ADDIS ABABA UNIVERSITY
ADDIS ABABA INSTITUTE OF TECHNOLOGY (AAiT)
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

ANOMALY-AUGMENTED DEEP LEARNING FOR ADAPTIVE FRAUD DETECTION IN MOBILE MONEY TRANSACTIONS

BY
MELAT KEBEDE

ADVISOR
Dr. BISRAT DEREBSSA

A thesis submitted to the School of Electrical and Computer Engineering in partial fulfillment of the requirements for the Degree of Master of Science in Computer Engineering

JUNE, 2024
ADDIS ABABA, ETHIOPIA

ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

The undersigned have examined the thesis titled:

ANOMALY-AUGMENTED DEEP LEARNING FOR ADAPTIVE FRAUD DETECTION IN MOBILE MONEY TRANSACTIONS

BY

MELAT KEBEDE

Approval by Boards of Examiners

Dr. Bisrat Derebssa	_____	_____
Dean, SECE, AAiT	Date	Signature
Dr. Bisrat Derebssa	_____	_____
Advisor	Date	Signature
Dr. Fitsum Assamnew	_____	_____
Internal Examiner	Date	Signature
Dr. Menore Tekeba	_____	_____
External Examiner	Date	Signature

Declaration

I, Melat Kebede Abraham, declare that this Masters Thesis entitled ”Anomaly-Augmented Deep Learning for Adaptive Fraud Detection in Mobile Money Transactions” is my original work. All sources of information in this study have been appropriately acknowledged through citation. I further confirm that this thesis has not been submitted either in part or in full for any other requirements to any other learning institution.

Student Name: Melat Kebede Abraham

Signature: _____

Date: _____

JUNE, 2024

Acknowledgments

First and foremost, I am deeply grateful to the Almighty God for His abundant blessings and unwavering guidance throughout my journey.

I extend my sincere appreciation to Dr. Bisrat Derebssa, my respected advisor, for his unwavering support, invaluable guidance, and insightful advice throughout my academic pursuit. His mentorship has played a pivotal role in my growth and achievement, inspiring me to strive for excellence.

I am deeply thankful to my father for his unwavering belief in my abilities, relentless encouragement and support. To my family, whose boundless love, support, and encouragement have been my pillar of strength through every challenge and success, I owe a debt of gratitude that words cannot fully express. Their unwavering faith in me has been a guiding light, significantly shaping both my academic and personal journey.

I am grateful to my fellow Computer Engineering MSc students at AAiT, as well as my friends and colleagues, whose dedication and passion have consistently inspired me. Working with them has greatly enriched my learning experience.

Abstract

Mobile Money, a revolutionary technology, enables individuals to manage their bank accounts entirely via their mobile devices, allowing for transactions like bill payments with unmatched ease and efficiency.This innovation has significantly reshaped financial landscapes, particularly in developing countries with limited access to traditional banking, by promoting financial inclusion and driving economic opportunity. However, the rapid growth of mobile money services has introduced significant challenges, such as fraud, where unauthorized individuals manipulate the system through various scams, creating serious risks that lead to financial losses and undermining trust in the system. We propose a fraud detection model that integrates deep learning techniques to identify fraudulent transactions and adapt to the dynamic behaviors of fraudsters in mobile money transactions. Given the private nature of financial data, we utilized a synthetic dataset generated using the Pay Sim simulator, which is based on a company in Africa. We evaluated three deep learning architectures, namely Restricted Boltzmann Machine (RBM), Probabilistic Neural Network (PNN), and Multi-Layer Perceptron (MLP) for fraud detection, emphasizing feature engineering and class distribution. The MLP achieved 95.70% accuracy, outperforming the RBM (89.91%) and PNN (73.36%) across various class ratios and on both the original and feature-engineered datasets. Among various techniques for anomaly detection, the Auto-Encoder consistently outperformed others, such as the Isolation Forest and Local Outlier Factor, achieving an accuracy of 82.85%. Our hybrid model employed a feature augmentation approach, integrating prediction scores from an Autoencoder model as additional features. These scores were then fed into the Multi-layer Perceptron (MLP) model along with the original dataset. This hybrid approach achieved 96.56% accuracy, 97.62% precision, 84.16% recall, and a 90.39% F1-score, outperforming the standalone MLP.The Hybrid model achieved an accuracy of 73.33% on unseen dataset, showing a 3.9% increase over the MLP model's 69.41% accuracy, and demonstrating its enhanced ability to capture and adapt to evolving fraud patterns.This study finds that the hybrid model's enhanced performance highlights the significance of anomaly detection and feature engineering in improving fraud detection.

Keywords: MLP, Autoencoder, Mobile Money Transfer (MMT), Mobile Money Fraud, Deep Learning Models

Table of Contents

List of Figures

List of Tables

List of Acronyms

PNN Probabilistic Neural Network

RBM Restricted Boltzmann Machine

MLP Multi-Layer Perceptron

MMT Mobile Money Transfer

PIN Personal Identification Number

MWT Manual Weights Tuning

SMOTE Synthetic Minority Oversampling Technique

ADASYN Adaptive Synthetic Sampling

SVM Support Vector Machine

NB Naïve Bayes

XGBoost Extreme Gradient Boosting

EDT Ensemble of Decision Trees

SAE Stacked Autoencoders

ROC Receiver Operating Characteristic

RNN Recurrent Neural Network

MSE Mean Squared Error

EM Expectation-Maximization

AUC Area under the Receiver Operating Characteristic (ROC) Curve

XGBOD Extreme Gradient Boosting Outlier Detection

ReLU Rectified Linear Unit

LOF Local Outlier Factor

Chapter 1

Introduction

Mobile Money is like having your own banker present with you everywhere you go. Think of it as managing your entire bank accounts entirely through the phone, without necessarily having to set foot inside a Bank. This is what you can do with it. This way, you are able to pay your bills among other things within a split second just by clicking a few icons right from your mobile device, as though there were such a thing as electronic cash in our modern-day society [1]. This technology is more than just a nice to have; it is changing how individuals obtain and handle their money, therefore ensuring that anybody can easily engage with banks [1]. This transformative capability reflects the significant impact of mobile money on modern financial landscapes.

Mobile Money is a crucial advancement in the field of digital payments that relies on wireless devices like mobile phones and smartphones. Hence, providing unmatched convenience it comes with reduced transaction fees significantly and increases the safety of online transactions. In addition, organizations gain insights into the buying habits and interactions with customers which ultimately make them more efficient [2]. Mobile money is more than just a convenience. It is a powerful tool for boosting financial inclusion and promoting individual and entrepreneurial empowerment by fitting in so easily with everyday schedules [2].

In developing countries with restricted traditional banking access, mobile money transfer (MMT) is becoming more and more important as a crucial financial facility. For developed countries, MMT is seen as just another extension of their banking services whereas in regions where banking remains hard to reach, its importance goes far beyond simply being convenient [3]. Mobile money transfer technologies are considered vital platforms of significant strategic and societal worth in these areas. They enhance the fight against financial exclusion among populations that are not served by banks and those excluded from formal banking via the use of cell phones for transactions, thus making mobile money an increasingly important form of financial services that can change this sector entirely [3].

Mobile devices growing rapidly are leading to the widespread adoption of mobile payment systems making it outpace other telecommunication infrastructure. The largest growth rate in mobile phone usage particularly in Africa is coupled with growing mobile coverage [2]. This development has allowed millions of people, especially those in remote places, to use mobile money as opposed to banking the traditional way which is more common in sub-Saharan Africa.

The rise in cell phone-based money transfers in economic developing countries such as India, Uganda, Argentina, Tanzania, Zambia, Nigeria, Ghana, and Kenya signify a growing appeal for such transactions in areas with scarce banking services [4]. This service bridges the gap for people without access to formal financial institutions as approximately 68 percent of these unbanked individuals out of an estimated 2.5 billion persons have mobile phones at their disposal [3]. Nonetheless the fast growth presents several problems, one of them being fraud cases involving mobile money as people are attracted to mobile money services by their convenience and easy access [3].

Mobile money services, with its remarkable achievement, are threatened by fraudsters. Operators experience a significant risk when fraudsters continue to be present inside the system. If nothing is done, those fraudsters can suddenly abrupt the business operations [4]. Mobile money transactions are attractive for several reasons: they are good at handling many small payments; they make it easier to transfer funds in digital currencies across borders; and there is absence of strong regulation which means that individuals who commit crimes like fraud think that such transactions are vulnerable targets for them thereby worsening the issue of fraudulent activity [4]. This increasing exposure to fraud necessitates a closer examination of what mobile money fraud involves.

Mobile money fraud occurs when some unauthorized individuals gain illegal access to phone subscribers', operators', and agents' money through wiring funds from one account to another or using a mobile phone to acquire data that may lead to unauthorized withdrawals [4].

Various types of fraudulent activities can occur in mobile money transactions, and the following are some of the most common types of fraud in mobile money transactions:

- **Phishing scams:** are fake text messages or emails that pretend to be from genuine mobile money service providers [5].
- **Fake Mobile Money agents:** are cases of fraud where individuals not representing the company call customers, ask for their Personal Identification Number (PIN)s or other such details of their accounts to steal the money [5].

- **Social Engineering:** for instance, fraudsters pretending to be providers and telling the customers that they have won a prize in a draw, but they must send money to the scammers' number in order to get their prizes [5].
- **SIM! (SIM!) card Fraud:** This occurs when fraudsters take control of someone's mobile money account through his or her **SIM!** card. These include illegal financial transactions performed using the operator's own database access without any permission from the owner.
- **Identity Theft:** occurs when a person's mobile wallet account is moved to someone else's **SIM!** card through dishonest offline SIM swaps. Consequently, fraudsters will use this opportunity to access the Consumer's monetary units and resulting in the access into personal financial details stored inside such a user's device [5].

Fraudulent mobile money operations may lead to significant amounts of money being lost by customers, agents, and intermediaries. For service operators these scams cause immediate financial loss while at the same time diminishing their status among other players in the sector. Mobile money fraud is a threat that affects the entire mobile money system because it leads to instability, unreliability, and loss of trust by users. It also has far-reaching implications on the entire financial system as well as lowering consumers' confidence with regards to the use of these technologies for transactions.

There are several machine learning algorithms that have been used for fraud detection in various financial sectors, such as supervised machine learning's, unsupervised machine Learning's, Ensemble machine learning, and Semi-supervised machine learning's. These algorithms have been shown to be effective in detecting fraudulent transactions in credit card transactions and insurance claims, among others. Recently, Machine learning techniques have shown great promise in detecting fraudulent transactions by analyzing large amounts of data and identifying patterns that may be indicative of fraud. Examples of these techniques include Restricted Boltzmann Machines (RBM) [6], Logistics Regression [7], Gradient Boosted Decision Tree [7], Multilayered Perceptron [8], Extreme Gradient Boost [9] and Stacked Recurrent Neural Network [10].

However, Fraudsters are continuously adapting their tactics and developing new advanced strategies. This poses a major limitation for current fraud detection methodologies, as they struggle to keep pace with the ever-changing behaviors of fraudulent activity. These simpler machine learning models trained on historical data may quickly become outdated and ineffective at identifying novel fraud patterns. There is a pressing need for more dynamic, adaptive approaches capable of learning and adjusting to detect emerging fraud techniques as fraudsters innovate.

The proposed study will investigate the use of machine learning techniques, mainly deep learning, for fraud detection in mobile money transactions. The study will use a large dataset of mobile money transactions to train and test various deep learning models, with the aim of identifying the most effective algorithms for fraud detection in Mobile money transactions.

1.1 Problem Statement

In an era where financial services are increasingly digitized, Mobile Money stands out as a revolutionary technology that allows individuals to manage their finances seamlessly through mobile devices, transforming everyday transactions. This innovation has become essential in developing countries, offering financial inclusion to millions who lack access to traditional banking. However, with this remarkable convenience comes a significant challenge: the rapid rise of fraud in mobile money transactions.

Given the potential risks associated with mobile banking services, it is crucial to develop effective fraud detection systems to mitigate financial fraud in mobile money transactions. Traditional methods often rely on rule-based approaches and linear machine learning algorithms, which fall short in capturing the complex, non-linear relationships inherent in fraudulent activities. This limitation underscores the need for more advanced techniques that can better detect and understand the complexities of fraud.

To further complicate matters, fraudsters often try to move illicit funds in smaller, less noticeable increments. These patterns are often non-linear in nature, meaning they do not follow a simple, straight-line relationship that linear models may struggle to capture effectively.

Moreover, the ever-evolving nature of fraudsters' tactics and the development of sophisticated strategies pose a significant challenge to the effectiveness of current fraud detection methodologies. As fraudsters continually adapt and innovate, machine learning models trained on historical data may quickly become outdated and inadequate at identifying new fraud patterns.

To address this issue, deep learning models, such as neural networks, are better suited due to their capacity to learn complex, non-linear patterns. However, the performance of these models is greatly dependent on thorough feature engineering and tuning, a process that is often insufficient in current approaches, thereby limiting their full potential in detecting fraud.

By developing an effective deep learning-based fraud detection system for mobile money transactions, This study's findings will contribute to the development of effective fraud detection systems for mobile money transactions, which can help protect customers from financial losses and improve the overall trust and adoption of mobile money services. The system will provide financial institutions with a hybrid machine learning model for detecting and preventing fraudulent activities, which can ultimately contribute to the growth and stability of the financial sector.

In this study, an attempt will be made to answer the following research questions:

RQ1 What impact does feature engineering have on the prediction performance of fraudulent mobile money transactions?

RQ2 Does the anomaly-augmented deep learning model have better performance than the existing state of the art models?

RQ3 Does the proposed model have better adaptability to evolving fraud patterns than the existing state of the art models?

1.2 Objectives

1.2.1 General Objective

The general objective of this study is to develop a fraud detection model that integrates anomaly-augmented deep learning techniques to detect fraudulent transactions and adapt to the dynamic behaviors of fraudsters in mobile money transactions.

1.2.2 Specific Objectives

The specific objectives that will be addressed to achieve the general objective:

- Identify and integrate relevant features to improve fraud detection model's accuracy.
- Compare various deep learning models to find the most effective one for detecting fraud in mobile money transactions.
- Investigate and propose methodologies to enhance the adaptability of fraud detection models to evolving fraud patterns over time.
- Investigate the impact of different parameters, such as the size of the training dataset or the choice of hyper-parameters, on the performance of the fraud detection system.

1.3 Contribution

The outcome of the research makes an important addition to research done on fraud detection methods. Highlighting important advancements made in feature engineering is significant because it has led to remarkable increases in accuracy rates concerning determining fraudulent mobile money transactions.

In addition, we have come up with new methodologies that are tailor-made to enhance the flexibility of the fraud detection model to withstand the continuously changing methods employed by the fraudsters. Our model has the ability to detect new fraudulent schemes because of its ability to constantly change depending on the changes in behavior patterns associated with fraudulent activities.

Our study uses improved feature engineering and adaptable methods to make progress in fraud detection that can be replicated while laying a foundation for more research in this area. Our contributions create a pathway for better and more effective fraud detection systems which in turn helps to protect the financial systems against fraud and keep stakeholders from suffering due to fraudulent activities.

1.4 Scope and Limitation

The scope of this research paper within the given time is to develop a fraud detection model that integrates deep learning techniques and feature engineering to detect fraudulent transactions and adapt to the dynamic behaviors of fraudsters in mobile money transactions.

The Proposed research has certain limitations that should be considered:

- **Limited Scope:** The Study focuses only on mobile money transactions within the "Pay Sim" simulator dataset, which makes it hard to apply the results beyond limited scenarios in the real world.
- **Dataset Constraints:** The study will use a confined dataset of mobile money transactions, and the results may be affected by data biases and imbalances, affecting the validity and generalizability of the result.
- **Data Quality Issues:** Challenges related to the availability and quality of data may arise and the quality of the data may be affected by issues such as missing values, outliers, and data errors.

1.5 Thesis Organization

The research paper is organized in five chapters. The first chapter provides a comprehensive overview of mobile money fraud, including a detailed examination of the statement of the problem, objectives of the research, and the scope and limitation of the investigation. Chapter two, a detailed review of existing machine learning-based approaches to financial fraud detection. In Chapter three, we examine the methods and techniques utilized in our research. The fourth chapter presents our experiments, results, and a comparative analysis of various state-of-the-art models. Finally, Chapter five presents our conclusions based on the findings, along with recommendations of future works for researchers in this field.

Chapter 2

Literature Review

Literature review has been conducted to assess concepts, techniques, and application of machine learning and to get domain knowledge about the problem. In order to select modeling techniques that best suit the problem stated above, different machine learning journals and published articles on the application of machine learning techniques in financial fraud detection were reviewed.

Financial fraud detection has been an important and challenging task for financial institutions. With the increasing number of financial transactions, the manual detection of frauds has become impractical. Machine learning techniques have emerged as a promising approach for detecting frauds in financial transactions. In recent years, several studies have investigated the performance of various machine learning algorithms for fraud detection.

