COMPARATIVE STUDY AND ANALYSIS OF CREDIT CARD FRAUD DETECTION WITH MACHINE LEARNING TECHNIQUES

A Project Report Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF COMPUTER APPLICATION

by

Akash Agarwal
(Roll. No. 2110143325030)

Shivendra Tripathi
(Roll. No. 2110014325024)

Muhammad Abdullah
(Roll. No. 2110014325010)

Under the Guidance of
Er. Shailendra Kumar Sonkar
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING AND TECHNOLOGY
UNIVERSITY OF LUCKNOW, LUCKNOW
June - 2023

DECLARATION

We compared and analysed various machine learning algorithms to assess their performance and suitability for specific tasks. Our project aimed to identify strengths and weaknesses, providing valuable insights into algorithm applications. We followed ethical guidelines, obtained necessary permissions, and conducted meticulous analysis. We express gratitude to Er. Shailendra Kumar Sonkar for guidance and support. Our work is accurate, authentic, and independent, with no academic misconduct. We appreciate your assessment of our project.

Akash Agarwal
(Roll No. 2110143325030)
Date:

Shivendra Tripathi
(Roll. No. 2110014325024)
Date:

Muhammad Abdullah
(Roll. No. 2110014325010)
Date:

ABSTRACT

This research paper presents a comparative study and analysis of credit card fraud detection using various machine learning techniques. The objective is to evaluate the performance of different algorithms in detecting fraudulent transactions and to identify the most effective approach.

The study utilises the "Online Payments Fraud Detection Dataset," which consists of 6,362,620 instances. The dataset includes information such as the transaction type, amount, old balance of the origin account, new balance of the origin account, old balance of the recipient account, new balance of the recipient account, and a binary indicator denoting fraudulent transactions. The dataset is highly imbalanced, with 6,354,407 non-fraudulent instances (0's) and 8,213 fraudulent instances (1's). To address the class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) is employed.

The machine learning algorithms employed in this study include Gradient Boosting, XGClassifier, Logistic Regression, Random Forest, Decision Tree, and K-Nearest Neighbors (KNN). Performance evaluation metrics such as accuracy, precision, F1 score and Matthews Correlation Coefficient (MCC) are used to assess the algorithms' performance and compare their effectiveness in detecting credit card fraud.

ACKNOWLEDGEMENT

We would like to express my heartfelt gratitude to Er. Shailendra Kumar Sonkar, our mentor, for his invaluable guidance, support, and expertise throughout the completion of our final year project titled **" COMPARATIVE STUDY AND ANALYSIS OF CREDIT CARD FRAUD DETECTION WITH MACHINE LEARNING TECHNIQUES". He** has extensive knowledge in the field of machine learning and has been instrumental in shaping the direction of our research and refining our understanding of various ML algorithms. His insightful suggestions and constructive feedback have significantly enhanced the quality of my project.

We are grateful for the time and effort dedicated to reviewing our work, providing valuable insights, and assisting us in navigating through complex concepts and methodologies. His mentorship has not only expanded our technical skills but also nurtured our critical thinking abilities, allowing us to approach challenges with confidence. His enthusiasm for the topic has been contagious, inspiring us to explore further and push the boundaries of our research.

Thank you, Er. Shailendra Kumar Sonkar, for being an exceptional mentor and for your invaluable contributions to our final year project.

Akash Agarwal

Shivendra Tripathi

Muhammad Abdullah

CERTIFICATE

Certified that Akash Agarwal (Roll No.: 2110143325030), Shivendra Tripathi (Roll No.: 2110014325024) and Muhammad Abdullah (Roll No: 2110014332510) have carried out the project work presented in this project report entitled "**COMPARATIVE STUDY AND ANALYSIS OF CREDIT CARD FRAUD DETECTION WITH MACHINE LEARNING TECHNIQUES"** for the award of degree of **MASTER OF COMPUTER APPLICATION** from **Faculty of Engineering and Technology, University of Lucknow, Lucknow** under my guidance. The project report embodies results of original work, and studies are carried out by the students themself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Supervisor: -

Er. Shailendra Kumar Sonkar
Assistant Professor
Department of Computer Science and Engineering
Faculty of Engineering and Technology
University of Lucknow, Lucknow

Table of Contents

List of Tables

List of Figures

List of Symbols and Abbreviations

ANN	Artificial Neural Network
FN	False Negatives
FP	False Positives
KNN	K-Nearest Neighbour
MCC	Matthews Correlation Coefficient
ML	Machine Learning
NB	Naive Bayes
RandomF	Random Forest Classifier
SMOTE	Synthetic Minority Over-Sampling Technique
TN	True Negatives
TP	True Positives
UndSamp	Under Sampling
XGBoost	Extreme Gradient Boosting

CHAPTER 1

INTRODUCTION

Credit card fraud has become a significant concern in the digital era, posing financial risks to individuals, merchants, and financial institutions. Detecting fraudulent transactions accurately and efficiently is of paramount importance to mitigate such risks. Traditional rule-based methods and statistical techniques have limitations in effectively identifying fraudulent activities in real-time. As a result, the use of machine learning algorithms has gained prominence in credit card fraud detection.

This research aims to compare and analyse various machine learning techniques for credit card fraud detection, considering their effectiveness in accurately classifying fraudulent transactions. The "Online Payments Fraud Detection Dataset" is utilised for this study, which comprises 6,362,620 instances. The dataset includes features such as the transaction type, amount, old balance of the origin account, new balance of the origin account, old balance of the recipient account, new balance of the recipient account, and a binary indicator denoting fraudulent transactions. The dataset is highly imbalanced, with 6,354,407 instances labelled as non-fraudulent (0) and 8,213 instances labelled as fraudulent (1).

To address the class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) is employed. SMOTE generates synthetic samples of the minority class (fraudulent transactions) to create a balanced dataset for training the machine learning models. This technique allows for a more robust and accurate analysis of the algorithms' performance in detecting credit card fraud.

The machine learning algorithms used in this study include Gradient Boosting, XGClassifier, Logistic Regression, Random Forest, Decision Tree, and K-Nearest Neighbors (KNN). These algorithms are evaluated based on performance evaluation metrics such as accuracy, precision, F1 score and Matthews Correlation Coefficient (MCC). By comparing the performance of these algorithms, the study aims to identify the most effective approach for credit card fraud detection.

The finding of this research will contribute to the existing body of knowledge in credit card fraud detection and assist in improving the accuracy and efficiency of fraud detection systems. Furthermore, the study will aid in minimising false positives and false negatives, enhancing the overall security of online payment transactions, and reducing financial risks associated with credit card fraud.

1.1 Machine Learning

Machine Learning is a subfield of artificial intelligence that focuses on developing algorithms and statistical models that enable computers to learn from data and make predictions or decisions without being explicitly programmed. It utilises mathematical and statistical techniques to extract patterns and insights from large datasets, allowing machines to improve their performance through experience. Regression and classification are two fundamental tasks in supervised machine learning. Regression involves predicting a continuous output variable based on input features, while classification involves assigning categorical labels to input instances. Both techniques have broad applications in various fields, including finance, healthcare, marketing, and robotics.

To train both regression and classification models, a dataset is divided into a training set and a test set. The training set is used to optimise the model's parameters, while the test set is employed to evaluate its performance on unseen data. Performance metrics such as accuracy, precision, recall, and mean squared error are commonly used to assess the quality of the models.

Unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Instead, models itself find the hidden patterns and insights from the given data. It can be compared to learning which takes place in the human brain while learning new things.

1.2 Models Used

1.2.1 Gradient Boosting

Gradient Boosting is a powerful machine learning algorithm that combines weak predictive models, typically decision trees, in a sequential manner to create a strong and accurate ensemble model. It starts with an initial model and then iteratively builds additional models to correct the errors made by the previous ones. Each new model focuses on learning the residuals or differences between the predicted and actual values from the previous models. The algorithm uses gradient descent optimization to update the weights of the training instances, placing more emphasis on instances with larger errors. By continuously learning from the mistakes and adjusting the weights, Gradient Boosting gradually improves the ensemble's performance, reducing the overall error and achieving higher accuracy. This iterative process allows the algorithm to effectively handle complex relationships in the data and capture nonlinear patterns. Gradient Boosting has become widely used in various domains due to its ability to produce robust and accurate predictions.

