1 using the logistic function rather than fitting a straight line or hyperplane.[19] The definition of the logistic function is:

$$\text{Logistic}(\eta) = 1/(1 + exp(-\eta)) \tag{2.1}$$

As η goes from $-\infty$ to ∞, logistic (η) goes from 0 to 1, a “squashing function”.

2.2.2 k-Nearest Neighbours (KNN)

(KNN) is a non-parametric method that we applied to the categorization of diabetic data. In a KNN, data is categorized by a majority vote of its neighbours. Based on an estimate of distance, the data is assigned to the class most mutual among its closest neighbours. The data is assigned to the class of its closest neighbour if K = 1. [20]:

2.2.3 Support Vector Machine (SVM)

A separating hyperplane defines the discriminative classifier as a support vector machine. In another way, the algorithm creates an ideal hyperplane that classifies the new data point using labelled training data. This hyperplane is a line that divides a plane into two sections in two dimensions, with each class lying on each side. It uses a kernel approach to classify non-linear data and linear data. A hyperplane is the collection of points $\vec{x}$ that meet the following equation:

$$w.\vec{x} - b = 0 \tag{2.2}$$

The offset of the hyperplane from the origin along the vector $\vec{w}$ is defined by the parameter $\frac{b}{\|\vec{w}\|}$, which must be maximized. SVM utilizes a "regularization parameter" to manage the trade-off between the complexity of the assumption space and experimental error. [21].

2.2.4 Decision Tree (DT)

A classifier that divides the instance space recursively is called a DT. The nodes in the decision tree combine to form a tree; the tree begins at a node known as the "root" with no incoming edges. Every other node has a single edge that enters. Decision nodes are leaf-like nodes. The process of computing Information Gain (IG) nominates the child node.

Information Gain is equal to [weights average] * Entropy(children) - Entropy(parent).

Since P(xi) represents the likelihood that child node i will occur, entropy(Ci) = -P(xi) log P(xi).

The parent node for the following level has the highest IG. This process is repeated until a leaf node is reached and the decision tree is finished. [22]:

2.2.5 Gaussian Naive Bayes (GNB)

A probability distribution function called the GNB classifier links neuronal activity to the averages and variances of activation under different impulse situations. A condition label is produced during the classifier's creation. The classes are assumed to have Gaussian normal distributions by the classifier.[23]

For every data point, the z-score distance—the difference between the distance from the class mean and the class standard deviation—is estimated.

$$Z_A = \frac{x - \mu_A}{\sigma_A} \tag{2.3}$$

A probability value for each z-score is derived from the equation for Gaussian normal distribution and applied to the observation of data point x. The co-variance across dimensions is not modelled by the GNB classifier.

2.2.6 Random Forest (RF)

The trees and bagging algorithms served as models for the collective algorithm known as RF. When the data set has many input variables, it functions well. Using several decision tree classifiers on various subsamples of the data set, it is a meta-estimator that uses the mean value to improve model accuracy and prevent overfitting. With the following labels: [L1, L2, L3, L4] for the training data set: [X1, X2, X3, X4]. The random forest method may produce three decision trees based on the input of subsets: [X1, X3, X4], [X2, X3, X4], and [X1, X2, X4]. Lastly, it uses the majority of votes from each decision tree to predict the class. Generally, a forest is stronger and more dependable with more trees. Similarly, the random forest classifier produces higher accuracy results in the more trees in the forest.[24].

2.2.7 eXtreme Gradient Boosting (XGBoost)

XGBoost is a powerful and efficient implementation of gradient boosting for supervised learning. It is widely used for classification and regression tasks due to its high performance and accuracy. The main components and features of the XGBoost algorithm include:

- **Gradient Boosting**: XGBoost builds an ensemble of decision trees in a sequential manner. Each new tree aims to correct the errors made by the previous trees.

- **Regularization**: XGBoost includes $L1$ and $L2$ regularization terms to control the complexity of the model and enhance generalization.

