School of Computing, Engineering and Digital Technologies
Department of Computing and Games
Teesside University
Middlesbrough TS1 3BA

Fraud Detection System in Financial Service Operations Using Machine Learning.

Analysis, Design and Implementation Report

Submitted in partial requirements for the degree of MSc in Data Science

Date: 8th May 2023

Author: Olatunji Jaiyeola

Supervisor: The Anh Han

ACKNOWLEDGEMENTS

I appreciate my supervisor Professor The Anh Han for his constant support and guidance. He was supportive throughout the duration of this project and his advise and mentoring has made it possible for me to successfully complete this project.

My appreciation equally goes to my professional colleagues; Adewunmi Sulaiman, Olatunji Olabisi and Oludeji Arale for training and mentoring me over the years, the professional knowledge you have passed on to me have made a difference throughout my programme.

To my amazing colleagues at Teesside University, Oluwafemi Olasupo thank you for the care, support and for always believing in me. Olajuwon Obaniyi you have been a resourceful and a great study partner, Omotola Uti-Bright and Gbadamosi Azeez for your encouragements and help.

Last but not the least, I wish to acknowledge the role my family played in helping me actualise my dreams, thank you for the support and encouragements.

To everyone that have contributed to the success of this project through your prayers, support and encouragements, THANK YOU.

Declaration

I **Olatunji Jaiyeola,** hereby declare that the project titled “Fraud Detection System in Financial Service Operations Using Machine Learning” has been completed entirely by me and has not been copied in part or in whole from any other source except where duly acknowledged. No other individuals have been involved in the completion of this dissertation and the work presented here is mine. I also declare that this dissertation has been prepared in accordance with the requirements of this course only and it has not been used for any other purpose before prior to this submission.

An analysis, design and implementation report for the development of an application to detect fraud in financial services operations using machine learning.

Abstract

This project explores how machine learning can be used to detect fraud in financial service operations. The current global reality that has adopted the use of technology in facilitating and conducting economic activities has increased the reliance on electronic transactions. This has helped organisations to expand beyond small geographical boundaries, however, this has also led to an increase in the volume of fraudulent transactions that affects both business and customers.

There is need for financial institutions to have systems in place that can detect and prevent frauds from taking place, this will prevent customers and financial institutions from losses and reputational damage. This project develops machine learning model can be used to detect fraud in financial service operations.

The project evaluates the performance of machine learning algorithms deployed and compares the performance with existing models from previous works. Synthetic data was used to train single and ensemble classifiers to evaluate the best performing algorithm. The project also applied both over-sampling and under-sampling methods to solve the problem of class imbalance that was created by the presence of more non-fraud instances in the dataset.

From the study, using various metrics as a basis of evaluation, Extreme gradient boosting algorithm produced the best result with a recall of 87%, F1-Score of 90%, false alarm rate of 0.07%, balance classification rate of 90% which proves the algorithm’s ability to maintain a balance between the recall and precision.

Table of Contents

Chapter 1

1.0. Introduction

The adoption of online transactions, mobile payments, and electronic banking has made the threat of financial fraud to become more sophisticated and common, costing financial institutions and customers billions of pounds annually. The development of technology in recent years has made it much harder to identify fraud because fraudsters now employ complex and modern tools to conduct their operations.

Financial institutions stopped £1.6 billion in illegal financial losses in 2020, or £6.73 out of every £10 in attempted fraud that was stopped (UK Finance, 2021). This shows the extent of damage that can be caused by fraudsters in financial services. As a result, financial institutions are always searching for fresh strategies to identify and stop fraud. Adopting the use of machine learning is one of such options as this allows for a more dynamic and effective way to combat the activities of fraudsters when compared to the traditional rule base method which is more expensive cost wise and can lead to bad customer experience.

The ability of computers to learn from data and make predictions or judgements without having to be explicitly programmed is known as machine learning (Brown, 2021). It has been used in the detection and prevention of fraud in numerous areas, including finance. Because of the level of success in identifying fraud, machine learning is being used more frequently in fraud detection. This study examines how historical data can be used by machine learning models to stop fraud in financial service operations.

1.1 Research Background and motivation

Kranacher and Riley (2020) defined "fraud as an intentional deception, whether by omission and co-mission, that causes its victim to suffer an economic loss and/or the perpetrator to realise a gain”. Fraud can have a severe negative effect on the reputation of the financial institution and confidence of the clients in the financial services industry.

Traditional fraud detection techniques, like rule-based systems, have been used by the financial services industry to identify fraud. However, since these techniques are dependent on a set of pre-established criteria, it is challenging to identify some types

of fraud (Preece, Shinghal and Batarekh, 1992). In addition, fraudsters can find ways to beat these systems, thereby making them less effective.

Machine learning can learn patterns in data and make decisions based on them. Machine learning models can identify patterns in data that are difficult to detect by human, this helps validate it as an effective way of detecting financial fraud. Machine learning can readjust to new data and new types of fraud, making it more effective in detecting fraud.

Algorithms can be used to create synthetic data as against collecting data from real-world environments. Synthetic data is used in many industries, including finance, to test and evaluate machine learning algorithms. It is useful in building machine learning models for financial sector as it involves the development of machine learning models in a controlled environment, without the risk of exposing real-life data.

1.2 **Research objectives**

The objective of this dissertation is to examine how machine learning can be used to prevent fraud in financial service operations using synthetic data. This project aims to achieve the objectives below:

- To review previous research on the use of machine learning to detect fraud in financial transactions
- To probe the prospects of using machine learning to prevent fraudulent transactions
- To design and implement different machine learning algorithms that can detect fraudulent financial transactions
- To evaluate the different machine learning algorithms developed for detecting fraud
- What are the limitations of using machine learning to prevent fraud in financial services operations?

1.3 Research Question

How can Machine Learning models be used to prevent fraud in financial service operations?

1.4 Overview of the report

This project consists of 6 chapters that are highlighted below:

Chapter 1- Introduction: This chapter provides the background context for the research, by focussing on the aims, objectives and motivation for the study.

Chapter 2- Literature Review: This section analyses the relevant academic literature that have been published on the use of machine learning to prevent fraud in financial service operations. It focuses on the type of dataset used, the approached used in solving class imbalance, the choice of algorithms and the performance of the models.

Chapter 3- Methodology: This chapter provides an analysis of the dataset, the different pre-processing steps carried out. It also provides a detailed analysis on the algorithms deployed and the model training process. The chapter also covers the review of the evaluation metrics that were used to assess the performance of the models

Chapter 4- Implementation: It explains the different pre-processing steps carried out such as exploratory data analysis, handling outliers, data encoding, data standardisation and feature selection

Chapter 5- Interpretation and discussion of results: This covers the performance report of the different algorithms and compares with previous research described in the literature review.

Chapter 6 Conclusion: The chapter summarizes the findings from the study, it also the discusses possible future works and areas for improvements.

Chapter 2

2.1 Literature Review

Many studies have been done on the use of machine learning in fraud, the papers reviewed agreed that machine learning algorithms are effective in detecting fraud as the classifiers can detect patterns in data that are difficult to detect by humans.

2.1.1 Availability of data

The lack of real-life data is common in the implementation of machine learning models in identifying fraudulent transactions due to data privacy and sensitivity. This necessitated some of the important but sensitive data to be masked or transformed using principal component analysis (PCA) (Dornadula and Geetha, 2019). Rtayli and Enneya (2020) used synthetic data which contained over 1million records. The dataset was generated from a simulation of mobile money transaction modelled from real-life transactions.

2.1.2 Class imbalance

Skewed data is common in fraud prediction because fraudulent transactions are usually rare compared to legitimate transactions. This imbalance between the number of fraudulent and non-fraudulent transactions can lead to a biased model that performs poorly in identifying fraudulent transactions. According to the research (Tae and Hung, 2019) dataset used in the study contains 284,807 transactions with 492 fraud transaction, this implies that the fraud samples account for 0.172 of the entire dataset. Zarepoor and Shamsolmoali (2015) in a study to evaluate the performance of three advanced data mining techniques for credit card detection. The dataset used for the study contains 100000 records of credit card transactions with 2.8% being fraudulent transactions and 97.2% being legitimate transactions.

The skewed data causes machine learning algorithms to be biased towards the majority class, resulting in poor performance in detecting instances of the minority class. To solve the issue of imbalanced dataset, a technique of Synthetic Minority Oversampling Technique (SMOTE) was used in research by Rtayli and Enneya, (2020). SMOTE generates synthetic samples of the minority class (in this case, the fraudulent transactions) to balance out the majority class in the dataset. A study to detect fraudulent credit card transactions in four fraud types of namely bankruptcy,

counterfeit/credit fraud, application fraud, and behavioural fraud was carried out by Thennakoon et al. (2019). The dataset used for the study were characterised by a highly imbalance distribution of the target classes, to resolve this, the study employed the use of both Over-sampling and Under-sampling. Synthetic Minority Oversampling Techniques (SMOTE) was used for over-sampling while Random under-sampling was used for Under-sampling.

