Online Payments Fraud Detection Using Machine Learning

A Mini-Project Report Submitted in the
Partial Fulfillment of the Requirements
for the Award of the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Aravapally Shivaramakrishna	**20881A05C4**
Arra Deekshitha reddy	**20881A05C5**
Gampa Rama Krishna	**20881A05D5**

SUPERVISOR

Ms. Rayeesa Tasneem

Assistant Professor

Department of Computer Science and Engineering

VARDHAMAN COLLEGE OF ENGINEERING

(AUTONOMOUS)

Affiliated to JNTUH, Approved by AICTE, Accredited by NAAC with A++ Grade, ISO 9001:2015 Certified

Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India

June, 2023

VARDHAMAN COLLEGE OF ENGINEERING
(AUTONOMOUS)
Affiliated to JNTUH, Approved by AICTE, Accredited by NAAC with A++ Grade, ISO 9001:2015 Certified
Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project titled **Online Payments Fraud Detection Using Machine Learning** is carried out by

Aravapally Shivaramakrishna	**20881A05C4**
Arra Deekshitha reddy	**20881A05C5**
Gampa Rama Krishna	**20881A05D5**

in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** during the year 2022-23.

Signature of the Supervisor	**Signature of the HOD**
Ms. Rayeesa Tasneem	**Dr. Karnati Ramesh**
Assistant Professor	**Head of CSE**

Acknowledgement

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express our deep sense of gratitude to **Ms. Rayeesa Tasneem**, Assistant Professor and Project Supervisor, Department of Computer Science and Engineering, Vardhaman College of Engineering, for his able guidance and useful suggestions, which helped us in completing the project in time.

We are particularly thankful to **Dr. Karnati Ramesh**, the Head of the Department, Department of Computer Science and Engineering, his guidance, intense support and encouragement, which helped us to mould our project into a successful one.

We show gratitude to our honorable Principal **Dr. J.V.R. Ravindra**, for providing all facilities and support.

We avail this opportunity to express our deep sense of gratitude and heartful thanks to **Dr. Teegala Vijender Reddy**, Chairman and **Sri Teegala Upender Reddy**, Secretary of VCE, for providing a congenial atmosphere to complete this project successfully.

We also thank all the staff members of Computer Science and Engineering department for their valuable support and generous advice. Finally thanks to all our friends and family members for their continuous support and enthusiastic help.

Aravapally Shivaramakrishna(20881A05C4)

Arra Deekshitha Reddy(20881A05C5)

Gampa Rama Krishna(20881A05D5)

Abstract

As we are approaching modernity, the trend of paying online is increasing tremendously. The increase of use of online transactions is causing an increase in fraud. The frauds can be detected by using various approaches but they lag at accuracy and have some drawbacks. The online payment method leads to fraud that can happen using any payment app. The network transaction has the characteristics of low cost, wide coverage and high frequency, which makes the detection of fraud more complex. Aiming at the problem of difficult fraud detection in network transactions Fraud Detection is very important. Recent research has shown that machine learning techniques have been applied very effectively to the problem of payments related fraud detection. Such Machine Learning based technique have the potential to evolve and detect previously unseen patterns of fraud.

So, we are going to use some machine learning algorithms to predict the transaction whether it is fraud or not. The selection of an algorithm to predict fraudulent transactions is based on attributes such as accuracy, recall, F1-score or etc. By considering these factors, we can identify the algorithm with the highest performance in terms of correctly identifying fraudulent transactions, ensuring the model's effectiveness and reliability. The model can analyze the transaction data uploaded and return the fraud detection results to users. We show that our proposed approaches are able to detect fraud transactions with high accuracy and reasonably low number of false positives.

Keywords: Machine Learning, Online, Fraud, Detection, model, Transaction, Prediction, Algorithm.

Table of Contents

List of Tables

List of Figures

Abbreviations

Abbreviation	**Description**
VCE	Vardhaman College of Engineering
ML	Machine Learning
XG Boost	Extreme Gradient Boosting
KNN	K-Nearest Neighbours
RMSE	Root mean squared error
MSE	Mean squared error
NumPy	Numerical Python
Scipy	Scientific Python
SVM	Support Vector Machine

CHAPTER 1

Introduction

1.1 Introduction

The risk of payment fraud has recently become an urgent worry for people, organisations, and financial institutions alike due to the exponential development of online transactions. Real-time fraud detection has grown to be a serious problem that necessitates effective and trustworthy solutions in order to protect the integrity of online payment systems. Machine learning has become a significant tool in the fight against online payment fraud, with the capacity to quickly and accurately identify fraudulent transactions.

This document provides as an introduction to the field of machine learning-based online payment fraud detection. It examines the importance of the issue, discusses the main difficulties encountered, and focuses on the benefits of using machine learning models for fraud detection. It also gives a brief review of the most widely used machine learning approaches and algorithms in this field.

Online payment fraud detection's main objective is to spot and stop fraudulent transactions in real time, minimising financial damages for both customers and enterprises. While still somewhat effective, conventional rule-based systems sometimes struggle to keep up with changing fraud tendencies and may produce a large number of false positives. On the other side, machine learning algorithms excel in analysing massive volumes of data and learning complicated patterns, enabling them to more effectively and correctly identify fraudulent actions.Here we will explore the many steps required in creating a powerful machine learning system for detecting online payment fraud. It will go over how crucial feature engineering, model selection, and data pre-treatment. The incorporation of machine learning models into current fraud detection systems will also be covered, underlining the necessity of ongoing model monitoring and updating to stay up with developing fraud strategies.

The document will also discuss evaluation criteria like accuracy, recall, precision, and F1-score that are used to gauge how well fraud detection models perform. To ensure an efficient fraud prevention strategy, it will emphasise the need of striking a balance between the fraud detection rate and reducing false positives.

Online payment fraud detection can be greatly improved by utilising the power of machine learning, allowing businesses and financial institutions to stay one step ahead of fraudsters. In order to tackle the growing problem of online payment fraud, this document intends to stimulate the development of more sophisticated and effective fraud detection systems by offering helpful insights and recommendations for researchers, practitioners, and stakeholders in the field.

1.2 Introduction to Algorithms

1.2.1 Decision Tree

A common and understandable machine learning approach used for both classification and regression applications is the decision tree. A series of if-else criteria learned from the training data are used to divide the input space into regions in this predictive model.

The Decision Tree algorithm divides the data recursively based on the input feature values to produce a tree-like structure. Each leaf node in the tree indicates a final prediction or conclusion, whereas each internal node reflects a judgement based on a particular attribute. A strong tool for capturing complex decision boundaries and locating key features, the splitting process seeks to maximise the homogeneity of the data inside each resulting segment.

1.2.2 KNN

A machine learning approach called K-Nearest Neighbours (KNN) is utilised for both classification and regression applications. It is a non-parametric technique that bases predictions on how closely fresh data points resemble

their nearby counterparts in the training set.

While using the KNN algorithm, "K" stands for the number of nearest neighbours that will be taken into account while making predictions. The method determines the distances between each new data point and every other point in the training set when a new data point needs to be categorised. On the basis of these distances, it then chooses the K closest neighbours. The prediction for the new data point is given as either the majority class or the average value of these K neighbours.

