National College of Ireland

Credit Card Fraud Detection using Ensemble Learning Algorithms

MSc Research Project
MSc Data Analytics

Eva Figuerola Ullastres
Student ID: x19209371

School of Computing
National College of Ireland

Supervisor: Dr. Majid Latifi

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Eva Figuerola Ullastres
Student ID:	x19209371
Programme:	MSc Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Dr. Majid Latifi
Submission Due Date:	30/05/2022
Project Title:	Credit Card Fraud Detection using Ensemble Learning Algorithms
Word Count:	8565
Page Count:	26

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	30th May 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Credit Card Fraud Detection using Ensemble Learning Algorithms

Eva Figuerola Ullastres
x19209371

Abstract

Credit card fraud is a type of financial crime where fraudsters use people's credit card details to purchase goods and services without the permission of the cardholder. The volume of online transactions has grown substantially in recent years, which has led to an increase in fraud attempts. Businesses lose billions due to fraud every year. Since only a small percentage of credit card transactions are fraudulent, fraud datasets are highly imbalanced, making fraud detection a challenging task. Machine learning plays an important role in credit card fraud detection. This study examines the performance of tree-based ensemble methods in detecting fraudulent transactions. Different classifiers; Random Forest, Bagging, XGBoost, LightGBM and CatBoost are implemented in this research. A hybrid sampling approach of random undersampling (RUS) and Borderline-SMOTE is used to handle the class imbalance. The results indicate that boosting classifiers outperformed bagging classifiers in detecting fraud. XGBoost and LightGBM achieved the best performance in terms of F1-Score (0.70), Matthews Correlation Coefficient (MCC) (0.71) and Area Under the Precision-Recall Curve (AUC-PR) (0.73).

keywords – Credit Card Fraud, Machine Learning, Class Imbalance, Ensemble Methods, XGBoost, LightGBM, RUS, BorderlineSMOTE.

1 Introduction

1.1 Background and Motivation

"Fraud is an uncommon, well-considered, imperceptibly concealed, time-evolving and often carefully organized crime which appears in many types and forms" (Baesens, Van Vlasselaer and Verbeke; 2015, p.3).

In today's digital world, credit cards are one of the most common payment method for online purchases. Credit card usage has risen sharply in recent years and the volume of fraudulent transactions has also increased. Credit card fraud is a serious and growing problem that not only affects the financial sector but also the global economy. According to the most recent Nilson Report data, credit card fraud losses reached $28.58 billion worldwide in 2020 [1], compared to $23.97 billion in 2017 [2]. This emphasises the importance of the early detection of fraudulent transactions.

Credit card fraud detection is a key priority for financial institutions. Although both supervised and unsupervised machine learning techniques can be used to detect credit

[1] `https://nilsonreport.com/mention/1515/1link/`
[2] `https://nilsonreport.com/mention/1313/1link/`

card fraud, this research only focuses on supervised classification techniques. Supervised fraud detection is the process of classifying a transaction as fraudulent or legitimate based on historical data. Despite fraudulent transactions being infrequent, failing to detect them can result in significant financial losses. Furthermore, misclassifying a genuine transaction as fraudulent has also a high cost associated, as it damages the brand's reputation and drives away loyal customers; hence the importance of having an effective credit card fraud detection system in place in order to accurately detect fraudulent transactions. By using classification algorithms that rely on historical data to identify patterns and anomalies, a potentially fraudulent transaction can be detected before it is completed. A supervised credit card fraud detection framework based on data mining and machine learning is shown in Figure 1.

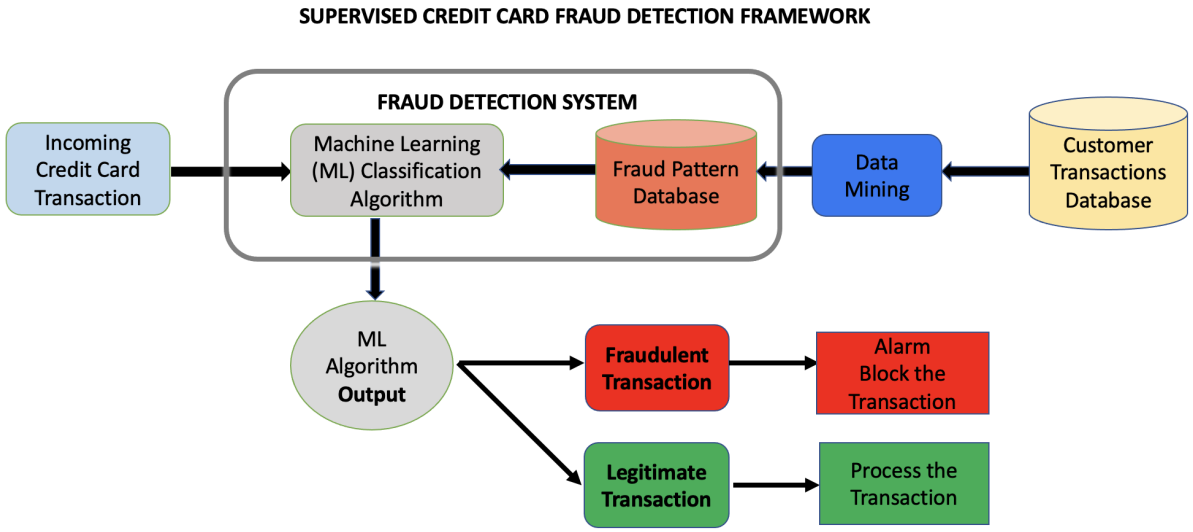

Figure 1: Supervised Credit Card Fraud Detection Framework

The motivation for this research is to create a predictive model to help financial institutions identify potentially fraudulent transactions. Achieving a low percentage of false positives and false negatives is a key challenge in fraud detection. Since the majority of credit card transactions are legitimate and only a small percentage are fraudulent, credit card fraud datasets are highly imbalanced, making more challenging the process of building reliable credit card fraud detection models. Another challenge in fraud detection is the constantly evolving fraud patterns and consumer's behaviour (concept drift). Because fraudsters are continuously looking for new and innovative ways to commit fraud, cutting-edge technologies must be used to keep pace with changing fraud trends and consumer shopping habits. The lack of publicly available real data in the credit card fraud domain is the main limitation in fraud detection research; instead, synthetic data is used.

1.2 Research Question, Objectives and Contributions

The goal of this research is to explore how tree-based ensemble algorithms can successfully be used to predict fraud. This research paper aims to answer the following question:

RQ: *"To what extent can tree-based ensemble learning algorithms correctly classify credit card transactions as fraudulent or legitimate using historical transactional data?"*

In order to address the research question, the specific sets of research objectives are defined in Table 1.

Table 1: Research Objectives

Objectives	Description
Investigation	**1.** Critically review the most recent literature on credit card fraud detection and identify research gaps.
Methodology	**2.** Propose a research methodology for credit card fraud detection.
Implementation	**3.** Exploratory data analysis and extensive data preprocessing. **4.** Adopt a sampling approach to handle the class imbalance. **5.** Build different classifiers using the following ensemble algorithms: **a)** Bagging, **b)** Random Forest, **c)** XGBoost, **d)** LightGBM, **e)** CatBoost.
Evaluation	**6.** Assess and select relevant key performance metrics based on the dataset and desired outcomes. **7.** Evaluate and compare the performance of the different models and select the best performing model. **8.** Identify the most important factors for predicting credit card fraud.

Due to the lack of publicly available fraud datasets, most of the current research on fraud detection is based on the European dataset, which only contains numerical input variables. The most significant contribution of this research is a detailed analysis of the performance of tree-based ensemble classifiers in fraud detection using a massive highly imbalanced dataset that contains a mix of continuous and categorical attributes. This research also identifies the most important factors for predicting credit card fraud.

1.3 Structure of the Report

The rest of the document is structured as follows: Section 2 contains a critical review of the current research on credit card fraud detection. Section 3 proposes a research methodology in order to answer the research question along with a detailed explanation of the different phases. Section 4 illustrates and explains the 2-tier architecture design approach adopted to conduct this project. Section 5 discusses the implementation of the different models. After that, Section 6 presents and evaluates the results and findings from the experiments along with a detailed discussion of the results. Lastly, Section 7 synthesizes the key points of this research and makes recommendations for future work.

2 Related Work

Fraudulent transactions pose a major threat to financial institutions; therefore, a great deal of research has been done on credit card fraud detection. This section provides a critical review of the state-of-the-art research that focuses on credit card fraud detection, and is divided into three subsections: 2.1. Research Dealing with the Class Imbalance Problem, 2.2. Machine Learning Algorithms for Fraud Detection, and lastly, 2.3. Summary and Research Gaps.

2.1 Research Dealing with the Class Imbalance Problem

Credit card fraud datasets are highly imbalanced since the majority of transactions are legitimate. This causes the classifier being biased towards the majority class, resulting in a poor predictive performance for the minority class (fraudulent transactions). In order to handle the class imbalance, sampling methods and class weight optimisation are frequently used. The performance of a classifier is greatly affected by the sampling approach applied on the dataset to balance the class distribution. Mishra and Ghorpade (2018) conducted a study with the aim to improve the fraud detection rate on the highly imbalanced European credit card fraud dataset. They used random undersampling (RUS), which consists in removing observations from the majority class to reduce the class imbalance, and compared the performance of Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), Stacked classifiers and Gradient Boosting Machine (GBM) in detecting fraud. Their results showed that RUS improved the fraud detection rate (recall) of the classifiers, and RF achieved the highest recall (96%) (Mishra and Ghorpade; 2018). However, undersampling may cause a loss of valuable information which would lead to underfitting of the model. On the other hand, Varmedja et al. (2019) conducted an experiment on the European dataset and compared the performance of LR, RF, Naive Bayes (NB) and Multilayer Perceptron (MLP) using the Synthetic Minority Oversampling Technique (SMOTE), which consists in generating synthetic observations for the minority class to reduce the class imbalance. Their results showed that RF achieved the best performance in terms of balance between precision (96.38%) and recall (81.63%); and it was concluded that SMOTE improves the performance of the classifiers (Varmedja et al.; 2019). However, in highly skewed datasets, such as fraud datasets, SMOTE may overgeneralise the minority class. In order to address this problem, different SMOTE extensions can be applied. Taneja, Suri and Kothari (2019) conducted research using the European dataset and compared different SMOTE-based techniques in conjunction with various ensemble classifiers. Their results showed that SVM SMOTE with RF achieved the best performance in terms of recall (80%), precision (91%) and F1-Score (0.85); but it was argued that SVM SMOTE has a higher computational cost compared to other sampling techniques (Taneja et al.; 2019).