2.1.3 **Supervised vs Unsupervised Learning**

In a study by Choi and Lee (2018) using data generated from IoT driven system, unsupervised learning method was used to identify the underlying threats while supervised learning was used for the accurate classification of fraud transactions. Unsupervised learning algorithm was applied in feature selection. It employed the use of filter-based feature selection algorithm for the feature selection. This depends on the attributes of the data to determine the importance of features. Scores are assigned to features based on the evaluation metrices, features with low scores were removed from dataset to be used. Sliding-Window method was used by Dornadula and Geetha (2019). The method sums transactions into respective groups, this selects some features from the window to identify the behavioural pattern of the cardholder.

The combination of supervised and unsupervised learning was implemented in the study by Carcillo et al. (2019). The supervised method uses the labelled historical data to learn and when it receives a new data it computes the probability of a transaction being fraudulent, while the unsupervised outlier method does not require the labelled transaction, it characterises the distribution of the transaction data and assumes that the outliers of the transaction distribution are fraudulent. Supervised method learns from past fraud transaction while the unsupervised method focuses on the identification of new types of fraud

2.1.4 Machine learning classifier of choice

Various studies have used several machine learning algorithms in building fraud prediction models. (Xu, Fan and Song, 2022) used four machine learning models; gradient boosting decision trees, random forest, support vector machine, and decision tree in the analysis of financial data, GBDT has an AUC value of 0.898 which showed to be significantly better that the other classifiers. The study showed that using a single model may result into excessive bias or poor generalization, therefore it suggests the

use of ensemble learning to combine several models to train the model that can produce a better and powerful performance. The research implemented a multi-model approach that perform a 5-fold cross-validation on the training set using the result of the selected model, gradient boosting model is further used to train the integrated data obtained from the previously trained dataset, this produced a better AUC value of the integrated model which is significantly higher that using a single model

Kamboj and Shankey (2016) examined the use of support vector machine (SVM) model to detect credit card fraud and reduce false alarm, The research used a training dataset with adequate proportion of fraud to non-fraud labels, to replicate a real-life scenario, a test dataset with a much lower ratio of fraud labels was used to measure the expected output of the model where the proportion of fraudulent transactions are typically low. The study had an accuracy of 94.4%. Bagging Ensemble classifier was deployed with other machine learning algorithms by Zarepoor and Shamsolmoali (2015) in a study to evaluate the performance of three advanced data mining techniques for credit card detection. The dataset was trained using different classifiers with the bagging ensemble classifier. The decision tree model with bagging ensemble returned a higher fraud catching rate and a low false alarm rate when compared with other models that returned a higher false alarm rate. It also showed a better performance with highly imbalanced dataset.

A study used the following Machine learning algorithms for the study Support Vector Machine, Naïve Bayes, K-Nearest Neighbor and Logistic Regression (Thennakoon et al., 2019). Their performances were measured against 4 types of fraud. One of the major contributions of the project was the use of the machine learning algorithm as part of the credit card transaction. This helps to identify fraud in real time as against deploying on historical data.

2.2 Justification for the research

With the increase in number of fraud cases recorded in the financial sector and the rise in the adoption of technology in today's business world and day to day life, it has become imperative to develop systems that can effectively detect financial fraud. Financial fraud has become more advanced and difficult to spot as online transactions, mobile payments, and electronic banking have grown in popularity, making business and customers lose a lot of money yearly.

The need to develop fraud detection tools with better efficiency is one of the justifications for this study, which seeks to use machine learning to detect fraud in financial service operations.

Traditional fraud detection system majorly uses pre-defined rules that can be readily avoided by fraudsters who constantly alter their methods of operation. Machine learning algorithms, however, can learn from historical data, detect anomalies, and discover patterns that shows fraudulent tendency. The use of machine learning models can enable financial institutions improve their ability to detect fraud in quickly, prevent losses, and protect their customers. (Preece, Shinghal and Batarekh, 1992).

The opportunity for cost reductions is also a reason for engaging in this study. Financial fraud result in financial losses for financial institutions, which includes the cost of fraud investigation, and compensating for fraudulent acts. Machine learning can assist in reducing these costs by automating the fraud detection process, allowing organisations to detect and respond to fraud quickly. Also, machine learning model has the potential to minimise false positives, which occur when normal transactions are identified as fraudulent, resulting in investigations that are not required and delays in customers transactions. By optimizing fraud detection with machine learning, financial institutions can save resources and allocate these resources more effectively to other areas of their operations.

Another reason for this study is the potential for societal influence. Financial fraud has a tremendous influence not just on financial institutions but also on consumers and the general public. Fraudulent activities can lead to the loss of personal and financial information, the deterioration of credit scores, and the erosion of trust in financial institutions. Consumers can be better protected against fraud by developing more effective machine learning-based fraud detection systems, leading to higher trust in financial transactions and a more secure financial ecosystem.

Chapter 3

3.0 Methodology

This section provides explanation on the deployment of the selected machine learning algorithms. It will include details about the dataset used in the project, the pre-processing stage, the deployment of classifier and performance evaluation.

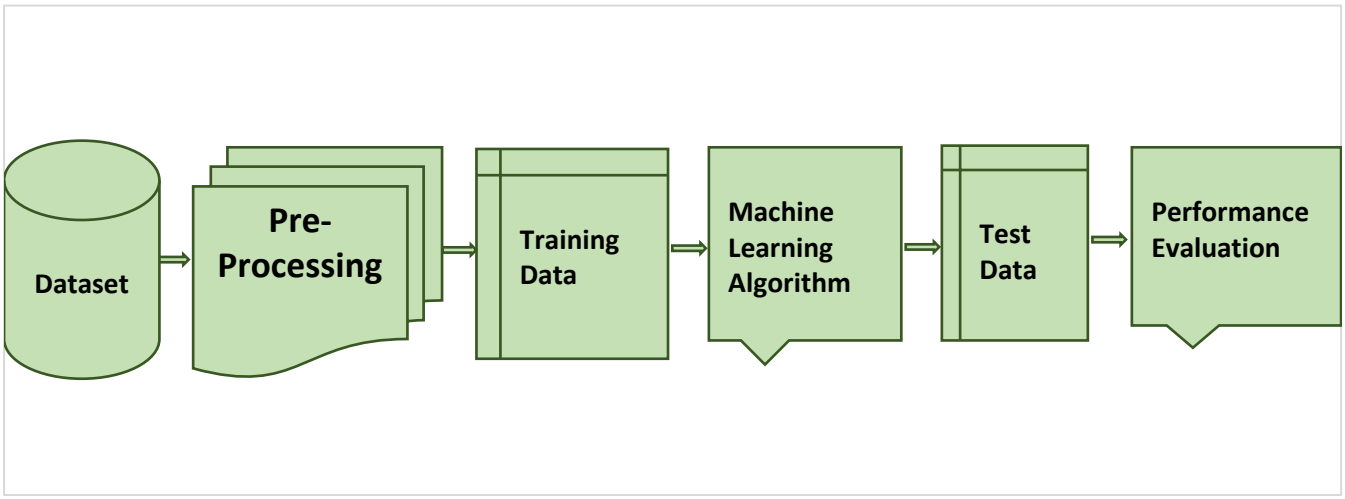

Figure 1: Project Methodology

3.1 Data collection – Dataset

The Bank Account Fraud Dataset Suite published at NeurIPS 2022 was obtained from kaggle.com. It is a synthetic bank account fraud dataset based on real-world pattern of transactions. The base dataset consists of 1million transactions which have undergone differential privacy techniques, feature encoding and trained generative model to anonymised the features to enhance data privacy. The dataset has about 1.1% of the positive class (fraud cases) and 98.9% representing the negative class (non-fraud cases). It has a combination of numerical and categorical input variables. There are 31 input features in the dataset with the fraud_bool feature being the output variable which is highly skewed towards the negative class.

Column name	Description
fraud_bool	Fraud status (Fraud transaction represented by 1while non fraud transaction is represented by 0)
income	Annual income of the applicant in quantiles. The range is between 0 and 1.
name_email_similarity	Metric of similarity between email and applicant's name. Higher values represent higher similarity.
prev_address_months_count	Number of months in applicant's previous registered address.
current_address_months_count	Months in currently registered address of the applicant. Ranges between [−1, 406] months
customer_age	Applicant's age in bins per decade (For example 20-29 is represented as 20).
days_since_request	Number of days passed since application was done.
intended_balcon_amount	Initial transferred amount for application.
payment_type	Credit payment plan type
zip_count_4w	Number of applications within same zip code in last 4 weeks. Ranges between [1, 5767].
velocity_6h	Average number of applications per hour in the last 6hours
velocity_24h	Average number of applications per hour in the last 24 hours)
velocity_4w	Average number of applications per hour in the last 4 weeks
bank_branch_count_8w	Total number of applications in the selected bank branch in last 8 weeks. Ranges between [0, 2521].
date_of_birth_distinct_emails_4w	Number of emails for applicants with same date of birth in last 4 weeks. Ranges between [0, 42].
employment_status	Employment status of the applicant.

credit_risk_score	Internal score of application risk. Ranges between [−176, 387].
email_is_free	Domain of application email (free or paid).
housing_status	Current residential status for applicant.
phone_home_valid	Validity of provided home phone.
phone_mobile_valid	Validity of provided mobile phone.
bank_months_count	How old is previous account (if held) in months. Ranges between [−1, 31] months (-1 is a missing value).
has_other_cards	If applicant has other cards from the same banking company.
proposed_credit_limit	Applicant's proposed credit limit. Ranges between [200, 2000].
foreign_request	If origin country of request is different from bank's country.
source	Online source of application. Either browser (INTERNET) or mobile app (APP).
session_length_in_minutes	Length of user session in banking website in minutes. Ranges between [−1, 107] minutes
device_os	Operating system of device that made request. options are Windows, Macintosh, Linux, X11, or other.
keep_alive_session	User option on session logout.
device_distinct_emails_8w	Number of distinct emails in banking website from the used device in last 8 weeks. Ranges between [0, 3].
device_fraud_count	Number of fraudulent applications with used device. Ranges between [0, 1].
month	Month where the application was made. Ranges between [0, 7].