1.2.3 XGBoost

XGBoost (Extreme Gradient Boosting) is a powerful machine learning algorithm that has gained popularity in various domains, including time series analysis. Originally developed by Tianqi Chen, XGBoost is an optimized implementation of the gradient boosting algorithm that combines the strengths of boosting with decision trees.

Gradient boosting is a process to convert weak learners to strong learners, in an iterative fashion. The name XGBoost refers to the engineering goal to push the limit of computational resources for boosted tree algorithms. Ever since its introduction in 2014, XGBoost has proven to be a very powerful machine learning technique and is usually the go-to algorithm in many Machine Learning competitions.

1.2.4 Random Forest

Due to its reliability and efficiency, Random Forest is a potent machine learning method that is frequently utilised in many fields, including fraud detection. Multiple decision trees are combined in this ensemble learning technique to produce predictions. To ensure diversity and minimise overfitting, each decision tree in the forest is built using a random subset of the training data and characteristics.

The Random Forest technique generates a final forecast by combining the

predictions of various decision trees. Each tree is independently constructed during training, and during prediction, the final result is determined by the majority vote, or the average of the forecasts from all trees. Accuracy, generalisation, and the capacity to handle high-dimensional datasets are all improved by this ensemble technique.

1.2.5 SVM

Support Vector Machines (SVM) is a powerful machine learning algorithm used for both classification and regression tasks. SVM is particularly effective in scenarios where the data is separable into distinct classes or when a clear margin between classes can be defined.

The primary objective of SVM is to find the best hyperplane that separates the data into different classes while maximizing the margin, which is the distance between the hyperplane and the closest data points from each class. These closest data points are called support vectors, hence the name of the algorithm.

1.2.6 Logistic Regression

An effective statistical modelling method that is particularly well suited for binary classification issues is logistic regression. Contrary to linear regression, which forecasts continuous numerical values, logistic regression aims to forecast the likelihood that an event or result will fall into one of two categories.

The logistic function, commonly called the sigmoid function, is the central idea of logistic regression. Any real-valued input is translated by this function into a value between 0 and 1, which represents the likelihood that the input belongs to a particular class. This probability is subsequently utilised in logistic regression to decide on a binary categorization.

1.2.7 Naive Bayes

A popular probabilistic machine learning approach for classification tasks is called Naive Bayes. It is based on the Bayes theorem, which determines the likelihood of an event occurring given the available information. Given the class variable, the Naive Bayes algorithm makes the assumption that the features are conditionally independent of one another.

This presumption makes probability calculations easier to understand and enables effective training and prediction. Naive Bayes has demonstrated strong performance in several real-world applications, including text classification, spam filtering, and, to some extent, fraud detection, despite its naive assumption.

CHAPTER 2

Literature Survey

2.1 Literature Survey

[1] Online Fraud Detection System: In this paper they had a behaviour based approach to classification using Support Vector machine is used to improve its accuracy. If there are any changes in the conduct of the transaction, the frauds are predicted and taken for further process. Due to large amount of data credit/debit card fraud detection problem is rectified by their proposed method.

[2] Machine Learning based Approach to Financial Fraud Detection Process in Mobile Payment System: In this paper their proposal was to detecting mobile payment fraud based on machine learning, supervised and unsupervised method to detect fraud and process large amounts of financial data. Moreover, our approach performed sampling process and feature selection process for fast processing with large volumes of transaction data and to achieve high accuracy in mobile payment detection. F-measure and ROC curve are used to validate our proposed model.

[3] Online Transaction Fraud Detection System Based on Machine Learning: Prediction: This paper designed two fraud detection algorithms based on Fully Connected Neural Network, whose AUC values can achieve 0.912 and 0.969 respectively. Meanwhile, we designed an interactive online transaction fraud detection system based on Random Forest model, which can automatically analyze the transaction data uploaded and return the fraud detection results to users.

[3] Online Transaction Fraud Detection System Using Machine Learning

and E-Commerce: In this paper they used BLA to detect this problem. An advantage to use BLA approach to reduce number of positive false transactions identified as malicious by an FDS although they are genuine. An FDS runs at a credit card issuing bank. Each incoming transaction is submitted to the FDS for verification. FDS receives the card details and transaction value to verify, whether the transaction is genuine or not. The types of goods that are bought in that transaction are not known to the FDS. Bank declines the transaction if FDS confirms the transaction to be fraud. User spending patterns and geographical location is used to verify the identity. If any unusual pattern is detected, the system requires re-verification. The previous data of the user the system recognizes unusual patterns in the payment procedure.

[4] The proposed solution is a Machine Learning model that will serve the purpose of detecting “fraudulent” and “genuine” transactions in real time. This is beneficial for all sectors that are even mildly aligned to finance. The solution will help them analyse based on various factors if the ongoing transaction can be harmful and will prevent many unfortunate incidents.

CHAPTER 3

Methodology

3.1 Objectives

1. We are developing a model that can accurately identify fraudulent transactions while minimizing the number of false positives.

2. The model identifies or predicts whether the transaction is fraud or not.

3. The model predicts whether fraud or not based on the training example in the shorter time.

3.2 Outcomes

i) The model can analyze the transaction data uploaded and return the fraud detection results to users.

ii) The model can which can effectively predict the type of transaction based on the training data of various features in a short time

3.3 About Dataset

We had Collected the Dataset from online Website Kaggle. The Dataset contain more than ten thousand records. It has different Attributes like:

Table 3.1: About Attributes in Dataset

Attributes	Description
step	represents a unit of time where 1 step equals 1 hour.
type	type of online transaction.
amount	the amount of the transaction.
nameOrig	customer starting the transaction.
oldbalanceOrg	balance before the transaction.
newbalanceOrig	balance after the transaction.
nameDest	recipient of the transaction.
oldbalanceDest	initial balance of recipient before the transaction.
newbalanceDest	the new balance of recipient after the transaction.
isFraud	class label

3.4 Dataset

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud
0	1	PAYMENT	9839.64	C1231006815	170136.00	160296.36	M1979787155	0.0	0.00	0.0
1	1	PAYMENT	1864.28	C1666544295	21249.00	19384.72	M2044282225	0.0	0.00	0.0
2	1	TRANSFER	181.00	C1305486145	181.00	0.00	C553264065	0.0	0.00	1.0
3	1	CASH_OUT	181.00	C840083671	181.00	0.00	C38997010	21182.0	0.00	1.0
4	1	PAYMENT	11668.14	C2048537720	41554.00	29885.86	M1230701703	0.0	0.00	0.0
...	...	...	...	...	...	...	...	...	...	...
110808	11	PAYMENT	36633.75	C825489209	40317.31	3683.56	M263628748	0.0	0.00	0.0
110809	11	PAYMENT	19512.89	C57441272	3683.56	0.00	M1965409129	0.0	0.00	0.0
110810	11	PAYMENT	21213.14	C1526823115	0.00	0.00	M1630743712	0.0	0.00	0.0
110811	11	TRANSFER	266490.27	C23079109	92890.00	0.00	C1025031603	0.0	266490.27	0.0
110812	11	CASH_OUT	118777.61	C1510313227	0.00	0.00	C601893033	1019293.0	NaN	NaN