Not all sampling approaches affect classifiers in the same way. Muaz, Jayabalan and Thiruchelvam (2020) conducted a study on the European dataset and compared several sampling techniques over different classifiers. Their results showed that SMOTE with RF obtained the highest recall of 81% and a precision of 86%. In comparison with Varmedja et al. (2019), Muaz et al. (2020) demonstrated that SMOTE with RF achieves promising results in credit card fraud detection. Sisodia, Reddy and Bhandari (2017) also analysed the European dataset and evaluated the impact of different sampling methods on the performance of some cost-sensitive and ensemble classifiers. Their experiments demonstrated that when oversampling was used, Bagging outperformed the other classifiers; but when undersampling was used, SVM outperformed the others. It was concluded that SMOTE-ENN, which uses Edited Nearest Neighbors (ENN) undersampling to remove noisy instances from the training data, achieved the best performance in detecting fraud when used in conjunction with ensemble classifiers (Sisodia et al.; 2017). A hybrid sampling approach was also used by Shamsudin et al. (2020), who analysed the European dataset and evaluated the performance of RF using different sampling strategies. They argued that combining RUS with oversampling helps to overcome the main limitation of RUS (loss of information) and that in large datasets the use of oversampling techniques

alone is not sufficient to sample the data as it might lead to overfitting and overgeneralisation in the model. Their results showed that a hybrid of RUS + SVM SMOTE obtained the best results in terms of F1-Score (80%) (Shamsudin et al.; 2020). In comparison with Sisodia et al. (2017), they demonstrated that a hybrid of SMOTE and undersampling achieves promising results when combined with ensemble classifiers. Singh, Ranjan and Tiwari (2021) also found that in extremely imbalanced credit card fraud datasets, a hybrid of oversampling and undersampling performs well when used with ensemble classifiers. Metaheuristic techniques have also been used to deal with highly skewed class distributions. Benchaji, Douzi and El Ouahidi (2019) demonstrated that grouping the minority class using K-means clustering and then applying genetic algorithms to generate new samples from the minority class is a suitable method to deal with the class imbalance in fraud datasets.

Apart from sampling techniques, there are other alternatives that can also be used to deal with imbalanced binary classification problems. Cost-sensitive learning takes the misclassification costs into account when training a model on datasets with a skewed class distribution. Sahu, GM and Gourisaria (2020) analysed the European dataset and built several classifiers for fraud detection using different methods to overcome the class imbalance; in the first method the minority class was oversampled, while in the second method a cost-sensitive approach giving a higher weight to the minority class was adopted. The results of their experiments revealed that both techniques achieved good results and RF outperformed the other classifiers achieving a F1-Score of 95.21% and 92.03% in the first and second approach, respectively.

In order to preserve the class proportions when splitting an imbalanced dataset into train and test sets, stratified sampling, which splits the classes proportionally into training and test sets, should be applied. Huang et al. (2021) analysed the European dataset and split the dataset in a stratified manner and then applied SMOTE on the training set in order to mitigate the effects of the class imbalance before building several tree-based models. It can be concluded that SMOTE is a widely used sampling approach to deal with highly imbalanced data and that hybrid sampling techniques have proven to be effective when used in conjunction with ensemble classifiers.

2.2 Machine Learning Algorithms for Fraud Detection

The choice of a machine learning algorithm is primarily driven by the type of input data. Nguyen et al. (2020) conducted research using different credit card datasets and argued that the number of attributes, the number of transactions and the correlation between attributes determine the model's performance in credit card fraud detection. An extensive research using the European dataset was conducted by Niu, Wang and Yang (2019). They compared the performance between multiple models in detecting fraudulent transactions using the AUC-ROC as evaluation metric, and concluded that supervised methods slightly outperformed unsupervised methods. XGBoost and RF achieved the best performance (AUC=0.989) among supervised models (Niu et al.; 2019). However, Saito and Rehmsmeier (2015) argued that when evaluating binary classifiers on highly imbalanced datasets, the ROC curve can be misleading as it is sensitive to the class imbalance; the Precision-Recall (PR) curve, instead, is more informative and hence a better performance metric for credit card fraud detection (Saito and Rehmsmeier; 2015).

Ensemble classifiers have proven to perform better than other state-of-the-art classifiers in credit card fraud detection. Dhankhad, Mohammed and Far (2018) compared the

performance of LR, RF, Stacking classifiers (SC), XGBoost, KNN, GBM, MLP, SVM, Decision Tree (DT) and NB using the European dataset. SC with LR as meta-classifier, RF and XGBoost achieved the best results (F1-score= 0.95); thus, they concluded that ensemble methods improve the performance of fraud detection models. Husejinovic (2020) also used the European dataset and compared the performance of NB, DT and Bagging in detecting fraud. The results indicated that Bagging with a DT as base learner achieved the best performance with a AUC-PR score of 0.825 followed by DT (0.745), whereas NB achieved a poor predictive performance (0.080) (Husejinovic; 2020). In comparison with Dhankhad et al. (2018) they demonstrated that tree-based ensemble methods perform well in fraud detection. A real e-commerce transactions dataset was used by Zareapoor and Shamsolmoali (2015) to investigate the performance of NB, SVM, KNN and bagged trees in fraud detection. Their results showed that Bagging with DT was the best performing classifier, achieving the highest Matthews Correlation Coefficient (MCC), the highest fraud detection rate and the lowest false alarm rate. It was also argued that Bagging has the capability to handle the class imbalance (Zareapoor and Shamsolmoali; 2015); and in comparison with Husejinovic (2020) they found that Bagging with DT as weak learner achieves promising results in credit card fraud detection.

Along with bagging, boosting is a popular ensemble learning method. In order to determine the best boosting algorithm to predict credit card fraud, Divakar and Chitharanjan (2019) used the European dataset and evaluated the performance of AdaBoost, GBM and XGBoost. They found that XGBoost (F1-Score=0.88) outperformed AdaBoost (F1-Score= 0.76) and GBM (F1-Score= 0.74). These results suggest that boosting algorithms are powerful techniques to detect credit card fraud and reflect those of Dhankhad et al. (2018) who also found that XGBoost performs well in credit card fraud detection. The rapid evolution of fraud patterns (concept drift) may deteriorate the performance of the state-of-the-art fraud detection models over time. Bayram, Köroğlu and Gönen (2020) used a dataset that contained transactions over a 4-month period and proposed a card-based incremental XGBoost model that not only detected fraudulent transactions in real time but also adapted to drifts in online transactions while preserving the knowledge from past transaction patterns. The proposed model achieved a higher Area Under the Curve (AUC) score (0.96) than the static XGBoost model (0.91).

A study conducted by Fang, Zhang and Huang (2019) compared the effectiveness of LightGBM with RF and GBM in detecting fraud. Despite all models performed well on different datasets, LightGBM slightly outperformed RF and GBM in terms of AUC score and training time; thus, they concluded that LightGBM is very effective and efficient in detecting fraud. Taha and Malebary (2020) also analysed different datasets and proposed an optimised LightGBM approach to detect fraud, in which a Bayesian-based hyperparameter optimisation algorithm was used to tune the parameters of the model. The results revealed that the proposed model outperformed other state-of-the-art classifiers in terms of AUC and F1-Score; it was also emphasised the importance of hyperparameter tuning in order to improve the predictive performance of the model (Taha and Malebary; 2020). Unlike other boosting algorithms, CatBoost can automatically handle categorical features. Hancock and Khoshgoftaar (2020) evaluated the performance of CatBoost and XGBoost in detecting fraudulent medical insurance claims using two datasets that contained a mix of numerical and categorical variables. They demonstrated that CatBoost outperforms XGBoost when dealing with high-cardinality categorical features.

Deep learning plays an important role in understanding and learning the complex relationships between transaction attributes. A use case of Autoencoders (AEs) is fea-

ture extraction for classification. Misra et al. (2020) conducted an experiment on the European dataset and used an AE to extract the relevant attributes from the input data, which were then fed into a classifier to determine whether a transaction was fraudulent or not. The results showed that an AE followed by MLP performed best in terms of F1-Score (0.8265) (Misra et al.; 2020). Artificial Neural Networks (ANNs) are more powerful than traditional algorithms, but they are slow to train. Jain et al. (2019) evaluated the performance of various supervised models using the KDD'99 intrusion dataset, and the results showed that ANN outperformed the other models; however, it was argued that ANNs are computationally expensive to train (Jain et al.; 2019). On the other hand, it has been demonstrated that deep learning not always outperforms traditional algorithms. Raza and Qayyum (2019) proposed a Variational AE (VAE) to detect fraudulent transactions and benchmarked it against different supervised classifiers using the European dataset. While their proposed VAE achieved the highest recall rate (81.5%), AdaBoost achieved a better overall performance in terms of F1-Score (0.854 vs 0.776).

The reviewed literature suggests that ensemble learning methods have proven to improve the performance of credit card fraud detection models. From an evaluation perspective, it has been observed that F1-Score is a popular performance metric in credit card fraud detection; however, Chicco and Jurman (2020) analysed a highly imbalanced genomic dataset and demonstrated that when evaluating binary classifiers on highly skewed datasets, Matthews Correlation Coefficient (MCC) produces a more accurate and informative result than F1-Score since MCC takes into consideration all the outcomes of the confusion matrix and also the total number of positive and negative observations.

