Real Time Credit Card Fraud Detection and Reporting System Using Machine Learning

Ahmed Oluwatoyin JOLAOSHO

LCU/PG/002404

Being a M.Sc Thesis Submitted to the Department of Computer Science, Faculty of Natural and Applied Sciences, Lead City University, Ibadan, Oyo State, Nigeria

In Partial Fulfillment of the Requirements for the Award of Master of Science Degree (MSc) in Computer Science

2023

Certification

This is to certify that Ahmed Oluwatoyin JOLAOSHO with matriculation number LCU/PG/002404 carried out this research work titled "Real Time Credit Card Fraud Detection and Reporting System Using Machine Learning" in the Department of Computer Science, Faculty of Natural and Applied Sciences, Lead City University, Ibadan, Oyo State, for the award of Master of Science (MSc) in Computer Science and that this has not been previously submitted.

…… ……

Prof. Akinola Date

Supervisor

…… ……..

Dr.Wilson Sakpere

Head of Department Date

Dedication

This research work is dedicated to God almighty, my Parents, wife and my children.

Acknowledgement

Foremost, I would like to express my gratitude to the leadership of the Lead City University, Ibadan for creating a medium for us to acquire knowledge for self-reliance.

I acknowledge my supervisor Prof.Akinola, for the continuous support for M.Sc study and research, for his motivation, enthusiasm, to guide me through the research. The immense knowledge and plentiful experience have encouraged me in all the time of my academic research and daily life. Besides my supervisor, my sincere thanks go to the Head of Department Dr. Wilson Sakpere, the Postgraduate Coordinator Dr. Azeez Waheed, all other lecturers and staff members in the department of computer science for their guidance encouragement, and insightful comments.

My sincere gratitude to my father, mentor and boss, who guide and give me the opportunity from ND to Msc level (Dcn. Sunday Agusa) and my colleagues at work Omolara, Dorcas, Aremo, Yinka, Lanre, John, paul Ezekiel, and all Obasanjo Farm (Ibadan Hatchery) staff I appreciate you all.

My special appreciation to my bunnies Wuraola, Arinola, Niniola JOLAOSHO and my wife Kafilat Idowu JOLAOSHO and my sisters Kafilat, Folashade, Remilekun, my parents and to Oluwaferanmi Mudashiru for your support. And all that I cannot mention, Thank you all

Table of Contents

List of Figures

List of Tables

List of Acronyms

Adaboost-	Adaptive Bboosting
ATM-	Automated Teller Machine
CATboost-	Category boosting
DT-	Decision Trees
EFB-	Exclusive Feature Bundling
ELM-	Extreme Learning Machines
ERP-	Effective Radiated Power
GNB-	Gaussian Naive Bayes
GOSS-	Gradient-Based Onside Sampling
KNN -	K-Nearest Neighbour
KNN-	K-Nearest Neighbour
LightGBM-	Light Gradient Boosting Machine
LR-	Logistics Regression
ML-	Machine Learning
MLP-	Multilayer Perceptron
NB-	Naive Bayes
NN-	Neural Network
POS-	Point of Sale
RF-	Random Forest
RTA-	Road Traffic Accidents
SHAP-	SHapley Additive exPlanations
SLFM-	Single Hidden Feed-Forward Artificial Neural Network
SVM-	Support Vector Machine
SVM-	Support Vector Machine
SVR-	Support Vector Regression
XGBoost-	Extreme Gradient Boosting

Abstract

In recent years, there has been a significant rise in fraudulent credit card activities, resulting in substantial financial losses for numerous organizations, companies, and government agencies. This study addresses the critical issue of credit card fraud detection in real-time using machine learning algorithms. The primary objective is to develop a robust prototype model capable of promptly identifying fraudulent transactions and notifying users. To achieve this, three algorithms, namely the Random Forest, the Decision Tree classifier and Linear Regression algorithms were used. Furthermore, various sampling techniques were employed to balance the dataset and improve model performance. 12 distinct models were developed, each offering varying levels of accuracy and effectiveness in fraud detection. From the findings, transactions are most commonly made after 12 noon. Also, older individuals above 75 years are more susceptible to fraud, possibly due to their unfamiliarity with evolving transaction methods. Also, transactions in the dataset are predominantly made the Female gender suggesting that transactions involving this gender may be more prone to fraud. Findings also showed that among these models, the Random Forest -SMOTE [Hyperparameter Tuned], emerged as the best classifier with remarkable performance metrics, including a 97% accuracy rate, an F1 score of 95%, and a precision rate of 98%. The study extended its focus to practical implementation by integrating the Random Forest -SMOTE [Hyperparameter Tuned] with Twilio for real-time notification. This integration successfully demonstrated the model's ability to send timely and accurate fraud alerts to users. The analysis and model development for fraud detection has therefore provided valuable insights and a robust solution for real time identifying and responding to fraudulent activities. It is recommended that periodic evaluations of the fraud detection model's performance be performed to ensure its effectiveness in detecting evolving fraud patterns.

Keywords: Accuracy, Credit Card Fraud Detection, Decision Tree Classifier, Fraud Detection Model, Fraudulent Transactions, Real-time Notification, Sampling Techniques

Word Count: 289 Words

Chapter One

Introduction

1.1 Background to the Study

The proliferation of internet banking systems for conducting cash transactions, bill payments, and shopping has significantly improved the convenience of daily life for many individuals. Notwithstanding the significant advantages of online transactions, users are facing a major issue due to financial fraud. Fraud occurs when an unauthorised individual or entity circumvents the bank's security protocols and impersonates a legitimate customer, thereby assuming the customer's identity. Financial fraud is a prevalent issue that is progressively escalating and has significant implications within the financial industry.

The credit card is a widely used financial product that enables users to make purchases when funds are not immediately available[1]. It can be used for a variety of purchases, including gas, groceries, electronics, travel expenses, and shopping bills. Credit cards offer significant value by providing a range of benefits in the form of reward points when used for various transactions. In 2021, the most commonly utilised payment methods for e-commerce transactions on a global scale were digital wallets, credit cards, and debit cards[2]. It was approximated that digital and mobile wallet payments constituted approximately 50% of global online transactions. In the Asia-Pacific Region, online wallets dominated the market with a market share of nearly 70% for e-commerce payments[3].

Credit card fraud is defined as the unauthorised utilisation of a credit card or its associated information without the owner's consent. In recent times, credit cards have become a popular means of conducting online transactions.

As a result, the frequency of transactions has increased significantly, leading to a corresponding rise in fraudulent transactions. Credit card fraud occurs when the physical credit card is stolen or when sensitive account information is compromised[4]. Since the inception of digital payments, the payments industry has endeavoured to establish a secure environment for financial transactions. Phishing emails are the prevalent form of fraud in internet or online banking, wherein individuals are deceived into disclosing confidential financial information[5]. Due to the growing number of individuals utilising credit cards as a means of payment in their daily routines, credit card companies are advised to prioritise the security and safety of their customers.

In 2019, the global count of credit card users was 2.8 billion. Furthermore, at least 70% of these users possess a minimum of one credit card[6]. The incidence of credit card fraud in the United States increased by 44.7% from 271,927 reports in 2019 to 393,207 reports in 2020[6]. As per the latest report, the global credit card fraud losses in 2021 were recorded at $32.34 billion, which is approximately 14% higher than the losses incurred in 2020, amounting to $28.43 billion. The total fraud losses in the United States for the year 2021 amounted to $11.91 billion, which represents an 18% rise from the $10.09 billion recorded in 2020[3]. The incidence of card not present fraud has increased by 81% compared to point of sale fraud. Credit card fraud is a prevalent issue in the United States, with banks, businesses, and cardholders reporting incidents on a daily basis. In fact, the country accounts for 38.6% of the world's reported payment card fraud losses[7].

From January to September 2020, Nigerian financial services firms experienced a loss of ₦5.2 billion due to fraudulent activities[8]. The majority of the loss was incurred during the period of July to September 2020, with companies experiencing losses of up to ₦3.36 billion. There was a significant surge in the amount lost to fraudsters during the same period in 2020, with a 510% increase from the ₦550 million lost in 2019. The NIBSS 2021 fraud report indicates that there was an 187% rise in the total number of fraud attempts in Nigeria between 2019 and 2020[9]. In 2020, the primary sources of fraud were identified as the web, mobile, ATM terminals, and POS terminals. These sources accounted for 47%, 36%, 9%, and 7% of fraud cases, respectively[9].

Credit card frauds are classified into different categories in the literature. Examples of fraud types include application frauds and behavioural frauds[10]. Application fraud involves the submission of a credit card application using fraudulent identification. On the other hand, behavioural fraud entails the acquisition of a cardholder's credentials to utilise an existing credit card[10]. Fraudulent transactions have been categorised into six types based on the fraudulent process. These categories include frauds resulting from lost or stolen cards, counterfeit cards, online transactions, bankruptcy, merchant fraud, and frauds from cards that were stolen during the expedition process[11].

Credit card fraud detection systems employ a range of techniques, including machine learning, statistical analysis, and pattern recognition, to scrutinise and detect potentially fraudulent transactions[10,11]. The system is capable of detecting various types of fraudulent activities, including but not limited to stolen credit cards, account takeover, and unauthorised transactions. The detection system operates by examining

different transaction attributes, including the transaction amount, location, and time, and contrasting them with historical transaction data.

Machine Learning is a subfield of Artificial Intelligence that enables systems to learn from experience without human intervention. Its objective is to predict future outcomes with high accuracy using different algorithmic models[4]. Machine Learning is a distinct approach from conventional computation methods, as it does not involve explicit programming of systems to perform calculations or problem-solving tasks. The field of machine learning involves the utilisation of input data to train a model.

During the training process, the model is designed to identify various patterns within the input data. This acquired knowledge is then applied to predict unknown results[17]. The scope of machine learning application is extensive. Machine learning finds application in diverse fields such as spam filtering, weather forecasting, stock market analysis, medical diagnosis, fraud detection, autonomous driving, real estate valuation, facial recognition, and numerous other domains[17]. Machine learning is generally classified into three categories: supervised, unsupervised, and reinforcement learning[18].

Supervised learning is a machine learning approach that involves training a model using both input and output labels. Unsupervised learning refers to a type of machine learning where a model is trained using unlabeled data, meaning that the dataset lacks input labels. The model learns various patterns and structures from this data. Typically, it is incorporated into various applications such as visual recognition, robotics, speech recognition, and other similar technologies[18]. Reinforcement learning is a type of machine learning that involves learning how to achieve a complex objective by incrementally maximising a specific dimension[18].

Financial companies and institutions are susceptible to significant financial losses due to fraudulent activities perpetrated by individuals who continuously devise new methods to circumvent regulations and engage in illegal actions. As such, fraud detection systems are crucial for all credit card-issuing banks to mitigate their losses. Various techniques are employed for identifying fraudulent activities, including Neural Network (NN), Decision Trees, K-Nearest Neighbour algorithms, and Support Vector Machines (SVM)[19]. Machine learning (ML) methods can be utilised in two ways: independently or collectively with ensemble or meta-learning techniques. These methods are employed to create classifiers[19].

The focus of this study is on supervised learning techniques and their application in detecting fraudulent activities related to credit card transactions. The study will analyse the data in two primary methods: categorical and numerical analysis. This thesis seek to determine the most suitable algorithms for detecting fraud patterns by conducting a comprehensive evaluation of machine learning techniques. The evaluation will be based on a performance measure that is effective in detecting and reporting fraudulent credit card transactions.

1.2 Statement of Problem

The issue of credit card fraud is a continuously expanding concern within the current financial market. In recent years, there has been a significant rise in fraudulent activities, resulting in substantial financial losses for numerous organisations, companies, and government agencies. Due to anticipated growth, numerous researchers in this field have concentrated on the early detection of fraudulent behaviours through the utilisation of advanced machine learning techniques.

However, most of the dataset utilised in their study exhibits a high degree of imbalance, wherein the frequency of genuine cases, representing the majority, significantly exceeds that of fraudulent cases. Their models exhibits a tendency to display bias towards the majority samples, resulting in the misrepresentation of a fraudulent transaction as a genuine one. This present research employs a data-level methodology, incorporating various resampling techniques like undersampling, oversampling, and hybrid strategies, in conjunction with an algorithmic approach to address this limitation.

Also, numerous credit card fraud detection systems using machine learning in previous literatures relied on predictions[7,11,13,14]. Previous studies are also very scarce on real time credit card fraud detection and reporting and empirical studies using real-time detection and reporting showed lower accuracy, lower f1 scores and precision. This study therefore tends to develop a real time credit card fraud detection and notification prototype model using machine learning algorithms for the purpose of achieving a better detection and reporting performance using three algorithms (Logistic Regression, the Random Forest model and the Decision Tree Classifier model) to identify instances of credit card fraud based on transaction time and amount. Upon detection, an automated text message will be sent to the card owner to alert them of the fraudulent activity.

1.3 Aim and Objectives of the Study

The aim of this study is to develop a real time credit card fraud detection and notification prototype model using machine learning algorithms. The specific objectives are to:

i. Perform data resampling using resampling techniques such as random undersampling, random oversampling, SMOTE to handle the data imbalance.
ii. Build machine learning models using Logistics Regression, Random Forest and Decision Tree Classifier algorithms
iii. Generate a fraud alert notification and
iv. Evaluate the models' performance using precision, recall,F1-score metric, and accuracy

1.4 Significance of the Study

The implementation of a prototype model utilising machine learning algorithms for real-time credit card fraud detection and notification is essential for the identification and reporting of fraudulent activities on credit cards. This study aims to enhance comprehension of credit card fraud detection and its integration into the classification task to improve fraud detection rates. The outcome of this project will enhance the existing technology of financial and card security, ultimately reducing losses. The implementation of this technology will enhance the ability of banks, card owners, and security agencies to promptly and accurately respond to incidents of theft or fraud by facilitating the identification and location of fraudulent activity.

The study aims to make a scholarly contribution to the existing knowledge and provide effective solutions to the problems associated with credit card fraud, including both card present and card not present scenarios. The proposed methodology has the potential to be applied to all supervised tasks that involve sequential datasets. The study's results will function as a point of reference and direction for computer science students, lecturers, and researchers. Additionally, it will stimulate further research on the topic. Furthermore, the discoveries could lead to the formulation of new theories concerning the utilisation of artificial intelligence in

the realm of fraud detection. Finally, the publication of the study would hold significant value in terms of contributing to the existing body of knowledge.

1.5 Scope of the Study

The purpose of this research is to create a prototype model for detecting and notifying credit card fraud in real time, utilising machine learning algorithms. The selected dataset is an open source simulated credit card transaction dataset. It encompasses credit card transactions of 1000 customers with a pool of 800 merchants. The dataset was generated using Sparkov Data Generation Harris. The project involved five main areas, which are credit card fraud detection for both card present and card not present transactions, feature selection, development of a machine learning model, analysis of the model, and interpretation of the results. The evaluation of the design was based on the metrics of precision, recall, and F1-score. The performance of the models on each class will be visualised using the confusion matrix. The accuracy of the models was used as an evaluation metric. The results will also be presented and analysed descriptively.

1.6Limitation of Research

The research is limited by the availability of real life and historical datacredit fraud data. The data used was an open source dataset for training and evaluating the machine learning algorithms. Also, another limitations is the potential challenges associated with adapting the model to evolving fraud patterns and the need for continuous model updates. Another limitation of the research is the potential for false positives, where legitimate transactions are incorrectly flagged as fraudulent.

1.7 Definition of Operational Terms

Credit Card: Credit card is a widely used financial product that enables users to make purchases when funds are not immediately available. It can be used for a variety of purchases, including gas, groceries, electronics, travel expenses, and shopping bills

Credit Card Fraud: Credit card fraud is defined as the unauthorised utilisation of a credit card or its associated information without the owner's consent.

Credit Card Fraud Detection System: A credit card fraud detection system is a software application that is specifically developed to automatically detect and prevent fraudulent transactions carried out using credit cards. It employ a range of techniques, including machine learning, statistical analysis, and pattern recognition, to scrutinise and detect potentially fraudulent transaction

Financial Fraud: Financial fraud is a prevalent issue that is progressively escalating and has significant implications within the financial industry

Fraud: Fraud occurs when an unauthorised individual or entity circumvents the bank's security protocols and impersonates a legitimate customer, thereby assuming the customer's identity

Machine Learning: Machine Learning is a subfield of Artificial Intelligence that enables systems to learn from experience without human intervention. Its objective is to predict future outcomes with high accuracy using different algorithmic models

Reinforcement Learning: Reinforcement learning is a type of machine learning that involves learning how to achieve a complex objective by incrementally maximising a specific dimension

Supervised Learning: Supervised learning is a machine learning approach that involves training a model using both input and output labels

Unsupervised Learning: Unsupervised learning refers to a type of machine learning where a model is trained using unlabeled data, meaning that the dataset lacks input labels

Endnotes

1. P Tiwari, S Mehta, N Sakhuja, J Kumar, A.K Singh. *Credit card fraud detection using machine learning: a study*. arXiv preprint arXiv:2108.10005. 2021 Aug 23

2. https://www.statista.com/statistics/348004/payment-method-usage-worldwide/

3. https://www.paymentsdive.com/news/card-industry-fraud-fighting-efforts-pay-off-nilson-report-credit-debit/639675/

4. R. Shakya. "*Application of machine learning techniques in credit card fraud detection"*. UNLV Theses, Dissertations, Professional Papers, and Capstones. 3454. http://dx.doi.org/10.34917/14279175. 2018

5. S Şentürk, E Yerli. and I Soğukpınar. *Email phishing detection and prevention by using data mining techniques*. In *2017 International Conference on Computer Science and Engineering (UBMK)* (pp. 707-712). IEEE., 2017, October

6. M AlEmad "*Credit card fraud detection using machine learning*". Thesis. Rochester Institute of Technology, 2022.

7. M Ashraf, M.A Abourezka, F.A Maghraby. *A comparative analysis of credit card fraud detection using machine learning and deep learning techniques*. InDigital Transformation Technology: Proceedings of ITAF 2020 2022 (pp. 267-282). Springer Singapore.

8. D.N Anowu, T Nyor, S.E Agbi, A.I Nelson, A.N Saliu.*financial forensic analysis and fraud deterrence in listed deposit money banks in Nigeria*. **Gusau Journal of Accounting and Finance**. 2021 Oct 1;2(4):18

9. NIBSS Insight, "fraud in nigerian financial services" 2021 https://nibss-plc.com.ng/media/PDFs/post/NIBSS%20Insights%20Fraud.pdf

10. C Soulé-Dupuy, E Gaussier, M Lux, G Gianini, S Calabretto, M Granitzer, P.E Portier. *Credit card fraud detection using machine learning with integration of contextual knowledge* (Doctoral dissertation, INSA Lyon).2019

11. Y Lucas. *Credit card fraud detection using machine learning with integration of contextual knowledge* (Doctoral dissertation, Université de Lyon; Universität Passau (Deutscheland)).2019

12. O.S Yee, S Sagadevan, N.H Malim. *Credit card fraud detection using machine learning as data mining technique*. **Journal of Telecommunication, Electronic and Computer Engineering** (JTEC). 2018 Jan 29;10(1-4):23-7.

13. A Thennakoon, C Bhagyani, S Premadasa, S Mihiranga and N Kuruwitaarachchi.*Real-time credit card fraud detection using machine*

learning. In 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 488-493). IEEE., 2019

14. G.K Singh, A Bhayye, S Dhamnaskar, S Patiland S.V Phulari. *Credit card fraud detection using isolation forest*. **International Journal of Recent Advances in Multidisciplinary Topics**, *2*(6), pp.118-119.2021.

15. K Gupta, K Singh, G.V Singh, M Hassan, U Sharma.*machine learning based credit card fraud detection-a review*. In2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC) 2022 May 9 (pp. 362-368). IEEE.

16. D Varmedja, M Karanovic, S Sladojevic, M Arsenovic, A Anderla. *Credit card fraud detection-machine learning methods*. In2019 18th International Symposium Infoteh-Jahorina (INFOTEH) 2019 Mar 20 (pp. 1-5). IEEE.

17. H Nozari, M.E Sadeghi. *Artificial intelligence and machine learning for real-world problems (a survey)*. **International Journal of Innovation in Engineering**. 2021 Oct 7;1(3):38-47.

18. R.K Dhanaraj, K Rajkumar, U Hariharan. *Enterprise IoT modeling: supervised, unsupervised, and reinforcement learning*. Business Intelligence for Enterprise Internet of Things. 2020:55-79.

19. Y Jain, N Tiwari, S Dubey, S Jain. *A comparative analysis of various credit card fraud detection techniques*. **Int J Recent Technol Eng.** 2019 Jan;7(5S2):402-7.

Chapter Two

Literature Review

2.1 Conceptual Review

2.1.1 Credit Card

E-commerce has come a long way since its inception. It has become an essential means for most organizations, companies, and government agencies to increase their productivity in global trade. One of the main reason for the success of e-commerce is the easy online credit card transaction. As credit card transaction is the most common method of payment in the recent years, the fraud activities have increased rapidly. Enterprises and public institutions are facing a massive problem as huge amount of financial loss are caused by fraud activities. The losses due to the credit card, debit card, and prepaid card fraud reached $16.31B worldwide in 2015[1].

Report shows that the gross fraud loss has reached $22.8B in 2018 which is 4% more than that in 2015 and it is expected to exceed by an even more significant amount in the coming years[1]. In the UK, unauthorised fraud losses across payment cards, remote banking and cheques reached £726.9 million in 2022, a decrease of less than one per cent compared to 2021[2]. Remote purchase fraud, where a criminal uses stolen card details to buy something online, over the phone or through mail order, remains the biggest category of losses at £395.7 million – although this figure was again down on the previous year. Fraud on lost and stolen cards increased by 30 per cent to £100.2 million and card ID theft, where a criminal opens or takes over a card account in someone else's name, almost doubled to £51.7 million[2].

In Nigeria, financial financial services companies lost ₦5.2 billion to fraud between January and September 2020[3].The bulk of this loss occurred between July and September 2020, when companies lost up to ₦3.36 billion. This was a 510% increase from the ₦550 million lost to fraudsters in the same period in 2019[4].According to a 2021 fraud report from the NIBSS, the total fraud attempts in Nigeria increased by 187% between 2019 and 2020. The top sources of fraud in 2020 were the web (47%), mobile (36%), ATM terminals (9%), and POS terminals (7%)[5].

According to report, the gross fraud reached $5.6B in 2012, whereas in 2018, the fraud loss has reached $9.1B, which is approximately two-fifths of the total loss1. In particular, 70% of these frauds are Card-Not-Present (CNP) frauds (i.e.,frauds conducted online or over the telephones), 20% are counterfeits and remaining 10% cases are related to losses due to lost or stolen cards1 . The solutions to the fraud can be categorized into prevention, which involves preventing the fraud in the source itself and detection, which is the action taken after the occurrence of the event.

The instances of fraudulent transactions by other authors have been classified into three distinct categories, namely card-related frauds, merchant-related frauds, and Internet frauds[6]. Two primary types of fraud can be identified in a given set of transactions: Card-not-present (CNP) fraud and Card-present (CP) fraud [7,8,9]. Hence, there are five distinct types of fraudulent scenarios[10]:

i. Lost/Stolen Cards: The occurrence of fraudulent transactions is estimated to be around 1%[11]. Fraudsters often target elderly cardholders and employ shoulder surfing techniques to obtain the card's PIN code. Subsequently, they proceed to steal the card.

ii. Cards not Received: Typically, the percentage of fraudulent transactions resulting from credit card theft during production or postal delivery is less than 1%. To prevent such fraudulent activities, banks may require customers to retrieve their cards from the bank agency or activate the card by calling the bank[10].

iii. ID Theft: The acquisition of a card through the use of fraudulent or stolen identification documents.

iv. Counterfeit Cards: Account for less than 10% of fraudulent transactions[10]. The replication of a card can occur either during a legitimate usage of the card or through a breach in the database, and subsequently be duplicated onto counterfeit plastic by organised crime syndicates operating on a global scale. The perpetrator acquires and duplicates the magnetic stripe data of the cards. Historically, this form of fraudulent activity was widespread. However, it has been partially resolved through the implementation of EMV technology[10].

v. Card not Present: The majority of credit card fraud incidents occur during e-commerce transactions, accounting for potentially more than 90% of all fraudulent transactions[11]. The retrieval of credentials (including card number, expiry date, and CVC) typically occurs through a database hacking event orchestrated by international criminal organisations. These credentials are subsequently sold on the dark web. The majority of merchants, approximately 90%, utilise the 3D SECURE technology to provide cardholders with dual identification protection. However, certain major merchant websites, such as Ebay or Amazon, do not offer this safeguard to their users[10].

2.1.1.1 Fraud Detection Process

Credit card fraud detection systems employ a range of techniques, including machine learning, statistical analysis, and pattern recognition, to scrutinise and detect potentially fraudulent transactions[10,11]. The system is capable of detecting various types of fraudulent activities, including but not limited to stolen credit cards, account takeover, and unauthorised transactions. The detection system operates by examining different transaction attributes, including the transaction amount, location, and time, and contrasting them with historical transaction data.

The transactions are first checked at the terminal point to be valid or not, as shown in figure 2.1. At the terminal point, certain essential conditions such as sufficient balance, valid PIN (Personal Identification Number), etc. are validated and the transactions are filtered accordingly. All the valid transactions are then scored by the predictive model, which then classifies the transactions as genuine or fraudulent. The investigators investigate each fraudulent alert and provide feedback to the predictive model to improve the model's performance[1].

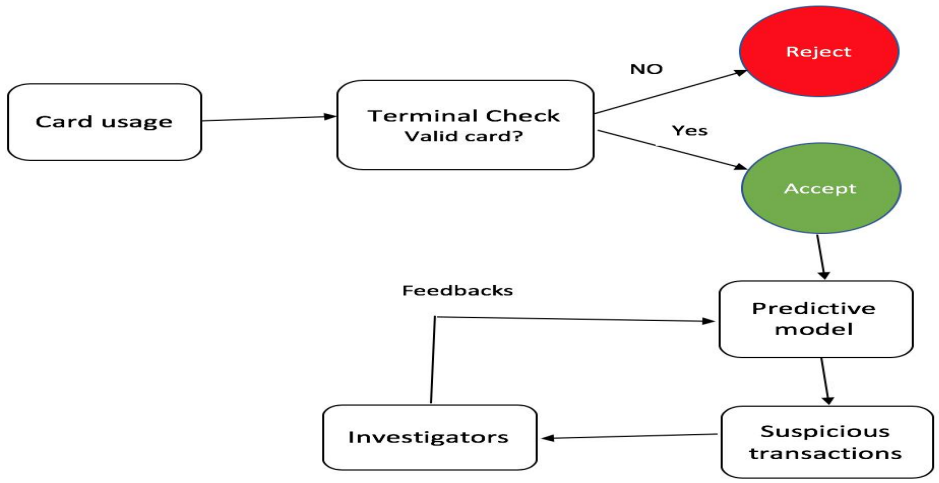


Figure 2.1: Fraud Detection Process[1].

2.1.1.2Challenges In Fraud Detection

In building a fraud detection system, there is need to determine which learning strategy to use (e.g., supervised learning or unsupervised learning), which algorithms to use (e.g., Logistic regression, decision trees, etc.), which features to use, and most importantly, how to deal with the class imbalance problem (fraudulent cases are extremely sparse as compared to the legitimate cases). Class imbalance is not only the major concern in fraud detection system[12]. Overlapping of the genuine and fraudulent classes due to limited information about the transaction records is another problem in the classification task, and most machine learning algorithms underperform under these scenarios.

In a real-life scenario, a fraud detection model predicts the nature of class (genuine or fraudulent) and gives the alert for the most suspicious transaction to the investigators. Investigators then perform a further investigation and provide feedback to the fraud detection system to improve its performance[1]. However, this process can be an overhead for the investigators due to which only a few transactions are validated on time by the investigators. In such a case, just a few feedbacks are provided to the predictive model, which generally results in a lesser accurate model.

Lastly, as financial institutes very rarely disclose the customer data to the public due to confidentiality issues, the real financial datasets are very hard to find. This is one of the major challenges in fraud detection research work.

2.1.2 Machine Learning

In general context, machine learning can be defined as a field in artificial intelligence that provides the system the capability to learn from the experience automatically without the human intervention and aims to predict the future outcomes as accurate as possible utilizing various algorithmic models[13].

Machine Learning is very different than the conventional computation approaches, where systems are explicitly programmed to calculate or solve a problem. Machine learning deals with the input data that are used to train a model where the model learns different patterns in the input data and uses that knowledge to predict unknown results. The application of machine learning is incredibly vast. It is used in various applications like the spam filter, weather prediction, stock market prediction, medical diagnosis, fraud detection, autopilot, house price prediction, face detection, and many more[14,15]. Typically, machine learning has three categories: supervised, unsupervised and reinforcement learning[16].

2.1.2.1 Supervised Learning

Supervised learning can be defined as a machine learning approach in which both input and output labels are provided to the model to train[17]. The supervised model uses the input and output labeled data for training, and it extracts the patterns from the input data. These extracted patterns are used to support future judgments.

Supervised learning can be formally represented as follows:

$$Y = f(x) \qquad 2.1$$

where x represent the input variables, Y denotes an output variable and f(x) is a mapping function.

The goal is to approximate mapping function such that when an unseen input is given to the mapping function, it can predict the output variable (Y) correctly[16]. Furthermore, supervised learning has two sub-categories: classification and regression[18]. In a classification problem, the output variable is a category, (e.g., fraud or genuine, rainy or sunny, etc.). In a regression problem, the output variable is a real value, (e.g., the price of a house, temperature, etc.).

Classification: Classification problem in machine learning can be defined as the task of predicting the class label of a given data point[1,19]. For example, fraud detection can be identified as a classification problem. In this case, the goal is to predict if a given transaction is fraud or genuine.

Generally, there are three types of classification: binary classification, where there are two output labels (e.g., classifying a transaction which may be fraud or genuine), multi-class classification, where there are more than two output labels (e.g., classifying a set of images of flowers) and multi-label classification, where the data samples are not mutually exclusive and each data samples are assigned a set of target labels (e.g., classifying a crab on the basis of the sex and color in which the output labels can be male/female and red/black)[1].

Regression: Regression algorithms are commonly employed to address regression problems that exhibit a linear correlation between input and output variables. Regression models are utilised to forecast continuous output variables, such as market trends, weather patterns, and other related phenomena[20]. Several commonly used regression algorithms include the Simple Linear Regression Algorithm, Multivariate Regression Algorithm, and Lasso Regression.

2.1.2.2 Unsupervised Machine Learning

Unsupervised machine learning involves training a machine using an unlabeled dataset, whereby the machine is capable of predicting output without any form of supervision. The models are trained using unclassified and unlabeled data, and subsequently operate on this data in an unsupervised manner. The primary objective of the unsupervised learning algorithm is to cluster or classify the unstructured dataset based on similarities, patterns, and dissimilarities. The machines are programmed to

identify concealed patterns within the input dataset[20]. Unsupervised Learning can be categorised into two distinct types, which are Clustering and Association

Clustering: The clustering methodology is employed to identify the intrinsic clusters within the dataset. Cluster analysis is a method of categorising objects into groups based on their similarities, with the aim of ensuring that objects within a group share the most similarities while having fewer or no similarities with objects in other groups[18]. One instance of the utilisation of a clustering algorithm is the categorization of customers based on their purchasing patterns. Several widely used clustering algorithms include the K-Means Clustering algorithm, Mean-shift algorithm, DBSCAN Algorithm, Principal Component Analysis, and Independent Component Analysis[21].

Association: Association rule learning is an unsupervised machine learning methodology that aims to discover significant associations between variables in a vast dataset. The primary objective of this particular learning algorithm is to identify the interdependence between two data items and subsequently establish a correlation between these variables, thereby optimising the potential for profit generation. Several widely used Association Rule Learning algorithms include the Apriori Algorithm, Eclat, and FP-growth algorithm. Unsupervised Learning finds various applications such as Network Analysis, Recommendation Systems, Anomaly Detection, and Singular Value Decomposition (SVD)[21].

2.1.2.3 Semi-Supervised Learning

Semi-supervised learning is a machine learning algorithm that occupies an intermediate position between supervised and unsupervised learning algorithms. Semi-supervised learning algorithms occupy a middle ground between supervised

learning, which utilises labelled training data, and unsupervised learning, which operates without labelled training data[21,22].

These algorithms leverage both labelled and unlabeled datasets during the training phase. Semi-supervised learning can be considered as an intermediary approach between supervised and unsupervised learning techniques, wherein the algorithm operates on a dataset that contains a limited number of labelled instances. The concept of Semi-supervised learning has been introduced as a means of addressing the limitations of both supervised and unsupervised learning algorithms. The primary objective of semi-supervised learning is to optimise the utilisation of all available data, as opposed to solely relying on labelled data as in the case of supervised learning. The first step involves clustering comparable data using an unsupervised learning algorithm[20]. This process subsequently facilitates the labelling of previously unlabeled data as labelled data. The reason for this is that obtaining labelled data is a more costly process compared to acquiring unlabeled data.

2.1.2.4 Reinforcement Learning

Reinforcement learning is a feedback-driven process whereby an artificial intelligence agent, which is a software component, autonomously explores its environment through trial and error. It takes actions, learns from its experiences, and enhances its performance. The reinforcement learning agent is incentivized to optimise its performance by receiving positive reinforcement for favourable actions and negative reinforcement for unfavourable actions, with the ultimate objective of maximising its cumulative reward[22,23].

Reinforcement learning is a type of machine learning that differs from supervised learning in that it does not rely on labelled data. Instead, agents acquire knowledge

solely through their experiences[23]. The process of reinforcement learning bears resemblance to that of human learning, whereby a child acquires knowledge through experiential encounters in their daily routine. Reinforcement learning can be exemplified through gameplay, wherein the game serves as the environment, the actions taken by an agent at each step determine the states, and the objective of the agent is to attain a high score.

Reinforcement learning has been utilised in various domains, including but not limited to Game theory, Operation Research, Information theory, and multi-agent systems, owing to its distinctive mode of operation. The formalisation of a reinforcement learning problem can be achieved through the utilisation of a Markov Decision Process (MDP)[22]. Within the framework of Markov Decision Processes (MDP), the agent engages in ongoing interactions with the environment by executing actions. Following each action, the environment responds by generating a new state.

Reinforcement learning is categorized mainly into two types of methods/algorithms. They include Positive Reinforcement Learning which specifies increasing the tendency that the required behaviour would occur again by adding something. It enhances the strength of the behaviour of the agent and positively impacts it, amd Negative Reinforcement Learning that operates in a manner that is diametrically opposed to that of positive reinforcement learning. By avoiding the negative condition, the likelihood of the specific behaviour recurring is heightened[22].

2.1.3 Supervised Learning Classification Algorithms

There are various classification algorithms used in machine learning. Some are discussed below;

2.1.3.1 K-Nearest Neighbor (KNN)

It is considered one of the simplest machine learning methods that are used for both regression and classification. KNN is commonly used to classify data models based on how their neighbors are classified[24]. This supervised machine learning algorithm requires to be used commonly in the following scenario. for instance, when the data is labeled, noise-free, small. Before using the KNN () function, the dataset should be prepared. After predicting the outcome with the kNN algorithm, the model's diagnostic performance should be evaluated. The k value, distance calculation, and selection of appropriate predictors all have a significant impact on model performance[24].

One method is to calculate the distance between the unlabelled observations and the neighbouring dataset. By default, the knn() function uses Euclidean distance, which can be calculated using the following equation[24]:

$$D(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2} \qquad 2.2$$

where p and q are subjects to be compared with n characteristics. Other methods of calculating distance exist, such as the Manhattan distance. Another concept is the k parameter, which determines how many neighbours are chosen by the kNN algorithm[25]. The selection of k has a significant impact on the diagnostic performance of the kNN algorithm. A large k reduces the impact of variance caused by random error, but it runs the risk of ignoring minor but significant patterns[25].

ANN-Benchmarks is a tool for evaluating the performance of approximate nearest neighbour algorithms in memory[26]. It provides a standardized interface for assessing the performance and quality of nearest Neighbour algorithms on various standard data sets. It supports a variety of k-NN algorithm integration methods, and its

configuration system automatically tests a variety of parameter settings for each algorithm.