110813 rows × 11 columns

Figure 3.1: Dataset

3.5 Types of Class Label in Dataset

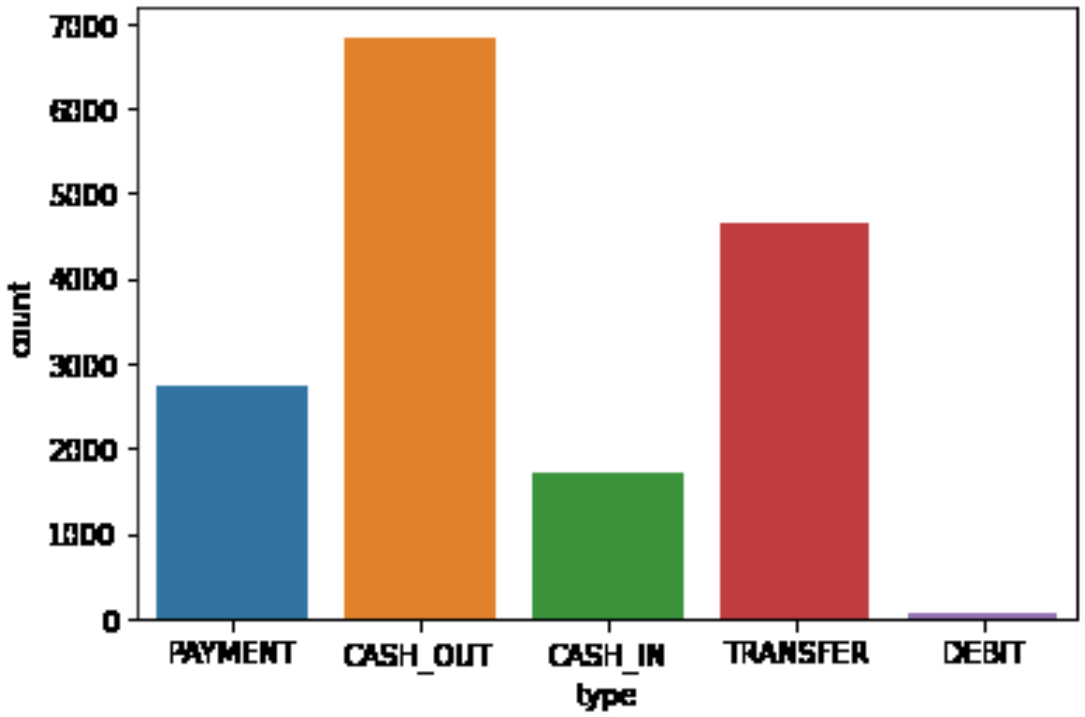

Figure 3.2: Class label

3.6 Data Pre-Processing

In order to guarantee that the data used to train models for machine learning are precise, consistent, and appropriate for analysis, data pre-processing is a crucial step. Here are some of the procedures we used to organise and clean up the data set.

3.6.1 Handling missing values:

Identify and handle missing data points in the dataset. This can involve imputing missing values using techniques such as mean, median, mode, or using more advanced methods like regression or K-nearest neighbors.

3.6.2 Removing duplicate data:

To prevent bias and inaccuracy in the analysis, locate and eliminate duplicate records from the dataset. Duplicates can be found using all or a portion of the features/columns.

3.6.3 Handling outliers:

Outliers are data points that significantly deviate from the rest of the data. Decide whether to remove or transform outliers based on the context of the problem and the impact they may have on the analysis. Outliers can be detected using statistical methods or domain knowledge.

3.6.4 Handling irrelevant or redundant features:

Identify and remove features that are irrelevant or redundant for the machine learning task. This can be done by performing feature selection techniques such as correlation analysis, feature importance ranking, or using domain knowledge.

3.6.5 Data normalization:

To guarantee that various characteristics are on a comparable scale, normalise the data. For algorithms like distance-based approaches (like k-nearest neighbours) that are sensitive to feature sizes, this phase is very crucial.

3.6.6 Splitting the dataset:

Create training, validation, and testing sets from the dataset. The validation set is used to fine-tune hyperparameters, the testing set is used to assess the performance of the final model, and the training set is used to train the model.

These are a few of the typical machine learning data cleaning procedures. Depending on the type of data being utilised and the machine learning issue at hand, several particular processes and methodologies may be employed.