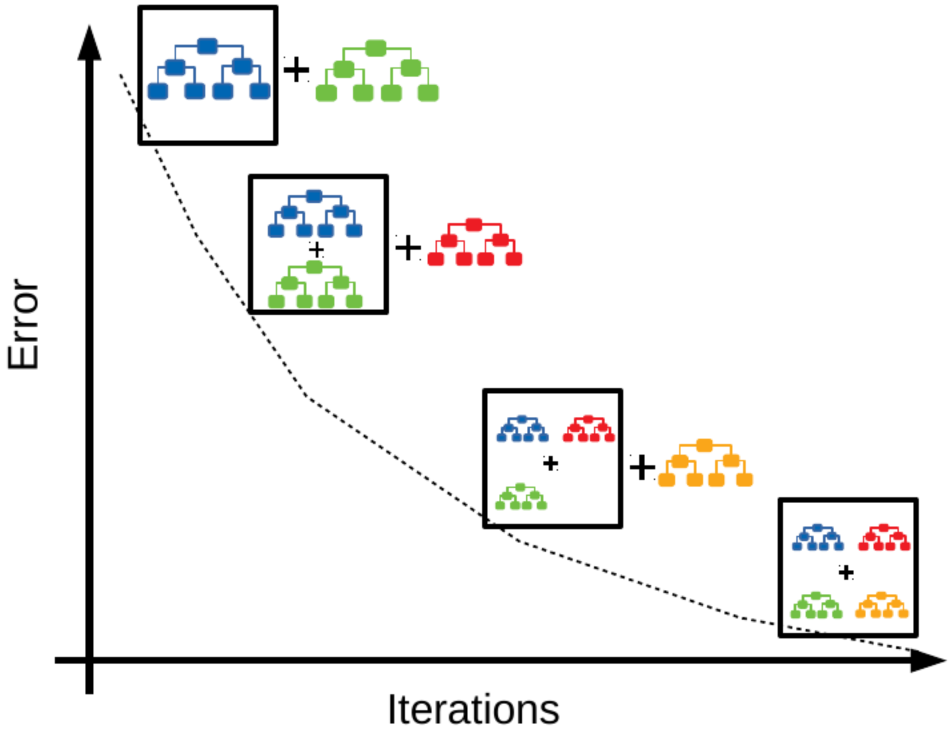

Figure 1.1 Gradient Boosting

1.2.2 XGBoost Classifier

XGBoost Classifier, also known as Extreme Gradient Boosting Classifier, is a machine learning algorithm that belongs to the gradient boosting family. It is specifically designed for classification tasks and is highly regarded for its accuracy and performance. XGBoost Classifier combines the principles of gradient boosting and decision trees to create a powerful ensemble model. It iteratively adds decision trees to the ensemble, with each tree correcting the errors made by the previous ones. The algorithm optimises an objective function by minimising a specified loss metric. XGBoost Classifier incorporates regularisation techniques to prevent overfitting and handles missing values efficiently. It provides feature importance measures and is known for its speed and scalability. Due to its impressive performance, XGBoost Classifier is widely used in various domains for classification tasks.

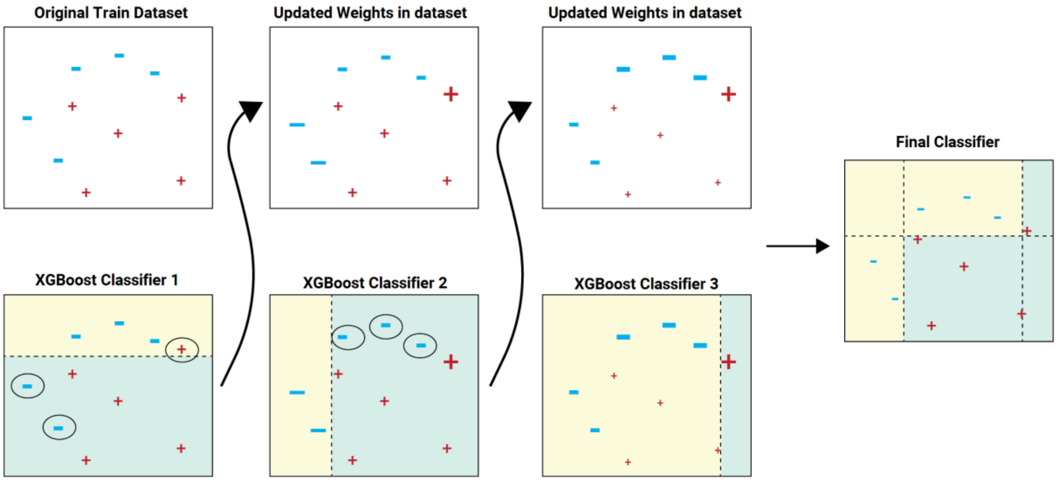

Figure 1.2 XGBoost Classifier

1.2.3 Logistic Regression

Logistic Regression is a widely used machine learning algorithm for binary classification tasks. It models the relationship between the input variables and the probability of the binary outcome. Unlike linear regression, Logistic Regression uses the logistic function (also known as the sigmoid function) to map the output to a probability between 0 and 1. The algorithm estimates the parameters of the logistic function by maximising the likelihood of the observed data. During training, it learns the weights assigned to each input feature, representing their influence on the probability of the positive class. To make predictions, Logistic Regression calculates the probability of the positive class based on the input features and applies a threshold to determine the final class label. Logistic Regression is known for its simplicity, interpretability, and efficiency. It can handle both numerical and categorical features, and it is less prone to over fitting compared to more complex models. However, it assumes a linear relationship between the features and the log-odds of the outcome and may struggle with nonlinear relationships. Nonetheless, Logistic Regression remains a fundamental algorithm in machine learning, widely used in various domains such as finance, healthcare, and marketing.

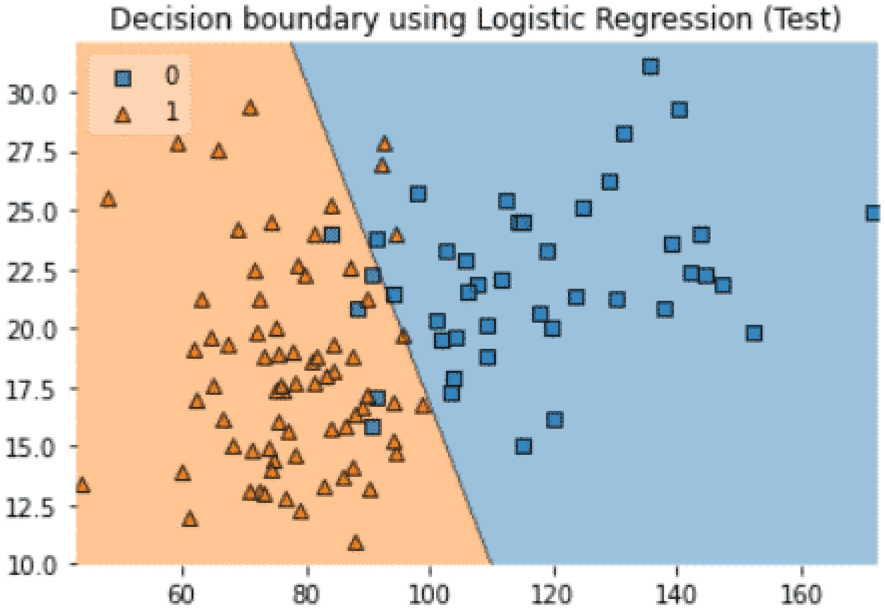

Figure 1.3 Logistic Regression

1.2.4 Decision Tree

Decision Tree algorithm is a popular and widely used machine learning algorithm for both classification and regression tasks. It builds a tree-like structure based on the provided training data, where each internal node represents a decision based on a feature, and each leaf node represents a class label or a numerical value for regression. The algorithm recursively splits the data based on different features, aiming to maximise information gain or minimise impurity at each node. Decision Trees are known for their interpretability, as the decision-making process can be easily understood by following the path from the root to the leaf nodes. They can handle both categorical and numerical features, automatically handle missing values, and capture complex relationships between features and target variables. However, Decision Trees are prone to over fitting, and techniques like pruning and ensemble methods can be applied to mitigate this issue. Despite their limitations, Decision Trees are widely used due to their simplicity, interpretability, and effectiveness in various domains.

