AI-POWERED FRAUD DETECTION & PREVENTION IN FINANCIAL INSTITUTIONS

BOAS SINGOGO

A Final Year Research Project submitted in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science

ZCAS University

2025

DECLARATION

Name: Boas Singogo

Student Number: 202303886

I hereby declare that this final year research project is the result of my own work, except for quotations and summaries which have been duly acknowledged.

Plagiarism check: 15% out of which 5% culminates from the cover page, declaration, dedication and table of contents.

Signature:  Date: 27th June 2025.

Supervisor Name: Dr Bob Jere

Supervisor Signature: 

Date: 30th June, 2025

AI-POWERED FRAUD DETECTION & PREVENTION IN FINANCIAL INSTITUTIONS

ABSTRACT

This research project investigates the development and implementation of AI-Powered Fraud Detection and Prevention in financial institutions. By using artificial intelligence and machine learning techniques like supervised learning, the proposed model aims to improve the accuracy, efficiency, and real-time capability of fraud detection. The study addresses key challenges faced by traditional methods such as high positive rates and failure to adapt to changing fraud tactics by developing a scalable and resource-efficient system. As emphasis is put on addressing data imbalance, ethical compliance, and user-friendly interfaces, this work contributes to improving risk management, reducing financial loses and increasing customer trust. The research findings show the potential of AI to revolutionize fraud detection, offering robust solutions aligned with regulatory standards and industry best practices.

Keywords: *Fraud Detection, Artificial-Intelligence, Supervised learning, Machine-Learning, Deep-Learning, Natural-Language- Processing, Risk management.*

ACKNOWLEDGEMENT

I would like to take this opportunity to express my gratitude and appreciation to my supervisor, **Dr Ezekiel Bob Jere** for his guidance, patience and invaluable advice throughout this project.

I would also like to express my appreciation to the entire ZCAS University Faculty for generously sharing your knowledge. Your lectures and advice greatly contributed to the success of the project.

THANK YOU.

DEDICATION

I would like to take this opportunity to express my gratitude and appreciation to my family, throughout my studies you gave me the unwavering support I needed to attend lectures and study. To my wife Emmy Tembo Singogo, thank you for your support and understanding my absence when you needed my attention. I dedicate this work to My Son Andrew Singogo, my daughters Natasha Singogo and Elida Singogo as your presence constantly remind me to continue working hard.

TABLE OF CONTENTS

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI…….. Artificial Intelligence

AML…… Anti-Money Laundering

API…… Application Programming Interface

AUPRC…… Area Under the Precision-Recall Curve

BNPL……. Buy Now, Pay Later

CCFD…… Credit Card Fraud Detection

CNN…… Convolutional Neural Networks

Fintech……. Financial Technology

GDPR…… General Data Protection Regulation

ID……..……. Identity

IP……. Internet Protocol

KYC…… Know Your Customer

LR……..……. Logistic Regression

LSTM……... Long Short-Term Memory

ML……. Machine Learning

PCA…… Principle Component Analysis

PCI DSS…….. Payment Card Industry Data Security Standard

RF…….……. Random Forest

ROC-AUC……... Receiver Operating Characteristic – Area Under the Curve

SMOTE…… Synthetic Minority Over-sampling Technique

CHAPTER 1 – INTRODUCTION

1.0 Overview

In today's fast changing financial environment, fraud has become a serious risk to the integrity and stability of financial institutions. The complexity of fraudulent activities have increased, making Conventional detection methods inadequate[1]. Financial fraud can result into major financial losses, undermine consumer trust, along with possing significant risks to the overall economic system of a country. This requires the development of advanced and robust systems to detect and prevent fraudulent activities effectively.

Fraud detection systems in financial services are known to operate on predefined rules and historical patterns which are perceived to be effective only to some extent, and fail to adapt with the changing approach of fraudsters. New types of fraud emerge regularly,and these Conventional methods often fall short in adapting to the changes quickly enough[2].

The emergency of artificial intelligence (AI) and machine learning (ML) innovations presents positive solutions to address the challenges of fraud detection. AI-based fraud detection systems can examine substantial amounts of transaction data, recognize patterns, and detect irregularities with high precision and speed. These systems can adjust to new and changing fraud tactics, offering a dynamic and proactive approach to fraud prevention[3].

For instance, as submitted by Najif [4], if an AI system detects an unusual pattern, whereby an account that was previously inactive suddenly experiences a large number of transactions or there is an unexpected surge in transactions from a foreign location, the system can promptly notify the institution’s fraud detection team. This allows for a quick response, often preventing significant damage before the fraudulent activity escalates.

As revealed in “AI-Driven Approaches for Real-Time Fraud Detection in US Financial Transactions: Challenges and Opportunities,”[5], incorporating AI in real-time detection of fraud systems marks a major step forward in the fight against financial fraud. With the increasing complexity and volume of financial transactions, AI-powered solutions will be vital in ensuring the security and reliability of the financial system.

This research project aims to review the current state of fraud detection systems in financial organizations, design and implement an AI-powered fraud detection and prevention model that can prevent fraudulent transactions in real-time. By employing advanced AI and ML techniques, the model will improve the accuracy and efficiency in detection of fraud, reduce false positives, and provide real-time monitoring and alerts. The project will also focus on creating a user-friendly interface for testing and ensure seamless adoption and usability by financial institutions.

The subsequent sections outline the objectives, methodology, and expected outcomes. The goal is to contribute to the development of a cutting-edge solution that addresses the pressing need for effective fraud detection and prevention in the financial sector.

1.1 Background to the Study

Fraud detection has been a critical concern in financial institutions for decades. Conventional methods of identifying fraud have relied on rule-based systems, manual audits, and statistical techniques. These methods aimed to identify activities related to fraud such as money laundering, identity theft, and transaction fraud.

One of the earliest approaches to fraud detection was manual review, where financial analysts examined transactions and account activities for irregularities. This method, however, was labor-intensive and vulnerable to human error, making it unsuitable for handling large volumes of financial data.

As technology advanced, rule-based systems became a standard approach. These systems depend on predefined rules that are set by financial experts to flag suspicious transactions. For example, a rule might activate an alert if a transaction is above a certain amount or occurs in a high-risk location. While rule-based systems improved efficiency, they had drawbacks, including high false-positive rates and an inability to adapt to arising fraud patterns.

Statistical methods, such as regression analysis and anomaly detection, were also employed to enhance fraud detection. These techniques used historical data to identify differences from normal transaction behavior. However, statistical models required constant updates and were limited in detecting sophisticated fraud schemes that evolved over time.

Despite their effectiveness to some extent, traditional fraud detection methods struggled with scalability, adaptability, and accuracy in detecting potential fraud. With an increase in digital transactions and more complex fraud tactics, financial organizations are progressively adopting advanced machine learning and artificial intelligence innovations to improve fraud detection capabilities.

1.2 Problem Statement

The existing fraud detection systems in banks and financial institutions may not effectively adapt to evolving fraud techniques and patterns. Conventional methods of detecting fraud, which depend heavily on rule-based systems and manual reviews, are becoming increasingly poor in identifying and preventing sophisticated fraudulent activities. These conventional methods often fail to keep-up with the fast changing tactics employed by fraudsters, leading to huge financial losses, compromised customer trust, and heightened operational risks.

To mitigate these challenges, this research project proposes the design and implementation of an AI-based fraud detection and prevention model that will be able to flag and stop fraudulent transactions as they occur. By exploiting machine learning algorithms and artificial intelligence, the proposed system aims to enhance the accuracy and speed of fraud detection, adapt to new and evolving fraud patterns, and provide real-time alerts. This system will help financial organizations reduce the risks associated with online transactions, point of sale, and credit/debit card related fraud, safeguard their assets and customers, and conserve trust and integrity in their operations.

1.3 Aim and Objectives of the Study

1.3.1 Aim

- The aim of this research is to investigate the current state of fraud detection systems in financial institutions, and to explore the application of AI to enhance the accuracy and efficiency of Debit and Credit card fraud detection and prevention in financial institutions.

1.3.2 Objectives of the Study

The objectives of the study are to:

a) develop and implement a machine learning model that can detect and prevent fraudulent transactions with at least 90% accuracy.

b) design and implement a user-friendly interface to interact with and test the fraud detection system.

c) evaluate the system's performance in terms of accuracy, precision, and recall.

1.4 Research Questions

The following research questions were used to interrogate the articles under review:

i. What are the current methods and technologies used for fraud detection in financial institutions?
ii. What are the key challenges and limitations in current fraud detection systems?
iii. How have advancements in artificial intelligence and machine learning improved fraud detection capabilities?

1.5 Scope and Limitation

1.5.1 Scope

The scope involves collection of the dataset that will be used to train the model. This will be followed by the developing a supervised machine learning model for detecting transactions that are fraudulent. To make interaction with the model possible and allow for practical testing, an API will be created. Lastly, the model will be thoroughly tested to evaluate its accuracy and performance.

1.5.2 Limitations:

Feature Availability: The features used in the study are limited to those present in the dataset. Important real-world features such as device ID, IP address, user biometrics, or user history might not be included.

Latency and Real-time Constraints: The model is tested in an offline environment. Issues related to real-time detection speed, system integration, stress testing, and response time are not covered in this study.

1.6 Significant of the Study

This research is significant because it contributes to the enhancement of fraud detection accuracy using advanced AI and machine learning algorithms, providing timely detection and prevention of fraudulent transactions. The scalable and efficient system reduces false positives and resource requirements, adapts to evolving fraud patterns, and improves customer trust by ensuring a secure banking experience. Additionally, it aligns with regulatory standards, reduces financial losses, and contributes to both academic research and industry best practices, ultimately benefiting financial institutions, their customers, and the overall economic landscape.

1.7 Preliminary Sections of the Project Report

Apart from this introductory chapter, this paper is made up of the following chapters:

CHAPTER 2 - LITERATURE REVIEW : In this section a broad Literature review of the research topic is done and followed by a critical review of related works. Comparison with related works is also undertaken and gaps identified. This is followed by the conceptual, Proposed model and Chapter Summary.

CHAPTER 3 - METHODOLOGY : The methodology section highlights the research design, adopted method and justification. The details about the source of research data and datasets are given, addresses data collection methods and data analysis techniques. Lastly it speaks to Ethical concerns related to the research and concludes with Chapter Summary.

CHAPTER 4 - DATA, EXPERIMENTS, AND IMPLEMENTATION : This chapter explores appropriate modelling in relation to the project. It highlights techniques, algorithms, mechanisms used to design the model, and explains the main functions of models used. Lastly, it delves into experiments and implementations and concludes with Chapter Summary.

CHAPTER 5 - RESULTS AND DISCUSSIONS: In this chapter, results of the research findings are presentation and Analysis of Results/Performance Metrics is done. Comparison to Related Work is undertaken, Implications of Results is discussed, and ends with Chapter Summary.

CHAPTER 6 - SUMMARY AND CONCLUSION: This section highlights the summary of main Findings, Academic contribution to the body of knowledge/Novelty, Limitations of the Research, and Future works.

Chapter Summary

Fraud has become a serious threat to financial institutions, with increasingly sophisticated techniques making traditional detection methods inadequate. Financial fraud leads to substantial financial losses, erodes consumer trust, and poses risks to economic stability. Conventional methods of detecting fraud, including rule-based systems and manual reviews, fail to adjust to changing fraud tactics. The development of artificial intelligence (AI) and machine learning (ML) provide a more effective solution, enabling real-time detection of fraud through advanced data analysis and anomaly detection.

In the past, fraud detection in financial institutions depended on manual audits, rule-based systems, and statistical models. While these methods provided some level of fraud prevention, they suffered from inefficiencies, high false-positive rates, and limited adaptability. The need for scalable and dynamic solutions has led to the adoption of AI and ML, which offer improved detection of fraud by identifying complex patterns and adapting to emerging threats. The next chapter considers the Literature Review related to the research of this paper.

CHAPTER 2 - LITERATURE REVIEW

2.0 Broad Literature Review of the Topic

As discussed by Philip Olaseni Shoetan and Babajide Tolulope Familoni[6], Financial institutions employ a variety of methods and technologies for fraud detection. The authors addressesed a critical and timely issue in the Financial and Technology (fintech) industry, where advanced technologies are increasingly being targeted by fraud. The authors submitted that emphasis on AI technologies stresses the importance of improving fraud detection mechanisms, which is pertinent given the rapid digital transformation in finance. However, the article did not provide clear directions for future research to set the stage for upcoming studies in this domain. Authors would have identified particular gaps in the existing research and suggested specific areas for additional study, this would have provided a useful road map for academics and practitioners alike.

Scholars, on the other hand have noted that despite these developments in the application of AI in the detection of fraud, maintenance of fraud detection system is difficult because financial fraud is dynamic in nature and requires constant learning and adaptation by AI systems to new data and patterns. They have also argued that the implementation and maintenance of models for fraud detection can be resource-intensive, as it requires continuous expenditure for data collection, model retraining, and system updates to guarantee systems efficiency. While the scholars addressed the importance of accountability and transparency in AI algorithms, which are essential for their ethical application, they also brought up the "black-box" nature of some AI models, more especially deep learning, which makes interpretability very complicated. This presents a contradiction, as the call for transparency clashes with the operational complexities that AI models entail, potentially making it difficult to maintain ethical standards.

Further research published by Mr. Yash Prajapati in an article titled "An Analysis of Financial Fraud Detection Methods Using Artificial Intelligence" [7] revealed that there are challenges involved with AI-based fraud protection such as requirement for large amounts of training data, the possibility of bias in the models, the models' lack of interpretability, and the possibility of conflicting results. However, the researcher argues that the utilization of AI in fraud protection offers many advantages, for example better accuracy, fewer false positives, and faster fraud detection. Although the paper brings out the possibility of bias in AI models, it fails to investigate

the strategies for alleviating these biases or improving model interpretability. Given the fact that financial decisions can have significant implications, understanding how models make their predictions is significant. The author's focus on ethical considerations in AI and implications of biased outcomes would be a valuable addition.

Moreover, the researcher,[8] had submitted that AI technology has changed the way fraud is detected in the financial sector by detecting fraud early, improving accuracy, enhancing speed, providing adaptive learning capabilities, offering cost efficiency, and ensuring scalability. The researcher submitted that AI algorithms can examine large volumes of data in real-time, flag abnormality in patterns, and adjust to new fraud techniques, which leads to a proactive fraud prevention, reduced false positives, and rapid response to fraudulent activities. On the other hand, the paper mentions the Deanship of Scientific Research at Majmaah University to be the source of funding and yet it does not discuss how this may influence the study's outcomes or conclusions. If the authors had discussed the potential biases related to the source of funding, they would probably had provided a measure of transparency.

Additionally, an Expert in the Banking sector and Financial Risk Management [9] submitted that an adoption of AI technologies in fraud detection and AML techniques presents an opportunity for countries to improve their financial security and regulatory compliance, while protecting their economies from the detrimental effects of financial crimes. The expert argued that by using the power of AI, many countries may build durable and effective systems to tackle fraud and money laundering in the financial sector. Overall, the article submits its findings in a concise and structured manner, strengthening readability and understanding. However, some sections could bring out more benefits from detailed explanations regarding complex financial concepts related to anti-money laundering measures in particularly. A more detailed introduction outlining the importance of the topic would have helped in understanding the discussion for readers less familiar with the subject.

The paper "Open AI and its Impact on Fraud Detection in Financial Industry," [10] revealed that Fraud significantly impacts financial institutions engaged in online payment services, especially with the advancements in technology. The researcher submitted that nearly 20% of customers switch financial institutions after experiencing fraud. This defection to rival institutions leads to financial losses and tarnished reputations for the victimized institution, particularly if the trend persists. According to the researcher, it is very important for financial institutions to deploy strong

fraud detection techniques within their systems such as the use of AI technology. The paper effectively exploited statistical data, such as the Nilson report by citing an increment of 14% in card fraud related losses, to stress the importance of implementing better fraud detection techniques

As submitted by Carter et al,[11] in an era of bulk and complex data, traditional methods of risk analysis and financial decision-making is not enough. The authors revealed that Artificial Intelligence (AI) undeniably provides financial institutions with the needed tools to increase precision and efficiency by processing large data sets and uncovering complicated patterns. They further argued that, Machine learning algorithms offer improved accuracy in fraud detection as compared to conventional techniques. In their opinion, application of natural language processing, and unstructured data from various sources such as news articles and social platform can be analyzed to gain deeper understanding into emerging risks, which may result into a proactive risk management by putting emphasis on potential issues before they manifest.

The paper had covered many aspects of AI integration in financial organizations, such as risk management, detection of fraud and regulatory compliance. This wide coverage makes it a valuable resource for understanding the current landscape of AI in finance. While the paper highlights a good overview, it could make more impact by explaining the techniques of AI technologies mentioned, such as specifying algorithms or models employed in financial applications. This lack of detail might however, leave technically-inclined readers wanting more.