2.1 Machine Learning Approaches

2.1.1 Supervised Machine Learning Models

In their exploration of the effects of various sampling methods on predicting Fraud in mobile money transactions using supervised machine learning approach, Botchey et al. [7]utilized the Pay Sim dataset where one of the major challenges was class imbalance. The investigators compared the performance of logistic regression models, created by different sampling methods, that are under sampling, logistic regression weighted, Synthetic Minority Oversampling Technique (SMOTE), SMOTE RS, Manual Weights Tuning (MWT) (Manual Weights Tuning), and Adaptive Synthetic Sampling (ADASYN). There was also feature reduction done through which some features were discarded if they were not considered suitable independent variables. According to the study, manual adjustment of weights leads to the highest accuracy rate of 92.74%. The research work has further emphasized the value of class imbalance treatment in fraud prevention models. An investigation into these authors' study showed how useful some sampling methods are to model performance [7]

Wirgen and Rube [11]investigated the performance of Logistic Regression, Random Forest, and Support Vector Machine in identifying fraud in mobile money transactions in their research using the Pay Sim dataset. The main goal of the study was to differentiate between these methods of classification under different imbalanced data distributions to evaluate their efficiency in detecting fraudulent activities. Under sampling techniques have been utilized by the researchers in handling class imbalance. In addition, they utilize feature engineering to improve the dataset by deriving new features such as 'Hour' and 'Day' from the raw 'step' feature. The purpose of this methodology is to achieve a better comprehension of transaction behavior for better model prediction outcomes [11].

The researchers compared three models: the Logistic Regression model, a Random Forest model, and Support Vector Machine model where each of them obtained a 99% average accuracy on different subsets of the dataset. Despite their high accuracy levels, it was found that while Support Vector Machine (Support Vector Machine (SVM)) and Logistic Regression (LR) performed differently on metrics, Random Forest (RF) had more robust and consistent results over varied distributions of imbalanced datasets based on what was observed based on the researchers work [11]. However, a limitation was noted in their work due to insufficient feature engineering, which may have impacted the overall performance of the models.

Botchey, Qin and Hughes-Lartey [7] examined different machine learning algorithms applied on the Pay Sim dataset so as to ascertain the possibility of fraud in mobile money transfers. The study is mainly into measuring the efficiencies of Gradient Boosted Decision Trees (**GBDT!** (**GBDT!**)) (tree-based), Support Vector Machines (SVM) (kernel-based) and Naïve Bayes (Naïve Bayes (NB)) (probabilistic-based) algorithms. To tackle the problem of class imbalance, different sampling techniques such as, Near Miss, Random Under sampling, Random Oversampling, SMOTE were used by researchers. feature engineering was implemented by carrying out feature elimination where features possess p-values surpassing 0.5 during their construction to ensure the inclusion of statistically significant variables [7].

The study utilized various evaluation metrics, including accuracy, precision, recall and F1-score to evaluate how well these models were performing during the experiments. Surprisingly enough, it turns out that even when there is no specific class imbalance handling in the data set, Gradient Boosted Decision Trees still manage to have an accuracy going up to 99.90%. In contrast, utilizing Under sampling, the Support Vector Machine and Naïve Bayes models both achieve accuracy's of 86.34% and 88.97%. This study concludes that the predictions of fraudulent transactions in mobile money systems can be highly effective when ensemble methods are used, mainly **GBDT!**. The Bernoulli Naïve Bayes algorithm also performs relatively strongly, indicating its potential utility in fraud detection scenarios [7].

Nisha Balani, Meher Bhawnani, and Ankita Kamle [9] demonstrated the effectiveness of numerous supervised machine learning algorithms for predicting fraudulent mobile money transactions. They utilized four algorithms in the Pay Sim dataset which are Random Forest, KNeighbors, Logistic Regression and Extreme Gradient Boosting (Extreme Gradient Boosting (XGBoost)). The only thing they examined was how well each model performed in terms of precision, recall and F2-score without mentioning anything about class imbalance or feature engineering techniques within their study [9].

The results of classification of the data indicated that Extreme Gradient Boosting achieved the highest precision at 100%. Subsequently, random forests achieved 99%; while KNeighbors and Logistic Regression attained 90% each. In terms of fraudulent mobile money transactions identification, the authors' that no other model outperforms Extreme Gradient Boosting (XGBoost). Nonetheless, the absence of addressing imbalances between different classes and insufficient feature engineering resulted in generalization restrictions and potential overfitting during their investigation [9].

A study conducted by Mubalaike and Adali [6] has examined the advantageousness of deep-learning models in detecting financial fraud within a mobile money transaction dataset. By using the Pay Sim dataset, authors employed three different types of deep-learning methods: an Ensemble of Decision Trees (Ensemble of Decision Trees (EDT)), Stacked Autoencoders (Stacked Autoencoders (SAE)), and Restricted Boltzmann Machines (RBM). They depended on detailed creation of features, such as creating error amount variables for both sender and recipient accounts, and then leaving out non-informative attributes like 'isFlaggedFraud', 'NameOrig', or 'NameDest'. Thus, the evaluation metrics used for these models were based on accuracy rate alongside ROC curve, as well as confusion matrix. The results demonstrated that Restricted Boltzmann Machines (RBM) performed better than other models as far as precision goes, having attained a rate of 92.86% while Ensemble of Decision Trees (EDT) has 91.76% compared to 81.83% achieved by Stacked Autoencoders (SAE). This research has demonstrated the potential of using advanced deep learning techniques in detecting frauds in financial sectors [6].

Pech [8] notes that fraud detection in mobile money transfers could be addressed as a binary classification problem, leveraging the Pay Sim dataset. The author evaluated the three supervised learning methods such as, Multi-layer Perceptron (MLP), Naive Bayes (NB) and Support Vector Machine (SVM). Notably, the dataset exhibits class imbalance, which the researchers addressed through the use of random sampling. Feature engineering was kept minimal, with the removal of the variables 'isFlaggedFraud', 'nameOrig' and 'nameDest' was made on the original dataset [8].

Evaluation metrics such as accuracy, precision, recall and F1-score were used to assess the model's performance. According to the results, Multi-layer Perceptron (MLP) had the highest accuracy of 90.76% while Support Vector Machine (SVM) and Naive Bayes (NB) had an accuracy of 87.61% and 78.61% respectively. This research demonstrated that Multi-layer Perceptron (MLP) has potential use in fraud detection using mobile money transfer as it excelled more than NB or SVM. Having said that, the researchers concluded that SVM and MLP requires hyperparameter tuning to produce the best results and on the other hand, NB does not require hyperparameter tuning making it stand out for its efficiency [8].

Bandyopadhyay and Dutta [10] proposed a model of neural networking specifically intended at monitoring fraudulent transactions via mobile money transfers particularly in times affected by coronavirus pandemic in 2019. The researchers utilized the PaySim dataset for the detection of financial fraud focusing on a Stacked Recurrent Neural Network (Recurrent Neural Network (RNN)) model. The evaluation of the model's performance is then covered in terms of certain metrics which include accuracy, F1 score, and Mean Squared Error (Mean Squared Error (MSE)). Unlike many other studies, this research does not treat the issue of class imbalance or feature engineering as do other studies [10].

The Stacked Recurrent Neural Network has achieved a remarkable accuracy rate of 99.87% in the experimental results which suggests that it is efficient at detecting fraud cases in the financial sector. In overall terms, the study shows that deep learning models, especially RNNs, can be used to develop reliable and scalable solutions for fighting fraud in finance sector, this is possible because the system accuracy rate is high as well as low error rate which makes it useful in alerting clients about any possible fraudulent activity in their accounts [10].

Sa'adah and Pratiwi [12] studied the usage of a Probabilistic Neural Network (PNN) in the classification of customer behaviors in Pay Sim mobile money transactions. The main aim of the study was to employ a PNN system combined with a binary classification model aimed towards preventing frauds in mobile money transactions. The evaluation metrics that have been used in this study were accuracy. While specific feature engineering is not provided and no class imbalance handling has been employed, the result demonstrated that the model Probabilistic Neural Network (PNN) achieved an accuracy of 88%. According to the researchers, fraud detection mechanisms could be made more accurate and reliable if other machine learning techniques are combined with Probabilistic Neural Network (PNN) [12].

2.1.2 Semi-Supervised Machine Learning Models

Choi and Lee [13] conducted research focusing on the performance evaluation of the different data mining methods used in trying to detect mobile payment system financial fraud. The researcher's conducted this study using the semi-supervised machine learning technique, leveraging both unsupervised and supervised machine learning models using the sampled mobile payment dataset from Korea. Expectation-Maximization (EM), K-means, FarthestFirst, XMeans, and MakeDensity among others are examples of unsupervised methods. These methods are applied to group the data points into clusters. Alongside supervised methods such as Naive Bayes, SVM, Logistic Regression, and Random Forest. Feature engineering and selection were essential components of the methodology, utilizing filter-based feature selection algorithms to identify relevant features [13].

Performance of the models can be assessed using evaluation metrics such as F1-score, ROC, and Area under the ROC Curve (AUC) values. Their experiments showed that the application of EM, K Means, and MakeDensity algorithms into supervised learning models resulted in a 100% perfect accuracy rate for various data ratios. In the same way, in all supervised models but Naïve Bayes, 100% accuracy was achieved using FarthestFirst and XMeans clustering methods. This study has shown that it is possible to improve detection capabilities using both unsupervised clustering and supervised learning methods when applied to finance-related frauds [13].

Hajek, Abedin and Sivarajah [14] in their study proposed an advanced fraud detection framework for mobile payment systems using the Extreme Gradient Boost (XGBoost) algorithm for fraud detection through a semi-supervised approach that further boosts the performance of Extreme Gradient Boost Employing a semi-supervised approach, the study enhances the performance of the XGBoost method on highly imbalanced datasets by introducing outlier scores obtained from multiple unsupervised outlier detection methods, notably Extreme Gradient Boosting Outlier Detection (XGBOD). The model's effectiveness was comprehensively assessed using such evaluation criteria as accuracy, precision, F1-Score, recall and the area under the ROC curve (AUC). Both models, XGBOD, and XGBoost, had the accuracy of 99%, validating how effective the ensemble and semi-supervised are when it comes to detecting fraud transactions. Ensemble methods, particularly those relying on XGBoost are significantly more effective than either individual machine learning algorithms; neither do they compare well with any traditional anomalousness detection methods which depend solely on statistics, as stated on this research paper [14].

2.2 Summary

Related Papers	Dataset Used	Methodology	Limitation
Choi, D., & Lee, K. (2017)	Mobile Payment Dataset, Korea	Semi-Supervised Machine Learning	Not adaptive to evolving fraud patterns.
Mubalaike, A. M., & Adali, E. (2018)	Pay Sim Dataset	Deep Machine Learning	Doesn't address class imbalance. Insufficient feature engineering results in less informative features. Not adaptive to evolving fraud patterns.
Pech, R. (2019)	Pay Sim Dataset	Supervised Machine Learning	Insufficient feature engineering results in less informative features. Not adaptive to evolving fraud patterns.
Botchey, F. E., Qin, Z., & Hughes-Lartey, K. (2020)	Pay Sim Dataset	Supervised Machine Learning	Not adaptive to evolving fraud patterns.
NishaBalani, M., MeherBhawnani, M., & AnkitaKamle, M. (2020)	Pay Sim Dataset	Supervised Machine Learning	Doesn't address class imbalance. Insufficient feature engineering results in less informative features. Not adaptive to evolving fraud patterns.
Bandyopadhyay, S. K., & Dutta, S. (2020)	Pay Sim Dataset	Deep Machine Learning	Doesn't address class imbalance. Insufficient feature engineering results in less informative features. Not adaptive to evolving fraud patterns.

Related Papers	Dataset Used	Methodology	Limitation
Sa'adah, S., & Pratiwi, M. S. (2020)	Pay Sim Dataset	Deep Machine Learning	Doesn't address class imbalance. Insufficient feature engineering results in less informative features. Not adaptive to evolving fraud patterns.
Botchey, F. E., Qin, Z., Hughes-Lartey, K., & Ampomah, E. K. (2022)	Pay Sim Dataset	Supervised Machine Learning	Insufficient feature engineering results in less informative features. Not adaptive to evolving fraud patterns.
Wirgen, I., & Rube, D. (2021)	Pay Sim Dataset	Supervised Machine Learning	Insufficient feature engineering results in less informative features. Not adaptive to evolving fraud patterns.
Hajek, P., Abedin, M. Z., & Sivarajah, U. (2023)	Pay Sim Dataset	Supervised Machine Learning	Doesn't address class imbalance. Insufficient feature engineering results in less informative features. Not adaptive to evolving fraud patterns.

Chapter 3

Methodology

This chapter contains the research methodology which was used in preparing the dataset and the techniques used in achieving the research objectives. The proposed architecture aims to predict mobile money fraud using machine learning algorithms as illustrated in 3.1 while the research methodology starts with data collection, followed by data preprocessing to create a well-structured dataset for developing prediction models. Feature engineering is thereafter applied to transform raw data so that more meaningful representations are produced; it allows models to capture relevant patterns accurately. The selection of each feature is made on its predictive ability and level of redundancy. After that the data set was split into training and test data for machine learning models that would be able to predict fraud in mobile money transactions.

Various machine learning algorithms such as deep learning and anomaly detection were used to train and test the prediction models based on the selected features. The best-performing model was determined by doing a comparative analysis on performance evaluation metrics for each model. This chapter gives a comprehensive overview concerning the steps and methodology used in developing a Adaptive mobile money fraud prediction model.

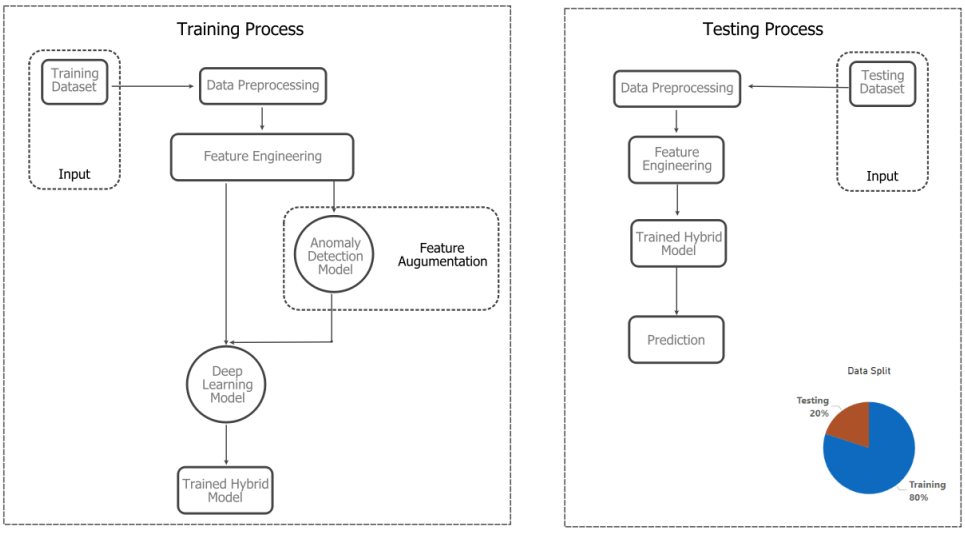

Figure 3.1: The Proposed Hybrid Model

3.1 Data Collection and Preparation

3.1.1 Data Collection

Due to the private nature of financial data, publicly available datasets for analysis are scarce. As a result, this work utilizes a synthetic dataset generated using the Pay Sim simulator. This simulator employs certain aggregated metrics from the private dataset of a multinational mobile financial services provider that offers mobile banking operating across 14 countries on a global scope[15].

The dataset is based on an actual application of mobile money transactions that were sampled over a month's period of financial logs from a mobile money service provider operating in an African country[15]. The simulated data aggregated both normal operation and injected malicious behavior to facilitate fraudulent behaviors, which are used in evaluating various methods employed to detect fraud transactions through mobile financial services. This synthetic dataset was made specifically for Kaggle[1] after being scaled down by a factor of four from its initial size[15].

There are 6,362,620 transactions, the dataset has 11 attributes which include is:

- Type of transactions
- Amount transacted.
- Customer ID and Recipient ID
- Old and New balance of Customer and Recipient
- Time step of the Transaction
- Whether the transaction was fraudulent or not

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M1979787155	0.0	0.0	0	0
1	1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M2044282225	0.0	0.0	0	0
2	1	TRANSFER	181.00	C1305486145	181.0	0.00	C553264065	0.0	0.0	1	0
3	1	CASH_OUT	181.00	C840083671	181.0	0.00	C38997010	21182.0	0.0	1	0
4	1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	M1230701703	0.0	0.0	0	0

Figure 3.2: PaySim: Sample Dataset

[1] https://www.kaggle.com/datasets/ealaxi/paysim1

Attribute	Description
Step	Maps a unit of time in the real world. (1 step = 1 hour of time)
Type	Transaction Type
Amount	Amount of the transaction in local currency
NameOrig	Customer who started the transaction
oldbalanceOrg	Initial balance before the transaction
newbalanceOrig	New balance after the transaction
nameDest	Customer who is the recipient of the transaction
oldbalanceDest	Initial balance recipient before the transaction
newbalanceDest	New balance recipient after the transaction
isFraud	The transactions made by the fraudulent agents inside the simulation
ifFlaggedFraud	Business model aims to control massive transfers from one account to another and flags illegal attempts. (> 200,000 in single transaction)

Table 3.1: Dataset Attributes and Descriptions

3.1.2 Data Pre-processing

One of the most important steps in the field of data analysis and prediction research is pre-processing, because it ensures the accuracy and reliability of the outcomes. This phase involves transforming raw data into a format that can be effectively and easily analyzed [16, 17].

The Key components of data pre-processing include data cleaning, converting from one structure or format into another, combining two or more data samples, reducing its size but retaining key information, and setting all values within a given scale so they range between some lower value and higher value within a particular scale [18].

3.1.2.1 Data Normalization

Data normalization is a pre-processing technique used in machine learning to convert numerical input data into a common scale. This ensures that all features contribute equally to the model's learning process, preventing features with larger scales from dominating the model's training. This process is also known as feature scaling or data standardization [19, 20, 19]. Some algorithms would have a bias towards variables with higher magnitudes. Normalization techniques ensure that no variable dominates the others through magnitude related issues. This in turn increases the efficiency of learning [16, 19].