2.1.3.2 Naïve Bayes

It is a probabilistic classifier based on strong assumptions that all predictor variables are independent of each other given the class to anticipate the class label for a given problem, this assumption is called class conditional independence[27]. Despite this assumption, the Naïve Bayes classifier's accuracy is comparable with other classifier's accuracy. It is based on Bayes' theorem which allows to calculate the conditional probability. it consists of two parts first is prior probability (it is obtained before any additional information as p(y) and p(x), secondly is posterior probability is adjusted probability using some of additional evidence and information p(x|y)[27].

$$P(Y|X)=\frac{p(y)\ p(x|y)}{p(x)} \qquad 2.3$$

It is used to calculated the probability of belonging to each class of the target variable .so we need to find the probability class of the object x, so it is very important to find the class which give the maximum probability[27]. Suppose a vector $(x_1, x_2, \ldots x_n)$represents the n features of X. Let θ represent the class label of X. The naïve theorem describes the conditional probability of observing X given the class label $\theta, P(\theta|X)$ as a product of several simpler probabilities as shown below[28]:

$$P(\theta|f_1, f_2, \ldots f_n) = \frac{P(\theta)P(f_1, f_2, \ldots, f_n|\theta)}{P(f_1, f_2, \ldots f_n)} \qquad 2.4$$

Where $(f_1, f_2, \ldots f_n)$ represents the features of vector X. Following the assumption of independence, the probability can be expressed as;

$$P(\theta|f_1, f_2, \ldots, f_{i-1}, f_{i+1}, \ldots f_n) = P(f_i|\theta) \qquad 2.5$$

For all i

$$P(\theta|f_1, f_2, \ldots f_n) = P(\theta)\frac{\prod_{i=1}^{n} P(f_i|\theta)}{P(f_1, f_2, \ldots f_n)} \quad 2.6$$

Assuming every feature is constant, the classification rule follows below:

$$P(\theta|f_1, f_2, \ldots f_n)\alpha\, P(\theta)\prod_{i=1}^{n} P(f_i|\theta)\,, \quad 2.7$$

And,

$$\hat{\theta} = arg\max_{\theta} P(\theta)\prod_{i=1}^{n} P(f_i|\theta) \quad 2.8$$

It is worth noting that for the GNB model, the probability of feature occurrence follows a Gaussian distribution

$$P(f_i|\theta) = \frac{1}{\sqrt{2\pi\sigma_\theta}}\exp\left(\frac{-(f_i-\mu_\theta)}{2\sigma_\theta^2}\right) \quad 2.9$$

whose parameters σ_θ and μ_θ are computed using maximum likelihood.

2.1.3.3 Support Vector Machine (SVM)

SVM offers very robust and accurate methods of all known methods. Alongside its sound theoretical foundation, it only requires a few examples for training and doesn't get affected by the number of dimensions. SVM definition can be extended from binary classification to problems that involve more classes to classify, as there are many studies in recent years studying the different kernels (functions used in SVM to help solve problems) for SVM classification[29]. For linearly separable data, the linear classification method creates a separating hyperplane f(x) that separates the data from the middle into two classes.

SVM additionally offers the best function by maximizing the margin between both classes. The margin can be defined as as the space or separation between both classes

defined by the hyperplane, which corresponds to the shortest distance between the closest data points to a point on the hyperplane[29]. When the data is not linearly separable, the kernel-based SVM is used.

Kernel-based SVM overcomes the problem of having non-linearly separable data by implicitly mapping data into feature space, where the linear threshold can be used[30]. Therefore, the non-linear SVM score is a linear combination but with the new variables derived from the kernel transformation. If the need for an additional feature arises, the SVM uses the kernel algorithm to change a low-dimensional input space into a high-dimensional one. In other words, it transforms problems that seem inseparable into integratable ones. If there is no error in this separation, then, the Expected value of error is given as

$$E[\Pr(error)] \leq \frac{E[number\ of\ support\ vectors]}{[number\ of\ training\ vectors]} \qquad 2.10$$

The decision function will be given as

$$D(x) = w\emptyset(x) + b \qquad 2.11$$

which is the best line that integrates the training data, w and b are parameters of the SVM, and $\emptyset(x)$ is the function which transforms the data into the new M dimension

$$\frac{D(x)}{||w||} \qquad 2.12$$

represents the line, which is the distance between item x and the hyperplane. The parameters of the linear decision function that will maximize M are:

$$w = \sum_k a_k y_k x_k \qquad 2.13$$

$$b = (y_k - w * x_k) \qquad 2.14$$

The principal function of the above training algorithms is to solve the equation quadratically[31].

$$J = \left(\frac{1}{2}\right)||w|| \qquad 2.15$$

There are two types of SVM:

Linear SVM which is obtainable when the data is integratable in linear form. This implies that the data points can be clearly integratable by a single straight line[25].

Non-Linear SVM which covers when data is not linearly separable, and advanced techniques (kernel tricks) are applied[32]. It is important to choose a kernel to work with, and this is largely dependent on the dataset at hand. If linear, then a linear kernel function must be adopted. Starting with the hypothesis that the data is linear is ideal, then working through other kernels to compare performance metrics.

SVM performs better on a linear dataset, and its efficiency is enhanced in high-dimensional data. SVM is very robust as it is non-sensitive to outliers[33]. SVM generalization to SVR is accomplished by presenting an ε-insensitive area around the function, named the ε-tube[34,35]. This tube redevelops the optimization problem to find the tube that best approximates the continuous-valued function. SVR is framed as an optimization issue by first defining a convex ε-insensitive loss function to minimize and discover the flattest tube that comprises most of the training instances.

SVR has achieved a better performance in terms of performance than an artificial neural network[36]. However, SVR may not fit with more than 10000 observations. Instead, a LinearSVR version (SVR with its linear kernel) can handle a prediction task with a large dataset of observations.

2.1.4 Class Imbalance

Most real-world applications possess unbalanced class distribution where the number of a class label heavily dominates the count of another class label. One of the best example to explain class imbalance problem is the fraud detection task, where the number of fraud class label is very low as compared to the normal class label.

Most machine learning algorithms work poorly in the presence of unbalanced class distribution (i.e., the predictive model tends to classify the minority example as the majority example). So some questions may arise such as[1]: (i) How to tackle the class imbalance problem? (ii) Which machine learning algorithms should be applied in the presence of unbalanced class distribution? (iii) What evaluation metrics should be used to assess the performance of a predictive model when the dataset is highly unbalanced[1].

There are different approaches in handling class imbalance problem. They can be classified into three categories: resampling approach, ensemble-based approach, and cost-sensitive learning approach. Cost-sensitive learning takes misclassification costs into consideration. For example, in medical diagnosis of cancer, the misclassification cost of missing a cancer is much higher than the cost of predicting that a healthy person has cancer. Hence, by considering the misclassification cost of minority class more heavily than that of majority class, the true positive rate of the model can be improved[1,19].

2.1.4.1 Resampling Approach

Basically, there are three resampling approaches: undersampling, oversampling, and hybrid[37]. Most of the predictive model works worst in the presence of unbalanced class distribution. Therefore, some data preprocessing task has to be performed before providing data as an input to the model. In the case of class imbalance problem, such

data preprocessing is performed using a data level approach, which is called resampling approach.

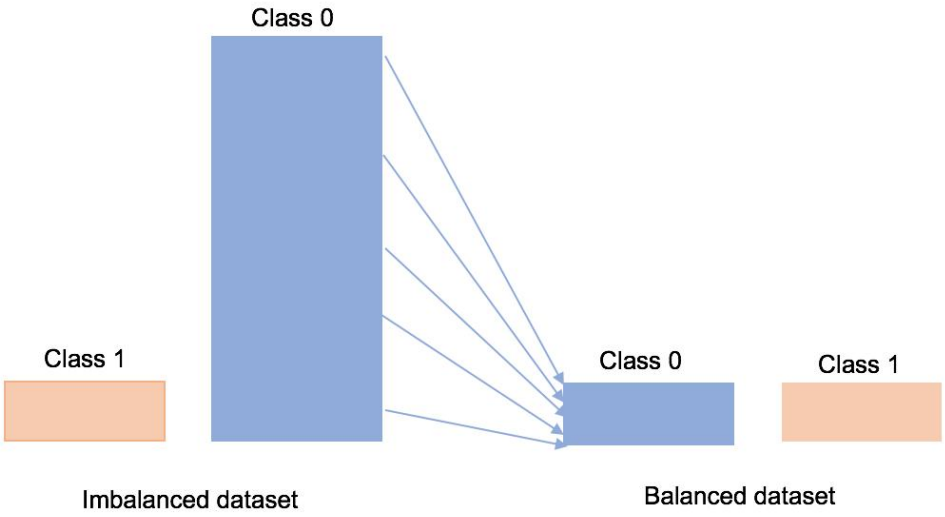


Figure 2.2: Undersampling Approach[38].

In the undersampling method, the majority class is reduced in order to make the dataset balanced, which is shown in figure 2.2. It is best to implement when the size of the dataset is huge and reducing the majority samples can greatly boost the runtime and reduce the storage troubles[38].

An oversampling method is exactly opposite to the undersampling method. This method works with the minority class. It replicates the observations from minority class to balance the ratio between majority and minority sample, which is shown in figure 2.3. At last, a hybrid method applies both undersampling and oversampling method for rebalancing purpose[37].

Random Undersampling: This method randomly eliminates the majority samples in order to make the dataset balanced[39]. This method is best to use when the size of the training data is huge. By reducing the frequency of majority samples, it improves the runtime and also reduces the storage troubles. However, the disadvantage of using

such an approach is that some useful information may be eliminated in the process of eliminating majority samples.

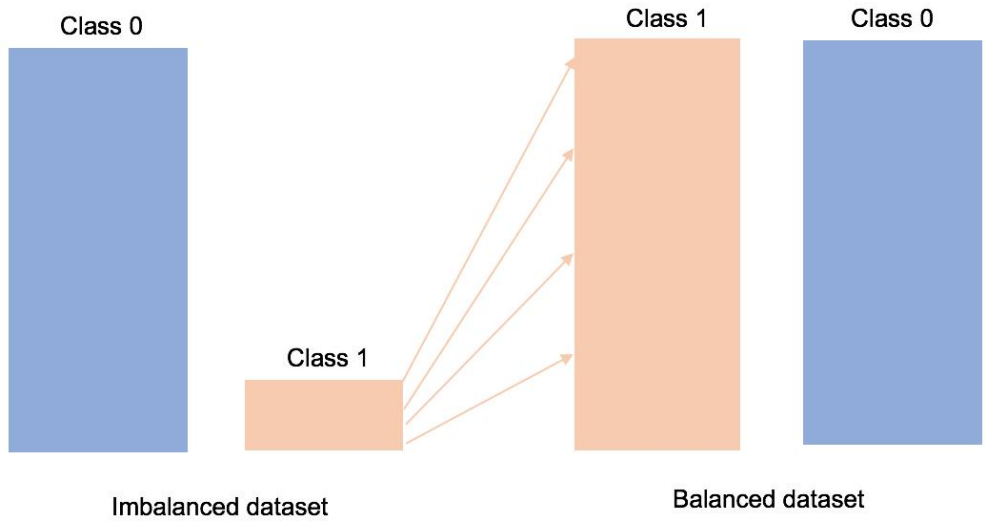


Figure 2.3: Oversampling Approach[38].

Random Oversampling: This method randomly replicates the minority samples to make the dataset balanced[40]. Unlike random undersampling, this approach does not lead to information loss. However, there's a high possibility of overfitting the data since it replicates the minority samples.

Synthetic Minority Over-sampling Technique (SMOTE): SMOTE aims to create new minority class examples (synthetic instances) by interpolating between several nearest minority examples rather than by oversampling with replacement[41]. As a result, it diminishes the problem of overfitting of the training data. Depending upon the amount of oversampling required, nearest neighbors of minority examples are randomly selected.

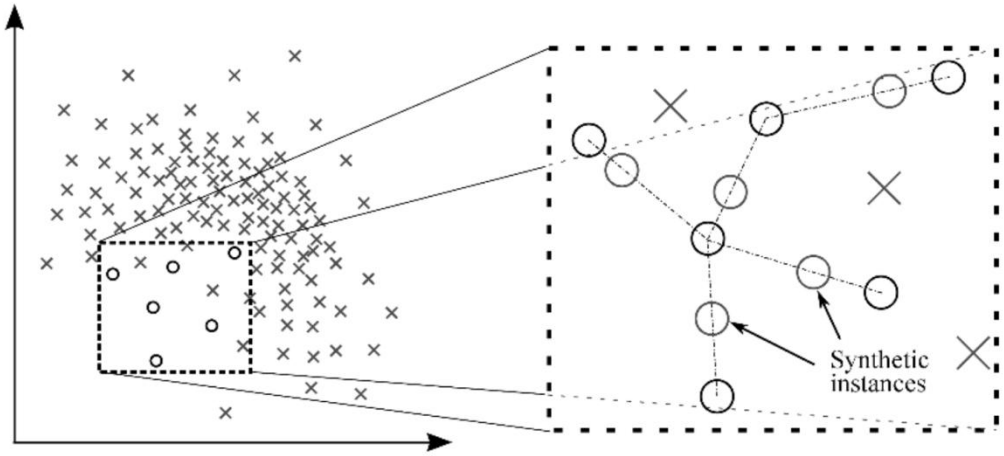


Figure 2.4: Generation of Synthetic Examples Using SMOTE[42].

2.1.4.2 Ensemble Approach

Ensemble approach deals with modifying existing classification algorithms to tackle the unbalanced class distribution[43]. In general, an ensemble approach is a learning algorithm that assembles a set of classifiers and applies them for classification by taking a vote of their predictions[44]. Typically, there are two types of the ensemble approach: bagging and boosting.

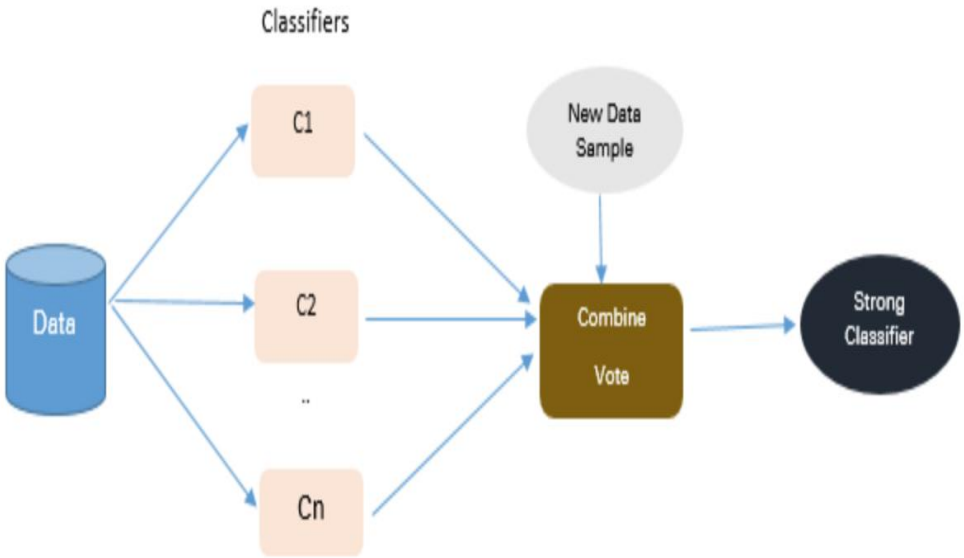


Figure 2.5: Ensemble Approach[45].

In machine learning, bootstrapping is the process of sampling the training data randomly with replacement[46]. Each bootstrap sample is sampled in such a way that each sample will have different characteristics as shown in Figure 2.6. When the models use these samples for training, they can learn various aspects of the data and can improve the prediction performance

Bagging: Bagging which is an abbreviation form of Bootstrap Aggregation is a simple yet very powerful ensemble technique. This method involves bootstrapping that generates new training samples from the original training set with replacement. These new training samples are called bootstrap training samples[47]. Each bootstrap sample is used for training the individual model separately, which are then used for prediction. Finally, the predictions from all the bootstrapped models are aggregated by averaging the output (for regression) or voting (for classification)[47]. Figure 2.7 gives a better picture of bagging. It helps to reduce overfitting and variance. Decision trees are usually used as a base model in bagging.

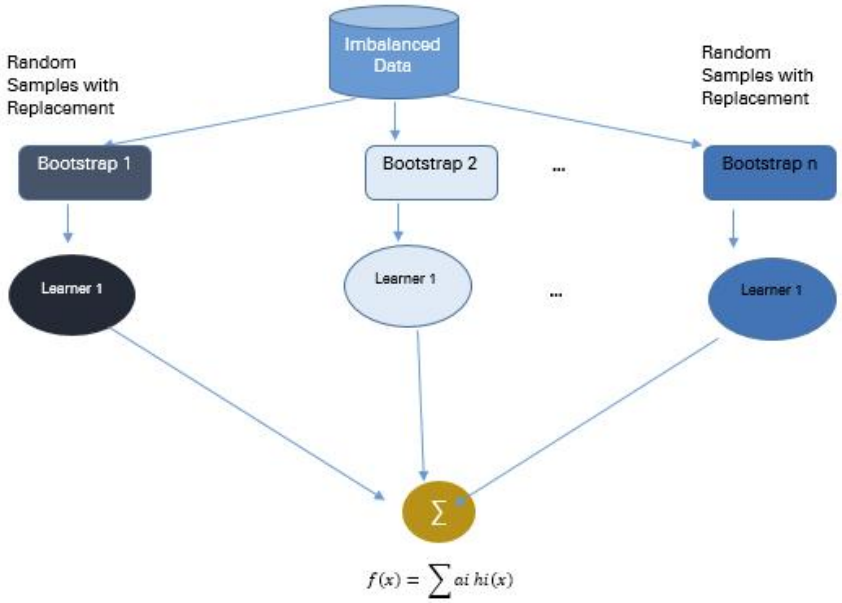


Figure 2.7: Overview of Bagging[48].

Boosting: Boosting is another very powerful ensemble technique. It involves combining weak learners, which are also called base learners, to create a strong learner that can give a better result as compared to the results generated by an individual learner[49]. Unlike bagging in which each model runs parallell and then the outputs are combined at the end, the boosting deals with training the weak learners sequentially, such that each learner tries to correct its predecessor by adding more weights to the samples that were previously misclassified[49]. Therefore, the future weak learner will focus more on the misclassified cases. Figure 2.8 gives a better picture of boosting. It also uses bootstrapping due to which it also avoids overfitting and variance. There are many examples of boosting algorithm like ada boost, gradient boost, xgboost, etc[50].

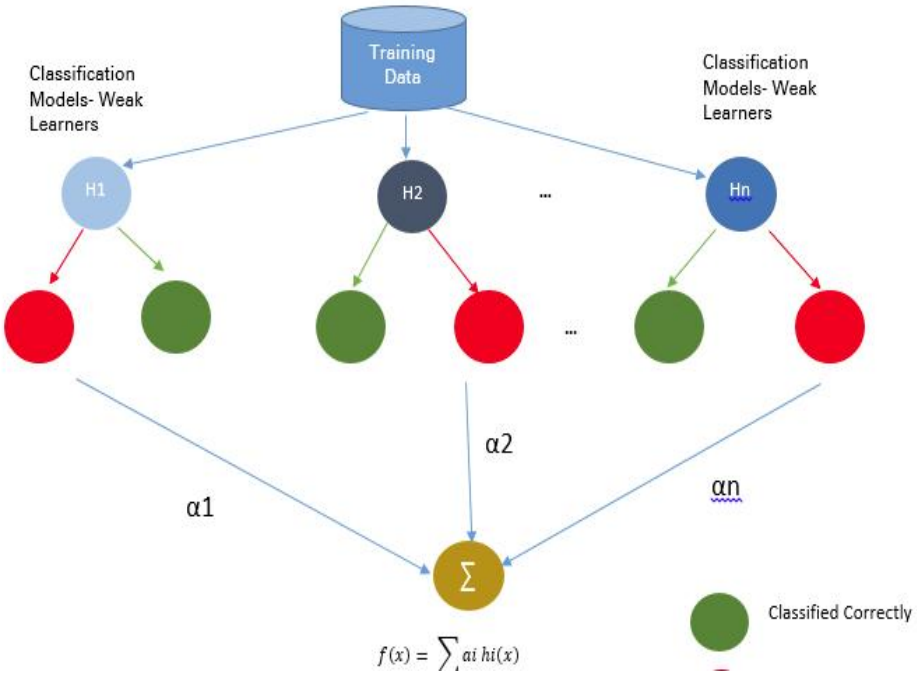


Figure 2.8: Overview of Boosting[51].

Gradient boosting Machines (GBoost)

A GBoost is a popular machine learning algorithm that builds an ensemble of shallow trees sequentially.

GBoost iteratively improves the predictions of y from x with respect to L by adding base learners or new weak that improve upon the previous ones, founding an additive ensemble model of size M[52]:

$g_o(x) = c, g_i(x) = g_{i-1}(x) + \gamma_i h_i(x), \quad i = 1, \ldots, M$ 2.16Where i the iteration index; h_i is the i^{th} base model, for example, a decision tree; γ_i is the weight or the coefficient of the i^{th} base model. GBoost has been used for several regression tasks and achieved better performance than alternative algorithms[52].

Adaptive Boosting: AdaBoost creates a strong learner through multiple iterations of adding a weak learner in each cycle[53]. The weight vector is also tweaked to account for previously misclassified sample points. Hence, the resulting classifier has higher accuracy. Adaboost is not robust to outliers and noise. The hyperparameters in Adaboost are learning_rate, n_estimators, and base_estimator.

Gradient Boosting: Gradient boosting is an improvement of Adaboost, which aims to minimize the loss function by using the gradient descent optimization method while combining weak learners[54]. The loss function estimates the best model depending on the problem task. Weak learners are added based on the additive model component. Gradient boosting is notably improved as it inculcates subsampling, which encourages randomness of the model, shrinkage reduces the impact of each added learner, and the size of added steps, thus penalising consecutive iteration.

Extreme Gradient Boost (XGboost): XGBoost is an improvement of the gradient boost designed to improve speed and performance. This technique employs regularized learning for smoothing and shrinkage to reduce the impact of each tree and make room for future trees to aid improvement, and feature subsampling to prevent over-fitting[55].

All these features, speed up the run time of the algorithm. Parameters of XGboost which can be tuned by the user are eta, gamma, max_depth, seed, eval_metric etc.

Light Gradient Boosting Machine: LightGBm uses two gradient-based methods: exclusive feature bundling (EFB) and gradient-based onside sampling (GOSS)[56]. GOSS operates by excluding the portion of the dataset with relatively small gradients and then uses the remaining data to compute the overall information gain. The EFB uses the mutually exclusive features in the dataset, and non-zero values simultaneously to minimise the number of features. This enhances the overall accuracy of the split point.

2.2 Methodological Review

This section gives a theoretical background of the main classification algorithms used in this study.

2.2.1 Logistic Regression

Logistic regression is one of the most popular machine learning algorithms that is used for classification. In logistic regression, the prediction is expressed in terms of probability of outcome belonging to each class. In a linear regression model, the real-valued outputs are predicted by combining the input variables (x) with the weights. To be more clear, consider there is just one input or independent variable 'x' and a dependent variable 'y'[62].

Logistic Regression is a supervised machine learning technique for predicting the probability of a given class or occurrence[62]. It is employed when the data is separated linearly, and the outcome is binary or categorical. That is, logistic regression is typically applied to binary classification issues. Estimating the output variable, which is discrete in two classes, is referred to as binary classification.

Process: Logistic regression forecasts the outcome of a categorical dependent variable. As a result, the end to be categorical or discrete value. Yes or no, can be 0 or 1, true or false, but instead of displaying exact values like 0 and 1, it displays probabilities in the range 0 to 1[63]. The algorithm can be developed using the equation

$$y = e^{(b0 + b1*x)} / (1 + e^{(b0 + b1*x)}). \quad 2.22$$

The hypothesis of linear regression can be expressed as

$$y = a0 + a1 * x \quad 2.23$$

where a0 is a bias term, and a1 is the weight for single input variable x. These weights are learned during the training. In this case, the value of the hypothesis can be less than 0 or greater than 1[1]. 14 Logistic regression also uses such a linear equation. However, since it should predict the probability of the outcome of belonging to each class, it uses a sigmoid function or a logistic function which is shown in equation 2.23, to squash the predicted real values between the range of 0 and 1.

Figure 2.8 shows how the sigmoid function looks like. In a classification problem where we have one independent variable 'x' and one dependent variable 'y', logistic regression can be represented as in equation 2.24[1]. By default, logistic regression uses a threshold of 0.5 such that any probability below 0.5 is classified as class 0, and any probability above 0.5 is classified as class 1. This threshold can be adjusted according to the needs.

$$P(y = 1) = sigm(a0 + a1 * x) \quad 2.24$$

where a0 and a1 are the parameters of the logistic regression model that are learned during training. Therefore, with a threshold of 0.5, the predicted output can be represented as follows.

$y = 1$ if $P(y=1) \geq 0.5$

y= 0 if P(y=1) < 0.5

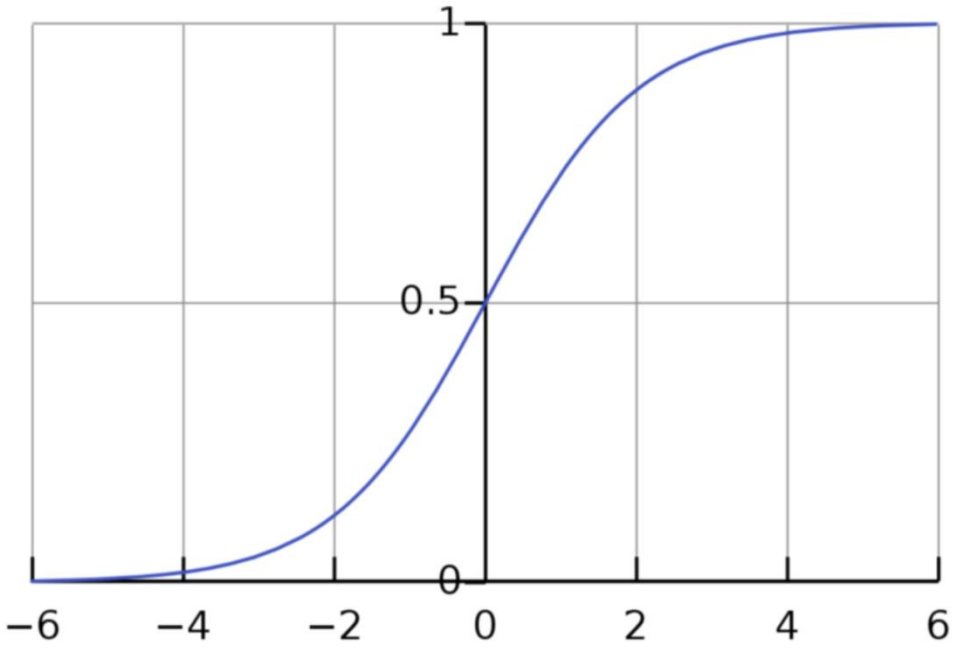


Figure 2.9: Sigmoid Function Graph[64].

2.2.2 Random Forest (RF) Algorithms

Random forest Random forest is an ensemble learning algorithm, which can be used for both regression and classification task[65]. Specifically, it is an extension of bagging. In a random forest, these weak learners are decision trees. In simple language, the random forest builds multiple decision trees and combines them to improve the performance of the model as a whole[65].

Random forest utilizes bootstrapping such that each decision tree will be trained with different subsamples of data[66.] Moreover, the random forest uses random subsets of features.

For example, if there are 50 features in the data, random forest will only choose a certain number of them, let's say 10, to train on each tree. Once there is a collection of decision trees, the results of each tree will be aggregated to get the final result (vote). The model trained in such a way will ensure generalization since not one, but multiple

decision trees are used for making the decision, and moreover, each tree is trained with different subsections of data.

Random forest is a supervised ensemble method that uses a collection of numerous decision trees to make predictions[67]. Random Forest is a classifier consisting of a set of tree-structured classifiers with identically distributed independent random vectors and each tree casting a unit vote at input x for the most popular class[68]. A random vector that is independent of the previous random vectors of the same distribution is generated and a tree is generated using the training test , an upper bound is extracted for Random Forests to get the generalization error in terms of two parameters Exactitude and interdependence of individual classifiers[69,70].

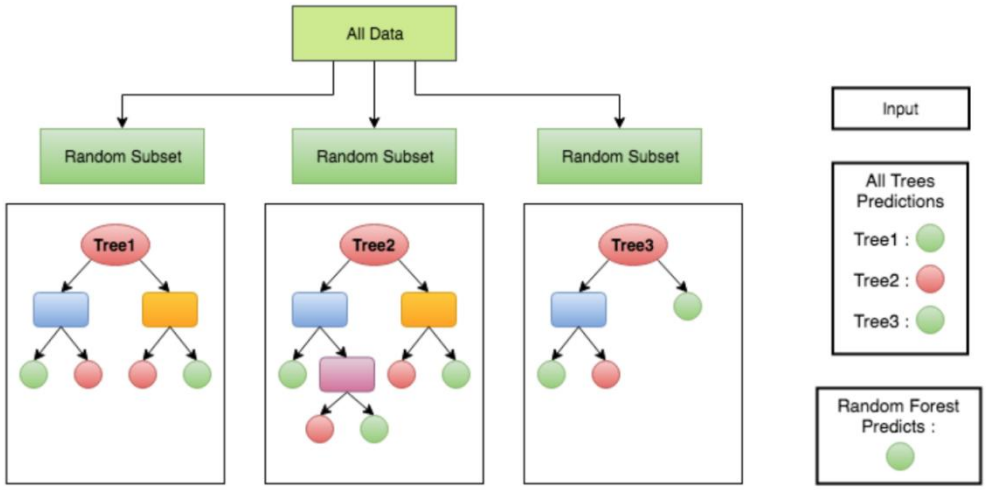


Figure 2.10: Random Forest Flow Chart[71]

To get multiple subsets of samples, it implements the bootstrap method, creates a Decision Tree utilizing each subset of samples, and combines several Decision Trees into a Random Forest[72]. When the sample to be classified is reached, the final outcome of the classification is decided by a vote on the Decision Tree[72]. Generally,

scholars increase the precision of the classifier starting from the classifier and reduce the association between classifiers[73].

Random Forest algorithm in the classification process, where the effects of the classification of each base classifier have a common distribution of errors, the final reduction of the classification effect is accomplished[74]. Takes the test characteristics and uses the rules of each randomly generated Decision Tree to forecast the result and store the expected result (target). Determine the votes for each predicted goal. Consider the predicted high-voted goal as the final prediction from the Random Forest algorithm[75,76].

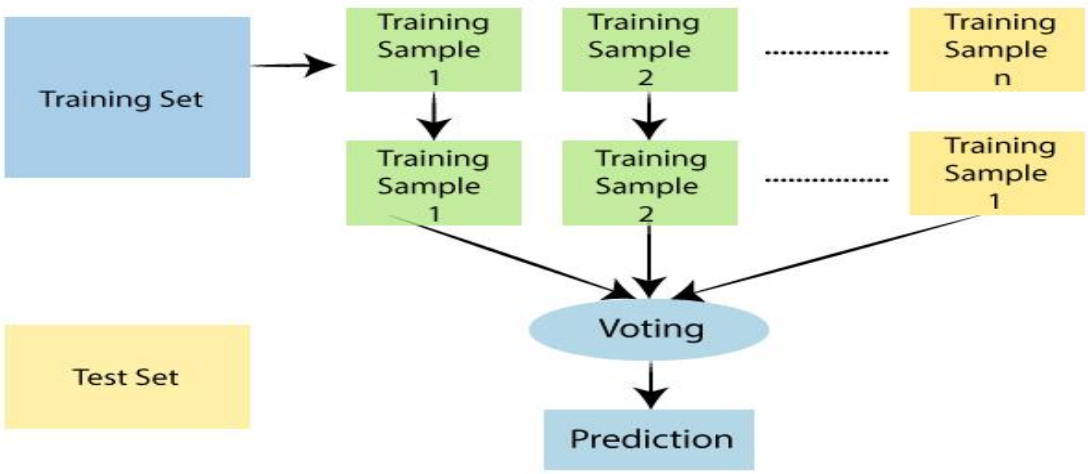


Figure 2.11: Random Forest Training Flow Chart[71].

The RF algorithm is very efficient, as it handles datasets that contain continuous variables, as well as categorical variables robustly. An RF classifier contains subsets of various tree classifiers $\{\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{\Theta_k}), \boldsymbol{k} = \boldsymbol{1}, \boldsymbol{2}, ...\}$ where the $\boldsymbol{\Theta_k}$ are independently and identically distributed random vectors, with each tree being able to specify the modal class at input $\boldsymbol{x}$62.The performance index, which solely approximates the confidence interval (CI) of the RF model is given as

$$mg(x,y) = av_k I(h_k(x,\Theta_k) = y) - \max_{j \neq y} av_k I(h_k(x,\Theta_k) = j) \quad 2.25$$

where $I(.)$ denotes an indicator function, and $av(.)$, the average value. It is observed that as the margin increases, the confidence level also increases. The generalisation error becomes

$$PE^* = P_{x,y}(mg(x,y) < 0), \quad 2.26$$

where $P(.)$ denotes probability. With an increase in trees for all sequences Θ_k, PE^* converges to

$$P_{x,y}(P_\Theta(h(x,\Theta) = y) - \max_{j \neq y} P_\Theta(h(x,\Theta) = j < 0) \quad 2.27$$

Convergence of this generalisation error proves that the RF model does not overfit as more trees are introduced. The upper bound for the generalisation error is given as

$$PE^* \leq \frac{\bar{\rho}(1-s^2)}{s^2}, \quad 2.28$$

where $\bar{\rho}$ is the average correlation value, s is the strength of each tree in the model. An increased strength of individual trees and a low correlation between them produces more accurate prediction results.

2.2.3 Decision Tree

Decision trees are one of the powerful methods commonly used in various fields, such as machine learning, image processing, and identification of patterns[77]. DT are a successive model that unites a series of the basic test efficiently and cohesively where a numeric feature is compared to a threshold value in each test. The conceptual rules are much easier to construct than the numerical weights in the neural network of

connections between nodes. Mainly for grouping purposes, DT is used. Moreover, DT is a usually utilized classification model in Data Mining. The nodes and branches are composed of each tree[77]. Each node represents features in a category to be classified and each subset defines a value that can be taken by the node. Because of their simple analysis and their precision on multiple data forms, decision trees have found many implementation fields.

Decision Tree is a type of supervised machine learning, used for either classification or regression, also used where the data is continuously split according to a certain parameter, and to provide a graphical representation of all the possible solutions. All decisions were dependent on a number of conditions[30]. It starts from the root node and branches off to the number of solutions, just like a tree. The tree starts from the root, then it grows branches and grows bigger and bigger. The main idea is to build a tree T from our set of observations S. if all S belongs to a class C, then the node is a leaf node and receives a label. If not, the algorithm goes to the next most informative attribute and builds sub-trees until goal is met.

To begin with, there is a need to define the most informative attribute, based on the entropy method that measures the homogeneity of the sampled data[30].

$$\text{Entropy S} = -\sum \text{P(x)}\log_2\text{P(x)} \qquad 2.29$$

Also,the information gain that measures the relative change in entropy with respect to the independent attribute is given as:

$$\text{Gain(S, A)} = \text{Entropy(S)} - \sum_{v\in A}\frac{|S_v|}{S}\ \text{x Entropy}(S_v) 2.30$$

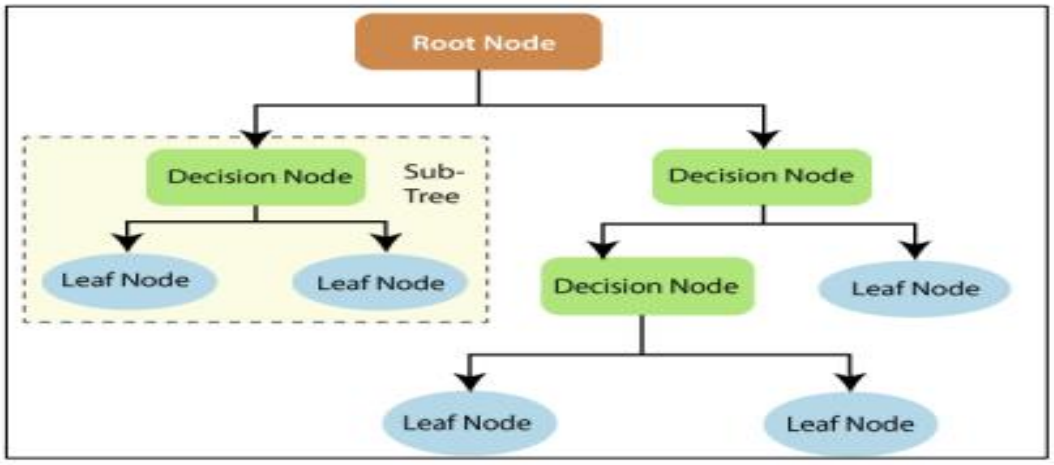


Figure 2.12. Decision Tree Flow Chart[78].

2.3 Review of Empirical Studies

2.3.1 Credit Card Fraud Detection Using Machine Learning

In a work that evaluated a subsection of Deep Learning topologies from the general artificial neural network to topologies with built-in time and memory components such as Long Short-term memory and different parameters with regard to their efficacy in fraud detection on a dataset of nearly 80 million credit card transactions that have been pre-labeled as fraudulent and legitimate. The authors utilize a high performance, distributed cloud computing environment to navigate past common fraud detection problems such as class imbalance and scalability. Their analysis provides a comprehensive guide to sensitivity analysis of model parameters with regard to performance in fraud detection. They also present a framework for parameter tuning of Deep Learning topologies for credit card fraud detection to enable financial institutions to reduce losses by preventing fraudulent activity[79].