- **Tree Pruning**: XGBoost uses a pruning strategy to remove branches that do not contribute to the final predictions, ensuring a more efficient model.

- **Handling Missing Values**: XGBoost can handle missing data internally, making it robust for real-world datasets.

- **Parallel Processing**: XGBoost supports parallel processing, significantly speeding up the training process.

- **Sparsity Awareness**: XGBoost is optimized for sparse data, which is common in many machine learning tasks.

Due to its flexibility, scalability, and performance, XGBoost has become a popular choice for many machine learning competitions and practical applications. [25]

2.3 Supervised Machine Learning Regression

This stage explains the supervised machine learning classifications of Lasso Regression Analysis, Ridge Regression, and Linear Regression.

2.3.1 Linear Regression

By fitting a linear equation to observed data, linear regression is a supervised machine learning approach that ascertains the linear relationship between the dependent variable and one or more independent features.[26]

As there is just one independent variable and one dependent variable, this is the most basic type of linear regression.

2.3.2 Ridge Regression

A model-tuning technique called ridge regression is applied to any data that exhibits multicollinearity. This technique carries out L2 regularization. Predicted values deviate significantly from fundamental values when multicollinearity is present, least-squares are impartial, and variances are high.[27]

2.3.3 Lasso Regression

In Lasso Regression, also known as the Least Absolute Shrinkage and Selection Operator, a penalty term has been added to the standard linear regression objective function. A punishment term is determined by the absolute values of the regression model's coefficients, and the degree of penalization is controlled by the hyperparameterλ.[28]

CHAPTER III

Methodology

3.1 Block Diagram

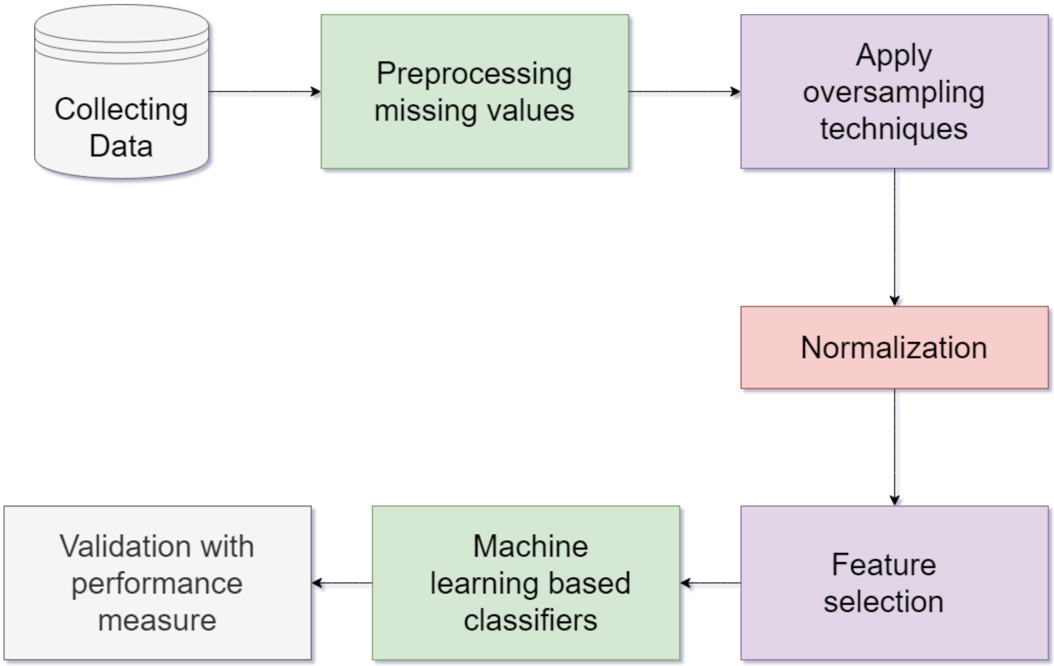
Collecting Data
Preprocessing missing values
Apply oversampling techniques
Normalization
Feature selection
Machine learning based classifiers
Validation with performance measure

Figure 3.1: Diagram of the workflow

Normalize the data: Scaling the data to give each feature a similar value is known as normalization. Increasing part comparability can enhance machine learning model performance.