2.3 Summary and Research Gaps

A SMOTE-based hybrid sampling approach has proven to be effective in dealing with the class imbalance problem in fraud datasets when used in conjunction with ensemble classifiers (Sisodia et al.; 2017; Shamsudin et al.; 2020). Ensemble classifiers have shown promising results in detecting fraudulent transactions. Dhankhad et al. (2018) concluded that ensemble learning improves the performance of fraud detection models; Husejinovic (2020) and Zareapoor and Shamsolmoali (2015) concluded that Bagging with DT as base learner achieves promising results in fraud detection; Divakar and Chitharanjan (2019) and Fang et al. (2019) demonstrated that XGBoost and LightGBM, respectively, outperformed other ensemble classifiers in detecting fraud. Since tree-based ensemble algorithms have demonstrated to outperform other state-of-the-art algorithms in detecting credit card fraud, they have been used in this research. The research gap found is that as almost all the reviewed studies are based on the European credit card dataset, in which all the predictors are continuous, it is unknown the extent to which their findings can be generalised to datasets that contain a mix of continuous and categorical predictors.

3 Research Methodology

Considering that this research intends to solve a business problem, a scientific methodology partially inspired on the CRoss-Industry Standard Process for Data Mining (CRISP-DM) approach has been proposed. A graphical representation of the proposed methodology in order to answer the research question outlined in Section 1.2 is shown in Figure 2 followed by a detailed explanation of the different phases.

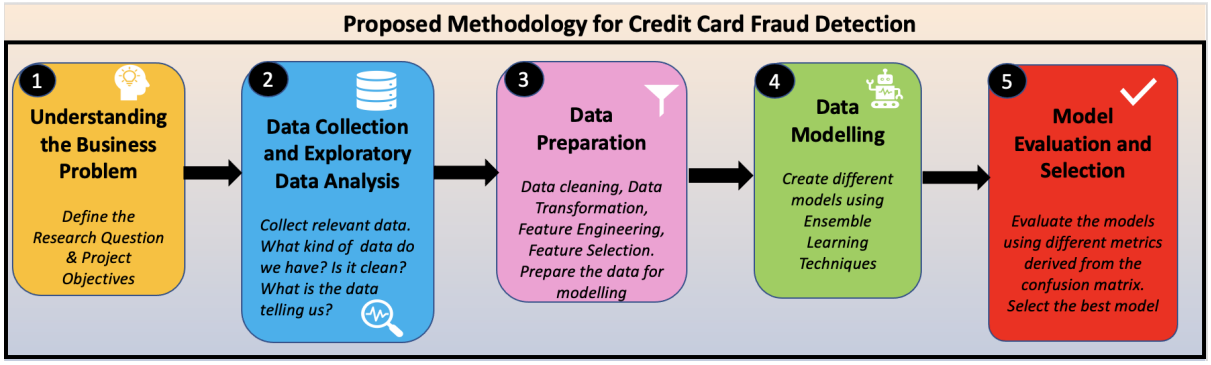

Figure 2: Proposed Research Methodology for Credit Card Fraud Detection

3.1 Understanding the Business Problem

In this phase, an understanding of the credit card fraud detection problem has been developed, the research question and the project objectives are defined, and the problem is formulated as a data mining classification problem. The goal is to create an ensemble classifier that based on past transactions can accurately detect fraudulent transactions.

3.2 Data Collection and Exploratory Data Analysis (EDA)

A simulated credit card transaction dataset that contains legitimate and fraudulent transactions made by American cardholders during the period from January 1st, 2019 to December 31st, 2020 has been used to conduct this research. The dataset[3] consists of 1,852,394 credit card transactions made by 1,000 customers and has 23 transaction attributes; it has been obtained from the Kaggle repository and is used to predict whether a transaction is legitimate or fraudulent. The outcomes of this binary classification problem are coded as **'0'** and **'1'**, which represent ***'legitimate'*** and ***'fraudulent'***, respectively. The original dataset contains two files which were concatenated in order to gain a better insight from the data. The features of the dataset are described in Table 2.

EDA was conducted to get an general understanding of the data and its underlying structure. The volume and value of legitimate and fraudulent transactions are shown in Figure 3. Since the majority of transactions are legitimate and only very few are fraudulent, the dataset is highly imbalanced. Although the fraudulent transactions only account for 0.52% of the total transaction volume, its monetary value is marginally higher at 3.95% of the total amount value. Furthermore, it was observed that the average amount for a fraudulent transaction is $530.66 in comparison with the average amount for a legitimate transaction which is $67.65. This demonstrates that despite fraudulent transactions being infrequent, failing to detect them can result in major financial losses. There are outliers in the variable 'amount' for legitimate transactions, but there are none for fraudulent transactions. There are no missing values and no duplicates in the dataset.

3.3 Data Preparation

In supervised learning, algorithms learn from the input data. Hence, it is essential to prepare the data before modelling as the quality of the data and the number of features in

[3]**Dataset Source:** `https://www.kaggle.com/kartik2112/fraud-detection`

the dataset have an impact on the performance of the models. The following tasks were performed in this phase: 3.3.1 data cleaning, 3.3.2 data transformation, 3.3.3 feature engineering, 3.3.4 feature selection, 3.3.5 train-test split, 3.3.6 categorical encoding and 3.3.7 resample the training data to deal with the class imbalance problem.

Table 2: Features of the Credit Card Transactions Dataset used in the Research

Feature/s	Type	Description
`is_fraud`	binary	Whether the transaction is fraud or not
amt	continuous	Amount of the transaction
city-pop	continuous	Population of the city the customer lives
unix-time	continuous	Time of the transaction in unix time
trans-day-trans-time	interval-scale	Date and Time of the transaction (trxn)
dob	interval-scale	Date of birth of the customer
first / last	nominal	First and Last name of the customer
gender	binary	Gender of the customer
merchant	nominal	Merchant the customer is paying to
merch-lat / merch-long	continuous	Merchant's Latitude and Longitude
street / city / state	nominal	Street, City, and State where customer lives
zip	nominal	ZIP code on credit card
lat / long	continuous	Latitude and Longitude of the customer
cc-num	nominal	Credit card number of the customer
trans-num	nominal	Unique trxn num. for each and every trxn
category	nominal	Shopping category
job	nominal	Job of the customer

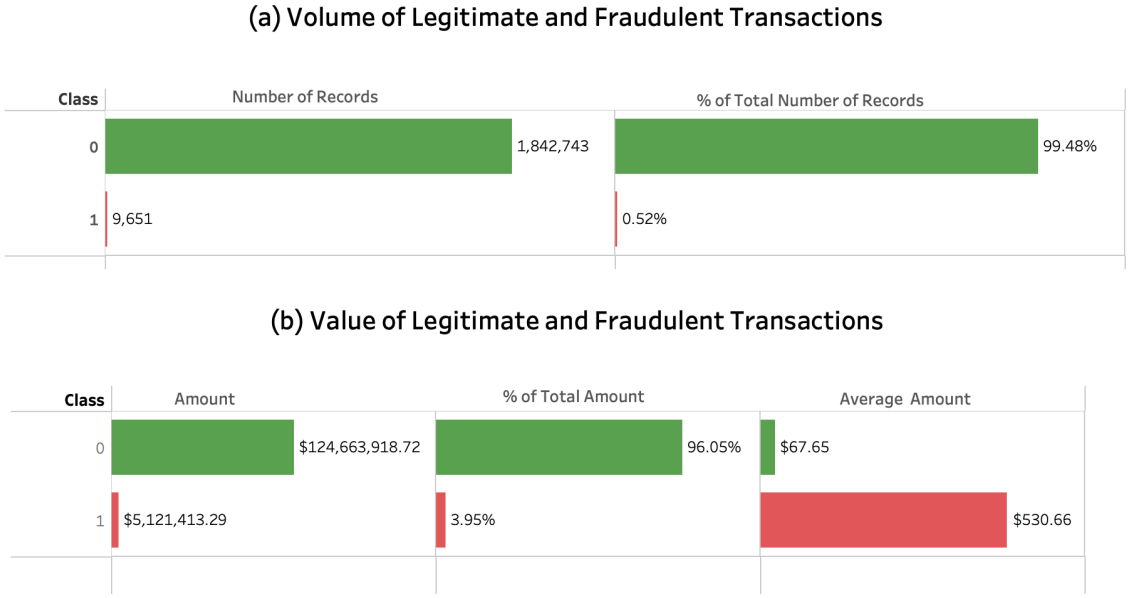

Figure 3: **(a)** Volume of Legitimate ('0') and Fraudulent ('1') credit card transactions. **(b)** Value of Legitimate ('0') and Fraudulent ('1') credit card transactions.

3.3.1 Data Cleaning

Irrelevant features such as unique identifiers, features that have almost unique values and redundant features were removed from the dataset in order to prevent overfitting of the models. The features *'unix-time'*, *'merch-lat'* and *'merch-long'* were dropped at the beginning of the analysis, whereas the features *'trans-num'* (transaction ID) and *'trans-date-trans-time'* were removed after being used for feature engineering. Outliers were not removed because the algorithms used in this research are unaffected by them.

3.3.2 Data Transformation

The purpose of data transformation is to make the data useful for modelling. In this research, data transformation was performed at two different stages of the analysis. In the first stage, the data type of the dichotomous target variable `'is_fraud'` was changed to 'category', and the features *'zip'* and *'cc-num'* were converted from integers to strings, as they are nominal variables whose values are represented by numbers. The features representing time were converted to datetime. The feature *'gender'* was converted to numerical using binary encoding. The rest of the categorical features, which most of them present a very high cardinality (high number of unique values) have not yet been encoded, as first it needs to be determined whether they have predictive significance in their raw format. Only the categorical features selected for modelling will be encoded after conducting feature selection.