Table 1: Dataset Description

3.2 Data Pre-processing

Pre-processing refers to the activities or techniques that are applied to input dataset before it is applied to train machine learning model. Pre-processing steps helps to clean, transform, and normalize the data, making it suitable for effective model training and prediction. Some pre-processing tasks performed in the study are checks for missing data, handling outliers, data encoding and data standardisation.

3.2.1 Handling missing data

Missing data occurs when some of the observations in a dataset are not complete. This can lead to a bias or wrong predictions when such data are applied to machine learning algorithms. To resolve this issue, different approaches can be applied to address this concern such as data imputation, deletion, substitution.

The method to be adopted in resolving this issue, is largely based on the proportion of missing data, nature of the missing values and goal of the analysis.

3.2.2 Handling Outlier

The existence of outliers in the data can negatively affect the effectiveness of machine learning models as it initiates noise to the data. Outliers are data points that have sharp deviation from the general pattern or distribution of the data (Li et al., 2020). Outliers can be handled by imputing extreme values with more representative values. This can be done using various techniques, such as mean, median, or mode imputation.

3.2.3 Data Encoding

This is the process of transforming categorical data into numerical data for machine learning models to utilize the data. Machine learning classifiers requires numerical input data, therefore before fitting in the data to the algorithms, categorical features need to be converted to numerical data (Dahouda and Joe, 2021). Binary encoding was used because the categorical features are nominal data with no order or ranking.

3.2.4 Data Standardisation

Data standardisation this is the process of scaling data to fit a standard normal distribution. The data has been scaled to have a mean of 0 and a standard deviation of 1. ScandardScaler class from scikit-learn was used for the purpose of scaling features to ensure that all the features have similar scale

3.2.5 Feature Selection

Feature selection refers to the process of obtaining a subset of data from an original dataset set according to specific selection criteria, this is used to select the features of interest from the dataset (Cai et al., 2018). Feature selection can be applied to improve the accuracy of the model. this also helps in reducing the time spent to train the model. This study computes the relationship between the target and the other variables in the dataset. Features with low correlation were dropped in order to improve the performance of the model.

3. 2.6 Class Imbalance

Class imbalance occurs when the instances of one class outnumber the instances of other classes (Guo et al., 2008). However, in many applications the important class is the class with the lower instance, most algorithms perform badly when the dataset is imbalance. The classifiers generally get outnumbered by the majority class (Buda, Maki and Mazurowski, 2018).

Imbalance datasets degrades the performance of data mining and machine learning techniques as the overall accuracy and decision making may be biased to the majority class, which lead to misclassifying the minority class samples or furthermore treating them as noise (Abd and Abraham 2013).

This study will implement some approaches that can be used to address class imbalance in a dataset, which occurs when the distribution of classes is significantly uneven. The techniques adopted in this study to address class imbalance are oversampling using Synthetic Minority Over-sampling Technique (SMOTE) and Under-sampling.

Oversampling: This involves creating synthetic copies of the minority class samples to increase their representation in the dataset. Techniques such as Synthetic Minority Over-sampling Technique (SMOTE) and its variants are popular oversampling methods.

Under-sampling: This involves randomly removing samples from the majority class to decrease its representation in the dataset. Technique such as Random Under-sampling is commonly deployed to under-sample datasets

3.3 Review of machine learning techniques

Four machine learning algorithms have been selected to be used in this project, a combination of single and ensemble classifiers have been selected to the classifiers selected are K-Nearest Neighbors, Random Forest, Logistic regression and Extreme gradient boosting classifiers.

3.3.1 K-Nearest Neighbors

KNN is a supervised learning algorithm used for classification and regression tasks. Predictions are made using the correlation of the data points in the features.

KNN works by finding the k closest data points in the feature space to the query point. The algorithm then makes a prediction using the most common label among the k neighbors for classification tasks or the average value for regression tasks. (Shokrzade et al., 2021)

KNN is widely preferred for its simplicity and flexibility. It can handle both binary and multi-class classification problems and can work with any number of features. Furthermore, it does not make any assumptions about the distribution of the data, making it suitable for both linear and nonlinear relationships between the features.

One of the limitations of KNN is that it can be computationally expensive, especially for large datasets, as it requires calculating the distance between the query point and all other data points. Second, the performance of KNN heavily depends on the choice of k, which can be a challenge to determine in practice. Finally, the algorithm is sensitive to the scaling of the features, as features with larger scales will dominate the distance metric.

3.3.2 Logistics Regression

Logistics regression is defined as a “statistical model which describe the relationship between a qualitative dependent variable (that is, one which can take only certain discrete values, such as the presence or absence of a disease) and an independent variable.” (Nick and Campbell 2007). The model is deployed to examine the implication of predictor on an outcome that is categorical. The logistic regression algorithm uses a sigmoid function to map input values to probability value that lies between 0 and 1. This probability value is then used to predict the class label of the input data.

It is a popular classification algorithm in machine learning because of its simplicity, interpretability, and effectiveness (Hosmer, Lemeshow, Sturdivant 2013).

A major advantage of logistic regression is the simplicity and interpretability. The algorithm is easy to deploy and does not need a large amount of computing resources. Additionally, the output of the algorithm is easy to interpret, which makes it useful for explaining the results to non-technical stakeholders (Hastie, Tibshirani, Friedman 2009)

However, logistic regression also has some disadvantages. It assumes a linear relationship between the independent features and the log odds of the dependent feature, which sometimes may not be valid in real-world scenarios. Additionally, logistic regression can only model binary outcomes, which may not be sufficient for some classification problems.

3.3.3 Random Forest

Random forest is an ensemble learning classifier that uses multiple decision trees to improve the accuracy and diminish the variance of the predictions.

According to Breiman (2001), the random forest algorithm builds a forest of trees using a type of decision tree algorithm, where each tree is built from a random subset of the training data and features. The algorithm then uses the ensemble of trees to predict the class label of the input data. Each decision tree in the forest predicts the class or value of the target variable and aggregating the predictions of all the trees produces the final prediction. In regression use case, the aggregation can be done through averaging while in classification use case, voting is used for aggregation. The random forest algorithm is frequently used in machine learning because it is robust to noise and overfitting and can handle high-dimensional datasets.

According to Barboza, Kimura and Altman (2017), Random Forest algorithm process follows the steps below

1. Creation of random subsets of the parent data, composed of an arbitrary number of observations and different features.
2. Each subset from step 1 produces a unique decision tree, and all elements of the set have a label (correct or not correct).

3. For each element, the forest aggregates the votes on the different trees. The class with the highest votes is chosen as the preferred classification of the element.

Random forest can handle high-dimensional datasets and nonlinear relationships between the variables. Additionally, the algorithm is robust to missing values and noisy data, which makes it useful for real-world datasets (Breiman, 2001). Furthermore, random forest can provide insights into the relative importance of the features in the classification problem, which can be useful for feature selection.

In conclusion, random forest has many applications in different fields and has the advantage of being robust to noise and overfitting.

3.3.4 Extreme Gradient Boosting

Extreme Gradient boosting is a machine learning technique that combines multiple weak classifiers, typically decision trees, to create a more accurate model.

XGBoost is based on gradient boosting, which is a machine learning technique that uses iteration to enhance the effectiveness of a model by introducing new models that correct the errors of the previous ones. (GhoshRoy, Alvi and Santosh 2022). A regularised term is added by the XGBoost algorithm to impose penalty on the complexity of the model in order to avoid overfitting.

XGBoost works by iteratively building decision trees and adding them to an ensemble model. During iteration, the classifier computes the gradient of the loss function with respect to the predictions and adjusts the parameters of the model to reduce the loss (Chen and Guestrin, 2016).

One of the advantages of XGBoost is its ability to handle missing data and outliers. The algorithm can use a variety of techniques to impute missing values, such as using the mean or median of the available data. XGBoost can also handle outliers by using a robust loss function that is less sensitive to extreme values.

In conclusion, XGBoost is a versatile machine learning algorithm that has become popular due to its speed, scalability, and ability to handle missing data and outliers.

3.4 Training and Test Data

Training and test sets are different subsets of data extracted from the dataset; they are used for different purposes during the model development process.

3.4.1 Training set

Machine learning model discovers patterns in the data and learns how to make correct predictions by using the training set. It contains a larger percentage of the data which is used to set the model's parameters during training.

3.4.2 Test set

Test set is the sample of data used to provide an unbiased evaluation of a final model fit on the training dataset. It used to measure how well the model generalizes to new data and gives an estimate of the efficiency with new real-world data.