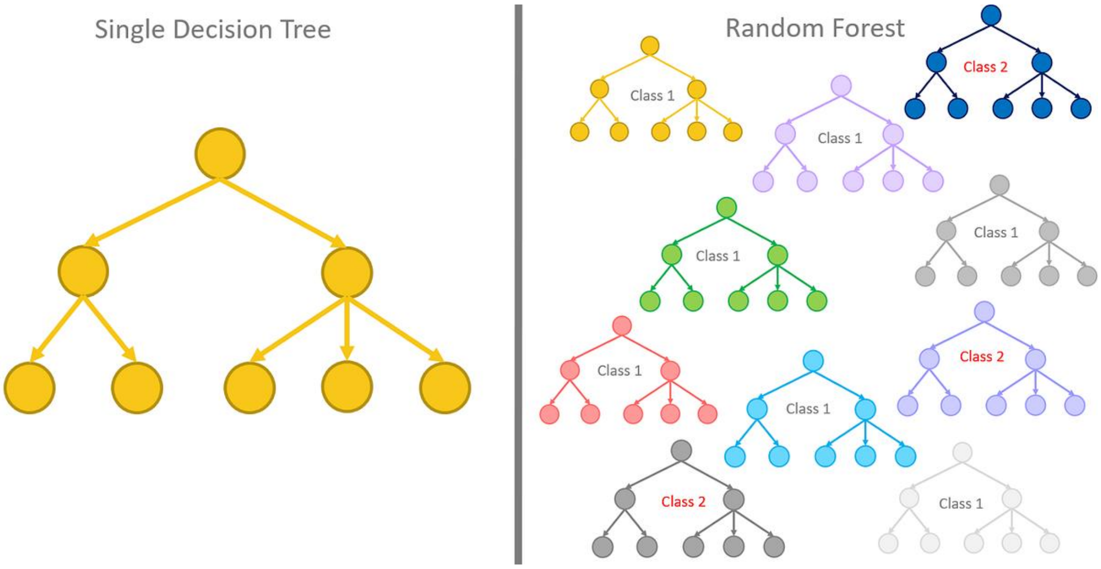

Figure 1.4 Decision Tree and Random Forest Classifier

1.2.5 Random Forest

Random Forest is a powerful and widely used machine learning algorithm that combines the principles of decision trees and ensemble learning. It forms an ensemble of decision trees by training them on random subsets of the training data. This random sampling introduces diversity and helps reduce over fitting. Moreover, Random Forest also introduces randomness in feature selection by considering only a subset of features at each node of the decision tree. The algorithm introduces randomness in two ways: random sampling of data and random feature selection. The final prediction is made by aggregating the predictions of all the trees through majority voting for classification tasks or averaging for regression tasks. Random Forest offers several advantages, including high accuracy, robustness to outliers and noisy data, and the ability to handle high-dimensional feature spaces. It is also resistant to over fitting and provides estimates of feature importance. While it may be less interpretable than individual decision trees, Random Forest's predictive performance and versatility make it a popular choice in various domains.

1.2.6 K-Nearest Neighbours

KNN is a non-parametric and instance-based learning algorithm used for classification and regression tasks. During the training phase, the algorithm stores the entire training dataset in memory. When making predictions for new data points, KNN finds the k nearest neighbours based on a chosen distance metric, such as Euclidean or Manhattan distance. The majority class among the neighbours determines the predicted class label for classification, while the average of the target variable values provides the prediction for regression. The value of k, representing the number of neighbours to consider, is an important parameter that affects the algorithm's performance. KNN is relatively simple to implement and can adapt to different data types, but it can be computationally expensive and sensitive to the choice of distance metric and scaling of features.

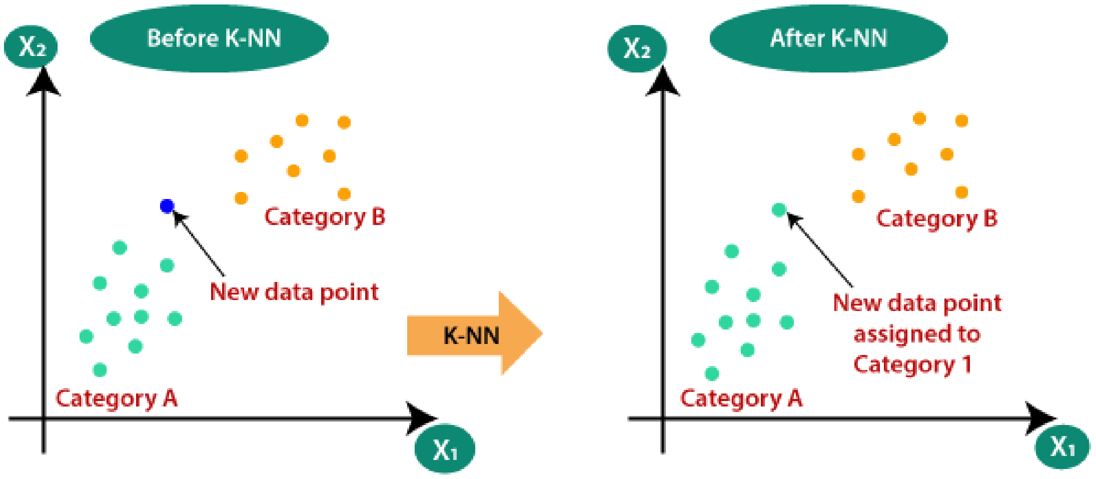

Figure 1.5 K- Nearest Neighbour

CHAPTER 2

LITERATURE REVIEW

Numerous studies have discussed fraudulent transactions. The study in [1] stated that Credit card frauds are easy and friendly targets. E-Commerce and many other online sites have increased the online payment modes, increasing the risk for online frauds. They focus on to design and develop a novel fraud detection method for Streaming Transaction Data, with an objective, to analyse the past transaction details of the customers and extract the behavioural patterns. They used a sliding window strategy to aggregate the transaction made by the cardholders from different groups so that the behavioural pattern of the groups can be extracted respectively. Then they used SMOTE operation on the data set for pre-processing. Later different classifiers like Logistic Regression, Decision Tree, Random Forest, Local Outlier Factor & Isolation Forest are trained over the groups separately. And then the classifier with better rating score can be chosen to be one of the best methods to predict frauds. Then they finally observed that Logistic regression, decision tree and random forest are the algorithms that gave better results.

The authors in [2] faced problems such as Class imbalance & transaction pattern. Their main aim is to detect anomalous activities, called Outliers. For that they have taken a Dataset from Kaggle consisting of 31 columns out of which 28 are named as v1-v28 to protect sensitive data. For Checking they have used Histogram. They majorly focused on only two models Local Outlier Factor & Isolation Forest then the algorithm does reach over 99.6% accuracy, the precision comes up to 33%.

In the study [3] problem faced was Data Mining Classification. For that they have conducted comparative analyses of identification of fraudulent activity using various models. The dataset presents 3075 transactions with 12 features of transactions in the CSV file. Support Vector Machine, K-nearest Neighbour, Naïve Bayes & Logistics Regression used as models. Their accuracy is 97.53%, 96.91%, 95.98% & 99.07% respectively. Hence, Logistics Regression was the most accurate.

In the study [4] the problem faced was an extremely imbalanced database. Their main aim is to identify systems with the feedback system. For that dataset it introduces transactions that occurred in two days, composed for 284,807 transactions, the number of 492 transactions being such a fraud. The data set has been divided into two categories of 70 % for training and 30 % for testing. Accuracy of different models on which data was trained are Random Forest is 95.98%, Support

Vector Machine is 93.22%, Logistics Regression is 92.89%, Naïve Bayes is 91.2%, Decision Tree is 90.9% and Gradient Boosting classifier is 93.99%. At last Random Forest is showing the best result.

In the study [5] the problem faced was lack of highly imbalanced & good dataset. Their aim is to predict legitimate or fraud transactions. Our model is BiLSTM - MaxPooling BiGRU - MaxPooling which is based on bidirectional long short-term memory (BiLSTM) and bidirectional Gated recurrent unit (BiGRU). The data set collected was containing four files, train transaction with 394 columns, train identity with 41 columns, test transaction with 393 columns, test identity with 41 columns. The identity and transaction files were merged based on the transaction id feature; resulting in 433 features in the dataset and 590540 instances. Pre-Processing was done through Random Under-Sampling, Random Over-Sampling & SMOTE. After that various classifier like Naïve base, Voting, Random Forest, Logistic Regression (LR), Decision Tree and Ada boosting were used. Their model with AUC (Area under the receiver operating characteristic curve) of 91.37% achieved.