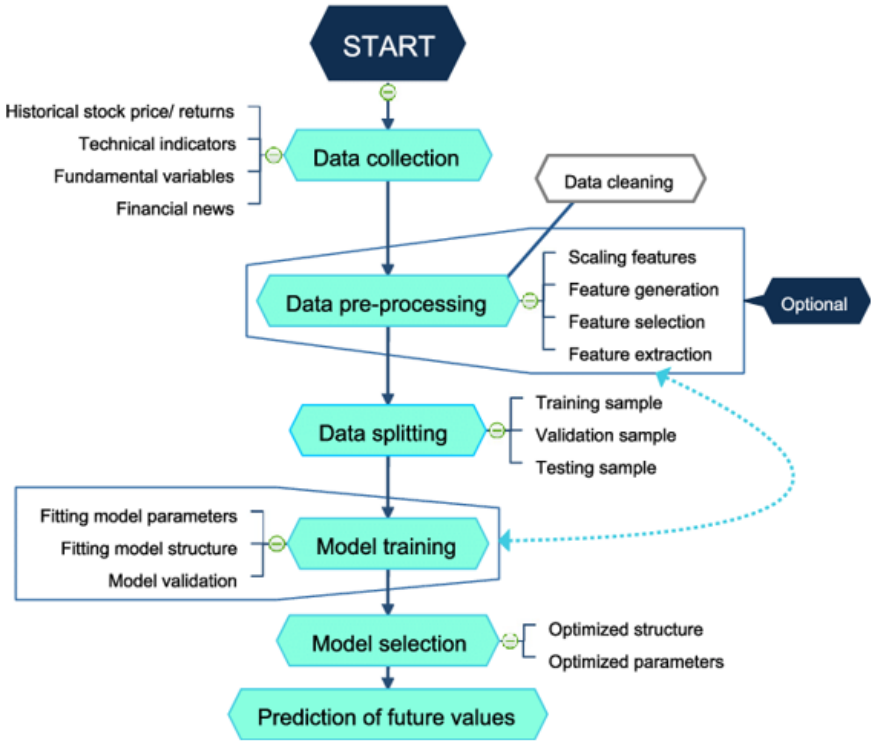

Figure 3.3: Different stages of Data

3.7 UML Diagrams

3.7.1 Use Case Diagram

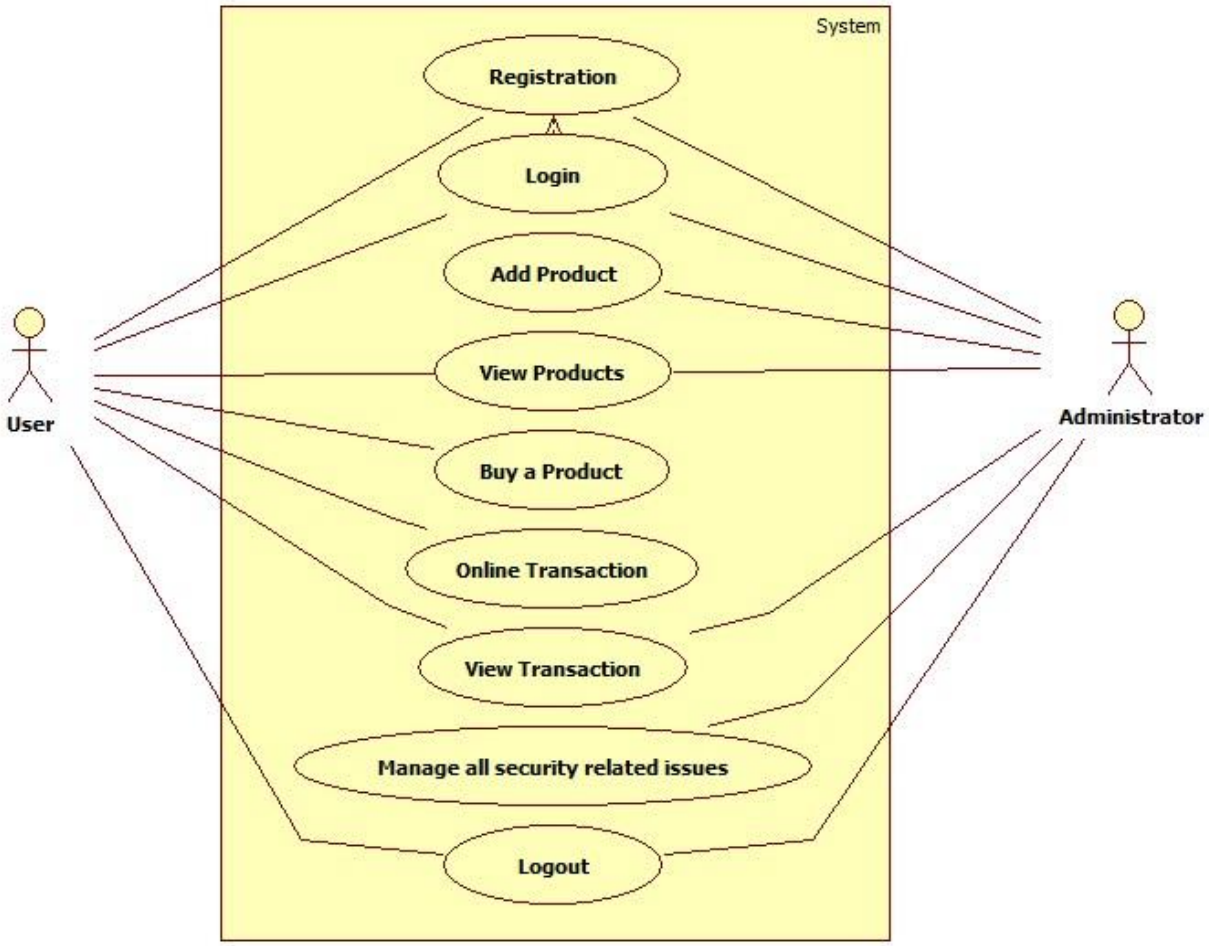

Figure 3.4: Use Case Diagram

3.7.2 Class Diagram

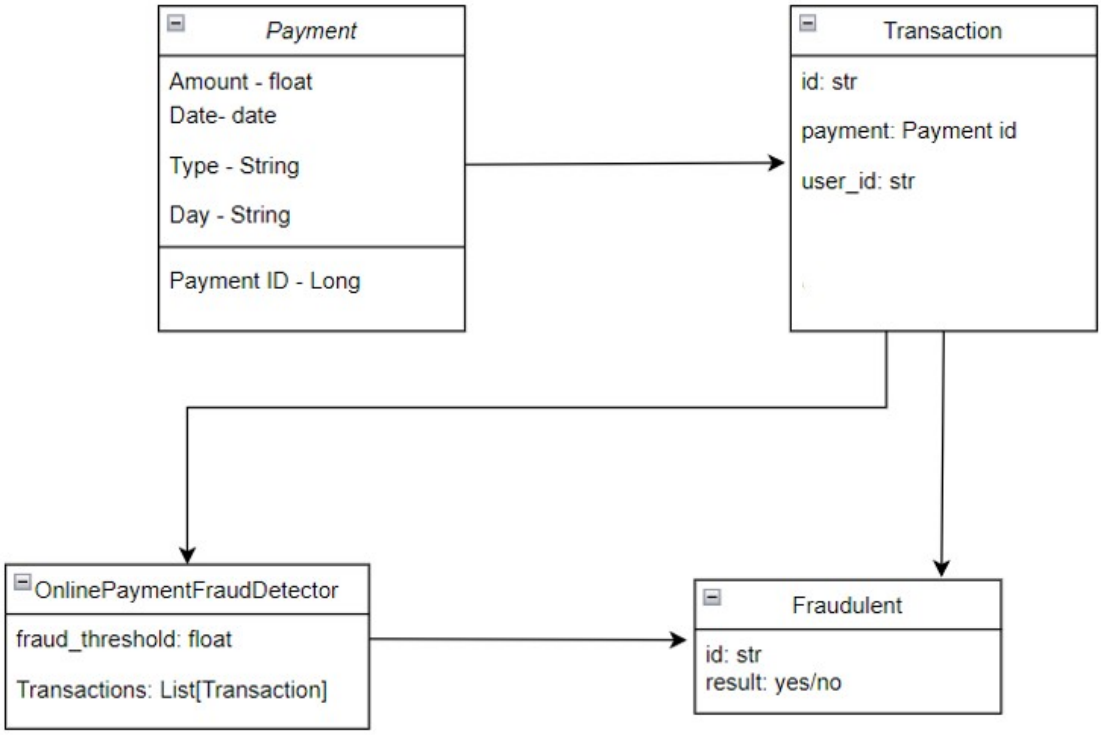

Figure 3.5: Class Diagram

3.7.3 Sequence Diagram

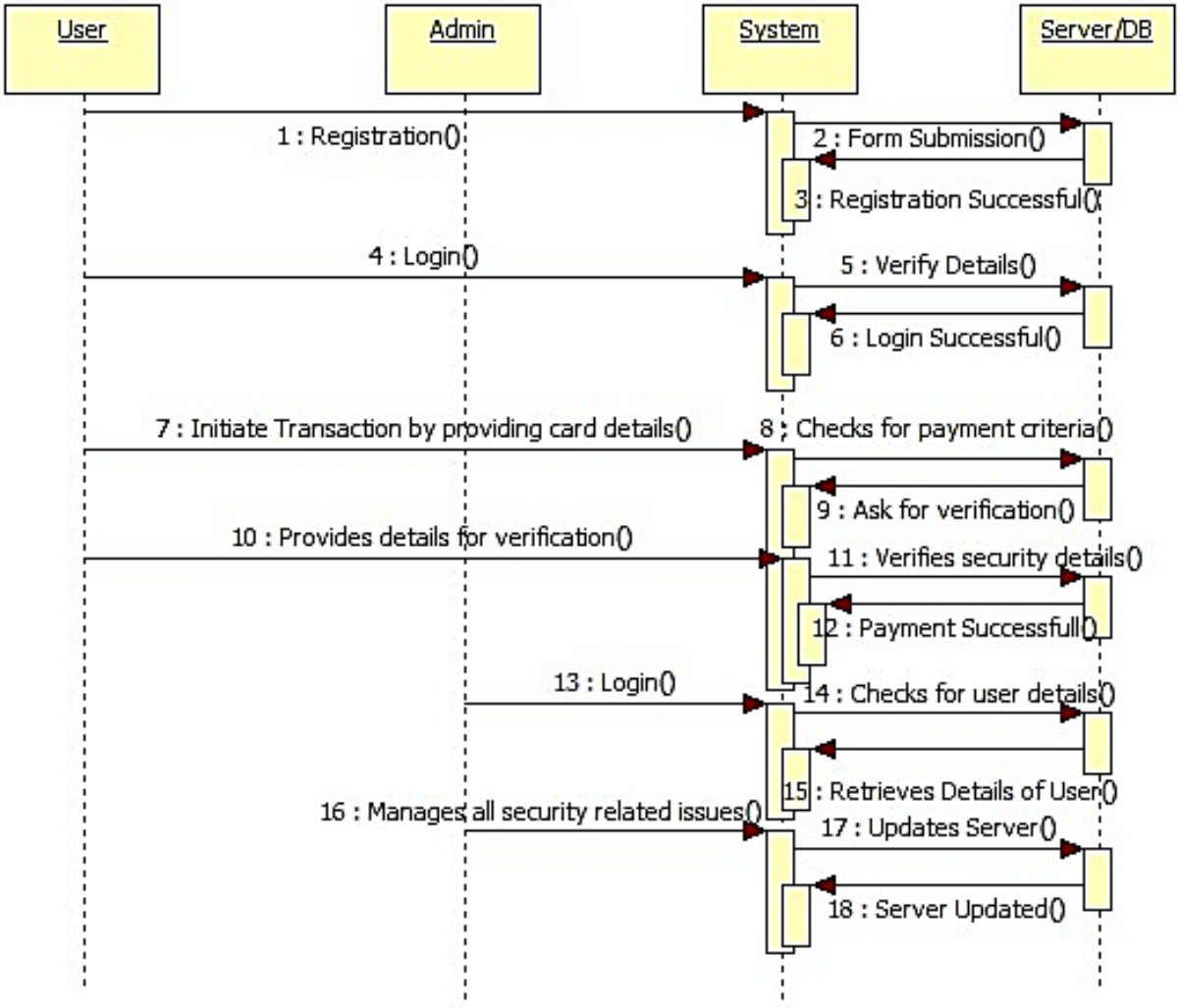

Figure 3.6: Sequence Diagram

3.7.4 Activity Diagram

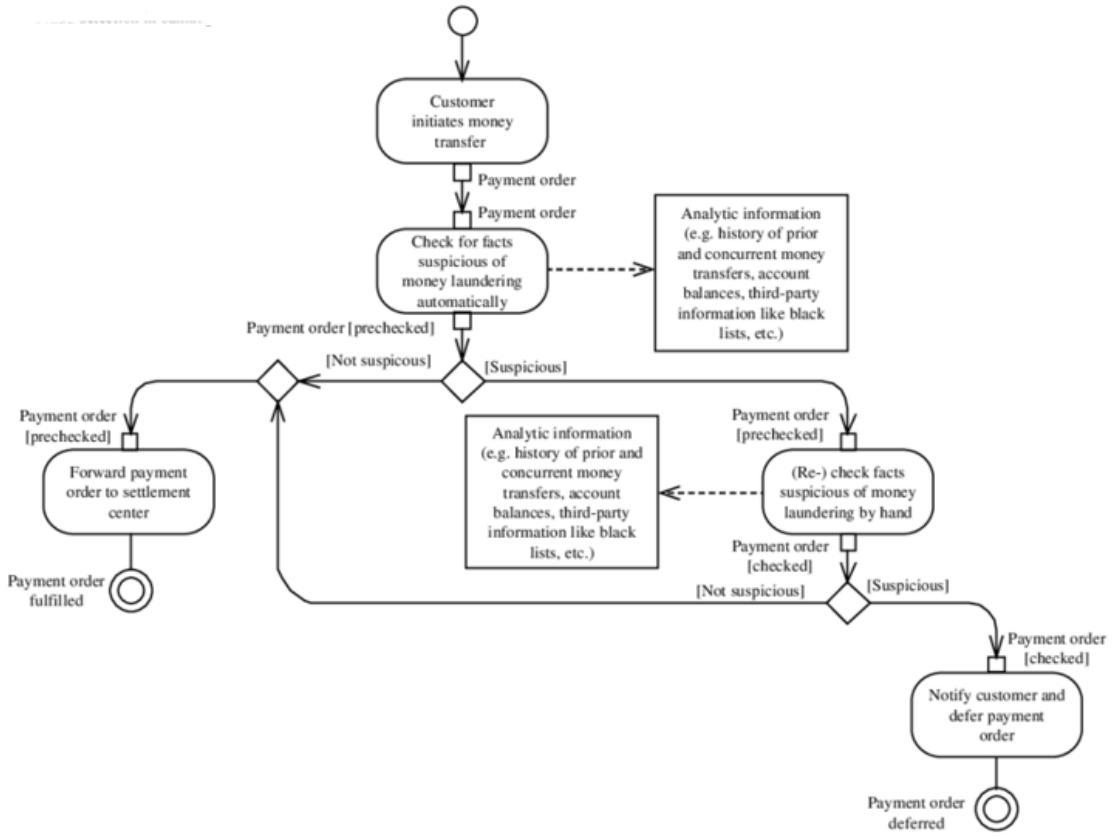

Figure 3.7: Activity Diagram

3.7.5 State Diagram

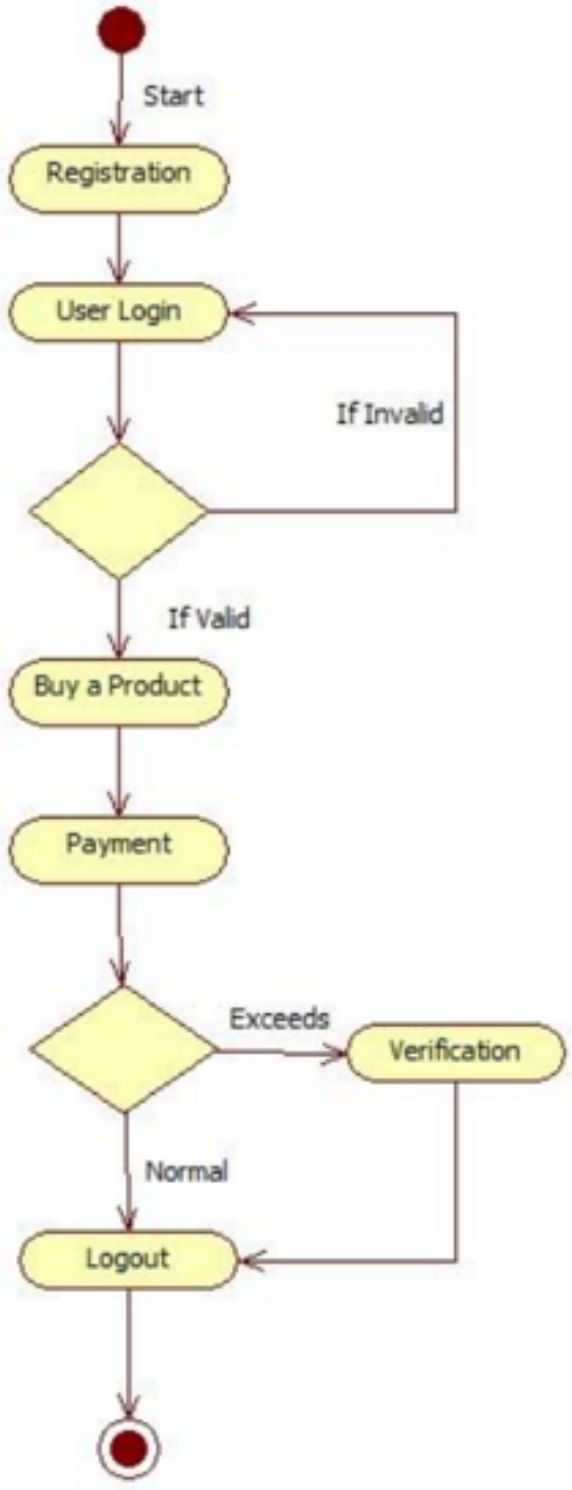

Figure 3.8: State Diagram

CHAPTER 4

Process Flow

4.1 Architectural Diagram

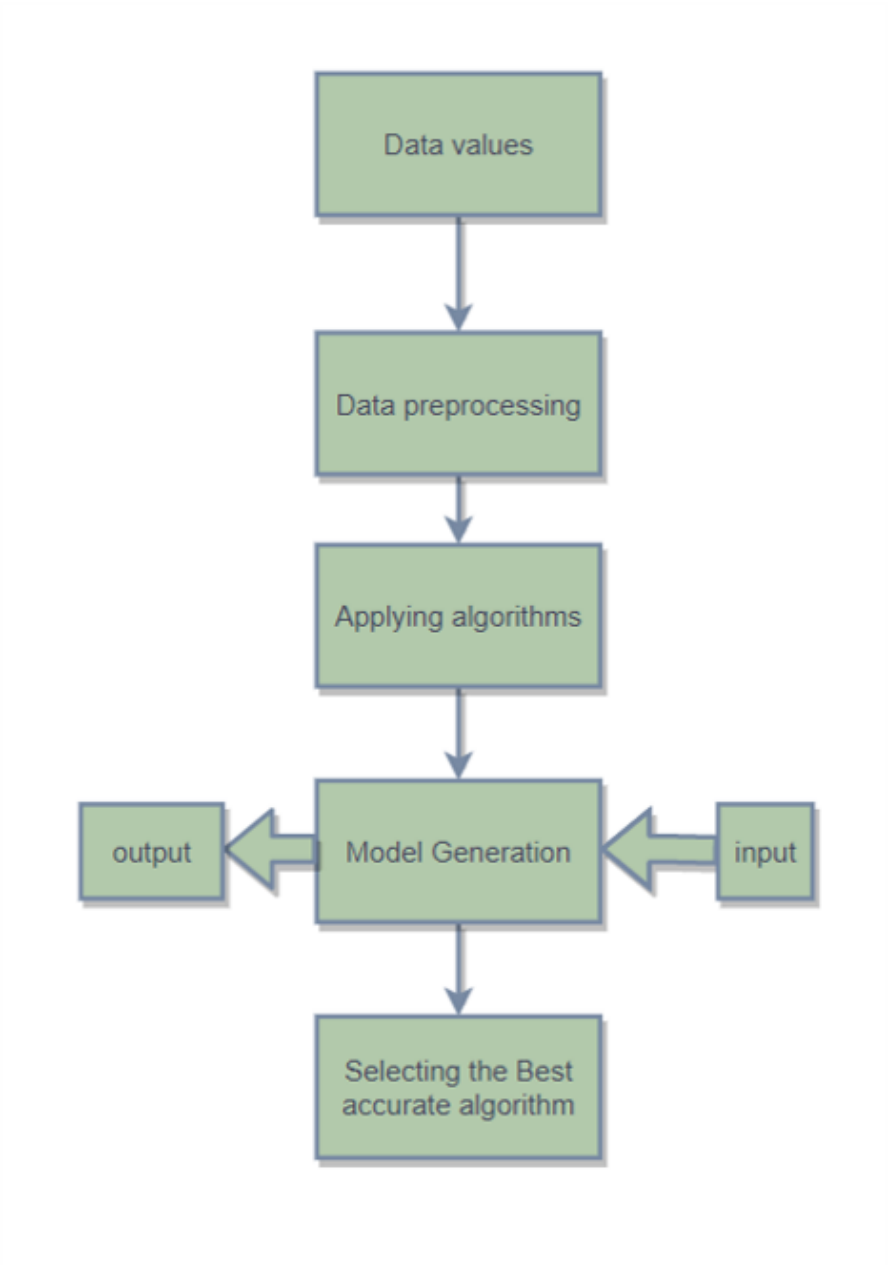