3.3.3 Feature Engineering

Since the features *'trans-date-trans-time'* and *'dob'* do not bring much information in their raw format, feature engineering was performed in order to derive more informative features. As a result, the features described in Table 3 were generated from the variable *'trans-date-trans-time'*, and the variable *'dob'* was also used to derive the feature *'age'*. After performing feature engineering, the variables *'trans-date-trans-time'* and *'dob'* were dropped from the dataset as they became redundant. The variable *'trans-num'* was also dropped as it was no longer needed. On the other hand, the feature *'region'* was derived from the variable *'state'* in order to have a feature that describes location and has a low cardinality.

3.3.4 Feature Selection

Feature selection was conducted in order to improve the performance and reduce overfitting and the training time of the models. Since the dataset used in this research contains a mix of continuous and categorical features, different statistical tests were performed to determine the most informative predictors. A correlation matrix was computed to examine the relationship between the continuous predictors and the target variable, and it was observed that the predictor *'amount'* followed by *'hourEncoded'* has the strongest relationship with the class of transaction. The univariate feature selection method SelectKBest from `sklearn.feature_selection`, which uses ANOVA F-values to compute the feature importance scores, was used in this research to select the most informative continuous predictors. The 8 most informative features to predict the class of transaction were selected out of the 15 continuous features. This filter method was chosen as opposed to wrapper methods, which are computationally expensive on large datasets.

Table 3: Features Generated

Feature Generated	Type	Description
age ((trans-day-trans-time)-dob)	numerical	Age at the time of the transaction
transaction-hour	numerical	Hour of the transaction
hourEncoded	binary	Whether trans occurs during day or night
day-of-week	nominal	Day of the week of the transaction
month-of-trans	nominal	Month of the transaction (trxn)
time-since-last-transaction	numerical	Time since last trxn (in seconds)
last-1-day-trans-count	numerical	Volume of trxn made the previous day
last-7-days-trans-count	numerical	Volume of trxn made in the past 7 days
last-14-days-trans-count	numerical	Volume of trxn made in the past 14 days
last-30-days-trans-count	numerical	Volume of trxn made in the past 30 days
last-60-days-trans-count	numerical	Volume of trxn made in the past 60 days
region (derived from 'state')	nominal	Region of the cardholder

A different approach was taken to select the most informative categorical predictors. Since most of the categorical variables have a very high cardinality, this makes more difficult the process of building the models. Therefore, it is important to determine whether these variables have predictive significance in their raw format before including them in the model. A Chi-Square test for independence was conducted in order to determine whether the categorical predictors are associated with the dichotomous target variable, and Cramer's V test was used to measure the strength of the association. Chi-Square statistic tests whether there is a statistically significant relationship between two categorical variables, and the reason why it was chosen in this research is because Chi-Square makes no assumptions regarding the underlying distribution of the data and unlike other statistics it does not assume that the data follows a normal distribution (McHugh; 2013). It was observed that the high-cardinality categorical predictors have a weak association with the target variable (class of transaction). Since Chi-Square is an omnibus test, when testing the association between a predictor that has multiple levels (categories) and the target variable, we can not tell which categories are responsible for the relationship between the predictor and the target variable. For this reason, a Post-Hoc test using Bonferroni correction was performed to examine the association between each level of the categorical predictors and the class of transaction. Due to the high cardinality of most of the categorical predictors, only the predictors whose levels are all associated with the class of transaction were selected for modelling.

After conducting the relevant statistical tests, the selected features for modelling are as follows: **'amt'**, **'age'**, **'hourEncoded'**, **'time-since-last-transaction'**, **'last-7-days-trans-count'**, **'last-14-days-trans-count'**, **'last-30-days-trans-count'**, **'last-60-days-trans-count'**, **'category'** and **'day-of-week'**.

3.3.5 Train-Test Split

In order to train and test the classification models, the dataset was split into training set (70%) and test set (30%). Because the dataset is extremely imbalanced, stratified splitting was applied in order to preserve the class proportions observed in the original dataset. The models were built on the training set and evaluated on the test set.

3.3.6 Categorical Encoding

Since most machine learning algorithms do not accept categorical input variables, they need to be encoded before building the models. Despite one-hot encoding being a very popular encoding method, it was not used in this research because tree-based ensemble algorithms, which were used to conduct the experiments, are sensitive to one-hot encoding. The categorical predictors selected for modelling were encoded using CatBoost encoder. CatBoost encoder is a supervised target-based encoder that introduces an ordering principle to reflect the ordering of the categories with respect to the target variable; and the target statistic for each observation is calculated by only using the target from the observed history. Since CatBoost encoding overcomes the target leakage problem, it was chosen for this research. In order to avoid data leakage, CatBoost encoding needs to be performed separately on the training and test set.

3.3.7 Class Imbalance

The dataset used in this research is highly imbalanced as only 0.52% of the transactions are fraudulent; therefore, the classifiers are very likely to predict the majority class which would result in a high classification accuracy, but this would only be a reflection of the underlying class distribution. To reduce the class imbalance, a hybrid of random undersampling (RUS) and Borderline-SMOTE oversampling was applied to the training set. The test set was not resampled since it is only used to test and evaluate the models, and it would not reflect 'real' data if the class distribution on the test set was modified. Shamsudin et al. (2020) argued that a hybrid sampling approach helps to overcome the loss of information caused by RUS and that SMOTE alone is not enough to sample the data in large datasets as it can lead to overfitting and overgeneralisation of the model. For this reason, a hybrid of RUS and Borderline-SMOTE was the chosen sampling method. Also, Singh et al. (2021) and Sisodia et al. (2017) argued that in extremely imbalanced fraud datasets a hybrid sampling approach is effective when used with ensemble classifiers.

The SMOTE sampling approach combines subsetting with replication of the minority class. A subset of data is randomly selected from the minority class and then new synthetic samples (observations) for the minority class are randomly generated based on its k nearest neighbors, which are found by using the Euclidean distance between observations. The new synthetic samples are then added to the original training set. This results in more fraudulent transactions being present on the training data, which facilitates the task of recognising fraudulent patterns from the data. However, since observations near the borderline are more likely to be misclassified than those further away, in order to facilitate the classifier to learn the borderline of each class in the training process, Borderline-SMOTE was used. Unlike SMOTE, Borderline-SMOTE only uses borderline samples to generate new synthetic samples; in other words, only creates synthetic data along the decision boundary between the two classes. While Borderline-SMOTE generates new synthetic instances for the minority class (fraudulent transactions), RUS removes instances from the majority class (legitimate transactions).

The large size of the training set (1,296,675 observations) would make training the models computationally expensive. In order to reduce the training time of the models, the training set was first undersampled (RUS) and then oversampled (Borderline-SMOTE). The sampling parameters were manually adjusted in a way that the majority class was reduced to 20 times the size of the minority class in the original traning set, and the minority class was 90% of the resultant size of the majority class. This resulted in

a training set of 256,728 transactions, in which 135,120 are legitimate and 121,608 are fraudulent. Figure 4 shows the effect of the combination of RUS with Borderline-SMOTE in the training set. The imbalanced-learn Python package was used to handle the class imbalance.

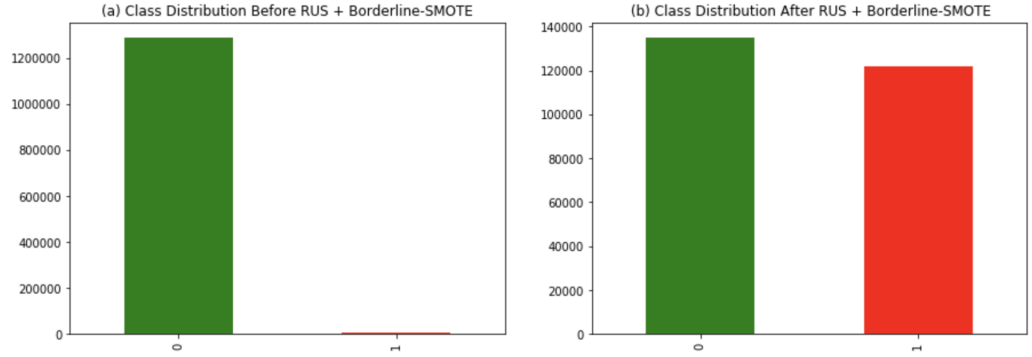

Figure 4: **(a)** Class Distribution Before Sampling the training set. **(b)** Class Distribution After Sampling the training set using a hybrid sampling of RUS + Borderline-SMOTE.

3.4 Data Modelling

Tree-based ensemble learning algorithms have been used in this research. The rationale for using them was established through the literature review, which found that ensemble learning methods based on decision trees (DTs) are state-of-the art techniques to detect credit card fraud. Ensemble methods combine multiple machine learning models (referred to as 'weak' learners) to derive an output and improve the accuracy of the predictions. Ensemble learning improves the stability and the predictive performance of the model. By combining predictions from multiple base models, ensemble learning reduces bias and variance in the final ensemble model (referred to as 'strong' learner).

DTs, which predict the class of the target variable by learning simple decision rules inferred from the input (training) data, were used as base learners to build the ensemble models. Bagging and boosting are two of the most popular ensemble learning techniques. While bagging reduces variance; boosting, on the other hand, reduces bias in the model's predictions. Figure 5 illustrates the differences between the structure of the bagging and boosting methodologies. The following ensemble models have been used in this research: 3.4.1 Bagging, 3.4.2 Random Forest , 3.4.3 XGBoost, 3.4.4 LightGBM and 3.4.5 CatBoost.