Model training: A machine learning algorithm will be trained in the second phase using the gathered data. SVM or another supervised learning technique will be used to prepare the model.

Model evaluation: The third phase will assess the machine learning model's performance on a holdout dataset of unseen transactions. We will use measures like accuracy, precision, and recall to evaluate the model's performance.

3.2 Data collection

We collect data using a dataset form kaggle. This dataset contains credit card transactions made by European cardholders in the year 2023. It comprises over 550,000 records, and the data has been anonymized to protect the cardholders' identities. The primary objective of this dataset is to facilitate the development of fraud detection algorithms and models to identify potentially fraudulent transactions.
Dataset : https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dataset-2023

3.3 Data Processing

3.3.1 Data Cleaning

- Impute the missing values with the column's mean, median, or mode.
- Drop the rows with missing values.
- Utilize machine learning models such as isnull() and heatmap() to forecast the absent values.

3.3.2 Oversampling

One method for addressing class imbalance issues where one class greatly outnumbers the others is oversampling, a data augmentation approach. By increasing the number of cases from the underrepresented class, it seeks to restore equilibrium to the distribution of training data. [29]

In machine learning, oversampling is used to rectify dataset class imbalance. Here, we used the random oversampling technique to balance the final column. Before oversampling, only 2% of the total data is at risk.

After applying the oversampling technique, the final column is the same with '0' and '1'. It's helpful to train a model efficiently.

3.3.3 Normalization

One type of feature scaling that converts the range of features to a standard scale is normalisation. Normalisation—or any data scaling technique, for that matter—is necessary only when your dataset contains features with different ranges. Normalisation includes a variety of methods designed for various data distributions and model specifications. To standardise the characteristics, scale them to unit variance and subtract the mean. A sample's standard score is determined as follows:

$$Z = \frac{x - \mu}{s}$$

where s is the standard deviation among the practice examples (or one if `with-std=False`), and μ is the mean among the practice examples (or zero if `with-mean=False`). [30]

3.4 Feature Selection

Filter Methods Rather than focusing on cross-validation performance, filter approaches extract the inherent characteristics of the features as assessed by univariate statistics. Compared to wrapper methods, these techniques are less computationally expensive and speedier. When working with high-dimensional data, filter approaches are computationally less expensive.

Chi-square Test : When analyzing categorical features in a dataset, the Chi-square test is employed. We determine the desired number of features with the optimal Chi-square scores by calculating the Chi-square between each feature and the target.

Correlation Coefficient : Correlation is a statistical measure of the linear relationship between two or more variables. Correlation allows us to predict one trait based on another. The idea behind using correlation for feature selection is that good variables have a strong connection with the target.

Embedded Methods : These techniques combine the advantages of the wrapper and filter approaches by incorporating feature interactions while keeping computing costs within acceptable bounds. Iterative in nature, embedded approaches handle every iteration of the model training process and meticulously identify the features that add the most to the training for a given iteration.

CHAPTER IV

Expected Results

4.1 Software and Tools Used

The system leverages a blend of open-source software, modern development frameworks, and cloud-based services to ensure an efficient and cost-effective solution. The following table summarizes the primary tools and their roles:

Tool/Platform	**Role in the System**
Python	Primary programming language for backend, data processing, and ML
TensorFlow/ PyTorch	Deep learning frameworks for building the RL model
Pandas & NumPy	Data manipulation and preprocessing
Flask / Django	Backend web frameworks for developing RESTful APIs
React / Vue.js	Frontend frameworks for building responsive user interfaces
Docker	Containerization for consistent deployment environments
Google Cloud / AWS	Cloud platforms for scalable deployment

