Suspicious Activity Reports: Enhancing the Detection of Terrorist Financing and Suspicious Transactions in Migrant Remittances

MASTER OF SCIENCE

in

MATHEMATICAL STATISTICS

in the

DEPARTMENT OF STATISTICS

of

RHODES UNIVERSITY

by

Stanley Munamato Mbiva
(ORCID iD: 0000-0002-0852-2343)

January 2024

Supervisor: Dr Fabio Correa

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Rhodes University will not infringe any third-party rights and that I have not previously in its entirety or part submitted for obtaining any qualification.

Date: 22nd July 2024

Abstract

Migrant remittances have become an important factor in poverty alleviation and microeconomic development in low-income nations. Global migrant remittances are expected to exceed US $630 billion by 2023, according to the World Bank. In addition to offering an alternate source of income that supplements the recipient's household earnings, they are less likely to be affected by global economic downturns, ensuring stability and a consistent stream of revenue. However, the ease of global migrant remittance financial transfers has attracted the risk of being abused by terrorist organizations to quickly move and conceal operating cash, hence facilitating terrorist financing. This study aims to develop an unsupervised machine-learning model capable of detecting suspicious financial transactions associated with terrorist financing in migrant remittances. The data used in this study came from a World Bank survey of migrant remitters in Belgium. To understand the natural structures and grouping in the dataset, agglomerative hierarchical clustering and k-prototype clustering techniques were employed. This established the number of clusters present in the dataset making it possible to compare individual migrant remittances in the dataset with their peers. A Structural Equation Model (SEM) and an Local Outlier Factor - Isolation Forest (LOF-IF) algorithm were applied to analyze and detect suspicious transactions in the dataset. A traditional Rule-Based Method (RBM) was also created as a benchmark algorithm that evaluates model performance. The results show that the SEM model classifies a significantly high number of transactions as suspicious, making it prone to detecting false positives. Finally, the study applied the proposed ensemble outlier detection model to detect suspicious transactions in the same data set. The proposed ensemble model utilized an Isolation Forest (IF) for pruning and a Local Outlier Factor (LOF) to detect local outliers. The model performed exceptionally well, being able to detect over 90% of suspicious transactions in the testing data set during model cross-validation.

Keywords: **Migrant Remittance, Terrorist Financing, Structural Equations, Outlier Detection, unsupervised learning**

Contents

List of Figures

List of Tables

Acronyms

AIC Akaike Information Criterion

AML/CTF Anti-Money Laundering / Counter Terrorism Financing

ANN Artificial Neural Networks

BBN Bayesian Belief Network

BIC Bayesian Information Criterion

BN Bayesian Networks

CFI Comparative Fit Index

FAFT Financial Action Task Force

FIC Financial Intelligence Center (South Africa)

FIU Financial Intelligence Unit

IF Isolation Forest

IST Information Systems and Technology

kNN k-Nearest Neighbour

LEA Law Enforcement Agents

LOF Local Outlier Factor

LOF-IF Local Outlier Factor - Isolation Forest

NPL Natural Processing Language

OECD Organisation for Economic Co-operation and Development

RMR Root Mean Square Residual

RMSEA Root Mean Square Error of Approximation

SAR Suspicious Activity Report

SEM Structural Equation Model

SRMR Standardized Root Mean Square Residual

STR Suspicious Transaction Reports

SVM Support Vector Machines

TF Terrorist Financing

TFAR Terrorism Financing Activity Report

TRTR Terrorism Financing Transaction Report

UNDP United Nations Development Programme

WCSS Within-Cluster- Sum of Squared Errors

Acknowledgements

First and foremost, I want to sincerely thank my parents for all of their support and advice throughout the years. Special thanks goes out to my uncle, Mr. Jabulani Mbiba, for his inspiration, dedication, and everlasting support. My sincere thanks also goes out to my supervisor, Dr. Fabio Correa, and Mr. Jeremy Baxter, who made this project a reality and provided support throughout the most difficult times. Lastly, the inspiration behind it all, my late grandfather Dr. Jeremiah Chese, is honoured in this work. Your unwavering dedication to knowledge and enlightenment has inspired me and will serve as a source of inspiration for generations to come.

May your soul rest in divine peace

Chapter 1

Introduction

1.1 Chapter Introduction

The complex ways in which terrorist organizations source, move, distribute, and store funds and the constant attempts to detect and suppress these channels have been a focal point of academic research. Over the past years, terrorism has morphed into a global threat, especially since the brazen September 11th attacks and the subsequent declaration of the *War on Terror* in the early 2000s. The threat of terrorism continues to grow despite the implementation of counter-terrorism measures. Keatinge & Keen (2020) acknowledges this fact, stating that global terrorism has become a network of multiple terror groups with diverse financing techniques. The introduction of new technologies, payment systems, and globalization are some of the environmental factors that have influenced global terrorism and terrorism financing (Keatinge & Keen 2020). This growth rate has been partly fuelled by donations from well-wishers who sympathize with the terrorist's ideological goals. The Financial Action Task Force (FAFT), an international organization mandated to combat money laundering and terrorist financing, states that donations from private individuals account for 33 % of the terrorist financing cases investigated by law enforcement in the United States (FAFT 2015). In addition, the use of online crowdfunding and social media platforms by terrorist groups has also facilitated terrorist financing (FAFT 2023). These sources of terrorist financing make it difficult to identify malicious financial transfers from legitimate transactions (Teichmann 2019).

Among the various counter-terrorism methods employed, targeting and detecting the movement of financial resources meant to aid terror operations and logistics has proven to be an effective, low-cost solution in reducing the operational capabilities of terrorist organizations (Keatinge & Keen 2020). Nonetheless, most terrorist groups have learned to swiftly adapt, employing unconventional, rudimentary methods to circumvent these preventative measures. These methods may include criminal activities, state sponsorship, controlling and selling local resources, and exploiting the financial system (Shokry et al. 2020).

This study seeks to develop an unsupervised machine learning classifier that identifies anomalous transactions that indicate the presence of terrorist financing. A Structural Equation Model (SEM) was developed to formulate the mathematical relationship between the latent factors that contribute to terrorism and their predictor measurements. The study also deploys an ensemble outlier detection model to achieve the same goals. The proposed ensemble model combines two outlier detection models namely, the Isolation Forest (IF) and the Local Outlier Factor (LOF) algorithm.

1.2 Motivation

On the issue of terrorist financing, South Africa is exposed to the dangers of terrorist financing including, financing facilitation networks and cells. The FAFT (2021) report describes the understanding of the terrorism financing risks by the South African authorities as crude and underdeveloped at best, noting the conservative approach of misclassifying terrorism as political acts of violence which in turn hampers investigations of potential terrorist financiers (FAFT 2021). The report also found that South African Law Enforcement Agents (LEA) had adequate operational financial intelligence and support from the Financial Intelligence Center (South Africa) (FIC) but, lacked the skills and resources to investigate these financial crimes. Most importantly, FAFT (2021) described the inability of the agencies to effectively identify, investigate, and prosecute terrorist financiers, lamenting the low number of cases prosecuted for terrorist financing.The report emphasized the importance of understanding Terrorist Financing (TF) risks associated with foreign and domestic terrorists in South Africa, noting the country's status as a regional economic powerhouse may attract terrorists to use the country as a financial and logistic hub. The growing threat of terrorism in Africa, especially with the 2017 terrorist insurgency in Cabo Delgado province in Mozambique, and the current instability in much of West Africa, Somalia, and the Democratic Republic of Congo, amplifies the importance of effectively detecting TF in the financial system. This was revealed in a Sunday Times report that detailed the transfer and distribution of R6 billion from different spaza outlets to regional terrorist groups in Kenya, Somalia, Nigeria, and Bangladesh through payment methods designed for remittances

and using approximately 57 000 unregistered phones between 2020 and 2021 (Graeme et al. 2022). Furthermore, inadequate Anti-Money Laundering / Counter Terrorism Financing (AML/CTF) policies have negative ramifications for the South African economy. In February 2023, South Africa alongside Nigeria, Senegal, and Mozambique was added to the FAFT's grey-list of countries that need to be closely monitored by the organization due to their AML/CFT compliance deficiencies (Kempen 2023).

The economic consequences of being grey-listed are severe as it increases the nation's money laundering and terror financing risk profile, casting doubt on the abilities of the state financial regulatory bodies. As a result, this leads to higher premiums, closure of business, and a tainted relationship with other global financial counterparts (Rangongo 2022). Considering these challenges, there is an urgent need for research to understand the complex nature of financing terrorism, and assist banks and other financial institutions improve their compliance requirements. Restricting the financial resources available to terrorist groups limits the impact of future terrorist attacks. This creates an effective deterrence against terrorist financiers and provides a better understanding of the terrorist logistical network (Biersteker et al. 2008).

1.3 Research Objectives

This thesis aims to implement an unsupervised machine-learning algorithm that is capable of effectively identifying terrorist financing in migrant remittances. Among the models considered such as the gaussian mixture model and Naive Bayes models, the structural equation model (SEM) and an ensemble outlier detection model were selected. The SEM was selected based on its ability to formulate mathematical relationships with unobserved latent factors with observed predictor measurement variables. Since the dataset is unsupervised, a preliminary analysis is performed utilising an agglomerative hierarchical clustering and a k-prototype clustering technique to find structures and clusters inherent in the dataset. Agglomerative hierarchical clustering technique partitions the data without the k-cluster initialization. The k-prototype clustering is suitable for clustering mixed datasets. Finally, the study develops an anomaly detection algorithm that identifies anomalous financial transactions. The principle concept is that suspicious transactions are seen as outliers or anomalous activities that do not conform with the standard variables (Sudjianto et al. 2010). Therefore, an Isolation Forest and the Local Outlier Factor were combined the develop the ensemble outlier detection model for detecting suspicious transactions. In addition, a traditional Rule-Based Model (RBM) was also develop as a benchmark model for evaluating model performance metrics of both unsupervised algorithms.

1.4 Assumptions

It is worth mentioning the assumptions that form the basis for this study which are in line with Biersteker et al. (2008). The first assumption states that all terrorist groups or organizations are similar and, hence treated equally. Although the methods and motives may differ, the assumption is the ultimate goal for most terrorist organizations is to obtain a political objective through intimidation and fear. The second assumption states that the same counter-terrorist measures and techniques can be applied to different terrorist organizations. Despite acknowledging the vulnerabilities of TF risks in informal financial sectors, the third assumption states that the formal financial sector provides the main channel for transferring funds for financing terrorism. Lastly, it is assumed that the regulations that govern formal financial institutions can be applied to regulate the informal sector as well.

1.5 Thesis Outline

The remainder of the thesis is arranged as follows. Chapter 2 explores the academic literature on the subject by previous academic scholars, emphasizing the statistical techniques and methods employed to effectively detect terrorist financing. This chapter also provides a brief review of the nature, trends, and patterns of global financing of terrorism, an introduction to supervised and unsupervised machine learning techniques employed to combat financial crimes, a discussion on the challenges encountered when devising AML/CFT techniques, and the successes and shortfalls of various methodologies. Chapter 3 shows a brief preliminary investigation done before the methodology. It's important to note that this is a self-contained, standalone chapter that is meant to investigate the structure of the unlabeled data as it is usually the first step in understanding unsupervised data. Thus, a k-prototype clustering algorithm is implemented in 3.8, and the results are noted. Clustering also identifies peer groups of similar behavior hence, enabling the comparison of clients' actions to their peers rather than as individual entities. Chapter 4 explores and examines the methods developed to accomplish the study objectives. After a brief explanation of the Belgium remittance data set, the structured equation modeling process and its application to exploring the theories of terrorism, with the intent of modeling the risk of terrorism are introduced. The chapter also presents a detailed formulation of the outlier detection algorithms; the Isolation Forest and the Local Outliers Factor machine learning algorithm and the model validation techniques. Chapter 5 presents the results obtained from the application of the methodology described in the previous chapters. This chapter also discusses the model validation results obtained from cross-validation. Chapter 6 summarises the results and presents conclusions and recommendations for future research areas.