The formula for standardization is:

$$X_{\text{std}} = \frac{X - \mu}{\sigma} \tag{3.1}$$

where:

- X is the original feature value,
- μ is the mean of the feature,
- σ is the standard deviation of the feature,
- X_{std} is the standardized feature value.

There are different approaches to normalizing data. Some of them are Min-Max scaling, Z-score normalization, robust scaling, standardization among others [19]. For instance, standardization converts the data in such a way that it has a mean value of zero and a standard deviation equal to one. Specifically, the mean value of each feature is subtracted from it before dividing it by the standard deviation of the feature [20, 19]. By doing so, this technique maintains the internal consistencies among numbers without bias. When we apply these normalization techniques, we are able to make sure that the model receives balanced data for learning thus increasing its general performance as well as accuracy.

3.1.3 Handling Class Imbalance

Since the dataset exhibits a high degree of imbalance, wherein 8,213 instances are fraudulent compared to 6,354,407 valid instances, fraudulent cases only represent 0.001% of the dataset.

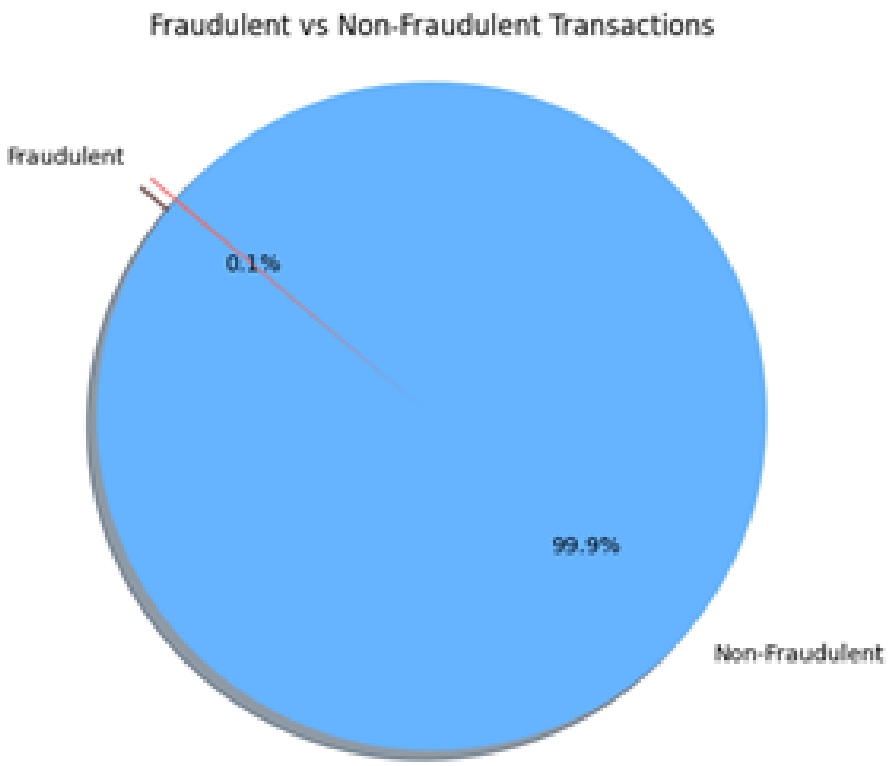

Figure 3.3: Class Imbalance

In order to address this issue and allow the model to learn effectively from all the classes we will be using under sampling techniques [16]. Under sampling techniques involve creating datasets with entries from the minority class (fraudulent cases) by reducing the number of entries from the majority class (valid cases). By doing so, we are able to create a balanced dataset that contains reasonable number of instances from both classes [16, 17]

We have experimented with several class ratios namely : 50:50, 60:40, 70:30, and 80:20 To explore the effects of different class distributions and to determine the optimal ratio for under-sampling [21, 22]. The 50:50 ratio ensures an equal number of instances for both classes thus helping create balanced representation while training. A 60:40 ratio, introduces slight imbalance closer to real world situations, it introduces minor imbalance. In most scenarios, the use of a 70:30 ratio is aligned with many real-world datasets, aiding the model in handling natural imbalances for realistic predictions. An 80:20 ratio tends to capture common real life distributions where the dominated class is emphasized [22].

Thus, using this method we can find a balance by which class imbalance is addressed while data integrity is maintained, thereby yielding stronger prediction models that are both robust and accurate [21, 22].

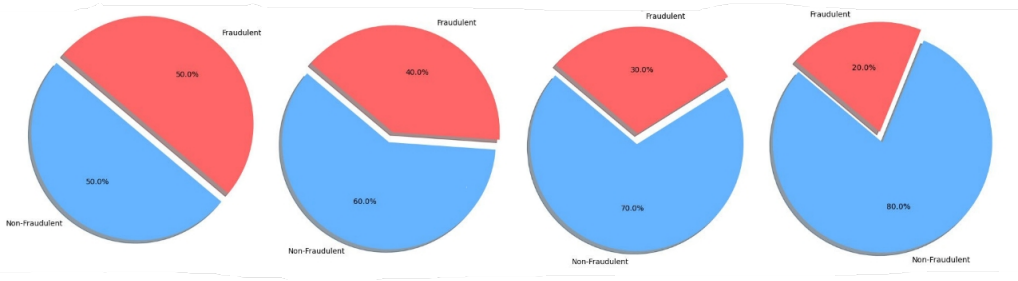

Figure 3.4: Under-sampling at various proportions of the dataset

3.2 Feature Engineering

In machine learning, feature engineering is an important part of data pre-processing. It involves selecting and designing the features that help improve the performance of the model so that it aims at making more accurate predictions through transforming raw data to meaningful one that captures significant patterns as well as relationships [23, 24, 25]. Feature engineering techniques include developing new features from existing ones, selecting the best-informative features, and encoding categorical variables [24, 26]. By transforming features that are important, it is possible to make correct predictions based on the dataset as well as improve generalization ability, ultimately improving its overall performance [16, 18].

3.2.1 Feature Creation

Feature creation involves developing a new attribute(s) that will greatly improve the capacity of the model to make predictions [27]. We have used data-driven feature engineering through domain knowledge and exploratory analysis. This involves identifying those variables in the data which are considered relevant for a predictive model then creating new attributes by combining existing ones using simple mathematical operations; namely sum, difference, product, and ratio. Feature creation aims at strengthening the model's predictive power through integration of new derived features thus enhancing precise predictions [18, 27].

3.2.2 Feature Splitting

Feature splitting is a concept that involves splitting one feature into several features, typically applied to continuous features to create binary or categorical features that represent various aspects or ranges of the original values [25]. This process can simplify complex datasets, enhance interpretability, and improve the performance of machine learning models by providing them with more detailed information. I split the transaction amount into separate features for whole dollars and cents. This approach allows for more efficient analysis and modeling by breaking attributes into smaller, more manageable subsets, ultimately leading to improved understanding and predictions [27].

3.2.3 Categorical Encoding

Categorical encoding is a way of transforming categorical or textual data into numerical format, so that it can be used as an input for machine learning algorithms. This process is crucial because many machine learning based algorithms can only process numerical representation of a dataset [26, 28]. One common way to encode categorical variables is through the use of one-hot encoding, whereby binary vectors are used to represent them. Each class gets a unique binary attribute; 1 means that it exists while 0 denotes no existence [26, 25].

In our case, we have used one-hot encoding to encode features such as transaction type and user types, ensuring that categorical data is appropriately represented in a format suitable for algorithmic processing. This enables our models to effectively interpret and analyze categorical variables, eventually improving the accuracy and performance of our predictive model [28].

3.2.4 Feature Selection

In the modeling process, the last stage is feature selection where irrelevant or redundant features are identified and removed from the dataset so as to leave behind only those that are highly informative as well as relevant to the predictive model [29]. The main purpose of this task is to get rid of any non-essential or repetitive items in the given data which do not add any value but rather diminish its prediction ability of the model.

The process aids in the reduction of over-fitting, while enhancing model interpretability and improving computational efficiency [29, 30]. Various techniques such as statistical tests, rankings of feature importance, and domain knowledge are used to determine the significance of every single feature. For our case, we utilized correlation coefficients and exploratory data analysis for feature selection. This involved identifying and retaining attributes that significantly impact the accuracy and robustness of the model's predictions [29, 30].

3.3 Hybrid Model

Studies have demonstrated that adding hybrid learning methods to deep learning systems can boost their generalization abilities and give rise to more accurate predictions, thereby paving the way for the development of robust hybrid machine learning models that leverage the strengths of multiple algorithms [31, 32].

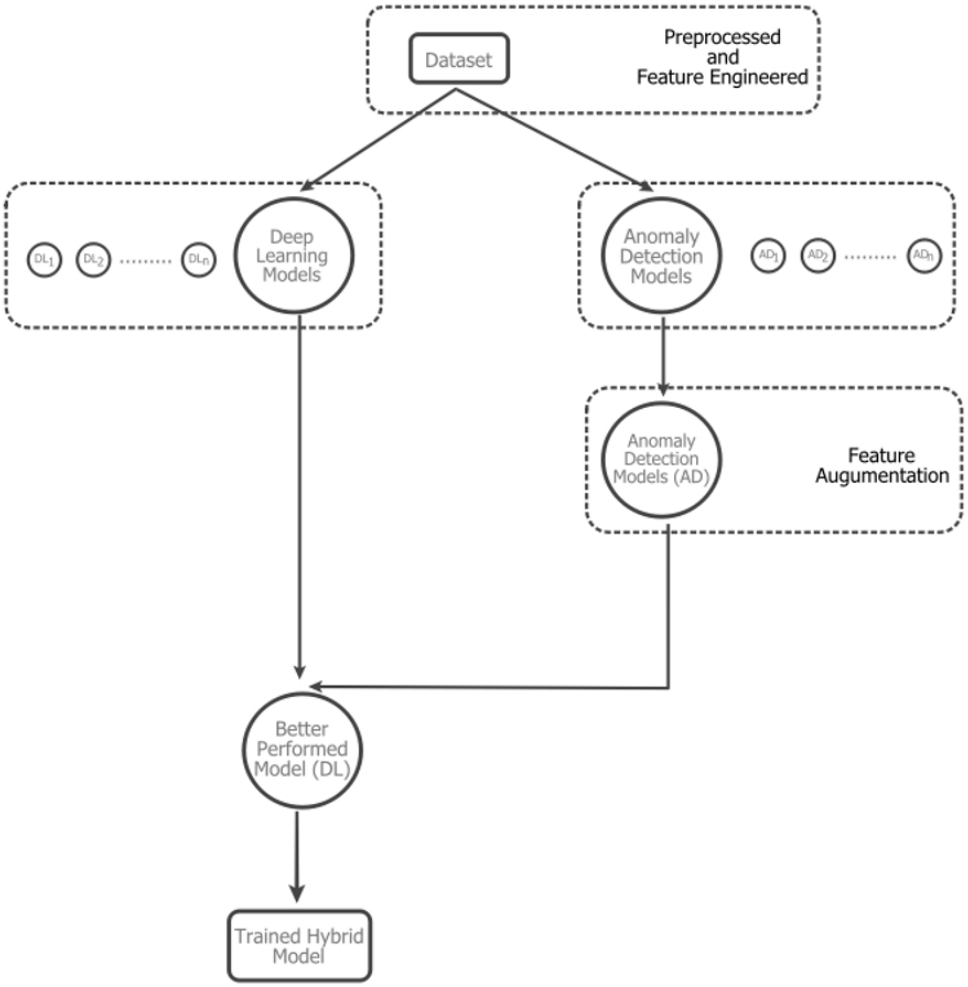

Figure 3.5: Proposed Hybrid Machine Learning Model

The model consists of five different stages. The first phase introduces the deep learning models that were used in initial experiments, and phase two demonstrates the anomaly detection methods applied at the initial dataset. Phase three involves selecting models that were best-performing among the selected Deep learning models, and likewise in phase four, we also selected the best-performing models from anomaly detection. In phase five, the hybrid model incorporates additional features by integrating prediction scores generated from the anomaly detection model and feeding them into the deep learning model alongside the original dataset features to enhance predictive capabilities. By augmenting the original data with informative feature, the hybrid model aims to improve model performance, enabling better generalization to unseen data and overall enhancement of predictive accuracy.

3.3.1 Deep Learning Model Selection

To identify the optimal Deep Learning Model, particularly from neural networks, for mobile money fraud prediction, we conducted a careful selection process. This important step involved evaluating three distinct models: Multi-layer Perceptron (MLP), Restricted Boltzmann Machines (RBM), and Probabilistic Neural Network (PNN).

Each of these models was chosen for its demonstrated advantages, including improved classification performance and widespread use in addressing prediction challenges in related previous research.

3.3.1.1 Probabilistic Neural Network (PNN)

The Probabilistic Neural Network (PNN) is a type of artificial intelligence network that comes from Bayesian networks and is used to solve classification problems. They work by looking for patterns in data collections and then trying to come up with mathematical relationships between the patterns in order to classify new data points [33, 12]. This means that the PNN is able to yield estimated possible values that describe the possibility of all classes after receiving the input information. PNNs are often known for their high training speed and their incredible ability to perform well in noisy environments [33, 12].

A Probabilistic Neural Network consists of four layers: the input layer, the pattern layer, the summation layer, and the output layer. Specifically, the input layer corresponds to each feature of the input data, and the pattern layer represents each training sample through the use of Gaussian activation functions for estimating probability density function [12].

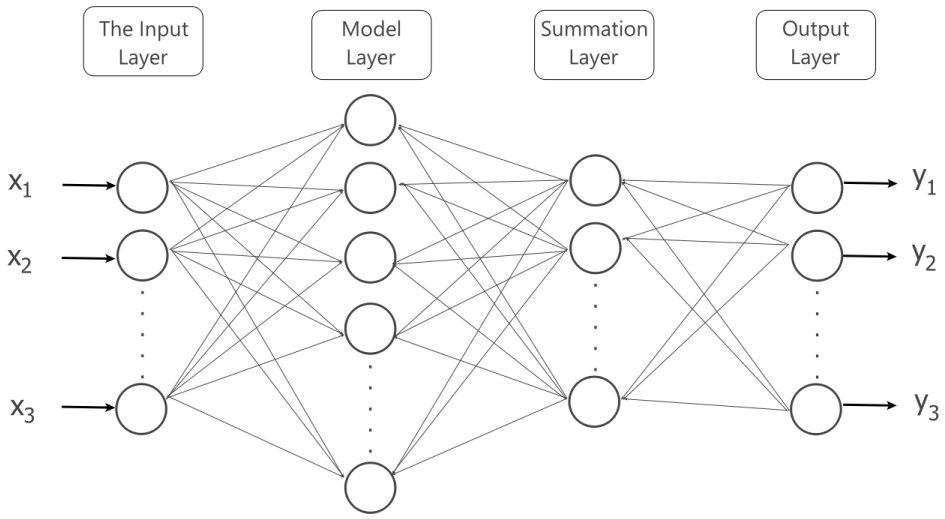

Figure 3.6: Probabilistic Neural Network (PNN) Architecture

Class Probability Estimation:

$$P(C_k \mid x) = \frac{1}{N_k} \sum_{i \in C_k} f_i(x) \tag{3.2}$$

where:

- N_k is the number of training samples in class C_k,
- $f_i(x)$ is the Gaussian kernel output for the i-th sample in class C_k.

The summation layer aggregates these outputs by class, adding up the Gaussian yields, while on the other end, the output layer gives the class which has the highest chance of occurring as the most-likely class hence providing a direct and interpret-able probabilistic output for classification [12].

3.3.1.2 Restricted Boltzmann Machine (RBM)

The Restricted Boltzmann Machine (RBM) is a generative stochastic neural network that can learn a probability distribution over its set of inputs. RBMs consist of two layers: a visible layer and a hidden layer. There are no connections inside individual layers opposite to traditional neural networks; each visible unit is connected to all hidden units, and vice versa. RBMs can be used for dimension reduction, classification, regression, collaborative filtering, feature learning, and topic modeling [34, 6].

A Restricted Boltzmann Machine (RBM) is made up of an input layer (visible units) and an output layer, with no interconnections within the layers. The visible layer represents the input characteristics while in the hidden layer, there are features operating upon visible units using weighted connections [6].

Probability of Hidden Unit Activation:

$$P(h_j = 1 \mid v) = \sigma(b_j + \sum_i v_i w_{ij}) \tag{3.3}$$

where:

- $\sigma(x)$ is the sigmoid function: $\sigma(x) = \frac{1}{1+e^{-x}}$,
- v represents the visible units,
- h_j represents the hidden unit,
- b_j is the bias term for the hidden unit j,
- w_{ij} is the weight connecting visible unit i to hidden unit j.

RBM employs a bipartite graph structure in order to capture interactions between these layers. In a process known as Gibbs sampling, probabilities based on the input data are used to set the activation's of the hidden units [6].