Table 4.1: Tools and Platforms in the System

4.2 Evaluation and Deployment

We assessed the model's performance using metrics like accuracy, precision, recall, specificity, and F1-score in order to compare different algorithms. Accuracy is most

	Actual class	
Predicted class	Fraud (1)	Not Fraud (0)
Fraud (1)	True Positive (TP)	False Positive (FP)
Not Fraud (0)	False Negative (FN)	True Negative (TN)

Table 4.2: Confusion matrix.

frequently used to gauge a model's performance [31]. Our dataset is quite unbalanced; thus, comparing the models using accuracy as the only performance metric may not be appropriate in this context. Instead, we must select the best model to identify fraudulent transactions by using other measurements such as area under the curve (AUC) [32] in addition to accuracy.

The entries in the confusion matrix are defined as follows:

- **False Positive (FP)**: The total number of incorrect predictions classified as positive.
- **False Negative (FN)**: The total number of incorrect predictions classified as negative.
- **True Positive (TP)**: The total number of true predictions classified as positive.
- **True Negative (TN)**: The total number of true predictions classified as negative.

Accuracy, as a measurement metric, measures the ratio of the total number of correct predictions of fraud to the total number of predictions (both fraud and not fraud) made by the model [33]. It is calculated as:

$$\text{Accuracy} = \frac{TN + TP}{TN + TP + FN + FP} \tag{4.1}$$

Precision metric measures the ratio of correctly classified fraud transactions (TP) to the total transactions predicted to be fraud transactions (TP + FP) [34]. It is calculated as:

$$\text{Precision} = \frac{TP}{FP + TP} \tag{4.2}$$

Recall/Sensitivity, as a metric, measures the ratio of correctly classified fraud transactions (TP) to the total number of fraud transactions [35]. It is calculated as:

$$\text{Recall/Sensitivity} = \frac{TP}{TP + FN} \tag{4.3}$$

Specificity measures the ratio of correctly classified not fraud transactions (TN) to the total number of not fraud transactions [36]. It is calculated as:

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{4.4}$$

The F1 score metric measures the weighted mean of precision and recall [37]. It ranges between zero and one, with a value close to one giving the highest value. It is computed using the expression:

$$F1\ Score = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{4.5}$$

CHAPTER V

Project Plan

5.1 Flowchart

This chapter outlines the comprehensive project plan, detailing the flowchart, and a detailed timeline for all project phases—from data collection to model deployment and continuous improvement.