Chapter 2

Literature Review

2.1 Chapter Introduction

The academic discourse surrounding the detection of financial crimes has been broad and diversified, with most statistical detection methods in the financial sector focusing on detecting money laundering and credit card fraud. (Sudjianto et al. 2010). However, financial crimes, such as sponsorship of terrorism, are relatively new and elusive (Keatinge & Keen 2020). The central purpose of this chapter is to present the scholarly research and methodological challenges related to the identification of terrorist financing in the financial migrant remittance system.

2.2 Economic Impact of Migrant Remittances

Until recently, migrant remittances played an important role in poverty alleviation and microeconomic development in developing nations (Rapoport & Docquier 2006). Yang (2011) defines migrant remittances as household income sent by migrants from their host country to their families in the home country. The World Bank characterizes migrant remittances as low-value, cross-border, non-commercial, peer-to-peer financial transactions (World Bank 2021). Migrant remittances have become an essential part of the modern financial system thus, it is important to have an in-depth analysis of the economic impact and potential risks associated with migrant remittances. The role and impact of migrant remittances as an alternative form of international funds to developing countries has been recently acknowledged, with much emphasis placed on the scale, growth rate, and stability of migrant remittances (Ratha 2013).

Yang (2011) notes that an estimated US $325 billion in migrant remittances was transferred globally in 2009. The study also claims the annual real growth rate of migrant remittances stood at 13 %, far outstripping the 11 % of foreign direct investments and 5.8 % of other governmental donation rates. Dilip et al. (2023) peg global migrant remittance inflows for low to middle-income countries at US $647 billion in 2022 and despite the slow economic growth, high inflation, and the wars in Ukraine and Sudan, migrant remittance inflows are expected to increase by 1.4% in 2023. Ratha (2003) posits that migrant remittances are a stable source of external income that supplements the individual recipient's household income since they are less responsive to cyclic economic changes than private capital. Yang (2011) concurs with this assertion, stating that in the aftermath of the 2008/09 financial crisis, migrant remittances only contracted by 5.2 % compared to the 39.7 % drop in foreign direct investment over the same period. Economic downturns tend to motivate migrants to transfer more funds, therefore the flow of migrant remittances to developing nations increases in reaction to economic negative trends, as migrants strive to alleviate the economic troubles faced in the migrant country of origin (Yang 2011).

Although Rapoport & Docquier (2006) and Shen et al. (2010) have drawn connections between migrant remittances and income inequality, the macroeconomic benefits of migrant remittances remain significant. For instance, Adams Jr & Cuecuecha (2010) study on the effects of remittances on household spending behavior in Guatemala, concludes that households that receive remittances spend more on human capital like education and housing compared to other consumables such as food and clothing. The study claims that remittance earnings are viewed as a short-term source of income that should be invested rather than consumed. Such investments create new business and job opportunities for construction workers and merchants while also contributing to human capital development. According to Lucas (1987), remittances from immigrants working in South African mines increased crop productivity and wealth accumulation in the form livestock. However, Biyase (2012) states that internal remittances have minimal impact on poverty reduction in South Africa. Despite the decline in agricultural activities and labour shortages caused by migration, the increase in crop productivity could be attributed to investing in improved mechanisation or provision of insurance against unforeseen risks, encouraging experimentation and risk-taking by subsistence farmers (Lucas 1987). Posel & Casale (2006) notes that the real-term value of remittances to rural households has significantly dropped due to the shift in investment choices. As more financial options become accessible, migrant workers may prefer to invest in pension funds and other saving instruments as compared to livestock (Posel & Casale 2006). Akram et al. (2017) also concurs, stating that seasonal migration indirectly improves the standard of living in rural areas. Akram et al. (2017) further states that subsidizing transport costs for Bangladesh seasonal workers increased male farm wages by 4.5%.

2.3 Financing of Terrorism

Mukhtar (2018) defines terrorist financing as funds acquired for the commission of future terrorist activities. This includes the provision of financial support for terrorist organizations from legal or illegal sources such as charity institutions, individual donations, and revenue from businesses. Zubair et al. (2015) further elaborates terrorist financing as funds or property unveiled for the facilitation of terrorist organizations and the proceeds generated from such activities. By incorporating the four stages of terrorist financing, Romaniuk (2014), defines terrorism financing as the ability to raise, move, store, and distribute funds and other resources to aid terrorist operations. Most studies and academic literature on financial crimes have emphasized preventative measures against money laundering and credit card fraud, which are prevalent and pose an enormous threat to the integrity of financial institutions (Bolton & Hand 2001, Sudjianto et al. 2010).

Despite the growing interest in the subject, Biersteker et al. (2008) expresses dissatisfaction with the available literature on terrorism financing and prevention efforts, noting that it is based on the generalization of a few case studies without considering the important changes in the financial regulation sectors. Freeman & Ruehsen (2013) also posits that many scholars have mainly focused on the sources of terrorist funds or the items acquired, overlooking the methods terrorists move or transfer funds and the methods this can be disrupted. The movement of money represents a crucial intermediary step and a point of weakness that states and law enforcement agents can target and disrupt terrorist operations more effectively (Freeman & Ruehsen 2013). According to the Organisation for Economic Co-operation and Development (OECD), terrorism financing has four stages namely funds collection, storage, funds transfer, and utilization (OECD 2019). As illustrated in Figure 2.1, the first stage involves soliciting funds from either legal or illegal channels.

Figure 2.1: The stages of terrorist financing (adapted from: (OECD 2019))

After obtaining sufficient finances for the required mission, the terrorists store the funds while developing a plan of action to achieve their goals. This leads to the third stage, which facilitates the distribution of funds to the cells of interest. Finally, the utilization stage leads to the use of the funds in logistics and purchasing equipment related to carrying out the desired operations (OECD 2019). It can be deduced that money transfer systems play a significant role in the first stage and the third stage. Keatinge & Keen (2020) claims the terrorist organization Al-Qaida used at least US$400 000 to finance the 9/11 attacks, with the bulk of the funds being transferred undetected through the financial system. Biersteker et al. (2008) emphasizes the essence of detecting and disrupting the flow of terrorist funds by noting that Ramzi Yousef, the organizer of the 1993 World Trade Centre attack, confessed that he would have made a bigger bomb if the necessary funds were available.

It is also important to note that most studies have lacked a true assessment of the state and nature of money laundering and terrorism financing due to the absence of credible statistical data (Sudjianto et al. 2010). Al-Suwaidi & Nobanee (2020) laments the quality and quantity of proper studies on this subject, emphasizing the lack of proper statistical interpretation and empirical analysis. The lack of raw statistical data has hampered the effort to carry out proper statistical studies. In addition, Al-Suwaidi & Nobanee (2020) acknowledges the lack of recommendations and insistence on providing appropriate detection methods and conducting further research. Only two of the thirty-two reviewed articles highlight the importance of enhancing the detection of money laundering and terrorism financing. The paper recommends future research should focus on machine learning detection schemes, the use of artificial intelligence systems to identify suspicious accounts with potential money laundering and terrorist financing activities, and examining alternative remittance systems such as the Hawala remittances system, their risks, and effective control.

2.4 Migrant Remittances and Financing of Terrorism

A major concern with migrant remittance transfers is the risk of terrorist financing and money laundering. The swift movement of financial resources is important for the seamless flow of operations for any terrorist organization (Keatinge & Keen 2020). The World Bank (2021) acknowledges this potential risk, stating that if the risks are ineffectively mitigated, criminals and terrorist organizations may abuse these financial channels to their advantage. Terrorist organizations have multiple legal and illegal ways of raising funds for their operations which may range from extortion to donations. These organizations are constrained to a handful of options when it comes to moving funds to facilitate their operations (Freeman 2011).

One of the systems exploited by terrorists in moving funds is the Hawala money transfer system (FAFT 2013). The Hawala system is usually described as a network of illegal or unlicensed money transmitters outside the traditional formal banking or financial channels (Jost & Sandhu 2000). This system is usually the preferred choice by terrorist and criminal organizations for its cost-effectiveness, efficiency, reliability, and extensive reliance on family and religious affiliation. It is the lack of financial scrutiny and bureaucracy found in financial institutions that attract most terrorist financiers to the Hawala remittance system. In the absence of financial transaction data, little can be done in data analysis and the extent to which Hawala is used or exploited by criminal organizations (Jost & Sandhu 2000). The FAFT (2013) report on Hawala systems demonstrates that the popularity of the Hawala system reflects the inefficiencies of banks and other financial institutions in providing financial services to the unbanked and under-banked sections of the community. The FAFT (2013) report discusses registration and licensing of Hawala agents as an alternative that would lessen the risks associated with informal Hawala networks. This also allows for efficient collection and analysis of financial transaction data from Hawala agents (FAFT 2013).

2.5 Suspicious Activity Reporting

As highlighted in the previous sections, migrant remittances are likely to be at risk of being misused as terrorist financing channels. Banks and other regulated financial institutions in South Africa must report suspicious activity as swiftly as possible (FIC 2020). Whenever a client has performed a transaction deemed to be malicious, financial institutions must compile a Suspicious Activity Report (SAR) or a Suspicious Transaction Reports (STR) that is sent to Financial Intelligence Unit (FIU) for analysis and interpretation. The FIU also requires a Terrorism Financing Activity Report (TFAR) when the suspicious act is linked to the financing of terrorism (FIC 2019).

According to the Financial Intelligence Centre Act of 2001, suspicious activity comprises action deemed to contravene Section 29 of the Act itself. This does not require the transaction itself to be completed. For instance, a transaction or series of transactions terminated upon the request of the client's identification may be deemed suspicious (FIC 2019). On the other hand, suspicious transactions are completed transactions that have been flagged in the system as suspicious. Any suspicious transaction related to terrorist financing would require a Terrorism Financing Transaction Report (TRTR).

Figure 2.2 summarises the different reports mandated by Section 29 of the Financial Intelligence Act (FIC 2019).

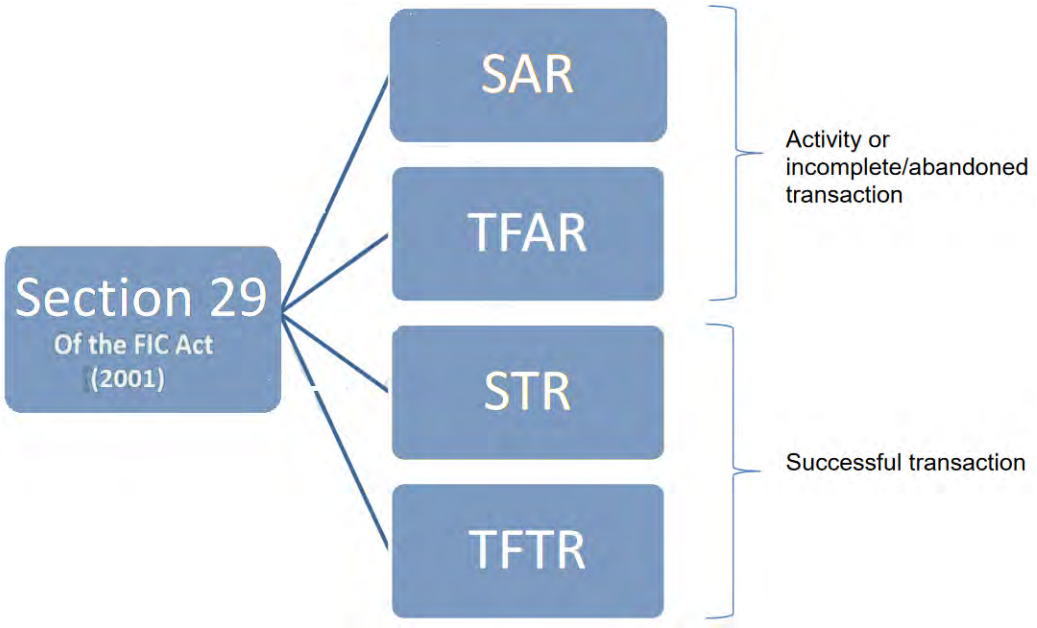

Figure 2.2: Different suspicious reports embedded in the FIC Act of 2001 (adapted from: (FIC 2019))

Despite significantly reducing the number of malicious transactions and apprehending culprits, such reporting has noticeable shortcomings. Firstly, the large number of SAR/STR reports filed by banks and financial institutions inundates FIC with thousands of reports that need to be sorted and thoroughly analyzed in a short space of time. According to the FIC-NAMIBIA (2023), the financial intelligence unit in Namibia received 1150 suspicious transaction reports (STR), and 203 suspicious activity reports (SAR) in 2023. In 2023, South African FIC organisation received 5.3 million reports from 45 392 financial institutions, 558 348 of these reports were suspicious transaction reports (STR) (FIC 2023). The high volume of SAR/STR reports could also be attributed to defensive filing. Defensive filing refers to the practice of filing SARs or STRs on activities or transactions that are not legitimately suspicious in order to limit the risk of regulatory penalties for failing to file such documents (Egmont 2011).