As subnitted by B. Mytnyk, O. Tkachyk, N. Shakhovska, S. Fedushko, and Y. Syerov [12] following the COVID-19 outbreak there has been an increase in fraudulent activities related to bank transactions due to the huge shift towards the online transactions. The researchers submitted that many varieties of banking fraud include wire fund, identity theft, compromised account, laundering of money, and accounting fraud. They suggested that to effectively flag fraudulent transactions, the supervised machine learning algorithm need access to large volumes of historical database in relation to such activities and use binary logic along with preprocessing techniques. The article thoroughly covers various types of fraudulent banking practices, including phishing scams, identity theft, money laundering, and account takeover fraud. Each type is briefly defined and explained, this provides a solid foundation for understanding the complexities involved. However, while the definitions are enough, the depth of analysis with regard the implications of

these practices on financial institutions and customers is somewhat missing. A more in-depth research on the long-term effects on victims and the wider economic impact would strengthen the article.

Further submissions from [13] stated that AI is the leading innovation imbedded with security and fraud detection techniques. It can examine an individual's pattern of spending and behavior towards different transactions and from which it would flag unusual spending activities. This old behavior may be utilizing of a card from one country and just a few hours later that card is used elsewhere in another country, or it may be an effort to take out the money suspiciously from an account. The article succeeds in addressing its targeted audience and presents insights that could be beneficial for stakeholders in the respective fields. However, there is room for improvement in ensuring that the arguments made resonate more powerfully with the reader by focusing on implications and future directions.

Another scholar [14] emphasized that as technology continues to change, so do the difficulties presented by payment fraud attacks. When fraudster establish a digital mark or pattern that makes the attacks not detectable, the innovative use of predictive models emphasize the value of AI as it assists in reduction of these attacks by providing an additional layer of security to the bank. According to the researcher, AI's ability to quickly detect large-scale fraudulent payments makes it a valuable asset for banks in handling such cases.

In as much as the article presents numerous advantages of AI in banking, it falls short in tackling potential drawbacks or challenges that come with deployment of these technologies. For example, the article brings out the challenges faced by USAA regarding data access and internal buy-in, but it does not explain in detail the effect of AI on the workforce in the sector or the ethical concerns associated with data privacy and cybersecurity risks. These aspects are very important, especially that the banking industry faces audits over how it handles personal data.

The article "Artificial Intelligence (AI) in Financial Sectors: Blessings or Threats?" by Rakib Ahmed[15] noted with concern that E-commerce deception and online fraud on the use of VISA, credit, debit and Master card is very difficult to avoid and proposed the use of AI in scam recognition as an effective means to detect and prevent fraud. The article discusses the importance and influence of AI in the banking and financial services sector. It explains the transition towards an automated and technology-driven global society, putting emphasis on various uses of AI, such as improving efficiency, reducing costs and providing faster customer assistance. The research is

based on secondary sources, emphasizing the potential benefits of AI in areas like detection of fraud, reducing risks, and improving customer service through chatbots. By referencing AI utilization in both Bangladesh and worldwide, the article adds to the understanding of AI's impact across different markets and economies, showing its global importance. However, the fact that the study is based on the secondary sources, it limits integrity of the analysis. Integrating primary research, such as interviews with industry professionals or surveys of financial institutions, would have given a strong foundation for the conclusions drawn.

Gautam .A [16] submitted that while Enhanced Fraud Detection Algorithms, empowered by AI, provides a very powerful weapon in the financial industry's ongoing battle against fraudulent activities, It presents challenges associated to data privacy and security. The researcher argued that the use of artificial intelligence in banking mostly requires access to large collection of sensitive customer information, which may include financial transactions, personal identification data, and behavioral patterns. It is therefore important for the financial institutions to apply strong measures to safeguard the privacy and security of the information and ensure compliance to data protection regulations and standards. All in all, the article provides as a solid foundation for understanding the impact of AI and IoT in banking. It successfully mentions the notable risks related to cybersecurity and bias while providing possible solutions. By incorporating more examples, addressing potential regulatory considerations, and discussing technological limitations, the article could further enrich its analysis and provide readers with even greater insights into the ethical deployment of AI in banking.

In their submission, C. Vijai and S. Natarajan, [17] revealed that the quick advancement of AI and ML technologies has created a high demand for skilled professionals. As a result, financial organizations face a serious scarcity of skills and needed expertise to develop, implement, and manage AI-driven innovations. In their view, this skill gap can slow the promotion and derived benefits of these technologies. The authors acknowledge the challenges, and difficulties of integration with legacy systems, which adds a critical viewpoint to their analysis. This balance between opportunities and challenges is essential for a good understanding of AI and ML implementation.

Other challenges as identified by C. Lloyd, M. R. Misheal, and N. Tavonga[18], include resistance to change, data quality issues and failure to document use of technology by auditors in their methodologies. In this article, the research questions are well-articulated, targeting specific areas of interest, such as understanding of AI tools, perceptions of their effectiveness, and challenges faced by auditors. However, the specific focus on Harare, Zimbabwe might overlook challenges and insights faced in other regions or countries. Comparative analysis with other regions could definitely add more meaning the discussion and allow for a stronger understanding of the issues.

The paper, "FinTech Futures: AI-Driven Payments and Supply Chain Innovations," by Huzaifa Arsalan [19] discusses the transformative effect of artificial intelligence (AI) on financial technology (FinTech), with a focus on electronic payment systems and supply chain management. The author argues that AI improves transaction security and efficiency by utilizing machine learning to combat fraud and personalize user experiences. According to the author, AI algorithms are able to detect current trends in fraud and adjust their detection strategies accordingly, thereby staying ahead of changing threats in an ever-changing landscape. By analyzing patterns and trends across different data sources, that may include transactional data, social media feeds, and dark web forums, AI-driven fraud detection systems can identify new fraud vectors and proactively implement countermeasures to protect against them. All in all, the paper appropriately identifies machine learning, natural language processing, and predictive analytics as a primary AI Components driving FinTech innovations. However, more specific citations within the text to support claims would enhance credibility.

In an article “AI-driven biometrics for secure fintech: Pioneering safety and trust,” [20] researchers submitted that Machine learning technologies applied in biometric systems is able to examine emerging patterns of fraud or unauthorized access and change their authentication rules as needed. According to the researchers, the ability for AI-driven biometrics to adapt to current trends makes it a perfect solution for FinTech security because of its ability to stay ahead of upcoming threats. The article discusses the practical ways of deploying AI-driven biometrics, which includes integration challenges, regulatory considerations and potential solutions. While the article provides a good overview, it could be more beneficial by adding more specific technical details about the AI algorithms and biometric methods used in FinTech applications.

In his article, Reddy[21] argues that AI has made a significant contribution to fraud prevention in cloud-based fintech applications. The paper investigates the integration of Artificial Intelligence (AI) with cloud computing in the fintech sector, bringing out its important role in improving application security. It discusses AI-powered techniques such as anomaly detection, which identifies unusual patterns that indicates unauthorized access and fraud prevention. The techniques utilize machine learning to detect deceptive activities. In this paper, two case studies shows the concepts of PayPal's anomaly detection framework, which reduces unauthorized access through real-time monitoring and adaptive learning, and Square's AI-driven techniques to combat payment fraud. The analysis shows how AI reinforces security measures and improves user trust and system efficiency, it tackles critical challenges and risks associated with AI implementation in fintech security. While the article includes a list of references, it could have benefited from more citations within the text to support specific claims and provide readers with resources for further reading.

As Submitted by Caprian [22] Scammers and hackers are continuously finding new methods of drawing off funds. His research findings revealed that from 2020 to 2021, fraud related losses reported to the Federal Trade Commission went up by more than 70% reaching a total of $5.8 billion globally. Alongside FinTechs and other financial organization, Banks are making extra effort to predict the activity of fraudsters. The paper presents findings from various reports indicating a rising trend in fraud, such as card related fraud losses projected to hit USD 32.3 billion in 2021 alone, with AI emerging to be a crucial tool for detection of fraud and prevention. What is notable is the rising cases related to mobile fraud like Buy Now Pay Later (BNPL) fraud, where criminals are able to exploit vulnerabilities in the financial system. In as much as the document put emphasis on the potential of AI in improving banking product quality and shows its important role in risk management and customer protection, the article attempts to cover many different areas of AI in banking, which may result in a lack of deep exploration of each topic.

According to the research submission [23] fraud prevention has been of key focus in the finance industry. Constantly, cybercriminals have caused havoc and increased the difficulty in financial management in many companies especially with credit fraud. In his view, AI-driven fraud detection software can help in analysing any stored client information, track all behavior, locations, and purchase trends. As a result, detection of any unusual activities that are not in line with the normal spending patterns of all clients becomes easy. However, the study would have been more

beneficial if it included empirical data or case studies, this would have strengthened the findings and provide more concrete evidence of AI's impact.

In addition, more research findings [24] revealed that HDFC Bank in India has been successful in the adoption and implementation of AI-driven innovations in its processes, this has resulted in enhanced efficiency, reduction in fraud related losses, and improved client trust. According to their findings the bank has continued investing in AI deployments and established the Center of Excellence to promote innovation in this domain. The authors made more submission that ICICI Bank's adoption of AI has led to huge refinement in areas like client service, prevention of fraud, assessment of risks and cost savings. In their opinion, the success in the implementation of AI-driven innovations can serve as a prototype for many banks in India and the world over. All in all, the paper offers a strong foundation for understanding the application and effectiveness of AI in the banking sector. The paper highlights the potential of AI while also acknowledging accompanying challenges. Addressing the limitations and expanding on the areas for improvement would further improve the paper's value and impact.

Kulkarni, Bansal[25] submitted that so many electronic transactions happen on a daily basis as users pay bills and withdraw money, and do many other things using online accounts. Fraud has continuously remained a serious concern for banks and financial organization. Such that, each year, they suffer huge financial losses amounting to billions of dollars, due to fraudulent related activities associated to identity theft, credit card fraud and money laundering. In the opinion of the authors, the increased understanding of fraud patterns make machine learning models to flag suspicious activities with greater precision and efficiency. This leads to faster detection and mitigation of fraudulent transactions, resulting in the reduction of financial losses that institutions may face.

Adding to the submissions of other scholars, Varma[26] submitted that artificial intelligence (AI)-driven innovations have revolutionized how financial organizations associate with their clients and protect their assets, from automated fraud detection systems to merchant chatbots and virtual assistants. According to the author’s perspective, investment organizations now have undisrupted access to data-driven perceptions facilitated by AI algorithms and machine learning techniques. This improves risk management and leads to more precise market forecasts which is key to improved decision-making. All in all, the article provides a good insight of the effect of AI on the financial sector, particularly in banking and investment. The article identifies key benefits and

challenges of AI application. The use of primary and secondary data, along with statistical analysis, strengthens the findings. However, more research may be needed to address the limitations related to data sensitivity and explore specific strategies for ensuring the transparency of AI systems.

As submitted by M. J. Madhurya, H. L. Gururaj, B. C. Soundarya, K. P. Vidyashree, and A. B. Rajendra [27] credit card related fraud has been on the increase such that the internet shopping Fraud Index contend that the rate of fraud in online stores had gone up from 0.06% in 2016 to 0.23% in 2017. Of all frauds, 10% were considered entirely of Credit cards related that had culminated in huge financial losses that distress companies. According to their research, AI powered fraud detection is the solution to look to. The researchers went a step further to analyze how the main AI tools can be used in this kind of fraud and submitted the results of their research as shown on Table 1 below:

Table 1
F1 score table.

Method	Accuracy	F1-Score
Logistic Regression	78.83	.015
Decision Tree	72.66	0.369
Random Forest Classifier	80.16	0.446
Naïve Bayes (Gaussian)	61.5	0.443
Naïve Bayes (Bernoulli)	75.83	0.491
K- Neighbour's Classifier	72.5	0.239
ANN-DL	77.63	0.44

The "Journal of Engineering and Applied The Transformative Impact of AI on Financial Institutions , with a Focus on Banking," [28] revealed that the use of AI in fraud detection marks a radical change in the security concept from reactive to proactive. The researcher revealed that instead of merely responding to incidents after they occur, AI enables financial institutions to deploy preemptive measures, stopping potential threats before they occur. This not only protects customer assets but also strengthen customers' trust in their financial institutions. While the article offers a solid groundwork for understanding AI's transformative impact on banking, it could be significantly strengthened by a more balanced presentation of risks and practical implementation frameworks, with deeper analytical content. By addressing these areas, the paper could serve as a more comprehensive resource for stakeholders in the financial industry.

In the area of regulatory compliance Devan etal, [29] submitted that AI-driven fraud detection systems help banks in sticking to regulatory obligations like anti-money laundering (AML) and know your customer (KYC) regulations through monitoring of transactions for dubious activities and reporting them to regulatory authorities. Authors suggested that by automating compliance

processes and improving detection capabilities, AI solutions can help banks meet their regulatory obligations and avoid the risk of regulatory penalties. The article covers a wide range of applications for AI in banking, including customer service, detection of fraud, personalized banking, credit scoring, and regulatory compliance. This breadth offers readers a holistic view of AI's impact. Also, the discussion on AI technologies reflects current trends in the banking sector, relevant to practitioners and academics alike. The inclusion of real-time analytics highlights the urgency and relevance of the topic.

In an article titled “Efficiency of Federated Learning and Blockchain in Preserving Privacy and Enhancing the Performance of Credit Card Fraud Detection (CCFD) Systems” [30].The scholars submitted that Fraudsters persistently change attack techniques and fraud tactics to trick Credit Card Fraud Detection Systems, and stop the detection of both newly and previously unseen fraudulent transactions. They employ different attack methods in the likes of data poisoning, evasion attacks and exploitation of input data to mislead the system. To mitigate this fraudulent activity specific to credit card fraud, they proposed their own framework that uses three machine learning and deep neural network algorithms: Random Forest (RF), Convolutional Neural Networks (CNNs), as well as long short-term memory (LSTM). In their opinion these are algorithms known for their high capability and performance in accurately detecting fraud instances, namely RF, CNN, and LSTM. In their proposal, they claimed that the framework differentiates itself by combining blockchain technology with federated learning, guaranteeing high privacy preservation and data protection. All in all, the article presents a notable contribution to the field of credit card fraud detection. The integration of federated learning and blockchain technology offers a promising approach to counteract the problems related to data privacy and security. While there are some limitations, the study provides a solid bedrock for future study and development in this area. Addressing the possible weaknesses, such as providing a more in-depth cost analysis, and further validating the system in real-world scenarios, would make the research even stronger.

A paper submited by Durga and Sambrow [31] revealed that as digital banking environment expand, different transaction methods are created, these may include online payments, mobile apps, traditional ATMs, and point-of-sale devices and Fraudsters often take advantage of inconsistencies between these methods. The researchers submitted that the use of an AI based fraud detection system, would offer a strong defense meassure through cross-channel analysis. In

their submission, they stated that AI systems combine data from every interaction point, allowing for a complete view of a client's behavior and spending pattern across all mediums. For example, in an event that a client's mobile application is used to launch a transfer just minutes after an ATM withdrawal in a different city, AI is able to detect the inconsistency and raise a flag for further review. Through interconnecting contrasting data from multiple channels, AI stops fraudsters from getting advantage of the gaps between them. Generally, the article offers a well-structured and clear overview of the adoptions and integration of AI in banking fraud prevention. The paper is comprehensive on the exploration of technology, its potential and adaptability. However, addition of empirical evidence, pointing out regulatory challenges more thoroughly and bringing out the risks associated with over-reliance on automated systems would have been beneficial. These improvements in the mentioned areas would have provided a more balanced outlook on the capabilities and limitations of AI in the banking sector.

Jahnavi and Tibrewal [32] submitted that AI assists in analysing data within seconds and efficiently detect complex patterns , otherwise it might be extremely hard for fraud analysts to detect fraud in banking. The authors advised that AI eliminates the labor-intensive work and makes the fraud analysts to concentrate on more serious cases, such as when risk scores are very high. The fraud analysts work and efficiency is improved as their tasks uses automated AI innovations. No wonder, the use of AI for prevention of fraud is the best alternative among large enterprises and financial institutions. The article noted lack of available data science skills, even as there is a high demand for these competencies in the banking industry to effectively utilize AI. This contradiction highlights a mismatch between industry needs and the workforce's preparedness.

2.1 Critical Review of Related Works

More research has been undertaken in this domain which provides further insights in the subject matter and various submissions of related works has been made.