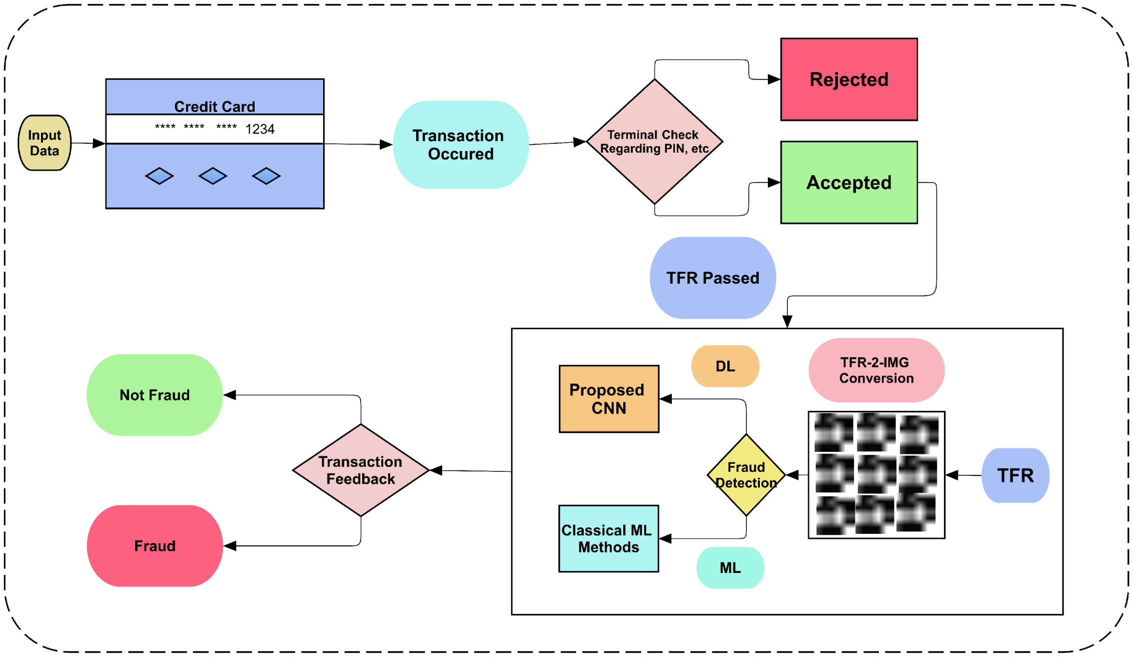

Figure 5.1: Flowchart of of proposed framework

The flowchart visually represents the sequential progression through each phase, providing a clear roadmap of the project's methodology.

5.2 Detailed Timeline

The project is segmented into several phases, each with clearly defined tasks and milestones. Below is the detailed timeline for the entire project, structured in weeks. This timeline is designed to be flexible, allowing adjustments as necessary based on interim feedback and testing results.

Phase 1: Project Planning & Requirement Analysis

- **Week 1:**
  - Define scope, objectives, and success criteria.
  - Identify key stakeholders.
- **Week 2:**
  - Gather system requirements.
  - Outline necessary resources.

Phase 2: Data Collection & Preparation

- **Week 3:**
  - Acquire credit card transaction data from sources.
- **Week 4:**
  - Analyze data distributions.
  - Detect anomalies.
- **Week 5:**
  - Handle missing values.
  - Detect and remove outliers.
- **Week 6:**
  - Normalize features.
  - Apply SMOTE for class balancing.

Phase 3: Feature Engineering & Model Selection

- **Week 7:**
  - Select and engineer key features for fraud detection.
- **Week 8:**
  - Apply scaling and transformations for improved model performance.
- **Week 9:**
  - Evaluate different machine learning models.
  - Choose the best candidates.
- **Week 10:**
  - Implement baseline models for comparison.

Phase 4: Model Training & Optimization

- **Week 11:**
  - Train selected models with preprocessed data.
- **Week 12:**
  - Optimize model parameters for better performance.
- **Week 13:**
  - Use k-fold cross-validation to prevent overfitting.
- **Week 14:**
  - Measure precision, recall, F1-score, and AUC-ROC.
- **Week 15:**
  - Adjust thresholds.
  - Improve false positive/negative rates.
- **Week 16:**
  - Choose the best-performing model for deployment.

Phase 5: Deployment & System Integration

- **Week 17:**
  - Integrate the model into a real-time system.
- **Week 18:**
  - Develop API/web interface for fraud detection.

Phase 6: Testing & Documentation

- **Week 19:**
  - Conduct thorough testing with real-world transaction data.
- **Week 20:**
  - Prepare the final report, user manuals, and presentations.