3.5 Evaluation metrics used to assess the performance of the models

There are many evaluation metrics used to assess the performance of machine learning models. The choice of evaluation metric is based on the problem being solved and the goals of the model. The evaluation metrics used in this study are:

3.5.1 Accuracy

Accuracy measures the proportion of correctly classified instances among all the instances in the dataset. It measures the proportion of correct predictions made by the model.

$$\text{Accuracy} = \frac{\text{number of correct prediction}}{\text{total number of prediction}} \qquad \text{Equation 1}$$

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \qquad \text{Equation 2}$$

Where *TP* = True Positives, *TN* = True Negatives, *FP* = False Positives, and *FN* = False Negatives.

True Positive (TP): • Reality: Fraud • ML model predicted: Fraud	False Positive (FP): • Reality: Not Fraud • ML model predicted: Fraud
False Negative (FN): • Reality: Fraud • ML model predicted: Not Fraud	True Negative (TN): • Reality: Not Fraud • ML model predicted: Not Fraud

Table 2: Confusion matrix

While accuracy is a commonly used evaluation metric, it is not always the most appropriate metric for all problems. In some cases, accuracy may not be a good measure of model performance. For instance, in cases where the dataset is imbalanced, where there are more instances of one class than the other, the model tend to predict the majority class with a high accuracy but would perform poorly with the minority class.

In such cases, precision, recall, or F1 score may be more appropriate evaluation metrics.

3.5.2 Precision

Precision measures the proportion of true positives among all the instances that the model has predicted as positive. It is used when the focus is on minimizing false positives.

$$\text{Precision} = \frac{\text{number of correctly classified instances}}{\text{total number of instances}} \quad \text{Equation 3}$$

This equation can also be written as:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive+False Positive}} \quad \text{Equation 4}$$

A high precision score shows that the model is making accurate positive predictions, while a low precision score indicates that the model is making large number of false positive predictions.

Precision is particularly useful when the cost of a false positive is high, such as in medical diagnosis, where a false positive result can lead to unnecessary treatment. In such cases, a model with high precision is preferable, even if its recall (the proportion of true positives identified among all actual instances of the class) is lower.

3.5.3 Recall

Recall is a metric used in machine learning to evaluate the effectiveness of a binary classification model. It measures the proportion of actual positive samples that the model can correctly identify.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive+False Negative}} \qquad \text{Equation 5}$$

True positives are the samples that are correctly identified as positive by the model, and false negatives are the instances that are incorrectly identified as negative by the model.

A high recall score indicates that the model is effective at identifying positive samples, while a low recall score indicates that the model is missing many positive samples. Recall is an important metric, especially when the positive class is rare or when the cost of missing positive samples is high.

3.5.4 F1-Score

F1 score is the harmonic mean of precision and recall. It is used when both precision and recall almost same level of importance. It is important when there is a class imbalance.

$$\text{F1 score} = 2 * \frac{\text{Precision}*\text{Recall}}{\text{Precision+Recall}} \qquad \text{Equation 6}$$

The portion of true positives among the predicted positives is known as the precision, and recall is the fraction of true positives among the true positives.

3.5.5 ROC AUC

Receiver Operating Characteristic (ROC) and Area Under the Curve (AUC) are used for binary classification problems. ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) for different threshold values, and AUC measures the area under the ROC curve. Higher AUC values indicate better performance.

3.5.6 False Alarm Rate

False Alarm Rate (FAR) is a performance metric used in fraud prediction to evaluate the ratio of legitimate transactions that are wrongly flagged as fraudulent by a classification model. In other words, FAR is the rate at which the algorithm produces false alarms or false positives.

In fraud prediction, a high FAR can result in many false alarms, which can lead to customer dissatisfaction and increased costs for investigating false alarms.

Overall, FAR is an important metric for fraud prediction, as it helps to evaluate the performance of the classification model and improve the accuracy of the system while minimizing false alarms.

3.5.7 Balanced Classification Rate

Balanced classification rate is used to measure the efficiency of classification models in fraud prediction because it considers both true positive and true negative rates. In fraud detection, it is important to have a low false alarm rate (FAR), which is the ratio of legitimate transactions that are flagged as fraudulent, while also having a high true positive rate (TPR), which is the ratio of fraudulent transactions that are correctly identified as fraud.

BCR is computed as the average of true positive rate and true negative rate and can be used to evaluate the overall performance of a fraud detection system. A high BCR indicates that the system is accurate in both identifying fraudulent transactions and avoiding false alarms.

3.6 Ethical considerations and potential limitations of the research

Machine learning is a great innovation, however, gaps between the design and operation of algorithms and our understanding of their ethical implications can have severe consequences for individuals, groups, and entire society (Mittelstadt et al., 2016). There are different ethical considerations and potential limitations that need to be considered for this study. Some of which include:

Privacy and Confidentiality: Financial dataset usually contains details like credit card numbers, personal identifying information, transaction details which are sensitive information. Data de-dentification methods must be applied to such data to remove

any personal identifiable information. Unauthorized access must be prevented by ensuring that data is stored securely.

Bias and Fairness: Financial dataset may include biases such as sampling bias, selection bias, or measurement bias, which may affect the validity and generalizability of the study result.

It may also include concerns about fairness because certain demographic groups may be unevenly represented or affected by financial dataset (Lee and Shin (2019). The study will carefully consider the potential bias and ensure that bias and discrimination are not in the results.

Legal and Regulatory Considerations: The use of financial dataset should follow the various legal and regulatory requirements, including data protection laws, financial regulations, and intellectual property rights. Some of the regulations are the Payment Card Industry Data Security Standard (PCI DSS), which defines the requirements for the protection of cardholder data, General Data Protection Regulation (GDPR) in the European Union which specifies data privacy requirements (Black and Murray, 2019).

The study will ensure compliance with all applicable laws and regulations and obtain any necessary approvals before conducting research using financial dataset.

Limited Data Availability: Financial dataset may have limitations in terms of size of sample data, data quality, and representation. This study will take considerations of these limitations and acknowledge them. The results gathered from the use of the datasets may not always be generalizable to other environments, and caution must be exercised in interpreting and extrapolating the findings.

In conclusion, this study on how machine learning can be used to detect fraud in financial service operation would be implemented with painstaking consideration of ethical considerations and potential limitations.

The study will ensure privacy and confidentiality, mitigate against biases and comply with all legal and regulatory requirements to execute credible research.

Chapter 4

4.0 Implementation

The project was implemented by carrying out different activities that are categorized as data pre-processing, model training and evaluation. The activities are explained below:

4.1 Data Pre-processing

4.1.1 Data Input

Data is ingested by Python, a popular programming language for data analysis, which reads fraud dataset using Pandas library. Data is loaded into Python using read_csv function and then transformed into data-frames, allowing for manipulation, analysis, and visualization. Python's extensive libraries provide powerful tools for processing and analysing fraud data.

```
#import pandas as pd
df = pd.read_csv("C:/Users/b1666681/OneDrive - Teesside University/Dissertation/Data/Base.csv")
```

Figure 2: Data input using pandas

4.1.2 Column Renaming

The target variable has been renamed to enable easy identification

```
# Rename the 'Class' column to 'Status'
df = df.rename(columns={'fraud_bool': 'Fraud_Status'})
```

Figure 3: Renaming column

4.2 Data Exploration Analysis

Data exploration analysis is an important step in machine learning for detecting financial fraud. It involves analysing the characteristics, patterns, and relationships within the fraud dataset using methods such as descriptive statistics, data visualization, and feature engineering. This process helps the study to gain insights, identify relevant features, pre-process data, and select appropriate algorithms.

```
#Data preview
df.head()
```

	Fraud_Status	income	name_email_similarity	prev_address_months_count	current_address_months_count	customer_age	days_since_request	intended_balco
0	1	0.9	0.166828	-1	88	50	0.020925	
1	1	0.9	0.296286	-1	144	50	0.005418	
2	1	0.9	0.044985	-1	132	40	3.108549	
3	1	0.9	0.159511	-1	22	50	0.019079	
4	1	0.9	0.596414	-1	218	50	0.004441	

5 rows × 32 columns

Figure 4: Viewing rows in the dataset.