In a project to predict credit card fraud using machine learning algorithms. The algorithms used are random forest algorithm and the Adaboost algorithm. The results of the two algorithms are based on accuracy, precision, recall, and F1-score. The ROC curve is plotted based on the confusion matrix. The Random Forest and the Adaboost

algorithms are compared and the algorithm that has the greatest accuracy, precision, recall, and F1-score is considered as the best algorithm that is used to detect the fraud[80].

In another related work, various machine learning and deep learning approaches are used for detecting frauds in credit cards and different algorithms such as Naive Bayes, Logistic Regression, K-Nearest Neighbor (KNN), Random Forest, and the Sequential Convolutional Neural Network are skewed for training the other standard and abnormal features of transactions for detecting the frauds in credit cards. For evaluating the accuracy of the model, publicly available data are used. The different algorithm results visualized the accuracy as 96.1%, 94.8%, 95.89%, 97.58%, and 92.3%, corresponding to various methodologies such as Naive Bayes, Logistic Regression, K-Nearest Neighbor (KNN), Random Forest, and the Sequential Convolutional Neural Network, respectively. The comparative analysis visualized that the KNN algorithm generates better results than other approaches[81].

The study's scientific novelty is the development of machine learning models for identifying fraudulent banking transactions and techniques for preprocessing bank data for further comparison and selection of the best results. This paper also details various methods for improving detection accuracy, i.e., handling highly imbalanced datasets, feature transformation, and feature engineering. The proposed model, which is based on an artificial neural network, effectively improves the accuracy of fraudulent transaction detection. The results of the different algorithms are visualized, and the logistic regression algorithm performs the best, with an output AUC value of approximately 0.946. The stacked generalization shows a better AUC of 0.954. The recognition of banking fraud using artificial intelligence algorithms is a topical issue in our digital society[82].

In another similar study which is based on index system construction using the Logistic regression model. The object of investigation is Chinese companies listed by China Stock Market & Accounting Research database, excluding J66 (remaining financial industry except for the monetary and financial services), J67 (capital market services), J68 (insurance industry) and J69 (other financial industry) enterprises. The period of investigation is 2017–2020. The data sample includes 53 fraudulent and 53 normal Chinese enterprises. The results show that the overall prediction accuracy of the developed model is 83% and robustness test results further verify the rationality and effectiveness of the method. The company's stakeholders could apply the proposed approach for fraud identification to improve the efficiency of financial fraud identification from the technical level[83].

In a work to solve the issue of class imbalance in credit card fraud detection, the authors re-sampled the dataset using the Synthetic Minority over-sampling TEchnique (SMOTE). This framework was evaluated using the following ML methods: Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), Extreme Gradient Boosting (XGBoost), Decision Tree (DT), and Extra Tree (ET). These ML algorithms were coupled with the Adaptive Boosting (AdaBoost) technique to increase their quality of classification. The models were evaluated using the accuracy, the recall, the precision, the Matthews Correlation Coefficient (MCC), and the Area Under the Curve (AUC). Moreover, the proposed framework was implemented on a highly skewed synthetic credit card fraud dataset to further validate the results that were obtained in this research[84].

Another similar research proposed a hybrid machine learning and swarm metaheuristic approach to address the challenge of credit card fraud detection. The novel, enhanced firefly algorithm, named group search firefly algorithm, was devised

and then used to a tune support vector machine, an extreme learning machine, and extreme gradient-boosting machine learning models. Boosted models were tested on the real-world credit card fraud detection dataset, gathered from the transactions of the European credit card users. The original dataset is highly imbalanced; to further analyze the performance of tuned machine learning models, in the second experiment performed for the purpose of this research, the dataset has been expanded by utilizing the synthetic minority over-sampling approach. The performance of the proposed group search firefly metaheuristic was compared with other recent state-of-the-art approaches. Standard machine learning performance indicators have been used for the evaluation, such as the accuracy of the classifier, recall, precision, and area under the curve. The experimental findings clearly demonstrate that the models tuned by the proposed algorithm obtained superior results in comparison to other models hybridized with competitor metaheuristics[85].

Further in a work that proposed a scheme, *RaKShA*, which presents explainable artificial intelligence (XAI) to help understand and interpret the behavior of black box models. XAI is formally used to interpret these black box models. The authors used XAI to extract essential features from the CC fraud dataset, consequently improving the performance of the LSTM model. The XAI was integrated with LSTM to form an explainable LSTM (X-LSTM) model. The proposed approach takes preprocessed data and feeds it to the XAI model, which computes the variable importance plot for the dataset, which simplifies the feature selection. Then, the data are presented to the LSTM model, and the output classification is stored in a smart contract (SC), ensuring no tampering with the results. The final data are stored on the blockchain (BC), which forms trusted and chronological ledger entries. They obtained an accuracy of 99.8%

with our proposed X-LSTM model over 50 epochs compared to 85% without XAI (simple LSTM model)[86].

Another work proposed a new method that can identify and predict financial fraud among listed companies based on machine learning. The authors collected 18,060 transactions and 363 indicators of finance, including 362 financial variables and a class variable. Then, eliminated 9 indicators which were not related to financial fraud and processed the missing values. After that, we extracted 13 indicators from 353 indicators which have a big impact on financial fraud based on multiple feature selection models and the frequency of occurrence of features in all algorithms. Then, they established five single classification models and three ensemble models for the prediction of financial fraud records of listed companies, including LR, RF, XGBOOST, SVM, and DT and ensemble models with a voting classifier. Finally, we chose the optimal single model from five machine learning algorithms and the best ensemble model among all hybrid models. In choosing the model parameter, optimal parameters were selected by using the grid search method and comparing several evaluation metrics of models. The results determined the accuracy of the optimal single model to be in a range from 97% to 99%, and that of the ensemble models as higher than 99%. According to the authors, the optimal ensemble model performs well and can efficiently predict and detect fraudulent activity of companies. Thus, a hybrid model which combines a logistic regression model with an XGBOOST model is the best among all models[87].

In a study that proposes an adaptive boosting algorithm, that was subjected to the optimization process by the social network search algorithm. The proposed hybrid approach has been validated on the imbalanced synthetic credit card fraud detection benchmark dataset, and acquired outcomes were compared to other cutting-edge

machine learning models. The evaluation was performed by utilizing the standard performance indicators—accuracy, recall, precision, Matthews correlation coefficient, and area under the curve. The experimental findings have shown that the proposed SNS-based AdaBoost approach obtained superior results, clearly outperforming all other machine learning models included in the analysis[88].

This research study analyzes the different data mining techniques used for the detection of credit card frauds. Additionally, different data mining techniques are properly described in this study. The steps of data mining are clearly explained in this study, which are helpful for recognizing the fraudulent activities. This study is discussing that Bayesian network, and decision tree are the effective techniques of data mining. The issues of data mining are also discussed in this study, which is creating performance issues, and user interaction issues in the financial institutions. Therefore, this study is providing the importance of RF algorithm in the determination of the credit card fraud. Data cleaning, and data visualization, therefore, the machine learning process is allowed to be developed with the support of the data mining process of credit cards. Clustering is the main process that is takes place in this data mining process. Additionally, the sequential pattern is being highlighted by this process, which helps to improve the fraud detection process of credit cards[89].

In a related study that focused on the application level of CCF detection using Genetic Algorithm (GA) as a feature selection technique. The GA feature selection technique is in two phases, the first phase is designated as the first priority features where eight (8) attributes were selected as the fittest attributes. At second stage which is referred to as the second priority features where another set of eight (8) attributes were considered and selected. The Naïve Bayes (NB), Random Forest (RF) and Support Vector Machine (SVM) supervised machine learning techniques were used for the

detection of CCF on German credit card dataset which is an imbalance dataset. The experimental findings of the proposed model revealed that the first priority features are the most important features. Also, the obtained results showed that the RF algorithm outperformed NB and SVM in terms of accuracy, fraud detection rate and precision[90].

A study aim to improve costs by reducing baseline with a new sample process, discuss the different choices of the optimization metric, and try to show the best performing models for fraud detection. The study has used Baseline logistic regression, Gradient boosted model (GbM), Logistic regression, Logistic Regression Baseline Model (LgBm), Extreme Gradient (XGBoost) classifier with Synthetic Minority Oversampling Technique (SmOtE) and Adaptive Synthetic Sampling Method (AdAsYn), showed the accuracy. In the end, this research study reminds you that fraud, which represents 0.1731% of the studied dataset. The findings and conclusions are based on real-world transactional data given by a big European card processing business[91].

Another study aim to detect credit card fraud using Novel Optimized Random Forest Technique (NORFT) and Gradient Boosting (GB). The Novel Optimized Random Forest Technique Algorithm uses parallel Decision Tree technique in addition with Random Forest Technique to improve the prediction of Credit Card Fadulants. Total sample size of 40 is used for testing and analysis, based on Gpower statistical analysis tool by considering gpower 0.8. In NORFT used N=20 and in GB used N=20 to measuring the performance of both algorithms. Novel Optimized Random Forest Technique provides mean precision of 92.52%, and compared with Gradient Boosting algorithm of mean precision is 88.56%. Statistical significance value was fixed as

(p>0.05) and obtained 0.477, this shows that NORFT is not statistically significant with alternative hypotheses[92].

Similarly, in a study to improve precision for credit card fraud detection by using Novel Optimized Random Forest Technique (NORFT) and comparison with Logistic Regression (LT). In NORFT, it uses multiple Decision Trees to detect the credit card fraud by culminating the maximum attained probability values. The groups consist of NORFT and LR for comparison analysis. The sample size was estimated by using Clinicalc online tool, which is determined as N=2500 for each group with g-power value as 80% and datasets are collected from various web sources with recent study findings and threshold 0.05%, confidence interval 95% mean and standard deviation. The implementation resulted in precision as such NORFT (92.52%) and LR (71.60%). The statistical significance was performed using Independent Sample T-test between the groups, the study has a significance value of (p>0.05) i.e. p=0.649 and states does not have any difference in research[93].

In a research to detect the fraudulent transactions made by credit cards by the use of machine learning techniques. Thedetection of the fraudulent transactions will be made by using three machine learning techniques KNN, SVM and Logistic Regression, those models will be used on a credit card transaction dataset[94].

In a study to demonstrate how modelling data sets are utilized in machine learning to detect credit card fraud. Credit Modelling historical credit card transactions, data from who look to be such fraud are key components of the Finding Card Fraud Problem. This model is then applied to determine if the activity is genuine or not. While reducing the types of fraudulent fraud, our aim is to identify 100% of false employment. A common sample separation to check for credit card scams. We are

concentrating on assessing and ranking data sets in this procedure, as well as providing a variety of perplexing algorithm postings, Local Outlier Factor and Isolation Forest method in PCA changed statistics about how credit cards are processed[95].

A paper proposed a Credit Card Fraud Detection system based on Operational & Transaction features using Support Vector Machine (SVM) and Random Forest (RF) classifiers. In this system, in the first phase, the operational features of users are extracted, and then a random forest classifier is used to classify the features into benign and suspected. In the second phase, the transaction features of users are extracted from the user records, and then the M-class SVM classifier is applied to classify the features into benign and suspected. The performance of the system is evaluated in terms of standard measures precision, accuracy, recall, and F-1 score. By results, it was shown that both RF and SVM classifiers achieve a higher detection rate with good accuracy[96].

A study presented an in-depth review of cutting-edge research on detecting and predicting fraudulent credit card transactions conducted from 2015 to 2021 inclusive. The selection of 40 relevant articles is reviewed and categorized according to the topics covered (class imbalance problem, feature engineering, etc.) and the machine learning technology used (modelling traditional and deep learning). The study shows a limited investigation to date into deep learning, revealing that more research is required to address the challenges associated with detecting credit card fraud through the use of new technologies such as big data analytics, large-scale machine learning and cloud computing. Raising current research issues and highlighting future research directions, our study provides a useful source to guide academic and industrial

researchers in evaluating financial fraud detection systems and designing robust solutions[97].

Another study solved class imbalance problem which have made it difficult for ML classifiers to achieve optimal performance. In order to solve this problem, this paper proposes a robust deep-learning approach that consists of long short-term memory (LSTM) and gated recurrent unit (GRU) neural networks as base learners in a stacking ensemble framework, with a multilayer perceptron (MLP) as the meta-learner. Meanwhile, the hybrid synthetic minority oversampling technique and edited nearest neighbor (SMOTE-ENN) method is employed to balance the class distribution in the dataset. The experimental results showed that combining the proposed deep learning ensemble with the SMOTE-ENN method achieved a sensitivity and specificity of 1.000 and 0.997, respectively, which is superior to other widely used ML classifiers and methods in the literature[98].

Another research proposed some of the classification ML algorithms such as Logistic regression(LR), Linear Discriminant Analysis (LDA), and Naïve Bayes(NB), additionally, the boosting algorithm XGBoost to create models capable of detecting fraud. The dataset from Kaggle. The authors used performance metrics such as accuracy, precision, f1, recall, AUC confusion matrix to evaluate the models' performance. The XGBoost model presented the best results compared to other models[99].

In a research that aims to detect credit card fraudusing random forest classifier in comparison with logistic regression**.** Novel random forest classifier algorithm with sample size =09 and novel logistic regression algorithm with sample size =09 were evaluated many times to predict the efficiency percentage.The G power taken as 0.8

and a =0.05.Random forest classifier has been efficiency(97.12%) when compared to logistic regression algorithm efficiency(95.34%). The statistical significance difference(two-tailed) is 0.001($p<0.05$).Based on the above it is observed that the Random forest classifier with 97.12% has better accuracy than 95.34% Logistic regression in Credit card fraud detection. There is a statistical 2-tailed significant difference in accuracy for algorithms is 0.02($p<0.05$) by independent t-test[100].

In a project that proposed to detect fraudulent transactions based on some methods of machine learning such like Logistic regression (LR), NaiveBayes (NB), as well as the Linear Discriminant Analysis(LDA), in addition to XGBoost algorithm. The techniques used to generate the models of the suggested system, they were trained and evaluated on the basis of two different datasets. Where the first dataset was the European Cardholders, and the second dataset was the Turkish dataset provided by the Yapi Kredi Company. These datasets suffer from a big imbalance problem. Therefore, the authors used SMOTE to fix this issue. The system models' effectiveness was measured employing a variety of metrics, specifically the confusion matrix, accuracy, F1score, recall, precision as well as AUC. Also, the fuzzy membership function was adopted to the dataset in order to raise the system's efficiency. The final results showed the high efficiency of XGBoost[101].

Another research aims to identify the frauds committed using a payment card such as credit cards, debit cards, and also an experiment is performed to find the best suitable algorithm among Random forest and Logistic Regression. Materials and Methods: To stop the fraud detections using Random forest (N=10) and Logistic regression (N=10) with supervised learning that gives insights from the previous data. Results: The precision of the random forest is 76.29% compared with Logistic regression with

accuracy of 74.65% with statistical significance value p=0.03 (p<0.05) using Independent sample t test. Conclusion: This results proved that Random forest was significantly better for Fraud detection than Logistic regression within the study's limits[102].

In a similar paper that proposed a method called autoencoder with probabilistic LightGBM (AED-LGB) for detecting credit card frauds. This deep learning-based AED-LGB algorithm first extracts low-dimensional feature data from high-dimensional bank credit card feature data using the characteristics of an autoencoder which has a symmetrical network structure, enhancing the ability of feature representation learning. The credit card fraud dataset comes from a real dataset anonymized by a bank and is highly imbalanced, with normal data far greater than fraud data. For this situation, the smote algorithm is used to resample the data before putting the extracted feature data into LightGBM, making the amount of fraud data and non-fraud data equal. After comparing the resampled and non-resampled data, it was found that the performance of the AED-LGB algorithm was not improved after resampling, and it was concluded that the AED-LGB algorithm is more suitable for imbalanced data. Finally, the AED-LGB algorithm is comparable with other commonly used machine learning algorithms, such as KNN and LightGBM, and it has an overall improvement of 2% in terms of the ACC index compared to LightGBM and KNN. When the threshold is set to 0.2, the MCC index of AED-LGB is 4% higher than that of the second-highest LightGBM algorithm and 30% higher than that of KNN. It shows that the AED-LGB algorithm has higher performance in accuracy, true positive rate, true negative rate, and Matthew's correlation coefficient[103].

In light of the importance of accurately predicting fraud incidents through payment procedures, this study investigated the credit card payment methods used for movie tickets, using the machine learning logistic regression method to analyze and predict such incidents. This study used a dataset from cinema ticket credit card transactions made in two days of September 2013 by European cardholders, including 284,807 transactions out of which 492 were fraudulent purchases. The results of the proposed method showed a prediction accuracy of 99%, proving its high prediction performance[105].

In a paper that proposed a technique for identifying credit card fraud that first accounts for customer spending patterns by aggregating transactions to creative new features based on periodic data. The authors considered benefits and costs when training an XGBoost classifier in order to achieve maximum benefits. They evaluated the performance of the classifier using benefits and costs. The authors demonstrated the effectiveness of our approach using data provided by a bank[106].

Another related work aims to concentrate on machine learning (ML) methods thereby proposing a credit card fraud discovery scheme to detect fraud. The ML techniques employed are Decision Tree (DT) and K-Nearest Neighbor (KNN) ML classification techniques. The performance outcomes of the two ML classification techniques are evaluated depending on accuracy, precision, specificity, recall, f1-score, and false-positive rate (FPR). The area under the ROC curve (AUC) of the receiver operating characteristics (ROC) curve was similarly drawn built on the confusion matrix for both classifiers. The two classification techniques were evaluated and compared using the performance metrics mentioned earlier and it was demonstrated that the KNN technique outperformed that of the DT with a greater ROC curve value of 91% for

KNN and 86% for DT. It was concluded that KNN is considered a better ML classification technique that can be employed to discover credit card fraudulent activities[107].

A research explored the use of a technique yet to be employed for credit card fraud detection (CCFD) namely Naïve Bayes. The classifier was compared using a confusion matrix for performance matrices like accuracy, precision, recall, f-measure, and ROC-AUC. It was discovered that NB outperformed most of the ML classifiers employed in state-of-the-art compared with an accuracy of 97.99%, recall of 98.02%, the precision of 99.97%, f-measure 98.98%, and FPR of 0.1971[108].

In this paper, the authors used ARO algorithm to classify the bank transactions into fraud and legitimate. ARO is taken from asexual reproduction. Asexual reproduction refers to a kind of production in which one parent produces offspring identical to herself. In ARO algorithm, an individual is shown by a vector of variables. Each variable is considered as a chromosome. A binary string represents a chromosome consisted of genes. It is supposed that every generated answer exists in the environment, and because of limited resources, only the best solution can remain alive. The algorithm starts with a random individual in the answer scope. This parent reproduces the offspring named bud. Either the parent or the offspring can survive. In this competition, the one which outperforms in fitness function remains alive. If the offspring has suitable performance, it will be the next parent, and the current parent becomes obsolete. Otherwise, the offspring perishes, and the present parent survives. The algorithm recurs until the stop condition occurs.Results showed that ARO had increased the AUC (i.e. area under a receiver operating characteristic (ROC) curve), sensitivity, precision, specificity and accuracy by 13%, 25%, 56%, 3% and 3%, in

comparison with AIS, respectively. The authors achieved a high precision value indicating that if ARO detects a record as a fraud, with a high probability, it is a fraud one. Supporting a real-time fraud detection system is another vital issue. ARO outperforms AIS not only in the mentioned criteria, but also decreases the training time by 75% in comparison with the AIS, which is a significant figure[109].

In a study that proposes to utilize a novel hybrid scheme that integrates two mechanisms: a universal model and a unique model. The universal model is a static mechanism that inspects transactions without regard to the cardholder's history or any other related transaction. It does so by implementing rules that are obtained via analyzing the complete population. On the other hand, the unique model is a dynamic, behavioral scheme that establishes a separate profile for each respective cardholder. In doing so, the model can establish a specific and accurate system that judges said cardholder's transactions. It was found that the integration of the two models greatly enhanced the performance of the overall system. The system is inherently capable of handling the class imbalance problem that is usually prevalent in credit card fraud classification. The proposed framework was implemented and tested on a typical dataset. The proposed framework exhibited superior performance when benchmarked with similar frameworks. It showed a very high fraud detection rate, high balanced classification rate, high Matthews' correlation coefficient and a very minimal false alarm rate110.

In this study, an approach for detecting credit card fraud using machine learning methods, such as K-Nearest Neighbors, random forest, decision trees, logistic regression, and support vector machines, is proposed. The research attempts to look at the effectiveness of the classification models while applying both the Oversampling and Undersampling techniques to find instances of fraud in the dataset for fraudulent

activities. The experimental study used two days of credit card transactions made by European cardholders in September 2013. To evaluate the models' performances, confusion matrix, precision, recall, f1_score, cross-validation score, and ROC_AUC score metrics were used. From different experiments of the tested model, it can be easily observed that the performance of all models was better compared with previous literature thus the KNN was the best in almost all metrics used[111].

In this study, the authors investigate the use of logistic regression, a machine learning algorithm, to detect fraudulent credit card transactions in an imbalanced dataset where only a small fraction of transactions are fraudulent. To address the issue of imbalanced data, the authors employed under sampling of the majority class and oversampling of the minority class. Our results show that the logistic regression model can achieve an accuracy of 94% in detecting fraudulent transactions. Additionally, they performed feature importance analysis to identify the most significant variables that contribute to fraud detection. Their findings suggest that variables such as transaction amount, country of origin, and time of day can be strong predictors of fraudulent transactions. our study highlights the potential of logistic regression for credit card fraud detection, even with an imbalanced dataset[112].

In a research that used ML algorithms like Decision Tree (DT), k Nearest Neighbors (KNN), Logistic Regression (LR) and Random Forest (RF) were analyzed and it is inferred that the RF algorithm works best on performance measures like precision, accuracy, recall, Matthews Correlation Coefficient, F-1 score. This paper also proposes to detect 100% fraud transactions while reducing false negatives by using SMOTE (oversampling technique) in combination with Random Forest algorithm[113].

In another paper that aims to perform an optimum solution of imbalance classification problem on a real-life scenario like a fraudulent transaction data. The main objective of this research is to improve testing accuracy in imbalance classification problem. For this purpose, a combination of Random Forest (RF) classifier and repeated stratified k-fold, grid search cross-validation, Synthetic Minority Oversampling Technique (SMOTE), and Random Under-Sampling (RUS) are applied to perform classification. From the experimental results, it is reported that the RF with grid search cross-validation provides the maximum performance in classification accuracy on a highly imbalanced credit card transaction data[114].

Another study demonstrates how to model utilizing multiple classifiers and data balance using machine learning approaches to learning about Credit Card Fraud Detection. The data has been observed as an imbalanced dataset that could have inferred not much optimal performance of models. The experimentation on the imbalanced data has been done and observed that XGBoost has yielded good performance with 0.91 precision score and 0.99 accuracy score. The different sampling techniques have been carried out in procedure so as to enhance the scores in terms of precision, recall, f1-score, and accuracy. The Random Oversampling technique has come out to be the best suited technique over the imbalance data and yields 0.99 precision and 0.99 accuracy score, when applied on the best model i.e., XGBoost.The models are then used to compare the results of all of the classifiers employed, resulting in varied conclusions and further research. While working on the study, many data balancing procedures such as oversampling, under sampling, and SMOTE are used, with XGBoost beating residual algorithms with a 99% accuracy score and precision score when Random Over-Sampling is considered[115].

To build optimal models, four techniques were used in this research to sample the datasets including the baseline train test split method, the class weighted hyperparameter approach, and the undersampling and oversampling techniques. Three machine learning algorithms were implemented for the development of the models including the Random Forest, XGBoost and TensorFlow Deep Neural Network (DNN). Our observation is that the DNN is more effcient than the other 2 algorithms in modelling the under-sampled dataset while overall, the three algorithms had a better performance in the oversampling technique than in the undersampling technique. However, the Random Forest performed better than the other algorithms in the baseline approach. After comparing our results with some existing state-of-the-art works, we achieved an improved performance using real-world datasets[116].

In a paper that implements the random forest (RF) algorithm to solve the issue in the hand. A dataset of credit card transactions was used in this study. The main problem when dealing with credit card fraud detection is the imbalanced dataset in which most of the transaction are non-fraud ones. To overcome the problem of the imbalanced dataset, the synthetic minority over-sampling technique (SMOTE) was used. Implementing the hyperparameters technique to enhance the performance of the random forest classifier. The results showed that the RF classifier gained an accuracy of 98% and about 98% of F1-score value, which is promising. We also believe that our model is relatively easy to apply and can overcome the issue of imbalanced data for fraud detection applications[117].

Another study proposed ATM fraud detection in static and streaming contexts respectively. In the static context, we investigated a parallel and scalable machine learning algorithms for ATM fraud detection that is built on Spark and trained with a

variety of machine learning (ML) models including Naive Bayes (NB), Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Gradient Boosting Tree (GBT), and Multi-layer perceptron (MLP). The authors employed several balancing techniques like Synthetic Minority Oversampling Technique (SMOTE) and its variants, Generative Adversarial Networks (GAN), to address the rarity in the dataset. In addition, we proposed a streaming based ATM fraud detection in the streaming context. The sliding window based method collects ATM transactions that are performed within a specified time interval and then utilizes to train several ML models, including NB, RF, DT, and K-Nearest Neighbour (KNN). The authors selected these models based on their less model complexity and quicker response time. In both contexts, RF turned out to be the best model. RF obtained the best mean AUC of 0.975 in the static context and mean AUC of 0.910 in the streaming context. RF is also empirically proven to be statistically significant than the next-best performing models[118].

In a study that used data set to simulate fraudulent transactions and construct the model after processing the characteristic data and selecting the features. In the model construction part, four algorithms are used as the trained classifiers, which are K Nearest Neighbor algorithm, Bagging algorithm, Logistic Regression algorithm and Gaussian Bayesian algorithm. First, the four algorithms are used to train the data set respectively, and then the parameters of each classifier are adjusted respectively, so that each individual classifier achieves the optimal performance. On this basis, a new model is constructed by using the method of model fusion. The classifier constructed by K Nearest Neighbor algorithm, Bagging algorithm and Gaussian Bayesian algorithm is used as the primary learner, and the classifier constructed by logistic regression algorithm is used as the secondary classifier to build a fused model as a

whole. The final experimental results will give the classification performance of the five models, including individual models. The results show that the performance of the model after integrating the four individual models is better than that of the single model, which can provide accurate feedback for credit card holders on whether there is fraud[119].

In a research that centers on cases of fraudulent activity from open-source data from kaggle.com. Fraudulent activities are examined by employing a sequence of machine learning models, and the optimal approach is determined through an extensive analysis process. We used three algorithms, namely the random forest algorithm, the Decision Tree classifier algorithm, linear regression and three sampling techniques in order to balance the dataset. We also used twelve (12) different models for the prediction of credit card fraud. The evaluation offers a comprehensive guide for the selection of an ideal algorithm based on the nature of fraudulent activities. Additionally, we demonstrate the evaluation process using a suitable metric for performance measurement. The twelve models were compared, and the best model, with an accuracy of 97.4%, was a Random Forest Classifier developed using the SMOTE sampling technique after hyperparameter tuning[120].

In a research that aims in using the multiple algorithms of Machine learning such as support vector machine (SVM), k-nearest neighbor (Knn) and artificial neural network (ANN) in predicting the occurrence of the fraud. Further, we conduct a differentiation of the accomplished supervised machine learning and deep learning techniques to differentiate between fraud and non-fraud transactions[121].

In a study that presented an approach to predict legitimate or fraud transactions on the IEEE-CIS Fraud Detection dataset provided by Kaggel. Our model is BiLSTM-

MaxPooling-BiGRU-MaxPooling which based on bidirectional Long short-term memory (BiLSTM) and bidirectional Gated recurrent unit (BiGRU). We also applied six machine learning classifiers which are: Naïve base, Voting, Ada boosting, Random Forest, Decision Tree, and Logistic Regression. Comparing the results from machine learning classifiers and our model the results show that our model achieved better as we got 91.37% score[122].

In this research study, the main aim is to detect such frauds, including the accessibility of public data, high-class imbalance data, the changes in fraud nature, and high rates of false alarm. Machine, Logistic Regression and XG Boost. However, due to low accuracy, there is still a need to apply state of the art deep learning algorithms to reduce fraud losses. The main focus has been to apply the recent development of deep learning algorithms for this purpose. Comparative analysis of both machine learning and deep learning algorithms was performed to find efficient outcomes. The detailed empirical analysis is carried out using the European card benchmark dataset for fraud detection. A machine learning algorithm was first applied to the dataset, which improved the accuracy of detection of the frauds to some extent. Later, three architectures based on a convolutional neural network are applied to improve fraud detection performance. Further addition of layers further increased the accuracy of detection. A comprehensive empirical analysis has been carried out by applying variations in the number of hidden layers, epochs and applying the latest models. The evaluation of research work shows the improved results achieved, such as accuracy, f1-score, precision and AUC Curves having optimized values of 99.9%,85.71%,93%, and 98%, respectively[123].

Another similar study proposes a hybrid feature-selection technique consisting of filter and wrapper feature-selection steps to ensure that only the most relevant features are used for machine learning. The proposed method uses the information gain (IG) technique to rank the features, and the top-ranked features are fed to a genetic algorithm (GA) wrapper, which uses the extreme learning machine (ELM) as the learning algorithm. Meanwhile, the proposed GA wrapper is optimized for imbalanced classification using the geometric mean (G-mean) as the fitness function instead of the conventional accuracy metric. The proposed approach achieved a sensitivity and specificity of 0.997 and 0.994, respectively, outperforming other baseline techniques and methods in the recent literature[124].

This paper describes the ways in which forensic credit card fraud detection can be achieved using deep neural network (DNN). The problem that this research addresses is to identify a technique that can be used by forensic experts in credit card fraud investigations to detect if a fraudulent transaction has occurred. It is an obligation that institutions providing financial services must implement relevant safeguards against credit card fraud to prevent possible loss of their investments or clients' funds. This paper presents a forensic detection model of credit card fraud that is based on sequential data modeling using Long Short-Term Memory (LSTM) DNNs. The current study determines whether LSTM-attention algorithm can identify the most important transactions in an input sequence that provides high-accuracy prediction of fraudulent transactions. The effectiveness of the LSTM-attention model is achieved by selecting the most relevant predictive features, uniform manifold approximation, transaction sequences, and attention mechanisms that improve the performance of the model. The results show that LSTM-attention algorithms can be used to conduct forensic credit card fraud detection with high accuracy and precision. The novelty of

the research paper is that it successfully uses an LSTM-attention algorithm to detect credit card fraud instances and proves the applicability of the model in mitigating fraudulent transactions in banking institutions[125].

In this study, the authors developed a hybrid CNN-SVM model for detecting fraud in credit card transactions. The effectiveness of our suggested hybrid CNN-SVM model for detecting fraud in credit card transactions was tested using real-world public credit card transaction data. The architecture of our hybrid CNN-SVM model was developed by replacing the final output layer of the CNN model with an SVM classifier. The first classifier is a fully connected layer with softmax that is trained using an end-to-end approach, whereas the second classifier is a support vector machine that is piled on top by deleting the final fully connected and softmax layer. According to experimental results, our hybrid CNN-SVM model produced classification performances with accuracy, precision, recall, *F*1-score, and AUC of 91.08%, 90.50%, 90.34%, 90.41, and 91.05%, respectively[126].

In a work that proposed a Data Characteristic Stability based feature selection by implementing the Random Forest algorithm for Credit Card Fraud Detection. After implementing the above optimized method for credit card fraud detection, the proposed detection engine result is compared against without applying Data Characteristic Stability using the following ML classifiers: Decision Tree (DT), Random Forest (RF), Logistic Regression(LR), XGBoost and etc. The Proposed solution is considered to be efficient an way because of the tendency of the drift happens after its deployed into production, where now we identify the indicative data drift during the Model development and helps to outperform in identifying the fraudulent cases way better than the traditional methods used in the market[127].

In a study that explores the opportunities, challenges, and future directions of blockchain-enabled federated learning for credit card fraud detection. The combination of federated learning and blockchain can provide a secure and private platform for credit card fraud detection. Blockchain-enabled federated learning offers several opportunities, including improved privacy, security, and collaboration among different financial institutions. The successful implementation of blockchain-enabled federated learning can revolutionize credit card fraud detection by providing a secure and private platform for collaborative model training. This paper emphasizes the potential of blockchain-enabled federated learning for credit card fraud detection and highlights the need to address the challenges associated with this technology. It is essential to continue exploring and developing blockchain-enabled federated learning to ensure the security and privacy of sensitive financial data while promoting collaboration and innovation in the financial industry[128].

Additionally, a study used the Logistic Regression technique, estimate the accuracy % of credit card fraudulent transactions. The accuracy percentage of credit card fraudulent transactions was predicted using a novel decision boundary logistic regression with a sample size of 100 and an artificial neural network (ANN) with a sample size of 100, a 95 percent confidence interval, and a pretest power of 80% iterated at various times. The sigmoid function is used in logistic regression to map values between 0 and 1, which aids in improving the accuracy percentage prediction. Logistic regression has a significantly higher accuracy rate (98.2 percent) than ANNs (88.8 percent). With (p=0.001) (p0.005), there was a statistically significant difference between Logistic regression and ANN. The Logistic Regression algorithm demonstrates a higher predictive accuracy percentage for fraudulent credit card transactions[129].

A research utilizes a brand-new dataset with all raw input variables and a Université Libre de Bruxelles (ULB) transaction dataset, which has been preprocessed with PCA technology. Since imbalanced datasets can affect the training quality, we further preprocess the datasets using random under-sampling and the Synthetic Minority Oversampling Techniques (SMOTE) to balance the datasets. Through the experimental results, we find that the networks perform well for the ULB dataset with 93% accuracy in prediction, but perform poorly for the independent input dataset, and the performance can be improved when increasing the complexity of the network. We also notice that the under-sampling method helps improve prediction accuracy better than the oversampling method. The results indicate that more complicated networks are required to detect fraud when the criterion for fraud is stricter, while balancing the dataset before training will improve the results[130].

In a paper that proposed an enhanced CSat (Customer Satisfaction)-related AdaBoost. Based on the traditional AdaBoost, the authors considered the expected loss of the impact of customer satisfaction and re-adjust the weight of different categories in the cost adjustment function of the basic classifier. Considering the serious consequences of fraud transactions, they also implemented a metric related to the Total Profit of Classification (TPC) to evaluate performance. The results show that the CSat-related AdaBoost performed better in F1-score and AUC score compared to the traditional AdaBoost and some mainstream models, the reliability and interpretability of TPC as an evaluation metric is also demonstrated in the paper[131].

A similar study used the Logistic Regression techniqueto forecast the accuracy % of credit card fraudulent transactions. For forecasting the accuracy percentage of credit card fraudulent transactions, Logistic Regression with sample size=100 and Random

forest(RF) with sample size=100 were iterated with 95 percent confidence interval and pretest power of 80 percent at various periods. Because the sigmoid function used in logistic regression maps the value between 0 and 1, it aids in the prediction of accuracy % by improving the accuracy percentage prediction. When compared to the accuracy of Random forest, the accuracy of Logistic regression is much higher (98.2 percent) (92.4). A statistically significant difference existed between Logistic regression and RF, with (p=0.000) (p0.005) indicating a statistically significant difference. The Logistic Regression method aids in the prediction of credit card fraudulent transactions with more accuracy than other algorithms[132].i

In a paper that focuses on current credit card fraud practices and fraud detection methods implemented in real time. Different ML algorithms like fuzzy-based SVM (FSVM), random forest (RF), logistic regression (LR), and support vector machine (SVM) for fraudulent transaction detection on the dataset collected from credit card users have been used to classify legitimate and fraudulent transactions. The comparative analysis of the credit card fraud detection scheme using these classification models was performed with precision, accuracy, sensitivity, and specificity. The comparative analysis outcomes showed that the highest performance was given by the FS VM over other algorithms with an accuracy of 98.61%[133].

In this research, the authors suggest a CatBoost-based system for detecting credit card fraud. The approach seeks to correctly identify fraudulent transactions while minimising false positives to avoid upsetting loyal consumers. A sizable dataset of credit card transactions, including details such transaction amount, time, and location, was used to train the model. In order to manage categorical variables and lessen the effects of imbalanced data, which is a prevalent problem in credit card fraud detection,

the CatBoost method is utilised. The model's efficiency in spotting fraudulent transactions is assessed using a variety of performance indicators, including precision, recall, and F1-score. The CatBoost algorithm, which can be used in real-world scenarios to reduce financial losses and defend customers' interests, is anticipated to be used by the suggested system to detect credit card fraud[134].

Another study is to classified a dataset of credit card security problems by employing six different machine learning (ML) approaches. The Support Vector Machine (SVM), Random Forest (RF), Bagged Tree, K-Nearest Neighbor (KNN), Naive Biased Classifier, and Extreme Gradient Boosting were selected as the classifiers to use (XGBoost). The classification accuracy of the machine learning algorithms was compared with that of a technique for categorization that is based on deep learning called Long Short-Term Memory (LSTM). The KNN machine learning approach had a maximum accuracy of 97.50 percent, while the LSTM machine learning method had an accuracy of more than 96 percent and promised to give biologically appropriate control of upper-limb movement. In addition to enhancing accuracy, the research has investigated how the effects of removing the channel with the most noise from the algorithms can have on accuracy. This was done in an effort to handle data in a more effective manner[135].