For instance, the paper "AI-Driven Fraud Detection Systems: A Comparative Study across Banking, Insurance, and Healthcare" submitted by Pankaj Zanke[33] offers very important information on the application and effectiveness of AI in detection of fraud across different sectors. The article's main focus is on evaluating and comparing the effectiveness, scalability, and adaptability of AI-driven fraud detection models in three distinct sectors: banking, insurance, and healthcare. The article seeks to understand how these systems operate differently in each sector,

given the unique challenges, types of fraud, and regulatory environments they face and how it relates to technological improvements in machine learning algorithms and anomaly detection methods. But there is a noticeable limitation on discussing potential pitfalls associated with existing AI-driven fraud detection systems. This involves handling problems related to data confidentiality and bias in algorithms, which are critical in the factors of fraud detection.

The article entitled “Detecting anomalies in financial statements using machine learning algorithm: The case of Vietnamese listed firms"[34] presents a timely and relevant examination of the role of AI in the domain of banking, particularly in combating fraud. However, the study acknowledges limitations regarding the accessibility of real accounting data from stock exchanges. This raises questions about how reliable the data that was used is, and how it could impact the validity of the anomaly detection results. All in all, the research would have been improved by exploring how possible it is to implement real-time data analysis in order to identify anomalies as they occur. This would have involved deploying machine learning algorithms in a production environment, allowing for timely detection of irregularities in financial reporting.

More research articles like the paper entitled “Harnessing AI for Next-Generation Financial Fraud Detection: A Data Driven Revolution” [35] provide a unique perspective on Comparative Analysis of Traditional and AI-Based Approaches. Unlike many studies that focus solely on AI technologies, this paper provides a structured comparison between Conventional methods of detecting fraud and AI-driven models. This analysis clearly points out the limitations of conventional methods in the context of increasingly complicated fraud tactics and positions AI as a necessary solution in fraud detection strategies. The paper rightly recognizes issues such as data quality, model transparency, and ethical considerations, making it a holistic discussion rather than a purely technical analysis. However, the paper would have made a strong impact by delving deeper into addressing specific implementation challenges faced by financial institutions, including the cost, resource allocation, and integration into existing systems.

2.2 Proposed Model

Extending the ideas discussed in the literature review, this research proposes a supervised machine learning model to predict credit/debit card fraud with at least 90% accuracy. The model will integrate:

1. **Logistic Regression (LR):** This will be use as a baseline classifier, offering interpretability for fraud probability estimation.
2. **Random Forest (RF):** This will improve fraud detection by reducing variance and handling non-linearity in transaction data.
3. **XGBoost:** This will utilize boosting techniques to improve classification performance and increase model generalization.
4. **Stacked Ensemble Learning:** This will be used in combining predictions from LR, RF, and XGBoost using a meta-classifier to enhance decision-making.

The proposed approach leverages ensemble learning to maximize fraud detection accuracy while addressing challenges identified in existing studies.

2.3 Chapter Summary

This chapter provides a careful review of related literature on AI-driven fraud detection systems, pointing out key studies and their contributions, limitations and areas for further research.

One outstanding study, “AI-Driven Fraud Detection Systems: A Comparative Study across Banking, Insurance, and Healthcare” by Pankaj Zanke, examines how effective AI is in fraud detection in three sectors. The paper looks into how scalable and adaptable of AI models are, but lacks a detailed discussion on potential pitfalls such as data confidentiality and algorithmic bias.

Overall, the literature review on the use of AI in fraud detection systems brings out notable advancements and challenges in the field. In as much many these studies show AI’s potential in improving fraud detection in banking, insurance, and healthcare, they also point out important gaps like data privacy concerns, biases in algorithms and difficulties in the implementation. It is suggested that real-time data integration, ethical compliance frameworks, and resource-efficient deployment techniques are important for improving AI's effectiveness in fraud prevention. Future studies should focus on addressing these limitations to develop more robust, transparent, and adaptable fraud detection solutions.

The following chapter is the methodology section which delves into the research design, adopted method and justification. It gives details about the source of research data and datasets are given, addresses data collection methods and data analysis techniques. Lastly it speaks to Ethical concerns related to the research.

CHAPTER 3 – METHODOLOGY

3.0 Research Design

This study tackles the challenge of credit/debit card fraud by developing a supervised machine learning model to catch fraudulent transactions with at least 90% accuracy. Logistic Regression (LR), Random Forest (RF) and XGBoost will be used to identify fraud, these methods will be combined in a stacked ensemble learning approach to improve fraud detection. The research design outlines the methodology, data collection process, model development, evaluation metrics, and ethical considerations.

3.1 Adopted Method and Justification

This study adopts a quantitative research approach which requires lots of data on past transactions, The methodology will use supervised machine learning techniques to classify transactions as fraudulent or legitimate. The research follows a predictive analytics methodology, where past transaction data is used in training the model and which will be used future fraud prediction[36]. Ensuring we are ethical and responsible in our methods.

The methodology was chosen for the following key reasons:

- **Effectiveness in Detection of Fraud:** Supervised machine learning models trained on past transactions are very good at identifying fraudulent activity.
- **Reliability through combined predictions:** Using multiple prediction models together makes our fraud detection more reliable and less prone to errors.
- **Finding the right balance between understanding and accuracy:** Logistic Regression helps us understand why a transaction is flagged as fraudulent, while others are better at simply identifying fraudulent transactions. Our combined approach gives us the best of both worlds.
- **Speed and scalability:** These methods are fast and efficient, making them suitable for banks that handle many transactions.
- **Handling uneven data:** We use special techniques SMOTE to account for the fact that fraudulent transactions are much rarer than legitimate ones[36].

3.2 Association of Research Method to Project:

The supervised learning approach without doubt aligns with the project objective of designing and implementing a machine learning model that is able to detect and prevent fraudulent transactions with at least 90% accuracy. When the supervised machine learning model is trained using past transaction data (with known outcomes), it learns how to generalize and make predictions on new transactions.

The fraud detection problem at hand is a binary classification task and makes supervised learning a good choice. Moreover, performance evaluation metrics such as Precision, Recall, and AUPRC (Area Under the Precision-Recall Curve) provide meaningful insight attributable to the inherent class imbalance.

3.3 Research Data and Dataset

- **Data Source:** Due to ethical issues regarding financial data, the dataset to train the model is sourced from well publicized and accessible credit card fraud datasets from the Kaggle Credit Card Fraud on https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud [available online]. This dataset includes credit card and debit card related transactions carried out by European cardholders in September 2013. It spans two days and comprises 284,807 transactions, of which 492 are fraudulent. This makes the dataset highly imbalanced, with fraud cases constituting only 0.172% of 284,807 transactions.
- **Features:** All variables are numeric. Excluding 'Time' and 'Amount', features are the result of a Principal Component Analysis (PCA) modification intended to protect sensitive financial information.

  The dataset has the following features:

  a) Time – seconds since the first transaction.
  b) Amount – transaction amount.
  c) V1-V28 – anonymized PCA components.
  d) Class – the target label which is either (0: legitimate, 1: fraud).

This dataset suits well for supervised learning models due to its labeling and structured format.

- **Pre-Processing:** Data will be cleaned to handle missing values, outliers, and unbalanced class distributions using sampling methods such as SMOTE (Synthetic Minority Over-sampling Technique).

3.4 Data Collection Methods and Data Analysis Techniques

A publicly available dataset from Kaggle was used. This dataset contains real transaction information. It requires no active data collection but it is sourced from an authentic transactional record. Below are the main activities involved:

3.4.1 Data Preprocessing

This is the processing of cleaning and converting raw data into acceptable format for training models**.** It is important as it improves model accuracy, reduce training time and enables the model to generalize well on the unseen data [37]. This involves the following steps:

a. Cleaning the data to remove any inconsistencies and irrelevant information.
b. Because of uneven dataset, handling class imbalance is done using techniques like under-sampling and over-sampling. E.g Synthetic Minority Over-sampling Technique (SMOTE).
c. Normalizing the data: the dataset is standardized to ensure all features.
d. Feature selection or dimensionality reduction (if needed).

3.4.2 Modeling and Evaluation

This is the process of selecting and training a supervised machine learning model to learn patterns from labeled data[38]. It involves the following steps:

a. Splitting data into training and test sets.
b. Applying cross-validation for performance robustness.

3.4.3 Using Evaluation Metrics Suited for Imbalanced Datasets:

These are quantitative measures used to examine how well a model perform on a given dataset, they give insight on the quality of predictions a trained model made on the unseen dataset such as the test data. The evaluation metrics used are as follows:

a. **Precision:** measures how many of the positive prediction made by the model are actually correct. High precision means few face positives.
b. **Recall:** measures how many of the actual positive cases the model correctly identified, high recall means the model caught most of the real positives.
c. **F1-Score:** this is the harmonic mean of precision and recall, it balances the two metrics. It is good for imbalanced dataset like fraud detection.
d. **AUPRC:** is the area under the curve that plots precision vs recall at various classification threshold. It is valuable in imbalanced datasets where ROC-AUC can be misleading.
e. **ROC-AUC:** area under the receiver operating characteristic curve plots the true positive rate (Recall) vs false positive rate (FPT).

3.5 Ethical Concerns Related to the Research

This research addressed several ethical isues related to fraud detection using machine learning. These concerns included:

- **Data Privacy:** Even though the dataset is anonymized, special care must be taken to ensure no attempt is made to de-anonymize the data.
- **Bias Mitigation:** Addressing potential algorithmic bias. Machine learning models may mistakenly discriminate against some groups due to biased training data. Techniques such as fairness-aware algorithms and bias audits must be employed.
- **Regulatory Compliance:** Adhering to financial data protection laws such as GDPR and PCI DSS. To ensure compliancy with these regulations the dataset was collected, stored, and processed ethically.

- **False Positives and Customer Impact:** Fraud detection models may flag legitimate transactions as fraudulent, resulting into inconveniencing of customers. Efforts to balance fraud detection sensitivity with minimizing false positives were applied.
- **Ethical Use of AI:** To avoid the misuse of machine learning models for surveillance or unjustified transaction monitoring. The model was used strictly for fraud prevention while respecting user rights.

3.6 Chapter Summary

This chapter walks you through the methodology used to develop a supervised machine learning model for detecting card fraud. It employs a data-driven, quantitative research design with supervised learning techniques well-suited for sorting transactions into categories like fraud or not fraud. Because there are usually far more legitimate transactions than fraudulent ones, we selected the methodology that justifies the use of specific algorithms and evaluation metrics tailored to handle imbalanced datasets. We also made sure to protect people's privacy by using data responsibly. Overall, the result is a reliable, understandable system that can be used in the real world to help prevent fraud. The next chapter explores appropriate modelling in relation to the project. It highlights techniques, algorithms, mechanisms used to design the model, and further delves into experiments and implementations and concludes with Chapter Summary.

CHAPTER 4 - PROTOTYPE, DATA, EXPERIMENTS, AND IMPLEMENTATION

4.0 Appropriate Modelling in Relation to Project

The hybrid ensemble model was designed to address the challenges of credit/debit card fraud detection, which include extreme class imbalance, non-linear feature relationships, and the need for high interpretability. By integrating **Logistic Regression (LR)**, **Random Forest (RF)**, **XGBoost**, and **stacked generalization**, the model balances interpretability and predictive power. LR provides a baseline understanding of fraud probability, RF handles non-linear interactions, XGBoost optimizes classification through gradient boosting, and the stacked ensemble minimizes overfitting while improving generalization.

4.1 Techniques, Algorithms, Mechanisms

Three primary models are integrated:

- **Logistic Regression (LR):** this gives a basic prediction of fraud and it serves as a baseline classifier and helps provide insights into the probability of fraudulent activity.
- **Random Forest (RF):** this handles complex relationships between transaction details and reduce model variance.
- **XGBoost:** this is a gradient boosting framework that enhances performance by minimizing bias and optimizing via regularized loss functions.

These models are combined using **Stacked Ensemble Learning**, where the outputs of individual models are used as features by a meta-classifier to make the final prediction. This aims to leverage the strengths of each base model to improve generalization[39].

4.2 Data Preprocessing:

Data was analyzed and the following activities are undertaken:

- Checking for null values from data before the model is trained.
- Resolving data imbalance with **SMOTETomek** which is a hybrid technique that combines **SMOTE**(Synthetic Minority Over-sampling Technique) and **Tomek links** – a data

cleaning method that identifies and removes overlapping examples between classes in order to avoid bias towards the majority class

- Stratified sampling to preserve class distribution.

4.3 Designed Prototype/ Model

The prototype is based on a layered architecture as shown below:

1. **Data Preprocessing Layer**: This Included scaling using StandardScaler, and train-test splitting.
2. **Base Learners**: Logistic Regression, Random Forest, and XGBoost were trained individually. Logistic regression was used as a meta classifier.
3. **Stacked Ensemble Layer**: Outputs of base learners were combined and fed into a logistic regression meta-classifier.
4. **Prediction and Evaluation Layer**: Outputs final prediction and evaluates accuracy, precision, recall, and F1-score.

The layered architecture is a combination of various models, and the meta-model's performance is based on strengths of base learners.

4.4 Main Functions, Models, Frameworks.

- **Logistic Regression()** from sklearn.linear_model: Used for both baseline prediction and as meta-classifier.
- **Random Forest Classifier()** from sklearn.ensemble: Captures complex patterns.
- **XGB Classifier()** from xgboost: Boosting model to enhance accuracy.
- **Stacking Classifier()** from sklearn.ensemble: Integrates base models and handles ensemble learning. It Maximizes AUC-ROC and F1-score by leveraging collective intelligence.
- **Standard Scaler()**: Ensures proper scaling for models sensitive to data scale.
- **train_test_split()**: Maintains stratification and ensures unbiased evaluation.

These components align with project objectives to maximize fraud detection, reduce false negatives, and offer explainability in results.

4.5 Key Metrics:

The key metrics used are as follows:

- Accuracy, Precision , Recall , F1-score, AUPRC.

4.6 Experiments:

In order to build the most accurate, efficient and reliable model, experiments were done by testing different configurations and comparing results. This helped to choose the best model for implementation. For all the experiments the base models and parameters were as follows:

```
base_models = [
    ('lr', LogisticRegression (max_iter=500, class_weight='balanced', random_state=42)),
    ('rf', RandomForestClassifier(n_estimators=100,class_weight='balanced', random_state=42)),
    ('xgb', XGBClassifier(n_estimators=100,eval_metric='logloss', random_state=42))
]
```

Experiment 1: Finding the Best Meta_model.

Based on the above defined base model, experiments were done on different meta classifiers to determine the best meta_model that would contribute to the performance of the prototype. Below is the outcome of the experiments on different meta classifier:

Meta_Classifier	Stacking_Model	Sampling Technique		Precision	Recall	F1-Score	AUPRC
LR	Stacking_classifier With CV=3	SMOTETomek		0.85	0.83	0.84	0.8703
RF	Stacking_classifier With CV=3	SMOTETomek		0.97	0.74	0.84	0.8637
XGBClassifier	Stacking_classifier With CV=3	SMOTETomek		0.98	0.65	0.79	0.8215

Table 4. 1 Performance of different meta classifiers

As shown in table 4.1 above, experiments with different meta classifiers were done using the stacking model whose cross validation (CV) was set to 3, and the sampling technique was SMOTETomek. In this experiment, Logistic Regression (LR) performed better with the overall AUPRC result of 0.8703.

Experiment 2: Finding the best Sampling Technique.

Dealing with imbalanced dataset requires application of under-sampling or over-sampling on the training data set. This is necessary as it addresses class imbalance. Class imbalance is a problem as the prototype may be biased toward the majority class, predicting it more accurately while neglecting the minority class. The model may not generalize well to new data and this may lead to misleading accuracy.

To find the best sampling technique, experiments were done on the sampling techniques which included the following in a pipeline:

- **SMOTE:** Synthetic Minority Over-sampling Technique
- **SMOTETomek:** A hybrid of SMOTE and Tomek Links
- **ADASY:** Adaptive Synthetic Sampling

Meta_Classifier	Stacking_Model	Sampling Technique	Precision	Recall	F1-Score	AUPRC
LR	*Stacking_classifier With CV=3*	*SMOTETomek*	*0.85*	*0.83*	*0.84*	*0.8703*
		ADAYSN	*0.20*	*0.91*	*0.32*	*0.8632*
		SMOTE	*0.86*	*0.85*	*0.85*	*0.8526*
RF	*Stacking_classifier With CV=3*	*SMOTETomek*	*0.97*	*0.74*	*0.84*	*0.8637*
		ADAYSN	*0.65*	*0.81*	*0.72*	*0.8339*
		SMOTE	*0.93*	*0.79*	*0.85*	*0.8634*
XGBClassifier	*Stacking_classifier With CV=3*	*SMOTETomek*	*0.98*	*0.65*	*0.79*	*0.8215*
		ADAYSN	*0.68*	*0.84*	*0.75*	*0.8403*
		SMOTE	*0.93*	*0.78*	*0.84*	*0.8667*

Table 4. 2 Finding the best Sampling Technique.

As show in table 4.2, experiments on three (3) sampling techniques which involved SMOTETomek, ADASYN and SMOTE were done using three (3) meta_clssifiers LR, RF and XGBClassifier, where the stacking_model had the Cross Validation parameter set to 3. The AUPRC column is indicative of the overall results.