3.4.1 Bagging (Bootstrap Aggregation)

In Bagging, bootstrap samples (random samples drawn with replacement) are generated from the training data. Multiple weak learners, such as DTs, are then trained in parallel on each sample. Finally, the predictions of all the individual models are aggregated using majority voting to get the final prediction. By combining the predictions from multiple models, bagging reduces variance in the model, which minimises the risk of overfitting. Bagging was chosen for this research because of its capability to avoid overfitting and because it produced promising results in the studies conducted by Husejinovic (2020) and Zareapoor and Shamsolmoali (2015).

3.4.2 Random Forest

Random Forest (RF) is an extension of Bagging that uses bootstrap samples from the training data to build multiple DTs. However, unlike Bagging, RF only uses a random subset of input features to split the data at each node in the tree. The predictions of each DT are combined and the most common output class (majority voting) is selected. RF was chosen for this research because it achieved promising results in the studies conducted by Mishra and Ghorpade (2018), Varmedja et al. (2019), Taneja et al. (2019), Muaz et al. (2020), Shamsudin et al. (2020), Sahu et al. (2020) and Dhankhad et al. (2018).

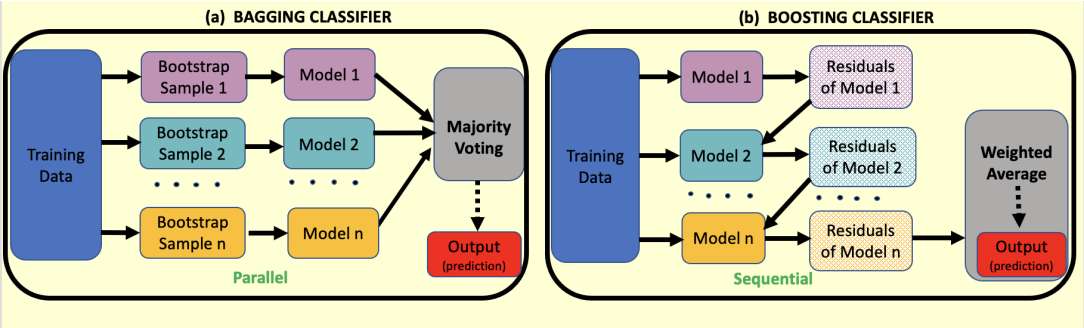

Figure 5: **(a)** Structure of the Bagging Classifier. **(b)** Structure of the Boosting Classifier.

3.4.3 XGBoost (Extreme Gradient Boosting)

Boosting is the process of combining a set of weak learners into a single strong learner in order to improve the predictive performance of the model. Unlike bagging, in boosting the weak learners are trained sequentially on different weighted versions of the training data, with each new model being trained on the residuals of the previous model in order to improve its predecessor. As a result, the final model has a lower bias. Gradient Boosting uses gradient descent algorithm in order to minimise the prediction errors in the sequential models. XGBoost is a gradient boosting tree algorithm which is optimised through parallel processing, tree pruning using depth-first approach, in-built cross-validation capability and regularisation to prevent overfitting[4]. XGBoost was chosen for this research because of the promising results achieved by Divakar and Chitharanjan (2019), Dhankhad et al. (2018), Bayram et al. (2020) and Niu et al. (2019) when using it to predict fraud.

3.4.4 LightGBM (Light Gradient Boosting Machine)

LightGBM is a gradient boosting tree algorithm that unlike other tree-based algorithms operates by growing trees vertically leaf-wise (best-first) and selects the leaf that minimises the loss to grow. By growing the trees leaf-wise instead of level-wise (depth-first), LightGBM improves the accuracy of the model. However, it can lead to overfitting as it creates much more complex trees. In order to prevent overfitting, the hyperparameters of the model should be adjusted. LightGBM allows for faster training time and requires less memory to run [5]. LightGBM was chosen for this research due to its effectiveness in handling large datasets when used by Fang et al. (2019) and Taha and Malebary (2020).

[4]https://medium.com/pursuitnotes/day-50-xgboost-1-4761634243d
[5]https://lightgbm.readthedocs.io/en/latest/

3.4.5 CatBoost (Categorical Boosting)

CatBoost is a relatively new gradient boosting tree algorithm that can automatically handle categorical features. Despite having already encoded the categorical features, CatBoost was chosen for this research as it reduces overfitting and provides a fast prediction [6]. Unlike other boosting algorithms, CatBoost implements symmetric trees which reduce the training time of the model, and it uses ordered boosting, which avoids target leakage and overfitting by training each tree on a subset of data while calculating the residuals on a different subset of new data . In the literature, CatBoost was used by Hancock and Khoshgoftaar (2020) who argued that it outperformed XGBoost when used in datasets with mixed data types.

3.5 Model Evaluation

Credit card fraud detection is an imbalanced binary classification problem in which the output label can either be 'fraudulent' or 'legitimate' transaction. Fraudulent transactions are the *'positive'* class and legitimate transactions are the *'negative'* class. Since the goal of this research is to build a model that can accurately detect fraudulent transactions, the following metrics based on the outcomes of the confusion matrix shown in Table 4 are used to evaluate the performance of the models: *Recall*, *Precision*, *F1-score*, *Mathews Correlation Coefficient*, *Geometric Mean* and *Area Under the Precision-Recall Curve*.

Table 4: Confusion Matrix for Credit Card Fraud Detection

	Predicted Legit. ('0')	**Predicted Fraud ('1')**
Actual Legit. ('0')	**True Negative (TN)**	**False Positive (FP)**
Actual Fraud ('1')	**False Negative (FN)**	**True Positive (TP)**

Recall (or Sensitivity) (1) is the percentage of fraudulent transactions correctly predicted by the model out of all fraudulent transactions; it is also called *fraud detection rate* or *true positive rate* (Baesens et al.; 2015). **Precision** (2) is the percentage of fraudulent transactions correctly predicted by the model among all transactions predicted to be fraudulent (Baesens et al.; 2015). There is a trade-off between *precision* and *recall*. **F1-score** (3) is the harmonic mean of *precision* and *recall*. Since fraud datasets are highly imbalanced, *accuracy* is a misleading evaluation metric; instead, *F1-score* is a better performance metric for imbalanced classification problems. (Baesens et al.; 2015).

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (1) \qquad \text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (2) \qquad \text{F1} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision+Recall}} \quad (3)$$

F1-score is a widely used metric to evaluate credit card fraud detection models; however, when dealing with highly imbalanced data, *F1-score* is not a complete performance measure because it does not take into account the TNs, and this could lead to biased results. **Matthews Correlation Coefficient (MCC)** (4), which considers the balanced ratio of the four outcomes of the confusion matrix to measure the quality of binary classifications, was also considered to evaluate the models as it looks at the performance of the classifier on both the positive and the negative classes (Chicco and Jurman; 2020).

[6] https://catboost.ai

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4}$$

The **Geometric Mean (G-Mean)** (5) combines **sensitivity** (1) and **specificity** (6) (*true negative rate*) to measure the balance between the classification performances on both classes, and is a suitable metric for highly imbalanced datasets (Tharwat; 2020).

$$G - Mean = \sqrt{Sensitivity * Specificity} \tag{5}$$

$$Specificity = \frac{TN}{TN + FP} \tag{6}$$

The **Precision-Recall (PR) Curve** depicts the trade-off between *precision* and *recall* for different thresholds, and is useful to evaluate binary classifiers. On highly imbalanced datasets the *PR Curve* is more informative and accurate than the *ROC Curve* (Saito and Rehmsmeier; 2015). The *AUC-PR (Area Under the Curve)* score measures the discriminatory power of the classifier.

4 Design Specification

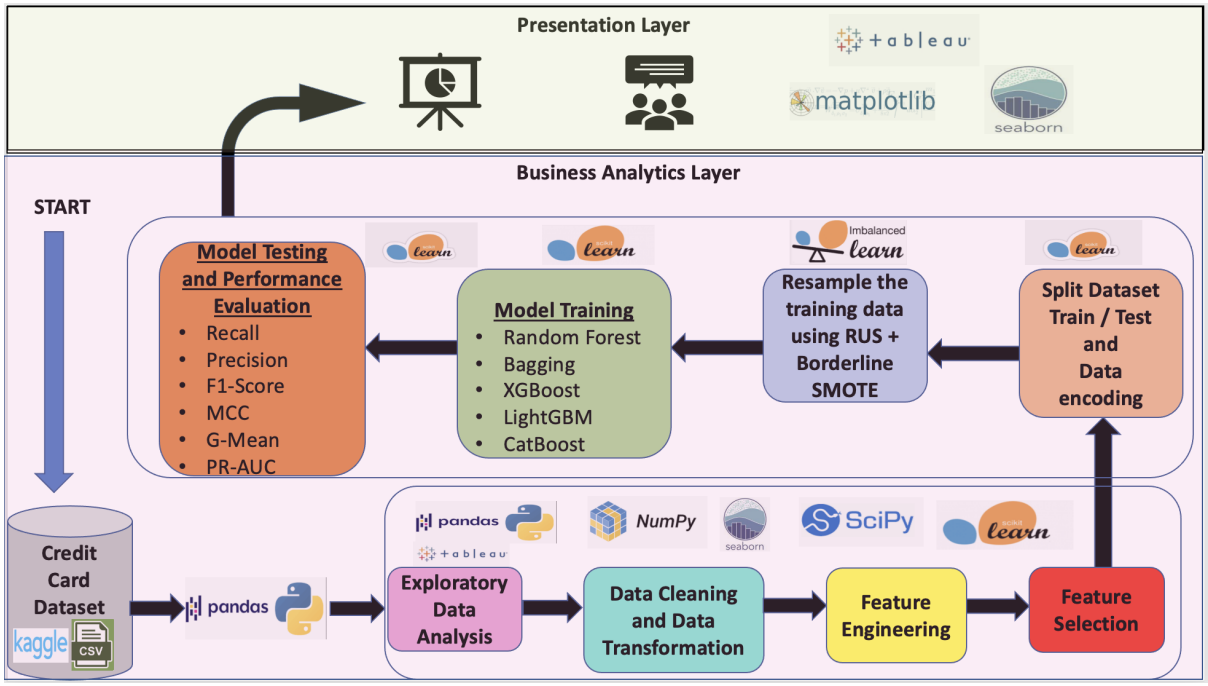

Figure 6: Project Design Specification

All the practical experiments were conducted using Python programming language (v. 3.7.4) and the code was executed on Jupyter Notebook 6.0.1. Tableau Desktop 2021.3 was also used to create visualisations. Python was identified to be the most suitable programming language to conduct this research as it is a very powerful language for data science and machine learning, and also was chosen because of its simplified syntax.