This work intends to illustrate the modelling of a data set using machine learning with Credit Card Fraud Detection. The Credit Card Fraud Detection Problem includes modelling past credit card transactions with the data of the ones that turned out to be fraud. This model is then used to recognize whether a new transaction is fraudulent or not. Our objective here is to detect 100% of the fraudulent transactions while minimizing the incorrect fraud classifications. Credit Card Fraud Detection is a

typical sample of classification. In this process, we have focused on analysing and pre-processing data sets as well as the deployment of multiple anomaly detection algorithms such as Local Outlier Factor and Isolation Forest algorithm on the PCA transformed Credit Card Transaction data[136].

In a work that aimedto design and develop a novel fraud detection method for Streaming Transaction Data, with an objective, to analyse the past transaction details of the customers and extract the behavioural patterns. Where cardholders are clustered into different groups based on their transaction amount. Then using sliding window strategy, to aggregate the transaction made by the cardholders from different groups so that the behavioural pattern of the groups can be extracted respectively. Later different classifiersare trained over the groups separately. And then the classifier with better rating score can be chosen to be one of the best methods to predict frauds[137].

In a project that focused on credit card fraud detection in real world scenarios. The author designed a model to detect the fraud activity in credit card transactions. They collected the credit card usage data-set by users and classify it as trained and testing dataset using a random forest algorithm and decision trees. Using this feasible algorithm, they analyze the larger data-set and user provided current data-set. Then augment the accuracy of the result data. Proceeded with the application of processing of some of the attributes provided which can find affected fraud detection in viewing the graphical model of data visualization. The performance of the techniques is gauged based on accuracy, sensitivity, and specificity, precision. The results is indicated concerning the best accuracy for Random Forest are unit 98.6% respectively[138].

In a work to survey on the various methods applied to detect credit card frauds. From the abnormalities, in the transaction, the fraudulent one is identified. The authors addressed this issue in order to implement some machine learning algorithm like random forest, logistic regression in order to detect this kind of fraud. They also increased the efficiency in finding the fraud. However, they discussed and evaluated employee criteria. They implement an intelligent algorithm which will detect all kind of fraud in a credit card transaction. The authors also found a pattern of each customer in between fraud and legal transaction. Isolation Forest Algorithm and Local Outlier Factor are used to predict the pattern of transaction for each customer and a decision is made according to them. In order to prevent data from mismatching, all attribute are marked equally[139].

Another research paper is focused on credit card fraud detection in real world scenarios. In this proposed paper the authors designed a model to detect the fraud activity in credit card transactions. This system can provide most of the important features required to detect Illegal and Valid transactions. Initially, they collect the credit card usage data-set by users and classify it as trained and testing dataset using a Random Forest algorithm, Bernoulli Naive Bayes Model and Logistic Regression algorithm. Using these algorithms, and analyse the larger data-set and user provided current data-set. Then augment the accuracy of the result data. Proceeded with the application of processing of some of the attributes provided which can find affected fraud detection in viewing the graphical model of data visualization. The performance of the techniques is gauged based on accuracy, sensitivity, and precision1[40].

2.3.2 Real-time Credit Card Fraud Detection and Reporting System Using Machine Learning

In a closely related work that focused on four main fraud occasions in real-world transactions. Each fraud is addressed using a series of machine learning models and the best method is selected via an evaluation. This evaluation provides a comprehensive guide to selecting an optimal algorithm with respect to the type of the frauds and we illustrate the evaluation with an appropriate performance measure. Another major key area that we address in our project is real-time credit card fraud detection. For this, we take the use of predictive analytic done by the implemented machine learning models and an API module to decide if a particular transaction is genuine or fraudulent. We also assess a novel strategy that effectively addresses the skewed distribution of data[141].

Another related work focused fraud detection system. The author proposed a live credit card fraud detection system based on a deep neural network technology. The proposed model is based on an auto-encoder and it permits to classify, in real-time, credit card transactions as legitimate or fraudulent. To test the effectiveness of our model, four different binary classification models are used as a comparison. The Benchmark shows promising results for our proposed model than existing solutions in terms of accuracy, recall and precision142.

In another study, a real time fraud detection system based on service oriented architecture (SOA) has been proposed to analyze the fraudulent activities in credit card transactions. The architecture is designed on the basis of Apache Kafka tool, which is used for real time streaming of transactional data to detect the fraudulent transactions. This process considers different services which form the backbone of

SOA. Further, five different machine learning classifiers namely support vector machine (SVM), multilayer perceptron (MLP), random forest regressor, autoencoder and isolation forest have been considered to identify the fraudulent activities instantly with the help of SOA based real time architecture[143].

2.4 Chapter Summary and Gap in Literature

This chapter was organised into four sub-headings - conceptual review, theoretical review/framework, review of empirical studies related to the research topic and conceptual model. The conceptual review explained in depth the concepts of the study. These concepts are - credit card fraud, fraud detection process, machine learning and types, supervised learning classification algorithm like KNN, Naive Bayes, Support Vector Machine (SVM), class imbalance, evaluation metrics. It also richly gave insights into sub-concepts on handling imbalance class like; resampling approach (under sampling, over sampling), essemble approach and cost-sensitive learning approach.

In the methodological review, the main classification algorithms used in this study were fully explained. They are Random Forest (RF) Algorithms, which involves two steps, one is Random Forest formation, and the other is to make a guess from the first step of the Random Forest classifier, Logistic Regression where prediction is expressed in terms of probability of outcome belonging to each class and Decision Tree, where each node represents features in a category to be classified and each subset defines a value that can be taken by the node

In the review of empirical studies, several studies on credit card fraud classification and detection using machine learning algorithms were presented. Also, few closely

related studies on real time credit card fraud detection and reporting were also presented.

The studies show that many empirical research works had been done on credit card fraud detection using machine learning algorithms. However, empirical studies using real-time detection and reporting showed lower accuracy, lower f1 scores and precision. Also, previous studies are also very scarce on real time credit card fraud detection and reporting which is evident in the few empirical studies presented. Therefore, this work tends to develop a real time credit card fraud detection and notification prototype model using machine learning algorithms for the purpose of achieving a better prediction and reporting performance using three algorithms (Logistic Regression, the Random Forest model and the Decision Tree Classifier model).

Endnotes

1. R Shakya, Ronish, "*Application of Machine Learning Techniques in Credit Card Fraud Detection*" (2018). UNLV Theses, Dissertations, Professional Papers, and Capstones. 3454. http://dx.doi.org/10.34917/14279175)

2. https://www.credit-connect.co.uk/news/consumer-lending/fraud/over-1-2bn-lost-to-fraud-in-2022/

3. D.N Anowu, T Nyor, S.E Agbi, A.I Nelson, A.N Saliu. *Financial Forensic Analysis And Fraud Deterrence In Listed Deposit Money Banks In Nigeria*. **Gusau Journal of Accounting and Finance.** 2021 Oct 1;2(4):18

4. https://techpoint.africa/2021/02/22/nigeria-lost-5b-fraud-2020/

5. NIBSS Insight, "Fraud in Nigerian Financial Services" 2021 https://nibss-plc.com.ng/media/PDFs/post/NIBSS%20Insights%20Fraud.pdf

6. O.S Yee, S Sagadevan, N.H Malim. *Credit card fraud detection using machine learning as data mining technique*. **Journal of Telecommunication, Electronic and Computer Engineering** (JTEC). 2018 Jan 29;10(1-4):23-7.

7. A Thennakoon, C Bhagyani, S Premadasa, S Mihiranga and N Kuruwitaarachchi. *Real-time credit card fraud detection using machine learning*. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 488-493). IEEE. , 201

8. G.K Singh, A Bhayye, S Dhamnaskar, S Patil and S.V Phulari. *Credit card fraud detection using isolation forest*. **International Journal of Recent Advances in Multidisciplinary Topics**, *2*(6), pp.118-119.2021.

9. K Gupta, K Singh, G.V Singh, M Hassan, U Sharma. *Machine Learning based Credit Card Fraud Detection-A Review*. In2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC) 2022 May 9 (pp. 362-368). IEEE.

10. C Soulé-Dupuy, E Gaussier, M Lux, G Gianini, S Calabretto, M Granitzer, P.E Portier. *Credit Card Fraud Detection using Machine Learning with Integration of Contextual Knowledge* (Doctoral dissertation, INSA Lyon).2019

11. Y Lucas. *Credit card fraud detection using machine learning with integration of contextual knowledge* (Doctoral dissertation, Université de Lyon; Universität Passau (Deutscheland)).2019

12. B Baesens, S Höppner, I Ortner, T Verdonck. *robROSE: A robust approach for dealing with imbalanced data in fraud detection*. Statistical Methods & Applications. 2021 Sep;30(3):841-61.

13. S.M Mathews. *Explainable artificial intelligence applications in NLP, biomedical, and malware classification: a literature review*. InIntelligent Computing: Proceedings of the 2019 Computing Conference, Volume 2 2019 (pp. 1269-1292). Springer International Publishing.

14. N Ahmed, R Amin, H Aldabbas, D Koundal, B Alouffi, T Shah. *Machine learning techniques for spam detection in email and IoT platforms: analysis and research challenges. Security and Communication Networks.* 2022 Feb 3;2022:1-9.

15. G.K Zewdie, C Valladares, M.B Cohen, D.J Lary, D Ramani, D.M Tsidu. *Data-driven forecasting of low-latitude ionospheric total electron content using the random forest and LSTM machine learning methods*. Space Weather. 2021 Jun;19(6):e2020SW002639.

16. R.K Dhanaraj, K Rajkumar, U Hariharan. *Enterprise IoT modeling: supervised, unsupervised, and reinforcement learning*. Business Intelligence for Enterprise Internet of Things. 2020:55-79.

17. P.C Sen, M Hajra, M Ghosh. *Supervised classification algorithms in machine learning: A survey and review*. InEmerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018 2020 (pp. 99-111). Springer Singapore.

18. S Mohamed, R Ashraf, A Ghanem, M Sakr, R Mohamed. *Supervised Machine Learning Techniques: A Comparison*.2022

19. P.C Sen, M Hajra, M Ghosh. *Supervised classification algorithms in machine learning: A survey and review.* InEmerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018 2020 (pp. 99-111). Springer Singapore.

20. J.A Sidey-Gibbons, C.J Sidey-Gibbons. *Machine learning in medicine: a practical introduction*. **BMC medical research methodology**. 2019 Dec;19:1-8.

21. I.H Sarker. *Machine learning: Algorithms, real-world applications and research directions*. SN computer science. 2021 May;2(3):160.

22. Y Bengio, A Lodi, A Prouvost. *Machine learning for combinatorial optimization: a methodological tour d'horizon*. **European Journal of Operational Research**. 2021 Apr 16;290(2):405-21.

23. T Bokaba, W Doorsamy, B.S Paul. *Comparative study of machine learning classifiers for modelling road traffic accidents*. Applied Sciences. 2022 Jan;12(2):828

24. A.A Nababan, M Khairi, B.S Harahap. *Implementation of K-Nearest Neighbors (KNN) Algorithm in Classification of Data Water Quality*. **Jurnal Mantik**. 2022 Mar 20;6(1):30-5.

25. S Uddin, I Haque, H Lu, M.A Moni, E Gide. *Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction*. Scientific Reports. 2022 Apr 15;12(1):1-1.

26. F Chern, B Hechtman, A Davis, R Guo, D Majnemer, S Kumar. *Tpu-knn: K nearest neighbor search at peak flop/s*. arXiv preprint arXiv:2206.14286. 2022 Jun 28.

27. U.S Shanthamallu, Spanias *A. Supervised Learning. InMachine and Deep Learning Algorithms and Applications* 2022 (pp. 9-21). Cham: Springer International Publishing.

28. S Ghosh, A Dasgupta, A Swetapadma. *A study on support vector machine based linear and non-linear pattern classification*. In2019 International Conference on Intelligent Sustainable Systems (ICISS) 2019 Feb 21 (pp. 24-28). IEEE.

29. O.M Ali, S.W Kareem, A.S Mohammed. *Evaluation of Electrocardiogram Signals Classification Using CNN, SVM, and LSTM Algorithm: A review*. In2022 8th International Engineering Conference on Sustainable Technology and Development (IEC) 2022 Feb 23 (pp. 185-191). IEEE.

30. S Mohamed, R Ashraf, A Ghanem, M Sakr, R Mohamed. *Supervised Machine Learning Techniques: A Comparison*.2022

31. J.A Andeta. *Road-traffic accident prediction model : Predicting the Number of Casualties [Internet] [Dissertation]. 2021*. Available from: http://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-20146

32. S Ghosh, A Dasgupta, A Swetapadma. *A study on support vector machine based linear and non-linear pattern classification*. In2019 International Conference on Intelligent Sustainable Systems (ICISS) 2019 Feb 21 (pp. 24-28). IEEE

33. I Ahmad, M Basheri, M.J Iqbal, A Rahim. *Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection*. IEEE access. 2018 May 30;6:33789-95

34. X Zhai, M Chen, W Lu. *Fuel ratio optimization of blast furnace based on data mining. ISIJ International*. 2020 Nov 15;60(11):2471-6.

35. M Sabzekar, S.M Hasheminejad. *Robust regression using support vector regressions. Chaos, Solitons & Fractals*. 2021 Mar 1;144:110738.

36. M.S Adnan, S Zaidi, P Bhargava. *A novel support vector regression (SVR) model for the prediction of splice strength of the unconfined beam specimens. Construction and building materials.* 2020 Jul 10;248:118475

37. Y Pristyanto, N.A Setiawan, I Ardiyanto. *Hybrid resampling to handle imbalanced class on classification of student performance in classroom.*

In2017 1st International Conference on Informatics and Computational Sciences (ICICoS) 2017 Nov 15 (pp. 207-212). IEEE.

38. N.A AL-SERW. Undersampling and oversampling: An old and a new approach. Analytics Vidhya. 2021.

39. S Mayabadi, H Saadatfar. *Two density-based sampling approaches for imbalanced and overlapping data*. Knowledge-Based Systems. 2022 Apr 6;241:108217.

40. C Arun, C Lakshmi. *Genetic algorithm-based oversampling approach to prune the class imbalance issue in software defect prediction*. Soft Computing. 2022 Dec;26(23):12915-31

41. D Elreedy, A.F Atiya, F Kamalov. *A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning*. Machine Learning. 2023 Jan 5:1-21.

42. A Arafa, N El-Fishawy, M Badawy, M Radad. *RN-SMOTE: Reduced noise smote based on DBSCAN for enhancing imbalanced data classification*. **Journal of King Saud University-Computer and Information Sciences**. 2022 Sep 1;34(8):5059-74.https://doi.org/10.1016/j.jksuci.2022.06.005

43. A Cano, B Krawczyk. *ROSE: Robust online self-adjusting ensemble for continual learning on imbalanced drifting data streams*. Machine Learning. 2022 Jul;111(7):2561-99

44. G Kunapuli. *Ensemble Methods for Machine Learning*. Simon and Schuster; 2023 May 2

45. B Seijo-Pardo, V Bolón-Canedo, A Alonso-Betanzos. *Testing different ensemble configurations for feature selection. Neural Processing Letters*. 2017 Dec;46:857-80.DOI:10.1007/s11063-017-9619-1

46. S.P McPherron, W Archer, E.R Otárola-Castillo, M.G Torquato, T.L Keevil. *Machine learning, bootstrapping, null models, and why we are still not 100% sure which bone surface modifications were made by crocodiles*. **Journal of Human Evolution**. 2022 Mar 1;164:103071.

47. G Ngo, R Beard, R Chandra. *Evolutionary bagging for ensemble learning*. Neurocomputing. 2022 Oct 21;510:1-4.

48. M.L Kadali, V.S Ramakrishna, V.S Chandra Mouli, G.J Rajasekhar. *Prediction of Cardiovascular Disease using Machine Learning Algorithms with Relief and Lasso Feature Selection Techniques*. Mathematical Statistician and Engineering Applications. 2022 Oct 18;71(4):5356-72

49. A Aldrees, H.H Awan, M.F Javed, A.M Mohamed. *Prediction of water quality indexes with ensemble learners: Bagging and Boosting*. Process Safety and Environmental Protection. 2022 Dec 1;168:344-61

50. S Demir, E.K Sahin. *An investigation of feature selection methods for soil liquefaction prediction based on tree-based ensemble algorithms using AdaBoost, gradient boosting, and XGBoost*. Neural Computing and Applications. 2023 Feb;35(4):3173-90

51. https://www.analyticsvidhya.com/blog/2017/03/imbalanced-data-classification/

52. A.V Konstantinov, L.V Utkin. *Interpretable machine learning with an ensemble of gradient boosting machine*s. Knowledge-Based Systems. 2021 Jun 21;222:106993

53. E Suganya, CRajan. *An adaboost-modified classifier using particle swarm optimization and stochastic diffusion search in wireless IoT networks*. Wireless Networks. 2021 May;27:2287-99

54. R Sibindi, R.W Mwangi, A.G Waititu. *A boosting ensemble learning based hybrid light gradient boosting machine and extreme gradient boosting model for predicting house prices*. Engineering Reports. 2022:e12599

55. X Yang, Y Wang, R Byrne, G Schneider, S Yang. *Concepts of artificial intelligence for computer-assisted drug discovery*. Chemical reviews. 2019 Jul 11;119(18):10520-94

56. C Zhang, X Lei, L Liu. *Predicting metabolite–disease associations based on LightGBM model*. Frontiers in Genetics. 2021 Apr 13;12:660275

57. A Tiwari. *Supervised learning: From theory to applications*. InArtificial Intelligence and Machine Learning for EDGE Computing 2022 Jan 1 (pp. 23-32). Academic Press

58. M Anand, A Velu, P Whig. *Prediction of loan behaviour with machine learning models for secure banking*. **Journal of Computer Science and Engineering (JCSE)**. 2022 Feb 15;3(1):1-3

59. D Hussain, I Hussain, M Ismail, A Alabrah, S.S Ullah, H.M Alaghbari. *A simple and efficient deep learning-based framework for automatic fruit recognition*. Computational Intelligence and Neuroscience. 2022 Feb 21;2022.

60. D.J Hand, P Christen, N.F Kirielle: a*n interpretable transformation of the F-measure*. Machine Learning. 2021 Mar;110(3):451-6

61. I.I.I Muschelli *J. ROC and AUC with a binary predictor: a potentially misleading metric*. **Journal of classification**. 2020 Oct;37(3):696-708

62. K Tretiak, G Schollmeyer, S Ferson. *Neural network model for imprecise regression with interval dependent variables*. Neural Networks. 2023 Apr 1;161:550-64.

63. J.S Jone, S Kipsy. *Early Prediction Of Heart Diseases Using Logistic Regression Algorithm.***EPRA International Journal of Multidisciplinary Research (IJMR)**. 2023 Mar 9;9(3):72-82.

64. M Sri, Vaddeboyina, Redrowthu, R.A VijayaVasavi. *Performance Comparison of Machine Learning Approaches on Intrusion Detection Dataset*. 782-788. 2021:10.1109/ICICV50876.2021.9388502.

65. M Schonlau, R.Y Zou. *The random forest algorithm for statistical learning*. **The Stata Journal**. 2020 Mar;20(1):3-29.

66. M Ibrahim. *Evolution of Random Forest from Decision Tree and Bagging: A Bias-Variance Perspective*. **Dhaka University Journal of Applied Science and Engineering**. 2022;7(1):66-71.

67. A.B Shaik, S Srinivasan. *A brief survey on random forest ensembles in classification model*. InInternational Conference on Innovative Computing and Communications: Proceedings of ICICC 2018, Volume 2 2019 (pp. 253-260). Springer Singapore.

68. I Reis, D Baron, S Shahaf. *Probabilistic random forest: A machine learning algorithm for noisy data sets*. **The Astronomical Journal.** 2018 Dec 20;157(1):16.

69. B.O Yigin, O Algin, G Saygili. *Comparison of morphometric parameters in prediction of hydrocephalus using random forests*. Computers in Biology and Medicine. 2020 Jan 1;116:103547.

70. N.M Abdulkareem, A.M Abdulazeez. *Machine learning classification based on Radom Forest Algorithm: A review*. **International Journal of Science and Business**. 2021;5(2):128-42.

71. A Arista. *Comparison Decision Tree and Logistic Regression Machine Learning Classification Algorithms to determine Covid-19*. Sinkron. 7. 59-65. 2022:10.33395/sinkron.v7i1.11243.

72. D Denisko, M.M Hoffman. *Classification and interaction in random forests. Proceedings of the National Academy of Sciences*. 2018 Feb 20;115(8):1690-2

73. L Demidova, M Ivkina. *Defining the Ranges Boundaries of the Optimal Parameters Values for the Random Forest Classifier.* In2019 1st International Conference on Control Systems, Mathematical Modelling, Automation and Energy Efficiency (SUMMA) 2019 Nov 20 (pp. 518-522). IEEE.

74. M.A Sulaiman. *Evaluating data mining classification methods performance in Internet of things applications*. **Journal of Soft Computing and Data Mining**. 2020 Dec 6;1(2):11-25.

75. M.L Kolhe, S Tiwari, M.C Trivedi & K.K Mishra (Eds.). *Advances in Data and Information Sciences: Proceedings of ICDIS 2019* (Vol. 94). Springer Singapore. https://doi.org/10.1007/978- 981-15-0694-9

76. K Gajowniczek, I Grzegorczyk, T Ząbkowski, C Bajaj. *Weighted random forests to improve arrhythmia classification*. Electronics. 2020 Jan 3;9(1):99

77. B Charbuty, A Abdulazeez. *Classification based on decision tree algorithm for machine learning*. **Journal of Applied Science and Technology Trends**. 2021 Mar 24;2(01):20-8.

78. A Paul, D.P Mukherjee, P Das, A Gangopadhyay, A.R Chintha, S Kundu. *Improved random forest for classification. IEEE Transactions on Image Processing*. 2018 May 10;27(8):4012-24.

79. A Roy, J Sun, R Mahoney, L Alonzi, A Adams, P Beling. *Deep learning detecting fraud in credit card transactions*. In2018 Systems and Information Engineering Design Symposium (SIEDS) 2018 Apr 27 (pp. 129-134). IEEE.

80. R Sailusha, V Gnaneswar, R Ramesh, G.R Rao. *Credit card fraud detection using machine learning*. In2020 4th international conference on intelligent computing and control systems (ICICCS) 2020 May 13 (pp. 1264-1270). IEEE.

81. A Mehbodniya, I Alam, S Pande, R Neware, K.P Rane, M Shabaz, M.V Madhavan. *Financial fraud detection in healthcare using machine learning and deep learning techniques. Security and Communication Networks*. 2021 Sep 9;2021:1-8.

82. B Mytnyk, O Tkachyk, N Shakhovska, S Fedushko, Y Syerov. *Application of Artificial Intelligence for Fraudulent Banking Operations Recognition*. Big Data and Cognitive Computing. 2023 May 10;7(2):93.

83. H Guan, S Li, Q Wang, O Lyulyov, T Pimonenko. *Financial Fraud Identification of the Companies Based on the Logistic Regression Model*. **Journal of Competitiveness**. 2022 Dec 1(4).

84. E Ileberi, Y Sun, Z Wang. *Performance evaluation of machine learning methods for credit card fraud detection using SMOTE and AdaBoost*. IEEE Access. 2021 Dec 15;9:165286-94.

85. D Jovanovic, M Antonijevic, M Stankovic, M Zivkovic, M Tanaskovic, N Bacanin. *Tuning machine learning models using a group search firefly algorithm for credit card fraud detection*. Mathematics. 2022 Jun 29;10(13):2272.

86. J Raval, P Bhattacharya, N.K Jadav, S Tanwar, G Sharma, P.N Bokoro, M Elmorsy, A Tolba, M.S Raboaca. *RaKShA: A Trusted Explainable LSTM Model to Classify Fraud Patterns on Credit Card Transactions*. Mathematics. 2023 Apr 17;11(8):1901.

87. Z Zhao, T Bai. *Financial Fraud Detection and Prediction in Listed Companies Using SMOTE and Machine Learning Algorithms*. Entropy. 2022 Aug 19;24(8):1157.

88. M Djuric, JL ovanovic, M Zivkovic, N Bacanin, M Antonijevic, M Sarac. *The AdaBoost Approach Tuned by SNS Metaheuristics for Fraud Detection*. InProceedings of the International Conference on Paradigms of Computing, Communication and Data Sciences: PCCDS 2022 2023 Feb 24 (pp. 115-128). Singapore: Springer Nature Singapore.

89. S.R Krishna, V Agarwal, D.E Rao, V.U Kakde, S Kumari, P.S Vadar. *Machine Learning based Data Mining for Detection of Credit Card Frauds*. In2023 International Conference on Inventive Computation Technologies (ICICT) 2023 Apr 26 (pp. 72-77). IEEE.

90. Y.K Saheed, M.A Hambali, M.O Arowolo, Y.A Olasupo. *Application of GA feature selection on Naive Bayes, random forest and SVM for credit card fraud detection*. In2020 international conference on decision aid sciences and application (DASA) 2020 Nov 8 (pp. 1091-1097). IEEE.

91. M.B Islam, C Avornu, P.K Shukla, P.K Shukla. *Cost Reduce: Credit Card Fraud Identification Using Machine Learning*. In2022 7th International Conference on Communication and Electronics Systems (ICCES) 2022 Jun 22 (pp. 1192-1198). IEEE.

92. M.S Baig, K Jaisharma. *Comparison of Novel Optimized Random Forest Technique and Gradient Boosting for Credit Card Fraud Detection with Improved Precision*. **Journal of Pharmaceutical Negative Results**. 2022 Sep 27:851-6.

93. M.S Baig, K Jaisharma. *Comparison of Novel Optimized Random Forest Technique and Logistic Regression for Credit Card Fraud Detection with Improved Precision*. **Journal of Pharmaceutical Negative Results**. 2022 Sep 27:723-7.

94. M AlEmad. "*Credit Card Fraud Detection Using Machine Learning". Thesis. 2022*. Rochester Institute of Technology. Accessed from https://scholarworks.rit.edu/theses/11318

95. S Bhatia, B.B Naib, G Ashraf. *Credit Card Fraud Detection using Classification Algorithm*. TechRxiv. Preprint.2023: https://doi.org/10.36227/techrxiv.23377547.v1

96. C Sudha, D Akila. *Credit card fraud detection system based on operational & transaction features using svm and random forest classifiers*. In2021 2nd International Conference on Computation, Automation and Knowledge Management (ICCAKM) 2021 Jan 19 (pp. 133-138). IEEE.: **DOI:** 10.1109/ICCAKM50778.2021.9357709

97. A Cherif, A Badhib, H Ammar, S Alshehri, M Kalkatawi, A Imine. *Credit card fraud detection in the era of disruptive technologies: A systematic review*. **Journal of King Saud University-Computer and Information Sciences**. 2022 Dec 5.https://doi.org/10.1016/j.jksuci.2022.11.008

98. I.D Mienye, Y Sun. *A Deep Learning Ensemble With Data Resampling for Credit Card Fraud Detection*. IEEE Access. 2023 Mar 27;11:30628-38.**DOI:** 10.1109/ACCESS.2023.3262020

99. A.Q Abdulghani, O.N Uçan, K.M Alheeti. *Credit card fraud detection using XGBoost algorithm*. In2021 14th International Conference on Developments in eSystems Engineering (DeSE) 2021 Dec 7 (pp. 487-492). IEEE.**DOI:** 10.1109/DeSE54285.2021.9719580

100. K Lavanya. *A Comparison of Logistic Regression Classifier and Random Forest Classifier for the Accurate Classification of Credit Card Fraudulent Transactions*. **Journal of Survey in Fisheries Sciences**. 2023 Mar 8;10(1S):2008-17.DOI: https://doi.org/10.17762/sfs.v10i1S.435

101. A Abdulghani. *Employing machine learning techniques and fuzzy membership for detecting fraud transactions in credit card*. 2022 (Yayınlanmamış yüksek lisans tezi). Altınbaş Üniversitesi, Lisansüstü Eğitim Enstitüsü, İstanbul.

102. M.V Krishna, J Praveenchandar. *Comparative Analysis of Credit Card Fraud Detection using Logistic regression with Random Forest towards an Increase in Accuracy of Prediction*. In2022 International Conference on Edge Computing and Applications (ICECAA) 2022 Oct 13 (pp. 1097-1101). IEEE.**DOI:** 10.1109/ICECAA55415.2022.9936488

103. H Du, L Lv, A Guo, H Wang. *AutoEncoder and LightGBM for Credit Card Fraud Detection Problems*. Symmetry. 2023 Apr 6;15(4):870.

104. H Du H, L Lv, A Guo, H Wang. AutoEncoder and LightGBM for Credit Card Fraud Detection Problems. Symmetry. 2023 Apr 6;15(4):870.https://doi.org/10.3390/sym15040870

105. A Alshutayri. *Fraud Prediction in Movie Theater Credit Card Transactions using Machine Learning*. **Engineering, Technology & Applied Science Research.** 2023 Jun 2;13(3):10941-5. https://doi.org/10.48084/etasr.5950

106. S Shellyann, H Patrick. *Framework for Credit Card Fraud Detection Using Benefit-Based Learning and Periodic Features*, 14 March 2023, PREPRINT (Version 1) available at Research Square [https://doi.org/10.21203/rs.3.rs-2652853/v1]

107. R.O Ogundokun, S Misra, O.E Ogundokun, J Oluranti, R Maskeliunas. *Machine learning classification based techniques for fraud discovery in credit card datasets*. InApplied Informatics: Fourth International Conference, ICAI

2021, Buenos Aires, Argentina, October 28–30, 2021, Proceedings 4 2021 (pp. 26-38). Springer International Publishing.

108. R.O Ogundokun, S Misra, O.J Fatigun, J.K Adeniyi. *Naïve Bayes Based Classifier for Credit Card Fraud Discovery*. InEuropean, Mediterranean, and Middle Eastern Conference on Information Systems 2021 Dec 8 (pp. 515-526). Cham: Springer International Publishing.

109. G.A Farhang, T Mansouri, M.R Sadeghi Moghaddam, N Bahrambeik, R Yavari, M Fani Sani. *Credit card fraud detection using asexual reproduction optimization*. Kybernetes. 2022 Sep 5;51(9):2852-76.

110. A.W Al-Faqeh, A Zerguine, M.A Al-Bulayhi, A.H Al-Sleem, A.S Al-Rabiah. *Credit card fraud detection via integrated account and transaction submodules*. **Arabian Journal for Science and Engineering**. 2021 Oct;46(10):10023-31.

111. O Sinayobye, R Musabe, A Uwitonze, A Ngenzi. *A Credit Card Fraud Detection Model Using Machine Learning Methods with a Hybrid of Undersampling and Oversampling for Handling Imbalanced Datasets for High Scores*. InInternational Conference on Applied Machine Learning and Data Analytics 2022 Dec 22 (pp. 142-155). Cham: Springer Nature Switzerland.

112. A Mahajan, V.S Baghel, R Jayaraman. *Credit Card Fraud Detection using Logistic Regression with Imbalanced Dataset*. In2023 10th International Conference on Computing for Sustainable Global Development (INDIACom) 2023 Mar 15 (pp. 339-342). IEEE..

113. N.S Pranavi, T.K Sruthi, B.J Sirisha, M.S Nayak, V.S Thadikemalla. *Credit Card Fraud Detection Using Minority Oversampling and Random Forest Technique*. In2022 3rd International Conference for Emerging Technology (INCET) 2022 May 27 (pp. 1-6). IEEE.

114. H Aktar, M.A Masud, N.J Aunto, S.N Sakib. *Classification Using Random Forest on Imbalanced Credit Card Transaction Data*. In2021 3rd International Conference on Sustainable Technologies for Industry 4.0 (STI) 2021 Dec 18 (pp. 1-4). IEEE.

115. P Gupta, A Varshney, M.R Khan, R Ahmed, M Shuaib, S Alam. *Unbalanced Credit Card Fraud Detection Data: A Machine Learning-Oriented Comparative Study of Balancing Techniques*. Procedia Computer Science. 2023 Jan 1;218:2575-84.

116. C.L Udeze, I.E Eteng, A.E Ibor. *Application of Machine Learning and Resampling Techniques to Credit Card Fraud Detection*. **Journal of the Nigerian Society of Physical Science**s. 2022 Aug 15:769-.

117. A.M Aburbeian, H.I Ashqar. *Credit Card Fraud Detection Using Enhanced Random Forest Classifier for Imbalanced Data*. InInternational

Conference on Advances in Computing Research 2023 May 8 (pp. 605-616). Cham: Springer Nature Switzerland.

118. Y Vivek, V Ravi, A.A Mane, L.R Naidu. *ATM Fraud Detection using Streaming Data Analytics*. arXiv preprint arXiv:2303.04946. 2023 Mar 8.

119. Q Meng. *Credit Card Fraud Detection Using Feature Fusion-based Machine Learning Model*. Highlights in Science, Engineering and Technology. 2022 Dec 3;23:111-6.

120. I Odeajo, O Akinmoluwa, S Ojo, T.D Otesanya. "F*inancial Fraud Detection using Machine Learning : Credit Card Fraud,*" **International Journal of Recent Engineering Science**, vol. 10, no. 3, pp. 23-32, 2023. Crossref, https://doi.org/10.14445/23497157/IJRES-V10I3P104

121. R.B Asha, S,K KR. C*redit card fraud detection using artificial neural network*. Global Transitions Proceedings. 2021 Jun 1;2(1):35-41.

122. H Najadat, O Altiti, A.A Aqouleh, M Younes. *Credit card fraud detection based on machine and deep learning*. In2020 11th International Conference on Information and Communication Systems (ICICS) 2020 Apr 7 (pp. 204-208). IEEE.

123. F.K Alarfaj, I Malik, H.U Khan, N Almusalla, M Ramzan, M Ahmed. *Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms*. IEEE Access. 2022 Apr 12;10:39700-15.

124. I.D Mienye, Y Sun. *A Machine Learning Method with Hybrid Feature Selection for Improved Credit Card Fraud Detection*. Applied Sciences. 2023 Jun 18;13(12):7254.

125. B Fakiha. *Forensic Credit Card Fraud Detection Using Deep Neural Network*. **Journal of Southwest Jiaotong University**. 2023;58(1).

126. T Berhane, T Melese, A Walelign, A Mohammed. *A Hybrid Convolutional Neural Network and Support Vector Machine-Based Credit Card Fraud Detection Model*. Mathematical Problems in Engineering. 2023 Jun 3;2023.

127. S Mugundhan, P Venkataramanan. *Data Characteristic Stability Based Random Forest Implementation of Credit Card Fraud Detection*. In2022 5th International Conference on Contemporary Computing and Informatics (IC3I) 2022 Dec 14 (pp. 1100-1104). IEEE.

128. P Chatterjee, D Das, D Rawat. *Securing Financial Transactions: Exploring the Role of Federated Learning and Blockchain in Credit Card Fraud Detection*.

129. M.A Alamri, M.A Ykhlef. *A Machine Learning-Based Framework for Detecting Credit Card Anomalies and Fraud*. In2023 27th International Conference on Information Technology (IT) 2023 Feb 15 (pp. 1-7). IEEE.

130. Z Zhang, S Huang. *Credit card fraud detection via deep learning method using data balance tools*. In2020 international conference on computer science and management technology (ICCSMT) 2020 Nov 20 (pp. 133-137). IEEE.

131. Y Yang, C Liu, N Liu. *Credit card fraud detection based on CSat-related AdaBoost*. InProceedings of the 2019 8th International Conference on Computing and Pattern Recognition 2019 Oct 23 (pp. 420-425).

132. K.V Mahalaxmi, K.S Rekha. *Accurate Credit Card Fraudulent Dataset using Logistics Regression Compared with Random Forest.* In2022 International Conference on Business Analytics for Technology and Security (ICBATS) 2022 Feb 16 (pp. 1-5). IEEE.

133. R Reshma, R Santhosh, N Mekala. *An Analytical Approach to Fraudulent Credit Card Transaction Detection using Various Machine Learning Algorithms*. In2023 Second International Conference on Electronics and Renewable Systems (ICEARS) 2023 Mar 2 (pp. 1400-1404). IEEE.

134. M.K Nikhil, M.B Maharshi, M.K Tanooj, M.D SriRam. Credit Card Fraud Detection Using Machine Learning Algorithms. **Journal of Engineering Sciences**. 2023;14(04).

135. M.N Hossain, M.M Hassan, R.J Monir. *Analyzing the Classification Accuracy of Deep Learning and Machine Learning for Credit Card Fraud Detection*. **Asian Journal For Convergence In Technology (AJCT)** ISSN-2350-1146. 2022 Dec 31;8(3):31-6.

136. S.P Maniraj, S Aditya, A Shadab, S Swarna. *Credit Card Fraud Detection using Machine Learning and Data Science*. **International Journal of Engineering Research** and. 2019.08. 10.17577/IJERTV8IS090031.