Alternatively, financial institutions resorted to de-risking strategies to mitigate the costs of compliance especially in regions with high risks of terrorist financing (Durner & Shetret 2015). De-risking refers to downsizing financial bank operations in areas deemed a "high bank risk". This reduces the cost of compliance incurred by the bank however, it negatively impacts the communities that rely on the bank, affecting the ability of individuals to manage savings and build capital. On a macroeconomic scale, de-risking affects the financing of small businesses and micro-enterprises (Durner & Shetret 2015). However, Elnahass et al. (2022) posits that al-

though banks operating in high-risk zones of terrorist attacks have significantly high credit risk and high insolvency rates, they exhibit very high financial performance and profitability. The high risk of terrorist activities is often correlated with high bank performance, which translates to high profitability and better cost efficiency (Elnahass et al. 2022). Banks in such regions receive favorable incentives such as interest rate cuts and economic stimulation packages offered by the government for providing financial services in high-risk, marginalized areas. Nonetheless, banks located in high-income earning economies are adversely affected by the increased risk of terrorism, thus, low financial performance and high credit risk are associated with exposure to terrorism (Elnahass et al. 2022).

Another conceivable problem is the constant reliance on a rule-based approach in identifying suspicious transactions. For a transaction to be classified as suspicious, certain rules and guidelines set by a regulatory body are employed. This proves to be ineffective, rigid, and slow to respond to the ever-dynamic financial environment (Jose-de Jesus et al. 2021). Furthermore, these archaic methods are incompatible with electronic money transfers where time is a crucial factor (Jose-de Jesus et al. 2021). The fixed rule approach generates a lot of false positives attributed to the inability of the system to adjust according to behavioral changes in criminal activities. There is a greater reputational risk of raising a false positive, financial institutions stand a chance of ruining their reputation if such erroneous classifications are made public.

2.6 Statistical Modeling of the Risk of Terrorism

The detrimental effects of terrorist activities have led to numerous attempts by researchers to develop mathematical models that adequately represent the phenomena. For instance, Boyd (2016) conducted a study that utilized a multilevel algorithm to compare the rate of domestic attacks and attacks against foreign targets. By equating the risk of terrorist attacks with catastrophic risks such as earthquakes, Major (2002) and Ezell et al. (2010) proposed the use of mathematical risk-based modeling of terrorist events. Both acknowledged that modeling terrorist activity, as opposed to catastrophic events or technical machine failures, is problematic due to the adaptability of terrorists. The need to understand the risks of terrorism and its impact on society has also aroused great interest in the field of social sciences. Borum (2004) explores the psychological aspects of terrorism and seeks to understand the psychological dimensions and behavioral dynamics of terrorists. Crenshaw (2000) sees terrorism as the product of a common ideology and group solidarity rather than the result of individual personal factors.

However, the search for a unifying theory of terrorism is fraught with challenges. Laqueur (2003) agrees that the idea of finding a general theory of terrorism is a misguided and futile endeavor. Nonetheless, theorizing about the concepts of terrorism provides a starting point for understanding terrorism. These theory-derived models combined with dynamic simulation, are capable of estimating the strength of the correlation between measurable attitudes, intentions, and manifest behaviors. Moreover, statistical models provide a decision-support tool to estimate the effects of interventions taken and counter ideologies that support terrorism (Borum 2004).

Nivette et al. (2017) used an ordinary least squares regression to model violent extremist attitudes. The study tests the application of the general strain theory among an ethnically diverse group of Swiss adolescents. Ekici & Akdogan (2020) performed a study to investigate the development of terrorist perceptions in Turkish high school students. After conducting a confirmatory factor analysis, religion, risk perception, and external influences were determined as major contributing factors. Risk perception measured the participant's notion of being under the potential risk of terrorism. The study also identifies religion as a key factor in shaping terrorism perceptions. Barton & Barton (2006) suggested using modern psychometric modeling and statistical analysis to comprehend society's vulnerabilities to terrorism. The paper introduced mathematical modeling based on Ajzen (1991)'s theory of planned behavior. The theory states that an individual's intentions to perform certain behaviors can be determined from their attitudes toward the behavior, societal norms or expectations, and perceived behavioral control (Ajzen 1991). An individual must believe that the intended behavior will result in the attainment of the desired outcome, be in line with social expectations, and possess the personal skills to perform the behavior (Barton & Barton 2006).

2.6.1 The General Strain Theory

Another psychological theory linked to explaining terrorism risks is Agnew (2010)'s general strain theory. The theory states that the conditions for terrorism are most likely to occur when people experience "collective strains or grievances" that are severe, unjust, and inflicted by a significantly powerful group of people (Agnew 2010). Strains are events or conditions that cause discomfort. These include negative treatment, loss of valuable assets, and the inability to attain set goals. Agnew (2010) identifies a list of collective strains that may lead to terrorism including material deprivation, territorial, ethnic, and religious disputes, military occupation, and political and socio-economic discrimination. Collective strains lead to strong emotional responses such as anger or hopelessness which reduces awareness and lack of concern for the consequences of an individual's behavior thus, involve frequent violent outbreaks which provide a modeling base for terrorism (Agnew 2010).

These assumptions are evident in the United Nations Development Programme (UNDP) survey report on identifying the influences that cause young African youths to join violent extremist groups. The report identifies religious education as a major determinant in one's decision to join a violent extremist organization. Fostering a greater knowledge of religion through methods that encourage engagement and critical thinking reduces the risk of misinterpretation of religious ideas and the chances of participating in a terrorist organization (UNDP 2017, 2023). Access to education plays an important role in predicting an individual's likelihood of being recruited into a terrorist organization. Therefore, being deprived of proper education may be viewed as a constraint. The reports further identify economic and geographical factors that influence terrorism beliefs in young adults and encourage them to join terrorist organizations.

Economic factors have been noted as the main contributors to determining the risk of terrorist recruitment. The relevance of economic factors is significant stating concerns associated with an upbringing in a severely poverty-stricken environment with the realities of unemployment provided a source of frustration and a justification to join violent extremist groups UNDP (2017). The report noted that employment was cited as the most frequent need at the time they joined a terrorist organization with 80% of the respondents joining an extremist organization in less than a year after their first contact. Thus, the lack of economic inclusion in society can be a metric to measure the risk of terrorist activity in that community UNDP (2017).

2.7 Machine Learning Techniques For the Detection of Financial Crimes

It is important to highlight the statistical and computational challenges in detecting financial crime. Sudjianto et al. (2010) summarises these challenges in four points. First, the volume and complexity of financial data pose enormous challenges for researchers. Large financial institutions process a large number of customer transactions per second, creating a large database of information collected across multiple, disparate systems. This puts an enormous strain on the algorithm's computing capacity, especially when important decisions need to be made promptly. Furthermore, such a large, heterogeneous dataset makes it more difficult to detect suspicious financial activity due to the phenomenon known as the curse of dimensionality, that is, the relative contrast between the closest data point and the furthest data point in a data set diminishes as dimensionality increases (Zimek et al. 2012, Hilal et al. 2022).

Secondly, the number of actual criminal cases or suspicious transactions is extremely low due to the infrequent occurrence of such crimes, resulting in an imbalanced data set. Kulatilleke (2022) posits that the frequency of credit card fraud is limited to 0.1% of all card transactions. However, the high amounts involved can lead to enormous financial losses. This class imbalance poses a fundamental challenge to the performance of the classifier, especially when considering the costs associated with misclassification (Chawla et al. 2002). The probability of a false negative result is high, hence, identifying suspicious transactions becomes a complicated task due to a large number of unsuspicious events. Zhu et al. (2017) also agrees and states that most classification algorithms have a negative impact as they tend to favor the majority class, resulting in high overall accuracy but unacceptably low precision with respect to the minority class interests. This can be costly for the financial institution and carries the risk of tarnishing the organization's reputation (Chawla et al. 2002).

Thirdly, the constant development of methods to circumvent existing financial recognition is a challenge. Risk models need to be constantly reassessed and updated to reflect these changing distributions and patterns and to maintain their robustness against money laundering and terrorist financing. Shokry et al. (2020) points out that rule-based systems alone are rigid and insufficient to monitor complex, ever-changing financial activities. Shokry et al. (2020) also recognizes that advances in artificial intelligence and machine learning are reducing the inefficiencies associated with rules-based risk models.

Finally, overlapping and mislabelling classes obstruct the effective detection of suspicious financial transactions. Class overlap refers to observations in a dataset that have similar features but belong to different classes (Qu et al. 2020). Overlapping classes in financial data arises when criminals try to conceal their activities by making illegal transactions appear as normal as possible, which leads to a blurry distinction between fraudulent and non-fraudulent classes (Sudjianto et al. 2010). The complications with overlapping classes become even more complicated as the dimensionality of the dataset increases and the distinction between classes becomes even more complex, affecting the performance of the machine learning algorithm (Basit et al. 2022). This causes masking and swamping effects in the data set. Swamping occurs when normal instances are close to anomalies and masking is defined as too many anomalous points that cluster together (Liu et al. 2008).

Jose-de Jesus et al. (2021) notes the deficiency in using artificial intelligence and machine learning algorithms to detect the movements of money in financial institutions predestined for financing terrorism. There is little knowledge in the existing literature about the use of machine learning models to monitor and detect the financing of terrorism in financial institutions. Samantha Maitland et al. (2011) supports this notion, stating that determining the risks of money laundering and terrorist financing and the techniques employed in virtual gaming worlds are relatively new. Although money laundering and terrorist financing have different goals and objectives, Samantha Maitland et al. (2011) study argues that both money laundering and terrorist financing should be viewed through the same lens since they share many techniques, and trends for acquiring, utilizing, and distributing financial resources to launder money or create terror. The article highlights the lack of research progress in AML/CTF techniques and emphasises the importance of machine learning detection models specifically for combating terrorist financing. Beeres et al. (2017) discusses the effectiveness of statistical profiling methods to combat the spread of terrorism. The study concludes that the use of statistical profiling in the fight against terrorism remains limited. Bolshibayeva et al. (2023) paper provides an overview of the machine learning approaches used to detect the financing of terrorist organisations in Kazakhstan. The study discovered k-means clustering as a promising technique for categorising financial transactions based on their similarities. However, the study acknowledged some limitations, including the inability to exit a local optima.

2.8 Supervised Learning Methods

The literature on the detection of illicit financial transactions is divided into two main classes The first group of statistical detection methods utilizes supervised learning algorithms and the other implements semi-supervised and unsupervised learning and anomaly detection models (Sudjianto et al. 2010, Hilal et al. 2022). Supervised machine learning algorithms have found a niche application in the detection of financial crimes. They are defined as a subset of machine learning techniques that make use of labeled, ground-truth data sets to train and test models. The defining characteristic of supervised learning is the availability of a response or dependent variable in the data set for model training (Cunningham et al. 2008). Classic applications include credit default risk assessment, Natural Processing Language (NPL), videos, and image analysis, where the input data are images, the training data is an annotated image database and the output is the corresponding number on the image (Tiwari 2022). In the presence of tagged or labeled transactions or cases, distributions of relevant variables can be constructed for legitimate and criminal behavior.