In the study [6] the problem faced was a highly imbalanced dataset. Their aim is to propose a machine learning (ML) based credit card fraud detection engine using the genetic algorithm (GA) for feature selection. Data set is Credit card transactions made by European cardholders for 2 days in Sept 2013, containing 284807 transactions in which 0.172% was fraudulent. SMOTE was used for data Pre-Processing. Techniques used are Random Forest, Decision Tree, ANN approach is used as the fitness method inside the Genetic Algorithm. The Genetic Algorithm selected attributes demonstrated that the Genetic Algorithm-Random Forest achieved optimal accuracy of 99.98%, the classifier Genetic Algorithm-Decision Tree achieved accuracy of 99.92% on our dataset.

In the study [7] they compare different machine learning algorithms to effectively and efficiently predict the legitimacy of financial transactions. Dataset was collected from Kaggle which consists of 6362620 rows and 10 columns. SMOTE is used for Pre-Processing of Data. Different Classifiers like Decision tree classifier, Random Forest, MLP Regressor, Compliment NB, MLP Classifier, Gaussian NB, Bernoulli NB, LGBM classifier, AdaBoost Classifier, K Neighbours Classifier, Logistic Regression, Bagging Classifier & Deep Learning are used. For that they have Accuracy, Precision, Recall & F1_score is used. Best Classifier with an unbalanced dataset was the Random Forest Classifier. However, the best classifier with a balanced dataset was the Bagging Classifier.

In the study [8] the problem faced is to handle categorical data and unbalanced datasets. Their aim is to use the CatBoost algorithm for detecting credit card frauds. A sizable dataset of credit card transactions, including details such as transaction amount, time, and location, was used to train the model. Resampling is used to handle data imbalance. They have used CatBoost model on the dataset and for model's efficiency, precision, recall, and F1-score are used. Hence, they stated that CatBoost is the most efficient model for detecting Credit card frauds.

CHAPTER 3

METHODOLOGY

3.1 Dataset

The "Online Payments Fraud Detection Dataset" is a dataset obtained from Kaggle, specifically designed for studying and analysing online payment fraud. This dataset contains a comprehensive set of attributes that capture key transactional details related to online payments. It includes features such as the transaction type, amount, customer and recipient information, as well as the initial and updated balances of the accounts involved in the transaction. The dataset comprises a total of 6,362,620 instances, with 6,354,407 instances labelled as non-fraudulent (0's) and 8,213 instances labelled as fraudulent (1's). With its large size and balanced representation of fraud and non-fraud cases, this dataset provides an ideal foundation for investigating and developing effective machine learning approaches for credit card fraud detection.

3.2 Data Pre-processing:

The "Online Payments Fraud Detection Dataset" underwent several pre-processing steps to ensure its suitability for analysis and model development. The dataset contained 6,362,620 instances with 6,354,407 instances labelled as non-fraudulent (0's) and 8,213 instances labelled as fraudulent (1's).

The following pre-processing steps were performed:

3.2.1 Handling Missing Values

No missing values were found in the dataset, eliminating the need for imputation techniques.

Features	Description
step	Maps a unit of time in the real world. In this case 1 step is 1 hour of time. Total steps 744 (30 days simulation).
type	CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER
amount	Amount of the transaction in local currency.
nameOrig	customer who started the transaction
oldbalanceOrg	initial balance before the transaction
newbalanceOrig	new balance after the transaction
nameDest	customer who is the recipient of the transaction
oldbalanceDest	Initial balance recipient before the transaction. Note that there is not information for customers that start with M (Merchants).
newbalanceDest	New balance recipient after the transaction. Note that there is not information for customers that start with M (Merchants).
isFraud	This is the transactions made by the fraudulent agents inside the simulation. In this specific dataset the fraudulent behavior of the agents aims to profit by taking control or customers accounts and try to empty the funds by transferring to another account and then cashing out of the system.

Table 1.1 Dataset Feature

3.2.2 Feature Selection

No feature selection was performed, and all the original attributes were retained for analysis.

3.3.3 Feature Removal

Two columns, "nameOrig" and "nameDest," were dropped from the dataset as they were deemed irrelevant to the fraud detection task.

3.3.4 Mapping Transaction Types

The "type" column, which represented the type of transaction, was transformed into numeric values to facilitate further processing. The mapping used was as follows:

CASH_IN:	1
CASH_OUT:	2
PAYMENT:	3
DEBIT:	4
TRANSFER:	5

3.3.5 Handling Class Imbalance

The dataset exhibited class imbalance, with significantly more non-fraudulent instances than fraudulent ones. To address this issue, the Synthetic Minority Over-sampling Technique (SMOTE) was employed to generate synthetic instances of the minority class, thereby balancing the class distribution.

3.3.6 Data Split

The dataset was divided into an 80% training set and a 20% testing set. This partitioning allowed for model training on the majority of the data while reserving a separate portion for evaluating the model's performance.

By performing these pre-processing steps, the "Online Payments Fraud Detection Dataset" was prepared for subsequent analysis and model development.

3.4 Data Imbalance Handling:

The "Online Payments Fraud Detection Dataset" exhibited a significant class imbalance, with a much larger number of non-fraudulent instances compared to fraudulent instances. This imbalance can pose challenges during model training and evaluation, as the models may become biased towards the majority class and perform poorly on the minority class.

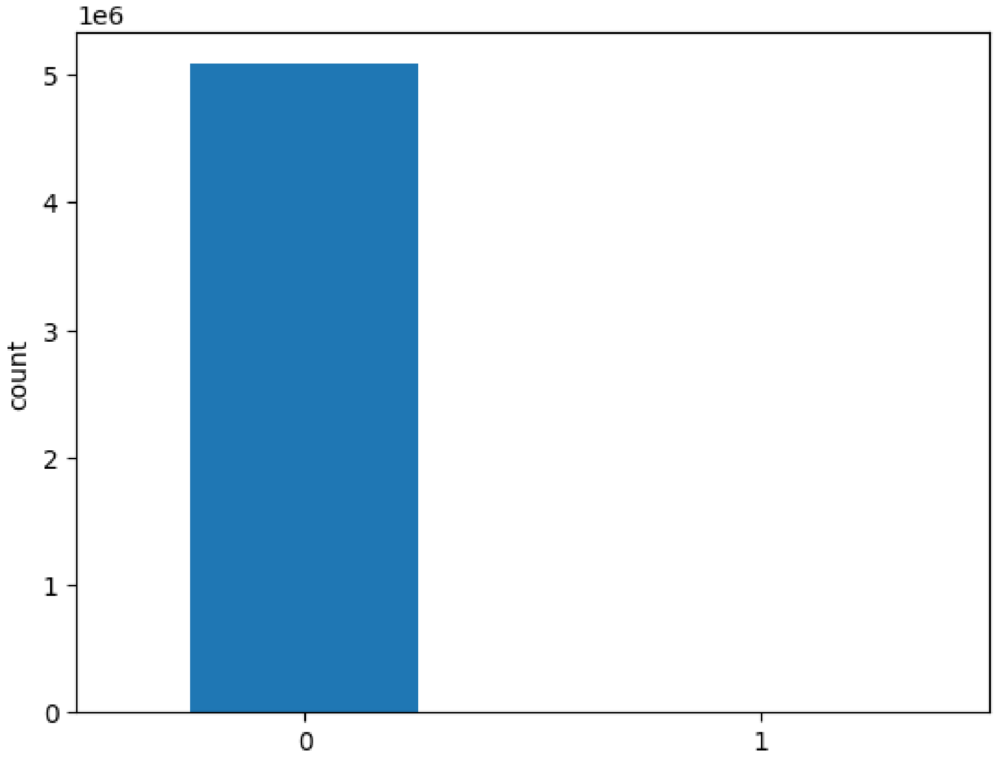

Figure 3.1 Highly Imbalanced Data

To overcome this issue, the Synthetic Minority Over-sampling Technique (SMOTE) was applied. SMOTE is a popular oversampling technique that generates synthetic instances of the minority class by interpolating the feature space between existing instances. By introducing synthetic instances, SMOTE helps balance the class distribution, thereby mitigating the impact of data imbalance on the modelling process.