4.2.1 Descriptive statistics

It provides the statistical properties of the data and insights into the data's characteristics. This is the summary of basic statistical measures for each numeric column in the Dataframe.

```
# Calculate summary statistics for each feature in the dataset
summary_stats = df.describe()

# Print the summary statistics
print(summary_stats)
```

```
         Fraud_Status          income  name_email_similarity  \
count  1000000.000000  1000000.000000         1000000.000000
mean         0.011029        0.562696               0.493694
std          0.104438        0.290343               0.289125
min          0.000000        0.100000               0.000001
25%          0.000000        0.300000               0.225216
50%          0.000000        0.600000               0.492153
75%          0.000000        0.800000               0.755567
max          1.000000        0.900000               0.999999

       prev_address_months_count  current_address_months_count  \
count              1000000.000000                1000000.000000
mean                    16.718568                     86.587867
std                     44.046230                     88.406599
min                     -1.000000                     -1.000000
25%                     -1.000000                     19.000000
50%                     -1.000000                     52.000000
75%                     12.000000                    130.000000
max                    383.000000                    428.000000

         customer_age  days_since_request  intended_balcon_amount  \
count  1000000.000000        1.000000e+06          1000000.000000
mean        33.689080        1.025705e+00               8.661499
std         12.025799        5.381835e+00              20.236155
min         10.000000        4.036860e-09             -15.530555
25%         20.000000        7.193246e-03              -1.181488
50%         30.000000        1.517574e-02              -0.830507
75%         40.000000        2.633069e-02               4.984176
max         90.000000        7.845690e+01             112.956928

         zip_count_4w     velocity_6h  ...  phone_mobile_valid  \
count  1000000.000000  1000000.000000  ...      1000000.000000
mean      1572.692049     5665.296605  ...            0.889676
std       1005.374565     3009.380665  ...            0.313293
min          1.000000     -170.603072  ...            0.000000
25%        894.000000     3436.365848  ...            1.000000
50%       1263.000000     5319.769349  ...            1.000000
75%       1944.000000     7680.717827  ...            1.000000
max       6700.000000    16715.565404  ...            1.000000
```

Figure 5: Descriptive statistics

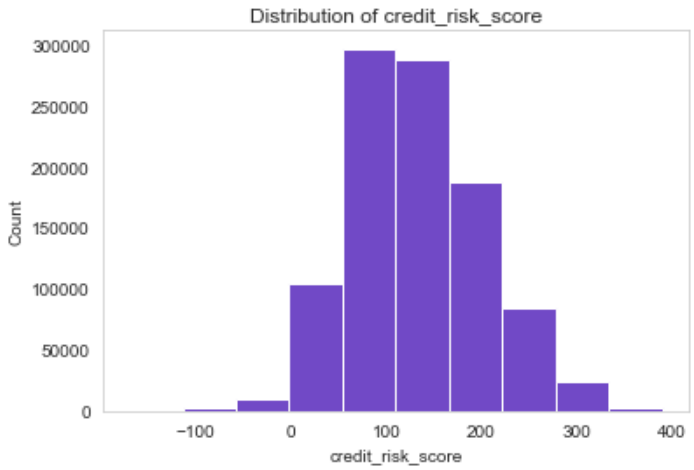

Figure 6: Distribution of the credit risk score

4.2.2 Target Class Analysis

There are 988971 instances of non-fraud cases in the dataset, this represent 98.9% of the data, while there are 11029 instances of fraud cases which represent 1.1% of the data

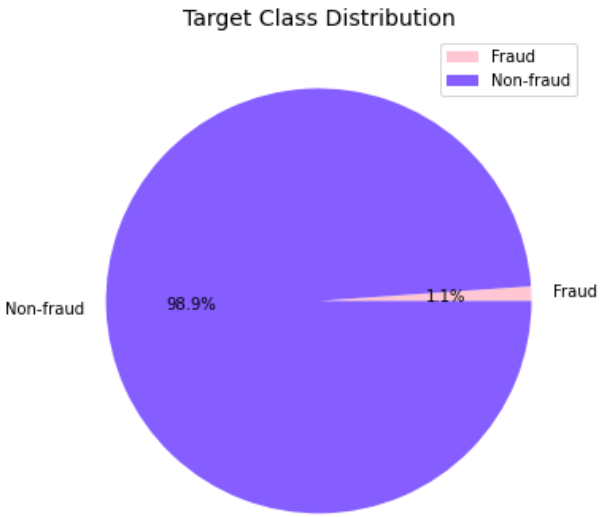

Figure 7: Target Class Distribution

4.2.3 Fraud Analysis

From the 11029 cases of frauds, Internet and Teleapp were the two payment modes used in fraudulent transactions. Internet mode was used in 10917 instances, while Teleapp was used in 112 instances.

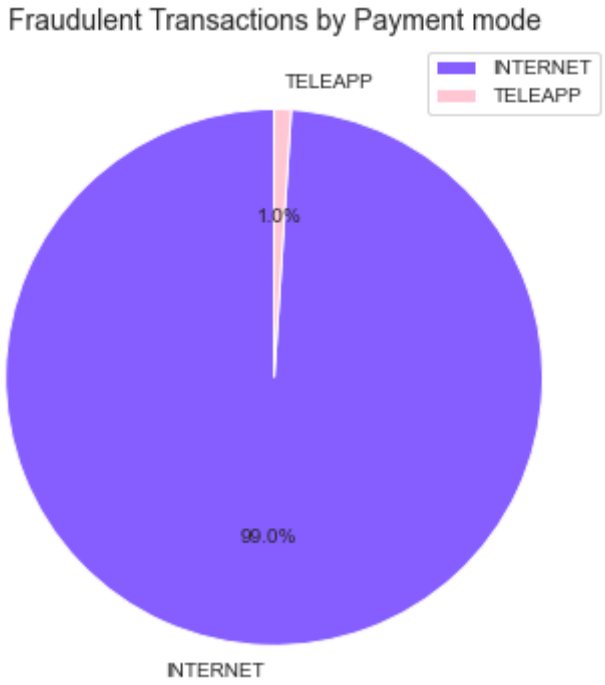

Figure 8: Payment mode used for fraud

4.2.4 Correlation Matrix

A correlation matrix is an important tool in fraud data analysis, it gives insights into the relationship between features. By calculating correlations between different features in a dataset, potential patterns, dependencies, that may indicate fraudulent activities can be discovered. These details help in feature selection, model development, and fraud detection strategy, aiding in the correct prediction of suspicious transactions. From the analysis, proposed_credit_limit, Credit_risk_score, customer_age, current_address_months_count, and income show a higher correlation to the target feature.

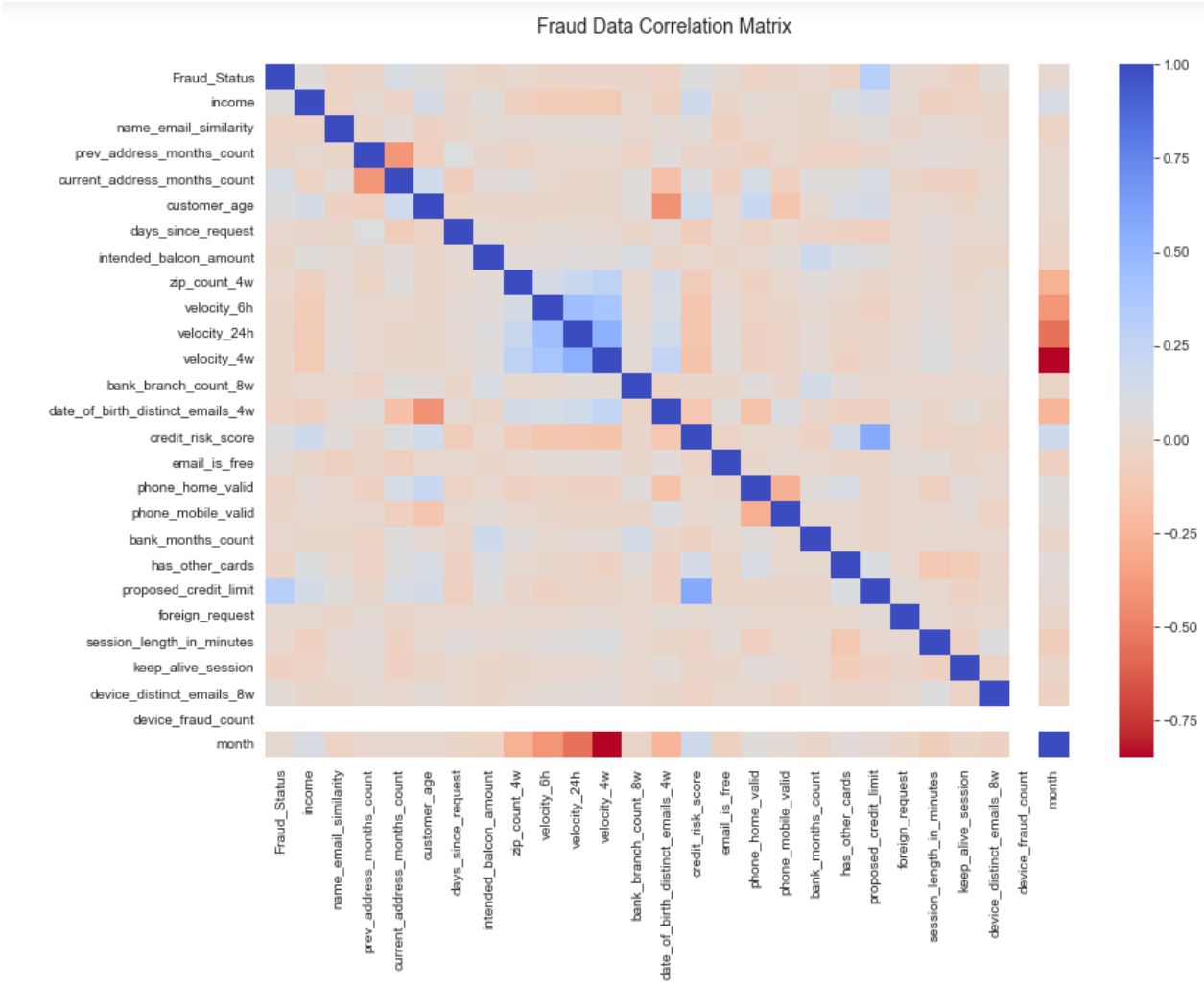

Figure 9: Correlation matrix

4.3 Data Cleaning

4.3.1 Handling missing data

Checking for missing values ¶

```
In [7]: df.isna().sum()

Out[7]: Fraud_Status                        0
        income                              0
        name_email_similarity               0
        prev_address_months_count           0
        current_address_months_count        0
        customer_age                        0
        days_since_request                  0
        intended_balcon_amount              0
        payment_type                        0
        zip_count_4w                        0
        velocity_6h                         0
        velocity_24h                        0
        velocity_4w                         0
        bank_branch_count_8w                0
        date_of_birth_distinct_emails_4w    0
        employment_status                   0
        credit_risk_score                   0
        email_is_free                       0
        housing_status                      0
        phone_home_valid                    0
        phone_mobile_valid                  0
        bank_months_count                   0
        has_other_cards                     0
        proposed_credit_limit               0
        foreign_request                     0
        source                              0
        session_length_in_minutes           0
        device_os                           0
        keep_alive_session                  0
        device_distinct_emails_8w           0
        device_fraud_count                  0
        month                               0
        dtype: int64
```

Figure 10: Verifying for missing data.