An application of supervised learning in detecting malicious financial crimes is supervised rule-based profiling of clientele transactions (Sudjianto et al. 2010, Jose-de Jesus et al. 2021). New transactions are compared to suspicious case profiles and subsequently flagged for further inspection based on their similarity to criminal behavior or deviations from expected legitimate behavior. Deviations from expected behavior or similarities to known fraudulent patterns may be indicative of criminal activity. Such a simple structure enables prompt decisions to be made on large amounts / streaming data (Sudjianto et al. 2010).

Rule-based profiling has proved to be a staple application by financial institutions owing much to its simplicity and ease of application (Colladon & Remondi 2017). The rules are generally Boolean conditions with thresholds based on historical data or business intuition, thus presenting a binary expert system for detecting suspicious activities. Supervised rule-based systems can also complement more sophisticated statistical techniques in filtering legitimate transactions and depend on rules that can be applied with little statistical expertise (Shokry et al. 2020). However, the challenge with supervised rule-based profiling is the need to constantly review and update the algorithm to reflect the dynamic changes in criminal as well as legitimate customer behavior. Failure to do so generates a high proportion of false positives as the system fails to dynamically adjust the set of rules by the changes in criminal behavior (Jose-de Jesus et al. 2021). Thresholds are usually based on supervised training of historical data and have to be updated when introduced to new information.

2.9 Classification

In the context of terrorism financing, the goal of statistical classifiers is to use labeled data to train models that determine the probability of each observation being a criminal transaction. Mercer (1990) applied least squares regression methods on high-level data for fraud audits while Foster & Stine (2004) utilized a fully automated step-wise regression model to predict the onset of bankruptcy. Among the traditional statistical discrimination techniques, logistic regression and linear discriminant analysis are popular techniques employed for the detection of suspicious financial activity (Sudjianto et al. 2010). Dumitrescu et al. (2021) applies a penalized logistic regression to develop a credit scoring model that effectively classifies credit risks in credit default datasets. Kulatilleke (2022) employs classification models to classify highly imbalanced fraud detection datasets. However, with the rapid advancement of computers, more recent non-linear classifiers such as support vector machines, Bayesian belief networks, and neural networks have made headway in a number of applications such as intrusion detection systems, sentiment analysis classifications, face detection and recognition systems and text recognition (Abdullah & Abdulazeez 2021).

2.9.1 Support Vector Machines

Support Vector Machines (SVM) are linear binary classifiers that attempt to find the hyper-plane that separates two binary classes with a margin which often leads to better approximation accuracy on unseen or unlabeled data (Cortes & Vapnik 1995, Kulatilleke 2022). Assuming that two classes can be separated by a hyper-plane, there exists an optimal hyper-plane that separates the data points with the widest margin. Sudjianto et al. (2010) finds that SVM performs exceptionally well on large, non-linear separable groups and requires large training datasets to converge to a unique solution, making them an attractive option for fraud and money laundering detection. SVM use kernel functions to transform low-dimensional data into a suitable higher dimension that enables linear separation (Kulatilleke 2022). A kernel function represents the dot product of high-dimensional projections of two data points (Zojaji et al. 2016). It is a transformation that resolves data by mapping the input space onto a new feature space where the instances are more likely to be linearly separable. Since there are many decision hyper-planes, the learning task is to find the best hyper-plane that maximizes the margin, the distance between the hyper-plane, and the closest points (support vectors) (Zojaji et al. 2016). Chen & Yuille (2004) utilized support vector machines to deal with credit card fraud and money laundering respectively.

2.9.2 Bayesian Belief Networks and Neural Networks

Bayesian classifiers have proven to be both popular and effective in detecting suspicious financial activity. Khan et al. (2013) makes use of Bayesian Networks (BN), based on rules suggested by the State Bank of Pakistan and transaction histories, to investigate and detect suspicious patterns among 8.2 million transaction records of more than a hundred thousand clients. According to Barbrook-Johnson & Penn (2022), a Bayesian Belief Network (BBN) is an acyclic network of conditional probabilities that specifies the likelihood or probability of variables represented by nodes assuming different states. These probabilities are based on the states of the parent nodes to which they are connected, and the nodes themselves are conditionally dependent on the states of the nodes with which they share a causal relationship. BBNs can assess the likelihood of achieving outcomes and quantify the impact of changes elsewhere in the system on outcomes (Barbrook-Johnson & Penn 2022). Bayesian classifiers derive the conditional probabilities of each attribute by learning from the training data given a specific class label. Bayes rules are applied to calculate the probability of a class provided a particular instance has occurred and predict the class with the highest posterior probability. This process is strongly hinged on the probabilistic assumption of independence. However, it is virtually impossible to ignore the correlations between instances for example age, income, and education when assessing credit risk (Friedman et al. 1997) Bayesian Networks can address the independence assumption associated

with Bayesian classifiers as they provide a mathematical structure for modeling complex relationships among variables while preserving a simplistic visualization of such relationships (Khan et al. 2013). On the other hand, Jose-de Jesus et al. (2021) admits that the model is based solely on transnational characteristics and fails to make peer group comparisons. Sudjianto et al. (2010) also warns against drawing causal conclusions from the results, emphasizing that there are several network structures with equivalent classification performance and therefore, several explanations for the observations. Yee et al. (2018) concluded that the Bayesian classifiers are capable of predicting and classifying transaction activities of credit card fraudsters when the data is appropriately processed. In addition to the use of Bayesian networks in the detection of financial crime, Artificial Neural Networks (ANN) have also proven to be popular and widely used. Abraham (2005) describes ANN as mathematical models that generalize the biological nervous systems. Islam et al. (2019) further states that ANNs are a form of artificial intelligence that attempts to mimic the human brain by making connections between processing elements and using the weights of the connection to determine the outcome. Neural networks are usually trained by using backpropagation, where the weights, which are initialized randomly, are changed recursively to minimize the training error (Arbib 2003). Once trained, neural networks analyze new data efficiently and in a short time, which is crucial for the real-time detection of suspicious transaction activity. However, training a neural network on a large training dataset is time-consuming. Multi-layer perceptrons trained with backpropagation tend to overfit and their result is difficult to interpret (Sudjianto et al. 2010, Zakaria et al. 2014).

2.9.3 Link Analysis

Terrorists like criminal organizations are often organized into crime rings and networks. Thus, some transactions and clients appear as legal transactions when examined individually however, when viewed in the context of a network of activity involving several related individuals, the pattern of criminal behavior becomes apparent (Sudjianto et al. 2010). Savage et al. (2016) used the same concept to develop an automated system for detecting groups of money launders in the financial system by combining network analysis and supervised learning techniques. The study represented transaction relationships as networks comprised of nodes and edges that, connect to different sending and receiving parties. Using network analysis, they established different relationships and parties of interest linked by actual remittance transactions and by shared accounts and agents. In addition, the weights on the edges can be seen as the probabilities of selecting a random group of transactions associated with some given evidence. To find evidence of money laundering, the study placed attention on identifying small sets of interacting parties who exhibit suspicious behavior. Although the paper managed to identify suspicious activity with a low rate of false positives, it acknowledged the inability to accommodate heterogeneous

networks and excessively large communities as limitations for detecting existing communities. Savage et al. (2016) also considered support vector machines using a linear, polynomial kernel and a random forest set of one hundred trees for their supervised machine learning classifiers. Both classifiers performed at a level acceptable for implementation in a real-world environment. The random forest slightly outperformed the support vector machines. Both models displayed high precision in detecting money laundering; the average recall of the classifiers was quite low (Savage et al. 2016). Though the system provides a general groundwork for identifying money laundering activity in a transaction network, its dependence on expert domain knowledge and the lack of awareness of community dynamics were noted as limitations to the model's capabilities.

2.10 Unsupervised Machine Learning Algorithms

The disadvantage of employing supervised machine learning algorithms to detect money laundering and financing terrorism stems from its core feature: the availability of response variables. This has proven to be a stumbling block in developing machine learning algorithms applicable to real-world scenarios for several reasons. Firstly, financial crime data can be unavailable or at best unreliable due to confidentiality and security reasons. Jose-de Jesus et al. (2021) acknowledges this fact by stating that supervised learning methods require ground-truth labeled data, something most financial institutions may have difficulty obtaining. The number of officially recognized money laundering and terrorist financing cases is limited, making it difficult to construct a labeled dataset (Liu et al. 2008, Sudjianto et al. 2010).It is also worth noting that assigning labels to previous transactions requires a lot of time and domain expertise, not to mention being prone to classification errors. In the case of money laundering and terrorist financing, it is virtually impossible to assign objective labels (Sudjianto et al. 2010). Shokry et al. (2020) also notes that supervised machine learning models can only detect suspicious patterns and transaction activities familiar with the patterns learned from the training data. Considering the shortfalls concerning supervised learning, unsupervised machine learning techniques may be utilized. Unsupervised machine learning techniques are more effective and able to promptly detect new patterns of suspicious activity without prior knowledge of the suspicious account (Shokry et al. 2020).

2.10.1 Clustering

Scholarly literature has extensively documented the utility of clustering as an unsupervised learning technique for detecting financial crime. Clustering is the analytic process that attempts to segregate the data into groups of similar observations using some measure of similarity, such as distance or density (Sudjianto et al. 2010, Kapp-Joswig & Keller 2022). Wierzchoń & Kłopotek (2018) states that the main feature of cluster analysis is that it generates disjoint subsets of the dataset so that the differences between the data points belonging to different groups are greater than those belonging to the same group. The main purpose of cluster analysis is to reveal the natural structure of the dataset (Wierzchoń & Kłopotek 2018). Gao (2009) used a cluster-based local outlier factor to detect suspicious money laundering transitional behavior patterns. However, clustering alone is not sufficient to detect anomalous financial transactions but is essential to creating peer groups for comparison (Sudjianto et al. 2010). Peer groups enable the comparison of individual transactions to other transactions of similar characteristics; thus, one transaction may appear unusual but would be normal when compared to a common group of transactions (Jose-de Jesus et al. 2021).

Clustering methods are also combined with supervised methods. Sánchez-Rebollo et al. (2019) used graphs and fuzzy clustering techniques to analyze social media messages to detect terrorist network groups. Raza & Haider (2011) combined distance-based clustering and dynamic Bayesian networks to identify anomalies in financial transactions. The model formed clusters of customers' monthly credit and used Dynamic Bayesian networks to capture patterns in a sequence of clients' financial transactions as well as compute the degree of the anomaly using the Anomaly Index Rank and Entropy. After testing the model on a dataset of non-corporate banking customers, the study concluded that the model was accurate in predicting the mode and number of incoming transactions. Clustering allows group comparison as the financial transaction behaviors of one customer are compared with other customers of similar traits. Although clustering algorithms may be adequate in detecting money launders in the presence of unlabelled data, they have been primarily focused on analyzing transactional features (Jose-de Jesus et al. 2021).