Experiment 3: Finding the best Cross Validation value.

Fine-tuning cross-validation (CV) parameters is an important aspect to be undertaken as it improves model evaluation by reducing bias, increasing robustness and avoid overfitting to the test dataset. Fine tuning CV parameters also help in comparing the models and select the one for a particular problem. Experiments were done on the Cross Validation value by adjusting the values from three (3) going upwards to 7 and below were the AUPRC results:

Meta_Classifier	Cross Validation Value	AUPRC
Logistic Regression	CV = 3	0.8703

Table 4. 3 Results When CV = 3

Meta_Classifier	Cross Validation Value	AUPRC
Logistic Regression	CV = 4	0.8685

Table 4. 4 Results When CV = 4

Meta_Classifier	Cross Validation Value	AUPRC
Logistic Regression	CV = 5	0.8615

Table 4. 5 Results When CV = 5

Meta_Classifier	Cross Validation Value	AUPRC
Logistic Regression	CV = 6	0.8661

Table 4. 6 Results When CV = 6

Meta_Classifier	Cross Validation Value	AUPRC
Logistic Regression	CV = 7	0.8634

Table 4. 7 Results When CV = 7

When Cross Validation was set to 7 the model failed to converge during training and gave the error below:

```
C:\Users\USER\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:469: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

Figure 4.7. 1 Error at Max_Iter = 500

Convergence was only achieved when the Max_Iteration value was set to 1000. And the AUPRC of 0.8634 was recorded. These experiments on Cross Validation parameters assisted in making a final decision on the parameters to be implemented in the final model.

4.7.0 IMPLEMENTATION:

The project was implemented in a controlled environment using Jupiter Notebook for the following reasons:

- **Interactive Development and Visualization** Jupyter Notebook allows for real-time code execution, data manipulation, and visualization. This makes it ideal for iteratively developing, testing, and refining fraud detection models with immediate feedback.

- **Reproducibility and Documentation** Each cell in a notebook can include both code and markdown explanations. This combination helps document the thought process and ensures that experiments are reproducible and well-structured for auditing and collaboration.

- **Integration with Data Science Libraries** Jupyter supports seamless integration with key machine learning libraries such as scikit-learn, TensorFlow, PyTorch, and pandas—essential for building and evaluating fraud detection algorithms.

- **Version Control and Experiment Tracking** In a controlled environment, Jupyter Notebooks can be versioned, allowing track of changes, experiment with model parameters, and revert to earlier versions when necessary.

- **Safe Testing Environment** A controlled environment ensures secure access to sensitive financial data, sandboxed model execution, and compliance with data governance policies—critical for building and testing fraud models responsibly.

4.8.0 STEP BY STEP CODING OF THE MODEL

STEP 1: Importing the Needed Libraries

Using the Jupyter Notebook, all the needed library required to build the model were imported as follows:

```
#Import needed Libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, cross_val_predict, StratifiedKFold
from sklearn.metrics import classification_report, confusion_matrix, precision_recall_curve, auc
from sklearn.metrics import roc_auc_score
from sklearn.pipeline import Pipeline
from sklearn.ensemble import StackingClassifier, RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from xgboost import XGBClassifier
from sklearn.preprocessing import StandardScaler
from imblearn.over_sampling import SMOTE
from imblearn.pipeline import Pipeline as ImbPipeline
import matplotlib.pyplot as plt
import seaborn as sns
from imblearn.combine import SMOTETomek
from imblearn.over_sampling import SMOTE, ADASYN
import joblib
```

Figure 4.8 1 Importing the needed libraries

STEP 2: Loading the Dataset

```
# Load Data
df = pd.read_csv('C:/Users/USER/Fraud/creditcard.csv')
X = df.drop('Class', axis=1)
y = df['Class']
```

Figure 4.8 2 Loading the dataset

The commands above were used to load a CSV file located at a given path into a Pandas DataFrame called df. The file creditcard.csv contains credit and debit card transaction data used for fraud detection.

X = df.drop('Class', axis=1) creates a new DataFrame x that contains all columns except the class column, this represents the features for training the model.

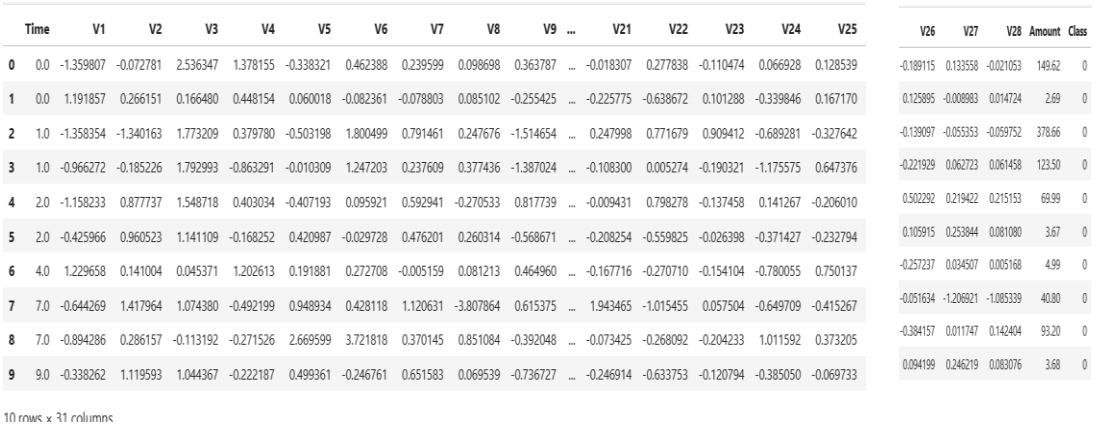
y = df['Class'] extracts the class column into a series y, which represents the target labels (e.g 1 for fraud, 0 for non-fraud).

To quickly inspect the loaded dataset, check if the data loaded correctly and review column names, types, and example values the following code below was run:

```
df.head(10)
```

Figure 4.8 3 display and inspect the first 10 rows in the dataset

The output of the above line of code is shown below:



	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0
5	2.0	-0.425966	0.960523	1.141109	-0.168252	0.420987	-0.029728	0.476201	0.260314	-0.568671	...	-0.208254	-0.559825	-0.026398	-0.371427	-0.232794	0.105915	0.253844	0.081080	3.67	0
6	4.0	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.005159	0.081213	0.464960	...	-0.167716	-0.270710	-0.154104	-0.780055	0.750137	-0.257237	0.034507	0.005168	4.99	0
7	7.0	-0.644269	1.417964	1.074380	-0.492199	0.948934	0.428118	1.120631	-3.807864	0.615375	...	1.943465	-1.015455	0.057504	-0.649709	-0.415267	-0.051634	-1.206921	-1.085339	40.80	0
8	7.0	-0.894286	0.286157	-0.113192	-0.271526	2.669599	3.721818	0.370145	0.851084	-0.392048	...	-0.073425	-0.268092	-0.204233	1.011592	0.373205	-0.384157	0.011747	0.142404	93.20	0
9	9.0	-0.338262	1.119593	1.044367	-0.222187	0.499361	-0.246761	0.651583	0.069539	-0.736727	...	-0.246914	-0.633753	-0.120794	-0.385050	-0.069733	0.094199	0.246219	0.083076	3.68	0

10 rows × 31 columns

Figure 4.8 4 the screenshot of the first 10 rows of the dataset

As shown in the output above:

- Most class values are 0, indicating non-fraudulent transactions.
- The PCA features (V1to V28) include both positive and negative values often with small magnitude.
- Time increases as you go down the rows, indicating the passage of time.
- Dataset is highly imbalanced, meaning there may be a need to balance the dataset or apply anomaly detection techniques.

STEP 3: Explore the Dataset

Checking for missing values is a basic part of data quality assessment. Missing data lead to biased results if not handled properly and usually machine learning algorithms do not accept data with missing values and give errors when training them. The code below was used to check for missing values:

```
print(df.isnull().sum())
```

Figure 4.8 5 code to check for missing values

No missing values were found as shown in the output below, hence no rows and columns were dropped or imputed.

```
Time      0
V1        0
V2        0
V3        0
V4        0
V5        0
V6        0
V7        0
V8        0
V9        0
V10       0
V11       0
V12       0
V13       0
V14       0
V15       0
V16       0
V17       0
V18       0
V19       0
V20       0
V21       0
V22       0
V23       0
V24       0
V25       0
V26       0
V27       0
V28       0
Amount    0
Class     0
dtype: int64
```

Figure 4.8 6 output for missing values

Further checks on the missing values were done using the command below:

```
df.isnull()
```

Figure 4.8 7 screenshot of the code to check null values

Looking at the output below, each cell as a False value indicating that there is no value missing.

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
284802	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
284803	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
284804	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
284805	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
284806	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False

284807 rows × 31 columns

Figure 4.8 8 output showing no missing values

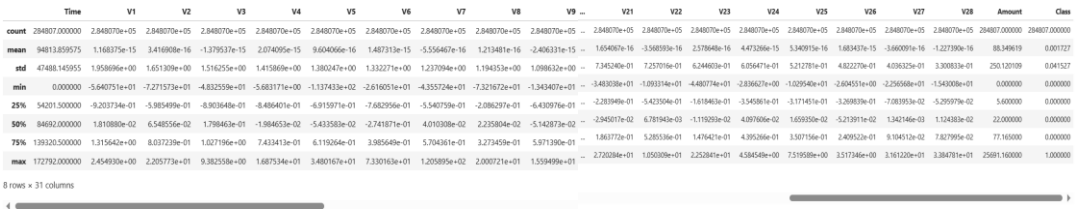
In order to detect outliers and skewed distribution, spot missing or abnormal values. The following code was used.

```
df.describe()
```

Figure 4.8 9 command to spot missing values

The output of the above code shown below revealed the following details.

- Time ranges from 0 to 172792, meaning it likely measures time elapsed since the first transaction.
- Most of the V1–V28 values have a **mean near 0**, suggesting they're **PCA components**.
- Amount has a **max value of 25,691.16**, with a **median of 22**, showing a wide range of transaction values.
- Class has mean = 0.001727, meaning **fraud cases are very rare** (about 0.17%) and min = 0, max = 1, which confirms it's a binary classification target.



	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	...	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	284807.000000	284807.000000
mean	94813.859575	1.168375e-15	3.416908e-16	-1.379537e-15	2.074095e-15	9.604066e-16	1.487313e-15	-5.556467e-16	1.213481e-16	-2.406331e-15	...	1.654067e-16	-3.568593e-16	2.578648e-16	4.473266e-15	5.340915e-16	1.683437e-15	-3.660091e-16	-1.227390e-16	88.349619	0.001727
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00	1.380247e+00	1.332271e+00	1.237094e+00	1.194353e+00	1.098632e+00	...	7.345240e-01	7.257016e-01	6.244603e-01	6.056471e-01	5.212781e-01	4.822270e-01	4.036325e-01	3.300833e-01	250.120109	0.041527
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00	-1.137433e+02	-2.616051e+01	-4.355724e+01	-7.321672e+01	-1.343407e+01	...	-3.483038e+01	-1.093314e+01	-4.480774e+01	-2.836627e+00	-1.029540e+01	-2.604551e+00	-2.256568e+01	-1.543008e+01	0.000000	0.000000
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01	-7.682956e-01	-5.540759e-01	-2.086297e-01	-6.430976e-01	...	-2.283949e-01	-5.423504e-01	-1.618463e-01	-3.545861e-01	-3.171451e-01	-3.269839e-01	-7.083953e-02	-5.295979e-02	5.600000	0.000000
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02	-2.741871e-01	4.010308e-02	2.235804e-02	-5.142873e-02	...	-2.945017e-02	6.781943e-03	-1.119293e-02	4.097606e-02	1.659350e-02	-5.213911e-02	1.342146e-03	1.124383e-02	22.000000	0.000000
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01	3.985649e-01	5.704361e-01	3.273459e-01	5.971390e-01	...	1.863772e-01	5.285536e-01	1.476421e-01	4.395266e-01	3.507156e-01	2.409522e-01	9.104512e-02	7.827995e-02	77.165000	0.000000
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01	7.330163e+01	1.205895e+02	2.000721e+01	1.559499e+01	...	2.720284e+01	1.050309e+01	2.252841e+01	4.584549e+00	7.519589e+00	3.517346e+00	3.161220e+01	3.384781e+01	25691.160000	1.000000

8 rows × 31 columns

Figure 4.8 10 output showing the mean and median in the dataset

The command below helped in understanding the shape of the DataFrame.

```
df.shape
```

Figure 4.8 11 command to display the shape of data

As shown in the output below the dataset has 284807 data records and each row has 31 features.

```
(284807, 31)
```

Figure 4.8 12

To calculate the percentage distribution of each class in the class column of the DataFrame the command below was used:

```
df['Class'].value_counts(normalize=True)*100
```

Figure 4.8 13 command to find the percentage distribution of class column

The results in the output below shows the percentage (proportion) of each class in the dataset.

```
Class
0    99.827251
1     0.172749
Name: proportion, dtype: float64
```

Figure 4.8 14 output of in percentage distribution

Based on the above output:

- Class 0: 99.83% of the transactions are non-fraud and
- Class 1: 0.17% of the transactions are fraudulent.

STEP 4: Visualizing the Dataset

The class distribution was further visualized using the following line of code:

```
import seaborn as sns

sns.countplot(x='Class', data=df, color='black')
plt.title('Fraud vs Non-Fraud Transactions')
plt.show()
```

Figure 4.8 15 command to visualize the class distribution

The code creates a bar chart of counts for each unique value in the class column. Class = 0 (being non-fraud) and class = 1 (fraud). The results in the Bar chart below shows how many times they appear:

Figure 4.8 16 visual display of the class distribution

The above bar chart helped in understating how imbalanced the dataset is. This is a common challenge in fraud detection. Unless this imbalance is handled properly, the model would fail to detect fraud.

In order to visually spot outliers which are points far outside the data distribution a boxplots for detecting outliers was created using the code below:

```
sns.boxplot(x=df['Amount'], color='black')
plt.title('Boxplot of Transaction Amount')
plt.show()
```

Figure 4.8 17 command to display the Boxplot of transaction

The output for the above line of code is shown below:

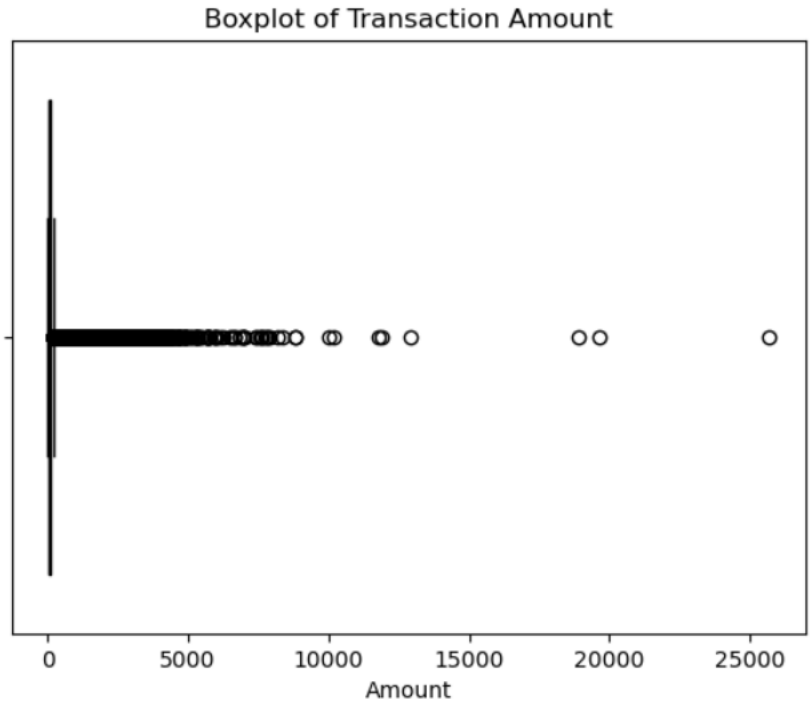


Figure 4.8 18 Boxplot of Transaction amount

As indicated in the boxplot of transaction amount above, the median is closer to the left edge of the box and whiskers are not balanced which means data is not symmetric. This reflects a large concentration of smaller values and a few large outliers. These outliers were not removed as they usually represent the minority class of fraud cases, therefore, removing them could result into deleting fraudulent transactions.

To confirm skewness of the above results the following line of code was used:

```
from scipy.stats import skew

skewness = skew(df['Amount'])
print(f"Skewness: {skewness}")
```

Figure 4.8 19 screenshot of code to check for skewness

The results below revealed the skewness in the dataset:

```
Skewness: 16.97763503663315
```

It was observed that this is common in datasets like credit card transactions where most amounts are small but few are very large.

STEP 5: Split the Dataset.

After understanding the dataset, it was time to split the dataset in readiness for building and training the model. The dataset was split into a training set and testing set as shown in the line of code below. The code performs a train-test split on the dataset for supervised learning, specifically a binary classification task like fraud detection.