137. N.D Vaishnavi, S Geetha. *Credit Card Fraud Detection using Machine Learning Algorithms*, Procedia Computer Science,Volume 165,2019,Pages 631-641,ISSN 1877-0509,https://doi.org/10.1016/j.procs.2020.01.057.

138. M Thirunavukkarasu, N Achutha, J Adusumilli. **International Journal of Computer Science and Mobile Computing**, Vol.10 Issue.4, April- 2021, pg. 71-79.DOI: 10.47760/ijcsmc.2021.v10i04.011

139. D Sayantan, U.R Zain, J Rehman, D Mangalesh, M Yash. Credit Card Fraud Detection System Using Machine Learning A Project Report, Bachelor Of Technology In Information Technology, MAY 2019

140. V Navaratna, P.A Reddy, P.S Avinash, T.A Jyothi. *Credit Card Fraud Detection Using Machine Learning.*Vol.11:Issue 6:June 2020:ISSN 0377-9254

141. A Thennakoon, C Bhagyani, S Premadasa, S Mihiranga, N Kuruwitaarachchi. *Real-time credit card fraud detection using machine learning*. In2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) 2019 Jan 10 (pp. 488-493). IEEE.

142. Y Abakarim, M Lahby, A Attioui. *An efficient real time model for credit card fraud detection based on deep learning*. InProceedings of the 12th international conference on intelligent systems: theories and applications 2018 Oct 24 (pp. 1-7).

143. A. Kumar, D. Prusti, I. S. Purusottam and S. K. Rath, "*Real time SOA based credit card fraud detection system using machine learning techniques,*" 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2021, pp. 1-6, doi: 10.1109/ICCCNT51525.2021.9579598.

Chapter Three

Methodology

3.1 Research Approach

The research work adopts a supervised learning approach, where a classification model is trained on a labelled dataset to predict the possibility of a customer being fraud or not. The study uses three classification algorithms: Logistics Regression, Random Forest and Decision Tree Classifier and also paid more attention in experimenting the most commonly used techniques for handling imbalance class such as SMOTE, Random Over Sampling, Random Under Sampling. The dataset is divided into training and testing sets, and cross-validation is employed to evaluate the performance of the models. This chapter provides an overview of the research design approach, data collection, preprocessing, model design, training, validation, and evaluation process and the deployment of sending text messages of fraud location and time to the customers about fraud alert. Additionally, the software and hardware requirements for the study are also discussed.

3.2 Requirement Specification

3.2.1 Hardware Requirement: The study was conducted on a personal computer with 16 GB of RAM and a 2.2 GHz Intel Core i7 processor.

3.2.2 Software Minimum Requirements: These are the computer programmes needed to put the the developed model into action. They include; Python programming language, Jupyter Notebook, and various libraries, including scikit-learn, Pandas, NumPy, Matplotlib, and Seaborn

3.3 System Design

A conceptual model of the proposed prediction model is shown in Figure 3.2.The system is designed such that, credit card comprehensive data will be collected from a simulated credit card transaction dataset containing legitimate and fraud transactions and preprocessed to normalize the data. Data sampling (Oversampling, undersampling and SMOTE) was done to cater for imbalanced data that may affect the performance of the model. Relevant features and variables will also be identified that may influence the credit card fraud detection rate through exploratory data analysis and domain expertise.

Machine learning models was built using 3 algorithms (Logistic Regression, Random Forest and Decision Tree) to detect credit card fraud, thus sending a real time notification of any fraudulent activity. The performance of the system will be evaluated and interpreted using the evaluation metrics (Accuracy, F1 Score, Recall and Precision).

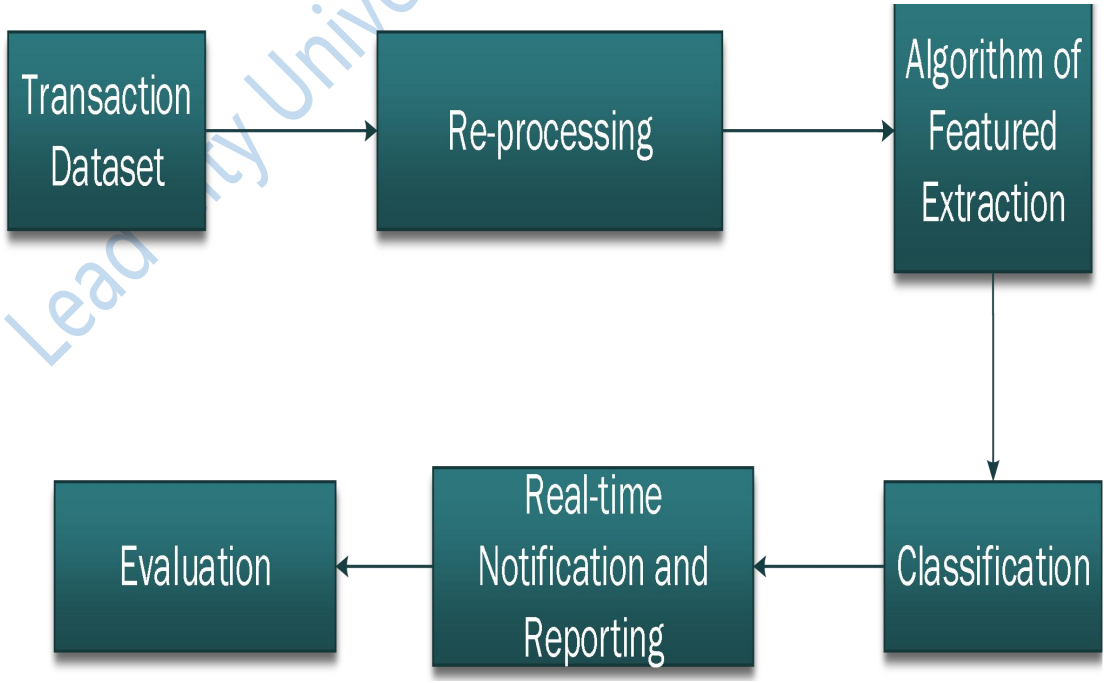


Figure 3.1: Conceptual Model of the Design

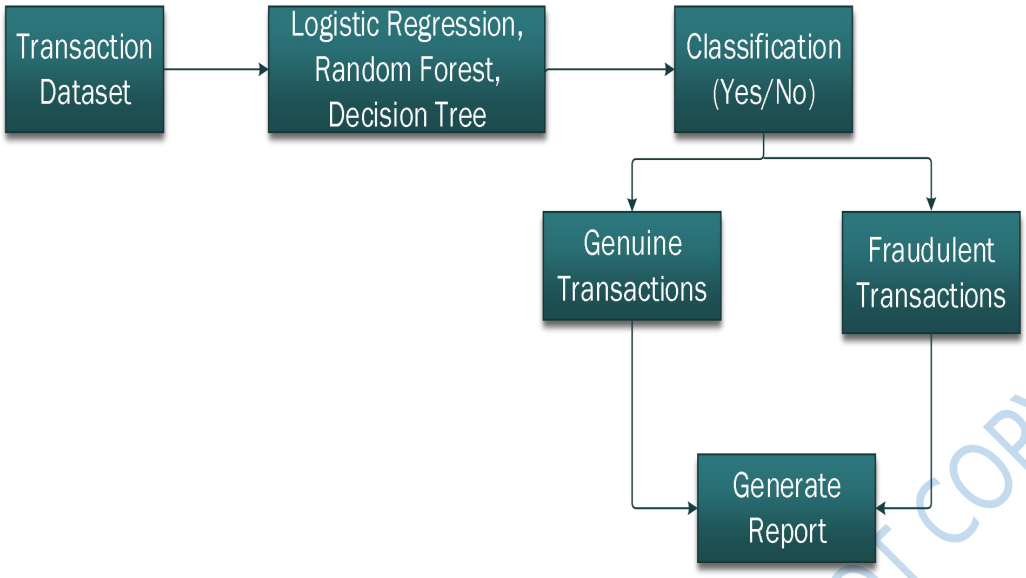


Figure 3.2: Design Framework(Researcher, JolaoshoA.O, 2023)

Transaction Dataset: This is an open source simulated credit card transaction datasetthat will be use used for analysis.

Logistic Regression, Random Forest and Decision Tree: These are the supervised machine learning algorithms that will be used for the classification of credit card fraud into genuine or fraudulent transaction.

Genuine/Fraudulent Transaction: At this stage, thedataset was classified into fraudulent or genuine transaction.

Genarate Report: This is the stage where the report is generated and a notification will be sent via a text message to notify about the transaction that occurred on a credit card

3.4 Research Method

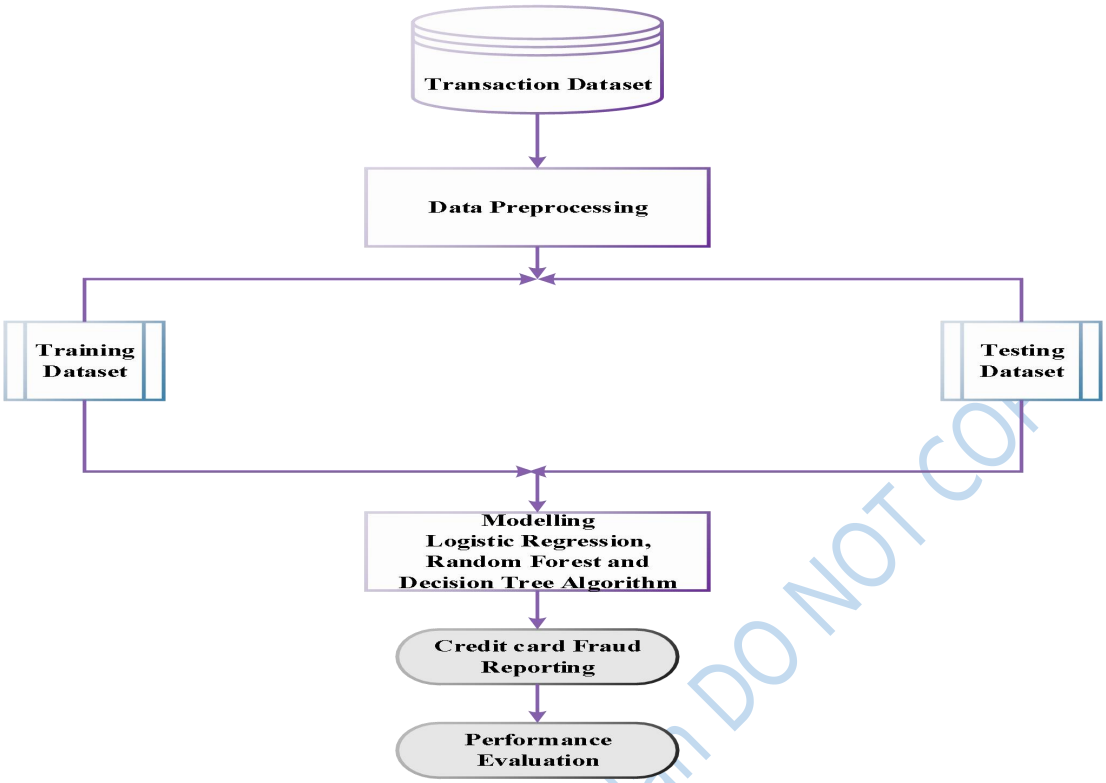


Figure 3.3: Flowchart of the Proposed Method(Researcher, JolaoshoA.O, 2023)

3.4.1 Data Collection

The dataset to be used in this study will be an open source simulated credit card transaction dataset. It encompasses credit card transactions of 1000 customers with a pool of 800 merchants. The dataset was generated using Sparkov Data Generation Harris[4]. This is a simulated credit card transaction dataset containing legitimate and fraud transactions from the duration 1st Jan 2019 - 31st Dec 2020. This was generated using Sparkov Data Generation via Github tool. The simulator has certain pre-defined list of merchants, customers and transaction categories. And then using a python library called "faker", and with the number of customers, merchants that is mentioned during simulation, an intermediate list is created.

After this, depending on the profile choosen for e.g. "adults 2550 female rural.json" (which means simulation properties of adult females in the age range of 25-50 who are from rural areas), the transactions will be created i.e "Sparkov | Github | adults_2550_female_rural.json", there are parameter value ranges defined in terms of min, max transactions per day, distribution of transactions across days of the week and normal distribution properties (mean, standard deviation) for amounts in various categories. Using these measures of distributions, the transactions will be generated using faker.

3.4.2 Dataset Description

trans_date_trans_time -> Transaction time stamp

cc_num -> Credit card number

merchant -> merchant name

category -> transaction category

amt -> Transaction amount

first -> First name of card holder

last -> Last name of card holder

gender -> Sex of card holder

street -> transaction address

city -> transaction city

state -> transaction state

zip -> transaction zipcode

lat -> transaction lattitude

long -> transaction longitude

city_pop -> Population of the city

job -> job of the card holder

dob -> date of birth of card holder

trans_num -> transaction number of transaction

unix_time -> time in unix format

merch_lat -> lattitude of the merchant

merch_long -> longitude of merchant

is_fraud -> nature of transaction (fraud or not fraud)

Here, the 'is_fraud' variables is our target variable

3.4.3Data Processing

Data Mining: This is toensure that the dataset is in a format that the machine learning algorithm can understand. In order to avoid over fitting or under fitting the model with data, in order to reduce the dimensionality of the dataset so that each stated parameter contributes at least equally to the proposed model's prediction. Sentence seems incomplete

Feature Engineering: feature engineering as the process of using domain knowledge of data to create features that make machine learning algorithm work. The features will be selected based on their relevance to predicting credit card fraud. Some features will be dropped due to their irrelevance or high correlation with other features.

3.4.4 Model Design, Training, and Validation.

Data splitting: The dataset was divided into two distinct subsets: a Training set of 70% and a Test set of 30% of the data. The training set was for resampling,

hyperparameter tuning, and training the model and the test set was used to test the performance of the trained model.

Data Resampling: This is use to handle the imbalanced dataset. If an imbalanced dataset is used, the model built tends to be biased towards the legitimate transactions, and hence, it results in the poor performance of model when tested in an unseen data. To tackle this problem, three resampling techniques was used such as random undersampling, random oversampling, SMOTE. The resampling technique was implemented on the training data separately to make it balanced.

Algorithm Models: The Logistics Regression, Random Forest and Decision Tree Classifier models wasdesigned and trained using the preprocessed dataset. Three (3) sampling techniques was used (over sampling, undersampling and SMOTE) in order to balance the dataset.

Logistic Regression: Forecasts the outcome of a categorical dependent variable. As a result, the end to be categorical or discrete value. Yes or no, can be 0 or 1, true or false, but instead of displaying exact values like 0 and 1, it displays probabilities in the range 0 to 1[4]. The algorithm can be developed using the equation

y = e^ (b0 + b1*x) / (1 + e^ (b0 + b1*x)). 3.1

The hypothesis of linear regression can be expressed as

y = a0 + a1 ∗ x 3.2

The performance index, which solely approximates the confidence interval (CI) of the RF model is given as

$$mg(x,y) = av_k I(h_k(x,\Theta_k) = y) - \max_{j \neq y} av_k I(h_k(x,\Theta_k) = j) \quad 3.3$$

where $I(.)$ denotes an indicator function, and $av(.)$, the average value. It is observed that as the margin increases, the confidence level also increases. The generalisation error becomes

$$PE^* = P_{x,y}(mg(x,y) < 0), \quad 3.4$$

where $P(.)$ denotes probability. With an increase in trees for all sequences Θ_k, PE^* converges to

$$P_{x,y}(P_\Theta(h(x,\Theta) = y) - \max_{j \neq y} P_\Theta(h(x,\Theta) = j < 0) \quad 3.5$$

Convergence of this generalisation error proves that the RF model does not overfit as more trees are introduced. The upper bound for the generalisation error is given as

$$PE^* \leq \frac{\bar{\rho}(1-s^2)}{s^2}\ , \quad 3.6$$

where $\bar{\rho}$ is the average correlation value, s is the strength of each tree in the model. An increased strength of individual trees and a low correlation between them produces more accurate prediction results.

The Decision Tree: If a target is a classification outcome taking on values 0,1,…,K-1, for node m,

Let

$$\mathbf{p_{mk}} = \frac{\mathbf{1}}{\mathbf{n_m}} \sum_{\mathbf{y \in Q_m}} \mathbf{I}\,(\mathbf{y = k}) \quad 3.7$$

be the proportion of class k observations in node m. If m is a terminal node, common measures of impurity are the following.

$$\mathbf{H(Q_m)} = \sum_{\mathbf{y \in Q_m}} \mathbf{p_{mk}(1 - p_{mk})} \quad 3.8$$

Random Forest is an ensemble learning method that constructs multiple decision trees and then combines their predictions to produce a final output.

Given an ensemble of classifiers h1(x),h2(x), ... ,hK (x), and with the training set drawn at random from the distribution of the random vector Y,X, define the margin function as

$$\mathbf{mg(X,Y) = av_k\, I(h_k\,(X)=Y) - max_{j \neq Y}\, av_k\, I(h_k\,(X)= j)} \qquad 3.9$$

where I(•) is the indicator function. The margin measures the extent to which the average number of votes at X,Y for the right class exceeds the average vote for any other class. The larger the margin, the more confidence in the classification. The generalization error is given by;

$$\mathbf{PE^* = P_{X,Y}\,(mg(X,Y) < 0)} \qquad 3.10$$

where the subscripts X,Y indicate that the probability is over the X,Y space.

3.4.5 Real Time Reporting: Twilio will be will be used in generating a real time text message if fraud is detected.Twilio is a web application programming interface (API) that can be used to add communications such as phone calling, messaging into Python applications[2].

3.5 Evaluation Metrics

Evaluation metrics is used for evaluating the performance of the model depending on the nature of the problem (whether it is a regression or classification). This thesis is limited to evaluation metrics related to the classification problem.

3.5.1 Confusion Matrix

It is the most commonly used evaluation metrics in predictive analysis mainly because it is very easy to understand and it can be used to compute other essential metrics such as accuracy, recall, precision, etc[6]. It is an NxN matrix that describes the overall

performance of a model when used on some dataset, where N is the number of class labels in the classification problem[7]. For binary classification, we have a 2x2 confusion matrix as shown in figure 3.4[7].

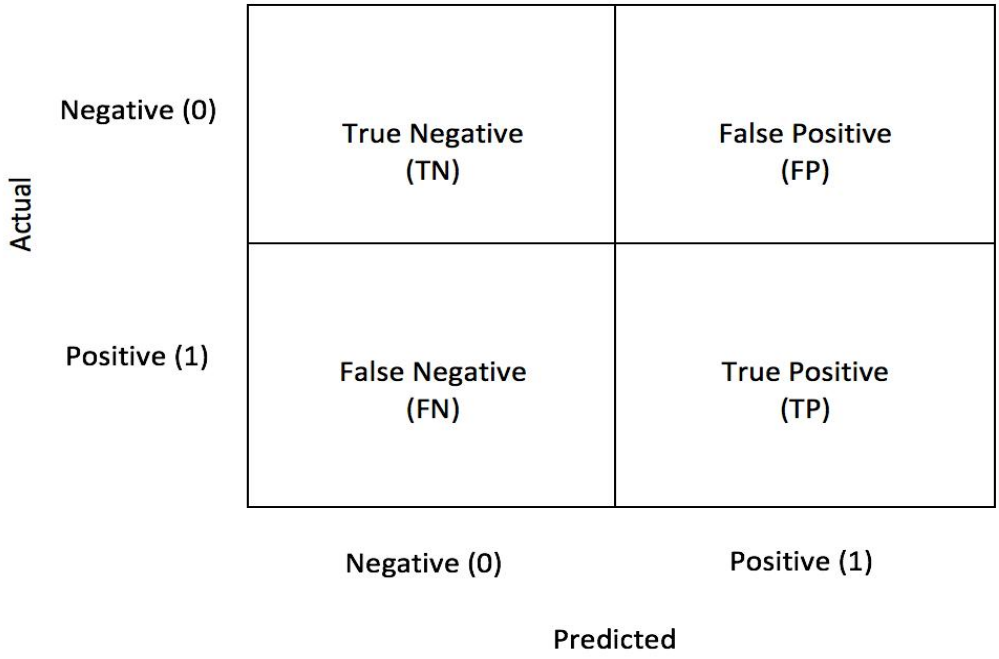


Figure 3.4: Confusion Matrix[8].

A confusion matrix is composed of statistics such as True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) which are calculated using the combination of actual and predicted values[8].

True Positive (TP) is a case where the actual value was positive (e.g., fraud) and the predicted value is also positive.

False Positive (FP) is a case where the actual value was negative (e.g., normal) but the predicted value is positive.

True Negative (TN) is a case where the actual value was negative (e.g., normal) and the predicted value is also negative.

False Negative (FN) is a case where the actual value was positive (e.g., fraud) but the predicted value is negative.

3.5.2 Recall

Recall, also known as sensitivity, is the fraction of true positives to the actual positive cases, which is shown in equation 3.12[9]. In simple terms, recall is how many of true positives were found (recalled) out of all the true positive cases.

$$\textbf{Recall} = \frac{TP}{TP+FN} \quad 3.12$$

3.5.3 Precision

As shown in equation 3.13, precision is the fraction of true positives over the true positives and false positives. In simple terms, precision is how many of the found cases were true positives.

$$\textbf{Precision} = \frac{TP}{TP+FP} \quad 3.13$$

3.5.4 F1 Score

F1 Score also called F score or F-measure is the harmonic mean of the recall and precision[10]. Its value ranges from 0 to 1, where 0 is considered worst, and 1 is considered best. It can be calculated as follows[6].

$$\textbf{F1} = \frac{2*(\text{Precision}*\text{Recall})}{(\text{Precision}+\text{Recall})} \quad 3.14$$

3.5.5 Area Under Receiver Operating Characteristic Curve

It is one of the most widely used evaluation metrics in predictive analysis. It tells us how good a model performs when used at different probability thresholds[6]. By default, a probability threshold of 0.5 is used for the classification problem. It is a plot between True positive rate (TPR), which is also called sensitivity and False Positive Rate (FPR). FPR is calculated as (1-Specificity)[6].

The equations of sensitivity and specificity are given in equations 3.15 and 3.16 respectively.

$$\textbf{Sensitivity} = \frac{TP}{TP+FN} \quad 3.15$$

$$\textbf{Specificity} = \frac{TN}{TN+FP} \qquad 3.16$$

Sensitivity and specificity are inversely related as we change the probability threshold, (i.e., when the threshold is decreased, sensitivity increases while specificity decreases and when we increase the threshold, sensitivity decreases while specificity increases). From the ROC curve, the area under the curve can be calculated which is the probability that a model will rank a randomly chosen positive instance higher than a randomly chosen negative one[11,12].

Endnotes

1. Z Shen, A Shehzad, S Chen, H Sun, J Liu. *Machine learning based approach on food recognition and nutrition estimation*. Procedia Computer Science. 2020 Jan 1;174:448-53

2. https://www.fullstackpython.com/twilio.html#:~:text=Twilio%20is%20a%20web%20application,authentication%20into%20their%20Python%20applications.

3. V Fleischhauer, A Feldheiser, S Zaunseder. *Beat-to-Beat Blood Pressure Estimation by Photoplethysmography and Its Interpretation*. Sensors. 2022 Sep 17;22(18):7037.

4. https://github.com/namebrandon/Sparkov_Data_Generation

5. G Kunapuli. *Ensemble Methods for Machine Learning*. Simon and Schuster; 2023 May 2

6. R Shakya, Ronish, "*Application of Machine Learning Techniques in Credit Card Fraud Detection*" (2018). UNLV Theses, Dissertations, Professional Papers, and Capstones. 3454. http://dx.doi.org/10.34917/14279175)

7. A Tiwari. *Supervised learning: From theory to applications*. InArtificial Intelligence and Machine Learning for EDGE Computing 2022 Jan 1 (pp. 23-32). Academic Press

8. M Anand, A Velu, P Whig. *Prediction of loan behaviour with machine learning models for secure banking*. **Journal of Computer Science and Engineering (JCSE)**. 2022 Feb 15;3(1):1-3

9. D Hussain, I Hussain, M Ismail, A Alabrah, S.S Ullah, H.M Alaghbari. *A simple and efficient deep learning-based framework for automatic fruit recognition*. Computational Intelligence and Neuroscience. 2022 Feb 21;2022

10. D.J Hand, P Christen, N.F Kirielle: a*n interpretable transformation of the F-measure*. Machine Learning. 2021 Mar;110(3):451-6

11. I.I Muschelli *J. ROC and AUC with a binary predictor: a potentially misleading metric*. **Journal of classification**. 2020 Oct;37(3):696-708

12. G Ngo, R Beard, R Chandra. *Evolutionary bagging for ensemble learning*. Neurocomputing. 2022 Oct 21;510:1-4.

Chapter Four

Results and Discussion of Finding

In this chapter, the results of the machine learning models for predicting credit card fraud was presented. Three models, Logistic Regression (LR), Random Forest (RF) and Decision Tree (DT) classifier, was used to classify the fraud into two categories: yes and no. Also, Twelve (12) different models was used for the prediction of credit card fraud. Additionally, and three sampling techniques was used in order to balance the dataset.The performance of the models was evaluated using precision, recall, and F1-score metrics. The evaluation offers a comprehensive guide for the selection of an ideal algorithm based on the nature of fraudulent activities.

4.1 Result on Data Collection and Description

An open-source data utilised in this study was sourced from kaggle.com (fraudTrain.csv dataset) which was used to train the Logistic regression, Random Forest and Decision tree classifier1. The dataset comprises simulated credit card transactions, encompassing both genuine and fraudulent transactions, spanning from the 1st of January 2019 to the 31st of December 2020. The dataset encompasses credit card usage data from a sample of 1000 customers engaging in transactions with a collective of 800 merchants with distribution of each variable "*trans_date_trans_time', 'cc_num', 'merchant', 'category', 'amt', 'first', 'last', 'gender', 'street', 'city', 'state', 'zip', 'lat', 'long', 'city_pop', 'job', 'dob', 'trans_num', 'unix_time', 'merch_lat', 'merch_long', 'is_fraud*.

	cc_num	merchant	category	amt	first	last	gender	street	city	state	zip	lat	long	city_pop	job	dob	trans_num	unix_time	me
2	2.291E+15	fraud_Kirlin	personal_c	2.86	Jeff	Elliott	M	351 Darlene	Columbia	SC	29209	33.9659	-80.9355	333497	Mechanical	3/19/1968	2da90c7d74	1.372E+09	33
3	3.573E+15	fraud_Spor	personal_c	29.84	Joanne	Williams	F	3638 Marsh	Altonah	UT	84002	40.3207	-110.436	302	Sales profe	1/17/1990	324cc20440	1.372E+09	39
4	3.598E+15	fraud_Swar	health_fitn	41.28	Ashley	Lopez	F	9333 Valen	Bellmore	NY	11710	40.6729	-73.5365	34496	Librarian, p	10/21/1970	c81755dbbl	1.372E+09	4
5	3.592E+15	fraud_Hale	misc_pos	60.05	Brian	Williams	M	32941 Kryst	Titusville	FL	32780	28.5697	-80.8191	54767	Set designe	7/25/1987	2159175b9e	1.372E+09	28
6	3.527E+15	fraud_John	travel	3.19	Nathan	Massey	M	5783 Evan F	Falmouth	MI	49632	44.2529	-85.017	1126	Furniture d	7/6/1955	57ff021bd3	1.372E+09	44
7	3.041E+13	fraud_Daug	kids_pets	19.55	Danielle	Evans	F	76752 Davic	Breesport	NY	14816	42.1939	-76.7361	520	Psychother	10/13/1991	798db04aa	1.372E+09	41
8	2.132E+14	fraud_Rom	health_fitn	133.93	Kayla	Sutton	F	010 Weave	Carlotta	CA	95528	40.507	-123.9743	1139	Therapist, c	1/15/1951	17003d7ce5	1.372E+09	41
9	3.589E+15	fraud_Reic	personal_c	10.37	Paula	Estrada	F	350 Stacy G	Spencer	SD	57374	43.7557	-97.5936	343	Developme	3/5/1972	8be473af4f	1.372E+09	44
10	3.596E+15	fraud_Goye	shopping_p	4.37	David	Everett	M	4138 David	Morrisdale	PA	16858	41.0001	-78.2357	3688	Advice wor	5/27/1973	71a1da150c	1.372E+09	41
11	3.547E+15	fraud_Kilba	food_dinin	66.54	Kayla	Obrien	F	7921 Rober	Prairie Hill	TX	76678	31.6591	-96.8094	263	Barrister	5/30/1956	a7915132c7	1.372E+09	31
12	2.243E+15	fraud_Feil,	food_dinin	7.01	Samuel	Jenkins	M	43235 Mcke	Westport	KY	40077	38.4921	-85.4524	564	Pensions cc	4/10/1996	3b8e4d02d	1.372E+09	38
13	5.715E+11	fraud_Gottl	kids_pets	42.4	Louis	Fisher	M	45654 Hess	Fort Washa	WY	82514	43.0048	-108.8964	1645	Freight forv	2/26/1976	fa3071565d	1.372E+09	42
14	6.593E+15	fraud_Conr	home	2.91	Melissa	Meza	F	244 Abbott	Loxahatche	FL	33470	26.7383	-80.276	26551	Paramedic	1/4/1977	a21cb82e7c	1.372E+09	2
15	4.988E+12	fraud_Bech	food_dinin	7.93	William	Thompson	M	977 Rita Gr	Rock Tavern	NY	12575	41.4575	-74.1659	2258	Building su	3/17/1937	d0d2b5cca5	1.372E+09	4
16	6.012E+15	fraud_Lubo	kids_pets	2.91	Ashley	Whitney	F	4038 Smith	Jones	AL	36749	32.5104	-86.8138	1089	Materials e	11/2/1971	61dca41a97	1.372E+09	32
17	4.571E+15	fraud_Welc	entertainm	24.73	Christine	Leblanc	F	5097 Jodi V	Deltona	FL	32725	28.8989	-81.2473	88735	Commercia	4/9/1988	ea11379e8a	1.372E+09	2
18	4.907E+18	fraud_Hickl	shopping_p	2.33	Charles	Moreno	M	838 Franklii	Key West	FL	33040	24.6557	-81.3824	32891	Town planr	2/13/1987	00da724953	1.372E+09	24
19	4.909E+15	fraud_Lang	kids_pets	16.6	Lauren	Torres	F	03030 Whit	Grandview	TX	76050	32.2779	-97.2351	5875	Radiograph	7/24/1992	67288141e8	1.372E+09	33
20	4.861E+18	fraud_Mori	entertainm	80.11	Ashley	Cruz	F	65417 Wals	Saint Aman	LA	70774	30.2385	-90.8435	10076	Surveyor, n	12/16/1977	71bb6ee81	1.372E+09	3
21	6.538E+15	fraud_Pros	personal_c	5.71	Gina	Grimes	F	444 Robert	Clarks Mills	PA	16114	41.3851	-80.1752	606	Energy mar	9/22/1997	2d7640ea4f	1.372E+09	40
22	2.284E+15	fraud_Corv	travel	8.53	Shannon	Williams	F	9345 Spenc	Alpharetta	GA	30009	34.077	-84.3033	165556	Prison offic	12/27/1997	1650f4f052	1.372E+09	33
23	4.56E+18	fraud_Gottl	kids_pets	37.95	Stacy	Villegas	F	20581 Pena	Colorado Sp	CO	80951	38.8881	-104.6556	525713	Museum/ga	5/9/1992	b14cd1ccf7	1.372E+09	39
24	4.563E+12	fraud_Tillm	travel	1.74	Christophe	Johnson	M	28711 Kristi	Greenville	OH	45331	40.0987	-84.6342	22930	Media plan	11/26/1971	d8edb8556a	1.372E+09	40
25	2.132E+14	fraud_Veur	travel	6.02	Rebecca	Conley	F	181 Moren	Tomahawk	WI	54487	45.4963	-89.7273	9594	Seismic int	11/23/1997	79f931ffc97	1.372E+09	4

Figure 4.1 Snapshot of the Sample Dataset(Researcher, JolaoshoA.O, 2023)

Required packages were imported and also, the distribution of each variableplotted as shown in figure 4.2

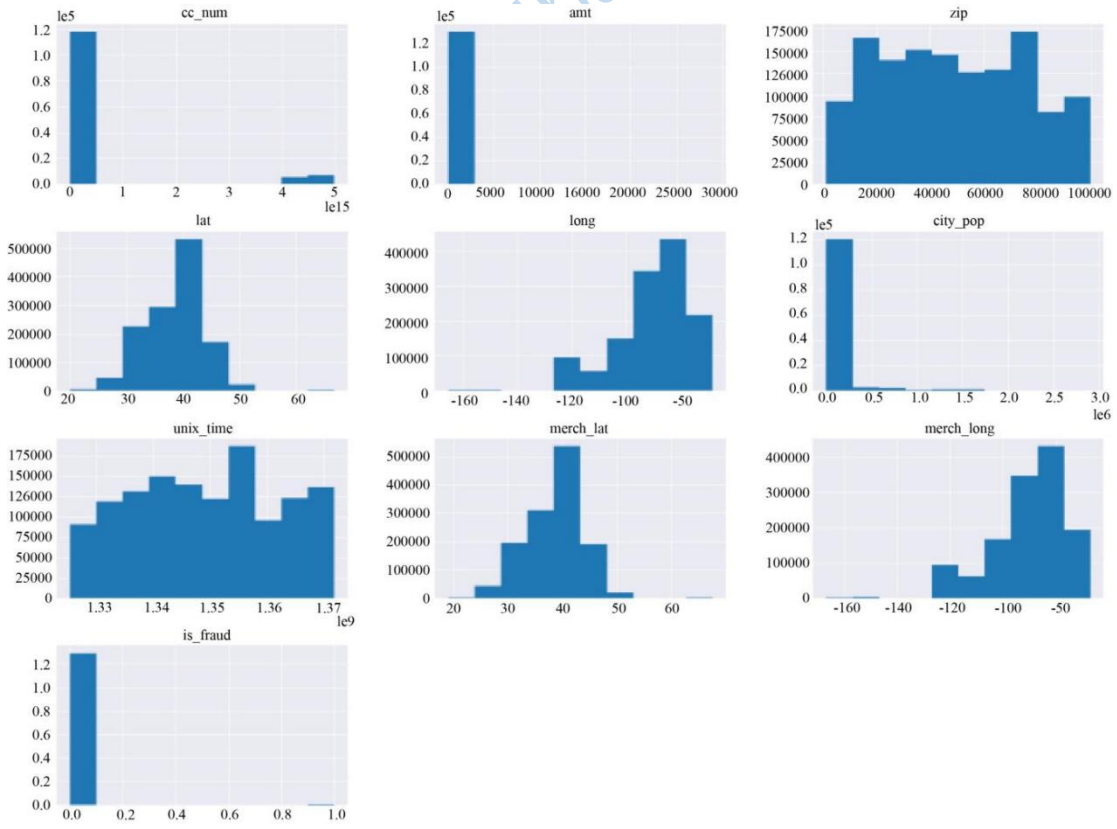


Figure 4.2: Plot of the Distribution of Each Variable (Researcher, JolaoshoA.O, 2023)

4.1.1 Result on Dataset Preprocessing and Exploratory Data Analysis

The numeric values in the dataset was looked into, to get a sense of what it was like. The output of the describe function on the dataset is shown in Figure 4.3.

	cc_num	amt	zip	lat	long	city_pop	unix_time
count	1296675.000000	1296675.000000	1296675.000000	1296675.000000	1296675.000000	1296675.000000	1296675.000000
mean	417192042079641088.000000	70.351035	48800.671097	38.537622	-90.226335	88824.440563	1349243636.726123
std	1308806447000789248.000000	160.316039	26893.222476	5.075808	13.759077	301956.360689	12841278.423360
min	60416207185.000000	1.000000	1257.000000	20.027100	-165.672300	23.000000	1325376018.000000
25%	180042946491150.000000	9.650000	26237.000000	34.620500	-96.798000	743.000000	1338750742.500000
50%	3521417320836166.000000	47.520000	48174.000000	39.354300	-87.476900	2456.000000	1349249747.000000
75%	4642255475285942.000000	83.140000	72042.000000	41.940400	-80.158000	20328.000000	1359385375.500000
max	4992346398065154048.000000	28948.900000	99783.000000	66.693300	-67.950300	2906700.000000	1371816817.000000

Figure 4.3: Snapshot of the Numerical Values in the Dataset (Describe Function Analysis) (Researcher, JolaoshoA.O, 2023)

The provided table represents a summary of descriptive statistics for different columns in a dataset

i. cc_num: This column seems to represent credit card numbers. The mean and standard deviation values are quite large, suggesting that these numbers are being treated as continuous variables. However, credit card numbers are typically categorical identifiers and should not be treated as numerical variables for analysis.

ii. amt: This column represents transaction amounts. The mean transaction amount is approximately $70.35, with a standard deviation of $160.32. The minimum transaction amount is $1, and the maximum is $28,948.90. This indicates a wide range of transaction amounts in the dataset.

iii. zip: This column likely represents zip codes. The mean zip code value is around 48,800, which may not be very meaningful. The standard deviation is 26,893, indicating significant variability in zip codes.

iv. lat and long: These columns represent latitude and longitude coordinates, respectively. The mean latitude is around 38.54, and the mean longitude is around -90.23. The standard deviations for both are relatively small, indicating that the data points are not spread out widely.

v. city_pop: This column represents city populations. The mean city population is around 88,824, with a significant standard deviation of 301,956. This suggests a wide variation in city populations across the dataset.

vi. unix_time: This column represents Unix timestamps (seconds since January 1, 1970). The mean Unix timestamp is approximately 1.35 billion seconds, with a standard deviation of around 12.8 million seconds. The range of Unix timestamps is quite extensive.

vii. merch_lat and merch_long: These columns represent latitude and longitude coordinates for merchant locations. The mean latitude is around 38.54, and the mean longitude is around -90.23, similar to the lat and long columns.

viii. is_fraud: This column appears to be a binary indicator (0 or 1) that represents whether a transaction is fraudulent (1) or not (0). The mean value is 0.0058, indicating that a very small portion of transactions are flagged as fraudulent. The standard deviation is 0.0759, which implies some variability in fraud occurrences.