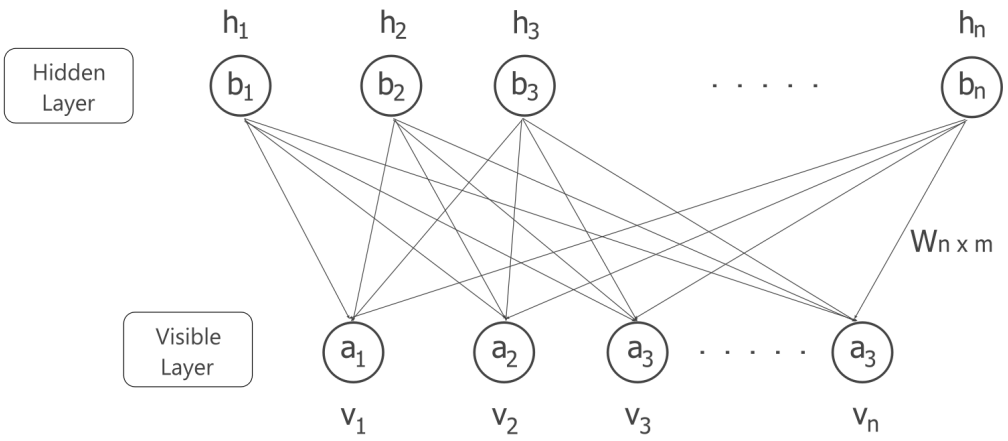

Figure 3.7: Restricted Boltzmann Machine (RBM) Architecture

3.3.1.3 Multi-Layer Perceptron (MLP)

The Multi-Layer Perceptron (MLP) model is a type of feed-forward artificial neural network model with multiple layers of nodes, each node being fully connected to all nodes in the subsequent layer. The MLP is designed to map input features to target output through a series of learned weights and biases. Since it can capture complex, non-linear relationships within the data this model is well suited to classification tasks [35, 36].

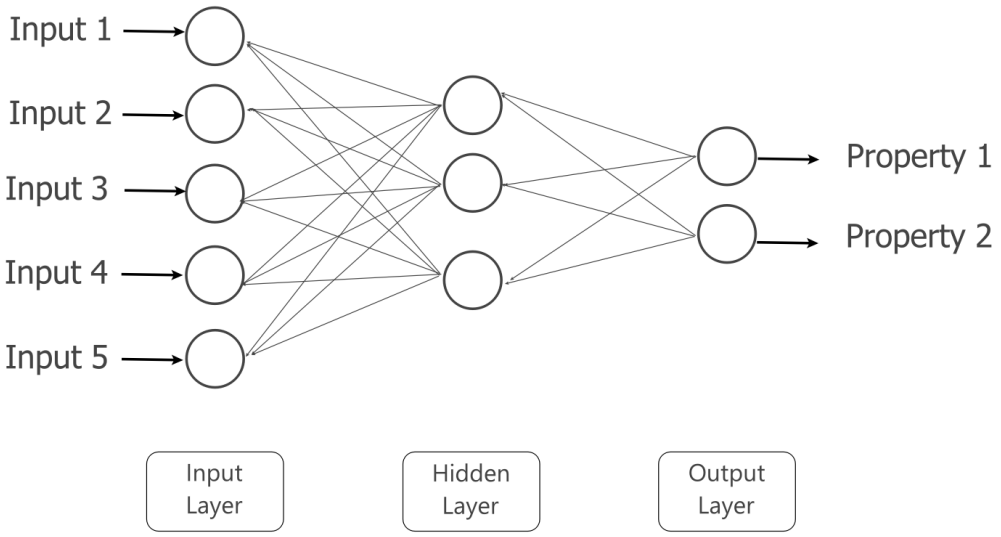

Figure 3.8: Multi-Layer Perceptron (MLP) Architecture

The Rectified Linear Unit (ReLU) (Rectified Linear Unit) activation function is utilized in each hidden layer to ensure a non-linear pattern is introduced into the model, while for binary classification, the soft-max activation function is applied; as for multi-class classification, the output layer uses the sigmoid activation function based on the type of classification task [35, 36].

Rectified Linear Unit (ReLU):

$$f(x) = \max(0, x) \tag{3.4}$$

where:

- $f(x)$ is the ReLU activation function,
- x is the input to the function.

In a Multi-layer Perceptron (MLP), the forward propagation through the output layer involves computing the activation's of output neurons using weighted sums of activation's from the preceding hidden layer. Each output neuron's activation is adjusted by a bias term and transformed by an activation function such as softmax or ReLU for classification tasks. This process yields the final predictions or outputs of the MLP, representing the network's response to the input data [35, 36].

Forward Propagation (Output Layer):

$$\hat{y}_k = \sigma\left(\sum_{j=1}^{m} w_{jk}^{(2)} z_j + b_k^{(2)}\right) \tag{3.5}$$

Where:

- $\hat{y}_k$ is the predicted output for class k,
- $\sigma(x)$ is the softmax function: $\sigma(x)_k = \frac{e^{x_k}}{\sum_{c=1}^{C} e^{x_c}}$ for C classes,
- $w_{jk}^{(2)}$ is the weight connecting hidden neuron z_j to output neuron k,
- $b_k^{(2)}$ is the bias term for output neuron k.

3.3.2 Anomaly Detection Model Selection

To enhance our fraud detection model, we incorporated three anomaly detection-based models: Autoencoder, Isolation Forest, and Local Outlier Factor (Local Outlier Factor (LOF)). These models were selected for their proven performance, including anomaly detection ability and widespread adoption in anomaly detection challenges within related research.

3.3.2.1 Isolation Forest

Isolation Forest (also known as iForest) is an unsupervised learning algorithm for detecting anomalous instances in a dataset. Instead of using the common measure of density or distance, it isolates anomalies by partitioning datasets recursively [37, 38, 39].

The key idea is that only a few anomalies are different, making them quicker to separate. The isolation forest will create many binary trees or isolation trees, selecting randomly one feature and then dividing it at some point chosen uniformly between that feature's minimum and maximum values. Anomalies will likely be found near the root of the tree due to their uniqueness [37, 38].

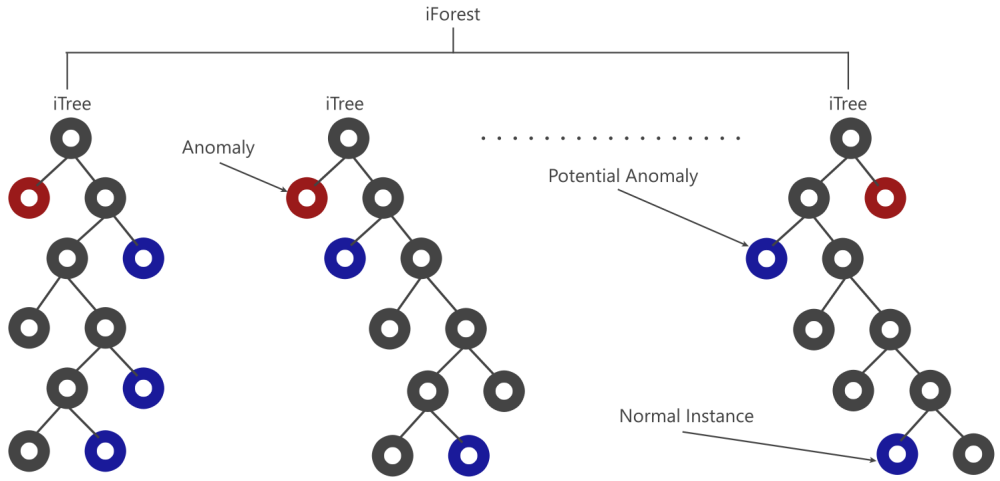

Figure 3.9: Isolation Forest Architecture

The architecture of the Isolation Forest consists of a group of isolation trees each of which is formed from a segment of the training data that is divided up repeatedly until all the dots are isolated or until it reaches a fixed depth. Nodes in each tree correspond to divisions based on randomly selected features at random thresholds [37, 38, 39].

Anomaly Score:

$$s(x, n) = 2^{-\frac{E(h(x))}{C(n)}} \tag{3.6}$$

where:

- $E(h(x))$ is the average path length of x across all trees,
- $C(n)$ is the average path length of unsuccessful searches in a binary search tree, approximated by $C(n) = 2H(n-1) - \frac{2(n-1)}{n}$,
- $H(i)$ is the i-th harmonic number, $H(i) = \ln(i) + \gamma$ (Euler's constant).

The depth isolates a data point, which consists of an anomaly score. A forest derives the general anomaly score through combining these scores. Such an architecture is parallelized inherently so that it is possible to build every tree on its own [38, 39].

3.3.2.2 Local Outlier Factor (LOF)

The Local Outlier Factor (LOF) is an unsupervised anomaly detection algorithm that detects anomalies by comparing the density deviations that exist for any given data point with the one that is close by. Instead of taking into account each element individually like with global methods for finding outlying values, this area-specific method takes the nearby surrounding area into consideration, which implies that it can spot deviations occurring in close proximity to density points, in contrast to other global methods [37, 40].

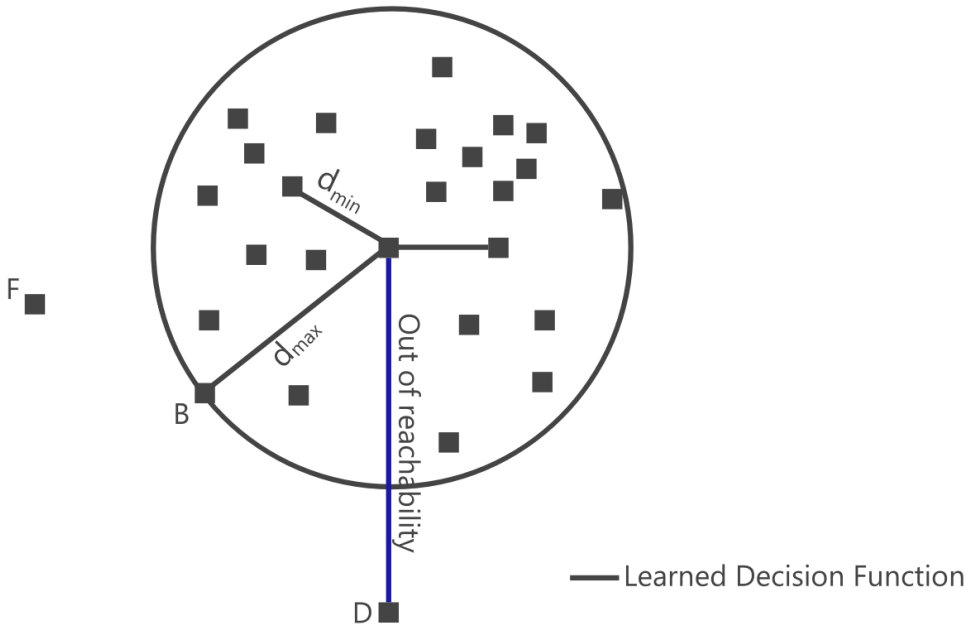

Figure 3.10: Local Outlier Factor Architecture

The presence of one or more abnormal points in the entire dataset is detected by the Local Outlier Factor (LOF) model [37, 40]. In this method, the density of points in the local neighborhood with regards to various data points is calculated.

Local Outlier Factor (LOF):

$$\text{LOF}_k(p) = \frac{\sum_{o \in N_k(p)} \frac{\text{lrd}_k(o)}{\text{lrd}_k(p)}}{|N_k(p)|} \tag{3.7}$$

where:

- $N_k(p)$ is the set of k-nearest neighbors of p,
- $\text{lrd}_k(p) = \left(\frac{\sum_{o \in N_k(p)} \text{reach-dist}_k(p,o)}{|N_k(p)|} \right)^{-1}$ is the local reachability density of p.

For each single data point, the number k data points which are close to this point are found in the dataset and then the distance is computed [37, 40].

3.3.2.3 Autoencoder

An Autoencoder, a type of artificial neural network, is used for unsupervised learning. It aims to learn a concise representation of a dataset (encoding) for a set of data, for dimensional reduction purposes or feature learning [41, 42].

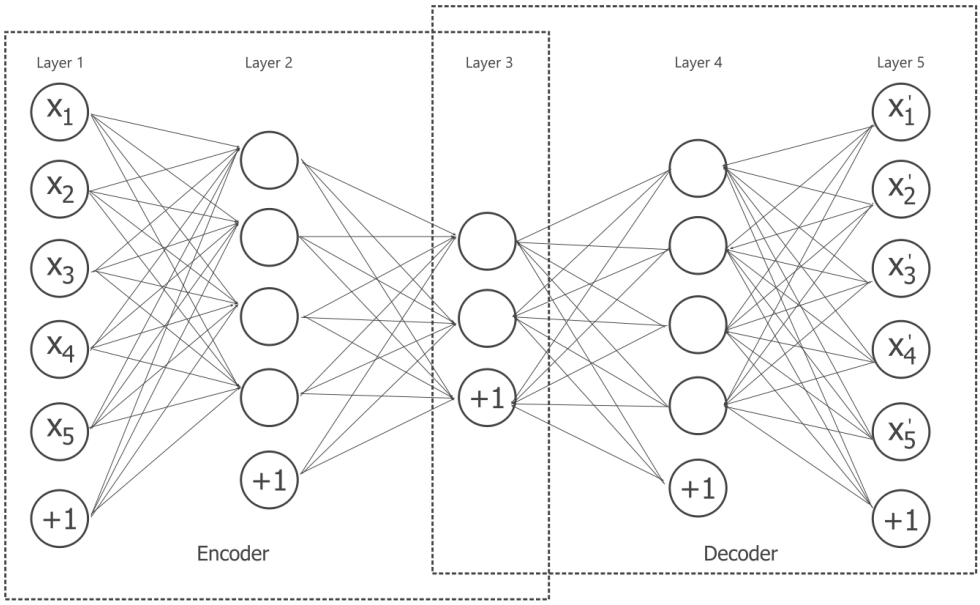

Figure 3.11: Typical Autoencoder Architecture

There are two main parts to the Autoencoder: encoder and decoder. Encoder reduces input data into a lower-dimensional latent space; in return decoder translates it back into input data [41, 42].

Latent Representation:

$$z = f(W_e x + b_e) \tag{3.8}$$

where:

- W_e is the weight matrix of the encoder,
- b_e is the bias vector of the encoder,
- f is an activation function (e.g., ReLU).

The network is trained to minimize the reconstruction error, which measures the difference between the input and the reconstructed output [41, 42].

Reconstruction Error:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \tag{3.9}$$

where:

- n is the number of data points,
- x_i is the original input data,
- $\hat{x}_i$ is the reconstructed data.
- x_i is the original input data,
- $\hat{x}_i$ is the reconstructed data produced by the Autoencoder.

3.3.3 Best Performing Model Selection

For our final testing phase, we conducted thorough evaluations using the Pay Sim dataset to assess different models. The objective was to understand how well deep learning models performed alongside anomaly detection models. This systematic evaluation aimed to identify the top-performing models that could be closely integrated into our fraud detection system. We prioritized models that demonstrated high accuracy and robustness in fraud classification within the dataset. This method allowed us to rank models based on their consistent ability to detect fraudulent activities, providing a solid foundation for improving our detection system.

3.3.4 Hybrid Model

Our proposed fraud prediction model is a hybrid approach that combines the strengths of anomaly detection and deep learning. We aim to leverage the best-performing anomaly detection model by integrating its predictions as additional features into a top-performing deep learning model. This feature augmentation strategy allows our hybrid model to learn new trends in the data as it evolves over time. The hybrid model incorporates additional features by integrating prediction scores generated from the anomaly detection model and feeding them into the deep learning model alongside the original dataset features as illustrated in 3.5.

3.3.5 Classification on a Unseen Dataset

3.3.5.1 K-Means Cluster

K-means clustering is a popular unsupervised machine learning algorithm used for partitioning a dataset into a set of distinct, non-overlapping subgroups, or clusters, where each data point belongs to the cluster with the nearest mean. The algorithm aims to minimize the variance within each cluster by iteratively assigning data points to clusters and updating the cluster centers, or centroids [43, 44].

A key formula in K-means clustering is the Euclidean distance, which measures the distance between data points and centroids [44], defined as:

Distance between Data Point and Centroid:

$$d(x, c) = \sqrt{\sum_{i=1}^{n}(x_i - c_i)^2} \tag{3.10}$$

where:

- x represents a data point,
- c represents a centroid.

To evaluate our proposed model's effectiveness in detecting "ever-changing" fraud patterns, we employed a dynamic approach to dataset management. Initially, we clustered the dataset based on customers' spending habits. This strategy ensured that our proposed algorithm was not trained uniformly on the same data set and allowed us to capture various transaction patterns. By using the clustered dataset that showcased different transaction behaviors compared to those seen during the model training phase, we aimed to test the model's adaptability to new fraud trends.

We subsequently concentrated on a portion of the dataset, conducting all machine learning tasks on one segment at a time. This data segmentation approach was essential in assessing whether our model could effectively generalize to new, unseen data, especially data with distinct spending pattern trends that were not encountered during the initial training phase. By isolating specific subsets for evaluation, we could rigorously test the model's robustness and its ability to handle diverse and evolving fraudulent behaviors.