```
#Train/Test Split
X = df.drop('Class', axis=1)
y = df['Class']
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, test_size=0.2, random_state=42)
```

Figure 5. 1 screenshot of code deviding the dataset into training set and test dataset

The above code splits data into:

- X_train and y_train for training the model
- X_test and y_test for testing

The test_size = 0.2 allocates 20% of the data for testing.

Radom_state = 42 ensures reproducibility of the split.

Stratify = y keeps the same class distribution in both train and test sets which is extremely important for imbalanced data.This ensures fair evaluation, prevents skewed splits and prepares cleanly separated training test datasets.

STEP 6: Choose and Train the Model

To choose and train the model the following code was applied and hyperparameters tuned.

```
#Define Base Models
base_models = [
    ('lr', LogisticRegression(max_iter=500, class_weight='balanced', random_state=42)),
    ('rf', RandomForestClassifier(n_estimators=100, class_weight='balanced', random_state=42)),
    ('xgb', XGBClassifier(n_estimators=100,eval_metric='logloss', random_state=42))
]
#Meta Classifier
meta_model = LogisticRegression(max_iter=500)

#Stacking Model
stacking_model = StackingClassifier(estimators=base_models, final_estimator=meta_model, cv=3, passthrough=True)

#Full Pipeline with SMOTE
pipeline = ImbPipeline(steps=[
    ('scaler', StandardScaler()),
     #('smote', SMOTE(random_state=42, k_neighbors=6, sampling_strategy=0.2)), # Experiment with SMOTE parameters
    #('adasyn', ADASYN(random_state=42)),
    ('smotetomek', SMOTETomek(random_state=42)),
    ('model', stacking_model)
])

#Train the Model
pipeline.fit(X_train, y_train)

#Evaluate on Test Data
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
y_pred = pipeline.predict(X_test)
y_proba = pipeline.predict_proba(X_test)[:, 1]
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

#AUPRC
precision, recall, _ = precision_recall_curve(y_test, y_proba)
auprc = auc(recall, precision)
print(f"AUPRC: {auprc:.4f}")
# Plot Precision-Recall Curve
plt.figure(figsize=(8,6))
plt.plot(recall, precision, marker='.')
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.title('Precision-Recall Curve')
plt.grid()
plt.show()
print("\nAccuracy Score:", accuracy_score(y_test, y_pred))
#ROC-AUC SCORE
print("ROC-AUC Score:", roc_auc_score(y_test, y_pred))
plt.show()
```

Figure 6. 1 screenshot of code to train the model

The code above builds and evaluates a stacked ensemble classification model within a full pipeline designed to handle imbalance dataset in fraud detection. The base models are clearly defined, and the meta_classifier specified. The model is then trained using the command **Pipeline.fit(x_train, y_train)** and later evaluated on the test data. Lastly the performance metrics are defined.

STEP 7: Make Prediction and Evaluate the Model

Making predictions and evaluating the model is critical in machine learning as it reveals how well the model will perform on new, unseen data. In order to predict and evaluate the model the following line of code was applied:

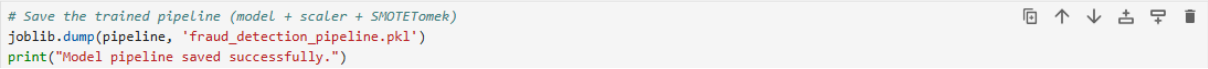
```
#Evaluate on Test Data
y_pred = pipeline.predict(X_test)
y_proba = pipeline.predict_proba(X_test)[:, 1]
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

#AUPRC
precision, recall, _ = precision_recall_curve(y_test, y_proba)
auprc = auc(recall, precision)
print(f"AUPRC: {auprc:.4f}")
# Plot Precision-Recall Curve
plt.figure(figsize=(8,6))
plt.plot(recall, precision, marker='.')
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.title('Precision-Recall Curve')
plt.grid()
plt.show()
# 10. Cross-validation scores
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
cv_preds = cross_val_predict(pipeline, X, y, cv=cv, method='predict')
print("Cross-validated Classification Report:\n", classification_report(y, cv_preds))
```

Figure 7. 1 screenshot of code to predict and evaluate the model

The above line of code evaluates both test set performance and cross-validation performance. It applies metrics and visuals for thorough analysis. The cross-validation score uses the StratifiedFold to preserve class distribution, and cross_val_predict() is used to make predictions and lastly prints a classification report based on all cross-validation predictions.

STEP 8: Save the Model

To the save the model in readiness for deployment the following line of code were used as shown below:



```
# Save the trained pipeline (model + scaler + SMOTETomek)
joblib.dump(pipeline, 'fraud_detection_pipeline.pkl')
print("Model pipeline saved successfully.")
```

Figure 8. 1 Saving the model

The above line of code used joblib to save the model under the name 'fraud_detection_pipeline.pk1'.

STEP 9: Creating the API to Load and Test the Model

The API to load and test the model's accuracy was created using the line of code below:

Figure 9. 1 screenshot of code to create an API

The above line of code uses Flask to load the model and defines the features to be used during testing.

STEP 10: Running the Model

To run the app.py as a background process and interact with the model using the API the following line of code below were applied

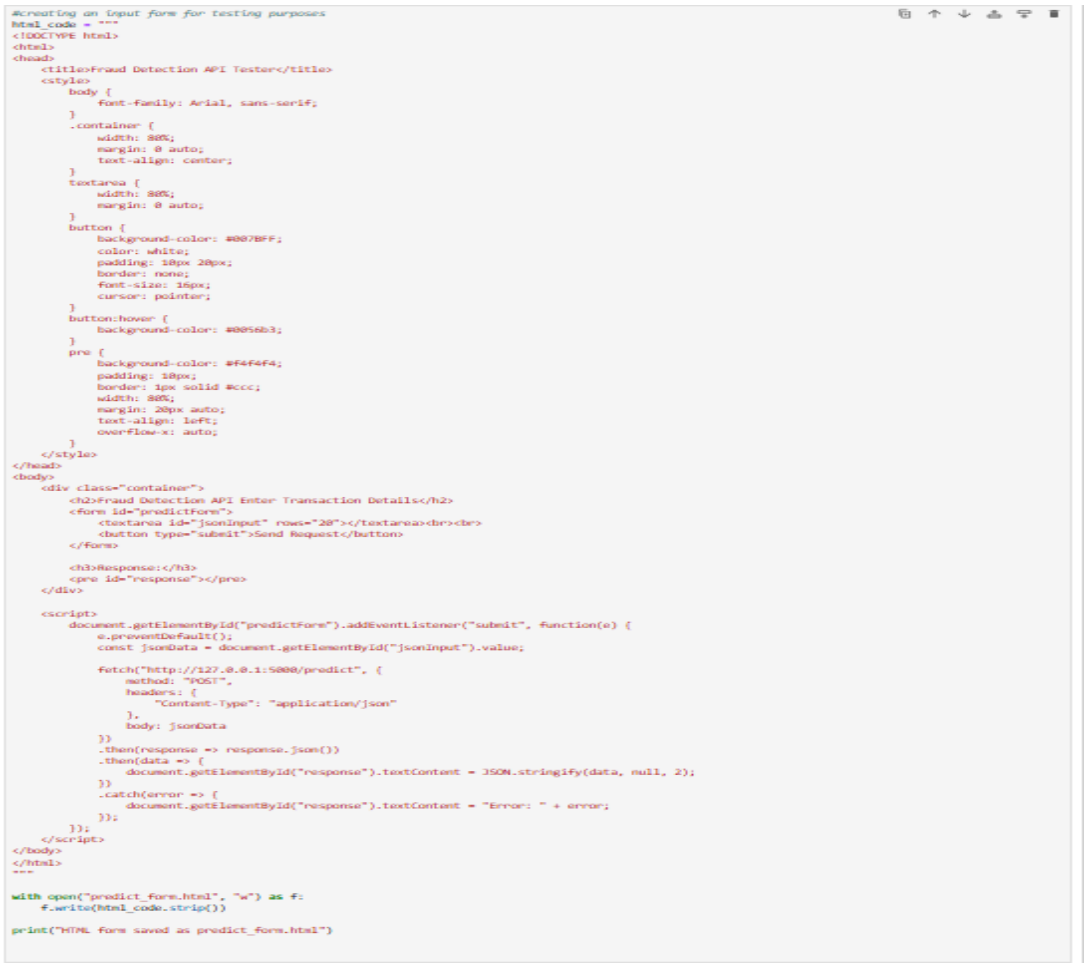
```
import subprocess

# Run app.py as a background process
process = subprocess.Popen(["python", "app.py"])

# Optional: Print the process ID to confirm it's running
print(f"Flask app is running with PID: {process.pid}")
```

```
Flask app is running with PID: 8776
```

Figure 10. 1 screenshot of code to run the model

STEP 11: Creating the form for Inputting Transactions

The form to enter transaction information for testing purposes was created and linked to the model using the API as shown in the code below:



```
#creating an input form for testing purposes
html_code = """
<!DOCTYPE html>
<html>
<head>
    <title>Fraud Detection API Tester</title>
    <style>
        body {
            font-family: Arial, sans-serif;
        }
        .container {
            width: 80%;
            margin: 0 auto;
            text-align: center;
        }
        textarea {
            width: 80%;
            margin: 0 auto;
        }
        button {
            background-color: #007BFF;
            color: white;
            padding: 10px 20px;
            border: none;
            font-size: 16px;
            cursor: pointer;
        }
        button:hover {
            background-color: #0056b3;
        }
        pre {
            background-color: #f4f4f4;
            padding: 10px;
            border: 1px solid #ccc;
            width: 80%;
            margin: 20px auto;
            text-align: left;
            overflow-x: auto;
        }
    </style>
</head>
<body>
    <div class="container">
        <h2>Fraud Detection API Enter Transaction Details</h2>
        <form id="predictForm">
            <textarea id="jsonInput" rows="20"></textarea><br><br>
            <button type="submit">Send Request</button>
        </form>

        <h3>Response:</h3>
        <pre id="response"></pre>
    </div>

    <script>
        document.getElementById("predictForm").addEventListener("submit", function(e) {
            e.preventDefault();
            const jsonData = document.getElementById("jsonInput").value;

            fetch("http://127.0.0.1:5000/predict", {
                method: "POST",
                headers: {
                    "Content-Type": "application/json"
                },
                body: jsonData
            })
            .then(response => response.json())
            .then(data => {
                document.getElementById("response").textContent = JSON.stringify(data, null, 2);
            })
            .catch(error => {
                document.getElementById("response").textContent = "Error: " + error;
            });
        });
    </script>
</body>
</html>
"""

with open("predict_form.html", "w") as f:
    f.write(html_code.strip())

print("HTML form saved as predict_form.html")
```

Figure 11. 1 screenshot of code to create an input form for testing

STEP 12: Retrieving Transactions for Testing

To retrieve sample transactions from the test data the following code was used:

```
# Retrieving the transactions for testing the model
import pandas as pd
from sklearn.model_selection import train_test_split

# Load your dataset
df = pd.read_csv("C:/Users/USER/Fraud/creditcard.csv")

# Split features and labels
X = df.drop("Class", axis=1)
y = df["Class"]

# Split into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Get the first 10 rows from the test set
samples = X_test.iloc[:10]

# Convert to a list of dictionaries
sample_list = samples.round(6).to_dict(orient="records")

# Display or use them
import json
print(json.dumps(sample_list, indent=2))
```

Figure 12. 1 screenshot of code to retrieve test transactions from test dataset

The above line of code retrieved about 10 transaction records from the unseen test dataset as shown in the code **samples = x_test.iloc[:10]**

STEP 13: Stress Testing / Bulk Processing

To test how the model perform when bulk transactions are submitted the following code was used:

```
import requests
import random
from concurrent.futures import ThreadPoolExecutor

# Prepare a sample transaction
features = ['Time'] + [f'V{i}' for i in range(1, 29)] + ['Amount']

def generate_transaction():
    return {feature: round(random.uniform(-5, 5), 6) for feature in features}

# Function to send a request
def send_request():
    try:
        response = requests.post("http://127.0.0.1:5000/predict", json=generate_transaction())
        return response.status_code, response.elapsed.total_seconds()
    except Exception as e:
        return "Error", str(e)

# Run with many parallel threads
num_requests = 500
with ThreadPoolExecutor(max_workers=20) as executor:
    results = list(executor.map(lambda _: send_request(), range(num_requests)))

# Analyze results
successes = [r for r in results if r[0] == 200]
failures = [r for r in results if r[0] != 200]

print(f"Total Requests: {num_requests}")
print(f"Successful: {len(successes)}")
print(f"Failed: {len(failures)}")
print("Average Response Time (s):", round(sum([r[1] for r in successes]) / len(successes), 4) if successes else "N/A")
```

```
Total Requests: 500
Successful: 500
Failed: 0
Average Response Time (s): 0.4878
```

Figure 13. 1 screenshot of code for offline stress testing.

As shown above 500 transactions were submitted at once for processing and the model processed all the transactions in 0.4878 seconds.

4.9 Chapter Summary

This chapter presented the hybrid machine learning approach applied in detecting credit/debit card fraud. The stacked ensemble integrates Logistic Regression, Random Forest, and XGBoost to improve accuracy and generalizability. The prototype was built with scalability and modularity in mind, ensuring high detection performance and interpretability. The outcome from the experiments validate how effective the stacked ensemble model is.

In the following chapter, results of the research findings are presented and Analysis of Results and Performance Metrics is done. Comparison to Related Work is undertaken, and Implications of Results is discussed.

CHAPTER 5 - RESULTS AND DISCUSSIONS

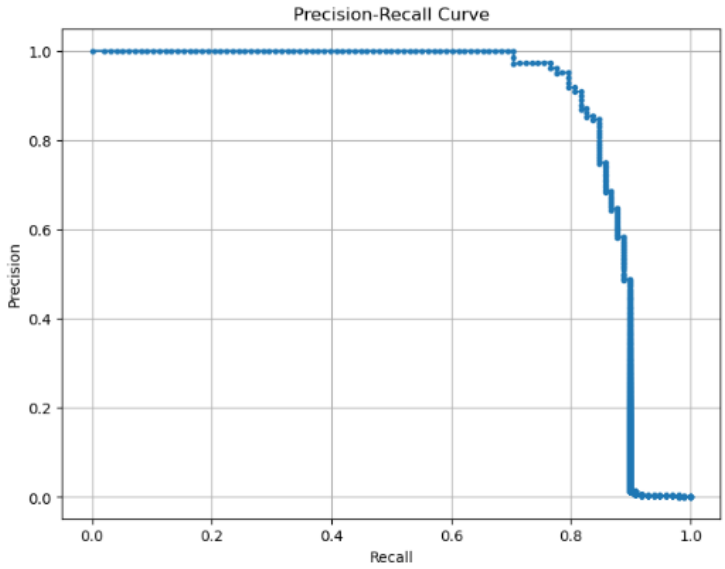
5.0 Results Presentation

Following the subsequent experimentation and implementation of the model in Chapter 4, results are presented as follows:

5.1 Model 1: Results

For this model the Base models, included Logistic regression, Random Forest classifier, and XGBClassifier. The meta_model was set to LogisticRegression with max_iteration set to 500. The full pipeline used the standard scaler and **SMOTETomek** to handle the imbalanced data set, and below were the results after training and evaluating the model.