The diagram shown in Figure 6 illustrates the two-tier architecture adopted and outlines the different sequential steps taken when conducting this research. The process

starts in the business analytics layer, with the dataset being downloaded from the Kaggle repository in a csv format and then converted into a pandas dataframe for analysis. The dataset is analysed, pre-processed and modeled using the Python pandas, seaborn, numpy, scipy, sklearn and imbalanced-learn packages. The performance of the classifiers is then evaluated using different metrics derived from the confusion matrix. In the presentation layer, the results of the models and findings of the research are presented using visualisations created with Tableau and Python's matplotlib and seaborn packages.

5 Model Implementation

The implementation of the initial stages of the proposed methodology has already been discussed in Sections 3.2 and 3.3. This section focuses on the implementation of the five different classification models described in Section 3.4. The models were built on the training dataset. As previously mentioned in Section 3.3.7, the class distribution of the training data was balanced using a hybrid sampling approach of RUS + Borderline-SMOTE prior to implementing the models. The size of the training set was also reduced from 1,296,675 to 256,728 observations in order to have a dataset of a manageable size to train the different models.

Bagging and Random Forest classifiers were built using their respective packages from the Python sklearn.ensemble library. On the other hand, in order to build the XGBoost, LightGBM and CatBoost classifiers, their respective Python libraries had to be installed (using Python's pip package) before the relevant packages could be imported. Each classifier was applied to the training set to build the models, and then the models were applied to the test data to make predictions (predict the class of transaction).

In order to improve the predictive performance of the classification models, hyperparameter optimisation was performed using Randomized Search, which consists in finding the best hyperparameter values that maximise the performance of the model by using random combinations of the values defined in the search space. Randomized Search was conducted on each model using the RandomizedSearchCV() function from `sklearn.model_selection`. This technique was chosen as opposed to the grid search technique, which is computationally expensive on large datasets. Due to the large size of the training set, hyperparameter optimisation was computed by using 3-fold cross-validation. Table 5 shows the selected hyperparameter values for the different models.

Table 5: Selected Hyperparameter Values

Model	**Hyperparameter Values**
Random Forest	Default hyperparameter values.
Bagging	`n_estimators`= 50; `max_samples` = 1; `max_features` = 0.5
XGBoost	`n_estimators`= 500; **subsample** = 0.75; `max_depth`= 6; `min_child_weight`= 1; `colsample_bytree`= 0.5; `learning_rate`= 0.3; **gamma** = 0.2
LightGBM	`n_estimators`= 1000; `max_depth` = 6; `max_bin` = 910; `min_data_in_leaf`= 1630; `learning_rate`= 0.2; **subsample** = 0.5; `num_leaves`= 4272; `colsample_by_tree`= 1
CatBoost	**iterations** = 2000; **depth** = 4; `learning_rate`= 0.3; `12_leaf_reg`= 1.0

Only the most important hyperparameters of each model were optimised. For the Random Forest model, it was found that the model with default hyperparameter values performed better than the model with optimised values; hence, it was decided to keep the model with the default values.

6 Results and Evaluation

The goal of this project was to create a model that can accurately detect fraudulent transactions. The different classifiers were evaluated on the test set, which consists of 555,719 observations and is highly imbalanced (only 0.52% of the transactions are fraudulent). The class distribution of the test data was not balanced since real credit card fraud data always has a highly skewed class distribution. The evaluation metrics discussed in Section 3.5, which were computed using the sklearn.metrics module, are used to evaluate the classifiers. The results of the different classification models are shown in Table 6, and the most important credit card fraud predictors for each model are listed in Table 7.

Table 6: Model Results

Model	Recall	Precision	F1-Score	MCC	G-Mean	AUC-PR
Random Forest	0.87	0.35	0.50	0.55	0.93	0.61
Bagging	0.59	**0.60**	0.59	0.59	0.77	0.60
XGBoost	0.87	0.58	**0.70**	**0.71**	0.93	**0.73**
LightGBM	**0.89**	0.57	**0.70**	**0.71**	**0.94**	**0.73**
CatBoost	0.88	0.53	0.66	0.68	**0.94**	0.71

Table 7: Best Predictive Features

Model	Best Predictors of Credit Card Fraud
Random Forest	Transaction Amount.
XGBoost	Hour Encoded and Transaction Amount.
LightGBM	Last 7 Days Transaction Count, Time Since Last Transaction, Transaction Amount, and Category.
CatBoost	Last 7 Days Transaction Count.

6.1 Experiment 1: Random Forest (RF)

The RF classifier has a high fraud detection rate (87%); however, it has a low precision. Only 35% of the transactions predicted to be fraudulent are correctly classified. RF has the lowest precision (35%) among all the classifiers. The low precision results in a worse overall performance in terms of F1-Score (50%) and MCC (55%) when compared to the other classifiers. Like the other models, RF has a high G-Mean (93%), which indicates a high balanced accuracy in the classification of the minority (fraudulent) and

majority (legitimate) classes. The AUC-PR score, which summarises the performance of the classifier across different thresholds, is 0.61, indicating that there is a 61% chance that the RF will be able to discriminate between fraudulent and legitimate transactions. According to the RF model, the transaction amount is the most important predictor of credit card fraud. The confusion matrix, AUC-PR and the feature importance ranking for the RF classifier can be seen in Figure 7.

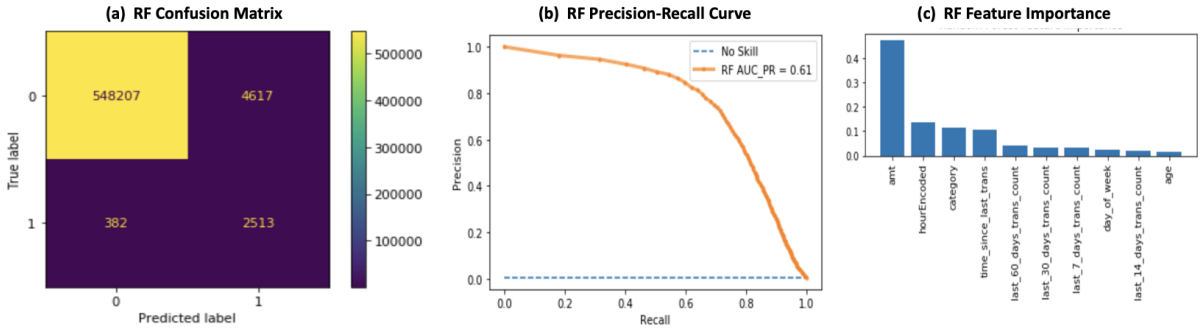

Figure 7: **(a)** Confusion Matrix for RF classifier. **(b)** Precision-Recall AUC for RF classifier. **(c)** Feature Importance for RF classifier.

6.2 Experiment 2: Bagging

The Bagging classifier has a fraud detection rate of 59%, which is by far the lowest among all the classifiers. On the other hand, it has a precision of 60%, being the highest of all the classifiers. The model detects 59% of the fraudulent transactions, and 60% of the transactions predicted to be fraudulent are actually fraudulent. This results in an overall performance of 59% in terms of F1-Score and MCC, which is lower compared to the boosting classifiers. In comparison with the other models, Bagging has a lower G-Mean (77%), which indicates an inferior balance between sensitivity *(true positive rate (TPR))* and specificity *(true negative rate (TNR))*. The AUC-PR score is 0.60, indicating that the model performs better than a random classifier (which has a AUC-PR of 0.5). However, boosting classifiers achieved a higher AUC-PR score than bagging classifiers. The confusion matrix and the AUC-PR for the Bagging classifier can be seen in Figure 8.

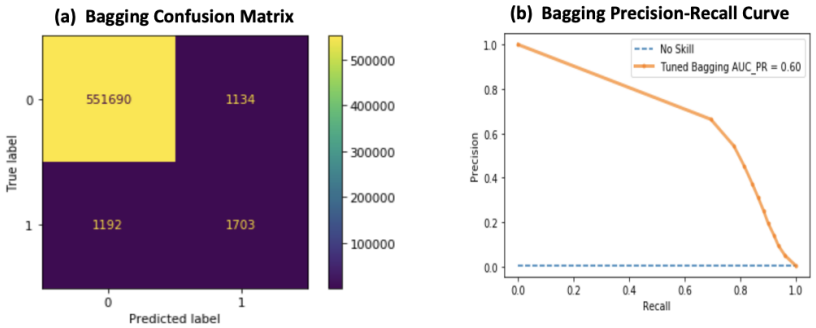

Figure 8: **(a)** Confusion Matrix for Bagging classifier. **(b)** Precision-Recall AUC for Bagging classifier.

6.3 Experiment 3: XGBoost

The XGBoost classifier has a high fraud detection rate (87%) and a precision of 58%. The model detects 87% of the total volume of fraudulent transactions, and 58% of the transactions identified as fraudulent are correctly classified. This results in an overall performance of 71% in terms of F1-Score and MCC, which are the highest among the classifiers; along with LightGBM. In comparison to the other models, XGBoost has a very high G-Mean (93%), which indicates that the model has a high accuracy in the classification of both the minority (positive) and majority (negative) classes; in other words, the model has a high sensitivity *(TPR)* and specificity *(TNR)*. XGBoost, along with LightGBM, has also the highest AUC-PR, with a score of 0.73, indicating that there is a 73% chance that the model will be able to discriminate between legitimate and fraudulent transactions. This suggests that the classifier has a good discriminatory ability. According to the XGBoost model, the time and the amount of the transaction are the most important predictors of credit card fraud. The confusion matrix, AUC-PR and the feature importance ranking for the XGBoost classifier can be seen in Figure 9.