2.10.2 Anomaly Detection

Mandhare & Idate (2017) defines an outlier as any data point that is dissimilar, inconsistent, irrelevant, or malicious from the dataset. Chalapathy & Chawla (2019) further elaborates an outlier as an observation that deviates so drastically from the other observations that it raises the possibility that it was generated by a separate mechanism. Thus, the underlying principle is that anomalies are indicative of a new, unknown underlying process (Chalapathy & Chawla

2019). Anomaly detection can be defined as the process of identifying patterns that stray from expected behavior. The process involves identifying a region of normal behavior and any occurrence outside the region of normalcy would be declared an anomaly. It can also be equated to novelty detection which aims to detect emergent patterns in the data (Markou & Singh 2003, Chandola et al. 2009). Chandola et al. (2009) provides a comprehensive survey of the existing anomaly detection techniques and the challenges associated with anomaly detection. These challenges include defining the regions of normalcy especially when the boundary differentiating normal and anomalous behavior is unclear. Chalapathy & Chawla (2019) expands on the survey by integrating concepts of deep learning with anomaly detection methods. Even though there is an improvement in performance and adaptability, Chalapathy & Chawla (2019) notes that the same issues and difficulties persist, pointing out the lack of well defined normal boundaries will present difficulties for both traditional and deep learning anomaly detection methods. Most outlier detection algorithms fall into two classes; density-based outlier detection and distance-based outlier detection where the former identifies outliers as observations in regions of low concentration while the latter takes into consideration how spaced or "far apart" the observations are from the "center" of the data set (Sudjianto et al. 2010, Mandhare & Idate 2017). A common measure for the distance-based algorithms is the Euclidean Distance and the Mahalanobis distance. A study by Jose-de Jesus et al. (2021) proposes the use of neural networks and an abnormality indicator based on the variance to design a model that improves both self and group comparison of customer transactions to detect instances of money laundering and terrorism financing in financial systems. The study goes on further to highlight the importance of including non-transnational variables such as geographical zones, state of residence, type of customer, and length of business relations with the customer, noting that they have not been sufficiently explored in previous studies.

2.11 Information Systems in Countering Terrorist Finance and Money Laundering

Information Systems and Technology (IST) has proved to be a vital tool in combating financial crimes, especially when combined with machine learning algorithms. Leonov et al. (2019) states that the benefits of automated transaction monitoring systems stem from the need to introduce speed and efficiency to human-dependent systems. The study developed a financial transaction monitoring information system purposed to detect money laundering in an automated environment. Leonov et al. (2019) presented a prototype of a business process monitoring transactions in an automated environment. The prototype is a multi-agent system based on utilizing knowledge-based intelligent agents, with the mandate of autonomously monitoring and reporting suspicious

activities to the bank management.Gao et al. (2009) performed similar research. Alexandre & Balsa (2023) study proposed a multi-agent system that incorporates machine learning and risk components to detect suspicious bank customers. The system also assisted human specialists in analyzing the suspicious behavior of these customers. Jennings (2000) defines knowledge-based intelligent agents as autonomous computer systems located in a particular environment with the capability of executing a list of actions without human or other systems interfering. Emphasis is placed on the agent's ability to act independently and thus, have control over their internal state and actions. Gao et al. (2009) further postulates that an agent's flexibility also includes reactivity, pro-activeness, and social capabilities. Gao et al. (2009) system aims to function as an independent anti-money laundering system, linked with the existing bank system through which all client transactions can be monitored.

2.12 Chapter Summary

This chapter explored the academic literature and assessed the statistical methods for detecting financial crimes. The importance of the application of machine learning techniques in detecting financial crimes is relevant. This chapter also concludes that the reliance on supervised machine-learning algorithms is undesirable and introduced the utilizing unsupervised machine learning algorithms for designing such models as they are not reliant on labeled datasets.

Chapter 3

Preliminary Investigation

3.1 Chapter Introduction

Prior to the implementation of the research methodology, a preliminary investigation of the dataset was conducted. As a result, this standalone chapter focuses on cluster analysis as an initial step in analyzing unsupervised data. The fundamental goal of clustering is to discover partitions in the dataset. Section 3.2 introduces the migrant remittance dataset. A brief exploratory data analysis of the dataset is presented in Section 3.3. The agglomerative hierarchical clustering approach and the k-prototype clustering algorithm, which are suitable for identifying the presence of clusters in the mixed dataset, are thoroughly explained in Sections 3.4 and 3.6 respectively. Sections 3.7 and 3.8 provides the summary of the clustering results for both procedures.

3.2 Migrant Remittance Data

The dataset on migrant remittances was taken from the World Bank Repository (*World Bank Database Repository* 2015). The World Bank conducted a survey in 2015, interviewing people from the Democratic Republic of Congo (DRC), Nigeria, and Senegal who lived in Belgium to investigate and better understand migration trends and migrant remittance patterns. Accessing financial data with labels has proven difficult and time-consuming due to security and confidentiality concerns. As a result, the migrant remittance dataset used in this study is unlabeled or unsupervised, which means it lacks a response variable that flags questionable transactions. Nonetheless, the absence of a response variable was expected, and contingencies were planned.

The dataset has three distinct sections. The first section contains logistical and administrative information. These attributes will not be used in the data analysis however, they will provide useful information about particular variables of interest upon comparing results. The second section describes the personal, social, and economic attributes of the migrants residing in Belgium. Important variables to note in this section include the migrant's age, the highest level of education, and the family relationship with the recipient. In addition, this section captures key socio-economic characteristics which include the total number of people residing with the migrant in Belgium, the number of senders if multiple senders contribute funds for a single transaction, and the different channels used to transfer funds. The last section of the dataset contains the recipient's financial and household information.

This section details the financial transactions and the recipient's socio-economic status. The variables included in this section are the recipient's age, educational status, and household size. This also contains other geographical attributes such as the recipient's access to basic social amenities such as electricity and water. Other important attributes include the time to complete a transaction, the cost of remittance, the transfer fees, and the exchange commission. Moreover, the net amount transferred and the money received by the recipient have all been expressed in three different currencies. To enable a fair comparison, all monetary values in the dataset were converted into United States dollars. It is vital to highlight that all information contained in the dataset is fully confidential, and no personal information, such as migrant's names and addresses was revealed.

3.3 Exploratory Data Analysis

The migrant remittance dataset is a mixed dataset of 1087 observations with 64 variables, 17 numerical variables, 26 categorical variables and, the remaining variables are character and identity variables. The dataset was converted to a CSV format from its original Excel format. The descriptive statistical results, shown in Table 3.1 were compiled using the *R* programming software.

Table 3.1: The descriptive statistical results for the numerical variables

Variable	Mean	Stan Dev	25th percentile	50th percentile	75th percentile
Total number of residence	3.47	2.08	2	3	5
Total Number of Recipients	2.11	1.25	1	2	3
Age	29.9	20.0	17	30	38
Year of migration	14.0	11.4	0	15	21
Frequency of remittances	8.06	14.9	2	4	12
Total Value of Remittance	1145.0	1726.0	300	600	1200
Recipient's Age	48.1	18.2	32	50	62
Size of Recipient household	6.21	5.25	3	5	8
Last Remittance (Months)	34.8	32.8	0	60	62
Net Costs of Remittances	166.0	622.0	0	22	165
Amount Received	119.0	1118. 0	0	0	45.5
Remitting Time (Hours)	13.5	41.8	0	1	12
Total Remittance Cost	12.5	14.5	0	8	15
Transfer Costs	14.8	22.0	8	10	16
Exchange Commission	74.5	203.0	0	0	0
Additional Money Sent	147.0	232.0	0	50	206.0
Total Value of Goods sent	706.0	1330. 0	900	100	900

The results in Table 3.1 show that migrant individuals preferred transfers that take less than a day, with a mean of 13.50 hours and a standard deviation of 41.8 hours. This shows that speed and simplicity are key attributes considered by migrants when transferring funds. The results also show that the average migrant age is 29.9 years and they remit on average 8 times per year. The recipient's average age was pegged at 48.1 years with a standard deviation of 18 years. This is in line with recent patterns of young adults migrating abroad to find ways of supporting their parents or guardians back to their countries of origin.

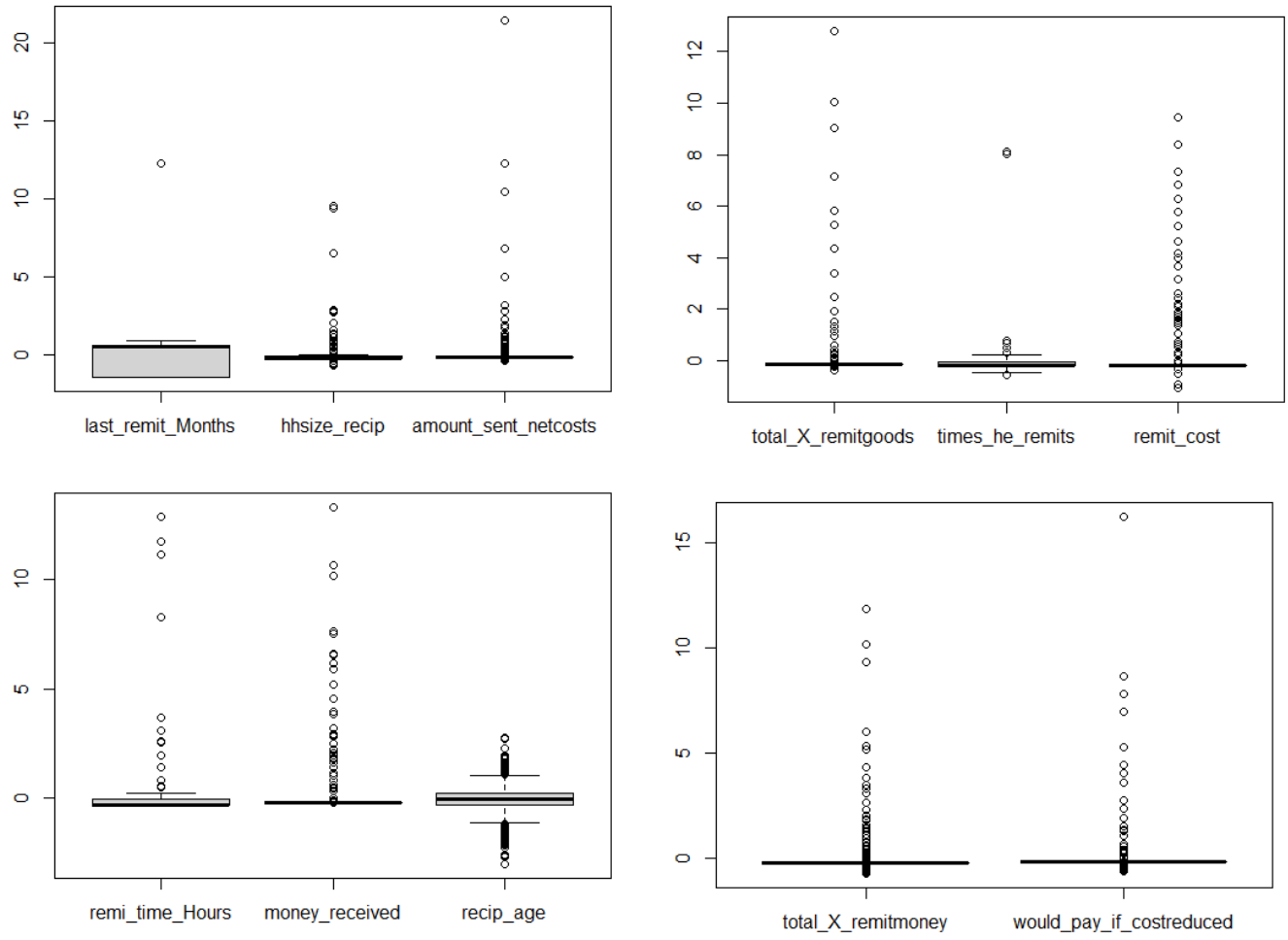

Figure 3.1: Boxplot diagrams of the numerical variables

Using the standardized numerical variables, Figure 3.1 depicts boxplots of some of the numerical variables associated with financial transactions. The boxplots suggest the presence of extreme outliers in numerical variables.The distribution of most of the variables is one-sided, an indication of a skewed distribution with a long tail. The recipient household size attribute has a similar distribution to the other financial-related variables as shown in Figure 3.1.