```
Confusion Matrix:
 [[56850    14]
 [   17    81]]

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56864
           1       0.85      0.83      0.84        98

    accuracy                           1.00     56962
   macro avg       0.93      0.91      0.92     56962
weighted avg       1.00      1.00      1.00     56962

AUPRC: 0.8703
```




```
Accuracy Score: 0.9994557775359011
ROC-AUC Score: 0.9131422053908789
```

Figure 5.5. 1 Model 1 Results

5.2 Model 2: Results

For this model the Base models, included Logistic regression, Random Forest classifier, and XGBClassifier. The meta_model was set to LogisticRegression with max_iteration set to 500. The full pipeline used the standard scaler and **ADASYN** to handle the imbalanced data set, and below were the results after training and evaluating the model.

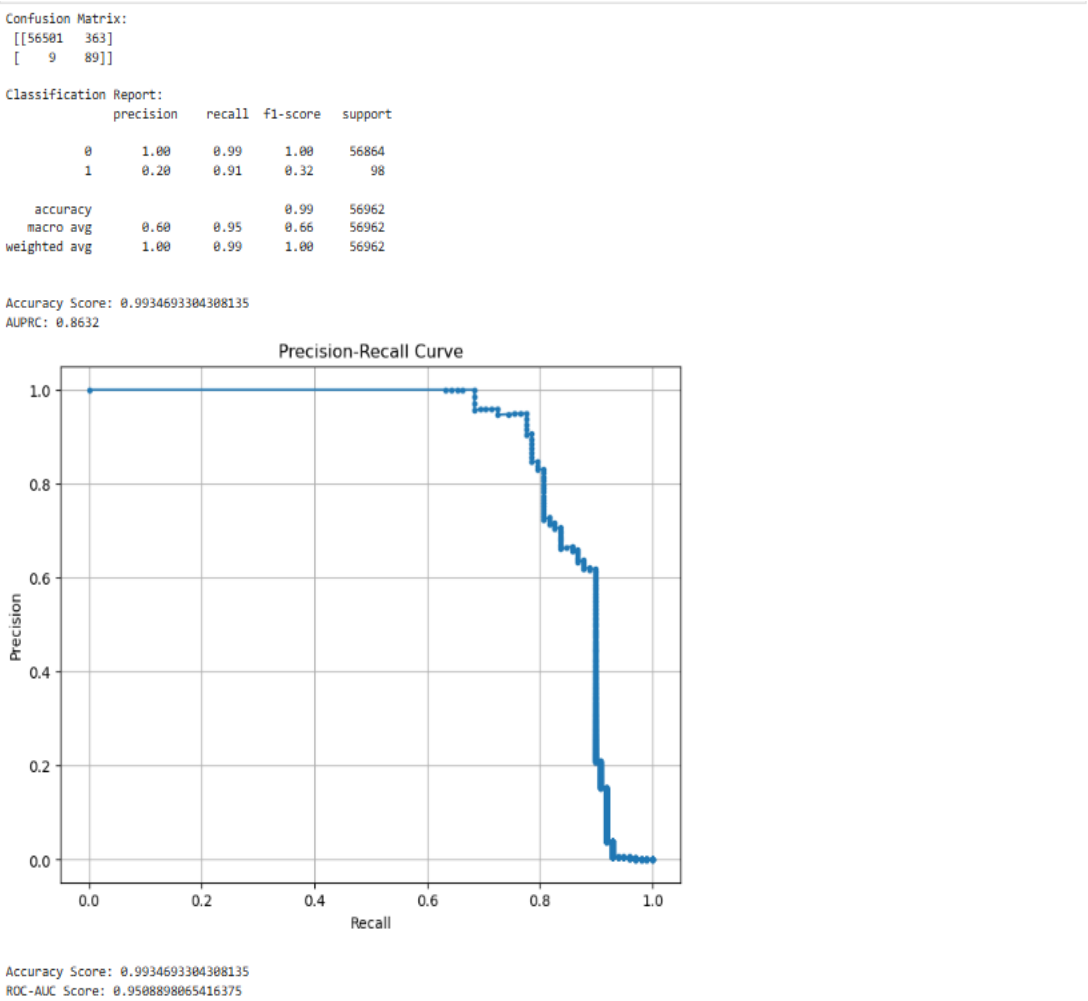


Figure 5.5. 2 Model 2 results

5.3 Model 3: Results

For this model the Base models, included Logistic regression, Random Forest classifier, and XGBClassifier. The meta_model was set to LogisticRegression with max_iteration set to 500. The full pipeline used the standard scaler and **SMOTE** to handle the imbalanced data set, and below were the results after training and evaluating the model.

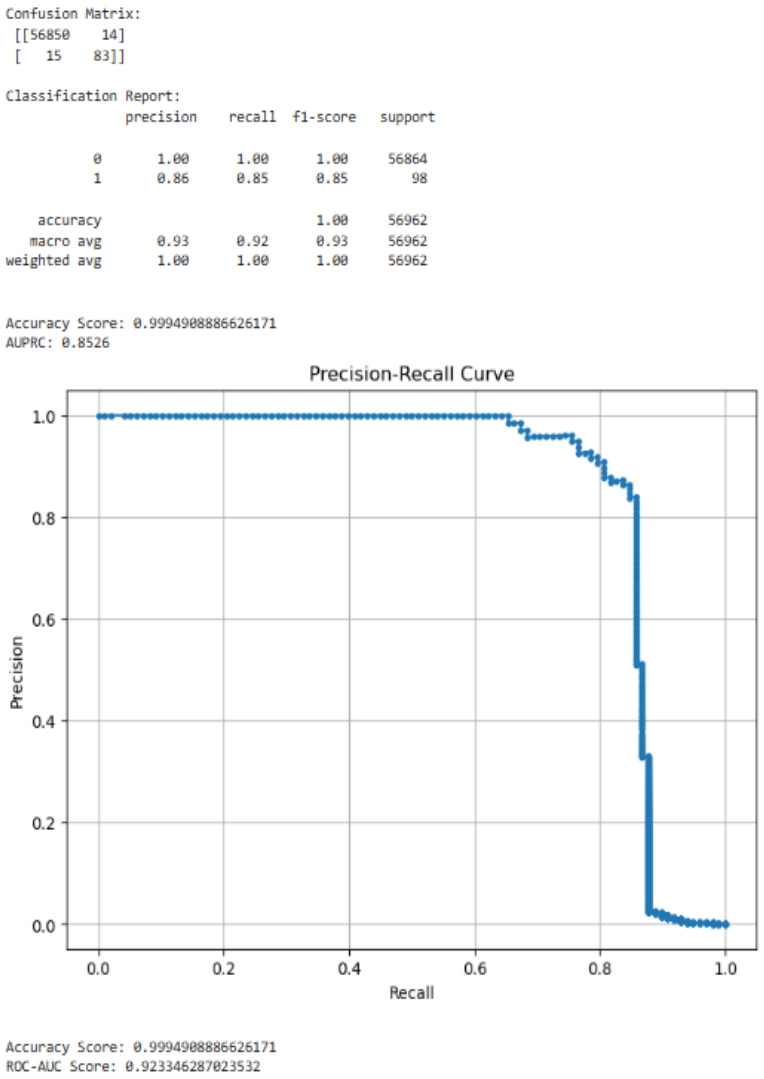


Figure 5.5. 3 Model 3 results

Summary of Results

METRIC	MAIN MODEL USING SMOTETomek	MODEL 2 USING ADASYN	MODEL 3 USING SMOTE
Accuracy	*0.9995*	*0.9935*	*0.9995*
Precision (class 1)	*0.85*	*0.20*	*0.86*
Recall (class 1)	*0.83*	*0.91*	*0.85*
F-1 Score (class 1)	*0.84*	*0.32*	*0.85*
ROC-AUC Score	*0.9131*	*0.9509*	*0.9233*
AUPRC	*0.8703*	*0.8632*	*0.8526*
False Positives (class 0 wrongly predicted as class 1)	*14*	*363*	*14*
False Negatives (class 1 wrongly predicted as class 0)	*17*	*9*	*15*

Table 5. 1 Summary of Results:

5.4.0 Analysis of Results/Performance Metrics

Based on the results presented in section 5.1, below is the analysis of the results and performance metrics of the models under experimentation.

5.4.1 Model 1: Main model (SMOTETomek)

This exhibits best performance overall and balance because of the following:

- The model has excellent precision and recall on the minority class (class 1).
- The model has very few false positives (14) and false negatives (17).
- The model has high AUPRC (0.8703) which is good for imbalanced classification
- The model has a strong balanced performance, and likely the best choice compared to the other two (2) models especially when both precision and recall matter.

5.4.2 Model 2 (ADASYN)

This model has high recall and poor precision and may not be a good choice because of the following:

- Recall is highest (0.91) which means it detects most of class 1 instances.
- But precision is extremely poor (0.20) indicating main false alarm (363 false positives).
- ROC-AUC is the highest (0.9509), but this is misleading due to poor precision.

The model with this performance can only be used in instances where catching every positive case is critical and false positives are accepted.

5.4.3 Model 3 (SMOTE)

This is similar to and the results are close to model 1 because of the following:

- It has slightly better recall (0.85) and fewer false positives (15).
- Precision is slightly higher than model 1(0.86).
- It has slightly lower AUPRC (0.8526), but still high.

This is also a strong model with slightly better recall than model 1.

5.5 Comparison to Related Works

Analysis of the performance of stacked ensemble models in this project was inspired by previous related works undertaken by different researchers. For instance, in the Journal of Artificial Intelligence Research 10 (1999) 271-289 [40] the outcome of the study reveal that stacking achieves higher accuracy by leveraging machine learning to learn optimal combination of different models. The paper looked at what type of top-level model works best to combine with lower models, and it turned out that using confidence levels from original models rather than just their predictions give better results.

The practical take away in another research [41] reveals that ignoring imbalance in dataset can lead to optimistic assessments of model performance that do not generalize well to the minority class. The paper discussed how the failure to address imbalance in dataset may lead to the risk of producing misleading estimates of a model's true performance.

5.6 Implications of Results

The implications of the results depend on the **use case**, the **costs of false positives and false negatives**, and the **requirements for model performance in production**. The implications based on the performance of the three models on experimentation are as follows:

a. Class Imbalance Handling Matters

Three different resampling techniques applied were SMOTETomek, ADASYN, and SMOTE. During experimentation, the results revealed that the choice of resampling strategy significantly affects precision and recall, especially for the minority class (class 1). ADASYN boosted recall but drastically affected precision. SMOTE and SMOTETomek gave a much better balance, suggesting they are more reliable for this dataset.

Implication:

To achieve reliable and fair model performance proper handling of class imbalance is very important, especially when the minority class carries important information (e.g., fraud detection, disease diagnosis).

b. Precision-Recall Tradeoff is Crucial

The results showed that Model 2 (ADASYN) had high recall (0.91) but very low precision (0.20), meaning many false positives. On the other hand Model 1 and 3 achieved a much better balance between precision and recall.

Implication:

If false positives are costly like flagging legitimate transactions as fraud, Model 2 is risky. But if missing a true positive is worse such as missing a cancer case, then Model 2 might be acceptable despite false alarms.

c. Metrics like Accuracy Can Be Misleading

As exhibited in the results, all three models show very high accuracy (99.3%–99.95%), but that is largely due to the overwhelming number of majority class samples (class 0). However, there is a difference in the minority class metrics precision, recall, and F1. This indicates that accuracy alone is not a reliable metric for imbalanced classification.

Implication:

To measure the models performance, there is a need to focus on **F1-score, precision, recall, AUPRC, and ROC-AUC** as they offer a more complete view of model performance under imbalance.

Observetions:

Unless the application demands maximum recall at all costs, Model 1 (SMOTETomek) or Model 3 (SMOTE) are better-suited for deployment due to:

- Balanced performance
- Low false positives and false negatives
- High F1-scores
- Stable ROC and PR metrics

5.7 Recommendations of Model Adaption to New Fraud Patterns:

Adapting a fraud detection system to new fraud patterns is very important because fraudsters constantly evolve their tactics. As revealed in this research, fraud pattern keeps evolving at a rapid pace and a static system quickly become outdated and ineffective.

Therefore, the following key strategies has to be applied in order to continuously adapt the fraud detection system:

1. Online Continuous Learning or Incremental Learning

This can be made possible when the model is implemented in such a way that it's able to learn continuously from new data without retraining from scratch. As revealed in [42], this can be achieved by using algorithms like online logistic regression, incremental decision trees, or frameworks like River (Python) for streaming ML[43].

2. Regular Model Retraining

The second way as shown in [42] is to continuously adapt the fraud detection systems is by:

- Scheduling periodic retraining (e.g., weekly, monthly) using recent labeled data to keep the model up-to-date.
- Using rolling windows to ensure the training data reflects current patterns [44].
- Implementing a model drift detection mechanism to trigger retraining when needed [44].

3. Anomaly Detection Layer

Adding an unsupervised anomaly detection module such as Isolation Forest, Autoencoders, One-Class SVM [42], these can assist the model to detect new fraud patterns not present in the training data, especially zero-day frauds.

5.8 Chapter Summary

In this chapter, results of the research findings were presented and analysed, further analysis of Performance Metrics were also done. Additionally, comparison to related work was undertaken, and the implications of results discussed. The next chapter, will highlight the summary of main Findings, Academic contribution to the body of knowledge, Limitations of the Research, and Future works.

CHAPTER 6 - SUMMARY AND CONCLUSION

6.0 Summary of Main Findings

The Projects main findings indicates that advanced AI and supervised machine learning models, such as those that are based on SMOTE and SMOTETomek resampling techniques, can attain high accuracy of over 90% in identifying fraudulent transactions. These models exhibit balanced performance with low false positives and false negatives, high F1-scores, and stable ROC-AUC and PR metrics. The study emphasizes that relying on accuracy alone is deceptive in imbalanced datasets. Instead, metrics like F1-score, precision, and recall provide a clearer evaluation of performance. All in all, the model developed is suitable for real-world deployment, offering efficient, interpretable, and adaptable fraud detection solution.

6.1 Discussion and Implications in Relation to Objectives

The study has achieved the primary objective of developing and implementing a machine learning model that is able to detect fraudulent transactions with high accuracy, exceeding 90%. By using techniques such as SMOTETomek, the model had demonstrated a balanced performance, effective handling of class imbalance, and low false positive and false negative rates. This aligns well with the objective of developing a reliable fraud detection system. The results also confirm that evaluating metrics beyond accuracy, such as F1-score, precision, and recall, is very important for accurate assessment in imbalanced datasets.

The implications of the findings are that implementing supervised machine learning models in real-world financial systems can improve fraud detection capabilities, reduce financial losses, and improve customer trust. Additionally, focusing on more comprehensive performance metrics ensures that the system remains effective and fair, especially in sensitive contexts where false negatives or false positives carry significant consequences. Generally, the research reinforce the importance of balanced model design and continuous adaptation to new fraud patterns to meet practical operational needs.

6.2 Academic Contribution to The Body of Knowledge

The study contributes to the academic body of knowledge by showing the effectiveness of advanced resampling techniques, such as SMOTE and SMOTETomek, in improving the performance of machine learning models for imbalanced fraud detection datasets. It offers a comparative analysis of multiple models, emphasizing the importance of evaluation metrics beyond accuracy, notably F1-score, precision, and recall, to assess model efficacy reliably. The research also adds insight to the understanding of balancing false positives and false negatives, guiding the development of more accurate AI-based fraud detection systems. The findings also add to the understanding of applying stacked ensemble strategies and performance metrics in financial fraud detection, promoting further research into more robust, fair, and ethically compliant AI applications in Finance.

6.3 Limitations of the System/Model

The study brings out several limitations of the proposed fraud detection system. First, the features used in the models are limited to those available in the dataset, excluding very important real-world indicators such as device ID, IP address, biometrics, or user behavior, which could improve detection accuracy. Secondly, system's testing was done in an offline environment. Meaning issues related to real-time detection speed, system integration, stress testing, and response time were not addressed. The outlined limitations may impact the system's effectiveness and deployment readiness in live production environments. Also, the challenges related to data privacy concerns and possibility biases in the training data are acknowledged as serious limitations that could affect the fairness and ethical deployment of the system.

Lastly finding a dataset that addresses all types of fraud across Financial Institutions proved to be practically impossible. Therefore, the model is limited to detecting Fraud related with POS transactions, credit/debit card based transaction and online purchasing.

6.4 Future Works

In the future works, there is a need to focus on studying how the model can be integrated in the payment gateway with the view of improving the system's capabilities for real-time data analysis and deployment. Specifically, the system needs to be adapted for real-time detection to enable immediate alerts and responses to fraud activities. Furthermore, research works on the inclusion of real-world features such as device information, IP addresses, user behavior, and biometric data to improve detection accuracy is needed. Lastly, operational challenges related to real-time stress testing need to be addressed, this will ensure robustness in production environments.

6.5 Chapter Summary

This Chapter delved into highlighting the summary of main Findings, and brought out Academic contributions to the body of knowledge. Limitations of the Research were acknowledged, and concluded with suggested Future works

REFERENCES

[1] Oluwatoyin Ajoke Farayola, "Revolutionizing Banking Security: Integrating Artificial Intelligence, Blockchain, and Business Intelligence for Enhanced Cybersecurity," *Financ. Account. Res. J.*, vol. 6, no. 4, pp. 501–514, 2024, doi: 10.51594/farj.v6i4.990.

[2] A. Yuille, "AI-Powered Financial Services : Enhancing Fraud Detection and Risk Assessment AI-Powered Financial Services : Enhancing Fraud Detection and Risk Assessment with Predictive Analytics," no. August, 2024, doi: 10.13140/RG.2.2.23580.09603.

[3] A. S. George, "Securing the Future of Finance: How AI, Blockchain, and Machine Learning Safeguard Emerging Neobank Technology Against Evolving Cyber Threats," *Partners Univers. Innov. Res. Publ.*, vol. 1, no. 1, pp. 54–66, 2023, doi: 10.5281/zenodo.10001735.

[4] N. Khalid, "AI-Powered Fraud Detection and Risk Assessment : The Future of Financial Services," no. September, 2024, doi: 10.13140/RG.2.2.32932.08324.

[5] O. Adijat Bello, A. Ogundipe, D. Mohammed, A. Folorunso, and O. Ayodeji Alonge, "AI-Driven Approaches for Real-Time Fraud Detection in US Financial Transactions: Challenges and Opportunities," *Eur. J. Comput. Sci. Inf. Technol.*, vol. 11, no. 6, pp. 84–102, 2023, doi: 10.37745/ejcsit.2013/vol11n684102.