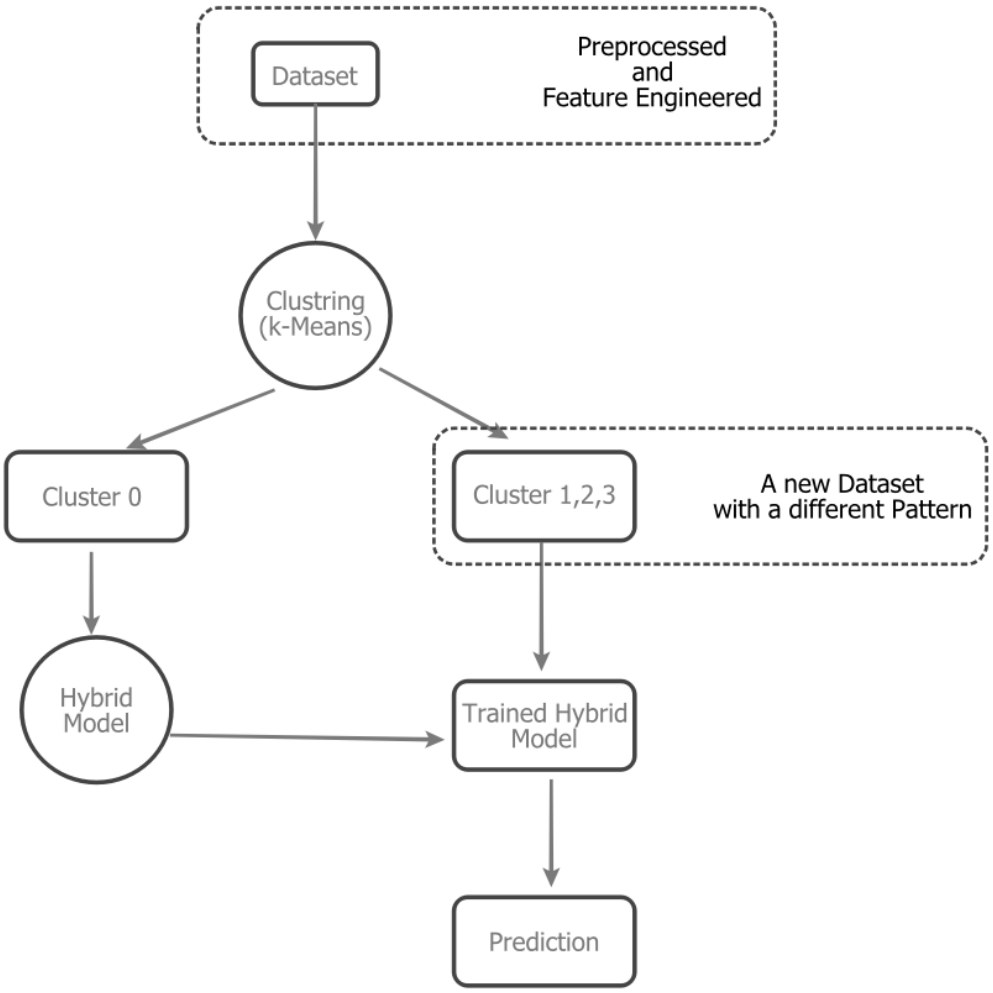

Figure 3.12: Classification on Unseen Dataset

3.3.6 Development Tools

3.3.6.1 Hardware Tools

Below is a table listing the hardware tools used throughout this research:

Number	Device Name	Usage for the Research
1	GPU	To train machine learning models
2	Hard disk	To store the datasets and different models
3	RAM	To enhance hardware performance

Table 3.2: Hardware tools

3.3.6.2 Software Tools

Different software and coding tools are utilized to implement the research through coding, and those are given below with descriptions:

Anaconda[2]: Anaconda is a distribution that includes Python and R, streamlining deployment and package management. It is renowned for its ability to provide a multitude of tools for machine learning and data analysis through a single installation. Conda, the package manager for Anaconda, facilitates the installation of libraries and offers a useful interaction with pip, allowing the installation of libraries not available through Anaconda's package manager. Anaconda is platform-independent, making it usable on Linux, macOS, and Windows.

Jupyter Notebook[3]:Jupyter Notebook is an open-source web application that integrates software code, explanatory text, computational output, and multimedia resources into a single document. It is integrated with the Anaconda environment and is widely used for editing code and viewing results.

Keras[4]: Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. It is designed to enable fast experimentation with deep neural networks and supports both convolutional and recurrent networks, as well as combinations of the two. Keras is user-friendly, modular, and extensible, making it a popular choice for building and evaluating deep learning models. It can be easily installed and used within the Anaconda environment.

3.3.7 Evaluation Metric

The evaluation metrics are important because they have an impact on how well different kinds of fraud detection systems will perform using machine learning algorithms for mobile money transactions. To determine the accuracy and reliability of such a prediction model, it is important to define and measure appropriate evaluation metrics that capture key aspects of the system's performance.

These four evaluation metrics, which are widely used to evaluate the performance of classification models:

[2] `https://www.anaconda.com/`
[3] `https://jupyter.org//`
[4] `https://keras.io/`

1. **Accuracy:** This metric measures the proportion of correctly classified instances out of all instances in the dataset. In the context of fraud detection, accuracy is important to ensure that the system correctly identifies both fraudulent and legitimate transactions. This is determined based on the count of predictions that were correctly divided by all predictions made [45].

$$\text{Accuracy} = \frac{\text{TruePositive} + \text{TrueNegative}}{\text{TruePositive} + \text{TrueNegative} + \text{FalsePositive} + \text{FalseNegative}} \tag{3.11}$$

2. **Recall:** This metric measures the proportion of true fraudulent instances that are correctly classified as fraud. In other words, it measures how well the system can identify fraudulent transactions without missing any. A high recall rate is important to minimize the number of false negatives, where fraudulent transactions are classified as legitimate. It is calculated as the ratio of the number of correctly identified fraudulent transactions to the total number of fraudulent transactions [45].

$$\text{Recall} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}} \tag{3.12}$$

3. **F1-score:** This metric is a weighted average of precision and recall and provides a single score that summarizes the performance of the system in identifying both fraudulent and legitimate transactions. A high F1-score indicates a balanced trade-off between precision and recall and is therefore desirable for a fraud detection system [45].

$$\text{F1-score} = \frac{2 * (\text{precision} * \text{recall})}{\text{precision} + \text{recall}} \tag{3.13}$$

4. **Precision:** This metric measures the proportion of true fraudulent instances out of all instances classified as fraud by the system. It reflects the ability of the system to correctly identify fraudulent transactions without falsely flagging legitimate transactions as fraudulent. It is calculated as the ratio of the number of correctly identified fraudulent transactions to the total number of transactions identified as fraudulent [45].

$$\text{Precision} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}} \tag{3.14}$$

Where:

- **TruePositive:** is the number of correctly identified fraudulent transactions.
- **TrueNegative:** is the number of correctly identified non-fraudulent transactions.
- **FalsePositive:** is the number of non-fraudulent transactions incorrectly identified as fraudulent.
- **FalseNegative:** is the number of fraudulent transactions incorrectly identified as non-fraudulent.

Chapter 4

Result and Discussion

This chapter shows the experimental setup; it also discusses the methodologies and procedures applied in the research conducted. After that, it describes the experimental results and conducts a comprehensive discussion that helps analyze and interpret the findings with regard to the research objectives.

4.1 Experimentation Setup

All Experiments were conducted using:

Software:

- Python 3.9.16
- TensorFlow 2.12.0
- Keras 2.12.0

Hardware:

- Dell Precision 7920 Tower server
- Intel(R) Xeon(R) Gold 6230R CPU
- 64GB RAM
- NVIDIA RTX A4000 GPU

Operating System:

- Ubuntu 22.04.2 LTS Server

4.2 Feature Engineering

As outlined in the methodology, feature engineering techniques have been applied to the dataset with the aim of enhancing its predictive ability and uncovering underlying patterns that might not be immediately apparent in the raw data. This enables a dual evaluation to be conducted, allowing for a direct comparison of model performance and the measurement of improvements made by feature engineering. The significance of Feature-Engineered features over original ones can be assessed by observing evaluation metrics like accuracy. Through a thorough comparison of the performance on both datasets, the strength and generalization ability of our models can be validated, ensuring that the enhancements from feature engineering are both meaningful and reliable in real-world applications.

We split the dataset using random division into an 80% training set and a 20% testing set. This separation enabled us to train the models on the larger training set, learning data patterns, and then evaluate their performance on the independent testing set, ensuring reliable assessments of predictive capability.

4.2.1 Feature Creation

Analyzing Time Distribution

For this step, we examine how the distribution of time data within our dataset could provide additional information on transaction patterns.

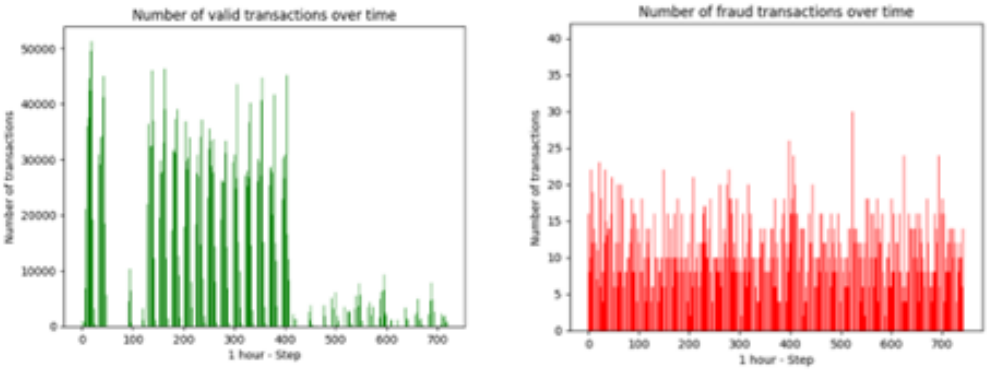

Figure 4.1: Analyzing Time Distribution

Our analysis indicates that most genuine transactions occur mainly within two-time windows, specifically from time step 0 to time step 60 and from time step 110 to time step 410. On the other hand, fraudulent transactions seem to have almost the same consistent frequency during different hours.

This observation motivates further investigation into potential trends in time or temporal patterns and what they may imply for fraud detection algorithms. To analyze the temporal patterns in more comprehensive details, we will investigate transactions that happen at particular hours of the day and on specific days of the week.

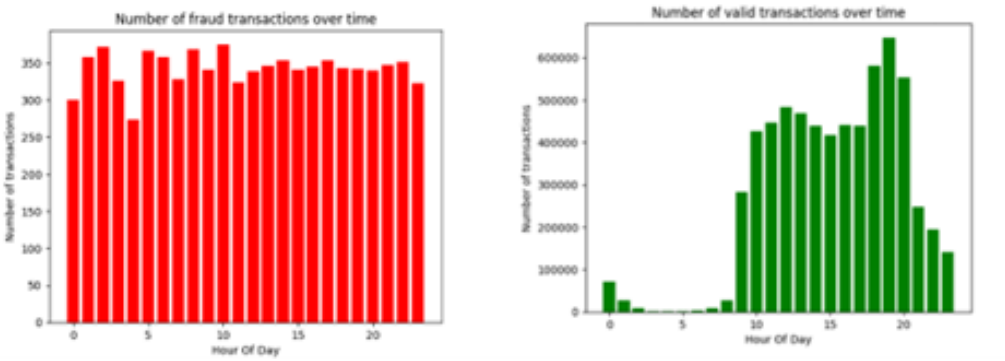

Figure 4.2: Analyzing Time Distribution: Hour of the Day

Our assessment has revealed that between 0 and 9 o'clock there are very few valid transactions. However, cases of fraud occur at an unchanging pace during all 24 hours, even during that time span.

In order for us to better represent the dynamics that vary with time, and maybe improve how well predictions are made by it; we suggest introducing another column-based feature on the time interval 'HourOfDay'. This feature will be created from the existing column "step" by calculating the reminder of each step number divided by 24, thus mapping each step to the hour it represents.The objective of this feature enhancement is to help the model gain more information about when transactions occur, and thereby assist in determining if activities such as fraud happen at specific hours.

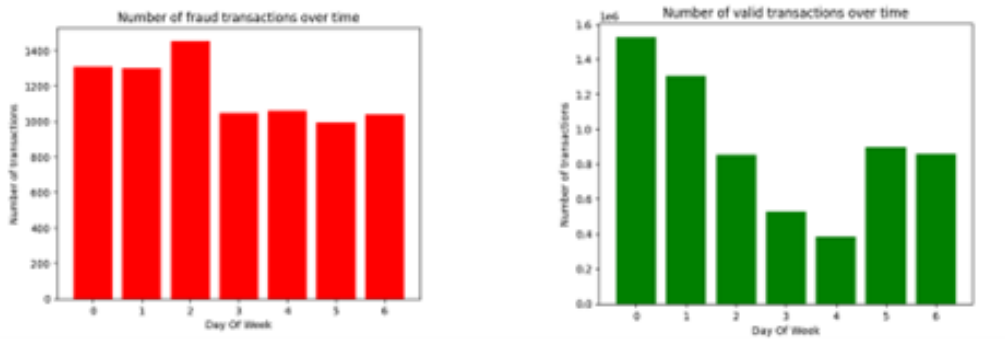

Figure 4.3: Analyzing Time Distribution: Day of the Week

Looking at the graphs that show when transactions happen during the week, there isn't much indication of fraudulent transactions occurring on particular days more than others. However, a noticeable decrease in the number of legitimate transactions takes place during two consecutive days of the week.

This observation implies that fraudulent activities do not show any noticeable pattern from one day to another day of the week; however, there might be days of the week with less frequent valid transactions. This could serve as a valuable feature for the models to learn from resulting in a better predictive ability.

Account Holders ID

We introduce a new feature that is created by extracting the first letter of each value from 'nameOrig' and 'nameDest' columns appended to the dataset. The feature makes use of existing account types within the records as source of its information; hence it is denoted as "CC" for Customer to Customer, "CM" for Customer to Merchant, and "MC" for Merchant to Customer.

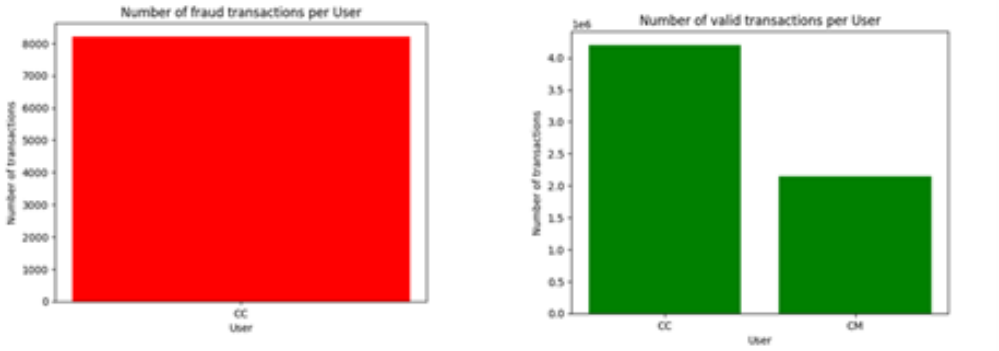

Figure 4.4: Analyzing: Account Holders ID

Upon analysis, it becomes evident that most fraudulent transactions happen in the category of "customer to customer". Interestingly, there are no transactions involving Merchants to Customer or Customer to Merchants. Mobile money transactions in nature put merchants into the business process and assist customers in sending funds. This observation aligns with the nature of mobile money transactions, because it is uncommon for merchants to use mobile money platforms to carry out transactions or make withdrawals from these platforms. As a result, our focus is mainly on transactions initiated by customers, especially those that are customer to customer (CC) since it is where most of the fraud transactions take place within the dataset.

New Origin Balance

The feature 'newbalanceorg' tells us about the amount left in the account of the individuals who initiated the transaction after it was completed.

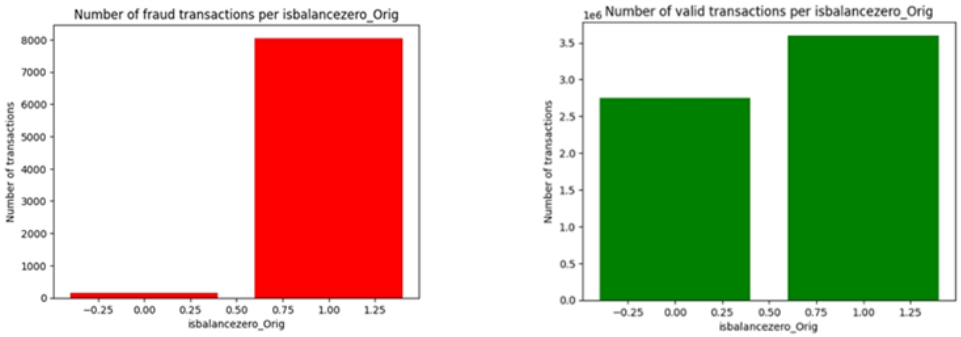

Figure 4.5: Analyzing: New Origin Balance

After looking at the balance values of the transactions, in the event that any fraudulent activity occurred, the balance in the account holders after would be reduced to zero in 95% of fraudulent transactions.This observation suggests that it may be important for the model to learn whether the initial balance becomes 0 or not after a transaction.

We aim to take advantage of the significant association between zero origin balance and fraudulent transactions by including this feature in our predictive model. This feature has the capability to improve the model's ability. Based on this insight, we have also created a corresponding feature for the destination account holders. This feature similarly indicate whether the balance in the destination account becomes zero after a transaction

4.2.2 Feature Splitting

Exploring Cents in Transaction amounts

We split the cents from the amount of a transaction in order to generate a new attribute named "Cents Amount" which denotes only the exact cents that are taken separately from the total amount of transaction.