3.4 Clustering

As mentioned in Chapter 2, the use of unsupervised learning algorithms has several merits in modeling real-world data. Clustering is the most frequently used methodology in understanding the structure and groupings found in unlabeled data (Jose-de Jesus et al. 2021). Rokach & Maimon (2005) defines clustering as that which groups data similar data instances and different instances belong to different groups, thus if a set $\mathbf{S}$ contains $\mathbf{C}_1, \mathbf{C}_2, ..., \mathbf{C}_k$ clusters then, the clustering structure would be $\mathbf{S} = \cup_{i=1}^{k} \mathbf{C}_i$ and $\mathbf{C}_i \cap \mathbf{C}_j = \{\}$ given that $i \neq j$. With supervised algorithms, accuracy is an essential measurement of an algorithm's validity and performance however, the task becomes increasingly difficult in the absence of a response variable in the dataset, especially when ascertaining an unsupervised model's accuracy and performance. Thus, assessing the compactness and separation of individual clusters within the dataset provides a more logical assessment of evaluating unsupervised clustering models (Flach 2012). Compactness, in this case, refers to how close similar objects or data points are relative to each other, and separation refers to how far dissimilar points are to each other (Jose-de Jesus et al. 2021). Classical metrics that determine a cluster's compactness and separation are variance and distance between the cluster centers, although Said et al. (2017) notes that distance fails to reflect the quality of the separation of clusters.

3.5 Agglomerative Hierarchical Clustering

The Agglomerative Hierarchical Clustering technique is a descriptive clustering technique that initially represents each data point as a cluster and at each iteration, the algorithm makes a new partition of data by merging the two closest clusters until the whole data set has been merged into one cluster (Flach 2012). Unlike decision trees that make use of features to split the vector space, hierarchical clustering algorithms indirectly use the set features as a basis on which the distance is computed thus, partitioning the data instead of the feature space itself (Flach 2012). The Agglomerative Hierarchical Clustering technique utilizes the *Minkowski distance* metric between the two points which is defined as shown in Equation 3.1:

$$Dis_p(\mathbf{X}, \mathbf{Y}) = \left(\sum_{j=1}^{n} |x_j - y_j|^p \right)^{\frac{1}{p}} ; \forall\ p \geq 1, p \in \mathbb{Z}^+, \tag{3.1}$$

where p is a positive integer and the $\mathbf{X} = \{x_1, x_2, ..., x_n\}$ and $\mathbf{Y} = \{y_1, y_2, ..., y_n\}$ are two points in the $\mathbb{R}^n$ dimensional space.

Minkowski distance provides a generalization of the different distance measurements. For instance, the *Manhattan distance* and the *Euclidean distance* are Minkowski distances of order $p = 1$ and $p = 2$ respectively. The final output is displayed as a dendrogram, which is a binary tree with the dataset elements as its leaves and the internal node of the tree representing a subset/cluster of similar leaves thus representing the distance between two clusters (Flach 2012). To calculate the distance between clusters, linkage functions are employed. The single linkage function computes the distance between two clusters as the smallest distance between elements from each cluster while the complete linkage function calculates the distance between two clusters as the largest point-wise distance. The average linkage function finds the cluster distance as the average point-wise distance. Lastly, the centroid linkage finds the cluster distance between cluster mean points (Flach 2012). It is important to note that the single linkage function appears to be the easiest to compute the distances between clusters. However, when considering the time complexity for each function, the single linkage function has a function $O(n)$ while other functions require at least $O(n^2 log\ n)$ time for n points which indicates that the single linkage function may initially outperform the other linkage models but will gradually deteriorate as the values of n become larger (Flach 2012). One of the merits of the agglomerative hierarchical clustering is the lack of dependence on the k clusters initialisation. The algorithm can partition the data without initially declaring the optimal number of cluster. However, the optimal number of clusters can be estimated by implementing the techniques like the silhouette technique and elbow method.

3.5.1 Elbow Method

The elbow method estimates the optimal number of k clusters in a dataset by plotting the Within-Cluster- Sum of Squared Errors (WCSS) against different numbers of k clusters. The WCSS measures the squared mean distances of points within a cluster from a cluster center (Flach 2012). Yuan & Yang (2019) defines the elbow method as the use of a square of the distance between the sample points and the centroid of the cluster to give a series of k values making the error sum of squares or the within-cluster-sum of squares (WCSS) a measure of performance.

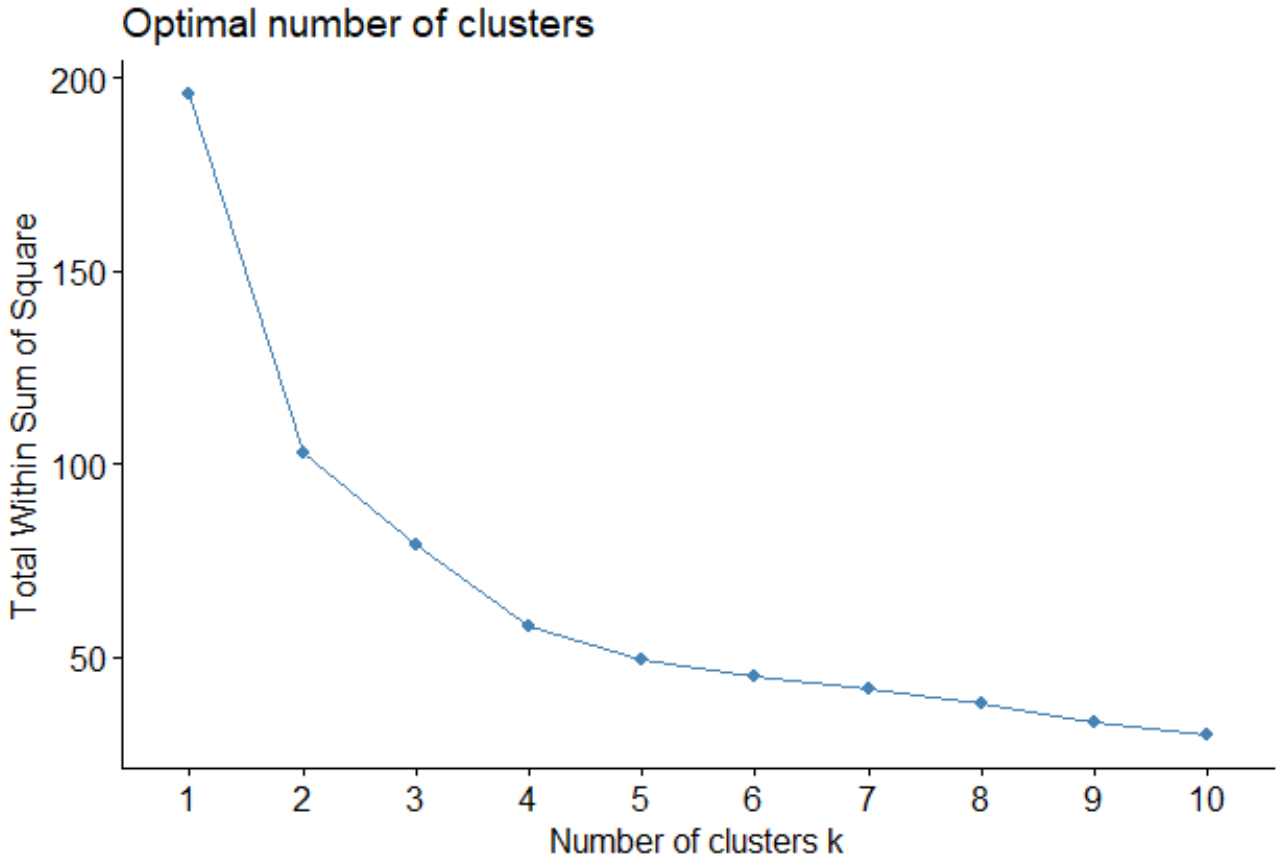

Figure 3.2: An illustration of elbow method depicting $k = 4$ as the optimal number of clusters

The optimal value of k is when the WCSS begins to diminish or tapers out, forming an elbow-like configuration as shown in Figure 3.2. The explained variations change rapidly for smaller clusters, as compared to larger numbers of k clusters leading to an elbow point in the graph. When the value of clusters exceeds this point of inflection, the WCSS will continue to decrease at a much slower rate.

3.6 *k*-prototype Algorithm

The main goal of the k-prototype algorithm is to cluster mixed datasets that contain numerical and categorical data (Huang 1998). This is achieved by combining the k-means and k-modes algorithms (Huang & Ng 2003). The k-prototype algorithm is the combination of the k-means and the k-modes algorithm that can be utilized to partition mixed-type data objects into clusters (Huang 1998). Thus, it is important to explain the concepts behind the k-means and the k-modes algorithms.

3.6.1 *k*-Means Algorithm

The k-means clustering algorithm is a non-probabilistic technique that seeks to allocate centroids to data clusters that minimize the squared Euclidean distance (Flach 2012). Centroids are points defined in a p-dimensional space that finds the average measurement values along each dimension (Kaufman & Rousseeuw 2009). The algorithm achieves this goal by repeatedly partitioning the data with the nearest centroid and updating the value of the centroids from a partitioned dataset until it converges to a stationary point. However, it is hard to ascertain whether the stationary point is a local or global minimum (Flach 2012). Given a set of numerical data objects $\mathbf{X}$ and an integer number k, the k-means algorithm searches for a partition of $\mathbf{X}$ into k clusters that minimize the WCSS (Huang 1998). This can be formulated into a minimization problem and the objective function $P(W, \mathbf{Q})$ shown in Equation 3.2:

$$\begin{aligned} &Minimise \quad P(W, \mathbf{Q}) = \sum_{l=1}^{k} \sum_{i=1}^{n} w_{i,l} |d(X_i, Q_l)|, \\ &subject\ to \quad \sum_{l=1}^{k} w_{i,l} = 1, \\ &\qquad\qquad w_{i,l} \in [0,1], \quad 1 \leq\ i \leq\ n,\ 1 \leq\ l \leq\ k, \end{aligned} \tag{3.2}$$

where $W = [w_{i,l} \in \{0,1\}]$ is an $n \times k$ partition matrix representing binary elements that partitions the n vectors into k clusters. $\mathbf{Q} = \{Q_1, Q_2, ..., Q_k\}$ are is the set of cluster centroids for k clusters, and the function $d(.)$ is the Minkowski distances of order $p = 2$ or the squared Euclidean distance between two objects. The aim is to find the values of $w_{i,l}$ and Q_k that minimize the objective function, $P(W, \mathbf{Q})$. This can be achieved by iterations of two steps that optimize $w_{i,l}$ and Q_k respectively until convergence is attained. The algorithm updates the cluster centers and membership with:

$$w_{i,l} = \begin{cases} 1,\ if\ k = \arg\min_j\ d(X_i, Q_k), \\ 0,\ otherwise. \end{cases} \tag{3.3}$$

For the optimization of Q_k, we find the derivative of the objective function, Equation 3.2 with respect to Q_k and equating it to zero thus, we get Equation 3.4

$$Q_k = \frac{\sum_{i=1}^{n} w_{i,l} x_{i,j}}{\sum_{i=1}^{n} w_{i,l}}. \tag{3.4}$$

Though it is possible to obtain the optimal number of clusters needed to partition a dataset by trial and error, it will become impossible to visualize the clusters present in higher dimensional

data sets. Techniques such as the elbow method discussed in Section 3.5.1 can be used to determine the optimal number of clusters. However, a study by Sinaga & Yang (2020) devised a modified k-means algorithm that automatically assigns the optimal number of clusters by introducing a penalty term to the k-means model.

3.6.2 *k*-Modes Algorithm

The application of the k-means algorithm shown in Section 3.6.1 is restricted only to numerical datasets yet real-world datasets also contain categorical data. The k-modes algorithm extends the logic of the k-means algorithm to categorical variables (Huang 1998). This is made possible by using a simple matching dissimilarity instead of the Euclidean distance as a measurement metric and replacing the centroids used in the k-means with modes and applying a frequency-based approach to find the modes that minimize the dissimilarity function (Huang 1998, Huang & Ng 2003). As an illustration of the simple matching dissimilarity, suppose X and Y are two categorical objects described by m categorical attributes. The dissimilarity between X and Y is the total mismatch of the corresponding attribute categories of the two objects (Huang 1998). The smaller differences indicate more similar objects. This can be formulated as :

$$d_1\left(X,Y\right) = \sum_{j=1}^{m} \delta(x_j, y_j), \tag{3.5}$$

where

$$\delta(x_j, y_j) = \begin{cases} 0, \ if \ x_j = y_j \\ 1, \ if \ x_j \neq y_j. \end{cases} \tag{3.6}$$

Thus, the dissimilarity measure for categorical variables is defined as the total mismatches of corresponding categorical attributes of two objects (Huang 1998, Kaufman & Rousseeuw 2009).