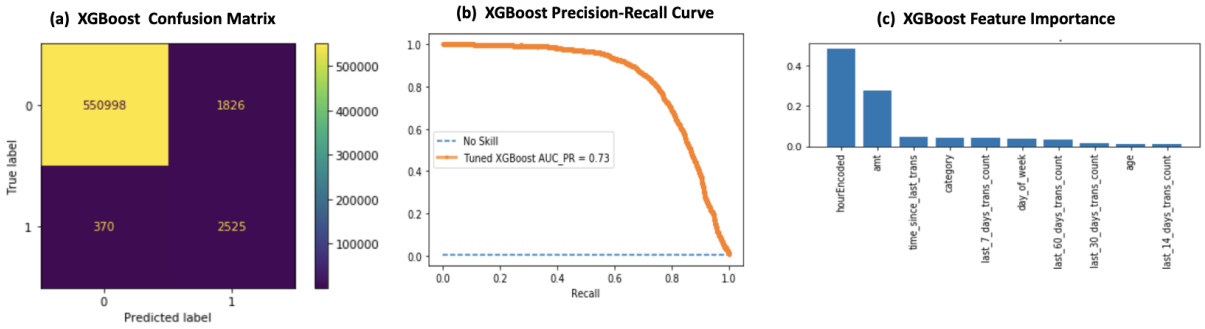

Figure 9: **(a)** Confusion Matrix for XGBoost classifier. **(b)** Precision-Recall AUC for XGBoost classifier. **(c)** Feature Importance for XGBoost classifier.

6.4 Experiment 4: LightGBM

The LightGBM classifier has the highest fraud detection rate (89%) with a precision of 57%. The model can detect 89% of the fraudulent transactions, and 57% of the transactions identified as fraudulent are actually fraudulent. This results in an overall performance of 71% in terms of F1-Score and MCC, which are the highest among the classifiers; along with XGBoost. In comparison with the other models, LightGBM achieves a very high G-Mean (94%), indicating that the model has a high sensitivity *(TPR)* and specificity *(TNR)*; in other words, the model has a high classification performance on both the fraudulent and legitimate transactions. LightGBM also has the highest AUC-PR score (0.73), along with XGBoost, which indicates that the classifier has a good discriminatory ability; there is a 73% chance that the model will be able to distinguish between legitimate and fraudulent transactions. According to the LightGBM model, the volume of transactions over the past 7 days, the time since last transaction, the amount of the transaction and the product category are the most important predictors of credit card fraud. The confusion matrix, AUC-PR and the feature importance ranking for the LightGBM classifier can be seen in Figure 10.

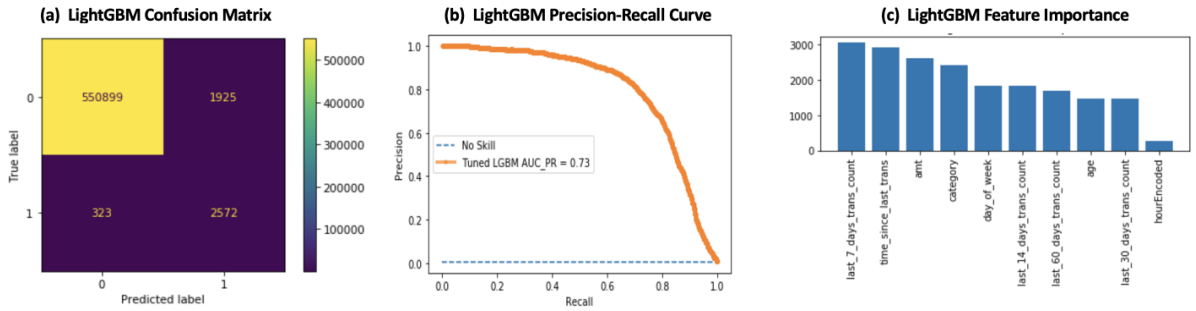

Figure 10: **(a)** Confusion Matrix for LightGBM classifier. **(b)** Precision-Recall AUC for LightGBM classifier. **(c)** Feature Importance for LightGBM classifier.

6.5 Experiment 5: CatBoost

The CatBoost classifier has a high fraud detection rate (88%) and a precision of 53%. The model detects 88% of the fraudulent transactions, and 53% of the transactions identified as fraudulent are correctly classified. This results in an overall performance of 68%, which is only slightly surpassed by XGBoost and LightGBM. Like the other models, CatBoost has a very high G-Mean (94%), which indicates a high sensitivity *(TPR)* and specificity *(TNR)*; in other words, the model has a high accuracy in the classification of both the minority (fraudulent) and majority (legitimate) classes. In comparison with XGBoost and LightGBM, CatBoost has a good discriminatory ability, with a AUC-PR score of 0.71, indicating that there is a 71% chance that the model will be able to distinguish between the two classes. According to the CatBoost model, the volume of transactions over the past 7 days is the most important predictor of credit card fraud. The confusion matrix, AUC-PR and the feature importance ranking for the CatBoost classifier can be seen in Figure 11.

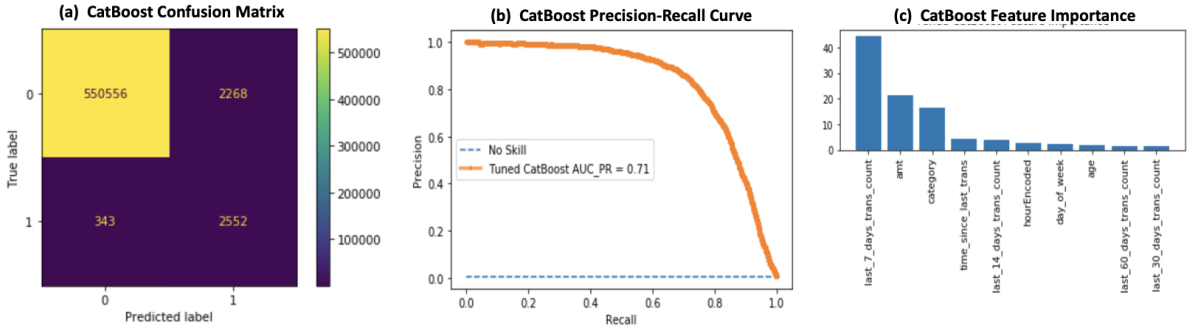

Figure 11: **(a)** Confusion Matrix for CatBoost classifier. **(b)** Precision-Recall AUC for CatBoost classifier. **(c)** Feature Importance for CatBoost classifier.

6.6 Discussion

The aim of this research was to create a model that can accurately predict credit card fraud by applying tree-based ensemble learning algorithms to credit card data. Bagging and boosting techniques are examined in this study. Due to the lack of credit card fraud

datasets, most of the reviewed literature focuses on the European dataset, in which all variables are continuous and most have been anonymised. In order to get a background information about the data, a different dataset which contains mixed data types has been used in this study. To the best of my knowledge, this dataset has yet to be used in any research.

From the results displayed in Table 6, it is observed that boosting classifiers outperformed bagging classifiers in detecting fraudulent transactions. All the models achieved a high ($\geq$ 87%) fraud detection rate (also known as recall, sensitivity or *TPR*) except the bagging classifier (59%). Although bagging had a better precision (60%) than the other models, boosting classifers were only surpassed by bagging by 2% to 7% in terms of precision. On the other hand, RF had the lowest precision (35%), indicating that only 35% of the transactions predicted to be fraudulent were actually fraudulent. While *precision* is an indicator of quality in predicting the positive (fraudulent) class, *recall* is an indicator of quantity (how many positive instances are detected). The precision-recall trade-off is a challenging problem in highly imbalanced fraud datasets, since in credit card fraud detection, both false positives *(FP)* and false negatives *(FN)* have a cost associated and should be kept to a minimum.

F1-score, which combines precision and recall into a single metric to measure the model's accuracy, was the main evaluation metric used in the reviewed literature, along with AUC. However, F1-score does not consider the number of true negatives, which is very high in the domain of credit card fraud detection; hence, it could lead to biased results. MCC (Matthews Correlation Coefficient), which considers all the four entries of the confusion matrix to assess the classifier's performance, is a more informative metric in highly imbalanced credit card fraud datasets. In this study, XGBoost and LightGBM outperformed the other classifiers in terms of MCC (71%) and F1-score (70%) followed by CatBoost (MCC= 68%; F1 = 66%). Furthermore, XGBoost and LightGBM also outperformed the other classifiers in terms of AUC-PR score (0.73) followed by CatBoost (0.71). This suggests that boosting classifiers have a better ability to discriminate between legitimate and fraudulent transactions than bagging classifiers.

All the classifiers had a very high specificity, which means that the models had a very high accuracy in classifying the majority (legitimate) class; in other words, they had a low false positive rate *(FPR = 1 - specificity)*. Similarly, all the classifiers except Bagging had a high sensitivity, which means a high fraud detection rate; in other words, they had a low false negative rate *(FNR = 1 - sensitivity)*. This results in a high G-Mean score, which measures the balance between sensitivity and specificity. Bagging had the lowest G-Mean among all classifiers and this is because Bagging had a lower sensitivity compared to the other models. The G-Mean score is a very useful metric for imbalanced data as it evaluates the performance of the model in the classification of both majority and minority classes.