Further, fraud distribution of the dataset was examined dataset using the snippet code below

*100*df.is_fraud.value_counts(normalize=True)*
0 99.421135
1 0.578865
Name: is_fraud, dtype: float64

This shows, the imbalance class of the target is_fraudtwo categories, 0 and 1, indicating non-fraudulent (0) and fraudulent (1) cases. In non-fraudulent cases (0), approximately 99.42% of the data falls into this category. In fraudulent cases (1), approximately 0.58% of the data falls into this category.

Also, the number of fraud per month was identified and analyzed as shown in figure 4.4

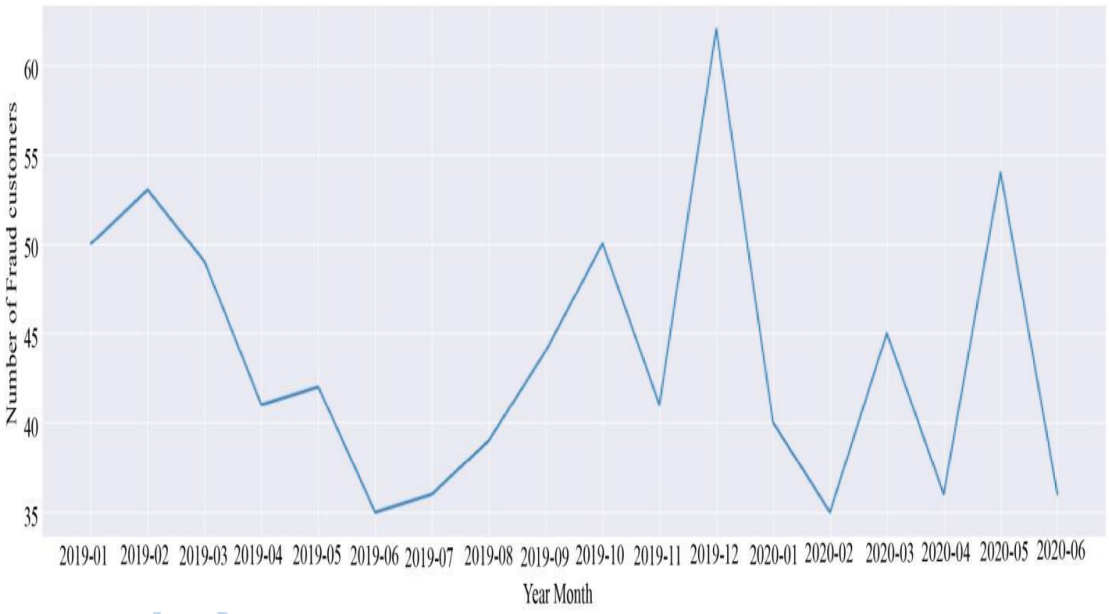


Figure 4.4: Number of Fraud Per Month (Researcher, JolaoshoA.O, 2023)

From the above visualizations it can be deduced that,

i. Most number of transactions happened in the afternoon times of the day, i.e., after 12 pm in the afternoon.

ii. Also, most of the transactions according to the 'trans_week' data have happened during 'sunday' and 'monday' weekdays of the week.

iii. Moreover, according to the 'trans_year_month' data, most transactions have happened in the January, February, October, December months of 2019 and in the May month of 2020.

iv. Also, it can be noted that the number of fraud customers and the number of fraud transaction have increased in the time of December, which again is the holiday season.

Gender fraud distribution was grouped and plotted andage-fraud distribution was created as shown in Figures 4.5 and 4.6

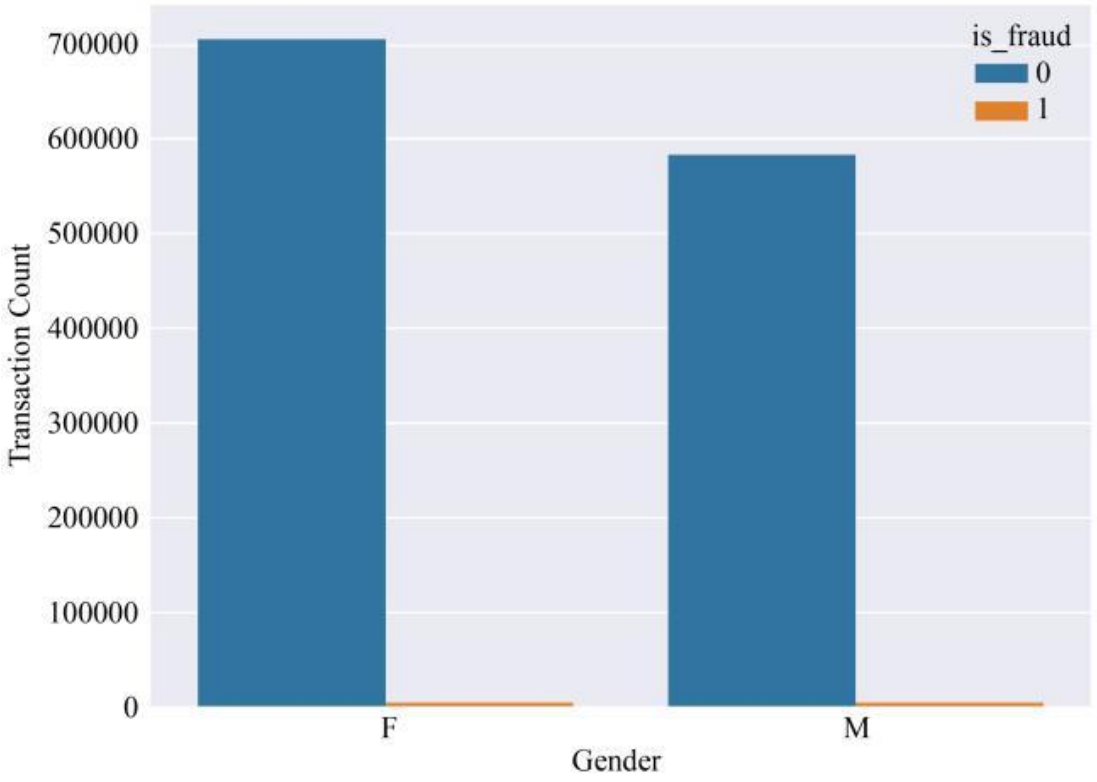


Figure 4.5: Distribution of Fraud Over Gender Chart (Researcher, JolaoshoA.O, 2023)

From the above visualization it can be observed that women contribute the most to the amount of the transaction frequencies.Although women do participate in fraud, the amount of women involved in fraud with respect to the number of transactions

involving women is 0.52% whereas the same for men is about 0.64%.It can be concluded that women are involved in most of the transactions and hence, they be more prone to frauds.Therefore, while there is a need for all sexes in the data to be knowledgeable about the frauds and their methods happening due to credit cards, in order to reduce the amount of frauds women should be educated and trained to be a bit more vigilant since they are much more prone to frauds.It can be concluded that men are a bit more inclined to be involved in fraud although both the sexes appear to be almost equally involved in all fraudulent transactions.

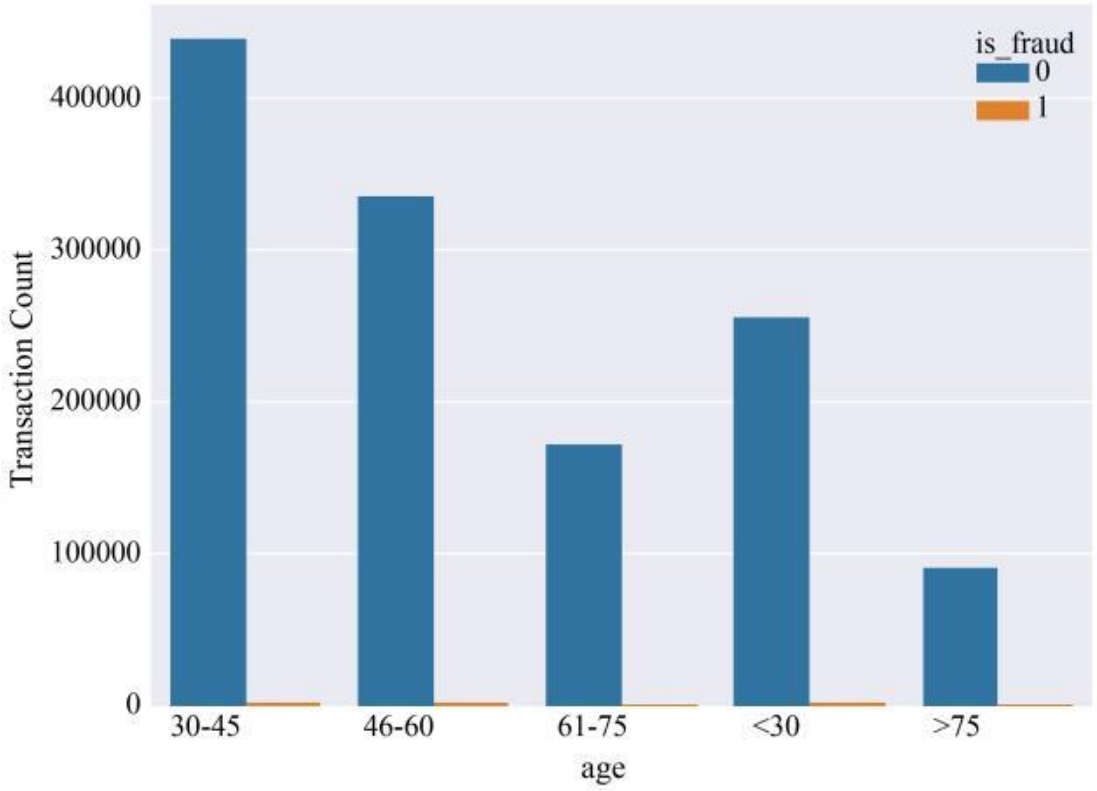


Figure 4.6: Age-Fraud Distribution Chart(Researcher, JolaoshoA.O, 2023)

From the plots it can be observed that the most number of transactions in the datset have been done by the people in the 30-45 age group.Also, the 46-60 age group has done significant number of transactions.With respect to the total number of transaction made by a particular age group the people in the >75 age group are the

most affected, wherein, about 1% transaction made by these people have been fraudulent.These are the people that are much prone to frauds and hence their transactions can be monitored with much more vigilance and they need to be educated regarding the frauds happening in order to reduce fraudulent transactions.The old age people might be targeted by fraudsters to take advantage of their lack of knowledge towards finance.

Correlation Analysis of the Numeric Features

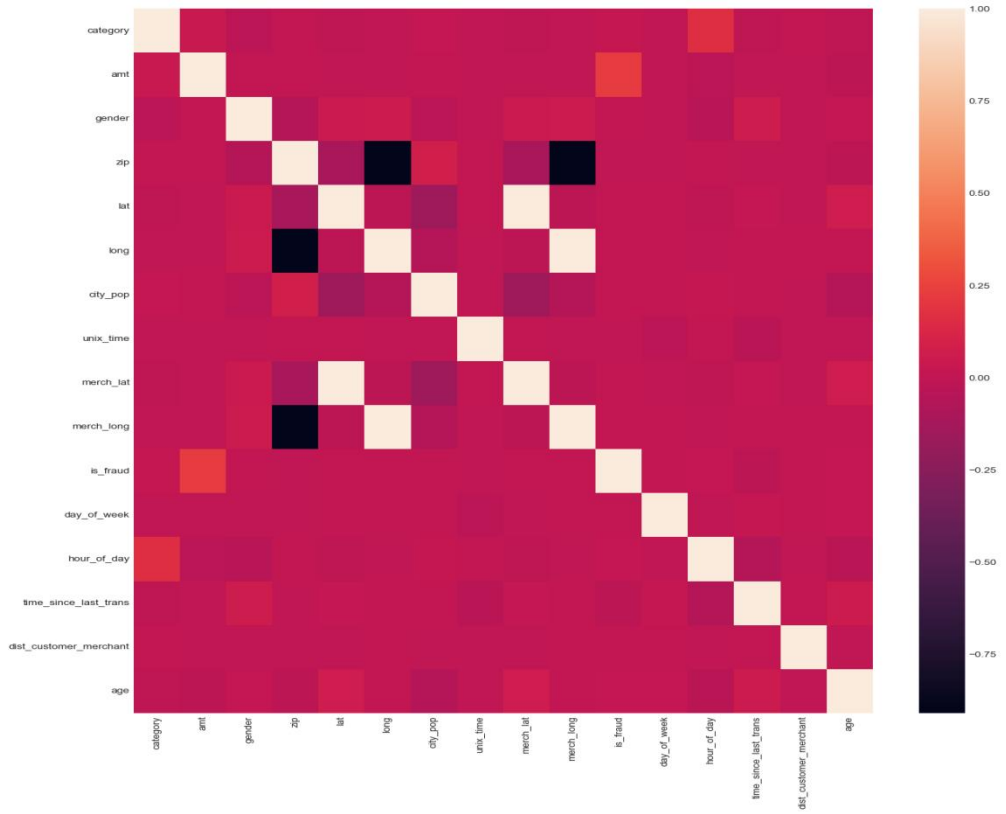


Figure 4.7: A Correlation Matrix of the Numeric Features (Researcher, JolaoshoA.O, 2023)

The correlation matrix shows the Pearson correlation coefficients between different pairs of columns in the dataset. These correlation coefficients quantify the linear relationships between variables.The main focus is on the correlation of the is_fraud column (indicating fraud) with other columns.The highest correlation value is approximately 0.22, indicating a positive correlation between the amt (transaction

amount) and the likelihood of fraud (is_fraud). This suggests that higher transaction amounts might be associated with a slightly higher likelihood of fraud.

Strong Correlations: There are some strong positive correlations, as indicated by coefficients close to 1:Strong positive correlation between unix_time (correlation of approximately 0.999). This suggests that these two variables are almost perfectly linearly related. It's important to investigate whether there's a data collection or preprocessing reason for this strong correlation.Strong positive correlation between long and merch_long, as well as between lat and merch_lat. This is likely due to the relationship between transaction locations and merchant locations.

Correlations Close to 0:Most other correlations are close to 0, indicating weak or no linear relationships between the variables.For example, there are weak correlations between transaction locations (lat, long) and the target variable (is_fraud).Investigate variables like cc_num and unix_time that have strong correlations with other columns. These columns might need special handling, such as transformation or exclusion from the feature set, depending on their relevance.

4.2 Machine Learning Models and Performance Evaluation

Three (3) different algorithms was used on the processed dataset and three(3)sampling techniques to balance the dataset was used. Eight (8) different models was created and used for the prediction of credit card fraud. The features 'zip', 'lat', 'long', 'city_pop', 'unix_time', 'merch_lat', and 'merch_long' have been assumed to provide no significant information in the model-building phase. Hence, they, along with the original features that have been encoded, have been dropped from the dataset.

4.2.1 Decision Tree Classifier

Confusion Matrix for Decision Tree Classifier

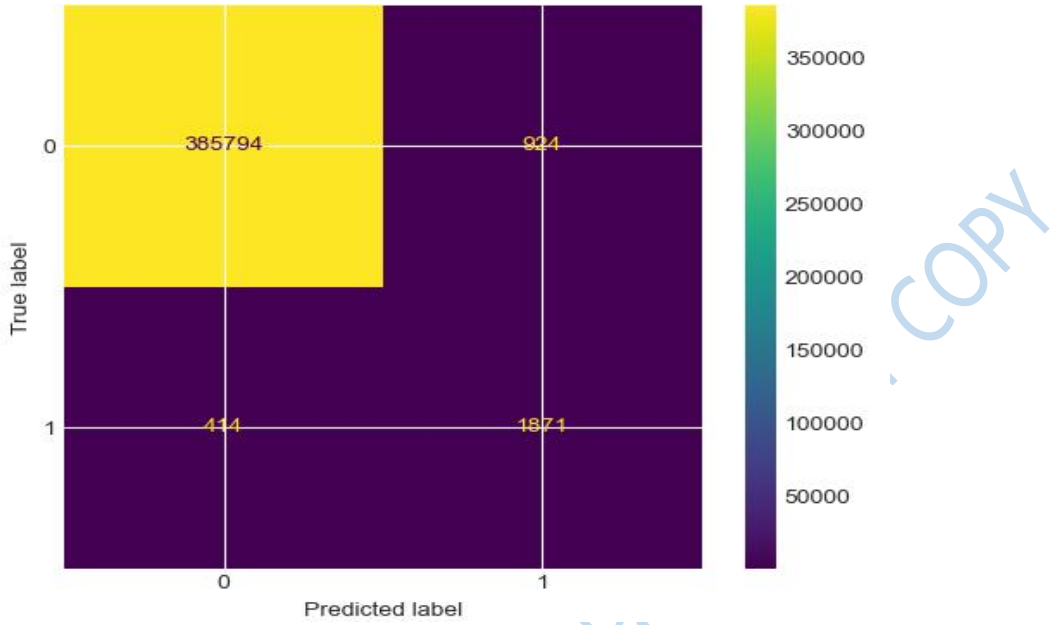


Figure 4.8: Decision Tree Confusion Matrix (Researcher, JolaoshoA.O, 2023)

From the figure 4.8 True Positive (TP) is1871 (The model correctly predicted 1871 instances as "isfraud".), False Positive (FP) is 924 (The model predicted 924 instances as "isfraud" when they were actually "notfraud").True Negative (TN) is 385794 (The model correctly predicted 385794 instances as "notfraud").False Negative (FN) is 414 (The model predicted 414 instances as "notfraud" when they were actually "isfraud"). This implies that the model correctly identified 1871 instances as "isfraud", correctly identified 385794 instances as "notfraud", the model made 924 false positive predictions and the model made 414 false negative predictions.

Table 4.1 Classification Report For Decision Tree Classifier(Researcher, JolaoshoA.O, 2023)

	Model Name	Training Score	Testing Score	Accuracy	F1 Score	Precision	Recall
1	Decision Tree - imbalance class	0.998622	0.954263	0.998026	0.997946	0.890376	0.757112
2	Decision Tree - Random Under Sampling	0.987438	0.954263	0.954263	0.954263	0.951370	0.957295
3	Decision Tree - Random Over Sampling	0.991297	0.946100	0.946100	0.946100	0.937265	0.955243
4	Decision Tree - SMOTE[Hyperparameter Tuned]	0.994949	0.940217	0.940217	0.940217	0.950549	0.930108

In Table 4.1, key metrics accuracy, F1 score, precision, and recall was used. The Imbalanced Class model exhibits extremely high accuracy and F1 score, suggesting excellent performance on the training data. It has relatively low recall (true positive rate), which is a concern for fraud detection. It may not be effective at capturing all fraudulent cases. However the modeltends to be biased towards the legitimate transactions thus, three resampling techniques was used; random undersampling, random oversampling, SMOTE

Decision Tree - Random Under Sampling: This model maintains a high level of accuracy and F1 score while achieving better recall compared to the Imbalanced Class model. The precision is also high, indicating that when it predicts fraud, it is usually correct.

Decision Tree - Random Over Sampling: This model, similar to the Random Under Sampling model, shows a good balance between precision and recall. It addresses class imbalance by oversampling the minority class.

Decision Tree - SMOTE: The SMOTE model also balances precision and recall, but it has slightly lower recall compared to the Random Under Sampling and Random Over Sampling models. SMOTE is an oversampling technique that generates synthetic samples to balance the dataset.

4.2.2 Random Forest

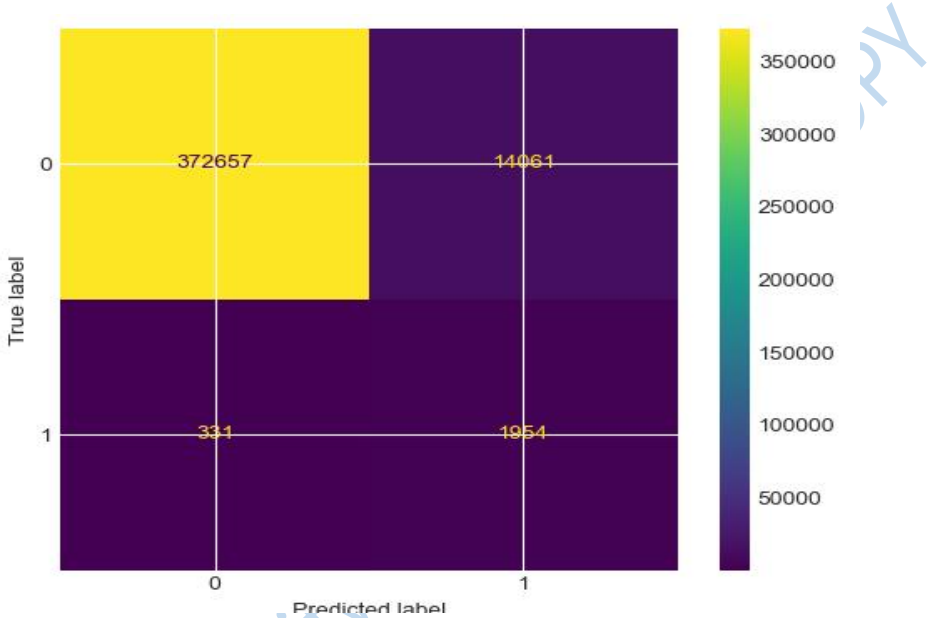


Figure 4.9: Confusion Matrix for Random Forest Classifier**(Researcher, JolaoshoA.O, 2023)**

From figure 4.9, the model correctly predicted 1954 instances as "isfraud" (True Positive (TP)=1954). The model predicted 14061 instances as "isfraud" when they are actually "notfraud" (False Positive (FP)=14061). Also, the model correctly predicted 372657 instances as "notfraud" (True Negative (TN)=372657).The model predicted 331 instances as "notfraud" when they were actually "isfraud" (False Negative (FN)=331). This implies that the model correctly identified 1954 instances as "isfraud".The model correctly identified 372657 instances as "notfraud", the model made 14061 false positive predictions and the model made 331 false negative predictions.

Table 4.2 Classification Report For Random Forest (Researcher, JolaoshoA.O, 2023)

	Model Name	Training Score	Testing Score	Accuracy	F1 Score	Precision	Recall
1	Random Forest - imbalance class	0.999997	0.998257	0.998257	0.998147	0.954211	0.738731
2	Random Forest - Random UnderSampling	1.000000	0.965808	0.965808	0.965803	0.977211	0.953737
3	Random Forest - Random Over Sampling	1.000000	0.954978	0.954978	0.954965	0.967148	0.941176
4	Random Forest - SMOTE [Hyperparameter Tuned]	1.000000	0.953804	0.973804	0.953805	0.955117	0.953405

From Table 4.2, Imbalanced Class shows extremely high accuracy and F1 score on the test data, which indicates excellent performance on the training data. It has a relatively low recall, suggesting that it may not effectively capture all fraudulent cases. Also, the modeltends to be biased towards the legitimate transactions thus, three resampling techniques was used; random undersampling, random oversampling, SMOTE.

Random Forest - Random Under Sampling: This model maintains a high level of accuracy, F1 score, and precision. It also achieves a significantly improved recall compared to the Imbalanced Class model, which is important for fraud detection.

Random Forest - Random Over Sampling: Random Under Sampling model, balances precision and recall well and performs impressively.

Random Forest - SMOTE [Hyperparameter Tuned]**:** This model, with hyperparameter tuning, achieves high accuracy, F1 score, and precision. It also maintains a strong recall rate, making it a well-rounded model.

The Random Forest models with data sampling techniques (Random Under Sampling, Random Over Sampling, and SMOTE) generally outperform the Imbalanced Class model in terms of capturing fraudulent cases (higher recall). Hyperparameter tuning further improves the performance metrics offer a well-balanced performance across various metrics.

4.2.3 Logistic Regression

Table 4.3 Classification Report For Logistic Regression (Researcher, JolaoshoA.O, 2023)

	Model Name	Training Score	Testing Score	Accuracy	F1 Score	Precision	Recall
1	Logistic Regression-Imbalance Class	0.993701	0.993747	0.993747	0.991093	0.000000	0.000000
2	Logistic Regression-with Random Undersampling	0.836315	0.835258	0.835258	0.833685	0.916207	0.738011
3	Logistic Regression - Random Over Sampling	0.837639	0.832594	0.832594	0.830971	0.913249	0.734772
4	Logistic Regression - SMOTE [Hyperparameter Tuned]	0.835276	0.841486	0.841486	0.839747	0.890376	0.737319

From Table 4.3, the Imbalance Class model achieves high accuracy but has low precision and recall. It seems to have issues with classifying the minority class

("isfraud"). Random Undersampling shows lower accuracy compared to the Imbalance Class model, but it has significantly improved precision and recall, making it better at capturing fraud cases. Random Over Sampling model has good precision and recall, addressing the class imbalance issue. Logistic Regression - SMOTE [Hyperparameter Tuned] achieves improved accuracy, precision, and recall compared to the Imbalance Class model. This implies that the Imbalance Class model has high accuracy but struggles with precision and recall, indicating issues with identifying "isfraud" cases. Models with data sampling techniques (Random Undersampling, Random Over Sampling, and SMOTE) show significant improvements in precision and recall. Hyperparameter tuning and SMOTE further enhance the model's performance.

Table 4.4: Summary of all Classification Model (Researcher, JolaoshoA.O, 2023)

	Model Name	Training Score	Testing Score	Accuracy	F1 Score	Precision	Recall
0	Logistic Regression-Imbalance Class	0.993701	0.993747	0.993747	0.991093	0.000000	0.000000
1	Logistic Regression-with Random Undersampling	0.836315	0.835258	0.835258	0.833685	0.916207	0.738011
2	Logistic Regression - Random Over Sampling	0.837639	0.832594	0.832594	0.830971	0.913249	0.734772
3	Logistic Regression - SMOTE [Hyperparameter Tuned]	0.835276	0.841486	0.841486	0.839747	0.890376	0.737319
4	Decision Tree -imbalance class	0.998622	0.954263	0.998026	0.997946	0.890376	0.757112
5	Decision Tree -Random Under Sampling	0.987438	0.954263	0.954263	0.954263	0.951370	0.957295
6	Decision Tree -Random Over Sampling	0.991297	0.946100	0.946100	0.946100	0.937265	0.955243
7	Decision Tree - SMOTE[Hyperparameter Tuned]	0.994949	0.940217	0.940217	0.940217	0.950549	0.930108
8	Random Forest -imbalance class	0.999997	0.998257	0.998257	0.998147	0.954211	0.738731
9	Random Forest -Random UnderSampling	1.000000	0.965808	0.965808	0.965803	0.955117	0.953737
10	Random Forest -Random Over Sampling	1.000000	0.954978	0.954978	0.954965	0.967148	0.941176
11	Random Forest -SMOTE [Hyperparameter Tuned]	1.000000	0.953804	0.973804	0.953805	0.955117	0.953405

From table 4.4, a total of twelve distinct models have been generated. Among the twelve (12) models constructed, the Random Forest Classifier, developed through the utilisation of the SMOTE sampling technique and subsequent hyperparameter tuning, has yielded the most desirable outcome, exhibiting an accuracy rate of 97.4%. Therefore, Random Forest Classifier - SMOTE sampling with hyperparameter tuning is the optimal model for predicting credit card fraud in this work.

Also to determine which model is the best based on Accuracy, F1 Score, Precision, and Recall, the specific objectives and priorities of your fraud detection task should be considered.

Best Accuracy: Random Forest - SMOTE [Hyperparameter Tuned]:This model achieved the highest accuracy (0.973804).

Best F1 Score:Random Forest - SMOTE [Hyperparameter Tuned]:This model achieved the highest F1 Score (0.953805), which is a balanced measure of precision and recall.

Best Precision:Random Forest - SMOTE [Hyperparameter Tuned]:This model achieved the highest precision (0.977211), indicating that when it predicted "isfraud," it was usually correct.

Best Recall:Random Forest - Random Under Sampling: This model achieved the highest recall (0.953737), indicating that it was effective at capturing a large proportion of "isfraud" cases.

Howeever, the choice of the "best" model depends on the specific goals and constraints. To prioritize precision (minimize false positives), consider the model with

the highest precision. To prioritize recall (minimize false negatives), consider the model with the highest recall. To seek a balanced performance, the model with the highest F1 Score might be the choice.

4.3 Real Time Notification

Twilio was used in generating a real time text message if fraud is detected. Using Twilio to generate real-time text messages if fraud is detected is a proactive and effective way to respond to potentially fraudulent activities. Twilio is a cloud communications platform that allows to send SMS messages, among other communication channels. The steps below was used to integrate Twilio the fraud detection system:

Detect Fraudulent Activity: The machine learning model (RF hyperparameter Tuned) will continuously monitor transactions or activities for signs of fraud.

Trigger an Alert: When the fraud detection system identifies a potentially fraudulent transaction or activity, it them an alert to notify the appropriate personnel or system.

Integrate Twilio: Integrate Twilio's API into the model. Twilio provides APIs for sending SMS messages programmatically. Twillo was integrated using the snippet code below. The full programming code is attached in appendix I.


```
pip install twilio

import twilio
from twilio.rest import Client

# Twilio credentials
account_sid = 'your_account_sid'
auth_token = 'your_auth_token'
```

Craft the Message: The default message was defined to notify when a fraud is detected. For this research, the crafted message is “Fraudulent transaction detected on

your credit card! Please contact your bank immediately". The snippet code is given below

Send the SMS: When a fraud alert is triggered, the model used Twilio's API to send the SMS message to designated recipients

```
# Function to send an SMS alert
def send_sms_alert(message):
    try:
        client = Client(account_sid, auth_token)
        message = client.messages.create(
            body=message,
            from_=+2347066484968,
            to=+2347066484968
        )
        print(f"SMS Alert sent with SID: {message.sid}")
    except Exception as e:
        print(f"Error sending SMS Alert: {str(e)}")
```

Testing and Validation: The model was tested with integration with Twilio to ensure that messages are sent accurately and in a timely manner. The snippet code for the testing and validation is given below

```
# Make a prediction using the trained model
prediction = model.predict(transaction_data)

# If fraud is detected, send a text message alert
if prediction == 1:
    message = client.messages.create(
        body="Fraudulent transaction detected on your credit card! Please conta
        from_='',
        to=''
    )

    print(message.sid)
```

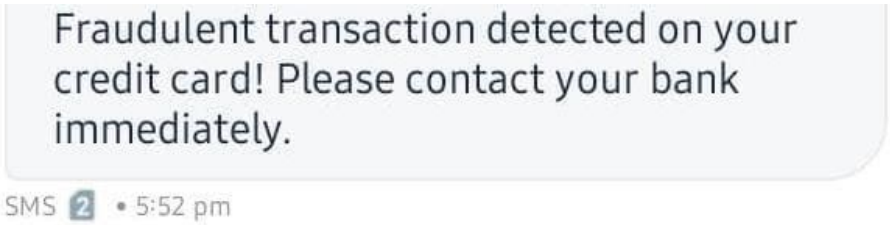


Figure 4.10: Sample of Simulated SMS Alert (Researcher, JolaoshoA.O, 2023).

4.4 Discussions of Findings

Various features of the data set have been analyzed and several insights have been obtained. The 'trans_date_trans_time' feature has been broken down into several components like 'Age', 'day of the week', 'month' in order to facilitate our analysis. These features have been thoroughly analyzed. It has been found that most transaction are being done after 12 noon and that during holiday seasons, the number of transaction along with the number of fraudulent transactions will increase. Old age people above 75 years are more susceptible to frauds. This is because, fraudsters might try to take advantage of their lack of knowledge about the constantly changing ways of how transactions are made.

The 'Female' gender people have been observed to do much of the transaction according to the dataset. Hence, transactions involving might be much prone to fraud. Also, by analyzing several demographic variables like city, state, zip etc it has been found that, several places like 'DE' state has 100% fraud rate and about 50 zip codes and 70 cities have 100% fraud rate. There might be some ill practices happening at the ground level at these places since all the transactions happening there are shown as fraudulent.

The features 'zip', 'lat', 'long', 'city_pop', 'unix_time', 'merch_lat','merch_long' have been asumed to provide no significant information in the model building phase. Hence, they along with the original features that have been encoded have been dropped from the dataset. 3 different algorithms have been implemented upon the processed dataset. Three sampling techniques in order to balance the dataset have also been implemented. The algorithms have also been implemented upon the dataset before balancing the dataset for demonstration purposes. Hence, about 12 different

models have been created the results of which have been summarized above. Out of the 12 models that have been built, the Random Forest Classifier built using the SMOTE sampling technique after hyper parameter tuning has provided the most preferable model with a accuracy of 0.97, f1 score of 0.95 and precision of 0.98. Hence, it can be said the the Radom Forest Classifier - SMOTE [Hyperparameter Tunned] sampling is the best model. Further, the model (Radom Forest Classifier - SMOTE [Hyperparameter Tunned]) was tested with integration with Twilio and the message was sent accurately and in a timely manner.

The result of this study has higher accuracy compared to a result in a work that reported that the machine learning models (LR, NB, LR and SVM) used in detecting credit card fraud captured the four fraud patterns (Risky MCC, Unknown web address, ISOResponse Code, Transaction above 100$)with an accuracy rate of 74%, 83%, 72% and 91% accuracy rates respectively[2]. This findings also corroborates the result in a study that reported an accuracy of 97% when compared with Decision Tree and Naive Bayes Technique for credit card fraud detection using supervised learning approach[3].

Endnotes

1. https://www.kaggle.com/datasets/kartik2112/fraudetection?select=fraudTrain.csv

2. A Thennakoon, C Bhagyani, S Premadasa, S Mihiranga, N Kuruwitaarachchi. *Real-time credit card fraud detection using machine learning*. In2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) 2019 Jan 10 (pp. 488-493). IEEE.

3. R More, C Awati, S Shirgave, R Deshmukh, S Patil. *Credit card fraud detection using supervised learning approach*. **International journal of scientific & technology research**. 2021;9(10):216-9.

Chapter Five

Conclusion

5.1 Summary of Findings

The Kaggle dataset was used for training the model Random Forest Classifier and Decision Tree Classifier and Logistic Regression. The 'trans_date_trans_time' feature was broken down into components like 'Age', 'day of the week', and 'month' to facilitate analysis. From the findings, transactions are most commonly made after 12 noon. During holiday seasons, both the number of transactions and the number of fraudulent transactions increase. Also, older individuals above 75 years are more susceptible to fraud, possibly due to their unfamiliarity with evolving transaction methods.

Transactions in the dataset are predominantly made by individuals identified as 'Female,' suggesting that transactions involving this gender may be more prone to fraud. Demographic variables such as city, state, and zip code were analyzed. 'DE' state showed a 100% fraud rate, and around 50 zip codes and 70 cities also had a 100% fraud rate. These areas might be associated with questionable practices or issues at the ground level, as all transactions from there were flagged as fraudulent. Further, features like 'zip', 'lat', 'long', 'city_pop', 'unix_time', 'merch_lat', 'merch_long' were deemed to provide no significant information and were dropped from the dataset.

Three different algorithms were implemented on the processed dataset. Three sampling techniques were used to balance the dataset. The models were also implemented on the dataset before balancing for demonstration purposes. In total, 12 different models were created and evaluated. Among the 12 models, the "Random Forest Classifier" built using the "SMOTE" sampling technique after hyperparameter

tuning provided the most preferable results. This model achieved an accuracy of 0.97, an F1 score of 0.95, and a precision of 0.98.

Finally, the chosen model, "Random Forest Classifier - SMOTE [Hyperparameter Tuned]," was tested with integration with Twilio. SMS alerts were sent accurately and in a timely manner when fraud was detected, enhancing the proactive response to potential fraudulent activities. The analysis involved thorough exploration of the dataset, feature engineering, model building, and integration with Twilio for real-time fraud alerts. The selected "Random Forest Classifier - SMOTE [Hyperparameter Tuned]" model demonstrated strong performance in fraud detection, making it the preferred choice for deployment in a fraud detection system.