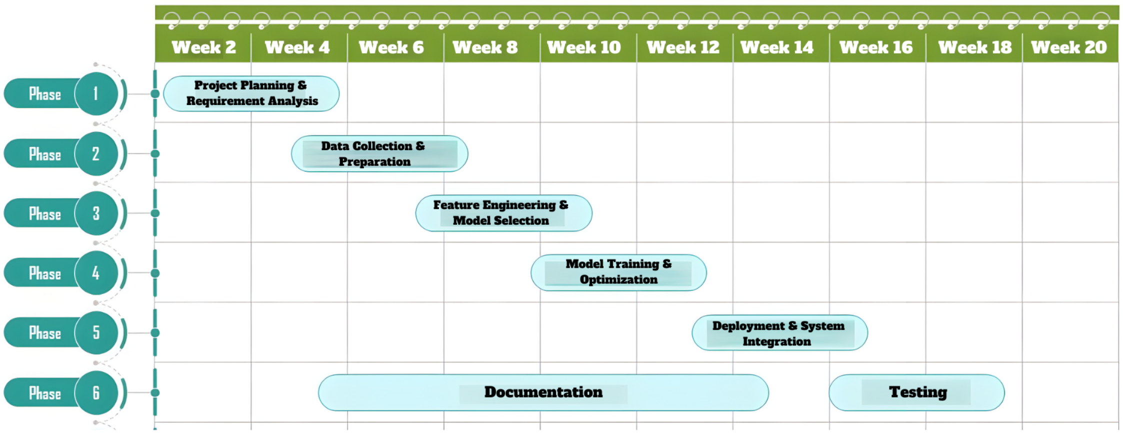

Figure 5.2: Gantt chart representing detailed timeline for each phase.

The project plan is divided into five phases. Each phase has specific tasks and milestones, ensuring a systematic approach to developing and deploying the AI-driven Crop Rotation Advisor. This detailed timeline not only guides the project through its various stages but also provides a framework for evaluating progress and ensuring that the final system is both robust and user-centric.

References

[1] N. of Authors, "Credit card fraud detection using machine learning algorithms," *ResearchGate*, 2020.

[2] N. of Authors, "Credit card fraud detection: Challenges and solutions - a review," *ResearchGate*, 2024.

[3] "Fraud detection using machine learning." `https://www.researchgate.net/publication/378258753_Fraud_Detection_using_Machine_Learning`, 2023.

[4] "Comparative analysis of selected machine learning algorithms." `https://www.eajournals.org/wp-content/uploads/Comparative-Analysis-of-Selected-Machine-Learning-Algorithms.pdf`, 2023.

[5] "Data preprocessing in data mining." `https://www.geeksforgeeks.org/data-preprocessing-in-data-mining/`, 2023.

[6] "Confusion matrix in machine learning." `https://www.geeksforgeeks.org/confusion-matrix-machine-learning/`, 2023.

[7] "Optimizing credit card fraud detection: A multi-algorithm approach with artificial neural networks and gradient boosting model." `https://www.researchgate.net/publication/387335228_OPTIMIZING_CREDIT_CARD_FRAUD_DETECTION_A_MULTI-ALGORITHM_APPROACH_WITH_ARTIFICIAL_NEURAL_NETWORKS_AND_GRADIENT_BOOSTING_MODEL`, 2023.

[8] "Machine learning in financial transaction fraud detection and prevention." `https://www.researchgate.net/publication/379494304_Machine_Learning_in_Financial_Transaction_Fraud_Detection_and_Prevention`, 2023.

[9] "Fraud detection in financial transactions using machine learning," *ScienceDirect*, 2024.

[10] Z. et al, "Analysis on credit card fraud detection techniques: Based on certain design criteria." `https://research.ijcaonline.org/volume52/number3/pxc3881538.pdf`, 2012. Accessed: 26-oct-2023.

[11] . A. N. O. Alenzi, H. Z., "Fraud detection in credit cards using logistic regression." `https://thesai.org/Publications/ViewPaper?Volume=11&Issue=12&Code=IJACSA&SerialNo=65`, 2020. Accessed: 26-oct-2023.

[12] S. A. A. S. . S. S. D. Maniraj, S. P., "Credit card fraud detection using machine learning and data science." `https://doi.org/10.17577/ijertv8is090031`, 2019. Accessed: 25-oct-2023.

[13] . D. R. Dheepa, V., "Analysis on credit card fraud detection techniques: Based on certain design behaviour-based credit card fraud detection using support vector machinescriteria." `https://doi.org/10.21917/ijsc.2012.0061`, 2012. Accessed: 26-oct-2023.

[14] . P. M. Malini, N., "Analysis of credit card fraud identification techniques based on knn and outlier detection." `https://doi.org/10.1109/aeeicb.2017.7972424`, 2017. Accessed: 26-oct-2023.