[6] Philip Olaseni Shoetan and Babajide Tolulope Familoni, "Transforming Fintech Fraud Detection With Advanced Artificial Intelligence Algorithms," *Financ. Account. Res. J.*, vol. 6, no. 4, pp. 602–625, 2024, doi: 10.51594/farj.v6i4.1036.

[7] A. I. M. P. E-journal, "Vidhyayana - ISSN 2454-8596," vol. 8, no. 7, pp. 79–95.

[8] S. Mishra, "Exploring the Impact of AI-Based Cyber Security Financial Sector Management," *Appl. Sci.*, vol. 13, no. 10, 2023, doi: 10.3390/app13105875.

[9] M. L. Algorithms and B. Analytics, "American Journal of Economics and Pioneering AI-Driven Fraud Detection and AML Strategies : Transforming Azerbaijan ' s Banking Landscape through Innovative Machine Learning Algorithms and Behavioral Analytics," vol. 7, no. 4, pp. 31–36, 2024.

[10] A. Sina, "Open AI and its Impact on Fraud Detection in Financial Industry," *J. Knowl. Learn. Sci. Technol. ISSN 2959-6386*, vol. 2, no. 3, pp. 263–281, 2024, doi: 10.60087/jklst.vol2.n3.p281.

[11] E. Carter, J. Anderson, E. Carter, and J. Anderson, "EasyChair Preprint Optimizing Financial Services with AI : Enhancing Risk Management and Strategic Decision Making Optimizing Financial Services with AI : Enhancing Risk Management and Strategic Decision Making," 2024.

[12] B. Mytnyk, O. Tkachyk, N. Shakhovska, S. Fedushko, and Y. Syerov, "Application of Artificial Intelligence for Fraudulent Banking Operations Recognition," *Big Data Cogn. Comput.*, vol. 7, no. 2, 2023, doi: 10.3390/bdcc7020093.

[13] A. B. Malali and S. Gopalakrishnan, "Application of Artificial Intelligence and Its Powered Technologies in the Indian Banking and Financial Industry: An Overview," *IOSR J. Humanit. Soc. Sci. (IOSR-JHSS*, vol. 25, no. 6, pp. 55–60, 2020, doi: 10.9790/0837-2504065560.

[14] R. Vp, "Role of Cognitive Technologies and Ai in Banking".

[15] R. Ahmed, "Artificial Intelligence (AI) in Financial Sectors : Blessings or Threats ?," vol. 23, no. 3, pp. 20–25, 2021, doi: 10.9790/487X-2303012025.

[16] A. Gautam, "The evaluating the impact of artificial intelligence on risk management and fraud detection in the banking sector," *AI, IoT Fourth Ind. Revolut. Rev.*, vol. 13, no. 11, pp. 9–18, 2023.

[17] C. Vijai and S. Natarajan, "Yugato artificial intelligence, machine learning and big data in finance," no. May, 2024.

[18] C. Lloyd, M. R. Misheal, and N. Tavonga, "Harnessing Machine Learning and Artificial Intelligence for Ear- ly Fraud Detection Among Banks in Harare, Zimbabwe: Internal Auditors' Perspective," *Met Manag. Rev.*, vol. 11, no. 01, pp. 01–11, 2024, doi: 10.34047/mmr.2024.111.

[19] H. Arsalan, "FinTech Futures: AI-Driven Payments and Supply Chain Innovations," no. April, 2024, doi: 10.13140/RG.2.2.16246.41282.

[20] Oseremi Onesi-Ozigagun, Yinka James Ololade, Nsisong Louis Eyo-Udo, and Damilola Oluwaseun Ogundipe, "AI-driven biometrics for secure fintech: Pioneering safety and trust," *Int. J. Eng. Res. Updat.*, vol. 6, no. 2, pp. 001–012, 2024, doi: 10.53430/ijeru.2024.6.2.0023.

[21] K. Reddy, "Artificial Intelligence Advantages in Cloud Fintech Application Security," *Cent. Asian J. Math. Theory Comput. Sci.*, vol. 4, no. 8, pp. 48–53, 2023, [Online]. Available: https://cajmtcs.centralasianstudies.org

[22] I. Caprian, "The Application of Artificial Intelligence for the Purpose of Combating Bank Fraud," *Probl. Econ.*, vol. 2, no. 56, pp. 204–212, 2023, doi: 10.32983/2222-0712-2023-2-204-212.

[23] H. Nobanee, "Fatima Alzaabi," no. October, 2021.

[24] M. A. Galdinus, S. Vinoth, and C. Gopalakrishna, "Investigating the Utilization and Efficacy of Artificial Intelligence in the Indian Banking Industry," *Int. Res. J. Mod. Eng. Technol. Sci.*, no. 03, pp. 3484–3492, 2023, doi: 10.56726/irjmets35089.

[25] N. D. Kulkarni and S. Bansal, "The Impact of Artificial Intelligence in Banking and Finance Sector A Review," *Int. Res. J. Mod. Eng. Technol. Sci.*, no. 01, pp. 4036–4043, 2024, doi: 10.56726/irjmets49043.

[26] A. Varma, "Revolution of Financial Services: Analysing the Impact of Ai on Banking and Investment," vol. 19, no. 12, pp. 565–575, 2021, doi: 10.48047/nq.2021.19.12.NQ21256.

[27] M. J. Madhurya, H. L. Gururaj, B. C. Soundarya, K. P. Vidyashree, and A. B. Rajendra,

"Exploratory analysis of credit card fraud detection using machine learning techniques," *Glob. Transitions Proc.*, vol. 3, no. 1, pp. 31–37, 2022, doi: 10.1016/j.gltp.2022.04.006.

[28] F. Mossavar-rahmani, "Journal of Engineering and Applied The Transformative Impact of AI on Financial Institutions , with a Focus on Banking," vol. 2023, no. December, 2023, doi: 10.47363/JEAST/2023(5)192.

[29] M. Devan, S. Prakash, S. Jangoan, and M. Devan, "Predictive Maintenance in Banking : Leveraging AI for Real-Time Data Analytics Fidelity Investments , USA," vol. 2, no. 2.

[30] T. Baabdullah, A. Alzahrani, and D. B. Rawat, "Efficiency of Federated Learning and Blockchain in Preserving Privacy and Enhancing the Performance of Credit Card Fraud Detection (CCFD) Systems," 2024.

[31] V. Durga and P. Sambrow, "Integrating Artificial Intelligence in Banking Fraud Prevention : A Focus on Deep Learning and Data Analytics," vol. 6, no. 1, 2022.

[32] N. Jahnavi and K. V Tibrewal, "Banking Sector Transformation-Artificial Intelligence In The Modern Digital Era," *Int. J. Adv. Eng. Manag.*, vol. 2, no. 9, p. 404, 2008, doi: 10.35629/5252-0209404411.

[33] B. P. Zanke, "AI-Driven Fraud Detection Systems : A Comparative Study across Banking , Insurance , and Healthcare," vol. 3, no. 2, pp. 1–22, 2023.

[34] M. Lokanan, V. Tran, and N. H. Vuong, "Detecting anomalies in financial statements using machine learning algorithm: The case of Vietnamese listed firms," *Asian J. Account. Res.*, vol. 4, no. 2, pp. 181–201, 2019, doi: 10.1108/AJAR-09-2018-0032.

[35] U. Kingdom, M. Kamal, and A. Ismaeil, "Harnessing AI for Next-Generation Financial Fraud Detection : A Data- Driven Revolution," pp. 811–821, 2024.

[36] N. Mohammad, M. A. U. Imran, M. Prabha, S. Sharmin, and R. Khatoon, "Combating Banking Fraud With It: Integrating Machine Learning and Data Analytics," *Am. J. Manag. Econ. Innov.*, vol. 6, no. 7, pp. 39–56, 2024, doi: 10.37547/tajmei/volume06issue07-04.

[37] L. Anthony, C. Maimuna, P. Ordóñez, and J. Sebastian, *Leveraging Data Science for Global Health*.

[38] V. Goar and N. S. Yadav, *Foundations of machine learning*. 2024. doi: 10.4018/979-8-3693-1598-9.ch002.

[39] M. Valavan and S. Rita, "Predictive-Analysis-based Machine Learning Model for Fraud Detection with Boosting Classifiers," *Comput. Syst. Sci. Eng.*, vol. 45, no. 1, pp. 231–245, 2023, doi: 10.32604/csse.2023.026508.

[40] K. M. Ting and I. H. Witten, "Issues in stacked generalization," *J. Artif. Intell. Res.*, vol. 10, pp. 271–289, 1999, doi: 10.1613/jair.594.

[41] L. Pasteur and R. Koch, "The Supervised Learning No-Free?Lunch Theorems," vol. 74, no. 1934, pp. 535–546, 1941.

[42] P. Bhattacharya, S. Wagner, and M. Haug, "Concept Drift Detection and adaptation for machine learning," 2022.

[43] J. Montiel *et al.*, “River: Machine learning for streaming data in python,” *J. Mach. Learn. Res.*, vol. 22, pp. 1–8, 2021.

[44] A. R. Abdul, N. C. R, S. B. R, H. Lahza, and H. F. M. Lahza, “A survey on detecting healthcare concept drift in AI/ML models from a finance perspective,” *Front. Artif. Intell.*, vol. 5, 2022, doi: 10.3389/frai.2022.955314.

APPENDICES

Appendix A: Project Proposal

SCHOOL OF COMPUTING, TECHNOLOGY & APPLIED SCIENCES

MSc Computer Science

Project Proposal

AI-Powered Fraud Detection and Prevention in Financial Institutions

Proposal Submission Date: 10th March, 2025

Proposal Approval Date: 24th March, 2025

1.0 Project Background

Fraud detection has been a critical concern in financial institutions for decades. Conventional methods of detecting fraud have relied on rule-based systems, manual audits, and statistical techniques. These methods aimed to identify fraudulent activities such as money laundering, identity theft, and transaction fraud.

One of the earliest approaches to fraud detection was manual review, where financial analysts examined transactions and account activities for irregularities. This method, however, was labor-intensive and vulnerable to human error, making it ineffective for handling large volumes of financial data.

As technology advanced, rule-based systems became a standard approach. These systems operate on predefined rules set by financial experts to flag suspicious transactions. For example, a rule might trigger an alert if a transaction is above a certain amount or occurs in a high-risk location. While rule-based systems improved efficiency, they had drawbacks, including high false-positive rates and an inability to keep-up with emerging fraud patterns.

Statistical methods, such as regression analysis and anomaly detection, were also employed to enhance fraud detection. These techniques used historical data to identify variations from normal transaction pattern. However, statistical models required constant updates and were limited in detecting sophisticated fraud schemes that evolved over time.

Despite their effectiveness to some extent, traditional fraud detection methods struggled with scalability, adaptability, and accuracy in detecting potential fraud. With the rise of digital transactions and more complex fraud tactics, financial institutions are progressively turning to advanced innovations such as machine learning and artificial intelligence to improve fraud detection capabilities.

This research project aims to review the current state of fraud detection systems in financial institutions, design and implement an AI-powered fraud detection and prevention model that can prevent fraudulent transactions in real-time. By taking advantage of advanced AI and ML techniques, the model will improve the accuracy and efficiency of fraud detection, reduce false positives, and provide real-time monitoring and alerts. The project will also focus on creating a user-friendly interface for testing and ensure seamless adoption and usability by financial institutions.

1.2 Problem Statement

The existing fraud detection systems in banks and financial institutions may not effectively adapt to evolving fraud techniques and patterns. Conventional fraud detection methods, which rely heavily on rule-based systems and manual reviews, are becoming increasingly ineffective in flagging and preventing sophisticated fraudulent activities. These methods often fail to adapt to the rapidly changing tactics used by fraudsters, leading to serious financial losses, compromised customer trust, and heightened operational risks.

To handle these challenges, this research project proposes the design and implementation of an AI-driven fraud detection and prevention model that will be able to detect and stop fraudulent transactions as they occur. By using machine learning algorithms and artificial intelligence, the proposed system aims to improve the accuracy and speed of fraud detection, adapt to new and evolving fraud patterns, and give alerts as fraudulent transactions occur. This system will help financial institutions mitigate the risks associated with credit and debit card related fraud, protect their assets and customers, and maintain trust and integrity in their operations.

1.3 Aim and Objectives of the Project

1.3.1 Aim

- The aim of the project is to investigate the current state of fraud detection systems in financial institutions, and to explore the application of AI to improve the accuracy and efficiency of Debit and Credit card fraud detection and prevention in financial institutions.

1.3.2 Objectives of the project

The objectives of the project are to:

- develop and implement a machine learning model that can detect and prevent fraudulent transactions with at least 90% accuracy.
- evaluate the system's performance in terms of Accuracy, Precision, and Recall.
- develop strategies for continuously adapting the system to emerging fraud patterns.

2.0 Proposed model

This project proposes the development of a supervised machine learning model to predict credit/debit card fraud with at least 90% accuracy. The model shall integrate:

- **Logistic Regression (LR):** This will be use as a baseline classifier, offering interpretability for fraud probability estimation.
- **Random Forest (RF):** This will improve fraud detection by reducing variance and handling non-linearity in transaction data.
- **XGBoost:** This will utilize boosting techniques to improve classification performance and increase model generalization.
- **Stacked Ensemble Learning:** This will be used in combining predictions from LR, RF, and XGBoost using a meta-classifier to enhance decision-making.

The proposed approach will leverage ensemble learning to maximize fraud detection accuracy.

3.0 Project Deliverables:

The Fraud detection Model will be designed and developed as proposed above. The project deliverables are as follows:

- A Functional model that is able to detect fraud.
- The source code used to develop the model.
- Dashboard for testing the model.
- Project report.

4.0 Assumptions and Constraints

4.1 Assumptions

It is assumed that:

- There is sufficient historical fraud data available for model training.
- Compliance with financial regulations.
- Adequate infrastructure is in place to support AI development.

4.2 Constraints

The following are identified constraints:

- Regulatory and legal restrictions on data usage.
- Time constraints for project completion.

5.0 Risks and Mitigation Strategies

5.1 Key Risks

The following are identified risks

- Data Privacy Issues: Risk of violating customer data protection laws.
- Model Bias: Risk of biased AI models leading to unfair fraud detection.
- False Positives: Over-flagging legitimate transactions as fraudulent.

5.2 Mitigation Strategies

Below are the strategies to mitigate identified risks

- Develop and test the model under strict offline Lab environment using Jupyter notebook.
- Regular model evaluation and bias mitigation techniques.
- Optimize algorithms to minimize false positives.

6.0 Project Time Line and Milestone

Phase	Milestone	Estimated Completion
Initiation	Project approval	31st March 2025
Planning	Literature review, requirement gathering & methodology design	30th April 2025
Development	Experiments and Model implementation	26th May 2025
API	API development & integration with the model	30th May 2025
Model Testing	Testing & evaluation of the model	1st June 2025
Project Report	Finalizing the project report	27th June 2025
Project Submission	Submission of the project	29th June 2025

7.0 Project Stake Holders

- ZCAS University

Appendix B: Line of code