4.3.2 Handling Outliers

According to Boukerche, Zheng and Alfandi (2021) an outlier is “an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data”. Outliers are usually defined based on the following assumptions.

i. Outliers are different from the norm with respect to their features.
ii. Outliers are not common in a dataset compared to normal instances.

Outliers causes data discordance in data modelling, therefore there is need to isolate of outliers as this can lead to an improvement in the performance of predictive modelling by offering better data quality and reducing outlier's influence on the model fitting (Su and Tsai, 2011).

Figure 11: Box- plot showing outliers.

Figure 12: Box- plot without outliers.

4.4 Data Encoding

This is the process of transforming categorical data into numerical data for machine learning models to utilize the data. Binary encoding was used because the categorical features are nominal data with no order or ranking.

```
#The categorical data needs to be converted to numerical data
# create a BinaryEncoder object
encoder = ce.BinaryEncoder(cols=['payment_type','employment_status','housing_status','device_os','source'])

# fit and transform the data using the encoder
df1= encoder.fit_transform(df1)
```

Figure 13: Data encoding

4.5 Data Standardisation

The data has been scaled to have a mean of 0 and a standard deviation of 1. ScandardScaler class from scikit-learn was used for the purpose of scaling features to ensure that all the features have similar scale.

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Figure 14: Data Standardisation

4.6 Feature Selection

This study computes the relationship between the target and the other variables in the dataset. Features with low correlation were dropped in order to improve the performance of the model.

Feature Selection

```
In [36]: #Correlation with output variable
         correlation=df.corr()
         cor_target = abs(correlation["Fraud_Status"])
         #Selecting highly correlated features
         relevant_features = cor_target[cor_target < 1]
         relevant_features.sort_values(ascending=False)
```

```
Out[36]: proposed_credit_limit              0.310867
         current_address_months_count      0.117891
         credit_risk_score                 0.070624
         customer_age                      0.061629
         income                            0.058785
         keep_alive_session                0.050296
         date_of_birth_distinct_emails_4w  0.043224
         name_email_similarity             0.036720
         device_distinct_emails_8w         0.035704
         has_other_cards                   0.035156
         phone_home_valid                  0.035128
         email_is_free                     0.027758
         prev_address_months_count         0.026031
         intended_balcon_amount            0.024524
         velocity_6h                       0.018218
         foreign_request                   0.016885
         month                             0.013250
         phone_mobile_valid                0.013180
         bank_branch_count_8w              0.011577
         velocity_4w                       0.011536
         velocity_24h                      0.011183
         zip_count_4w                      0.009539
         bank_months_count                 0.003222
         session_length_in_minutes         0.002233
         days_since_request                0.000567
         Name: Fraud_Status, dtype: float64
```

Figure 15: Feature Selection

```
# Drop the unwanted features columns from the dataframe
df = df.drop(['days_since_request', 'session_length_in_minutes','bank_months_count','zip_count_4w'], axis=1)
```

Figure 16: Dropping unwanted columns.

4.7 Model training

The different selected classifiers are trained using the pre-processed dataset. The dataset is split into training set and test set, the model is trained on the training set by adjusting its parameters iteratively to minimize the prediction errors and the model performance is evaluated on using the test set data.

```
X_train, X_test, Y_train, Y_test = train_test_split(x, y, test_size = 0.3, random_state = 42)
```

Logistic Regression Model

```
]: # Create Logistics Regression classifier
lg = LogisticRegression(solver='liblinear')

# Train the classifier
lg.fit(X_train, Y_train)

# Make predictions on the testing set
y_pred_lg = lg.predict(X_test)

# Evaluate the model
print('Accuracy:', accuracy_score(Y_test, y_pred_lg))
print('Confusion Matrix:', confusion_matrix(Y_test, y_pred_lg))
```

```
Accuracy: 0.99335
Confusion Matrix: [[296627      6]
 [  1989   1378]]
```

Figure 16: Model training using Logistic Regression Classifier

Chapter 5

5.0 Interpretation and discussion of results

The aim of this study is to develop a model that can detect fraudulent transactions, in essence the fraud class (1) is the major class of interest. This indicates the presence of a fraud.

5.1 Accuracy

Accuracy represents the proportion of labels that have been predicted correctly out of the total labels. However, there are more of non-fraud instances in the dataset used when compared to the fraud instances, this leads to class imbalance.

The resultant effect is that the model may have a high accuracy due to the large number of predictions for the majority class.

From the study, Random Forest and Extreme gradient boosting classifiers have the highest accuracy of 99.5%, however this can be linked to the class imbalance observed in the dataset. Oversampling and under sampling techniques were applied to deal with the problem of class imbalance, using smote method, Random Forest has the highest accuracy of 99.4% while Extreme gradient boosting has an accuracy of 89.7% when under-sampling method was used.

Due to class imbalance in the dataset used for this study and the resultant effect, using accuracy as the only basis of evaluating the performance of the model may be misleading. Other evaluation metrics will be used to measure the performance of the algorithms used in this study.

Classifier	Imbalance	Smote	Under_sampling
Logistic Regression Model	0.99335	0.86597	0.8372
Random Forest Classifier	0.99523	0.9946	0.8882
Extreme Gradient Boosting	0.9955	0.99074	0.8966
KNeighbors	0.99346	0.95277	0.8064

Table 3: Accuracy Result

5.2 Precision

This represents the proportion of true positive fraud cases out of the predicted positives. It measures the accuracy of positive predictions made by the model. Using Smote method, Random Forest has a precision of 88% while using Under-sampling Random Forest and Extreme gradient boosting both have a precision of 92%.

Classifier	Imbalance	Smote	Under_sampling
Logistic Regression Model	1	0.06	0.80
Random Forest Classifier	1	0.88	0.92
Extreme Gradient Boosting	1	0.67	0.92
KNeighbors	0.99	0.14	0.81

Table 4: Precision Result

5.3 Recall

Recall or sensitivity is used to measure the effectiveness of the model in identifying positive instances from the data.

Using smote method, Logistic regression has 80% recall rate, Extreme Gradient Boosting has a recall rate of 87% when used on an under-sampled data. The smote and under-sampling methods produced better results when compared to 66% recall rate obtained from imbalanced dataset.

Classifier	Imbalance	Smote	Under_sampling
Logistic Regression Model	0.41	0.80	0.79
Random Forest Classifier	0.58	0.60	0.84
Extreme Gradient Boosting	0.66	0.67	0.87
KNeighbors	0.42	0.61	0.79

Table 5: Recall Result

5.4 F1-Score

F1-score gives a balanced measure of the performance of the model, it is used to measure the trade-off between precision and recall. Identifying a non-fraudulent transaction as being fraudulent may lead to bad customer satisfaction as transaction will be unsuccessful and can lead to some reputational issue for financial institutions and identifying a fraudulent transaction as being non-fraudulent will lead to financial losses.

Extreme Gradient Boosting has the best F1-score of 90% using the under-sampling method, while Random Forest produced the best score of 71% when applied with oversampled data.

Classifier	Imbalance	Smote	Under_sampling
Logistic Regression Model	0.58	0.12	0.80
Random Forest Classifier	0.73	0.71	0.88
Extreme Gradient Boosting	0.75	0.62	0.90
KNeighbors	0.59	0.23	0.80

Table 6: F1-Score Result

5.5 ROC AUC

The ROC AUC measures a model's performance by considering both sensitivity and specificity, it is useful in imbalanced dataset where the target class have uneven representation. They also provide insights into the model's ability to discriminate between positive and negative instances at different classification thresholds, which can help in decision-making and model selection.

The Extreme Gradient Boosting classifier has the best F1-score of 97% when evaluated using the under-sampled data method.