In our study, SMOTE was applied to the training data, creating synthetic instances of fraudulent transactions. This balanced training set was then used to train the machine learning models, ensuring that they were exposed to an equal representation of both classes.

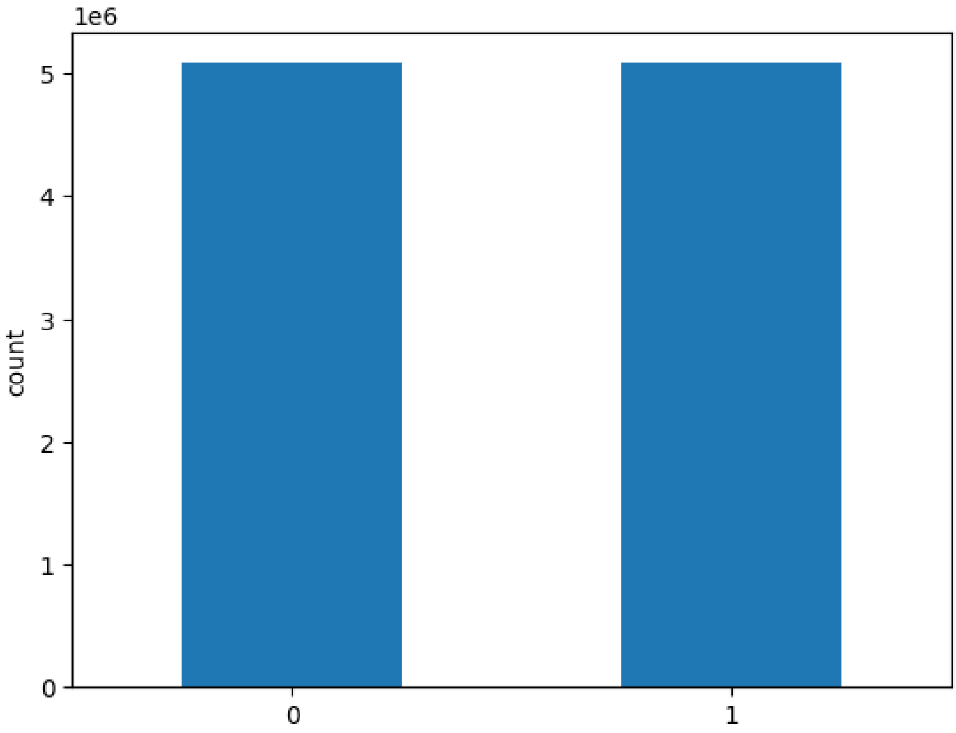

Figure 3.2 Data Resampling using Smote

By addressing the data imbalance, we aimed to improve the models' ability to capture patterns and make accurate predictions for both fraudulent and non-fraudulent transactions.

Including this information in the methodology section will provide a clear understanding of how you handled the data imbalance issue and the rationale behind using the SMOTE technique.

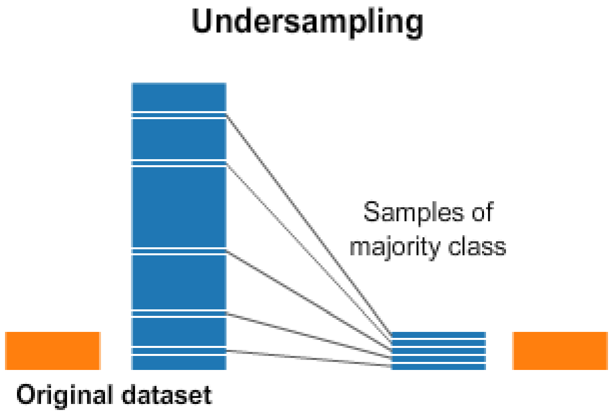

Figure 3.3 Under Sampling

Additionally, it is important to acknowledge that random under sampling, which was not explicitly mentioned earlier, was employed as a data re-sampling technique in this study. Random under sampling involves randomly selecting a subset of instances from the majority class (non-fraudulent transactions in this case) to create a more balanced dataset. While random under sampling can help mitigate the class imbalance, it should be noted that this technique may result in the loss of potentially valuable information present in the majority class instances. The decision to use random under sampling was made in conjunction with the SMOTE technique to explore the impact of different re-sampling approaches on the performance of the machine learning models.

CHAPTER 4

EXPERIMENTAL SETUP

4.1 Evaluation metrics

Evaluation metrics are quantitative measures used to assess the performance and effectiveness of a statistical or machine learning model. These metrics provide insights into how well the model is performing and help in comparing different models or algorithms.

4.1.1Accuracy

Accuracy measures the overall correctness of the model's predictions by calculating the ratio of correctly classified instances to the total number of instances. While accuracy is a useful metric, it can be misleading in imbalanced datasets where the majority class dominates. Therefore, it should be used in conjunction with other metrics, especially in cases of imbalanced classes.

In credit card fraud detection, accuracy indicates the proportion of correctly identified fraud and non-fraud transactions. However, accuracy can be misleading when dealing with imbalanced datasets, where the number of non-fraudulent transactions significantly outweighs the fraudulent ones.

Accuracy = (True Positives + True Negatives) / (True Positives + True Negatives + False Positives + False Negatives)

4.1.2 Precision

Precision quantifies the proportion of correctly predicted positive instances (true positives) out of all instances predicted as positive (true positives + false positives). Precision focuses on the accuracy of positive predictions and is useful when minimising false positives is crucial. For credit card fraud detection, precision is important as it measures the ability to correctly identify actual fraud cases, reducing false alarms. Precision represents the ability of the model to accurately identify fraudulent transactions while minimising false positives (non-fraud transactions wrongly classified as fraud). Precision:

Precision = True Positives / (True Positives + False Positives)

4.1.3 F1-Score

The F1 score is the harmonic mean of precision and recall. It provides a balanced measure that considers both precision and recall simultaneously. In credit card fraud detection, the F1 score captures the trade-off between precision and recall. A higher F1 score indicates a better balance between correctly identifying fraudulent transactions (precision) and minimising false negatives (missed fraud cases, recall). The F1 score is particularly useful when dealing with imbalanced datasets, where it is important to consider both false positives and false negatives.

F1 Score = 2 * (Precision * Recall) / (Precision + Recall)
Recall (Sensitivity) = True Positives / (True Positives + False Negatives)

4.1.4 Matthews Correlation Coefficient (MCC)

MCC is a correlation coefficient that takes into account true positives, true negatives, false positives, and false negatives. It provides a balanced measure that considers all aspects of a confusion matrix. MCC ranges from -1 to 1, with 1 representing a perfect prediction, 0 indicating a random prediction, and -1 representing total disagreement between predictions and true labels. MCC is a reliable metric for evaluating the performance of binary classification models, including credit card fraud detection models. In credit card fraud detection, MCC evaluates the overall performance of the model by considering the balance between true positives, true negatives, and errors. A high MCC score indicates a strong correlation between the model's predictions and the true labels. It is a reliable metric for assessing the performance of binary classification models, including credit card fraud detection models. Matthews Correlation Coefficient (MCC):

MCC = (TP * TN - FP * FN) / sqrt((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN))

where:
TP = True Positives
TN = True Negatives
FP = False Positives
FN = False Negatives

4.2 Training and testing procedure

The experimental setup for the research project on "Comparative study and analysis of credit card fraud detection with machine learning techniques" involved the following steps:

4.2.1 Dataset

The "Online Payments Fraud Detection Dataset" obtained from Kaggle was used as the primary source of data. The dataset consisted of 6,362,620 instances, with 6,354,407 instances labelled as non-fraudulent (0) and 8,213 instances labelled as fraudulent (1).

4.2.2 Data Pre-processing

The dataset underwent several pre-processing steps. There were no missing values in the dataset, and no imputation techniques were required. Two columns, "nameOrig" and "nameDest," were dropped as they were deemed irrelevant to the fraud detection task. The "type" column, representing transaction types, was transformed into numeric values (CASH_IN: 1, CASH_OUT: 2, PAYMENT: 3, DEBIT: 4, TRANSFER: 5) to facilitate further processing. The dataset was highly imbalanced, with a significant majority of non-fraudulent instances (0's) compared to fraudulent instances (1's). To address this imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was applied to generate synthetic instances of the minority class, ensuring a balanced class distribution.