```
#Import needed Libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, cross_val_predict, StratifiedKFold
from sklearn.metrics import classification_report, confusion_matrix, precision_recall_curve, auc
from sklearn.metrics import roc_auc_score
from sklearn.pipeline import Pipeline
from sklearn.ensemble import StackingClassifier, RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from xgboost import XGBClassifier
from sklearn.preprocessing import StandardScaler
from imblearn.over_sampling import SMOTE
from imblearn.pipeline import Pipeline as ImbPipeline
import matplotlib.pyplot as plt
import seaborn as sns
from imblearn.combine import SMOTETomek
from imblearn.over_sampling import SMOTE, ADASYN
import joblib
```

```
# Load Data
df = pd.read_csv('C:/Users/USER/Fraud/creditcard.csv')
X = df.drop('Class', axis=1)
y = df['Class']
```

```
df.head(10)
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0
5	2.0	-0.425966	0.960523	1.141109	-0.168252	0.420987	-0.029728	0.476201	0.260314	-0.568671	...	-0.208254	-0.559825	-0.026398	-0.371427	-0.232794	0.105915	0.253844	0.081080	3.67	0
6	4.0	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.005159	0.081213	0.464960	...	-0.167716	-0.270710	-0.154104	-0.780055	0.750137	-0.257237	0.034507	0.005168	4.99	0
7	7.0	-0.644269	1.417964	1.074380	-0.492199	0.948934	0.428118	1.120631	-3.807864	0.615375	...	1.943465	-1.015455	0.057504	-0.649709	-0.415267	-0.051634	-1.206921	-1.085339	40.80	0
8	7.0	-0.894286	0.286157	-0.113192	-0.271526	2.669599	3.721818	0.370145	0.851084	-0.392048	...	-0.073425	-0.268092	-0.204233	1.011592	0.373205	-0.384157	0.011747	0.142404	93.20	0
9	9.0	-0.338262	1.119593	1.044367	-0.222187	0.499361	-0.246761	0.651583	0.069539	-0.736727	...	-0.246914	-0.633753	-0.120794	-0.385050	-0.069733	0.094199	0.246219	0.083076	3.68	0

10 rows × 31 columns

```
print(df.isnull().sum())
```

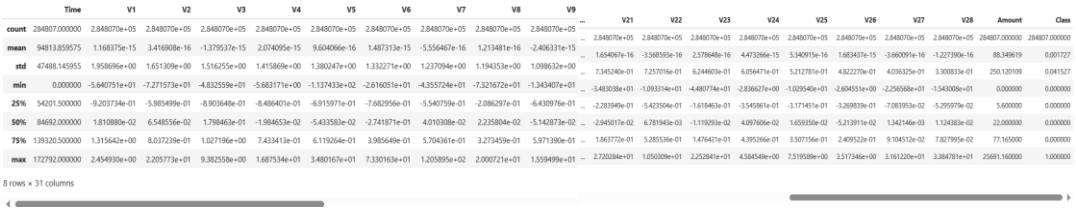
```
Time      0
V1        0
V2        0
V3        0
V4        0
V5        0
V6        0
V7        0
V8        0
V9        0
V10       0
V11       0
V12       0
V13       0
V14       0
V15       0
V16       0
V17       0
V18       0
V19       0
V20       0
V21       0
V22       0
V23       0
V24       0
V25       0
V26       0
V27       0
V28       0
Amount    0
Class     0
dtype: int64
```

```
df.isnull()
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
284802	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
284803	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
284804	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
284805	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
284806	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False

284807 rows × 31 columns

```
df.describe()
```



	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	...	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	284807.000000	284807.000000
mean	94813.859575	1.168375e-15	3.416908e-16	-1.379537e-15	2.074095e-15	9.604066e-16	1.487313e-15	-5.556467e-16	1.213481e-16	-2.406331e-15	...	1.654067e-16	-3.568593e-16	2.578648e-16	4.473266e-15	5.340915e-16	1.683437e-15	-3.660091e-16	-1.227390e-16	88.349619	0.001727
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00	1.380247e+00	1.332271e+00	1.237094e+00	1.194353e+00	1.098632e+00	...	7.345240e-01	7.257016e-01	6.244603e-01	6.056471e-01	5.212781e-01	4.822270e-01	4.036325e-01	3.300833e-01	250.120109	0.041527
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00	-1.137433e+02	-2.616051e+01	-4.355724e+01	-7.321672e+01	-1.343407e+01	...	-3.483038e+01	-1.093314e+01	-4.480774e+01	-2.836627e+00	-1.029540e+01	-2.604551e+00	-2.256568e+01	-1.543008e+01	0.000000	0.000000
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01	-7.682956e-01	-5.540759e-01	-2.086297e-01	-6.430976e-01	...	-2.283949e-01	-5.423504e-01	-1.618463e-01	-3.545861e-01	-3.171451e-01	-3.269839e-01	-7.083953e-02	-5.295979e-02	5.600000	0.000000
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02	-2.741871e-01	4.010308e-02	2.235804e-02	-5.142873e-02	...	-2.945017e-02	6.781943e-03	-1.119293e-02	4.097606e-02	1.659350e-02	-5.213911e-02	1.342146e-03	1.124383e-02	22.000000	0.000000
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01	3.985649e-01	5.704361e-01	3.273459e-01	5.971390e-01	...	1.863772e-01	5.285536e-01	1.476421e-01	4.395266e-01	3.507156e-01	2.409522e-01	9.104512e-02	7.827995e-02	77.165000	0.000000
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01	7.330163e+01	1.205895e+02	2.000721e+01	1.559499e+01	...	2.720284e+01	1.050309e+01	2.252841e+01	4.584549e+00	7.519588e+00	3.517346e+00	3.161220e+01	3.384781e+01	25691.160000	1.000000

8 rows × 31 columns

```
df.shape
```

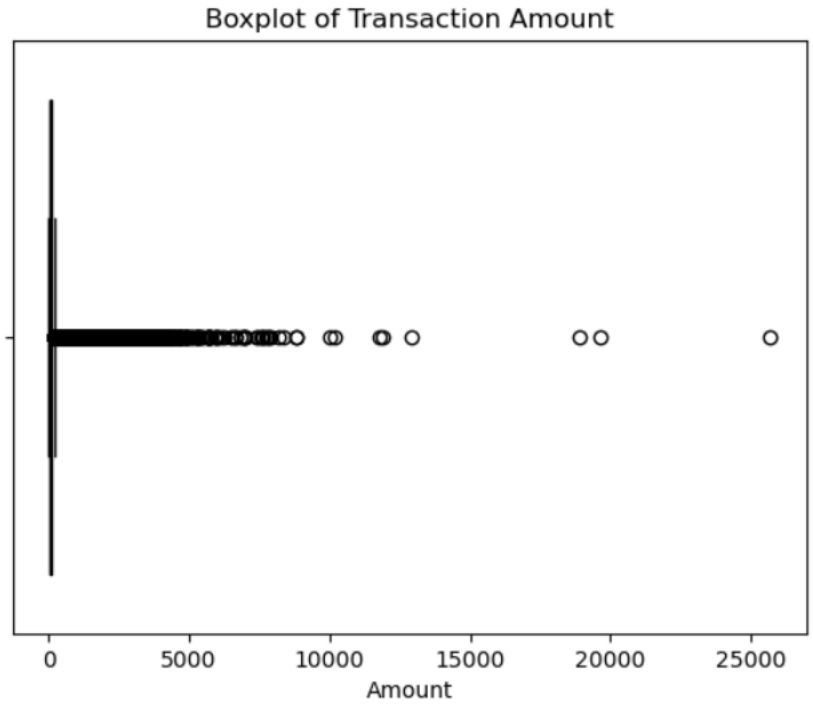
```
(284807, 31)
```

```
df['Class'].value_counts(normalize=True)*100
```

```
Class
0    99.827251
1     0.172749
Name: proportion, dtype: float64
```

```
import seaborn as sns

sns.countplot(x='Class', data=df, color='black')
plt.title('Fraud vs Non-Fraud Transactions')
plt.show()
```


```
sns.boxplot(x=df['Amount'], color='black')
plt.title('Boxplot of Transaction Amount')
plt.show()
```



```python
from scipy.stats import skew

skewness = skew(df['Amount'])
print(f"Skewness: {skewness}")
```

```
Skewness: 16.97763503663315
```

```python
#Train/Test Split
X = df.drop('Class', axis=1)
y = df['Class']
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, test_size=0.2, random_state=42)
```

```python
#Define Base Models
base_models = [
    ('lr', LogisticRegression(max_iter=500, class_weight='balanced', random_state=42)),
    ('rf', RandomForestClassifier(n_estimators=100, class_weight='balanced', random_state=42)),
    ('xgb', XGBClassifier(n_estimators=100,eval_metric='logloss', random_state=42))
]
#Meta Classifier
meta_model = LogisticRegression(max_iter=500)

#Stacking Model
stacking_model = StackingClassifier(estimators=base_models, final_estimator=meta_model, cv=3, passthrough=True)

#Full Pipeline with SMOTE
pipeline = ImbPipeline(steps=[
    ('scaler', StandardScaler()),
     #('smote', SMOTE(random_state=42, k_neighbors=6, sampling_strategy=0.2)), # Experiment with SMOTE parameters
    #('adasyn', ADASYN(random_state=42)),
    ('smotetomek', SMOTETomek(random_state=42)),
    ('model', stacking_model)
])

#Train the Model
pipeline.fit(X_train, y_train)

#Evaluate on Test Data
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
y_pred = pipeline.predict(X_test)
y_proba = pipeline.predict_proba(X_test)[:, 1]
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

#AUPRC
precision, recall, _ = precision_recall_curve(y_test, y_proba)
auprc = auc(recall, precision)
print(f"AUPRC: {auprc:.4f}")
# Plot Precision-Recall Curve
plt.figure(figsize=(8,6))
plt.plot(recall, precision, marker='.')
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.title('Precision-Recall Curve')
plt.grid()
plt.show()
print("\nAccuracy Score:", accuracy_score(y_test, y_pred))
#ROC-AUC SCORE
print("ROC-AUC Score:", roc_auc_score(y_test, y_pred))
plt.show()
```

```
#Evaluate on Test Data
y_pred = pipeline.predict(X_test)
y_proba = pipeline.predict_proba(X_test)[:, 1]
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

#AUPRC
precision, recall, _ = precision_recall_curve(y_test, y_proba)
auprc = auc(recall, precision)
print(f"AUPRC: {auprc:.4f}")
# Plot Precision-Recall Curve
plt.figure(figsize=(8,6))
plt.plot(recall, precision, marker='.')
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.title('Precision-Recall Curve')
plt.grid()
plt.show()
# 10. Cross-validation scores
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
cv_preds = cross_val_predict(pipeline, X, y, cv=cv, method='predict')
print("Cross-validated Classification Report:\n", classification_report(y, cv_preds))
```

```
# Save the trained pipeline (model + scaler + SMOTETomek)
joblib.dump(pipeline, 'fraud_detection_pipeline.pkl')
print("Model pipeline saved successfully.")
```

```python
#Create a Flask API to Load and Use the Model

from flask import Flask, request, jsonify
from flask_cors import CORS
import joblib
import pandas as pd
import numpy as np
from lime.lime_tabular import LimeTabularExplainer
from sklearn.model_selection import train_test_split

# Load trained pipeline
pipeline = joblib.load('fraud_detection_pipeline.pkl')

# Load dataset used during training
df = pd.read_csv('C:/Users/ANAK/Fraud/creditcard.csv')
X = df.drop('Class', axis=1)
y = df['Class']
features = list(X.columns)

# Recreate training set for LIME
X_train, _, _, _ = train_test_split(X, y, test_size=0.2, stratify=y, random_state=42)

# Initialize Flask
app = Flask(__name__)
CORS(app)

# LIME Explainer setup
explainer = LimeTabularExplainer(
    training_data=np.array(X_train),
    feature_names=features,
    class_names=['Not Fraud', 'Fraud'],
    mode='classification'
)

# Human-readable feature name map
feature_names_map = {
    "V0": "transaction signal",
    "Amount": "transaction amount",
    "V14": "signal pattern",
    "V3": "velocity anomaly",
    "V11": "anomaly patterns",
    "Time": "time since last transaction"
}

def translate_condition(condition):
    feature = condition.split()[0]
    readable = feature_names_map.get(feature, feature)
    if ">" in condition and "<=" not in condition:
        return f"{readable} was higher than normal"
    elif "<=" in condition and ">" not in condition:
        return f"{readable} was lower than usual"
    elif "<=" in condition and ">" in condition:
        return f"{readable} was in a borderline range"
    else:
        return f"{readable} had an unusual value"

def generate_plain_explanation(explanation, prediction, top_n=4):
    explanation = sorted(explanation, key=lambda x: -abs(x[1]))[:top_n]
    phrases = []
    for condition, score in explanation:
        reason = translate_condition(condition)
        weight = round(score * 100, 2)
        phrases.append(f"{reason} (+{weight}% contribution)")
    if prediction == 1:
        return "The model flagged this transaction as **fraudulent** based on: " + "; ".join(phrases) + "."
    else:
        return "The model determined this transaction is **legitimate** because: " + "; ".join(phrases) + "."

@app.route("/")
def home():
    return "Fraud Detection API is running!"

@app.route("/predict", methods=["POST"])
def predict():
    try:
        data = request.get_json()
        input_data = [data[feature] for feature in features]
        input_df = pd.DataFrame([input_data], columns=features)

        # Prediction
        pred = pipeline.predict(input_df)[0]
        proba = pipeline.predict_proba(input_df)[0][1]

        # LIME explanation (for predicted class)
        exp = explainer.explain_instance(
            data_row=np.array(input_data),
            predict_fn=pipeline.predict_proba,
            num_features=5,
            labels=[0, 1]
        )
        explanation = exp.as_list(label=pred)
        plain_explanation = generate_plain_explanation(explanation, pred)

        return jsonify({
            "prediction": int(pred),
            "fraud_probability": round(proba, 4),
            "message": "Fraud detected!" if pred == 1 else "Transaction is legitimate",
            "explanation": explanation,
            "plain_explanation": plain_explanation
        })

    except Exception as e:
        return jsonify({"error": str(e)})

if __name__ == '__main__':
    app.run(debug=False)
```

```python
import subprocess

# Run app.py as a background process
process = subprocess.Popen(["python", "app.py"])

# Optional: Print the process ID to confirm it's running
print(f"Flask app is running with PID: {process.pid}")
```

```
Flask app is running with PID: 8776
```

```python
html_code = """
<!DOCTYPE html>
<html>
<head>
    <title>Fraud Detection API Tester</title>
    <style>
        body {
            font-family: Arial, sans-serif;
        }
        .container {
            width: 80%;
            margin: 0 auto;
            text-align: center;
        }
        textarea {
            width: 80%;
            margin: 0 auto;
        }
        button {
            background-color: #007BFF;
            color: white;
            padding: 10px 20px;
            border: none;
            font-size: 16px;
            cursor: pointer;
        }
        button:hover {
            background-color: #0056b3;
        }
        pre {
            background-color: #f4f4f4;
            padding: 10px;
            border: 1px solid #ccc;
            width: 80%;
            margin: 20px auto;
            text-align: left;
            overflow-x: auto;
        }
    </style>
</head>
<body>
    <div class="container">
        <h2>Fraud Detection API Enter Transaction Details</h2>
        <form id="predictForm">
            <textarea id="jsonInput" rows="20"></textarea><br><br>
            <button type="submit">Send Request</button>
        </form>

        <h3>Response:</h3>
        <pre id="response"></pre>
    </div>

    <script>
        document.getElementById("predictForm").addEventListener("submit", function(e) {
            e.preventDefault();
            const jsonData = document.getElementById("jsonInput").value;

            fetch("http://127.0.0.1:5000/predict", {
                method: "POST",
                headers: {
                    "Content-Type": "application/json"
                },
                body: jsonData
            })
            .then(response => response.json())
            .then(data => {
                document.getElementById("response").textContent = JSON.stringify(data, null, 2);
            })
            .catch(error => {
                document.getElementById("response").textContent = "Error: " + error;
            });
        });
    </script>
</body>
</html>
"""

with open("predict_form.html", "w") as f:
    f.write(html_code.strip())

print("HTML form saved as predict_form.html")
```

```python
# Retrieving the transactions for testing the model
import pandas as pd
from sklearn.model_selection import train_test_split

# Load your dataset
df = pd.read_csv("C:/Users/USER/Fraud/creditcard.csv")

# Split features and labels
X = df.drop("Class", axis=1)
y = df["Class"]

# Split into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Get the first 10 rows from the test set
samples = X_test.iloc[:10]

# Convert to a list of dictionaries
sample_list = samples.round(6).to_dict(orient="records")

# Display or use them
import json
print(json.dumps(sample_list, indent=2))
```

```python
import requests
import random
from concurrent.futures import ThreadPoolExecutor

# Prepare a sample transaction
features = ['Time'] + [f'V{i}' for i in range(1, 29)] + ['Amount']

def generate_transaction():
    return {feature: round(random.uniform(-5, 5), 6) for feature in features}

# Function to send a request
def send_request():
    try:
        response = requests.post("http://127.0.0.1:5000/predict", json=generate_transaction())
        return response.status_code, response.elapsed.total_seconds()
    except Exception as e:
        return "Error", str(e)

# Run with many parallel threads
num_requests = 500
with ThreadPoolExecutor(max_workers=20) as executor:
    results = list(executor.map(lambda _: send_request(), range(num_requests)))

# Analyze results
successes = [r for r in results if r[0] == 200]
failures = [r for r in results if r[0] != 200]

print(f"Total Requests: {num_requests}")
print(f"Successful: {len(successes)}")
print(f"Failed: {len(failures)}")
print("Average Response Time (s):", round(sum([r[1] for r in successes]) / len(successes), 4) if successes else "N/A")
```

```
Total Requests: 500
Successful: 500
Failed: 0
Average Response Time (s): 0.4878
```

Link to the Artifact:

https://drive.google.com/drive/folders/18ceBrfuv9Hi-3p6_3rgGv4ABzhN6NPPL?usp=sharing

Appendix C: Screen Shorts of the Web Interface to interact and test the model