In order to improve the predictive performance of the models, the hyperparameters of the models were optimised. The RF model unlike the other classifiers achieved better results with the default hyperparameters. For the rest of classifiers it was found that increasing the number of estimators up to a limit improved the performance of the models; however, this was at the cost of a longer training time. For the Bagging classifier it was found that reducing the maximum number of features required to train each DT enhanced the performance of the model. For the boosting classifiers it was found that with a learning rate of 0.2-0.3, the models achieved good results with a reasonable training time. Also, increasing the minimum number of data in one leaf in the LightGBM model

reduced overfitting. For both bagging and boosting classifiers it was found that setting the maximum depth of each tree in the ensemble to 4 - 6 improved the performance of the models. Due to the large size of the training set, the model hyperparameters were tuned using only 3-fold cross validation.

Overall, it has been found that boosting models outperform bagging models in detecting credit card fraud. LightGBM and XGBoost achieved the best results in terms of MCC, F1-Score and AUC-PR followed by CatBoost. LightGBM has been selected as the best model as it has a fraud detection rate and a G-Mean slightly higher than XGBoost.

7 Conclusion and Future Work

This research examines the performance of bagging and boosting algorithms in credit card fraud detection. The goal of this research project was to create a tree-based ensemble classifier that can accurately predict whether a transaction is legitimate or fraudulent using historical transactional data. The dataset used in this study is heavily imbalanced and contains a mix of continuous and categorical data. New features were generated and feature selection was also conducted. It was found that the high cardinality demographic attributes have a weak association with the class of transaction. A hybrid sampling approach of RUS + Borderline SMOTE was applied on the training data to deal with the class imbalance before building the models.

The results indicate that boosting classifiers outperformed bagging classifiers in predicting credit card fraud. LightGBM and XGBoost achieved the best performance in terms of MCC (71%), F1-score (70%) and AUC (0.73), followed by CatBoost. All boosting classifiers achieved a very high sensitivity and specificity, which resulted in a very high G-Mean (94%). The last objective of this research was to identify the most important predictors of credit card fraud. It was found that the volume of transactions over the last 7 days, the time since last transaction, the time of the day, the transaction amount and category are the most influencing factors for predicting credit card fraud.

Future work on this study could involve the use of different approaches to handle the class imbalance. It would be interesting to see how a different hybrid sampling method performs on this dataset. On the other hand, instead of sampling the dataset, a class weight approach giving a higher weight to the minority (fraudulent) class could also be applied. It would be interesting to find out if a class weight approach improves the results obtained in this study. Ensemble classifiers with a base learner other than DT should also be investigated. Deep learning algorithms, such as ANN, and the use of autoencoders for feature extraction, could also be examined.

8 Acknowledgments

First and foremost, I would like to thank my supervisor Dr.Majid Latifi for his valuable support, guidance, feedback and motivation throughout this research project. I would also like to extend my thanks to my parents, friends, colleagues and managers for their moral support when I was working on this research project.

References

Baesens, B., Van Vlasselaer, V. and Verbeke, W. (2015). *Fraud analytics using descriptive, predictive, and social network techniques: A guide to data science for fraud detection*, Hoboken, NJ: John Wiley Sons.

Bayram, B., Köroğlu, B. and Gönen, M. (2020). Improving fraud detection and concept drift adaptation in credit card transactions using incremental gradient boosting trees, *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA). Miami, Florida, USA, 14-17 December 2020,* pp. 545–550 , IEEE Xplore. doi: 10.1109/ICMLA51294.2020.00091.

Benchaji, I., Douzi, S. and El Ouahidi, B. (2019). Using genetic algorithm to improve classification of imbalanced datasets for credit card fraud detection, *Journal of Experimental & Theoretical Artificial Intelligence* **66**: doi: 10.1007/978–3–030–11914–0–24.

Chicco, D. and Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation, *BMC Genomics* **21**(6): doi: 10.1186/s12864–019–6413–7.

Dhankhad, S., Mohammed, E. and Far, B. (2018). Supervised machine learning algorithms for credit card fraudulent transaction detection: A comparative study, *2018 IEEE International Conference on Information Reuse and Integration (IRI). Salt Lake City, UT, USA, 6-9 July 2018,* pp. 122–125 , IEEE Xplore. doi: 10.1109/IRI.2018.00025.

Divakar, K. and Chitharanjan, K. (2019). Performance evaluation of credit card fraud transactions using boosting algorithms, *International Journal of Electronics Communication and Computer Engineering* **10**(6): 262–270.

Fang, Y., Zhang, Y. and Huang, C. (2019). Credit card fraud detection based on machine learning, *Computers, Materials & Continua,* **61**(1): 185–195. doi: 10.32604/cmc.2019.06144.

Hancock, J. and Khoshgoftaar, T. M. (2020). Performance of catboost and xgboost in medicare fraud detection, *19th IEEE International Conference on Machine Learning and Applications (ICMLA)* pp. 572–579, doi: 10.1109/ICMLA51294.2020.00095.

Huang, D., Lin, Y., Weng, Z. and Xiong, J. (2021). Decision analysis and prediction based on credit card fraud data, *ESCC '21: The 2nd European Symposium on Computer and Communications* p. 20–26. doi: 10.1145/3478301.3478305.

Husejinovic, A. (2020). Credit card fraud detection using naive Bayesian and C4.5 decision tree classifiers, *Periodicals of Engineering and Natural Sciences,* **8**(1): pp. 1–5. Available at SSRN: `https://ssrn.com/abstract=3521283`.

Jain, Y., Tiwari, N., Dubey, S. and Jain, S. (2019). A comparative analysis of various credit card fraud detection techniques, *International Journal of Recent Technology and Engineering (IJRTE), ISSN: 2277-3878,* **7**(5S2,): pp. 402–407.

McHugh, M. L. (2013). The chi-square test of independence, *Biochemia Medica* **23**(2): 143–149, doi: 10.11613/BM.2013.018.

Mishra, A. and Ghorpade, C. (2018). Credit card fraud detection on the skewed data using various classification and ensemble techniques, *2018 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS). Bhopal, India, 24-25 February 2018,* pp. 1–5, IEEE Xplore. doi: 10.1109/SCEECS.2018.8546939.

Misra, S., Thakur, S., Ghosh, M. and Saha, S. K. (2020). An autoencoder based model for detecting fraudulent credit card transaction, *Procedia Computer Science,* **167**: 254–262, ScienceDirect. doi: 10.1016/j.procs.2020.03.219.

Muaz, A., Jayabalan, M. and Thiruchelvam, V. (2020). A comparison of data sampling techniques for credit card fraud detection, *(IJACSA) International Journal of Advanced Computer Science and Applications* **11**(6): 477–485. doi: 10.14569/IJACSA.2020.0110660.

Nguyen, T. T., Tahir, H., Abdelrazek, M. and Babar, A. (2020). Deep learning methods for credit card fraud detection. Available at: `https://arxiv.org/ftp/arxiv/papers/2012/2012.03754.pdf` [accessed 18 june 2021].

Niu, X., Wang, L. and Yang, X. (2019). A comparison study of credit card fraud detection: Supervised versus unsupervised. Available at: `https://arxiv.org/pdf/1904.10604.pdf` [accessed 18 june 2021].

Raza, M. and Qayyum, U. (2019). Classical and deep learning classifiers for anomaly detection, *2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST). Islamabad, Pakistan, 8-12 January 2019,* pp. 614–618, IEEE Xplore. doi: 10.1109/IBCAST.2019.8667245.

Sahu, A., GM, H. and Gourisaria, M. K. (2020). A dual approach for credit card fraud detection using neural network and data mining techniques, *2020 IEEE 17th India Council International Conference (INDICON)* pp. 1–7, doi: 10.1109/INDICON49873.2020.9342462.

Saito, T. and Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets, *PLOS ONE,* **10**(3): 1–21, Academic Search Complete. doi: 10.1371/journal.pone.0118432.

Shamsudin, H., Yusof, U. K., Jayalakshmi, A. and Akmal Khalid, M. N. (2020). Combining oversampling and undersampling techniques for imbalanced classification: A comparative study using credit card fraudulent transaction dataset, *2020 IEEE 16th International Conference on Control Automation (ICCA). Singapore, 9-11 October 2020,* pp. 803–808, IEEE Xplore. doi: 10.1109/ICCA51439.2020.9264517.

Singh, A., Ranjan, R. K. and Tiwari, A. (2021). Credit card fraud detection under extreme imbalanced data: A comparative study of data-level algorithms, *Journal of Experimental & Theoretical Artificial Intelligence* pp. 1–28, doi: 10.1080/0952813X.2021.1907795.

Sisodia, D. S., Reddy, N. K. and Bhandari, S. (2017). Performance evaluation of class balancing techniques for credit card fraud detection, *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI).Chennai, India, 21-22 September 2017,* pp. 2747–2752, IEEE Xplore. doi: 10.1109/ICPCSI.2017.8392219.

Taha, A. A. and Malebary, S. J. (2020). An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine, *IEEE Access* **8**: 25579–2558, doi: 10.22044/jadm.2016.788.

Taneja, S., Suri, B. and Kothari, C. (2019). Application of balancing techniques with ensemble approach for credit card fraud detection, *2019 International Conference on Computing, Power and Communication Technologies (GUCON). New Delhi, India, 27-28 September 2019,* pp. 753–758, IEEE Xplore.

Tharwat, A. (2020). Classification assessment methods, *Applied Computing and Informatics* **17**(1): pp. 168–192, doi: 10.1016/j.aci.2018.08.003.

Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M. and Anderla, A. (2019). Credit card fraud detection - machine learning methods, *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH).East Sarajevo, Bosnia and Herzegovina, 20-22 March 2019,* pp. 1–5, IEEE Xplore. doi: 10.1109/INFOTEH.2019.8717766.

Zareapoor, M. and Shamsolmoali, P. (2015). Application of credit card fraud detection: Based on bagging ensemble classifier, *Procedia Computer Science)* **48**: 679–685, doi: 10.1016/j.procs.2015.04.201.