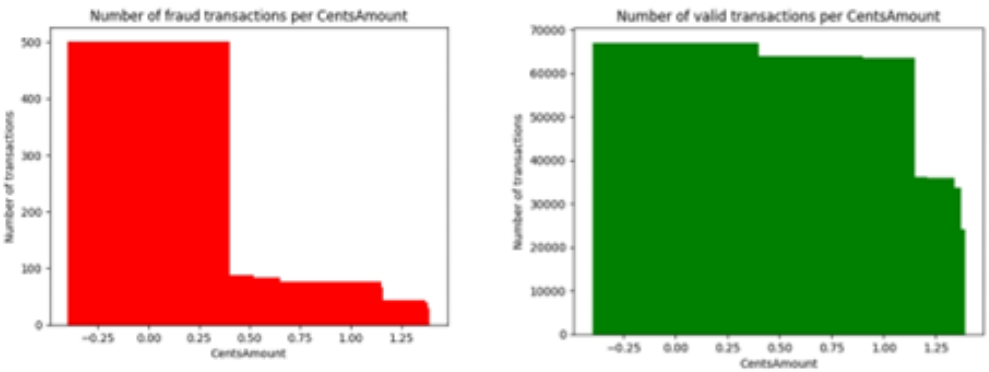

Figure 4.6: Analyzing Cents in Transaction amounts

When this feature is examined, it is noted that; there is a clear trend where most cases where frauds happen include zero cents as part of the money involved and for about 90% cases in the dataset under consideration, there are no cents involved in them. On the other hand, cent amounts are more spread out in valid transactions hence indicating that transactions with cents are more common and diverse among usual transactions. It implies that having no cents may differentiate between fraud and non-fraudulent transactions. Based on this observation, we created a binary feature called ”Cents” to capture this relationship. This feature indicates whether the transaction amount in ”CentsAmount” includes any cents, enhancing our ability to differentiate between fraudulent and non-fraudulent transactions.

4.2.3 Feature Selection

Analyzing balances before and after transactions

Upon inspecting the dataset, some significant portions of observations had errors, especially in making calculations for amounts in the accounts before and after the transactions. Notably, 85% of the errors made on the balances were observed in the sending or the transaction initiator account's balances while 100% were observed receiving the account's balances. The cause of these errors is unclear, making it challenging to correct them accurately. As a result, we decide to exclude these observations from our modeling process due to their irrelevance.

'isFlaggedFraud' Feature

The objective of the 'isFlaggedFraud' feature, in the dataset, aims to determine and label the possibly fraudulent transactions, especially the ones with high value transfers above and beyond 200,000. However, upon examination only 16 out of the 1,673,570 which exceed this benchmark are highlighted as fake. The low detection rate implies that the feature cannot differentiate well between legitimate and fraudulent transactions, hence, it is considered useless for modeling and would be removed from the dataset. By excluding this feature, we ensure that our model concentrates on those more informative features that significantly contribute meaningfully to detection of fraudulent activities; hence improving its general effectiveness and accuracy.

Assessing the Relevance of the 'Amount' Feature

While the 'Amount' feature has long been considered important in classification models for financial fraud detection, our analysis suggests it may not always be as relevant in determining the fraudulent nature of transactions. Traditionally, transaction sums were significant in the classification efforts, yet it seems that using just the 'Amount' attribute may not provide enough distinctive information to enable one accurately to classify the transaction as fraudulent or non- fraudulent transactions.

Fraudulent activities can occur in different monetary amounts which makes the 'Amount' variable biased on its own, if considered in isolation. Relying solely on this feature might introduce bias into the models; Instead, efforts should be directed towards deriving other features which show spending patterns thereby providing more information on fraud behavior beyond just looking at transaction amounts.

Therefore, an attribute can be developed that measures deviation of transaction amounts from the customer's usual spending behavior. The new feature can give a much clearer view on instances when transactions are different from the usual ones, thus increasing the effectiveness in detecting frauds. To accomplish this, we calculated the transaction frequency for each customer and computed the deviation of each transaction amount from the average amount spent by that customer. This resulted in the creation of the "amount deviation" feature for both the sender and receiver, along with an indicator for transaction frequency. These features collectively provide a more nuanced understanding of transaction patterns and enhance our ability to identify potentially fraudulent activities.

Following exploratory data analysis and correlation coefficient assessments, we identified several features in our initial dataset (dataset one) that showed weak or negligible linear relationships with the target variable "isFraud"[46]. These features, including "CentsAmount", "Freq_Org", "amount_deviationOrig", and "freq_Dest", were initially removed from dataset one due to their low correlation coefficients.

To further investigate the possibility of non-linear relationships in our dataset, we have preserved these removed features in dataset two. By including these features in dataset two, we aim to assess whether they exhibit non-linear associations with the target variable 'isFraud'. By utilizing both the original dataset and the two feature-engineered datasets, we aim to assess the performance gains achieved through our feature engineering efforts. This comprehensive approach allows us to leverage both linear and non-linear insights to enhance our understanding and predictive accuracy in fraud detection

4.3 Experiment 1: Deep Learning Models Selection Result

Selecting the most effective deep learning model is important for achieving optimal performance in complex tasks. In this study, we aim to identify the best-performing model among three distinct deep learning architectures. By systematically experimenting with and evaluating these models, we seek to determine which one offers superior accuracy, robustness, and efficiency.

The chosen model will then serve as a key component in the development of our hybrid model, designed to leverage the strengths of multiple approaches for improved predictive capability. This selection process is essential to ensure that our hybrid model is built on a solid foundation, capable of delivering high-quality results in predicting fraud in mobile money transactions.

Overfitting occurs when the model learns the noise and details in the training data to the extent that it negatively impacts the model's performance on new data. To mitigate this, several techniques and parameters can be employed during the training process. The following table outlines the parameters used in our deep learning models to avoid overfitting:

Parameter	**Value**
Learning Rate	0.0001
Early Stopping	True
Validation Fraction	0.1
Patience (Iterations)	10

Table 4.1: Parameters used to avoid overfitting during training for deep learning models

4.3.1 Restricted Boltzmann's Machines (RBM)

We used a Restricted Boltzmann Machine to train on the dataset, and the results are shown in Table 4.2 . To analyze and compare the model's performance, we measured its accuracy on different parts of the dataset, including both the original and feature-engineered versions. This method helped us thoroughly assess how well the model works in different class distribution and on different dataset.

For the original dataset, a 50:50 ratio achieved an accuracy of 78.36%, providing a baseline when both classes are equally represented. As the class imbalance increased to 60:40, accuracy improved to 80.81%, but further imbalances to 70:30 and 80:20 saw a decrease to 79.93% and 79.04%, respectively.

Dataset	**50 50**	**60 40**	**70 30**	**80 20**
Original	78.36%	80.81%	79.93%	79.04%
Feature-Engineered Dataset One	82.96%	81.28%	85.83%	89.91%
Feature-Engineered Dataset Two	82.58%	81.28%	85.60%	89.85%

Table 4.2: RBM Model Accuracy of different datasets with varying class ratios

This indicates a potential reduction in the model's sensitivity to the minority class with increasing imbalance. For Feature-Engineered Dataset One, the accuracy significantly improved across all ratios, starting at 82.96% for the 50:50 ratio. However, accuracy slightly decreased to 81.28% for the 60:40 ratio but then increased to 85.83% and peaked at 89.91% for the 70:30 and 80:20 ratios, respectively. This suggests that the first feature-engineered dataset is more robust to class imbalance.

Similarly, Feature-Engineered Dataset Two showed an initial accuracy of 82.58% for the 50:50 ratio, consistent with the first feature-engineered dataset. For the 60:40 ratio, accuracy remained at 81.28%, identical to the first feature-engineered dataset, and increased to 85.60% and 89.85% for the 70:30 and 80:20 ratios, respectively. These results indicate that both feature-engineered datasets handle class imbalance effectively, significantly enhancing RBM performance compared to the original dataset.

4.3.2 Probabilistic Neural Network (PNN)

The performance of the Probabilistic Neural Network (PNN) on various datasets and class ratios demonstrates the impact of class distribution and feature engineering on classification accuracy. For the original dataset, the model achieved an accuracy of 77.81% with a balanced 50:50 ratio of valid to fraudulent transactions. As the class imbalance increased, accuracy decreased notably: 72.29% for a 60:40 ratio, 68.68% for a 70:30 ratio, and 64.34% for an 80:20 ratio. This decline suggests that the PNN's ability to accurately classify fraudulent transactions diminishes as the dataset becomes more imbalanced.

Dataset	50 50	60 40	70 30	80 20
Original	77.81%	72.29%	68.68%	64.34%
Feature-Engineered Dataset One	78.36%	73.36%	69.40%	65.25%
Feature-Engineered Dataset Two	78.36%	73.36%	69.40%	65.25%

Table 4.3: PNN Model Accuracy of different datasets with varying class ratios

When applying feature engineering, the PNN showed slight improvements. For Feature-Engineered Dataset One, the accuracy was 78.36% with a 50:50 ratio, marginally better than the original dataset. As the cPNN Model Accuracy of different datasets with varying class ratioslass imbalance increased, the accuracy followed a similar declining trend, achieving 73.36% for a 60:40 ratio, 69.40% for a 70:30 ratio, and 65.25% for an 80:20 ratio. This pattern indicates that while feature engineering provided some benefit, the PNN still struggled with increased class imbalance.

The results for the second feature-engineered dataset were identical to those of the first the accuracy was again 78.36% for the 50:50 ratio and decreased to 73.36% for the 60:40 ratio, 69.40% for the 70:30 ratio, and 65.25% for the 80:20 ratio. These results suggest that the second feature-engineered dataset did not provide any additional advantage over the first.

Overall, feature engineering provided a slight improvement in PNN performance across different class ratios compared to the original dataset. However, significant class imbalances still led to decreased accuracy, underscoring the challenge of accurately detecting fraud in imbalanced datasets even with enhanced features.

4.3.3 Multi-Layer Perceptron (MLP)

The performance of the Multi-Layer Perceptron (MLP) on various datasets and class ratios showcases the model's robustness and the positive impact of feature engineering on classification accuracy. For the original dataset, the model achieved an accuracy of 85.64% with a balanced 50:50 ratio of valid to fraudulent transactions. As the class imbalance increased, the accuracy improved: 86.68% for a 60:40 ratio, 87.55% for a 70:30 ratio, and 89.05% for an 80:20 ratio. This trend suggests that the MLP benefits from a slight majority class presence, potentially due to better learning of the underlying patterns in the larger class.

Dataset	50 50	60 40	70 30	80 20
Original	85.64%	86.68%	87.55%	89.05%
Feature-Engineered Dataset One	90.83%	92.33%	93.09%	94.75%
Feature-Engineered Dataset Two	91.79%	93.19%	93.41%	95.70%

Table 4.4: MLP Model Accuracy of different datasets with varying class ratios

For Feature-Engineered Dataset One, the accuracy showed a significant improvement across all class ratios. With a 50:50 ratio, the accuracy was 90.83%, and it increased as the imbalance grew, achieving 92.33% for a 60:40 ratio, 93.09% for a 70:30 ratio, and 94.75% for an 80:20 ratio. This indicates that feature engineering considerably enhances the MLP's ability to detect fraudulent transactions, even when they are in the minority.

Feature-Engineered Dataset Two demonstrated even higher accuracy gains. The model achieved 91.79% for the 50:50 ratio, 93.19% for the 60:40 ratio, 93.41% for the 70:30 ratio, and 95.70% for the 80:20 ratio. These results suggest that the second feature-engineered dataset provides additional advantages over the first, leading to even higher performance, especially in scenarios with significant class imbalances.

In summary, the MLP's performance improves with class imbalance in both the original and feature-engineered datasets, but feature engineering plays a crucial role in significantly enhancing accuracy. The second feature-engineered dataset demonstrates the highest performance, indicating the value of careful feature selection and engineering in developing robust models for fraud detection in mobile money transactions.

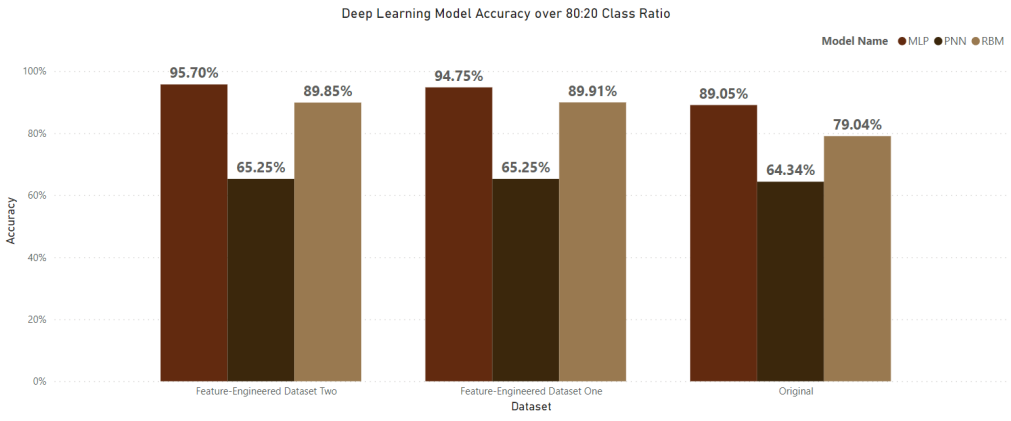

Figure 4.7: Accuracy of Deep Learning models over Class Distribution 80:20

In comparison to the other models, the Multi-Layer Perceptron (MLP) stands out as a formidable contender for mobile money fraud detection. With consistently high accuracy across various data setups, the MLP demonstrates robust performance unmatched by the Restricted Boltzmann Machine (RBM) and Probabilistic Neural Network (PNN). While the RBM and PNN showed moderate to declining accuracy trends with increasing dataset ratios, the MLP consistently improved its accuracy, showcasing its superior ability to leverage additional training data for enhanced predictive capability.

Feature engineering played a pivotal role in elevating the accuracy of all models, including the MLP, RBM, and PNN. By enriching the datasets with engineered features, the models were able to capture and leverage intricate patterns and relationships within the data more effectively. This led to substantial improvements in accuracy across all dataset configurations, particularly evident in the MLP's case, where the engineered features propelled accuracy to impressive levels. The success of feature engineering underscores its importance in enhancing the predictive capabilities of machine learning models for mobile money fraud detection.

4.3.4 Experiment 2: Anomaly Detection Models Selection Result

In Experiment 2, we focus on selecting the most accurate anomaly detection model among three prominent options: Isolation Forest, Auto-Encoder, and Local Outlier Factor (LOF). The primary objective is to identify the model with superior accuracy, which will serve as a pivotal component in our endeavor to develop a robust hybrid model for mobile money fraud detection. This hybrid model aims to adapt to evolving fraud detection challenges by combining the strengths of selected deep learning models with advanced anomaly detection techniques. By comparing the accuracy of these anomaly detection models, we aim to determine the most effective approach for augmenting features in our hybrid model, thereby enhancing its capability to detect and mitigate emerging fraudulent activities in mobile money transactions.

4.3.4.1 Isolation Forest

Training the dataset with Isolation Forest yielded results depicted in Figure X. To comprehensively evaluate and compare its performance, we employed an accuracy evaluation method. This approach enabled us to analyze how well the Isolation Forest model performed across various dataset ratios, considering both the original and feature-engineered datasets.

The performance of the Isolation Forest for anomaly detection on different feature-engineered datasets and class ratios provides insights into its effectiveness for identifying fraudulent mobile money transactions. For Feature-Engineered Dataset One, the model achieved an accuracy of 36.11% with a balanced 50:50 ratio of valid to fraudulent transactions. As the class imbalance increased, the accuracy showed significant improvement: 43.06% for both the 60:40 and 70:30 ratios, and 54.65% for the 80:20 ratio. This trend suggests that the Isolation Forest model becomes more effective at detecting anomalies when there is a higher proportion of valid transactions, possibly due to a clearer distinction between normal and fraudulent behavior in the dataset.

Dataset	**50 50**	**60 40**	**70 30**	**80 20**
Feature-Engineered Dataset One	36.11%	43.06%	43.06%	54.65%
Feature-Engineered Dataset Two	25.40%	34.77%	48.02%	54.15%

Table 4.5: Isolation Forest Model Accuracy of different datasets with varying class ratios

In contrast, Feature-Engineered Dataset Two exhibited lower initial accuracy but similar improvements with increasing class imbalance. The model achieved an accuracy of 25.40% with a 50:50 ratio, which improved to 34.77% for a 60:40 ratio, 48.02% for a 70:30 ratio, and 54.15% for an 80:20 ratio. However, as the imbalance increased, the model's performance improved, aligning closely with the trends observed in Feature-Engineered Dataset One.

Isolation Forest performed better on the highly imbalanced datasets because the model assumes anomalies are found in rare cases. When there is a higher proportion of valid transactions, the distinction between normal and fraudulent activities becomes more pronounced, making it easier for the model to identify the rare instances of fraud. Overall, these results indicate that the Isolation Forest's effectiveness for anomaly detection in mobile money transactions improves with class imbalance, with both feature-engineered datasets showing enhanced accuracy at higher ratios of valid to fraudulent transactions. The better performance at higher imbalances suggests that the Isolation Forest is more capable of isolating anomalies when there is a clearer majority of normal transactions, although the choice of feature engineering has a notable impact on the model's baseline performance.

4.3.4.2 Local Outlier Factor (LOF)

To evaluate the capability of the Local Outlier Factor (LOF) for anomaly detection, we conducted experiments across different feature-engineered datasets and class ratios to determine its effectiveness in identifying fraudulent mobile money transactions. For Feature-Engineered Dataset One, the model achieved an accuracy of 50.88% with a balanced 50:50 ratio of valid to fraudulent transactions. As the class imbalance increased, the accuracy improved significantly: 60.06% for a 60:40 ratio, 66.97% for a 70:30 ratio, and 74.34% for an 80:20 ratio. This trend indicates that the LOF model becomes more effective at detecting anomalies when there is a higher proportion of valid transactions, likely due to its capability to better distinguish the normal patterns from the outliers.