Classifier	F1-Score	Classifier
Imbalance	0.97	Extreme Gradient Boosting
SMOTE	0.96	Extreme Gradient Boosting
Under-sampling	0.97	Extreme Gradient Boosting

Table 7: ROC AUC Result

5.6 False Alarm Rate:

False alarm rate is the rate at which non-fraud instances are incorrectly flagged as fraudulent leading to investigations which are not required. The objective is to accurately predict the true fraud cases while minimizing false positives. A high false

Extreme Gradient Boosting implemented with smote gave the lowest FAR of 0.005, however a low false alarm rate may be as a result of some fraud cases being missed which can lead to financial losses and reputational demand. It is imperative to constantly review fraud prediction system to as fraud patterns evolve constantly

Classifier	F1-Score	Classifier
Imbalance	0	Extreme Gradient Boosting
Smote	0.005	Extreme Gradient Boosting
Under-sampling	0.07	Extreme Gradient Boosting

Table 8: False Alarm Rate Result

5.7 Balanced Classification Rate

Balanced Classification Rate represents a balanced trade-off between sensitivity (the true positive rate) and specificity (the true negative rate) in identifying both fraudulent and non-fraud transactions.

In this study, achieving high sensitivity is important to detect as many true fraud cases as possible, while high specificity is equally important to minimize false positives or false alarms. BCR gives a balance between the two goals as this provides the model's capability to prevent fraud and non-fraud. Extreme Gradient Boosting classifier using under-sampled data gave the best rest of 90%. Finding the right middle ground is key to optimising the performance of fraud prediction.

	BCR	Classifier
Imbalance	0.80	Extreme Gradient Boosting
SMOTE	0.83	Extreme Gradient Boosting
Under-sampling	0.90	Extreme Gradient Boosting

Table 9: Balanced Classification Rate Result

5.8 Analysis of result

The result of the study on using machine learning to detect fraud in financial services operation was highly promising. Using XGBoost classifier returned the best result, the classifier achieved a precision of 92%, recall of 87%, and an F1-score of 90%. These metrics indicate the ability of the classifier to accurately identify fraudulent activities while minimizing false positives and false negatives. The Receiver Operating Characteristic Area Under the Curve (ROC-AUC) was 97%, which further proves the classifier's effectiveness in distinguishing between fraudulent and non-fraudulent transactions. The false alarm rate was impressively low at 0.07%, indicating the classifier's ability to minimize false positives, this has the potential to save financial institutions from unnecessary investigation costs. The Balanced Classification Rate (BCR) was also high at 90%, which demonstrates the classifier's ability to maintain a balance between precision and recall.

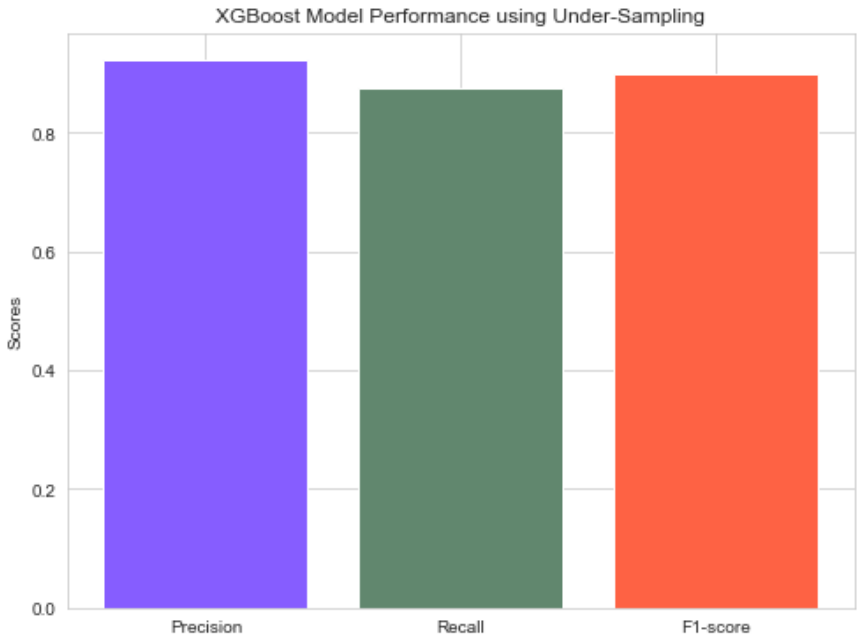

Figure 17: XGBoost Model Result

Chapter 6

6.1 Interpretation and discussion of results in relation to existing literature

In a study to predict credit card fraud by Xu, Fan and Song (2022) that deployed the use of ensemble and single machine learning algorithms, gradient boosting decision tree gave an AUC value of 90%, this project also used both ensemble and single classifiers with an ensemble classifier (Extreme gradient boosting) producing the best ROC-AUC value of 97%. This shows that ensemble classifier performs better than single classifiers.

Support vector machine (SVM) model was used to detect credit card fraud and reduce false alarm in a study using a dataset with adequate proportion of fraud and non-fraud instances, the study produced an accuracy of 94.4%. Kamboj and Shankey (2016), in comparison this current study produced an accuracy of 90% after the dataset has been under-sampled, this project however will be evaluating the performance based on other metrics aside accuracy.

Zarepoor and Shamsolmoali (2015) in a study to evaluate the performance of three advanced data mining techniques for credit card detection. The dataset was trained using different classifiers with the bagging ensemble classifier. The decision tree model with bagging ensemble returned a higher fraud catching rate and a low false alarm rate when compared with other models that returned a higher false alarm rate, it also showed a better performance with highly imbalanced dataset. This current project used a boosting ensemble classifier that also produced a low false alarm rate of 0.07%.

6.2 Analysis of implications, significance, and limitations of the research

There are some issues that were encountered in the process of carrying out this study on using machine learning for fraud prevention in financial services operations:

1. Data availability and quality:

Financial data are usually not readily available as it contains many sensitive information this led to the use of synthetic data for this study. The effectiveness of machine learning models depends on the quality and representativeness of the available data used for training. Data which is not closely related to real life data can lead to a bias.in the prediction.

2. Class imbalance:

There are more non-fraud instances in the dataset compared to the ratio of fraud instances. This created a problem of class imbalance which can cause a bias in the model's performance.

3. Model updates: It is important to regularly update machine learning models used for fraud predication because fraud patterns change constantly. Ensuring continuous model monitoring and maintenance is essential to maintaining the effectiveness of the model in predicting fraud.

4. Ethical considerations: Using machine learning for fraud prevention comes with some ethical considerations, such as potential biases in the models, fairness in model predictions, and privacy concerns with the use of sensitive financial data.

The model must be designed with necessary steps put in place to ensure that fairness assessment, bias mitigation, and data privacy protection measures are put in place, these are important factors in building trust and complying with regulatory requirements.

6.3 Suggestions for future research and potential areas of improvement

There are several future research areas that could be explored in using machine learning for fraud prevention in financial service operations:

- Handling evolving fraud patterns: Future research can explore methods to adapt machine learning model to fraud patterns that constantly change over time, such as online learning techniques or incorporating temporal features to capture time-dependent patterns.

- Robustness against adversarial attacks: Adversarial attacks such as data poisoning attacks or evasion attacks can be used by fraudsters to manipulate the behaviour of machine learning models to avoid detection. Future research could examine the strength of fraud detection models against such attacks and develop techniques to improve the security of fraud detection models against sophisticated attacks.
- Generalizability: The findings of the analysis may not be generalizable to all financial service operations or datasets. Future research could explore the applicability and generalizability of machine learning for fraud prevention in different financial sectors, such as credit card fraud, insider trading, or insurance fraud, and assess the model's performance in various contexts.

In summary, future research could focus on handling evolving fraud patterns, developing real-time fraud detection systems and assessing generalizability to further enhance the effectiveness and responsible use of machine learning for fraud prevention in financial service operations.

6.4 Conclusion

In conclusion, deploying machine learning models to prevent fraud in financial service operations has proven to be an assuring method. In the face of current realities of increasing complexities and sophistication of fraud activities, rule-based model which is the traditional approach used to combat this problem sometimes can be grossly inadequate and insufficient in detecting frauds. Machine learning models, however, can analyse big data, detect patterns in data and readjust to ever-changing pattern in fraud activities, therefore in financial service operations machine learning is an effective and efficient tool in fraud detection.

One of the goals of the study was to determine the best performing algorithm, this was an important factor in the choice of classifier. A combination of single classifiers was used as well as ensemble learning algorithms. Using the Bank Account Fraud Dataset Suite, the study trained the data on four algorithms namely, logistic regression, random forest, K-nearest neighbor and extreme gradient boosting.

The use of Extreme Gradient Boosting (XGBoost) classifier as an ensemble learning technique, has shown significant promise in preventing fraud in financial service operations. It produced the best performing metrics when compared with other classifier deployed, it gave a precision of 92%, recall of 87%, F1-score of 90%, ROC-AUC of 97%, false alarm rate of 0.07, and Balanced Classification Rate (BCR) of 90%. The application of under-sampling method to fix class imbalance further improved the model's performance.

The findings of this study highlight the effectiveness of the XGBoost classifier in detecting and preventing fraud in financial service operations. The high precision rate of 92% and recall rate of 87% indicate that the model can accurately identify instances of fraud while minimizing false positives and false negatives. The F1-score of 90%, provides a balanced evaluation of the model's performance, this further proves the model's accuracy and consistency in fraud detection. The ROC-AUC score of 97% indicates the model's strong predictive power in distinguishing between fraud and non-fraud cases. The low false alarm rate of 0.07% also suggests that the model has a low rate of flagging non-fraudulent transactions as fraudulent, reducing inconveniences for legitimate customers. The BCR of 90% further validates the model's performance in handling both classes, even with class imbalance.