4.2.3 Data Split

To explore the impact of different data split ratios on the performance of the machine learning models, the dataset was divided into training and testing sets using various ratios. The following ratios were considered: 80-20, 70-30, 60-40, 40-60, 30-70, 20-80, 10-90, and 5-95. These ratios represent the proportions of instances allocated to the training set and the testing set, respectively. By varying the data split ratios, the study aimed to assess how different training-testing set distributions affected the models' performance and generalization capabilities. This analysis helps to understand the robustness and stability of the machine learning models across different data split scenarios and provides insights into the optimal allocation of data for training and testing purposes.

4.2.4 Machine Learning Models

The selection of machine learning models was based on previous research papers in the field of credit card fraud detection. The chosen models included Gradient Boosting, XGBoost, Logistic Regression, Random Forest, Decision Tree, and KNN. These models have been widely studied and reported to perform well in fraud detection tasks.

4.2.5 Evaluation Metrics

The performance of the models was assessed using multiple evaluation metrics, including accuracy, precision, F1-score, Matthews Correlation Coefficient (MCC). These metrics provided a comprehensive analysis of the models' predictive capabilities and their ability to distinguish between fraudulent and non-fraudulent transactions.

The experimental setup involved no specific techniques for regularisation or validation to address over fitting. The pre-processing steps, except for resampling, were applied to the entire dataset. Resampling, performed exclusively on the training data using SMOTE, helped alleviate the class imbalance issue. No cross-validation or holdout validation was conducted during the training phase.

It is important to note that this experimental setup has certain limitations, such as the absence of hyper parameter tuning and a defined criteria for metric selection. These considerations should be taken into account when interpreting the results.

By outlining this experimental setup, the research project aimed to investigate and compare the performance of various machine learning algorithms in credit card fraud detection.

CHAPTER 5

RESULTS AND DISCUSSION

The research aimed to evaluate the performance of different machine learning models on various data splits and a specific data sampling technique. The accuracy, MCC, F1-score and precision results for each model and split configuration are presented below:

5.1 Accuracy

Accuracy	**Gradient**	**RandomF**	**Decision**	**KNN**	**Logistic**	**XGBoost**
80-20	0.98232	0.99934	0.99945	0.99541	0.9182	0.99818
70-30	0.98272	0.9993	0.99944	0.99537	0.93202	0.99836
60-40	0.98408	0.99929	0.9994	0.99511	0.93334	0.99827
40-60	0.98334	0.99922	0.9993	0.99448	0.93106	0.99831
30-70	0.98493	0.99915	0.99926	0.99387	0.94035	0.99866
20-80	0.98397	0.99911	0.99914	0.99338	0.93283	0.99848
10-90	0.98487	0.99902	0.9989	0.99254	0.92681	0.99855
5-95	0.98597	0.99886	0.99866	0.99069	0.92235	0.99856
UndSamp	0.98162	0.98707	0.98765	0.94366	0.90972	0.99114

Table 5.1 Accuracy Comparison

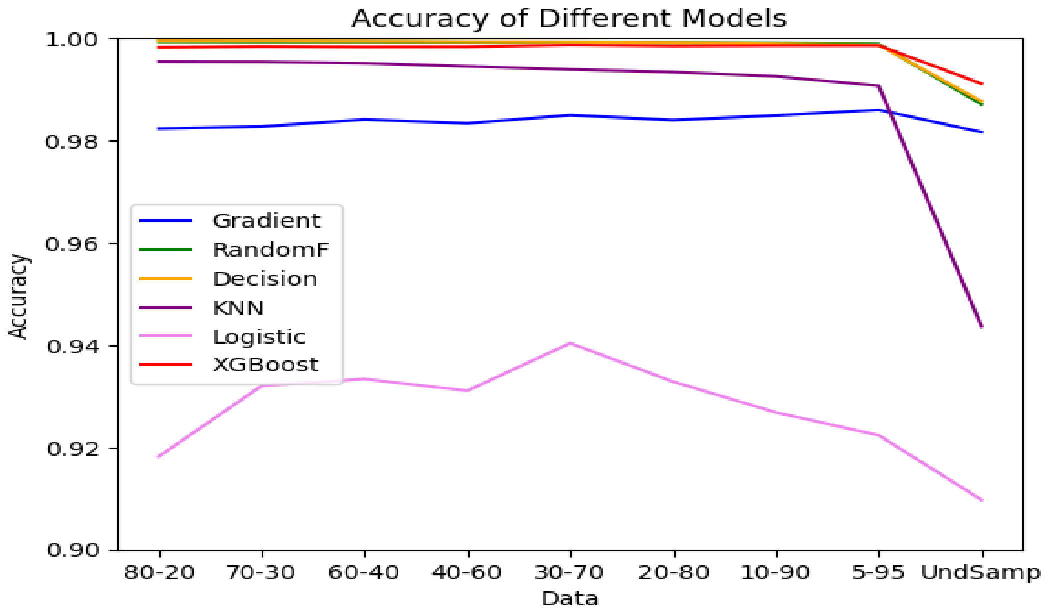

Figure 5.1 Accuracy Comparison

The results demonstrate varying levels of accuracy achieved by different machine learning models across different data splits and the random undersampling technique. Overall, the RandomF model consistently outperformed other models, achieving the highest accuracy in all data splits except for the 80-20 split, where the Decision model showed the highest. While under sampling XGBoost has the best performance.

5.2 MCC

MCC	Gradient	RandomF	Decision	KNN	Logistic	XGBoost
80-20	0.25583	0.79872	0.82896	0.42535	0.10412	0.63919
70-30	0.25855	0.78953	0.82522	0.42446	0.11618	0.65791
60-40	0.26912	0.78766	0.81451	0.41674	0.11841	0.64885
40-60	0.26436	0.76879	0.79315	0.39404	0.11657	0.65324
30-70	0.27677	0.75324	0.78139	0.3758	0.12788	0.6972
20-80	0.26895	0.73993	0.75185	0.36393	0.11972	0.67285
10-90	0.27541	0.71249	0.69729	0.34165	0.11348	0.67729
5-95	0.28453	0.67457	0.6428	0.3038	0.1096	0.67437
UndSamp	0.25158	0.2966	0.30253	0.13912	0.09827	0.352428

Table 5.2 MCC Comparison

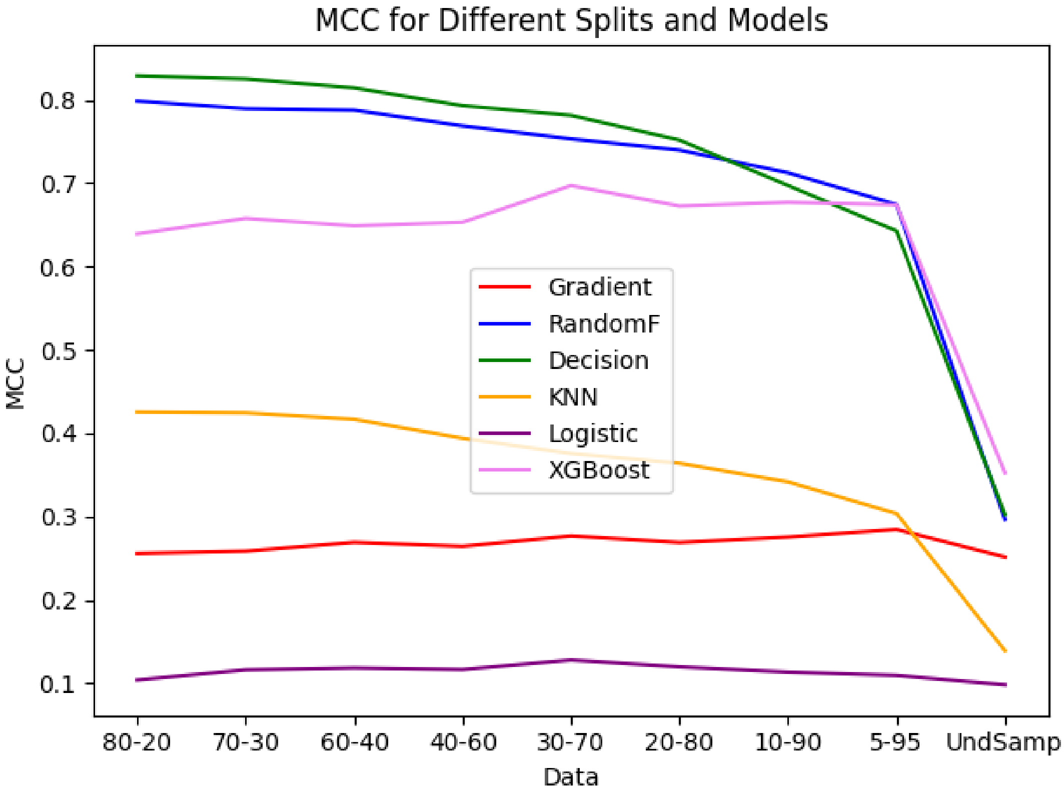

Figure 5.2 MCC Comparison

The results indicate the performance of different machine learning models in terms of the Matthews Correlation Coefficient (MCC) across various data splits and the utilization of random undersampling.