Dataset	**50 50**	**60 40**	**70 30**	**80 20**
Feature-Engineered Dataset One	50.88%	60.06%	66.97%	74.34%
Feature-Engineered Dataset Two	49.77%	59.53%	66.79%	74.00%

Table 4.6: Local Outlier Factor Model Accuracy of different datasets with varying class ratios

Similarly, Feature-Engineered Dataset Two exhibited comparable performance improvements with increasing class imbalance. The model achieved an accuracy of 49.77% with a 50:50 ratio, which improved to 59.53% for a 60:40 ratio, 66.79% for a 70:30 ratio, and 74.00% for an 80:20 ratio. The consistent improvement across both datasets suggests that the LOF model's performance is robust to different feature engineering approaches, enhancing its ability to detect fraud as the dataset becomes more imbalanced.

The improved performance of the LOF model on highly imbalanced datasets can be attributed to its underlying assumption that anomalies are more likely to be found in rare cases. When the dataset has a higher proportion of valid transactions, the distinction between normal and fraudulent activities becomes more pronounced, enabling the LOF model to identify the fraudulent transactions more effectively. Overall, these results indicate that the LOF's effectiveness for anomaly detection in mobile money transactions improves with class imbalance, demonstrating enhanced accuracy at higher ratios of valid to fraudulent transactions. The consistent performance across different feature-engineered datasets underscores the LOF model's robustness in isolating anomalies amidst varying levels of class imbalance.

4.3.4.3 Autoencoder

We conducted experiments to assess the capability of autoencoder for anomaly detection, utilizing error reconstruction as a feature. Our investigation aimed to determine its effectiveness across various feature-engineered datasets and class ratios for identifying fraudulent mobile money transactions. In Feature-Engineered Dataset One, the model achieved an accuracy of 50.00% with a balanced 50:50 ratio of valid to fraudulent transactions. As the class imbalance increased, the accuracy improved significantly: 62.40% for a 60:40 ratio, 73.77% for a 70:30 ratio, and 82.85% for an 80:20 ratio. This trend indicates that the autoencoder model becomes increasingly effective at detecting anomalies as the proportion of valid transactions decreases, allowing it to better differentiate between normal and fraudulent activities.

Dataset	**50 50**	**60 40**	**70 30**	**80 20**
Feature-Engineered Dataset One	50.00%	62.40%	73.77%	82.85%
Feature-Engineered Dataset Two	49.31%	60.15%	71.99%	80.71%

Table 4.7: Autoencoder Model Accuracy of different datasets with varying class ratios

Similarly, Feature-Engineered Dataset Two exhibited comparable performance improvements with increasing class imbalance. The model achieved an accuracy of 49.31% with a 50:50 ratio, which improved to 60.15% for a 60:40 ratio, 71.99% for a 70:30 ratio, and 80.71% for an 80:20 ratio. These results suggest that the autoencoder model's performance is robust across different feature-engineered datasets, demonstrating its capability to detect anomalies amidst varying levels of class imbalance.

Overall, the consistent improvement in accuracy with higher ratios of valid to fraudulent transactions indicates the autoencoder's effectiveness in identifying anomalies in mobile money transactions. Its ability to leverage error reconstruction as a feature makes it a promising approach for detecting fraudulent activities, especially in scenarios with significant class imbalance.

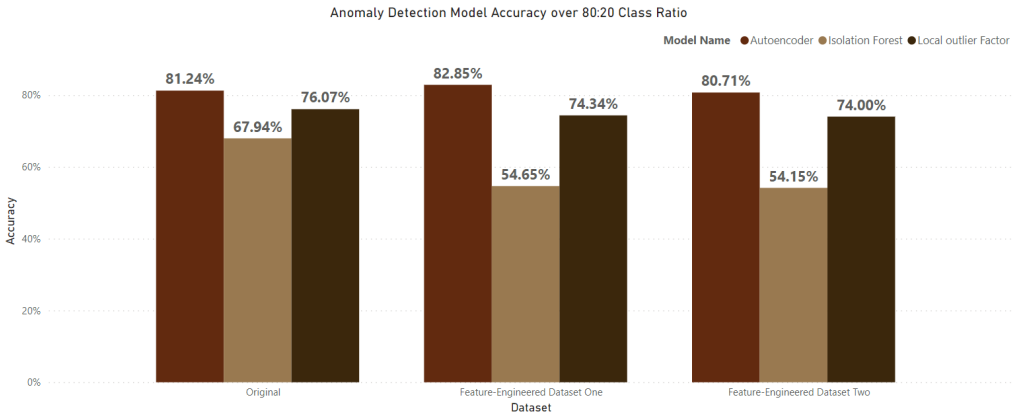

Figure 4.8: Anomaly Detection Model Accuracy

In comparison to the other models, Autoencoder emerges as a robust candidate for mobile money fraud detection and subsequent feature augmentation with selected deep learning models. While Isolation Forest demonstrates its own strengths in anomaly detection, Autoencoder's consistent performance across various dataset configurations and its adaptability to evolving fraud detection challenges make it a compelling choice for feature augmentation in hybrid models.

The implementation of feature engineering has significantly contributed to the improvement in accuracy across all models, including Autoencoder and Isolation Forest. By enriching the datasets with engineered features, the models have been able to capture and utilize intricate patterns and relationships within the data more effectively. This has led to substantial enhancements in accuracy across different dataset configurations, underscoring the importance of feature engineering in enhancing the predictive capabilities of machine learning models for mobile money fraud detection.

4.3.5 Experiment 3: Hybrid Model

Based on the results of our experiments, the deep learning model, specifically the Multi-Layer Perceptron (MLP), demonstrated superior performance in classification tasks. In contrast, during Experiment 2, the Autoencoder significantly outperformed both the Isolation Forest and Local Outlier Factor methods in detecting anomalies. These findings highlight the strengths of both models in their respective domains.

Given these insights, we propose the development of a hybrid model that leverages the strengths of both approaches. Specifically, we will utilize the Autoencoder for anomaly detection and use its outputs as augmented features for the MLP model. The Autoencoder's ability to accurately identify anomalies will enhance the dataset by providing additional, informative features that reflect the likelihood of fraudulent activity.

This augmented dataset, enriched with anomaly detection scores from the Autoencoder, will then be fed into the MLP model. The enhanced feature set will improve the MLP's capability to recognize and adapt to evolving fraud patterns in mobile money transactions. By combining the robust anomaly detection of the Autoencoder with the high-performance classification of the MLP, our hybrid model aims to achieve superior detection and classification accuracy, ultimately providing a more reliable and adaptive fraud detection system.

Autoencoder Feature Augmentation

The process of feature augmentation begins with An Autoencoder model is defined with an input layer matching the dimensionality of the scaled features, followed by an encoding layer with 14 neurons, and a decoding layer that reconstructs the input features. This Autoencoder is trained using only the normal transactions from the training set to learn the typical patterns of non-fraudulent data.

Once trained, the Autoencoder generates predictions for both the training and testing sets, allowing us to calculate the reconstruction error for each sample by measuring the mean squared differences between the original and reconstructed features. These reconstruction errors, which serve as indicators of anomaly likelihood, are then appended as additional features to both the training and test sets.

With the augmented feature sets (original features plus the reconstruction error), we train an MLPClassifier, a neural network model designed for classification tasks. Finally, this MLP model is used to make predictions on the test set, and its performance is evaluated using accuracy, precision, recall, and F1 score metrics, showcasing the effectiveness of the feature augmentation process in enhancing the model's ability to detect fraudulent activities.

The hybrid model combining Multi-Layer Perceptron (MLP) and Autoencoder, utilizing error reconstruction as a feature, demonstrates strong performance across different feature-engineered datasets and class ratios for detecting fraudulent mobile money transactions. In Feature-Engineered Dataset Two, the model achieved an accuracy of 91.67% with a balanced 50:50 ratio of valid to fraudulent transactions. As the class imbalance increased, the accuracy improved significantly: 94.27% for a 60:40 ratio, 94.52% for a 70:30 ratio, and 96.56% for an 80:20 ratio. This trend indicates the effectiveness of the hybrid model in accurately identifying anomalies, even in highly imbalanced datasets.

Ratio	**Accuracy**	**Precision**	**Recall**	**F1 Score**
50 50	91.67%	93.71%	89.68%	91.65%
60 40	94.27%	93.88%	91.55%	92.70%
70 30	94.52%	95.93%	85.37%	90.35%
80 20	96.56%	97.62%	84.16%	90.39%

Table 4.8: The Proposed Performance metrics with varying class ratios on Feature-Engineered Dataset Two

Moreover, the precision, recall, and F1-score metrics further validate the hybrid model's performance. Precision consistently remains high across all ratios, indicating a low false positive rate. Recall shows a slight decrease as the imbalance increases, suggesting a slight decrease in the model's ability to correctly identify all fraudulent transactions. However, the F1-score, which balances precision and recall, remains high, indicating the overall effectiveness of the hybrid model in detecting anomalies.

These results indicate that the hybrid model effectively leverages the Autoencoder's anomaly detection capabilities to enhance the MLP's classification performance. The high precision across all splits suggests that the model is particularly adept at correctly identifying true positives, which is crucial in fraud detection scenarios where false positives can be costly. However, the recall values indicate some variance, particularly in higher training proportions, suggesting a need for further tuning to balance sensitivity and specificity. Overall, the increasing accuracy and precision with larger training sets highlight the hybrid model's potential to improve fraud detection systems by adapting to evolving patterns in mobile money transactions.

Comparison of Hybrid Model and MLP Model Results

When comparing the performance of the Multi-Layer Perceptron (MLP) against the Hybrid model (Autoencoder + MLP) in detecting fraudulent mobile money transactions with an 60:40 ratio, using Feature-Engineered Dataset Two, notable differences emerge. The proposed model achieves a higher accuracy of 94.27% compared to the MLP's 93.19%, indicating it is better at making correct predictions overall. Furthermore, the proposed model's recall of 91.55% is significantly higher than the MLP's 87.6%, suggesting it is more effective at identifying true positives and reducing false negatives. Although theMLP model has a slightly higher precision of 94.80% compared to the proposed model's 93.88%, the overall performance of the proposed model is superior due to its improved recall.

Model	Accuracy	Precision	Recall	F1 Score
MLP	93.19%	94.80%	87.67%	91.10%
Proposed Model	94.27%	93.88%	91.55%	92.70%

Table 4.9: Comparison of Hybrid Model vs MLP on Feature-Engineered Dataset Two with 60:40 class Distribution

The F1 score, which balances precision and recall, is higher for the proposed model at 92.70%, compared to the MLP model's 91.10%. This indicates that the proposed model achieves a better balance between precision and recall, making it more robust in scenarios where both metrics are crucial. In summary, while the MLP model has a slight advantage in precision, the proposed model's higher accuracy, recall, and F1 score suggest it is a more effective and reliable model for Mobile money fraud detection.

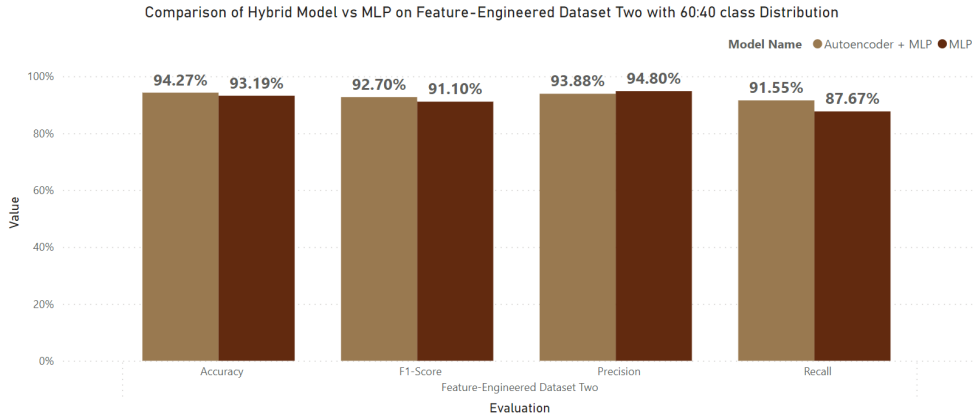

Figure 4.9: Comparison of Hybrid Model vs MLP on Feature-Engineered Dataset Two with 60:40 class Distribution

The F1-score, which balances precision and recall, provides a comprehensive measure of the model's overall performance. In this regard, the Hybrid model achieved an F1-score of 90.39%, while the MLP achieved an F1-score of 88.54%. This indicates that the Hybrid model strikes a better balance between precision and recall, resulting in higher overall performance in identifying fraudulent transactions.

Overall, these results suggest that the Hybrid model combining Autoencoder with MLP is more effective than the standalone MLP in detecting fraudulent mobile money transactions, particularly in scenarios with slightly class imbalance. The Hybrid model achieves higher accuracy, precision, and overall performance, making it a promising approach for fraud detection in mobile money transactions.

4.3.6 Experiment 4: Classification on Unseen Dataset

In Experiment 4, we employed the Hybrid model to evaluate its effectiveness in adapting to evolving fraud patterns. This involved clustering a dataset based on the spending patterns of customers, ensuring that it represented a distinct pattern from those the model had been trained on previously. By using this newly clustered dataset, we aimed to assess the performance and robustness of our model in identifying fraudulent activities amidst evolving patterns.

This approach allowed us to test the model's adaptability and efficacy in real-world scenarios, providing valuable insights into its potential to handle dynamically changing fraudulent behaviors. The experiment's findings will be instrumental in determining the model's capability to maintain high detection accuracy and reliability as fraud patterns continue to evolve.

In order to create a dataset that would challenge its adaptability to unseen data patterns. We employed the K-means clustering technique to achieve this. By clustering the dataset based on the deviations in transaction amounts, we were able to capture and segregate various spending patterns of the customers. This approach allowed us to generate distinct clusters that represented different spending behaviors, in the mobile money transactions.

This dataset was instrumental in evaluating how well the hybrid model could adapt to and accurately detect fraudulent transactions among varying and evolving customer spending patterns. This approach allowed us to simulate real-world scenarios where fraudsters may adopt new strategies or patterns to evade detection. By utilizing this newly clustered dataset, we sought to comprehensively evaluate the performance and robustness of our Hybrid model. Such insights are essential for enhancing the model's capabilities and ensuring its reliability in real-world fraud detection applications.

To determine the optimal number of clusters within our dataset, we employed the Elbow Method. This technique helps in identifying the appropriate number of clusters by plotting the within-cluster sum of squares (WCSS) against the number of clusters and observing the point where the rate of decrease sharply slows down, forming an "elbow" shape.

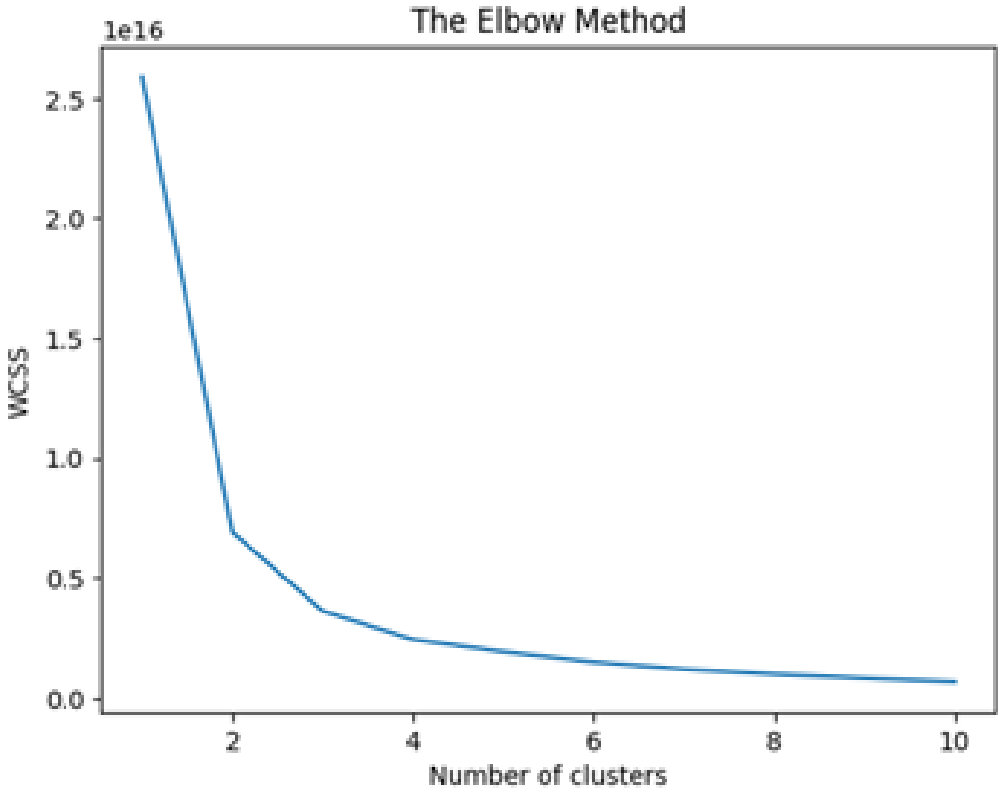

Figure 4.10: The Elbow Method

This point indicates the optimal number of clusters, where adding more clusters beyond this number results in diminishing returns in terms of clustering quality. By applying the Elbow Method, we were able to identify the most effective number of clusters systematically and accurately for our dataset, ensuring that the clustering process was both meaningful and efficient.