Let $\mathbf{X}$ be a set of categorical objects described by categorical attributes, $A_1, A_2 ... A_m$. The mode of $\mathbf{X}$ is a vector, $Q = [q_1, q_2, ... q_m]$ that minimises:

$$D(\mathbf{X}, Q) = \sum_{i=1}^{n} d_1(X_i, Q). \tag{3.7}$$

Q is not necessarily an element of $\mathbf{X}$. Now let $n_{c_{k,j}}$ be the number of objects having the kth category $c_{k,j}$ in attributed A_j and $f_r(A_j = c_{k,j}|\mathbf{X}) = \frac{n_{c_{k,j}}}{n}$ be the relative frequency of the category $c_{k,j}$ in $\mathbf{X}$. The function $D(\mathbf{X}, Q)$ is minimised if and only if $f_r(A_j = q_j|\mathbf{X}) \geq f_r(A_j = c_{k,j}|\mathbf{X}) \ for \ q_j \neq c_{k,j} \ for \ all \ j = 1, ..., m$. Using the dissimilarity measure in Equation 3.5, the

cost function becomes:

$$P(W, \mathbf{Q}) = \sum_{l=1}^{k} \sum_{i=1}^{n} \sum_{j=1}^{m} w_{i,l} \delta(x_{i,j}, q_{l,j}), \tag{3.8}$$

where $w_{i,l} \in W$ and $Q_l = [q_{l,1}, q_{l,2}, ..., q_{l,m}] \in \mathbf{Q}$.

3.6.3 *k*-prototype Algorithm

Now the k-prototype algorithm combines both the k-means and the k-modes algorithms. Thus, if $\mathbf{X}$ and $\mathbf{Y}$ are two mixed variable objects with attributes, $A_1^r, A_2^r, ..., A_p^r, A_{p+1}^c, ..., A_m^c$ where A^r are the numerical attributes and A^c are the categorical attributes, the dissimilarity between the two can be measured by

$$d_2(X, Y) = \sum_{j=1}^{p} (x_j - y_j)^2 + \gamma \sum_{j=p+1}^{m} \delta(x_j, y_j). \tag{3.9}$$

The first term in Equation 3.9 is the Euclidean distance measure on the numeric and the second term is the simple matching dissimilarity measure on the categorical attributes while γ acts as the weight used to avoid biases for a particular type of attribute. The modified cost function for the k-prototype clustering algorithm is shown in Equation 3.10:

$$P(\mathbf{W}, \mathbf{Q}) = \sum_{l=1}^{k} \left(\sum_{i=1}^{n} w_{i,l} \sum_{j=1}^{p} (x_{i,j} - q_{l,j})^2 + \gamma \sum_{i=1}^{n} w_{i,l} \sum_{j=p+1}^{m} \delta(x_{i,j}, q_{l,j}) \right). \tag{3.10}$$

Since both $\sum_{i=1}^{n} w_{i,l} \sum_{j=1}^{p} (x_{i,j} - q_{l,j})^2$ and$\gamma \sum_{i=1}^{n} w_{i,l} \sum_{j=p+1}^{m} \delta(x_{i,j}, q_{l,j})$ are non negative, minimising $P(\mathbf{W}, \mathbf{Q})$ is equivalent to minimising the Equation 3.2 for $1 \leq l \leq k$. The approach in Section 3.2 can still be used to determine a locally optimal $\mathbf{Q}^*$ and W^*, as only $d(.)$ has changed. Equation 3.9 calculates W using the k-means method for a given $\hat{\mathbf{Q}}$. To get $\mathbf{Q}$, we minimize Equation 3.10 for $1 \leq l \leq k$. Similar to the k-means algorithm, the k-prototype needs to determine the k number of clusters prior to clustering, hence techniques such as the elbow method can be implemented to determined the optimal k clusters.

3.7 Preliminary Results: Agglomerative Hierarchical Clustering

This section presents the results obtained from the preliminary investigation. The clustering was first done on the numerical variables as finding the distance matrix of such variables carries a relative meaning of closeness. The resultant dendrogram is shown in Figure 3.3.

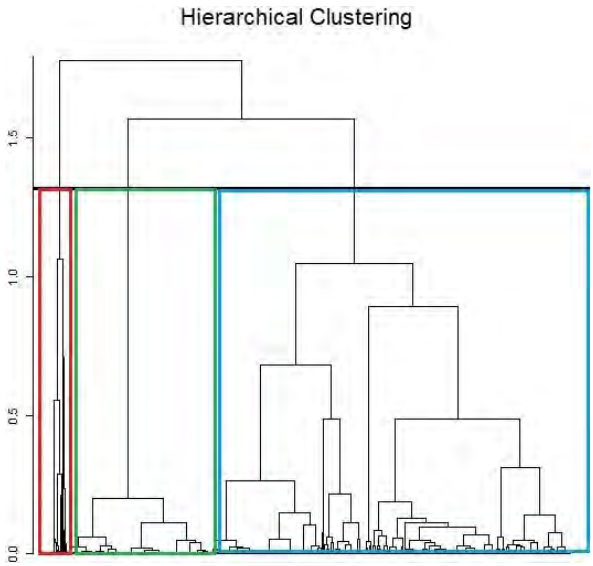

Figure 3.3: The dendrogram of the migrant remittance dataset

After using the elbow method to determine optimal number of clusters Johnson & Wichern (2007), the cluster configuration from the dendrogram is evaluated by counting the how many times the horizontal line (the threshold line) touches the vertical lines (individual data points) denoting clusters in the dendrogram. Each intersection point represents a cluster division.

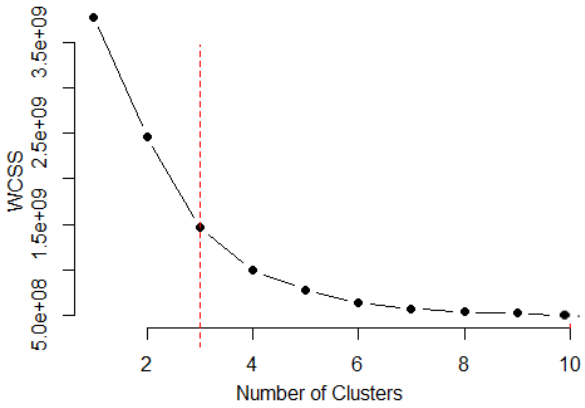

Figure 3.4: The graph depicting the loss in Within-Cluster Sum of Square (elbow method)

Figure 3.4 shows the elbow method graph shows the elbow is formed when the number of k clusters is approximately equal to 3. The first three clusters have the largest variation as shown by the steep gradient when $k < 4$.

3.8 *k*-prototypes Algorithm

The k-prototype algorithm was implemented to cluster the entire migrant remittance dataset. The R programming software was utilized for this task. Using $k = 3$ as our initialization number

of clusters, the scatter plot showing the k-prototype clustering output is shown in Figure 3.5 below. The summary of k-prototype clustering results from R programming are presented in the Appendix.

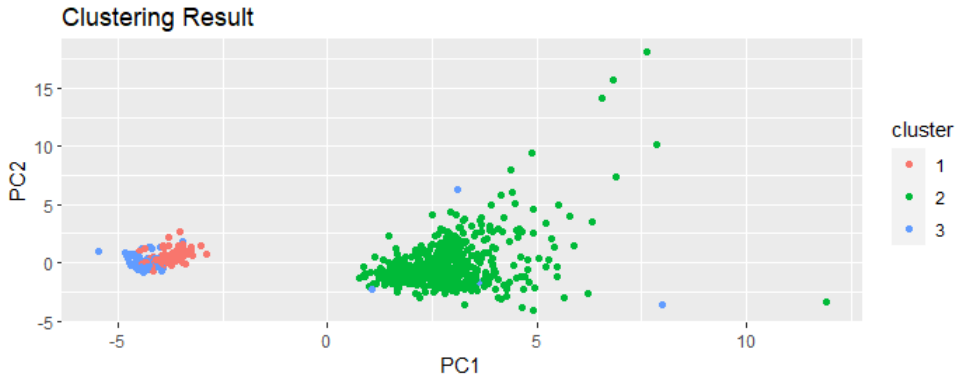

Figure 3.5: The results of the k-prototype clustering

3.8.1 First Cluster

From the results, the first cluster has a mean household family size of 4.31 and a much smaller mean household size of 2.16, which resides in the native country. The average age of the migrant sender is 20.15 years old making it the youngest age cluster. This corresponds to the average number of years a person has lived in Belgium, which is on average, 6.79 years. In addition, 50.2% of individuals in the first cluster listed miscellaneous work as their main economic activity for the past four weeks while 26.4% stated they temporary jobs for wages. This suggests that the main source of income for individuals in the first cluster is derived from temporary, part-time work. This also coincides with the fact that 27.3% of the individuals in the first cluster have mentioned secondary school education as their highest form of education. Without proper education, it is difficult to obtain permanent employment. When taking into account the total income earned in the past four weeks, 65.4% of migrants in the first cluster had no net income, whereas 0.02% earned €2500.00 or higher. Western Union was the most chosen money transfer agent, with 77% used Western Union for their last transaction.

3.8.2 Second Cluster

In the second cluster, both the mean household sizes in the native country and in Belgium are similar to the first cluster. However, there is a marked difference in the age of the migrant senders, who average 37.5 years. In addition, second-cluster migrants have resided in Belgium three times longer when compared to those in the first cluster. Individuals in the second cluster have a broad range of occupations. A little over 33.3% of the migrants work miscellaneous jobs

for wages, while 25.9% are self-employed and 22.2% have retired. Thus, unlike the first cluster which predominately had young individuals, the second cluster has a much older and experienced individuals, well acclimatised to the Belgian way of life. Furthermore, 70.3% of individuals in the second cluster have tertiary education. The total income for second cluster individuals ranges from €1000 to more than €2500, significantly more than the first cluster individuals. Unlike the individuals in the first cluster, the preferred channel of remittance is informally through family and friends. This may be attributed to the established relations and numerous connections built over the years and the need to reduce the high costs incurred when transferring funds with money agents such as Western Union.

3.8.3 Third Cluster

The household size for migrant individuals in the third cluster is 2.7 which is a smaller household compared to the other clusters. The average age of 37.9 years is similar to the second cluster however, their academic experience is vastly different. Moreover, 21.2% of the respondents have little to no academic attendance, 41.1% have attended secondary school and 12.2% have an undergraduate degree. Thus, due to the low qualifications, the main occupation for the majority of individuals in the third cluster is informal and labeled as other. Despite 14.3% refusing to disclose their total earnings, 48.9% made a total monthly income of between €500 to €1500.

3.9 Chapter Summary

This chapter investigated the natural clusters in the dataset. For clustering, an agglomerative hierarchical clustering, and the k-prototype clustering methods were used for this task. The R output results are in the Appendix. The results from the k-prototype algorithm where conclusive. The total income earned by individuals in all the three clusters depended on the level of education. Migrants who attained a higher form of education where most likely capable of earning a substantially higher income. Interestingly, most of the individuals in all the clusters, preferred either to use Western Union money transfer agent or family or returning relatives to remit funds back to their countries of origin. This reflects the conscious decision among migrants to prioritise the safety and security of the remitting funds. Moreover, it reflects either the awareness of the dangers of using informal systems like Hawala or the lack of prevalence of such systems. Another important note is that all individual migrants remit to family members. Cases of migrants remitting funds to unknown individuals were inconsequential.