Observing the MCC values, it is evident that the Random Forest model consistently demonstrates the highest MCC scores for most data splits, indicating its effectiveness in capturing the true positive rate, true negative rate, and the balance between them. The Decision tree model also shows competitive performance, consistently achieving high MCC values across multiple splits.

Furthermore, when applying random undersampling, the MCC value decreases compared to the other models and data splits. This suggests that random undersampling may affect the model's ability to accurately classify instances, potentially due to the loss of valuable information present in the original dataset.

5.3 F1-score

F1-score	Gradient	RandomF	Decision	KNN	Logistic	XGBoost
80-20	0.125432	0.78665	0.8197	0.33161	0.02685	0.58238
70-30	0.12805	0.77639	0.81629	0.33035	0.03242	0.60704
60-40	0.13801	0.77546	0.80469	0.3201	0.0334	0.59594
40-60	0.13334	0.75499	0.78028	0.29357	0.03246	0.60178
30-70	0.14557	0.73894	0.76811	0.27233	0.03797	0.65755
20-80	0.13823	0.72481	0.73682	0.25781	0.03382	0.62804
10-90	0.14451	0.69821	0.67771	0.23311	0.03085	0.63572
5-95	0.15368	0.65807	0.6204	0.19288	0.02904	0.63435
UnderSamp	0.06464	0.1641	0.17033	0.04178	0.02432	0.22304

Table 5.3 F1-Score Comparison

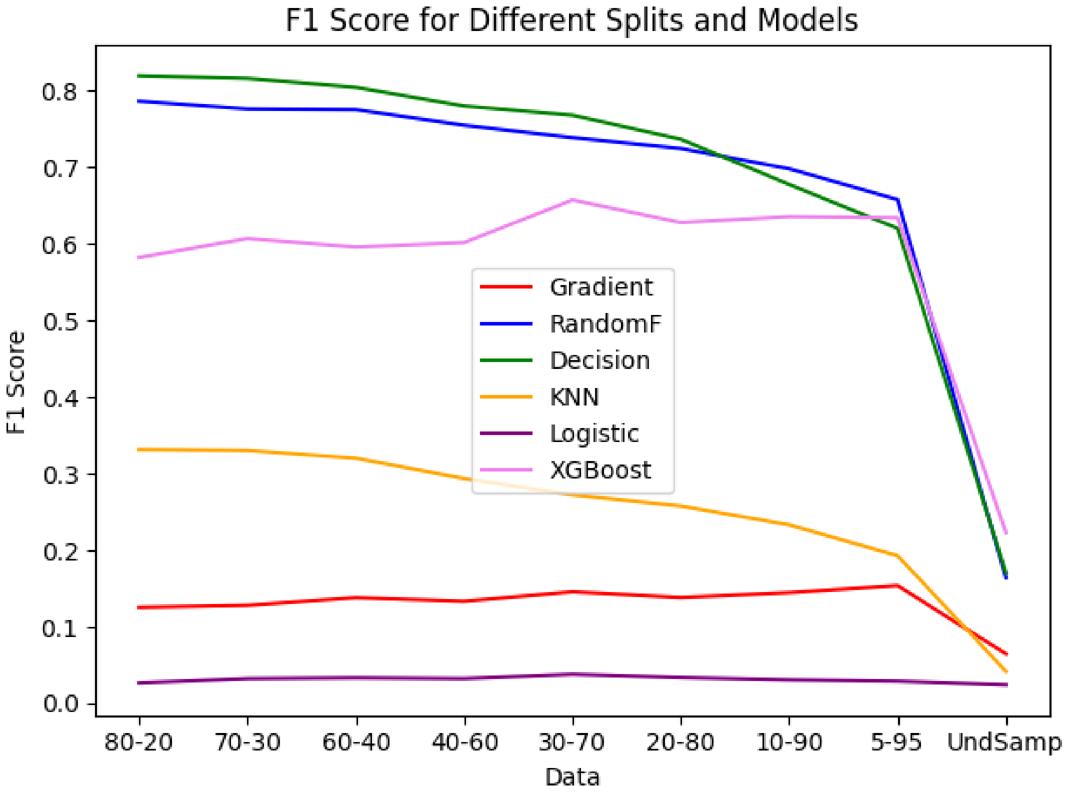

Figure 5.3 F1-Score Comparison

Based on the F1 scores, it is evident that the Decision and XGBoost models consistently achieve higher F1 scores across different data splits. These models demonstrate the ability to balance precision and recall effectively, making them suitable for binary classification tasks.

The Random Forest model also performs well, consistently obtaining competitive F1 scores. However, it tends to have slightly lower F1 scores compared to the Decision and XGBoost models.

On the other hand, the Gradient Boosting, KNN models show lower F1 scores across different data splits, indicating their limitations in accurately predicting positive instances.

Additionally, when applying random undersampling, the F1 score decreases significantly compared to the other models and data splits. This suggests that random undersampling may lead to a loss of F1 score and overall performance.

5.4 Precision

Precision	Gradient	RandomF	Decision	KNN	Logistic	XGBoost
80-20	0.06693	0.66941	0.7122	0.20357	0.01363	0.41187
70-30	0.06843	0.6559	0.71059	0.20258	0.01651	0.43753
60-40	0.07416	0.6586	0.6956	0.19482	0.01701	0.42645
40-60	0.07146	0.634	0.66046	0.1758	0.01652	0.43265
30-70	0.07856	0.61769	0.64778	0.16095	0.01939	0.49282
20-80	0.0743	0.6022	0.61335	0.15088	0.01723	0.46154
10-90	0.07795	0.58084	0.54744	0.13443	0.01569	0.47199
5-95	0.08334	0.53807	0.4905	0.10861	0.01476	0.47292
UndSamp	0.06464	0.08941	0.09314	0.02135	0.01233	0.12555