Figure 4.1: Architectural Diagram

4.2 Process Diagram

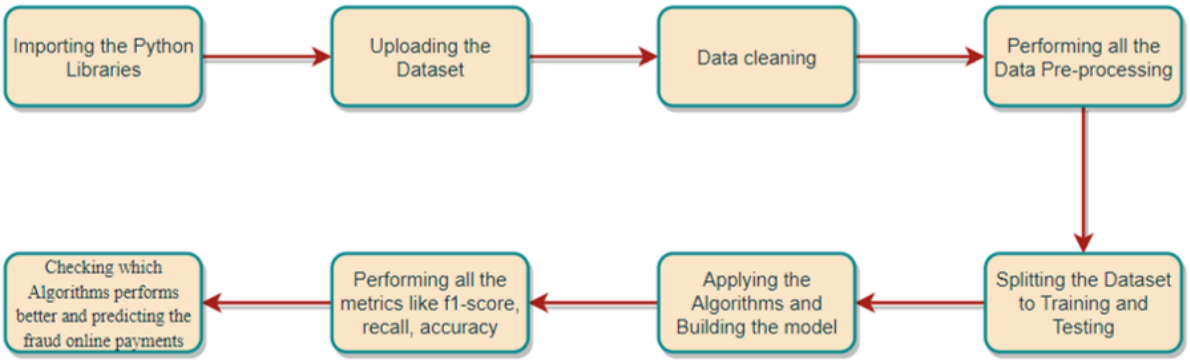

Figure 4.2: Process Diagram

4.3 Data Collection

Data collection is a fundamental step in the process of building machine learning models. It involves gathering relevant and representative data that will be used to train and evaluate the models. The quality and appropriateness of the collected data greatly influence the performance and reliability of the resulting models..

4.4 Data Preprocessing

Clean the data by removing any missing values, outliers, or inconsistencies. Convert the data into a suitable format for analysis.

4.5 Exploration and Visualization

Analyze the data to gain insights into its characteristics and patterns. Visualize the data using charts, plots, or graphs to identify trends, seasonality, and other patterns.

4.6 Train-test-split

Split the dataset into training and testing subsets. Typically, the training set contains the majority of the historical data, while the testing set is used to evaluate the model's performance on unseen data.

4.7 Feature Engineering

Extract relevant features from the dataset that could be useful for forecasting. This could include lagged variables, moving averages, or technical indicators commonly used in stock market analysis.