5.2 Conclusion

In conclusion, the analysis and model development for fraud detection have provided valuable insights and a robust solution for real time identifying and responding to fraudulent activities. The combination of data analysis, feature engineering, and the selection of an effective machine learning model has resulted in a powerful fraud detection system. The "Random Forest Classifier - SMOTE [Hyperparameter Tuned]" model, along with Twilio integration, provides a comprehensive solution for detecting and responding to fraud in a timely and accurate manner. This system is well-equipped to safeguard against fraudulent transactions and protect both the business and its customers.

5.3. Recommendations

Based on the findings from this study, the following recommendations were made:

1. Continuous use and deploy the "Random Forest Classifier - SMOTE [Hyperparameter Tuned]" model as the primary fraud detection model. It has demonstrated strong performance across multiple metrics, including accuracy, F1 score, and precision.
2. Maintain the integration with Twilio for real-time SMS alerts. This is to ensure that the alerting system is continuously monitored and tested to confirm its reliability and responsiveness.
3. Perform periodic evaluations of the fraud detection model's performance to ensure its effectiveness in detecting evolving fraud patterns. Revisit and retrain the model as necessary to adapt to changing circumstances.
4. Continue to explore and engineer features that could enhance the model's predictive capabilities by considering additional data sources or variables that may provide valuable insights into fraud detection**.**
5. Given the observation that older individuals are more susceptible to fraud, consider implementing educational initiatives or materials to empower older customers with knowledge about safe transaction practices.
6. Implement a robust monitoring and reporting system to track the model's performance in a real-world environment. Ensure that alerts are acted upon promptly and that false positives/negatives are reviewed and addressed.

7. Encourage a culture of continual improvement in fraud detection strategies. Regularly review and update the system to stay ahead of emerging fraud tactics andfoster collaboration between data scientists, fraud analysts, and business stakeholders to exchange insights and align the fraud detection system with evolving business needs.
8. Consider the scalability of the system to accommodate increased transaction volumes and data growth over time.

Implementing these recommendations will help ensure the continued effectiveness of the fraud detection system and enhance the organization's ability to detect and respond to fraudulent activities in a proactive and efficient manner.

5.4 Contribution to Knowledge

The analysis and findings in the fraud detection project contribute to knowledge in several important ways:

1. Effective Fraud Detection Techniques: The project demonstrates the application of various machine learning models and data sampling techniques for fraud detection. It provides insights into which techniques are most effective in handling imbalanced datasets and improving the accuracy of fraud detection systems.
2. Feature Engineering and Analysis:The project showcases the importance of feature engineering and in-depth data analysis in understanding transaction patterns and identifying potential fraud indicators. Breaking down transaction timestamps and analyzing demographic variables offer valuable insights.
3. Geographic Patterns in Fraud:The observation of geographic patterns, such as areas with a high fraud rate, highlights the significance of localized fraud

detection strategies. It contributes to the understanding of how fraud can vary across different regions.

4. Gender-Based Analysis: The gender-based analysis sheds light on how transaction behavior may differ among demographic groups, emphasizing the need for tailored fraud detection approaches.
5. Integration with Real-time Alerting:The integration of the fraud detection model with Twilio for real-time SMS alerts demonstrates a practical and proactive approach to fraud prevention. It showcases how technology can be leveraged to respond swiftly to potential fraudulent activities.
6. Model Performance Metric: The project assesses models using a range of performance metrics, including accuracy, precision, recall, and F1 score. It provides a comprehensive understanding of how different models perform and the trade-offs between precision and recall.
7. Continual Improvement: The emphasis on continual improvement underscores the dynamic nature of fraud detection. It encourages organizations to adapt and evolve their strategies to stay ahead of fraudsters.

The project contributes to the body of knowledge by demonstrating best practices in fraud detection, showcasing the importance of data analysis and feature engineering, and highlighting the value of proactive alerting systems in mitigating fraud risks. These insights can be valuable for organizations across various industries aiming to enhance their fraud detection capabilities and protect against financial losses and risks.

5.5 Suggestions for Further Research

The following are the suggestions for further research:

1. Explore the application of deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), for fraud detection.

Deep learning models can capture intricate patterns in data and may improve fraud detection accuracy.

2. Investigate advanced anomaly detection techniques, including autoencoders and one-class SVMs, to identify unusual and potentially fraudulent patterns in transaction data.
3. Further research can be done by combining multiple data sources, such as transaction data, user behavior data, and network logs, to create a multi-modal fraud detection system. Fusion techniques can improve accuracy by leveraging complementary information.
4. Also, models can be developed with built-in explainability to provide clear insights into why a particular transaction or activity is flagged as fraudulent. Explainable AI can enhance trust and transparency in fraud detection systems.
5. More work can be done to investigate the challenges and opportunities of fraud detection in real-time data streams, where transactions and events are continuously generated. Implement stream processing techniques for timely detection.
6. Investigation of advanced feature engineering techniques can be explored, including natural language processing (NLP) for analyzing transaction descriptions and customer reviews, to extract valuable information for fraud detection.
8. Study the vulnerability of fraud detection models to adversarial attacks and develop robust models that can withstand attacks by fraudsters attempting to evade detection.

9. More work can also be done to investigate fraud detection in blockchain-based systems and cryptocurrencies, where traditional fraud patterns may differ significantly.

Bibliography

International Conference

Abakarim Y, Lahby M, Attioui A. *An efficient real time model for credit card fraud detection based on deep learning*. InProceedings of the 12th international conference on intelligent systems: theories and applications 2018 Oct 24 (pp. 1-7).

Abdulghani AQ, Uçan ON, Alheeti KM. Credit card fraud detection using XGBoost algorithm. In2021 14th International Conference on Developments in eSystems Engineering (DeSE) 2021 Dec 7 (pp. 487-492). IEEE.DOI**:** 10.1109/DeSE54285.2021.9719580

Aburbeian AM, Ashqar HI. *Credit card fraud detection using enhanced random forest classifier for imbalanced data*. InInternational Conference on Advances in Computing Research 2023 May 8 (pp. 605-616). Cham: Springer Nature Switzerland.

Adamopoulou E, Moussiades L. *An overview of chatbot technology. In IFIP International Conference on Artificial Intelligence Applications and Innovations* 2020 Jun 5 (pp. 373-383). Springer, Cham.

Aktar H, Masud MA, Aunto NJ, Sakib SN. *Classification using random forest on imbalanced credit card transaction data*. In2021 3rd International Conference on Sustainable Technologies for Industry 4.0 (STI) 2021 Dec 18 (pp. 1-4). IEEE.

Alamri MA, Ykhlef MA. *A machine learning-based framework for detecting credit card anomalies and fraud*. In2023 27th International Conference on Information Technology (IT) 2023 Feb 15 (pp. 1-7). IEEE.

Ali OM, Kareem SW, Mohammed AS. *Evaluation of electrocardiogram signals classification using CNN, SVM, and LSTM Algorithm*: A review. In2022 8th International Engineering Conference on Sustainable Technology and Development (IEC) 2022 Feb 23 (pp. 185-191). IEEE

Ashraf M, AbourezkaM.A, Maghraby F.A. A *comparative analysis of credit card fraud detection using machine learning and deep learning techniques*. InDigital Transformation Technology: Proceedings of ITAF 2020 2022 (pp. 267-282). Springer Singapore.

Demidova L, Ivkina M. *Defining the ranges boundaries of the optimal parameters values for the random forest classifier.* In2019 1st International Conference on Control Systems, Mathematical Modelling, Automation and Energy Efficiency (SUMMA) 2019 Nov 20 (pp. 518-522). IEEE.

Djuric M, Jovanovic L, Zivkovic M, Bacanin N, Antonijevic M, Sarac M. *The adaboost approach tuned by SNS* Conference on Paradigms of Computing, Communication and Data Sciences: PCCDS 2022 2023 Feb 24 (pp. 115-128). Singapore: Springer Nature Singapore.

Ghosh S,Dasgupta A, Swetapadma A. *A study on support vector machine based linear and non-linear pattern classification*. In2019 International Conference on Intelligent Sustainable Systems (ICISS) 2019 Feb 21 (pp. 24-28). IEEE.

Gupta K, Singh G, Singh V, Hassan M, Sharma U.*Machine learning based credit card fraud detection-A Review*. In2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC) 2022 May 9 (pp. 362-368). IEEE.

Islam MB, Avornu C, Shukla PK, Shukla PK. Cost Reduce: *Credit card fraud identification using machine learning*. In2022 7th International Conference on Communication and Electronics Systems (ICCES) 2022 Jun 22 (pp. 1192-1198). IEEE.

Krishna MV, Praveenchandar J. *Comparative Analysis of Credit Card Fraud Detection using Logistic regression with Random Forest towards an Increase in Accuracy of Prediction*. In2022 International Conference on Edge Computing and Applications (ICECAA) 2022 Oct 13 (pp. 1097-1101). IEEE.**DOI:** 10.1109/ICECAA55415.2022.9936488

Krishna SR, Agarwal V, Rao DE, Kakde VU, Kumari S, Vadar PS. *Machine learning based data mining for detection of credit card frauds*. In2023 International Conference on Inventive Computation Technologies (ICICT) 2023 Apr 26 (pp. 72-77). IEEE.

Kumar A, Prusti D, Purusottam I.S & Rath S.K, "*Real time SOA based credit card fraud detection system using machine learning techniques*," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2021, pp. 1-6, doi: 10.1109/ICCCNT51525.2021.9579598.

Mahajan A, Baghel VS, Jayaraman R. *Credit card fraud detection using logistic regression with imbalanced dataset*. In2023 10th International Conference on Computing for Sustainable Global Development (INDIACom) 2023 Mar 15 (pp. 339-342). IEEE.

Mahalaxmi KV, Rekha KS. *Accurate credit card fraudulent dataset using logistics regression compared with random forest*. In2022 International Conference on Business Analytics for Technology and Security (ICBATS) 2022 Feb 16 (pp. 1-5). IEEE.

Mathews SM. *Explainable artificial intelligence applications in NLP, biomedical, and malware classification: a literature review*. InIntelligent Computing: Proceedings of the 2019 Computing Conference, Volume 2 2019 (pp. 1269-1292). Springer International Publishing.

Mugundhan S, Venkataramanan P. *Data characteristic stability based random forest implementation of credit card fraud detection*. In2022 5th International Conference on Contemporary Computing and Informatics (IC3I) 2022 Dec 14 (pp. 1100-1104). IEEE.

Najadat H, Altiti O, Aqouleh AA, Younes M. *Credit card fraud detection based on machine and deep learning*. In2020 11th International Conference on

Information and Communication Systems (ICICS) 2020 Apr 7 (pp. 204-208). IEEE.

Ogundokun RO, Misra S, Fatigun OJ, Adeniyi JK. *Naïve bayes based classifier for credit card fraud discovery*. InEuropean, Mediterranean, and Middle Eastern Conference on Information Systems 2021 Dec 8 (pp. 515-526). Cham: Springer International Publishing.

Ogundokun RO, Misra S, Ogundokun OE, Oluranti J, Maskeliunas R. *Machine learning classification based techniques for fraud discovery in credit card datasets*. InApplied Informatics: Fourth International Conference, ICAI 2021, Buenos Aires, Argentina, October 28–30, 2021, Proceedings 4 2021 (pp. 26-38). Springer International Publishing.

Pranavi NS, Sruthi TK, Sirisha BJ, Nayak MS, Thadikemalla VS. *Credit card fraud detection using minority oversampling and random forest technique*. In2022 3rd International Conference for Emerging Technology (INCET) 2022 May 27 (pp. 1-6). IEEE.

Pristyanto Y, Setiawan NA, Ardiyanto I. *Hybrid resampling to handle imbalanced class on classification of student performance in classroom*. In2017 1st International Conference on Informatics and Computational Sciences (ICICoS) 2017 Nov 15 (pp. 207-212). IEEE\

Reshma R, Santhosh R, Mekala N. *An analytical approach to fraudulent credit card transaction detection using various machine learning algorithms*. In2023 Second International Conference on Electronics and Renewable Systems (ICEARS) 2023 Mar 2 (pp. 1400-1404). IEEE

Saheed YK, Hambali MA, Arowolo MO, Olasupo YA. *Application of GA feature selection on Naive Bayes, random forest and SVM for credit card fraud detection*. In2020 international conference on decision aid sciences and application (DASA) 2020 Nov 8 (pp. 1091-1097). IEEE.

Sailusha R, Gnaneswar V, Ramesh R, Rao GR. *Credit card fraud detection using machine learning*. In2020 4th international conference on intelligent computing and control systems (ICICCS) 2020 May 13 (pp. 1264-1270). IEEE.

Sen PC, Hajra M, Ghosh M. *Supervised classification algorithms in machine learning: A survey and review*. InEmerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018 2020 (pp. 99-111). Springer Singapore.

Şentürk S, Yerli, E &Soğukpınar, I. *Email phishing detection and prevention by using data mining techniques*. In 2017 International Conference on Computer Science and Engineering (UBMK) (pp. 707-712). IEEE., 2017, October

Shaik, A.B, Srinivasan S. *A brief survey on random forest ensembles in classification model*. InInternational Conference on Innovative Computing and Communications: Proceedings of ICICC 2018, Volume 2 2019 (pp. 253-260). Springer Singapore.

Sinayobye O, Musabe R, Uwitonze A, Ngenzi A. *A credit card fraud detection model using machine learning methods with a hybrid of undersampling and*

oversampling for handling imbalanced datasets for high scores. InInternational Conference on Applied Machine Learning and Data Analytics 2022 Dec 22 (pp. 142-155). Cham: Springer Nature Switzerland.

Sudha C, Akila D. *Credit card fraud detection system based on operational & transaction features using svm and random forest classifiers*. In2021 2nd International Conference on Computation, Automation and Knowledge Management (ICCAKM) 2021 Jan 19 (pp. 133-138). IEEE.: **DOI:** 10.1109/ICCAKM50778.2021.9357709

Thennakoon A,Bhagyani C,Premadasa S,Mihiranga S&Kuruwitaarachchi N.*Real-time credit card fraud detection using machine learning*. In 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 488-493). IEEE., 2019

Varmedja D, Karanovic M, Sladojevic S, Arsenovic M, Anderla A. *Credit card fraud detection-machine learning methods*. In2019 18th International Symposium Infoteh-Jahorina (INFOTEH) 2019 Mar 20 (pp. 1-5). IEEE.

Yang Y, Liu C, Liu N. *Credit card fraud detection based on csat-related adaboost*. InProceedings of the 2019 8th International Conference on Computing and Pattern Recognition 2019 Oct 23 (pp. 420-425).

Zhang Z, Huang S. Credit card fraud detection via deep learning method using data balance tools. In2020 international conference on computer science and management technology (ICCSMT) 2020 Nov 20 (pp. 133-137). IEEE.

Journal

Abdulkareem N.M, Abdulazeez A.M. *Machine learning classification based on Radom Forest Algorithm: A review*. **International Journal of Science and Business**. 2021;5(2):128-42.

Adnan M.S, Zaidi S, Bhargava P.*A novel support vector regression (SVR) model for the prediction of splice strength of the unconfined beam specimens. Construction and building materials.* 2020 Jul 10;248:118475

Ahmad I, BasherMi, Iqbal M.J, Rahim A. *Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection*. IEEE access. 2018 May 30;6:33789-95

Ahmed N, Amin R, Aldabbas H, Koundal D, Alouffi B, Shah T. *Machine learning techniques for spam detection in email and IoT platforms: analysis and research challenges*. Security and Communication Networks. 2022 Feb 3;2022:1-9.

Alarfaj FK, Malik I, Khan HU, Almusallam N, Ramzan M, Ahmed M. *Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms*. IEEE Access. 2022 Apr 12;10:39700-15.

Aldrees A, Awan HH, Javed MF, Mohamed AM. *Prediction of water quality indexes with ensemble learners: Bagging and boosting*. Process Safety and Environmental Protection. 2022 Dec 1;168:344-61

Al-Faqeh AW, Zerguine A, Al-Bulayhi MA, Al-Sleem AH, Al-Rabiah AS. *Credit card fraud detection via integrated account and transaction submodules.***Arabian Journal for Science and Engineering**. 2021 Oct;46(10):10023-31.

Al-Serw NA. *Undersampling and oversampling: An old and a new approach*. Analytics Vidhya. 2021

Alshutayri A. *Fraud prediction in movie theater credit card transactions using machine learning*. Engineering, Technology & Applied Science Research. 2023 Jun 2;13(3):10941-5. https://doi.org/10.48084/etasr.5950

Anand M, Velu A, Whig P. *Prediction of loan behaviour with machine learning models for secure banking*. **Journal of Computer Science and Engineering(JCSE)**. 2022 Feb 15;3(1):1-3

Anowu DN, Nyor T, Agbi SE, Nelson AI, Saliu AN. *Financial forensic analysis and fraud deterrence in listed deposit money banks in Nigeria*. **Gusau Journal of Accounting and Finance**. 2021 Oct 1;2(4):18

Anowu, D.N T Nyor, S.E Agbi, A.I Nelson, A.N Saliu. *Financial forensic analysis and fraud deterrence in listed deposit money banks in Nigeria*. **Gusau Journal of Accounting and Finance.** 2021 Oct 1;2(4):18

Arafa A, El-Fishawy N, Badawy M, Radad M. RN-SMOTE: *Reduced noise smote based on DBSCAN for enhancing imbalanced data classification*. **Journal of King Saud University-Computer and Information Sciences**. 2022 Sep 1;34(8):5059-74.https://doi.org/10.1016/j.jksuci.2022.06.005

Arista, A. *Comparison decision tree and logistic regression machine learning classification algorithms to determine Covid-19*. Sinkron. 7. 59-65. 2022:10.33395/sinkron.v7i1.11243.

Arun C, Lakshmi C. *Genetic algorithm-based oversampling approach to prune the class imbalance issue in software defect prediction*. Soft Computing. 2022 Dec;26(23):12915-31

Asha RB, KR SK. *Credit card fraud detection using artificial neural network*. Global Transitions Proceedings. 2021 Jun 1;2(1):35-41

Baesens B, Höppner S, Ortner I, Verdonck T. robROSE: *A robust approach for dealing with imbalanced data in fraud detection*. Statistical Methods & Applications. 2021 Sep;30(3):841-61

Baig MS, Jaisharma K. *Comparison of novel optimized random forest technique and gradient boosting for credit card fraud detection with improved precision*. **Journal of Pharmaceutical Negative Results**. 2022 Sep 27:851-6.

Bengio Y, Lodi A, Prouvost A. *Machine learning for combinatorial optimization: a methodological tour d'horizon*. **European Journal of Operational Research**. 2021 Apr 16;290(2):405-21

Berhane T, Melese T, Walelign A, Mohammed A. *A hybrid convolutional neural network and support vector machine-based credit card fraud detection model*. Mathematical Problems in Engineering. 2023 Jun 3;2023

Bhatia, S; Naib, B.B; Ashraf, G. Credit card fraud detection using classification algorithm. TechRxiv. Preprint.2023: https://doi.org/10.36227/techrxiv.23377547.v1

Bokaba T, Doorsamy W, Paul B.S. *Comparative study of machine learning classifiers for modelling road traffic accidents*. Applied Sciences. 2022 Jan;12(2):828

C Zhang, X Lei, L Liu.*Predicting metabolite–disease associations based on LightGBM model*. Frontiers in Genetics. 2021 Apr 13;12:660275

Cano A, Krawczyk B. ROSE: *Robust online self-adjusting ensemble for continual learning on imbalanced drifting data streams*. Machine Learning. 2022 Jul;111(7):2561-99

Charbuty B, Abdulazeez A. *Classification based on decision tree algorithm for machine learning*. **Journal of Applied Science and Technology Trends**. 2021 Mar 24;2(01):20-8.

Chatterjee P, Das D, Rawat D. *Securing Financial Transactions: Exploring the role of federated learning and blockchain in credit card fraud detection*.TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.22683403.v1

Cherif A, Badhib A, Ammar H, Alshehri S, Kalkatawi M, Imine *A. Credit card fraud detection in the era of disruptive technologies: A systematic review*. **Journal of King Saud University-Computer and Information Sciences**. 2022 Dec 5.https://doi.org/10.1016/j.jksuci.2022.11.008

Chern F, Hechtman B, Davis A, Guo R, Majnemer D, Kumar S. Tpu-knn: *K nearest neighbor search at peak flop/s.* arXiv preprint arXiv:2206.14286. 2022 Jun 28.

Demir S, Sahin EK. *An investigation of feature selection methods for soil liquefaction prediction based on tree-based ensemble algorithms using AdaBoost, gradient boosting, and XGBoos*t. Neural Computing and Applications. 2023 Feb;35(4):3173-90

Denisko D, Hoffman M.M. *Classification and interaction in random forests. Proceedings of the National Academy of Sciences*. 2018 Feb 20;115(8):1690-2

Dhanaraj R.D, Rajkumar K, Hariharan U. *Enterprise IoT modeling: supervised, unsupervised, and reinforcement learning*. Business Intelligence for Enterprise Internet of Things. 2020:55-79

Dhanaraj R.K, Rajkumar K, Hariharan U. *Enterprise IoT modeling: supervised, unsupervised, and reinforcement learning*. Business Intelligence for Enterprise Internet of Things. 2020:55-79.

Du H, Lv L, Guo A, Wang H. *Autoencoder and lightgbm for credit card fraud detection problems*. Symmetry. 2023 Apr 6;15(4):870.https://doi.org/10.3390/sym15040870

Du H, Lv L, Guo A, Wang H. *Autoencoder and lightgbm for credit card fraud detection problems*. Symmetry. 2023 Apr 6;15(4):870.

Elreedy D, Atiya AF, Kamalov F. *A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning*. Machine Learning. 2023 Jan 5:1-21

Fakiha B. Forensic credit card fraud detection using deep neural network. **Journal of Southwest Jiaotong University**. 2023;58(1)

Farhang Ghahfarokhi A, Mansouri T, Sadeghi Moghaddam MR, Bahrambeik N, Yavari R, Fani Sani M. *Credit card fraud detection using asexual reproduction optimization*. Kybernetes. 2022 Sep 5;51(9):2852-76.

Fleischhauer V, Feldheiser A, Zaunseder S. *Beat-to-Beat Blood Pressure Estimation by Photoplethysmography and Its Interpretation*. Sensors. 2022 Sep 17;22(18):7037.

Gajowniczek K, Grzegorczyk I, Ząbkowski T, Bajaj C. *Weighted random forests to improve arrhythmia classification*. Electronics. 2020 Jan 3;9(1):99

Guan H, Li S, Wang Q, Lyulyov O, Pimonenko T. *Financial fraud identification of the companies based on the logistic regression model.* **Journal of Competitiveness**. 2022 Dec 1(4).

Gupta P, Varshney A, Khan MR, Ahmed R, Shuaib M, Alam S. *Unbalanced credit card fraud detection data: A machine learning-oriented comparative study of balancing techniques*. Procedia Computer Science. 2023 Jan 1;218:2575-84.

Hand DJ, Christen P, Kirielle N. F: *an interpretable transformation of the F-measure. Machine Learning*. 2021 Mar;110(3):451-6

Hossain MN, Hassan MM, Monir RJ. Analyzing the classification accuracy of deep learning and machine learning for credit card fraud detection. **Asian Journal For Convergence In Technology (AJCT)** ISSN-2350-1146. 2022 Dec 31;8(3):31-6.

Hussain D, Hussain I, Ismail M, Alabrah A, Ullah SS, Alaghbari HM. *A simple and efficient deep learning-based framework for automatic fruit recognition*. Computational Intelligence and Neuroscience. 2022 Feb 21;2022.

Ibrahim M. *Evolution of random forest from decision tree and bagging: A bias-variance perspective*. **Dhaka University Journal of Applied Science and Engineering**. 2022;7(1):66-71.

Ileberi E, Sun Y, Wang Z. *Performance evaluation of machine learning methods for credit card fraud detection using SMOTE and AdaBoos*t. IEEE Access. 2021 Dec 15;9:165286-94.

Jain Y, Tiwari N, Dubey S, Jain S. *A comparative analysis of various credit card fraud detection techniques*. **Int J Recent Technol Eng.** 2019 Jan;7(5S2):402-7.

Jone JS, Kipsy S. *Early prediction of heart diseases using logistic regression algorithm. EPRA***International Journal of Multidisciplinary Research (IJMR)**. 2023 Mar 9;9(3):72-82.

Jovanovic D, Antonijevic M, Stankovic M, Zivkovic M, Tanaskovic M, Bacanin N. *Tuning machine learning models using a group search firefly algorithm for credit card fraud detection*. Mathematics. 2022 Jun 29;10(13):2272.

Kadali ML, Ramakrishna VS, Chandra Mouli VS, Rajasekhar GJ. *Prediction of cardiovascular disease using machine learning algorithms with relief and lasso feature selection techniques*. Mathematical Statistician and Engineering Applications. 2022 Oct 18;71(4):5356-72

Kolhe M.L, Tiwari S, Trivedi M.C & Mishra K.K (Eds.). *Advances in data and information sciences: Proceedings of ICDIS 2019* (Vol. 94). Springer Singapore. https://doi.org/10.1007/978- 981-15-0694-9

Konstantinov A.V, Utkin L.V. *Interpretable machine learning with an ensemble of gradient boosting machine*s. Knowledge-Based Systems. 2021 Jun 21;222:106993

Kunapuli G. *Ensemble Methods for Machine Learning*. Simon and Schuster; 2023 May 2

Kunapuli G. *Ensemble methods for machine learning*. Simon and Schuster; 2023 May 2

Lavanya K. A *Comparison of Logistic Regression Classifier and Random Forest Classifier for the Accurate Classification of Credit Card Fraudulent Transactions*. **Journal of Survey in Fisheries Sciences**. 2023 Mar 8;10(1S):2008-17.**DOI:** https://doi.org/10.17762/sfs.v10i1S.435

Maniraj S.P & Saini, A & Ahmed, S & Sarkar, S. *Credit card fraud detection using machine learning and data science*. **International Journal of Engineering Research**. 2019:08. 10.17577/IJERTV8IS090031.

Mayabadi S, Saadatfar H. *Two density-based sampling approaches for imbalanced and overlapping data.* Knowledge-Based Systems. 2022 Apr 6;241:108217

McPherron SP, Archer W, Otárola-Castillo ER, Torquato MG, Keevil TL. *Machine learning, bootstrapping, null models, and why we are still not 100% sure which bone surface modifications were made by crocodiles*. **Journal of Human Evolution**. 2022 Mar 1;164:103071.

Mehbodniya A, Alam I, Pande S, Neware R, Rane KP, Shabaz M, Madhavan MV. *Financial fraud detection in healthcare using machine learning and deep learning techniques*. Security and Communication Networks. 2021 Sep 9;2021:1-8.

Meng Q. *Credit card fraud detection using feature fusion-based machine learning model*. Highlights in Science, Engineering and Technology. 2022 Dec 3;23:111-6.

Mienye ID, Sun Y. *A deep learning ensemble with data resampling for credit card fraud detection*. IEEE Access. 2023 Mar 27;11:30628-38.**doi:** 10.1109/access.2023.3262020

Mienye ID, Sun Y. *A machine learning method with hybrid feature selection for improved credit card fraud detection*. Applied Sciences. 2023 Jun 18;13(12):7254.

Mohamed S, Ashraf R, Ghanem A, Sakr M, Mohamed R. *Supervised machine learning techniques*: A Comparison.2022

Mohamed S, Ashraf R, Ghanem A, Sakr M, Mohamed R. *Supervised machine learning techniques: A comparison*.2022

More R, Awati C, Shirgave S, Deshmukh R, Patil S. *Credit card fraud detection using supervised learning approach*. **International journal of scientific & technology research**. 2021;9(10):216-9.

Muschelli III J. *ROC and AUC with a binary predictor: a potentially misleading metric*. **Journal of classification**. 2020 Oct;37(3):696-708

Mytnyk B, Tkachyk O, Shakhovska N, Fedushko S, Syerov Y. *Application of Artificial Intelligence for Fraudulent Banking Operations Recognition.* Big Data and Cognitive Computing. 2023 May 10;7(2):93.

Nababan AA, Khairi M, Harahap BS. Implementation of *k-nearest neighbors (KNN) algorithm in classification of data water quality*. **Jurnal Mantik**. 2022 Mar 20;6(1):30-5

Navaratna V, Reddy PA, Avinash PS, Jyothi TA. Credit card fraud detection using machine learning. Vol.11:Issue 6:June 2020:ISSN 0377-9254

Ngo G, Beard R, Chandra R. *Evolutionary bagging for ensemble learning*. Neurocomputing. 2022 Oct 21;510:1-4.

Nikhil MK, Maharshi MB, Tanooj MK, SriRam MD. *Credit card fraud detection using machine learning algorithms*. **Journal of Engineering Sciences**. 2023;14(04).

Nozari H, Sadeghi M.E. *Artificial intelligence and machine learning for real-world problems (a survey)*. **International Journal of Innovation in Engineering**. 2021 Oct 7;1(3):38-47.

Odeajo I, Akinmoluwa O, Sharon O, Otesanya T D, "*Financial fraud detection using machine learning : Credit card fraud*," **International Journal of Recent Engineering Science**, vol. 10, no. 3, pp. 23-32, 2023. Crossref, https://doi.org/10.14445/23497157/IJRES-V10I3P104

Paul A, Mukherjee DP, Das P, Gangopadhyay A, Chintha AR, Kundu S. *Improved random forest for classification*. IEEE Transactions on Image Processing. 2018 May 10;27(8):4012-24.1026236868

Raval J, Bhattacharya P, Jadav NK, Tanwar S, Sharma G, Bokoro PN, Elmorsy M, Tolba A, Raboaca MS. RaKShA: *A trusted explainable lstm model to classify fraud patterns on credit card transactions*. Mathematics. 2023 Apr 17;11(8):1901.

Reis I, Baron D, Shahaf S. *Probabilistic random forest: A machine learning algorithm for noisy data sets*. **The Astronomical Journal.** 2018 Dec 20;157(1):16.

Roy A, Sun J, Mahoney R, Alonzi L, Adams S, Beling P. *Deep learning detecting fraud in credit card transactions*. In2018 Systems and Information Engineering Design Symposium (SIEDS) 2018 Apr 27 (pp. 129-134). IEEE.

Sabzekar M, Hasheminejad S.M. *Robust regression using support vector regressions. Chaos, Solitons & Fractals*. 2021 Mar 1;144:110738

Sarker IH. Machine learning: Algorithms, real-world applications and research directions. SN computer science. 2021 May;2(3):160

Schonlau M, Zou RY. *The random forest algorithm for statistical learning*. **The Stata Journal**. 2020 Mar;20(1):3-29.

Seijo-Pardo B, Bolón-Canedo V, Alonso-Betanzos A. *Testing different ensemble configurations for feature selection*. Neural Processing Letters. 2017 Dec;46:857-80.DOI:10.1007/s11063-017-9619-1

Sen PC, Hajra M, Ghosh M. *Supervised classification algorithms in machine learning: A survey and review*. InEmerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018 2020 (pp. 99-111). Springer Singapore

Shanthamallu US, Spanias A. *Supervised learning. Inmachine and deep learning algorithms and applications* 2022 (pp. 9-21). Cham: Springer International Publishing

Shellyann Sooklal, Patrick Hosein. *Framework for credit card fraud detection using benefit-based learning and periodic features*, 14 March 2023, preprint (Version 1) available at Research Square [https://doi.org/10.21203/rs.3.rs-2652853/v1]

Shen Z, Shehzad A, Chen S, Sun H, Liu J. *Machine learning based approach on food recognition and nutrition estimation*. Procedia Computer Science. 2020 Jan 1;174:448-53

Sibindi R,Mwangi R.W, Waititu A.G. *A boosting ensemble learning based hybrid light gradient boosting machine and extreme gradient boosting model for predicting house prices*. Engineering Reports. 2022:e12599

Sidey-Gibbons J.A, Sidey-Gibbons C.J. *Machine learning in medicine: a practical introduction*. **BMC medical research methodology**. 2019 Dec;19:1-8.

Singh G.K, Bhayye A, Dhamnaskar S, Patil S & S.V Phulari. *Credit card fraud detection using isolation forest*. **International Journal of Recent Advances in Multidisciplinary Topics**, *2*(6), pp.118-119.2021

Singh G.K, Bhayye A, Dhamnaskar S, Patiland S, Phulari S.V. *Credit card fraud detection using isolation forest*. **International Journal of Recent Advances in Multidisciplinary Topics**, *2*(6), pp.118-119.2021

Sri Manvith, V & Redrowthu, V & Vasavi, R. *A performance comparison of machine learning approaches on intrusion detection dataset*. 782-788. 2021:10.1109/ICICV50876.2021.9388502.

Suganya E, CRajan. *An adaboost-modified classifier using particle swarm optimization and stochastic diffusion search in wireless IoT networks*. Wireless Networks. 2021 May;27:2287-99

Sulaiman M.A. *Evaluating data mining classification methods performance in Internet of things applications*. **Journal of Soft Computing and Data Mining**. 2020 Dec 6;1(2):11-25.

ThirunavukkarasuM, Achutha N, Adusumilli J. **International Journal of Computer Science and Mobile Computing**, Vol.10 Issue.4, April- 2021, pg. 71-79.DOI: 10.47760/ijcsmc.2021.v10i04.011

Tiwari A. *Supervised learning: From theory to applications*. InArtificial Intelligence and Machine Learning for EDGE Computing 2022 Jan 1 (pp. 23-32). Academic Press

Tiwari P, Mehta S, Sakhuja N, Kumar J, Singh, A.K. *Credit card fraud detection using machine learning: a study*. arXiv preprint arXiv:2108.10005. 2021 Aug 23

Tretiak K, Schollmeyer G, Ferson S. *Neural network model for imprecise regression with interval dependent variables*. Neural Networks. 2023 Apr 1;161:550-64.

Uddin S, Haque I, Lu H, Moni MA, Gide E. *Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction*. Scientific Reports. 2022 Apr 15;12(1):1-1.

Udeze CL, Eteng IE, Ibor AE. *Application of machine learning and resampling techniques to credit card fraud detection*. **Journal of the Nigerian Society of Physical Sciences**. 2022 Aug 15:769-.

Vaishnavi N.D, Geetha S. *Credit card fraud detection using machine learning algorithms*,Procedia Computer Science,Volume 165,2019,Pages 631-641,ISSN 1877-0509,https://doi.org/10.1016/j.procs.2020.01.057.

Vivek Y, Ravi V, Mane AA, Naidu LR. ATM fraud detection using streaming data Analytics. arXiv preprint arXiv:2303.04946. 2023 Mar 8.

Yang X, Wang Y, Byrne R, Schneider G, Yang S. *Concepts of artificial intelligence for computer-assisted drug discovery.* Chemical reviews. 2019 Jul 11;119(18):10520-94

Yee O.S, Sagadevan S, Malim N.H. *Credit card fraud detection using machine learning as data mining technique*. **Journal of Telecommunication, Electronic and Computer Engineering** (JTEC). 2018 Jan 29;10(1-4):23-7.

Yee O.S, Sagadevan S, Malim N.H. *Credit card fraud detection using machine learning as data mining technique*. **Journal of Telecommunication, Electronic and Computer Engineering** (JTEC). 2018 Jan 29;10(1-4):23-7

Yigin B.O, Algin O, Saygili G. *Comparison of morphometric parameters in prediction of hydrocephalus using random forests*. Computers in Biology and Medicine. 2020 Jan 1;116:103547.

Zewdie GK, Valladares C, Cohen MB, Lary DJ, Ramani D, Tsidu GM. *Data-driven forecasting of low-latitude ionospheric total electron content using the random forest and LSTM machine learning methods*. Space Weather. 2021 Jun;19(6):e2020SW002639.

Zhai X, Chen M, Lu W. *Fuel ratio optimization of blast furnace based on data mining. ISIJ International*. 2020 Nov 15;60(11):2471-6.

Zhao Z, Bai T. *Financial fraud detection and prediction in listed companies using smote and machine learning algorithms*. Entropy. 2022 Aug 19;24(8):1157.

Thesis

Abdulghani, Ahmed. *Employing machine learning techniques and fuzzy membership for detecting fraud transactions in credit card.* (Yayınlanmamış yüksek lisans tezi). Altınbaş Üniversitesi, Lisansüstü Eğitim Enstitüsü, İstanbul.2022

AlEmad M."*Credit card fraud detection using machine learning*". Thesis. Rochester Institute of Technology, 2022.

AlEmad, M, "*Credit card fraud detection using machine learning*". Thesis. 2022. Rochester Institute of Technology. Accessed fromhttps://scholarworks.rit.edu/theses/11318

Andeta J.A. *Road-traffic accident prediction model : Predicting the number of casualties[Dissertation]. 2021*. Available from: http://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-20146

Lucas Y. *Credit card fraud detection using machine learning with integration of contextual knowledge* (Doctoral dissertation, Université de Lyon; Universität Passau (Deutscheland)).2019

Lucas Y. *Credit card fraud detection using machine learning with integration of contextual knowledge* (Doctoral dissertation, Université de Lyon; Universität Passau (Deutscheland)).2019

Sayantan D. *Credit card fraud detection system using machine learning a project report*, Bachelor of Technology in information technology, May 2019

Shakya R. "*Application of machine learning techniques in credit card fraud detection"*. UNLV Theses, Dissertations, Professional Papers, and Capstones. 3454. http://dx.doi.org/10.34917/14279175. 2018

Shakya, Ronish, "*Application of machine learning techniques in credit card fraud detection"*. Unlv theses, Dissertations, Professional Papers, and Capstones. 3454.2018: http://dx.doi.org/10.34917/14279175.