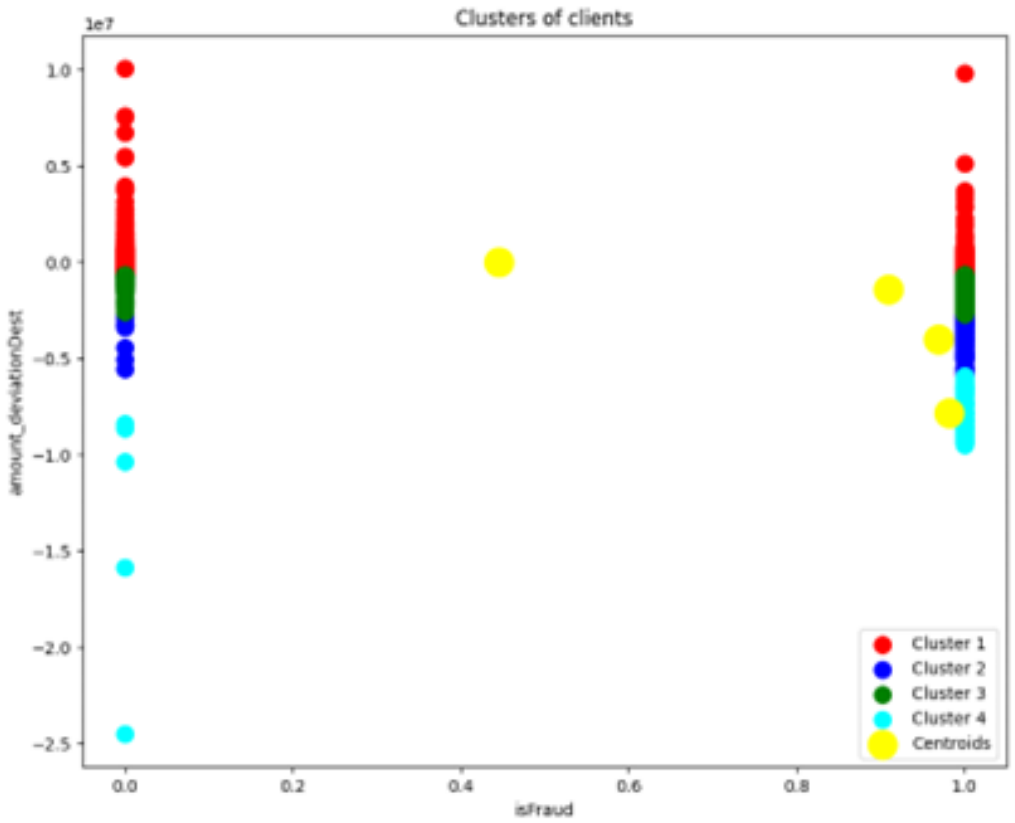

Figure 4.11: Clusters of Unseen Spending Pattern

From our graph, it is clear that we have identified four distinct clusters using K-means clustering. Among these, Cluster 1 stands out due to its larger number of instances compared to the other three clusters. Given its extensive data points, we selected Cluster 1 as the primary dataset for training our hybrid model. This approach ensures that the model is trained on a comprehensive and representative set of transaction behaviors.

For the evaluation phase, we combined the remaining three clusters (Clusters 2, 3, and 4) to form a diverse and varied test dataset. By using these combined clusters, we aim to assess the model's performance in detecting fraudulent transactions across different spending patterns.

The Hybrid model achieved an accuracy of 73.33% when evaluated on a dataset featuring an unseen pattern, specifically at a 60:40 ratio. Despite the inherent challenge posed by a slight class imbalance and the introduction of a novel spending pattern, the Hybrid model exhibited a creditable level of accuracy. This result underscores the model's resilience and adaptability to evolving fraud patterns, highlighting its potential efficacy in detecting rare instances of fraud among a vast majority of legitimate transactions.

Model	Dataset	Accuracy
MLP	Unseen Dataset	69.41%
Autoencoder + MLP	Unseen Dataset	73.33%

Table 4.10: Comparison of Hybrid Model vs MLP on Unseen Dataset with 60:40 class Distribution

In comparison to the MLP model, which achieved an accuracy of 69.41% on the same dataset and data ratio, the Hybrid model showcased a notably higher accuracy of 73.33%. This contrast highlights the effectiveness of the Hybrid model in adapting to the challenging class imbalance and novel spending patterns present in the dataset. While both models faced similar obstacles, the Hybrid model's incorporation of anomaly detection through Autoencoder likely contributed to its strong performance.

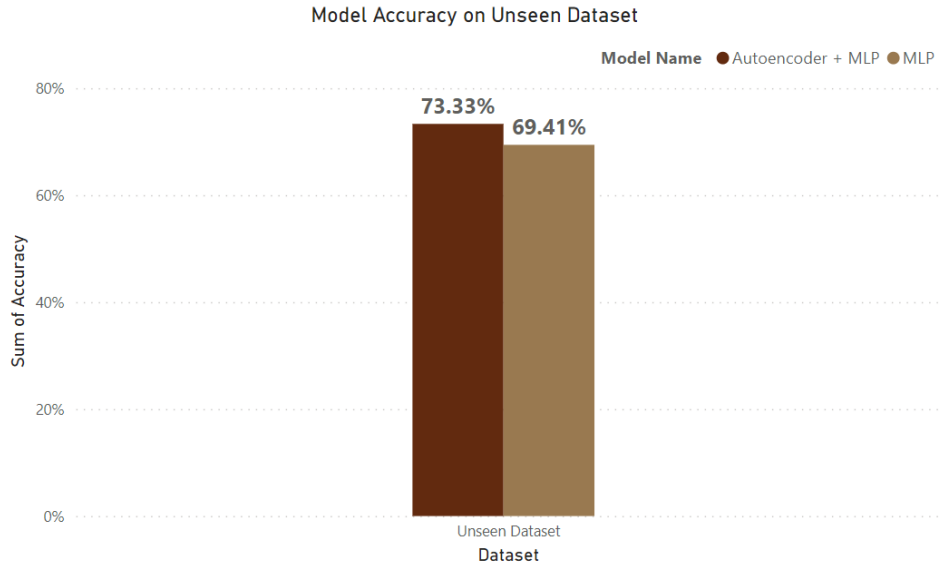

Figure 4.12: Comparison of Hybrid Model vs MLP on Unseen Dataset with 60:40 class Distribution

By leveraging the Autoencoder's ability to capture anomalies in the data, the Hybrid model demonstrated a better capacity to identify fraudulent transactions, resulting in a higher accuracy rate. the Hybrid model's slightly higher accuracy suggests that incorporating the Autoencoder for anomaly detection as a feature augmentation indeed provides additional value, enhancing the MLP's ability to distinguish between fraudulent and non-fraudulent transactions. This suggests that the Hybrid model's novel approach, combining deep learning with anomaly detection techniques, offers distinct advantages over traditional MLP models, particularly in scenarios with evolving fraud patterns.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Mobile money transactions are increasingly vulnerable to fraudulent activities, which can result in significant financial losses. To address this challenge, we have developed an anomaly-augmented deep learning approach for adaptive fraud detection. Our Hybrid model, which combines a Multi-Layer Perceptron (MLP) with an Autoencoder for anomaly detection, has demonstrated superior performance in identifying fraudulent transactions. Notably, the Hybrid model achieved an accuracy of 96.56%, precision of 97.62%, recall of 84.16%, and F1-score of 90.39%, outperforming the standalone MLP model. Additionally, the proposed Hybrid model has shown better adaptability to evolving fraud patterns, achieving an accuracy of 73.33%, compared to the MLP model's accuracy of 69.41%.The Hybrid model's ability to adapt to evolving fraud patterns and detect rare fraud instances makes it a promising solution for real-world fraud detection applications. Overall, this research contributes to the development of more effective fraud detection methods in mobile money transactions, which can help mitigate financial losses and improve the overall security of mobile payment systems.

5.2 Future Work

We propose that this method be examined on datasets other than the Pay Sim dataset to validate its effectiveness in different contexts and environments. This would help determine whether the Hybrid model's superior performance is consistent across various types of mobile money transaction data. Additionally, to further enhance the Autoencoder's capability, exploring different methods of incorporating error reconstruction as a feature could be beneficial. This might involve experimenting with various techniques to integrate the reconstruction error more effectively into the MLP, thereby improving the model's overall performance.

Another promising area for future research is to train the Autoencoder only on relevant subsets of the dataset. By focusing on specific portions of the data that are more pertinent to the Autoencoder's function, we might enhance its ability to detect anomalies and, consequently, improve the Hybrid model's fraud detection capability. These steps, aimed at refining both the feature engineering and training processes, could significantly bolster the robustness and accuracy of the Hybrid model in identifying fraudulent transactions.

References

[1] Morris Ayodele Peacock. An analysis of the potential risk and fraud involved in mobile money transaction in freetown sierra leone. *Global Journal of Computer Science and Technology*, 22(E1):31–35, 2022.

[2] Zlatko Bezovski. The future of the mobile payment as electronic payment system. *European Journal of Business and Management*, 8(8):127–132, 2016.

[3] Adeyinka Adedoyin. *Predicting fraud in mobile money transfer*. PhD thesis, University of Brighton, 2018.

[4] Isaac Akomea-Frimpong, Charles Andoh, Agnes Akomea-Frimpong, and Yvonne Dwomoh-Okudzeto. Control of fraud on mobile money services in ghana: an exploratory study. *Journal of Money Laundering Control*, 22(2):300–317, 2019.

[5] Mercy Wangari Buku and Rafe Mazer. Fraud in mobile financial services: protecting consumers, providers, and the system. Technical report, The World Bank, 2017.

[6] Aji Mubarek Mubalaike and Esref Adali. Deep learning approach for intelligent financial fraud detection system. In *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, pages 598–603. IEEE, 2018.

[7] Francis Effirim Botchey, Zhen Qin, Kwesi Hughes-Lartey, and Ernest Kwame Ampomah. Predicting fraud in mobile money transactions using machine learning: the effects of sampling techniques on the imbalanced dataset. *Informatica*, 45(7), 2022.

[8] Ratha Pech. Fraud detection in mobile money transfer as binary classification problem. *Eagle Tech. Inc Publ*, pages 1–15, 2019.

[9] Ms NishaBalani, Ms MeherBhawnani, and Ms AnkitaKamle. Implementation and design onfraud detection and prediction of mobile money transaction using ml techniques. *Annals of the Romanian Society for Cell Biology*, pages 261–269, 2020.

[10] Samir Kumar Bandyopadhyay and Shawni Dutta. Detection of fraud transactions using recurrent neural network during covid-19: fraud transaction during covid-19. *Journal of Advanced Research in Medical Science & Technology (ISSN: 2394-6539)*, 7(3):16–21, 2020.

[11] Isak Wirgen and Douglas Rube. Supervised fraud detection of mobile money transactions on different distributions of imbalanced data: A comparative study of the classification methods logistic regression, random forest, and support vector machine, 2021.

[12] Siti Sa'adah and Melati Suci Pratiwi. Classification of customer actions on digital money transactions on paysim mobile money simulator using probabilistic neural network (pnn) algorithm. In *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pages 677–681. IEEE, 2020.

[13] Dahee Choi and Kyungho Lee. Machine learning based approach to financial fraud detection process in mobile payment system. *IT CoNvergence PRActice (INPRA)*, 5(4):12–24, 2017.

[14] Petr Hajek, Mohammad Zoynul Abedin, and Uthayasankar Sivarajah. Fraud detection in mobile payment systems using an xgboost-based framework. *Information Systems Frontiers*, 25(5):1985–2003, 2023.

[15] Edgar Lopez-Rojas, Ahmad Elmir, and Stefan Axelsson. Paysim: A financial mobile money simulator for fraud detection. In *28th European Modeling and Simulation Symposium, EMSS, Larnaca*, pages 249–255. Dime University of Genoa, 2016.

[16] Rakesh Pandit Sheetal Bawane Jayesh Surana Pankaj Malik, Ankita Chourasia. Credit risk assessment and fraud detection in financial transactions using machine learning. *Journal of Electrical Systems*, 2024.

[17] Masoud Erfani, Farzaneh Shoeleh, and Ali A Ghorbani. Financial fraud detection using deep support vector data description. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 2274–2282. IEEE, 2020.

[18] Seyedeh Khadijeh Hashemi, Seyedeh Leili Mirtaheri, and Sergio Greco. Fraud detection in banking data by machine learning techniques. *IEEE Access*, 11:3034–3043, 2022.

[19] Kelsy Cabello-Solorzano, Isabela Ortigosa de Araujo, Marco Peña, Luís Correia, and Antonio J. Tallón-Ballesteros. The impact of data normalization on the accuracy of machine learning algorithms: A comparative analysis. In *International Conference on Soft Computing Models in Industrial and Environmental Applications*, pages 344–353. Springer, 2023.

[20] Huosong Xia, Wuyue An, and Zuopeng Justin Zhang. Credit risk models for financial fraud detection: A new outlier feature analysis method of xgboost with smote. *J. Database Manag.*, 34:1–20, 2023.

[21] Paria Soltanzadeh, M Reza Feizi-Derakhshi, and Mahdi Hashemzadeh. Addressing the class-imbalance and class-overlap problems by a metaheuristic-based undersampling approach. *Pattern Recognition*, 143:109721, 2023.

[22] Fadi Thabtah, Suhel Hammoud, Firuz Kamalov, and Amanda Gonsalves. Data imbalance in classification: Experimental evaluation. *Information Sciences*, 513:429–441, 2020.

[23] Alejandro Correa Bahnsen, Djamila Aouada, Aleksandar Stojanovic, and Björn Ottersten. Feature engineering strategies for credit card fraud detection. *Expert Systems with Applications*, 51:134–142, 2016.

[24] Xinwei Zhang, Yaoci Han, Wei Xu, and Qili Wang. Hoba: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture. *Information Sciences*, 557:302–316, 2021.

[25] Nabin Adhikari. Feature construction and feature splitting. *Medium*, 2021. Accessed: 2024-06-16.

[26] John T Hancock and Taghi M Khoshgoftaar. Survey on categorical data for neural networks. *Journal of big data*, 7(1):28, 2020.

[27] Soledad Galli. *Python feature engineering cookbook: over 70 recipes for creating, engineering, and transforming features to build machine learning models*. Packt Publishing Ltd, 2022.

[28] Cedric Seger. An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing, 2018.

[29] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM computing surveys (CSUR)*, 50(6):1–45, 2017.

[30] Alan Jović, Karla Brkić, and Nikola Bogunović. A review of feature selection methods with applications. In *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 1200–1205. Ieee, 2015.

[31] Cach N. Dang, María N. Moreno García, and Fernando de la Prieta. Hybrid deep learning models for sentiment analysis. *Complex.*, 2021:9986920:1–9986920:16, 2021.

[32] Talal S Qaid, Hussein Mazaar, Mohammad Yahya H Al-Shamri, Mohammed S Alqahtani, Abeer A Raweh, and Wafaa Alakwaa. Hybrid deep-learning and machine-learning models for predicting covid-19. *Computational Intelligence and Neuroscience*, 2021(1):9996737, 2021.

[33] Osman Musa Aydın and Ramazan Aktaş. Detecting financial information manipulation by using supervised machine learning technics: Svm, pnn, knn, dt. *Uluslararası İktisadi ve İdari İncelemeler Dergisi*, (29):165–174, 2020.

[34] Apapan Pumsirirat and Yan Liu. Credit card fraud detection using deep learning based on auto-encoder and restricted boltzmann machine. *International Journal of advanced computer science and applications*, 9(1), 2018.

[35] Bassam Kasasbeh, Balqees Aldabaybah, and Hadeel Ahmad. Multilayer perceptron artificial neural networks-based model for credit card fraud detection. *Indonesian Journal of Electrical Engineering and Computer Science*, 26(1):362–373, 2022.

[36] Aji Mubalaike Mubarek and Eşref Adalı. Multilayer perceptron neural network technique for fraud detection. In *2017 International Conference on Computer Science and Engineering (UBMK)*, pages 383–387. IEEE, 2017.

[37] Hyder John and Sameena Naaz. Credit card fraud detection using local outlier factor and isolation forest. *Int. J. Comput. Sci. Eng*, 7(4):1060–1064, 2019.

[38] Soumaya Ounacer, Hicham Ait El Bour, Younes Oubrahim, Mohamed Yassine Ghoumari, and Mohamed Azzouazi. Using isolation forest in anomaly detection: the case of credit card transactions. *Periodicals of Engineering and Natural Sciences*, 6(2):394–400, 2018.

[39] Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang. Deep isolation forest for anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[40] V Vijayakumar, Nallam Sri Divya, P Sarojini, and K Sonika. Isolation forest and local outlier factor for credit card fraud detection system. *International Journal of Engineering and Advanced Technology (IJEAT)*, 9:261–265, 2020.

[41] Mohamad Zamini and Gholamali Montazer. Credit card fraud detection using autoencoder based clustering. In *2018 9th International Symposium on Telecommunications (IST)*, pages 486–491. IEEE, 2018.

[42] MA Al-Shabi. Credit card fraud detection using autoencoder model in unbalanced datasets. *Journal of Advances in Mathematics and Computer Science*, 33(5):1–16, 2019.

[43] Shi Na, Liu Xumin, and Guan Yong. Research on k-means clustering algorithm: An improved k-means clustering algorithm. In *2010 Third International Symposium on intelligent information technology and security informatics*, pages 63–67. Ieee, 2010.

[44] Huda Hamdan Ali and Lubna Emad Kadhum. K-means clustering algorithm applications in data mining and pattern recognition. *International Journal of Science and Research (IJSR)*, 6(8):1577–1584, 2017.

[45] Emmanuel Ileberi, Yanxia Sun, and Zenghui Wang. Performance evaluation of machine learning methods for credit card fraud detection using smote and adaboost. *IEEE Access*, 9:165286–165294, 2021.

[46] Robert C Gardner and Richard WJ Neufeld. What the correlation coefficient really tells us about the individual. *Canadian Journal of Behavioural Science/Revue canadienne des sciences du comportement*, 45(4):313, 2013.