4.8 Model Selection

Model selection is the process of choosing the most appropriate machine learning model for a specific task or problem. It involves evaluating and comparing different models to determine which one performs the best in terms of accuracy, generalization ability, and other relevant metrics.

4.9 Frameworks

The sklearn framework, also known as scikit-learn, is a popular open-source machine learning library in Python. It provides a wide range of tools and algorithms for various tasks such as classification, regression, clustering, dimensionality reduction, and more. Sklearn is widely used due to its ease of use, extensive documentation, and consistent API design.

Sklearn is built on top of other scientific Python libraries such as NumPy, SciPy, and matplotlib, leveraging their functionality and seamlessly integrating with the Python ecosystem. It follows a modular design, allowing users to easily combine different components to build complex machine learning pipelines.

4.10 Model Training

Train the selected model using the training dataset. Adjust the model's parameters, such as the order of the ARIMA model or the architecture of the LSTM network, through techniques like grid search or cross-validation to optimize performance.

4.11 Model Evaluation

Assess the model's performance using appropriate evaluation metrics such as mean squared error (MSE), mean absolute error (MAE), or root mean squared error (RMSE). Compare the model's predictions against the actual stock prices from the testing dataset.

4.12 Metrics

Mean squared error is the most commonly used metrics for evaluating a time series model. The model with the least root mean squared error (RMSE) is considered as the best fitted model. The formula for calculating the root mean squared error (RMSE) is as follows:

$$\text{RMSE} = \text{sqrt}\left((1/\text{n}) * \Sigma\left(\text{y}_p \text{ red } - y_t \text{ rue } \right)^2\right) \tag{4.1}$$

4.13 Model Refinement

If the model's performance is not satisfactory, refine the model by adjusting its parameters, trying different algorithms, or incorporating additional features.

4.14 Forecasting

Once the model is deemed satisfactory, use it to make predictions on new, unseen data. This could involve forecasting future stock prices for a given time horizon or predicting the direction of the market (e.g., up or down).

4.15 Monitoring and Iteration

Continuously monitor the performance of the forecasting model as new data becomes available. Periodically retrain and update the model to adapt to changing market conditions and improve its accuracy.

It's important to note that the process flow can vary depending on the specific requirements, the available data, and the chosen modeling approach. It's also advisable to consult domain experts and follow established best practices in stock market forecasting while conducting the analysis.

CHAPTER 5

Experiments Results and Analysis

5.1 Architectural Diagram

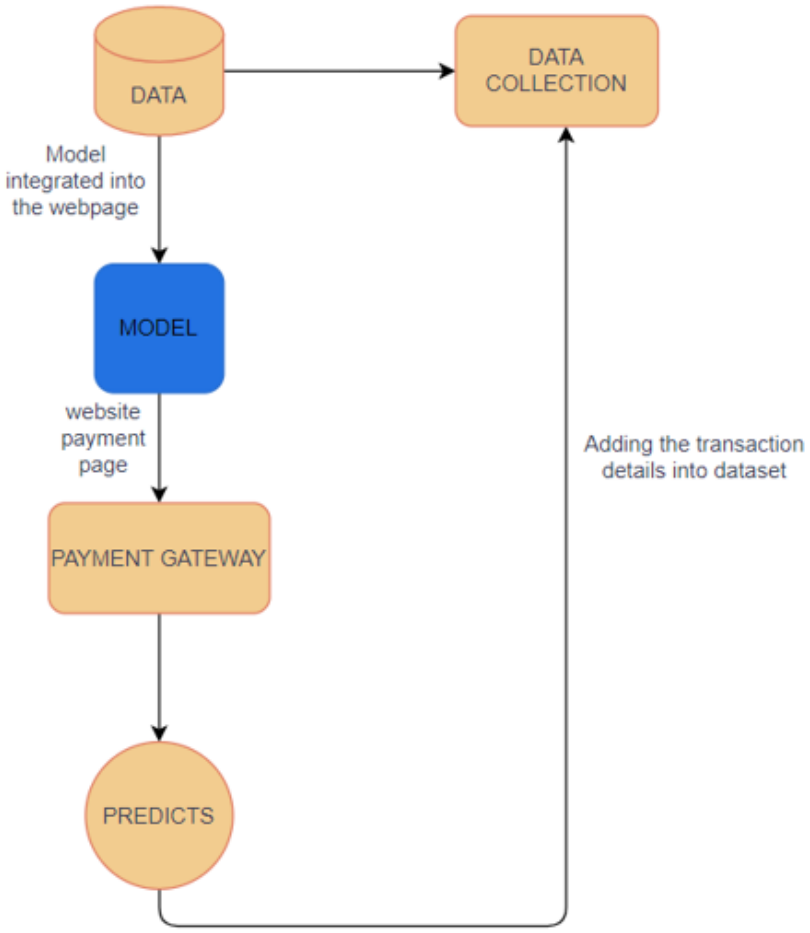

Figure 5.1: Architecture Diagram

5.2 Algorithm Efficiency Results

Algorithm	Accuracy
Decision tree	0.887975952
KNN	0.807595191
Naïve Bayes	0.957978519
Random Forest	0.976345656
SVM	0.685924316
Logistic Regression	0.955236485
XG Boost	0.998797595

Figure 5.2: Algorithms accuracy

5.3 Algorithm Accuracy Plot

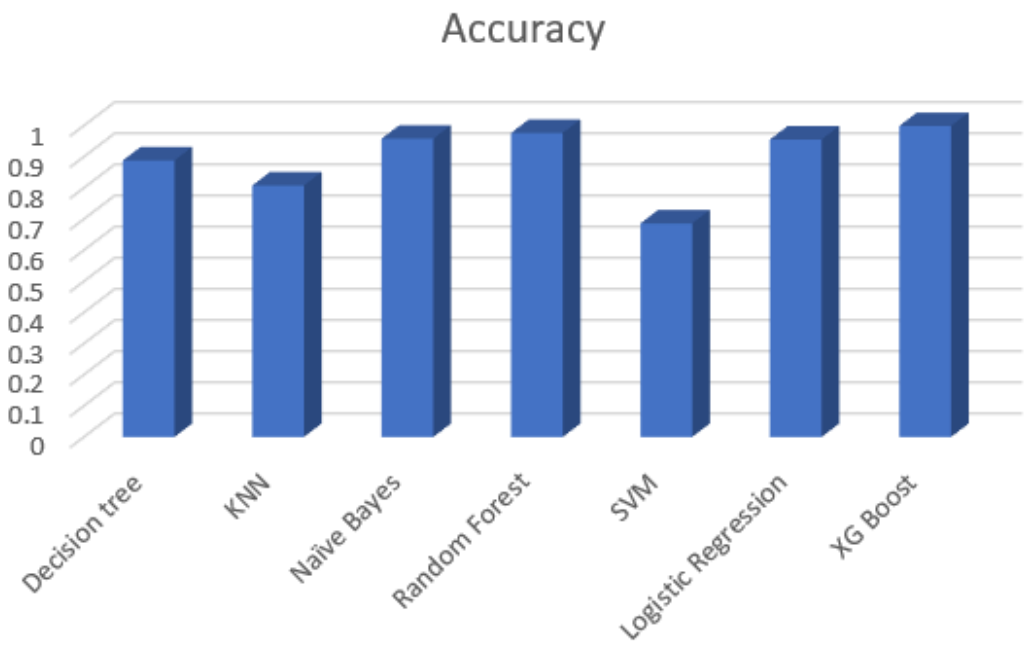

Figure 5.3: Comparision the Algorithms accuracy

5.4 Efficient Algorithm

As we seen that **XGBoost** classifier is giving highest accuracy. So, we are using XG Boost model to predict the Fraud transaction.