Table 5.4 Precision Comparison

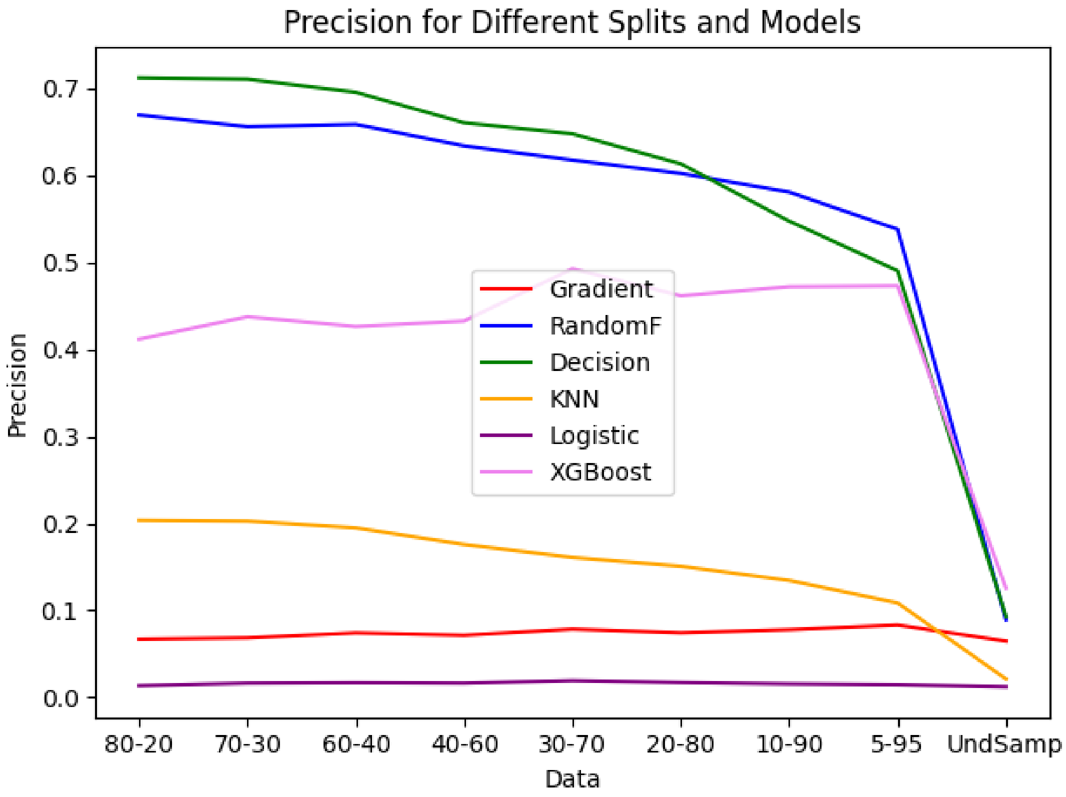

Figure 5.4 Precision Comparison

Based on the Precision scores, the Decision model consistently achieves higher precision scores across different data splits. It demonstrates a higher ability to accurately predict positive instances with a low false positive rate.

The XGBoost model also performs well, consistently obtaining competitive precision scores. It demonstrates a good balance between precision and recall, making it a suitable choice for classification tasks.

The RandomF model shows moderate precision scores, indicating its ability to predict positive instances. However, it tends to have slightly lower precision scores compared to the Decision and XGBoost models.

The GradientBoosting and KNN models exhibit lower precision scores across different data splits, suggesting their limitations in accurately predicting positive instances.

Furthermore, when applying random undersampling, the precision score decreases significantly compared to the other models and data splits. This indicates that random undersampling may lead to a loss of precision and overall performance.

	Our findings **(80/20)**	[1]	[3]	[4]	[6]	[7]
Logistic regression	0.98232	0.9718	99.074	90.448	97.70	96.694%
Decision tree	0.99945	0.9708	-	90.998	95.50	99.958%
Random forest	0.99934	0.9998	-	94.9991	-	99.950%
XGBoost	0.99818	-	-	-	-	-
KNN	0.99541	-	96.91	94.999	97.69	99.681%
Gradient Boosting	0.98232	-	-	94.001	-	-

Table 5.5 Accuracy Comparison

CHAPTER 6

CONCLUSION

In this study, we evaluated the performance of various machine learning models using different data splits and assessed their accuracy, Matthews Correlation Coefficient (MCC), F1 score, and precision. Here are the key findings and conclusions based on the results:

Accuracy: The Decision model consistently achieved the highest accuracy across different data splits. It demonstrated robust performance in correctly classifying instances and had the highest overall accuracy compared to other models. The Random Forest (RandomF) and XGBoost models also performed well, showing competitive accuracy scores.

MCC: The Decision model consistently achieved the highest MCC scores across different data splits. The MCC metric takes into account true positives, true negatives, false positives, and false negatives, providing a comprehensive measure of model performance. The RandomF and XGBoost models also demonstrated good MCC scores, indicating their effectiveness in capturing the underlying patterns in the data.

F1 Score: The Decision model consistently achieved the highest F1 scores across different data splits. The F1 score considers both precision and recall, providing a balanced measure of model performance. The RandomF and XGBoost models also showed competitive F1 scores, indicating their ability to achieve a balance between precision and recall.

Precision: The Decision model consistently achieved the highest precision scores across different data splits. Precision measures the ability of the model to correctly predict positive instances while minimising false positives. The XGBoost model also showed competitive precision scores, indicating its effectiveness in predicting positive instances with a low false positive rate.

Overall, based on the evaluation of accuracy, MCC, F1 score, and precision, the Decision model emerged as the top-performing model in this study. It consistently demonstrated high accuracy, MCC, F1 score, and precision across different data splits, indicating its effectiveness in accurately classifying instances and capturing the underlying patterns in the data.

The RandomF and XGBoost models also showed strong performance across the evaluation metrics, making them viable alternatives depending on the specific requirements of the classification task.

It is worth noting that the choice of the appropriate model depends on the specific context, dataset characteristics, and objectives of the classification task. Therefore, it is important to consider these factors when selecting the most suitable model for a given application.

In conclusion, the findings of this study highlight the importance of carefully evaluating and selecting machine learning models based on multiple performance metrics. The Decision model, with its consistent high performance in accuracy, MCC, F1 score, and precision, demonstrates its potential as a reliable choice for classification tasks.

CHAPTER 7

LIMITATIONS AND FUTURE SCOPE

7.1 Limitations

7.1.1 Evolving Fraud Techniques

Credit card fraudsters are continually evolving their techniques to bypass detection systems. As machine learning models are trained on historical data, they may struggle to identify new and sophisticated fraud patterns that have not been encountered before. This limitation calls for constant model updates and the integration of real-time data to stay ahead of emerging fraud tactics.

7.1.2 Imbalanced Data

One major limitation of credit card fraud detection models is the issue of imbalanced data. In real-world scenarios, fraudulent transactions are relatively rare compared to legitimate ones. This class imbalance can lead to biased models that may have a higher false positive or false negative rate. Ensuring a balanced representation of both fraudulent and legitimate transactions in the training data is crucial to mitigate this limitation.

7.1.3 Scarcity of Training Data

Due to the confidential nature of credit card information, banks and other financial institutions do not readily provide varied transaction data for training models. As a result, there is a limitation in accessing diverse transaction data to train the model.

7.2 Future Scope:

7.2.1 Real-time Fraud Detection

Real-time fraud detection is another promising future scope. Traditional machine learning models often operate in batch mode, processing transactions in predefined intervals. However, as fraudsters become more sophisticated, real-time detection becomes crucial. Implementing machine learning models capable of analysing and making decisions on transactions in real-time can help prevent fraudulent transactions more effectively.

7.2.2 Integration of Advanced Techniques:

Enhancing Model Performance using feature selection such as using Genetic Algorithm.

The above-mentioned limitations and future scopes reflect the ongoing challenges and advancements in the field of credit card fraud detection using machine learning. The field is dynamic and adaptive, and researchers and practitioners are actively working to address these limitations and explore new avenues for improving fraud detection models.

REFERENCES

[1] Geetha S & Vaishnavi, 2019. “Credit Card Fraud Detection using Machine Learning Algorithms.” *Procedia Computer Science* Vol. 165 (2019): pp. 631–641

[2] S P Maniraj, Aditya, Swarna & Shadab, 2019. “Credit Card Fraud detection using M.L & Data Science.” *International Journal of Engineering Research & Technology (IJERT)* Vol. 8 Issue 09, ISSN: 2278-0181

[3] Olawale, Julius, Shiwani & Hemaint, 2019. ”Comparative Evaluation of Credit Card Fraud Detection Using Machine Learning Techniques.” *IEEE* 978-1-7281-3694-

[4] Naresh, Sarita, Umesh & Sanjeev, 2020. “Efficient Credit Card Fraud Detection Model Based on Machine Learning.” *International Journal of Advanced Science and Technology(IJAST)* Vol. 29, No. 5, pp. 3414 - 3424

[5] Hassan, Ola Altiti, Ayah Abu & Mutaz, 2020. “Credit Card Fraud Detection Based on Machine Learning & Deep Learning.”

[6] Emmanuel Ileberi, Yanxia & Zenghui Wang, 2022. “A Machine Learning Based Credit Card Fraud Detection Using The GA Algorithm for Feature Selection.” *Journal of Big Data* Vol. 9:240.40537-022-00573-8

[7] Mosa M. Megdad, Bassem & Samy S. Abu-Naser, 2022. “Fraudulent Financial Transactions Detection Using Machine Learning.” *International Journal of Academic Information Systems Research (IJAISR)* Vol. 6 Issue 3,March - 2022, ISSN: 2643-9026

[8] Nikhil, Vinay, Tanooj & Dr.T.V.S.SriRam, 2023. “Credit Card Fraud Detection Using Machine Learning Algorithm.” *Journal of Engineering Sciences.* Vol. 14 Issue 04,2023, ISSN:0377-9254