[15] T. K. V. B. . M. B. Maes, S., "Credit card fraud detection using bayesian and neural networks." `https://www.ijert.org/research/credit-card-fraud-detection-using-machine-learning-and-data-science-IJERTV8IS pdf`, 2002. Accessed: 23-oct-2023.

[16] N. S. . J. S. Jain, Y., "A comparative analysis of various credit card fraud detection techniques," 2019. Accessed: 25-oct-2023.

[17] P. S. . K. S. Dighe, D., "Detection of credit card fraud transactions using machine learning algorithms and neural networks." `https://doi.org/10.1109/iccubea.2018.8697799`, 2018. Accessed: 26-oct-2023.

[18] . D. E. . Sahin, Y., "Detecting credit card fraud by decision trees and support vector machines," 2011. Accessed: 23-oct-2023.

[19] GeeksforGeeks, "Understanding logistic regression." `https://www.geeksforgeeks.org/understanding-logistic-regression/`. Accessed: 2024-07-11.

[20] IBM, "K-nearest neighbors (knn)." https://www.ibm.com/topics/knn. Accessed: 2024-07-11.

[21] GeeksforGeeks, "Support vector machine algorithm." https://www.geeksforgeeks.org/support-vector-machine-algorithm/. Accessed: 2024-07-11.

[22] B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, 2021.

[23] Z. jun Bi, Y. quan Han, C. quan Huang, and M. Wang, "Gaussian naive bayesian data classification model based on clustering algorithm," in *Proceedings of the 2019 International Conference on Modeling, Analysis, Simulation Technologies and Applications (MASTA 2019)*, pp. 396–400, Atlantis Press, 2019/07.

[24] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, pp. 197–227, 2016.

[25] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[26] GeeksforGeeks, "Ml — linear regression." https://www.geeksforgeeks.org/ml-linear-regression/. Accessed: 2024-07-11.

[27] G. C. McDonald, "Ridge regression," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 1, no. 1, pp. 93–100, 2009.

[28] Great Learning, "Understanding of lasso regression." https://www.mygreatlearning.com/blog/understanding-of-lasso-regression/. Accessed: 2024-07-11.

[29] A. Ashraf, "Oversampling for better machine learning with imbalanced data." https://medium.com/@abdallahashraf90x/oversampling-for-better-machine-learning-with-imbalanced-data-68f9b5ac2696, 2020. Accessed: 2024-06-29.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn:

Machine learning in python." https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html, 2011. Accessed: 2024-06-29.

[31] J. Zhao, L. Wang, R. Cabral, and F. Torre, "Feature and region selection for visual learning," *Pattern Recognition*, vol. 25, no. 3, pp. 1084–1094, 2016.

[32] G. Goy, C. Gezer, and V. C. Gungor, "Makine Öğrenmesi yöntemleri ile kredi kartı sahteciliği tespiti," in *Proceedings of a Conference*, pp. 350–354, 2019.

[33] S. Bagga, A. Goyal, N. Gupta, and A. Goyal, "Credit card fraud detection using pipeling and ensemble learning," in *Procedia Computer Science*, vol. 173, pp. 104–112, 2020.

[34] M. Chen, "Bankruptcy prediction in firms with statistical and intelligent techniques and a comparison of evolutionary computation approaches," *Computers and Mathematics with Applications*, vol. 62, no. 12, pp. 4514–4524, 2011.

[35] N. Rtayli and N. Enneya, "Enhanced credit card fraud detection based on svm-recursive feature elimination and hyper-parameters optimization," *Journal of Information Security and Applications*, vol. 55, no. 1, p. 102596, 2020.

[36] S. Mittal, *Sampling Approaches for Imbalanced Data Classification Problem in Machine Learning.* Springer, 2022.

[37] T. C. Tran, B. T. District, H. Chi, M. City, T. K. Dang, and L. T. Ward, "Machine learning research," 2022.