Soulé-Dupuy C, Gaussier E, Lux M, Gianini G, Calabretto S, Granitzer M,Portier P.E. *Credit Card Fraud Detection using Machine Learning with Integration of Contextual Knowledge* (Doctoral dissertation, INSA Lyon).2019

Soulé-Dupuy C, Gaussier E,Lux M,Gianini G, Calabretto S, Granitzer M, Portier P.E. *Credit card fraud detection using machine learning with integration of contextual knowledge* (Doctoral dissertation, INSA Lyon).2019

Website

https://github.com/namebrandon/Sparkov_Data_Generation

https://techpoint.africa/2021/02/22/nigeria-lost-5b-fraud-2020/

https://www.analyticsvidhya.com/blog/2017/03/imbalanced-data-classification/

https://www.credit-connect.co.uk/news/consumer-lending/fraud/over-1-2bn-lost-to-fraud-in-2022/

https://www.fullstackpython.com/twilio.html#:~:text=Twilio%20is%20a%20web%20application,authentication%20into%20their%20Python%20applications.

https://www.kaggle.com/datasets/kartik2112/fraudetection?select=fraudTrain.csv

https://www.paymentsdive.com/news/card-industry-fraud-fighting-efforts-pay-off-nilson-report-credit-debit/639675/

https://www.statista.com/statistics/348004/payment-method-usage-worldwide/

NIBSS Insight, "Fraud in Nigerian Financial Services" 2021 https://nibss-plc.com.ng/media/PDFs/post/NIBSS%20Insights%20Fraud.pdf

Appendices

Appendix I: Design Source Code

```
In [69]: x = np.arange(0,len(df_timeline1),1)
fig, ax = plt.subplots(1,1,figsize=(20,5))
ax.plot(x,df_timeline1['num_of_transactions'])
ax.set_xticks(x)
ax.set_xticklabels(df_timeline1['year_month'])
ax.set_xlabel('Year Month')
ax.set_ylabel('Num of Transactions')
plt.show()
```

```
In [51]: #importing required packages
#modelues for EDA steps
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
#modules for data cleaning and data analysis
from sklearn.model_selection import train_test_split
from sklearn.model_selection import StratifiedKFold
from sklearn.preprocessing import StandardScaler
import scipy.stats as stats
#modules for model building
#algorithms for sampling
from imblearn.under_sampling import RandomUnderSampler
```

```
from imblearn.over_sampling import RandomOverSampler
from imblearn.over_sampling import SMOTE
#baseline linear model
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
#modules for hyper parameter tuning
from sklearn.model_selection import GridSearchCV
#modules for model evaluation
from sklearn.model_selection import cross_val_score
from sklearn import metrics
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.metrics import precision_score, accuracy_score, f1_score, r2_score
from sklearn.metrics import precision_recall_curve, roc_curve
#modules for avoiding warnings
import warnings
warnings.filterwarnings('ignore')
#setting backend for matplotlib
%matplotlib inline
#setting formatting options
pd.options.display.max_columns = 100
pd.options.display.max_rows = 900
pd.set_option('float_format' , '{:f}'.format)
#setting plot style
plt.style.use('seaborn-darkgrid')
```

In [52]:
```
#loading the dataset
df = pd.read_csv('data/fraudTrain.csv')
```

Out[54]:

In [55]:
```
df.columns
```

Out[55]:

In [56]: df.drop('Unnamed: 0', axis=1, inplace=**True**)

In [57]: *# Plot the distribution of each variable*

df.hist(figsize=(20,15))

plt.show()4/18/23, 5:51 PM

fraud_detection

file:///C:/Users/owner/Downloads/fraud_detection.html

4/44

0 1289169

1 7506

Out[59]:

In [60]: *# Feature engineering*

 # Extract useful information from the "trans_date_trans_time" variable

#converting trans_date_trans_time into datetime

df['trans_date_trans_time'] = pd.to_datetime(df['trans_date_trans_time'])

df['trans_datetime'] = pd.to_datetime(df['trans_date_trans_time'])

df['day_of_week'] = df['trans_datetime'].dt.dayofweek

df['hour_of_day'] = df['trans_datetime'].dt.hour

df['trans_year_month'] = df['trans_date_trans_time'].dt.to_period('M')

df['time_since_last_trans'] = df.groupby(['cc_num'])['unix_time'].diff().fillna(0)

:

In [62]: *# Create a new variable that indicates the distance between the customer's location*

df['dist_customer_merchant'] = np.sqrt((df['lat'] - df['merch_lat'])**2 + (df['long

In [13]: *# Create a new variable that indicates the frequency of transactions made by each c*

df['freq_trans_customer_merchant'] = df.groupby(['cc_num', 'merchant'])['trans_num'

 # Create a new variable that indicates the time difference between the current tran

df['time_diff_customer_merchant'] = df.groupby(['cc_num', 'merchant'])['unix_time']

In [63]: *#finding age*

```
#converting 'dob' column to datetime
df['dob'] = pd.to_datetime(df['dob'])
df['age'] = np.round((df['trans_date_trans_time'] - df['dob'])/np.timedelta64(1, 'Y
df.age.head()
Out[63]:
In [64]: #dropping variables
df.drop(['trans_date_trans_time','first', 'last', 'dob'] , axis=1, inplace=True)
df.head()4/18/23, 5:51 PM
Out[66]:4/18/23, 5:51 PM
fraud_detection

plot = [0,0,0]
#plotting the 'trans_hour' feature
plot[0] = sns.countplot(df.hour_of_day, ax = plt.subplot(221))
#plotting the 'trans_day_of_week' feature
plot[1] = sns.countplot(df.day_of_week, ax = plt.subplot(222))
#plotting the 'trans_year_month' feature
plot[2] = sns.countplot(df.trans_year_month, ax = plt.subplot(212))
for i in plot:
i.set_xticklabels(i.get_xticklabels(), rotation=30)
plt.show()
In [68]: #year_month vs number of transactions
df_timeline1 = df.groupby(df['trans_year_month'])[['trans_num','cc_num']].nunique()
df_timeline1.columns = ['year_month','num_of_transactions','customers']
df_timeline
Out[72]:
In [73]: x = np.arange(0,len(df_timeline2),1)
fig, ax = plt.subplots(1,1,figsize=(20,5))
ax.plot(x,df_timeline2['fraud_customers'])
```

```
ax.set_xticks(x)

ax.set_xticklabels(df_timeline2['year_month'])

ax.set_xlabel('Year Month')

ax.set_ylabel('Number of Fraud customers')

plt.show()

In [74]: #creating the 'gender' distributed dataframe

df_gender = df[['gender','trans_num']].groupby(['gender']).count().reset_index()

df_gender.columns = ['Gender', 'gender_count']

#creating gender-fraud distribution

df_fraud_gender = df[['gender','trans_num', 'is_fraud']].groupby(['gender','is_frau

df_fraud_gender.columns = ['Gender', 'is_fraud', 'Transaction Count']

df_fraud_gender = df_fraud_gender.merge(df_gender[['Gender', 'gender_count']],
how=

df_fraud_gender['Transaction percentage'] = (df_fraud_gender['Transaction Count']/d

df_fraud_gender4/18/23, 5:51 PM

In [75]: sns.barplot(data=df_fraud_gender, y='Transaction Count', x='Gender',
hue='is_fraud'

plt.show()

In [76]: #let us first bin the age feature

for i in range(len(df.age)):

if df.age[i] <= 30:

df.age[i] = '< 30'

elif df.age[i] >30 and df.age[i] <= 45:

df.age[i] = '30-45'

elif df.age[i] >45 and df.age[i] <= 60:

df.age[i] = '46-60'

elif df.age[i] >60 and df.age[i] <= 75:

df.age[i] = '61-75'

else:

df.age[i] = '> 75'
```

df.age.head()4/18/23, 5:51 PM

In [77]: *#constructing the age-transaction count distribution*

df_age = df[['age','trans_num']].groupby(['age']).count().reset_index()

df_age.columns = ['age', 'age_count']

#creating the age-fraud distribution

df_fraud_age = df[['age', 'trans_num', 'is_fraud']].groupby(['age','is_fraud']).cou

df_fraud_age.columns = ['age', 'is_fraud', 'Transaction count']

df_fraud_age = df_fraud_age.merge(df_age[['age', 'age_count']], how='inner', on='ag

df_fraud_age['Transaction percentage'] = (df_fraud_age['Transaction count']/df_frau

df_fraud_age

Out[77]:

In [78]: sns.barplot(data=df_fraud_age, y='Transaction count', x='age', hue='is_fraud')

plt.show()4/18/23, 5:51 PM

fraud_detection

file:///C:/Users/owner/Downloads/fraud_detection.html

14/44

In [79]: *#constructing the zip-transaction count distribution*

df_job = df[['job','trans_num']].groupby(['job']).count().reset_index()

df_job.columns = ['job', 'job_count']

#creating the zip-fraud distribution

df_fraud_job = df[['job', 'trans_num', 'is_fraud']].groupby(['job','is_fraud']).cou

df_fraud_job.columns = ['job', 'is_fraud', 'Transaction count']

df_fraud_job = df_fraud_job.merge(df_job[['job', 'job_count']], how='inner', on='jo

df_fraud_job['Transaction percentage'] = (df_fraud_job['Transaction count']/df_frau

#viewing the top 20 jobs with high fraudulent transaction volumes

df_fraud_job[df_fraud_job['is_fraud'] == 1].sort_values(by = ['Transaction percenta4/18/23, 5:51 PM

Name: job, dtype: object

In [81]: *#job with more than one percent fraudulent transactions*

df_fraud_job.loc[(df_fraud_job.is_fraud == 1) &(df_fraud_job['Transaction percenta

Out[81]:4/18/23, 5:51 PM

VA

Index(['cc_num', 'merchant', 'category', 'amt', 'gender', 'street', 'city',

'state', 'zip', 'lat', 'long', 'city_pop', 'job', 'trans_num',

'unix_time', 'merch_lat', 'merch_long', 'is_fraud', 'trans_datetime',

'day_of_week', 'hour_of_day', 'trans_year_month',

'time_since_last_trans', 'dist_customer_merchant', 'age'],

dtype='object')

Out[84]:

In [85]: df.columns

Out[85]:

In [86]: df.drop(['cc_num','street','city','state','job'], axis=1, inplace=**True**)

In [87]: *#let us now check the correlations between the columns*

df_random_under_corr = df.corr()

#plotting the correlation heatplot

plt.figure(figsize=(15,15))

sns.heatmap(df_random_under_corr)

plt.show()4/18/23, 5:51 PM

fraud_detection

file:///C:/Users/owner/Downloads/fraud_detection.html

19/44

In [88]: *#function to return highly correlated column above a threshold*

def correlation(dataset, threshold):

col_corr = set() *# This set stores the highly correlated columns*

corr_matrix = dataset.corr() *#correlation matrix*

#traversing the correlation matrix

for i **in** range(len(corr_matrix.columns)):

for j **in** range(i):

if corr_matrix.iloc[i,j] >threshold:

colname = corr_matrix.columns[i] *#selecting columns above threshold*

col_corr.add(colname) *#adding columns to set*

return col_corr

In [89]: *#let us get the features with correlation above 85%*

corr_features = correlation(df,0.85)

corr_features4/18/23, 5:51 PM

44250.000000

1296675 rows × 8 columns

Out[89]:

In [90]: *#removing unnecessary variables*

df.drop(['zip', 'lat', 'long', 'city_pop', 'unix_time', 'merch_lat','merch_long','m

axis=1, inplace=**True**)

In [91]: df.head()

Out[91]:

In [124…

#split X and Y

X = df.drop(['is_fraud'],axis=1)

y = df.is_fraud

In [125…

X

Out[125]:

In [93]: *#scaling*

scaler = StandardScaler()4/18/23, 5:51 PM

fraud_detection

file:///C:/Users/owner/Downloads/fraud_detection.html

21/44

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1296675 entries, 0 to 1296674

Data columns (total 9 columns):

Column Non-Null Count Dtype

--- ------ -------------- -----

0 category 1296675 non-null int32

1 amt 1296675 non-null float64

2 gender 1296675 non-null int32

3 is_fraud 1296675 non-null int64

4 day_of_week 1296675 non-null int64

5 hour_of_day 1296675 non-null int64

6 time_since_last_trans 1296675 non-null float64

7 dist_customer_merchant 1296675 non-null float64

8 age 1296675 non-null int32

dtypes: float64(3), int32(3), int64(3)

memory usage: 74.2 MB

0 644585

1 3753

Name: is_fraud, dtype: int64

X = scaler.fit_transform(X)

In [94]: df.info()

In [95]: *#train-test split using stratified K fold*

skf = StratifiedKFold(n_splits=2)

skf.get_n_splits(X,y)

for train_index, test_index **in** skf.split(X,y):

X_train, X_test = X[train_index], X[test_index]

y_train, y_test = y[train_index], y[test_index]

y_train.value_counts()

Out[95]:

In [96]: lr = LogisticRegression(random_state=42)

model = lr.fit(X_train, y_train)

y_train_pred = model.predict(X_train)

y_test_pred = model.predict(X_test)

In [97]: *#evaluating the model*

model_name = 'Logistic Regression - imbalance class'

train_score = model.score(X_train,y_train)

test_score = model.score(X_test,y_test)

acc_score = accuracy_score(y_test,y_test_pred)

f_score = f1_score(y_test, y_test_pred, average='weighted')

precision = precision_score(y_test, y_test_pred)

recall = metrics.recall_score(y_test,y_test_pred)

#creating a dataframe to compare the performance of different models

model_eval_data = [[model_name, train_score, test_score, acc_score, f_score, precis

evaluate_df = pd.DataFrame(model_eval_data, columns=['Model Name', 'Training

Out[97]:

In [98]: *#random under sampling using imblearn*

rus = RandomUnderSampler()

X_rus, y_rus = rus.fit_resample(X_train,y_train)

y_rus.value_counts()

Out[98]:

In [99]: X_train, X_test, y_train, y_test = train_test_split(X_rus, y_rus, test_size=0.3, ra

In [100…

#evaluating the model

model_name = 'Logistic Regression - with Random Under Sampling'

train_score = model.score(X_train,y_train)

test_score = model.score(X_test,y_test)

acc_score = accuracy_score(y_test,test_pred)

f_score = f1_score(y_test, test_pred, average='weighted')

precision = precision_score(y_test, test_pred)4/18/23, 5:51 PM

#adding calculations to dataframe

model_eval_data = [model_name, train_score, test_score, acc_score, f_score, precisi

model_eval_dict = {evaluate_df.columns[i]:model_eval_data[i] **for** i **in** range(len(mod

evaluate_df = evaluate_df.append(model_eval_dict, ignore_index=**True**)

evaluate_df

Out[103]:

In [104…

#oversampling with imblearn

ros = RandomOverSampler()

X_ros, y_ros = ros.fit_resample(X_train,y_train)

y_ros.value_counts()

Out[104]:

In [105…

#train Test split

X_train, X_test, y_train, y_test = train_test_split(X_ros,y_ros, test_size=0.3, str

y_train.value_counts()

Out[105]:

In [134…

#implementing logistic regression

lr = LogisticRegression(random_state=42)

#creating model

model = lr.fit(X_train, y_train)

y_train_pred = model.predict(X_train)

y_train_pred

Out[134]:

In [107…

test_pred = model.predict(X_test)

test_pred

Out[107]:

In [108…

#printing classification report

In [109…

#evaluating the model

model_name = 'Logistic Regression - Random Over Sampling'

train_score = model.score(X_train,y_train)

test_score = model.score(X_test,y_test)

acc_score = accuracy_score(y_test,test_pred)

f_score = f1_score(y_test, test_pred, average='weighted')

precision = precision_score(y_test, test_pred)

recall = metrics.recall_score(y_test,test_pred)

#adding claculations to dataframe

model_eval_data = [model_name, train_score, test_score, acc_score, f_score, precisi

model_eval_dict = {evaluate_df.columns[i]:model_eval_data[i] **for** i **in** range(len(mod

evaluate_df = evaluate_df.append(model_eval_dict, ignore_index=**True**)

evaluate_df

Out[109]:

In [110…

#balancing using SMOTE method

smote = SMOTE(sampling_strategy='minority')

X_sm, y_sm = smote.fit_resample(X_train.astype('float'), y_train)

y_sm.value_counts()

Out[110]:

In [111…

#train test split

X_train, X_test, y_train, y_test = train_test_split(X_sm, y_sm, test_size=0.3, rand

y_train.value_counts()4/18/23, 5:51 PM

fraud_detection

file:///C:/Users/owner/Downloads/fraud_detection.html

25/44

0 1287

1 1287

#implementing logistic regression

lr = LogisticRegression(random_state=42)

#creating model

model = lr.fit(X_train, y_train)

y_train_pred = model.predict(X_train)

y_train_pred

Out[112]:

In [116…

model = model.predict(X_test)

model

Out[116]:

In [114…

#printing classification report

print(classification_report(y_test, test_pred))

In [115…

#evaluating the model

model_name = 'Logistic Regression - SMOTE'

train_score = model.score(X_train,y_train)

test_score = model.score(X_test,y_test)

acc_score = accuracy_score(y_test,test_pred)

f_score = f1_score(y_test, test_pred, average='weighted')

precision = precision_score(y_test, test_pred)

recall = metrics.recall_score(y_test,test_pred)

#adding claculations to dataframe

model_eval_data = [model_name, train_score, test_score, acc_score, f_score, precisi

model_eval_dict = {evaluate_df.columns[i]:model_eval_data[i] **for** i **in**
range(len(mod

evaluate_df = evaluate_df.append(model_eval_dict, ignore_index=**True**)

evaluate_df4/18/23, 5:51 PM

In [69]: *#Decisiontree classifier*

#train-test split

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3, random_stat

dtree = DecisionTreeClassifier(max_depth=10)

model = dtree.fit(X_train,y_train)

y_test_pred = model.predict(X_test)

print(classification_report(y_test, y_test_pred))

#evaluating the model

model_name = 'Decision Tree - imbalance class'

train_score = model.score(X_train,y_train)

test_score = model.score(X_test,y_test)

acc_score = accuracy_score(y_test,y_test_pred)

f_score = f1_score(y_test, y_test_pred, average='weighted')

precision = precision_score(y_test, y_test_pred)

recall = metrics.recall_score(y_test,y_test_pred)

#adding claculations to dataframe

model_eval_data = [model_name, train_score, test_score, acc_score, f_score, precisi

model_eval_dict = {evaluate_df.columns[i]:model_eval_data[i] **for** i **in**
range(len(mod

evaluate_df = evaluate_df.append(model_eval_dict, ignore_index=**True**)

evaluate_df4/18/23, 5:51 PM

In [70]: *#train-test split*

X_train, X_test, y_train, y_test = train_test_split(X_rus,y_rus, test_size=0.3, ran

dtree = DecisionTreeClassifier(max_depth=10)

model = dtree.fit(X_train,y_train)

test_pred = model.predict(X_test)

print(classification_report(y_test, test_pred))

#evaluating the model

model_name = 'Decision Tree - Random Under Sampling'

train_score = model.score(X_train,y_train)

```
test_score = model.score(X_test,y_test)

acc_score = accuracy_score(y_test,test_pred)

f_score = f1_score(y_test, test_pred, average='weighted')

precision = precision_score(y_test, test_pred)

recall = metrics.recall_score(y_test,test_pred)

#adding claculations to dataframe

model_eval_data = [model_name, train_score, test_score, acc_score, f_score, precisi

model_eval_dict = {evaluate_df.columns[i]:model_eval_data[i] for i in
range(len(mod

evaluate_df = evaluate_df.append(model_eval_dict, ignore_index=True)

evaluate_df4/18/23, 5:51 PM

In [71]: #train-test split

X_train, X_test, y_train, y_test = train_test_split(X_ros,y_ros, test_size=0.3, ran

dtree = DecisionTreeClassifier(max_depth=10)

model = dtree.fit(X_train,y_train)

test_pred = model.predict(X_test)

print(classification_report(y_test, test_pred))

#evaluating the model

model_name = 'Decision Tree - Random Over Sampling'

train_score = model.score(X_train,y_train)

test_score = model.score(X_test,y_test)

acc_score = accuracy_score(y_test,test_pred)

f_score = f1_score(y_test, test_pred, average='weighted')

precision = precision_score(y_test, test_pred)

recall = metrics.recall_score(y_test,test_pred)

#adding claculations to dataframe

model_eval_data = [model_name, train_score, test_score, acc_score, f_score, precisi

model_eval_dict = {evaluate_df.columns[i]:model_eval_data[i] for i in
range(len(mod

evaluate_df = evaluate_df.append(model_eval_dict, ignore_index=True)
```

```
evaluate_df4/18/23, 5:51 PM

#evaluating the model

model_name = 'Decision Tree - SMOTE'

train_score = model.score(X_train,y_train)

test_score = model.score(X_test,y_test)

acc_score = accuracy_score(y_test,test_pred)

f_score = f1_score(y_test, test_pred, average='weighted')

precision = precision_score(y_test, test_pred)4/18/23, 5:51 PM

model_eval_data = [model_name, train_score, test_score, acc_score, f_score, precisi

model_eval_dict = {evaluate_df.columns[i]:model_eval_data[i] for i in
range(len(mod

evaluate_df = evaluate_df.append(model_eval_dict, ignore_index=True)

evaluate_df

Out[72]:

In [73]: #train-test split

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3, random_stat

rf = RandomForestClassifier(n_estimators=100, criterion='gini')

model = rf.fit(X_train,y_train)

y_test_pred = model.predict(X_test)

print(classification_report(y_test, y_test_pred))4/18/23, 5:51 PM

#evaluating the model

model_name = 'Random Forest - imbalance class'

train_score = model.score(X_train,y_train)

test_score = model.score(X_test,y_test)

acc_score = accuracy_score(y_test,y_test_pred)

f_score = f1_score(y_test, y_test_pred, average='weighted')

precision = precision_score(y_test, y_test_pred)

recall = metrics.recall_score(y_test,y_test_pred)

#adding claculations to dataframe

model_eval_data = [model_name, train_score, test_score, acc_score, f_score, precisi
```

```
model_eval_dict = {evaluate_df.columns[i]:model_eval_data[i] for i in
range(len(mod

evaluate_df = evaluate_df.append(model_eval_dict, ignore_index=True)

evaluate_df4/18/23, 5:51 PM

#evaluating the model

model_name = 'Random Forest - Random Under Sampling'

train_score = model.score(X_train,y_train)

test_score = model.score(X_test,y_test)

acc_score = accuracy_score(y_test,test_pred)

f_score = f1_score(y_test, test_pred, average='weighted')

precision = precision_score(y_test, test_pred)

recall = metrics.recall_score(y_test,test_pred)

#adding claculations to dataframe

model_eval_data = [model_name, train_score, test_score, acc_score, f_score, precisi

model_eval_dict = {evaluate_df.columns[i]:model_eval_data[i] for i in
range(len(mod

evaluate_df = evaluate_df.append(model_eval_dict, ignore_index=True)

#evaluating the model

model_name = 'Random Forest - Random Over Sampling'

train_score = model.score(X_train,y_train)

test_score = model.score(X_test,y_test)

acc_score = accuracy_score(y_test,test_pred)

f_score = f1_score(y_test, test_pred, average='weighted')

precision = precision_score(y_test, test_pred)

recall = metrics.recall_score(y_test,test_pred)

#adding claculations to dataframe

model_eval_data = [model_name, train_score, test_score, acc_score, f_score, precisi

model_eval_dict = {evaluate_df.columns[i]:model_eval_data[i] for i in
range(len(mod

In [79]: #train-test split

X_train, X_test, y_train, y_test = train_test_split(X_sm,y_sm, test_size=0.3, rando
```

```
rf = RandomForestClassifier(n_estimators=100, criterion='gini')

model = rf.fit(X_train,y_train)

test_pred = model.predict(X_test)

print(classification_report(y_test, test_pred))

#evaluating the model

model_name = 'Random Forest - SMOTE'

train_score = model.score(X_train,y_train)

test_score = model.score(X_test,y_test)4/18/23, 5:51 PM

acc_score = accuracy_score(y_test,test_pred)

f_score = f1_score(y_test, test_pred, average='weighted')

precision = precision_score(y_test, test_pred)

recall = metrics.recall_score(y_test,test_pred)

#adding claculations to dataframe

model_eval_data = [model_name, train_score, test_score, acc_score, f_score, precisi

model_eval_dict = {evaluate_df.columns[i]:model_eval_data[i] for i in
range(len(mod

evaluate_df = evaluate_df.append(model_eval_dict, ignore_index=True)

evaluate_df4/18/23, 5:51 PM

In [80]: #train-test split

model_name = 'Random Forest - SMOTE'

train_score = model.score(X_train,y_train)

test_score = model.score(X_test,y_test)

acc_score = accuracy_score(y_test,test_pred)

f_score = f1_score(y_test, test_pred, average='weighted')

precision = precision_score(y_test, test_pred)

recall = metrics.recall_score(y_test,test_pred)

#adding claculations to dataframe

model_eval_data = [model_name, train_score, test_score, acc_score, f_score, precisi

model_eval_dict = {evaluate_df.columns[i]:model_eval_data[i] for i in
range(len(mod
```

evaluate_df = evaluate_df.append(model_eval_dict, ignore_index=**True**)

In [81]: best_grid = RandomForestClassifier(max_features = 'sqrt', n_estimators=200, random_

#train-test split

X_train, X_test, y_train, y_test = train_test_split(X_sm,y_sm, test_size=0.3, rando

model = best_grid.fit(X_train,y_train)

test_pred = model.predict(X_test)

print(classification_report(y_test, test_pred))

#evaluating the model

model_name = 'Random Forest - SMOTE [Hyperparameter Tuned]'

train_score = model.score(X_train,y_train)

test_score = model.score(X_test,y_test)

acc_score = accuracy_score(y_test,test_pred)

f_score = f1_score(y_test, test_pred, average='weighted')

precision = precision_score(y_test, test_pred)

recall = metrics.recall_score(y_test,test_pred)

#adding claculations to dataframe

model_eval_data = [model_name, train_score, test_score, acc_score, f_score, precisi

model_eval_dict = {evaluate_df.columns[i]:model_eval_data[i] **for** i **in** range(len(mod

evaluate_df = evaluate_df.append(model_eval_dict, ignore_index=**True**)

Out[119]:4/18/23, 5:51 PM

fraud_detection

file:///C:/Users/owner/Downloads/fraud_detection.html

42/44

Collecting twilio

Downloading twilio-8.0.0-py2.py3-none-any.whl (1.7 MB)

---------- 1.7/1.7 MB 366.0 kB/s eta 0:00:00

Requirement already satisfied: pytz in c:\users\owner\anaconda3\lib\site-packages (f

rom twilio) (2022.1)

Requirement already satisfied: requests>=2.0.0 in c:\users\owner\anaconda3\lib\site

packages (from twilio) (2.28.1)

Collecting aiohttp>=3.8.4

Downloading aiohttp-3.8.4-cp39-cp39-win_amd64.whl (323 kB)

---------- 323.6/323.6 kB 911.0 kB/s eta 0:00:00

Collecting aiohttp-retry>=2.8.3

Downloading aiohttp_retry-2.8.3-py3-none-any.whl (9.8 kB)

Collecting asyncio>=3.4.3

Downloading asyncio-3.4.3-py3-none-any.whl (101 kB)

---------- 101.8/101.8 kB 532.1 kB/s eta 0:00:00

Requirement already satisfied: PyJWT<3.0.0,>=2.0.0 in
c:\users\owner\anaconda3\lib\s

ite-packages (from twilio) (2.4.0)

Requirement already satisfied: attrs>=17.3.0 in c:\users\owner\anaconda3\lib\site-pa

ckages (from aiohttp>=3.8.4->twilio) (21.4.0)

Requirement already satisfied: multidict<7.0,>=4.5 in c:\users\owner\anaconda3\lib\s

ite-packages (from aiohttp>=3.8.4->twilio) (6.0.4)

Requirement already satisfied: yarl<2.0,>=1.0 in c:\users\owner\anaconda3\lib\site-p

ackages (from aiohttp>=3.8.4->twilio) (1.8.2)

Requirement already satisfied: aiosignal>=1.1.2 in c:\users\owner\anaconda3\lib\site

-packages (from aiohttp>=3.8.4->twilio) (1.3.1)

Requirement already satisfied: frozenlist>=1.1.1 in c:\users\owner\anaconda3\lib\sit

e-packages (from aiohttp>=3.8.4->twilio) (1.3.3)

Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in
c:\users\owner\anacond

a3\lib\site-packages (from aiohttp>=3.8.4->twilio) (4.0.2)

Requirement already satisfied: charset-normalizer<4.0,>=2.0 in
c:\users\owner\anacon

da3\lib\site-packages (from aiohttp>=3.8.4->twilio) (2.0.4)

Requirement already satisfied: certifi>=2017.4.17 in c:\users\owner\anaconda3\lib\si

te-packages (from requests>=2.0.0->twilio) (2022.9.14)

Requirement already satisfied: idna<4,>=2.5 in c:\users\owner\anaconda3\lib\site-pac
kages (from requests>=2.0.0->twilio) (3.3)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\owner\anaconda3\lib
\site-packages (from requests>=2.0.0->twilio) (1.26.11)
Installing collected packages: asyncio, aiohttp, aiohttp-retry, twilio
Attempting uninstall: aiohttp
Found existing installation: aiohttp 3.8.3
Uninstalling aiohttp-3.8.3:
Successfully uninstalled aiohttp-3.8.3
Successfully installed aiohttp-3.8.4 aiohttp-retry-2.8.3 asyncio-3.4.3 twilio-8.0.0
Index(['category', 'amt', 'gender', 'is_fraud', 'day_of_week', 'hour_of_day',
'time_since_last_trans', 'dist_customer_merchant', 'age'],
dtype='object')
In [121…
!pip install twilio
In [122…
from twilio.rest **import** Client
In [132…
df**.**columns
Out[132]:
In [ ]: *# Send a text message alert if fraud is detected*
auth_token = ''
client = Client(account_sid, auth_token)
for index, transaction **in** X**.**iterrows():
 # Extract relevant features from the transaction data
transaction_data = np**.**array([[transaction['category'], transaction['gender'], t
transaction['hour_of_day'],transaction['time_since
 # Make a prediction using the trained model
prediction = model**.**predict(transaction_data)

 # If fraud is detected, send a text message alert

Bio data

A. Personal Data

1. **Full Name:** Ahmed Oluwatoyin**, JOLAOSHO**
2. **Date and Place of Birth:** 15th March 1986.
3. **Nationality:** Nigerian
4. **Marital Status:** Married
5. **Place of Birth:** Abeokuta
6. **Local Govt. Area:** Abeokuta North
7. **State of Origin:** Ogun
8. **Permanent Address:** Green lane, New vista estate, Moganno Elebu off akala Express way Ibadan
9. **Email:** jolaosho1000@yahoo.com
10. **Department:** Computer Science

B. Educational Background

Educational Institutions Attended with Dates and Qualification:

i. Christ church primary school, Ilaro 1992-1998

ii. Adokun High School 2010-2015

iii. Moshood Abiola Polytechnic, Abeokuta 2005-2011

iv. Crescent University, Abeokuta 2015-2017

v. Lead City University, Ibadan 2021-Till date

Educational Qualification with Dates:

i. First school leaving certificate 1998

ii. Senior school certificate examination (SSCE) 2004

iii. ND Electrical/Electronic Engineering 2007

iv. HND Electrical/Electronic Engineering 2011

v. Bs.c Computer science 2017

vi. Ms.c Computer Science in-view

C. Work Experience: With Dates

Gods will school Ondo state 2012

Albert Metro Consultancy Limited 2014

Obasanjo Farm Nigeria Limited 2015 till date

D. Names and Addresses of Referees

Prof. A. Akinola

Senior Lecturer

Lead City University, Ibadan

Department of Computer science

Solom202@yahoo.co.uk

Engr. (Dr) Olujide A. Adeniran

HOD Computer Science

Moshood Abiola Polytechnic, Abeokuta

adenekanolujide@gmail.com

Dr. Ismail Olalekan Lasisi

HOD Computer science

olassbaba@gmail.com

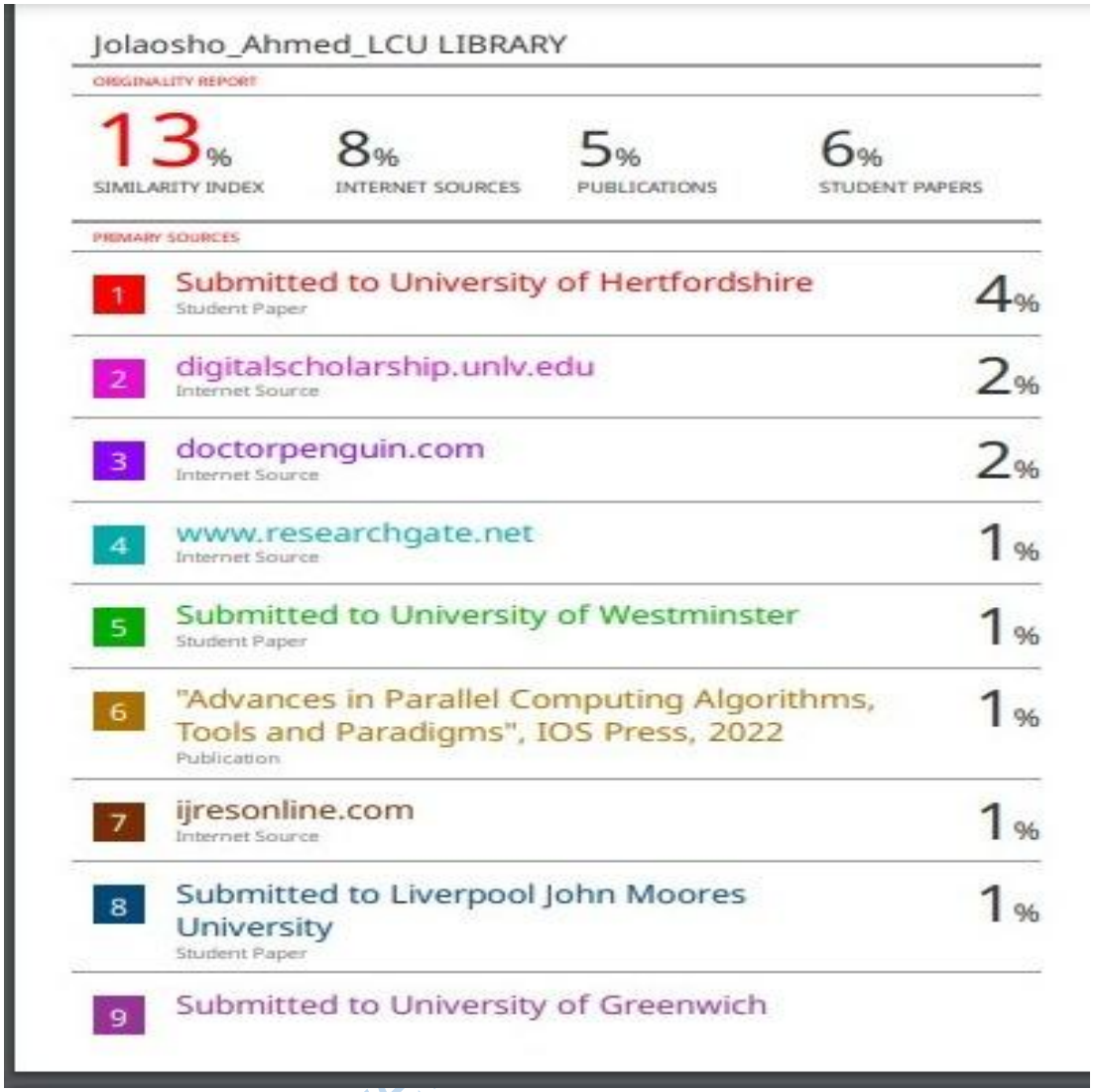
Crescent University, Abeokuta

The University Compliance Certification

This is to certify that this thesis by Ahmed Oluwatoyin Jolaosho with Matriculation Number LCU/PG/002404 in the Department of Computer Science, Faculty of Natural and Applied Sciences, Lead City University, Ibadan is in full compliance with the approval of the University's format and style.

…… ……..

Signature Date



Jolaosho_Ahmed_LCU LIBRARY

ORIGINALITY REPORT

13%	8%	5%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Hertfordshire Student Paper	4%
2	digitalscholarship.unlv.edu Internet Source	2%
3	doctorpenguin.com Internet Source	2%
4	www.researchgate.net Internet Source	1%
5	Submitted to University of Westminster Student Paper	1%
6	"Advances in Parallel Computing Algorithms, Tools and Paradigms", IOS Press, 2022 Publication	1%
7	ijresonline.com Internet Source	1%
8	Submitted to Liverpool John Moores University Student Paper	1%
9	Submitted to University of Greenwich	