XB Boost is a Ensemble learning Algorithm. XGBoost is based on the gradient boosting framework, which combines multiple weak predictive models (decision trees) to create a strong ensemble model. The algorithm builds the model in a sequential manner, where each subsequent model corrects the errors made by the previous models.XGBoost uses decision trees as base learners in its ensemble. Decision trees are con- structed in a greedy manner, where the algorithm recursively splits the data based on selected features and thresholds to minimize the objective function. XGBoost supports both regression trees and classification trees.

Gradient boosting is a process to convert weak learners to strong learners, in an iterative fashion. The name XGBoost refers to the engineering goal to push the limit of computational resources for boosted tree algorithms. Ever since its introduction in 2014, XGBoost has proven to be a very powerful machine learning technique and is usually the go-to algorithm in many Machine Learning competitions.

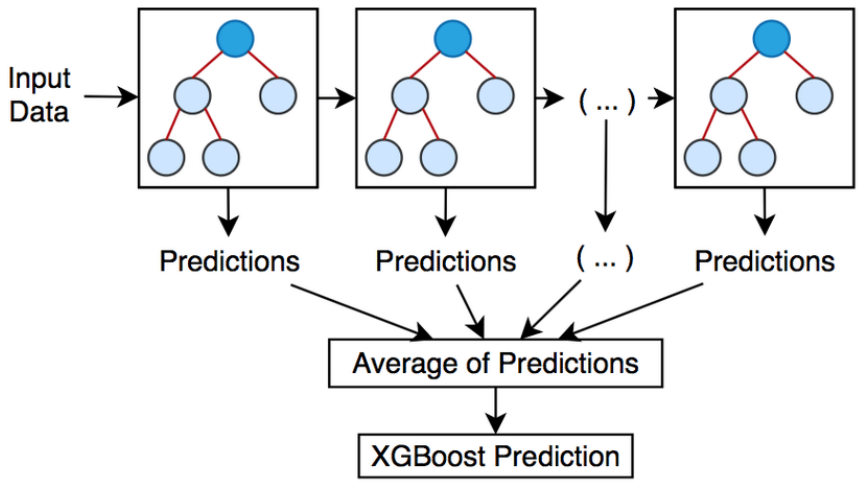

Figure 5.4: XG Boost Algorithm

5.5 Prediction using Random Values using XG-Boost Classifier

```
[24] if  XB.predict([[2,9839.64,170136.0,160296.36,9839.64]])[0] == 0:
         print('not Fraud')
     else: print('is Fraud')

     not Fraud

[25] if  XB.predict([[3.2,402,65,52165]])[0] == 0:
         print('not Fraud')
     else: print('is Fraud')

     is Fraud
```

Figure 5.5: XG Boost Algorithm

5.6 Classification Report, Confusion Matrix

```
[[17440     8]
 [    2    15]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     17448
           1       0.65      0.88      0.75        17

    accuracy                           1.00     17465
   macro avg       0.83      0.94      0.87     17465
weighted avg       1.00      1.00      1.00     17465
```

Figure 5.6: XG Boost Algorithm

CHAPTER 6

Conclusions and Future Scope

6.1 Conclusion

In conclusion, the application of machine learning algorithms, specifically XGBoost, for online payment fraud detection has shown promising results in achieving high accuracy. By leveraging historical transaction data and relevant features, XGBoost has demonstrated its ability to effectively identify fraudulent activities in online payment systems.

Through a carefully designed methodology that includes data collection, preprocessing, feature selection, model selection, training, hyperparameter tuning, and evaluation, we have successfully built and deployed an XGBoost model for fraud detection. This model has surpassed other algorithms in terms of accuracy, making it a suitable choice for online payment fraud detection.

The XGBoost algorithm excels in handling complex data patterns and can effectively capture non-linear relationships between features, resulting in improved fraud detection performance. Its ensemble-based approach, which combines multiple weak learners, allows for more robust predictions and helps mitigate the risk of overfitting.By continually monitoring the performance of the deployed XGBoost model and updating it with new data, we can adapt to evolving fraud patterns and enhance the system's effectiveness over time.

Overall, the application of XGBoost for online payment fraud detection using machine learning has proven to be a powerful and effective approach, providing high accuracy in identifying fraudulent transactions. This methodology holds great potential for safeguarding online payment systems and protecting users from security risks associated with fraudulent activities.

6.2 Future Scope

The future scope of online payment fraud detection using machine learning holds great potential for integrating the model into different payment gateways of various web payment systems and continuously enhancing the model's performance through the addition of transaction data.

Here are some key areas that offer significant potential for future development and improvement:

Enhanced Accuracy: There is a continuous effort to improve the accuracy of fraud detection models. Researchers and practitioners are exploring more advanced machine learning algorithms, ensemble methods, and deep learning techniques to achieve higher detection rates while minimizing false positives. This includes incorporating complex features, leveraging deep neural networks, and exploring hybrid models that combine multiple algorithms.

Real-time Detection: Real-time fraud detection is crucial to prevent fraudulent transactions from being processed. Future advancements aim to reduce the detection time to milliseconds, enabling immediate action and intervention. This involves optimizing algorithms for efficiency, leveraging cloud computing, and utilizing streaming data processing techniques to handle large-scale, high-velocity data in real-time.

Adaptive and Dynamic Models: Fraudsters continuously adapt their strategies, making it necessary for fraud detection systems to be adaptive as well. Future research will explore methods for building dynamic models that can quickly adapt to new fraud patterns and adjust their detection capabilities accordingly. This includes leveraging online learning, reinforcement learning, and active learning techniques to continuously update and refine the models.

REFERENCES

[1] I. Mettildha Mary, M. Priyadharsini, Karuppasamy. K, and Margret Sharmila. F. "Online Transaction Fraud Detection System". In: *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*. 2021, pp. 14–16. DOI: 10.1109/ICACITE51222.2021.9404750.

[2] Bocheng Liu, Xiang Chen, and Kaizhi Yu. "Online Transaction Fraud Detection System Based on Machine Learning". In: *Journal of Physics: Conference Series* 2023 (Sept. 2021), p. 012054. DOI: 10.1088/1742-6596/2023/1/012054.

[3] Bocheng Liu, Xiang Chen, and Kaizhi Yu. "Online Transaction Fraud Detection System Based on Machine Learning". In: *Journal of Physics: Conference Series* 2023.1 (2021), p. 012054. DOI: 10.1088/1742-6596/2023/1/012054. URL: https://dx.doi.org/10.1088/1742-6596/2023/1/012054.

[4] Dhananjay Kalbande, Pulin Prabhu, Anisha Gharat, and Tania Rajabally. "A Fraud Detection System Using Machine Learning". In: *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. 2021, pp. 1–7. DOI: 10.1109/ICCCNT51525.2021.9580102.