However, further research and continuous monitoring and improvement of the model's performance are necessary to ensure its effectiveness in real-world scenarios. Machine learning models, when used in conjunction with other preventive measures, can be a valuable tool in mitigating fraud risks and safeguarding the integrity of financial service operations.

References

Abd, S. and Abraham, A. (2013). A Review of Class Imbalance Problem. *Journal of Network and Innovative Computing*, [online] 1, pp.332–340. Available at: http://ias04.softcomputing.net/jnic2.pdf. [Accessed: 12 April 2023]

Alfaro-Navarro, J.-L., Cano, E.L., Alfaro-Cortés, E., García, N., Gámez, M. and Larraz, B. (2020). *A Fully Automated Adjustment of Ensemble Methods in Machine Learning for Modeling Complex Real Estate Systems. Complexity, 2020*, pp.1–12. Available at: https://doi.org/10.1155/2020/5287263. [Accessed: 25 April 2023]

Ao, Y., Li, H., Zhu, L., Ali, S. and Yang, Z. (2019). The linear random forest algorithm and its advantages in machine learning assisted logging regression modeling. *Journal of Petroleum Science and Engineering*, [online] 174, pp.776–789. Available at: https://doi.org/10.1016/j.petrol.2018.11.067. [Accessed: 9 April 2023]

Barboza, F., Kimura, H. and Altman, E. (2017). Machine learning models and bankruptcy prediction. *Expert Systems with Applications*, 83, pp.405–417. Available at: https://doi.org/10.1016/j.eswa.2017.04.006. [Accessed: 10 April 2023]

Belete, D.M. and Huchaiah, M.D. (2021). Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results. *International Journal of Computers and Applications*, pp.1–12. Available at: https://doi.org/10.1080/1206212x.2021.1974663. [Accessed: 24 April 2023]

Black, J. and Murray, A.D. (2019). Regulating AI and Machine Learning: Setting the Regulatory Agenda. European Journal of Law and Technology, [online] 10(3). Available at: https://www.ejlt.org/index.php/ejlt/article/view/722/980 [Accessed 8 May 2023].

Boukerche, A., Zheng, L. and Alfandi, O. (2021). Outlier Detection. *ACM Computing Surveys*, 53(3), pp.1–37. Available at: https://doi.org/10.1145/3381028. [Accessed: 15 April 2023]

Breiman, L. Random Forests. *Machine Learning* **45**, 5–32 (2001). Available at: https://doi.org/10.1023/A:1010933404324. [Accessed: 9 April 2023]

Brown, S. (2021). *Machine learning, explained*. [online] MIT Sloan. Available at: https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained. [Accessed: 10 April.2023]

Buda, M., Maki, A. and Mazurowski, M.A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106, pp.249–259. Available at: https://doi.org/10.1016/j.neunet.2018.07.011. [Accessed: 12 April 2023]

Cai, J., Luo, J., Wang, S. and Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, [online] 300, pp.70–79. Available at:https://doi.org/10.1016/j.neucom.2017.11.077. [Accessed: 12 April 2023]

Carcillo, F., Le Borgne, Y.-A., Caelen, O., Kessaci, Y., Oblé, F. and Bontempi, G. (2019). Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences*. Available at: https://doi.org/10.1016/j.ins.2019.05.042. [Accessed: 21 April 2023]

Chen, T. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pp.785–794. Available at: https://doi.org/10.1145/2939672.2939785. [Accessed: 10 April 2023]

Choi, D. and Lee, K. (2018). An Artificial Intelligence Approach to Financial Fraud Detection under IoT Environment: A Survey and Implementation. *Security and Communication Networks*, [online] 2018, pp.1–15. Available at: https://doi.org/10.1155/2018/5483472. [Accessed: 13 March 2023]

Dahouda, M.K. and Joe, I. (2021). A Deep-Learned Embedding Technique for Categorical Features Encoding. *IEEE Access*, [online] 9, pp.114381–114391. Available at: https://doi.org/10.1109/ACCESS.2021.3104357. [Accessed: 12 April 2023]

Dornadula, V.N. and Geetha, S. (2019). Credit Card Fraud Detection using Machine Learning Algorithms. *Procedia Computer Science*, 165, pp.631–641. Available at: https://doi.org/10.1016/j.procs.2020.01.057. [Accessed: 17 March 2023]

GhoshRoy, D., Alvi, P.A. and Santosh, K. (2022). Explainable AI to Predict Male Fertility Using Extreme Gradient Boosting Algorithm with SMOTE. *Electronics*, 12(1), p.15. Available at: https://doi.org/10.3390/electronics12010015. [Accessed: 9 April 2023]

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media. Available at: https://link.springer.com/chapter/10.1007/978-0-387-84858-7_1 [Accessed: 9 April 2023]

Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression (Vol. 398). John Wiley & Sons.

Lee, I. and Shin, Y.J. (2019). Machine learning for enterprises: Applications, algorithm selection, and challenges. *Business Horizons*, 63(2). Available at: https://doi.org/10.1016/j.bushor.2019.10.005. [Accessed: 13 April 2023]

Li, Y., Daochen Z, Praveen, K.V, Zou, N. and Hu, X. (2020). PyODDS: An End-to-end Outlier Detection System with Automated Machine Learning. *Companion Proceedings of the Web Conference 2020*. Available at: https://doi.org/10.1145/3366424.3383530. [Accessed: 13 April 2023]

Mary-Jo Kranacher and Riley, R. (2020). *Forensic accounting and fraud examination*. Hoboken, Nj: John Wiley & Sons, Inc. 2nd Ed

M. Kamboj and G. Shankey, “Credit Card Fraud Detection and False Alarms Reduction using Support Vector Machines,” International Journal ofAdvance Research, Ideas and Innovations in Technology, vol. 2, no. 4, 2016. Available at: https://www.ijariit.com/manuscripts/v2i4/V2I4-1145.pdf [Accessed: 13 March 2023]

Nick, T.G. and Campbell, K.M. (2007). Logistic Regression. *Topics in Biostatistics*, pp.273–301. Available at: https://doi.org/10.1007/978-1-59745-530-5_14. [Accessed:13 March. 2023]

Preece, A.D., Shinghal, R. and Batarekh, A. (1992). Principles and practice in verifying rule-based systems. *The Knowledge Engineering Review*, 7(2), pp.115–141. Available at: https://doi.org/10.1017/s026988890000624x. [Accessed: 10 April 2023]

Rtayli, N. and Enneya, N. (2020). Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization. *Journal of*

Information Security and Applications, 55, p.102596. Available at: https://doi.org/10.1016/j.jisa.2020.102596. [Accessed: 8 April 2023]

Shokrzade, A., Ramezani, M., Akhlaghian T, F. and Abdulla Mohammad, M. (2021). A novel extreme learning machine based kNN classification method for dealing with big data. *Expert Systems with Applications*, 183, p.115293. Available at: https://doi.org/10.1016/j.eswa.2021.115293. [Accessed: 10 April 2023]

Su, X. and Tsai, C. (2011). Outlier detection1. *WIREs Data Mining and Knowledge Discovery*, 1(3), pp.261–268. Available at: https://doi.org/10.1002/widm.19. [Accessed: 15 April 2023]

Tae, C.M. and Hung, P.D. (2019). Comparing ML Algorithms on Financial Fraud Detection. *Proceedings of the 2019 2nd International Conference on Data Science and Information Technology*. Available at: https://doi.org/10.1145/3352411.3352416. [Accessed: 8 April 2023]

Thennakoon, A., Bhagyani, C., Premadasa, S., Mihiranga, S. and Kuruwitaarachchi, N. (2019). *Real-time Credit Card Fraud Detection Using Machine Learning*. [online] IEEE Xplore. Available at: https://doi.org/10.1109/CONFLUENCE.2019.8776942. [Accessed: 15 March 2023]

ukfinance (2021). *FRAUD -THE FACTS 2021 THE DEFINITIVE OVERVIEW OF PAYMENT INDUSTRY FRAUD*. [online] Available at: https://www.ukfinance.org.uk/system/files/Fraud%20The%20Facts%202021-%20FINAL.pdf. [Accessed: 10 April 2023]

Xu, H., Fan, G. and Song, Y. (2022). Application Analysis of the Machine Learning Fusion Model in Building a Financial Fraud Prediction Model. *Security and Communication Networks*, 2022, pp.1–13. Available at: https://doi.org/10.1155/2022/8402329. [Accessed: 21 April 2023]

Yang, L. and Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415, pp.295–316. Available at: https://doi.org/10.1016/j.neucom.2020.07.061. [Accessed: 18 April 2023]

Zareapoor, M. and Shamsolmoali, P. (2015). Application of Credit Card Fraud Detection: Based on Bagging Ensemble Classifier. *Procedia Computer Science*, [online] 48, pp.679–685. Available at: https://doi.org/10.1016/j.procs.2015.04.201. [Accessed: 14 March 2023]