Chapter 4

Methodology

4.1 Chapter Introduction

After introduction of clustering techniques in Chapter 3, this chapter presents the more advanced methodology used to achieve the main research goal. This chapter details the methods and techniques used to design an unsupervised learning algorithm capable of detecting suspicious financial migrant remittances. Two proposed algorithms are developed simultaneously: the structural equation model, (SEM) in Section 4.2 and an ensemble outlier detection algorithm in Section 4.3. The aim is to determine suspicious migrant remittances transferred to aid terrorism and to assess the risk of legitimate transfers being misused for terrorist activities in the recipient country. Finally, the proposed ensemble outlier detection algorithm comprising of an Isolation Forest and a local outlier factor will be presented in Section 4.4 and Section 4.5.

4.2 Structural Equations Models

Kaplan (2008) defines structural equation models (SEMs) as a set of methods that aim to represent statistical assumptions of observed population parameters in the form of hypothesized structural parameters. Kline (2023) and Pearl (2012) further defined SEM as a causal inference method that converts a set of qualitative causal hypotheses or assumptions based on theories and empirical results into numerical estimates of model parameters. In short, SEM is a statistical technique that examines the relationship between latent and observed variables. Latent variables are evaluated indirectly by making connections to the observed variable, whereas observed variables are examined directly (Civelek 2018). Kline (2023) notes that other statistical methods are restricted versions of SEM, intended only for analyzing observed data. SEM can also be considered a multivariate analytical tool that incorporates multiple dependent and independent variables with indirect attributes, thus testing specific relationships between observed

and latent variables and enabling testing of events where experiments are considered impossible (Kline 1998) . Nonetheless, the principal aim for SEMs is not to fit the model but to determine whether the analysis answers fundamental theoretical questions (Kline 2023). Kaplan (2008) agrees and explains that the core premise of structural equations is to test basic theories. A typical SEM includes three main features, namely the indicator or observed (manifest) variables that indirectly measure a construct, latent variables that correspond to a hypothesized construct, and residual or error terms that represent the estimated variance that is not explained by a factor and is likely caused by random measurement error or unreliability of the outcome (Kline 2023).

The presence of the measurement error term and the relationships between errors distinguishes SEM from other standard regression analysis, which ignores possible error (Civelek 2018). Furthermore, SEM is based on the variance-covariance matrix, whereas regression analysis relies on the correlation matrix. Another important premise of SEM is that it requires a large dataset to provide statistically significant results. Although Jung (2013) has adapted an SEM model to work with smaller samples, SEMs are generally considered large-sample techniques that primarily use covariances as the fundamental statistic for data analysis (Kline 2023). SEMs are useful for understanding causal linear relationships in psychology and the social sciences, where theory formulation and factor analysis are prominent features in these fields of study. SEMs have proven their practicality in these fields but have been criticized for their inherent complexity in their application (Kaplan 2008).

SEM shares several assumptions with classical regression analysis. The first assumption is that both the observable and latent variables have a multivariate normal distribution, whereas the error terms are independent and follow a normal distribution (Civelek 2018). As a component of factor and regression analysis, SEM assumes the connection between the latent and measurement variables is considered to be linear. The presences of outliers play a crucial role in the performance of SEM models. Before implementing SEM models, the dataset is expected to be free of outliers (Civelek 2018). A significant number of outliers violates the normalcy assumptions (Jenatabadi 2015). Nonetheless, Hoyle (2012) is primarily interested in the assumptions that govern the inference of a causal relationship between two variables. It is expected that the presumed cause or the measurement variable comes before the presumed endogenous variable, implying temporal precedence. In addition to temporal precedence, there is a correlation, or observed covariation between the measurement and latent variables (Hoyle 2012). Furthermore, the causal relation is believed to be correctly defined, so the measurement variables do affect the latent components (Hoyle 2012).

4.2.1 Model Conceptualisation

A typical SEM consists of a structural part that links latent factor variables and a measurement part that links latent and observed manifest variables. The development and implementation of a structural equation model follows six steps: specification, identification, estimation, evaluation, re-specification, and interpretation and reporting (Hoyle 2012). The process is illustrated in Figure 4.1.

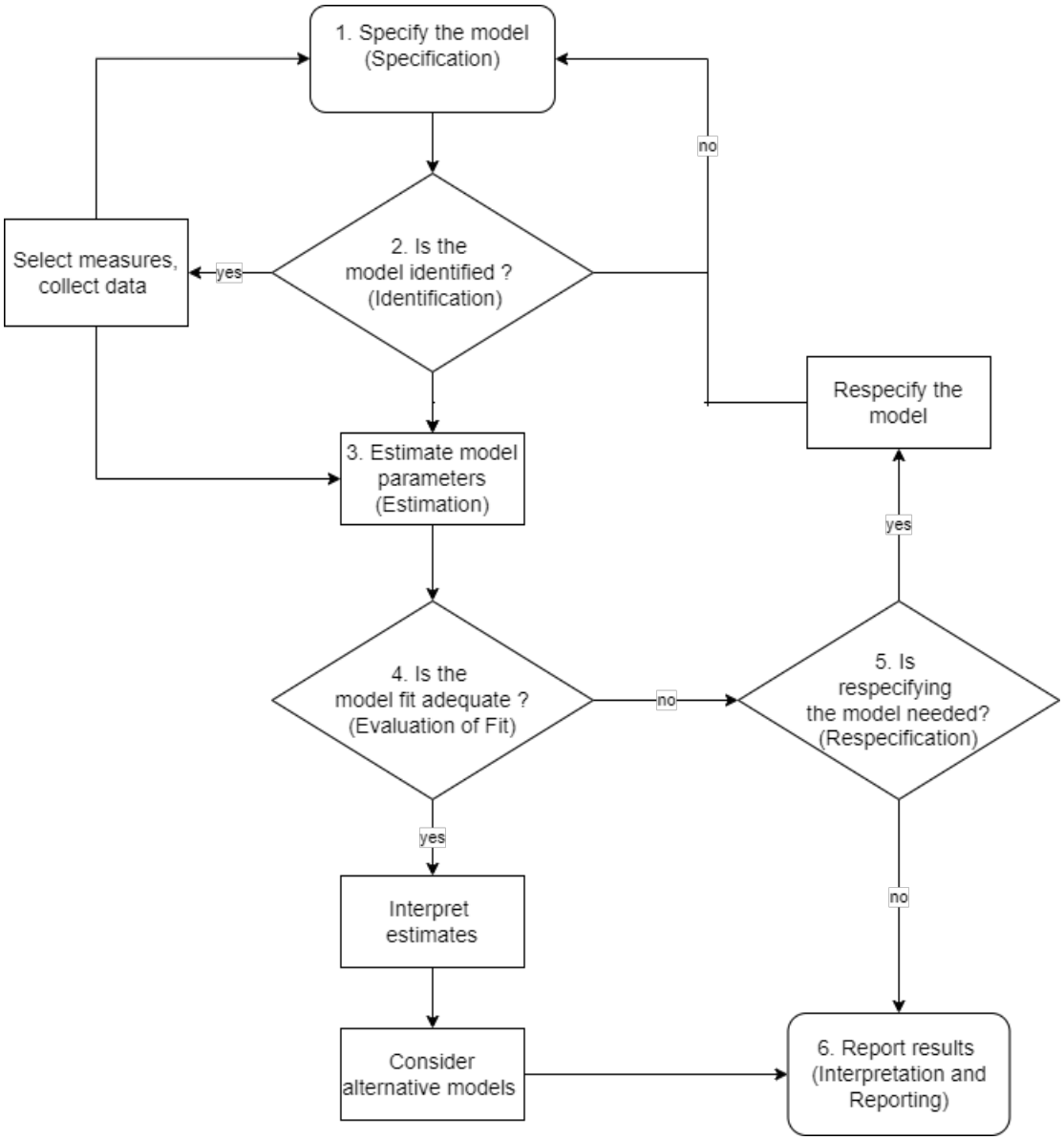

Figure 4.1: Flowchart depicting the development and implementation of the SEM steps (adapted from: (Kline 2023))

4.2.2 Specification

The model conceptualization begins with the specification of the underlying mechanism assumed to have generated the observed data. This involves variable selection, defining the relationship between variables and the status of the parameters in the model (Hoyle 2012). The initial stage of model formulation depends extensively on the researcher's domain expertise. They are responsible for selecting the latent and observed variables and must establish the relationships between these variables. A clear formulation and communication of the assumptions and their justification is important, as future results obtained in later steps assume that the model is true (Kline 2023). The model specification stage can be implemented before or after the data collection and preparation stage. It is the researcher's discretion to either formulate a substantive theory and then collect the data to back up their hypothesis or devise a model that explains the observed data already collected. When model specification precedes data collection and preparation, the hypothesized model acts as a guide to the choice of the data source or the type of data to be collected (Hoyle 2012).

4.2.2.1 Path Diagram

Some researchers may prefer to express their model hypotheses as graphical conceptual models or path diagrams, which provide a visual representation of theoretical variables of interest and expected relationships between them (Kline 2023). It is important to introduce the shape and features of the path diagram. Although there may exist several path diagram variations, the underlying principle is generally the same. The path diagram has three main features: squares, circles, and arrows. The squares represent the observed variables that were directly measured. The circles represent the hypothetical constructs or the latent variables (Hoyle 2012). Variables in a typical SEM can either have an association or be a direct effect of an independent variable on a dependent variable. An association between variables, represented by a double-headed or bidirectional arrow, reflects the covariation between two variables. The arrow can also represent the variance of a variable if it starts and ends at the same variable (Hoyle 2012). A direct arrow with a single point represents the direct effect of an exogenous or endogenous variable on another endogenous variable. Endogenous variables in SEM serve as outcome variables otherwise, they are referred to as exogenous variables (Hoyle 2012). In the path diagram, only the mean values, variances, and covariances of the exogenous variables are shown as parameters. It is important to specify the variable type, the type of parameters as either fixed or free, and the relationship between the variables (Hoyle 2012, Kline 2023). Free parameters are the variables whose factor loadings are estimated by the model. Fixed parameters, on the other hand, are determined by the researcher and are therefore not estimated. For instance, the factor loading of the $\mathbf{x_1}$ manifest variable, λ_1 can be fixed to 1.

4.2.3 Identification

The primary goal of this step is to translate the hypothesized model into a series of mathematical equations that define the model parameters corresponding to the presumed variable relations (Kline 2023). This is achieved when each parameter in the model assumes a particular value thus, the model is identified when unique values of the population parameters exist and can be estimated. In other words, a model is identified when it is theoretically possible to obtain a unique estimate for every model parameter (Kline 2023). An unidentified parameter occurs when the estimation process does not yield a single value, resulting in the model being re-specified or attempts to analyze them being deemed fruitless. On the contrary, the model is over-identified when the estimation computations produce the same values (Bollen et al. 2010, Hoyle 2012). As shown in Figure 4.1, the model identification process is linked to both the specification and respecification steps as the research may need to ensure that all model parameters are identified before model estimation (Hoyle 2012).

4.2.4 Estimation

The next step, after model specification and identification, is to estimate the model parameters. The goal is to find the best values for the free parameters that minimize the discrepancy between the observed sample variance-covariance matrix denoted by $\mathbf{S}$ and $\boldsymbol{\Sigma}(\hat{\boldsymbol{\Theta}})$, the estimated or model-implied covariance matrix. A good estimator needs to be asymptotic, consistent, unbiased, and efficient. An estimator is termed consistent if it approaches the true parameter as the sample size increases, unbiased when the expected value of the estimate is equal to the parameter estimated, and efficient when there is little variability among its consistent estimators (Hoyle 2012). Maximum likelihood estimation is listed by Hoyle (2012) as the most widely used estimate method and the one that most SEM software defaults to. The first step in model estimating, according to Hoyle (2012), is to determine the components of the observed sample variance-covariance matrix, which offers the fundamental components for model estimation. Individual observed variables can be expressed as a function of unknown parameters and other observed or latent variables in the model. The causal relationships between variables, described by these equations, also referred to as structural equations (Hoyle 2012).

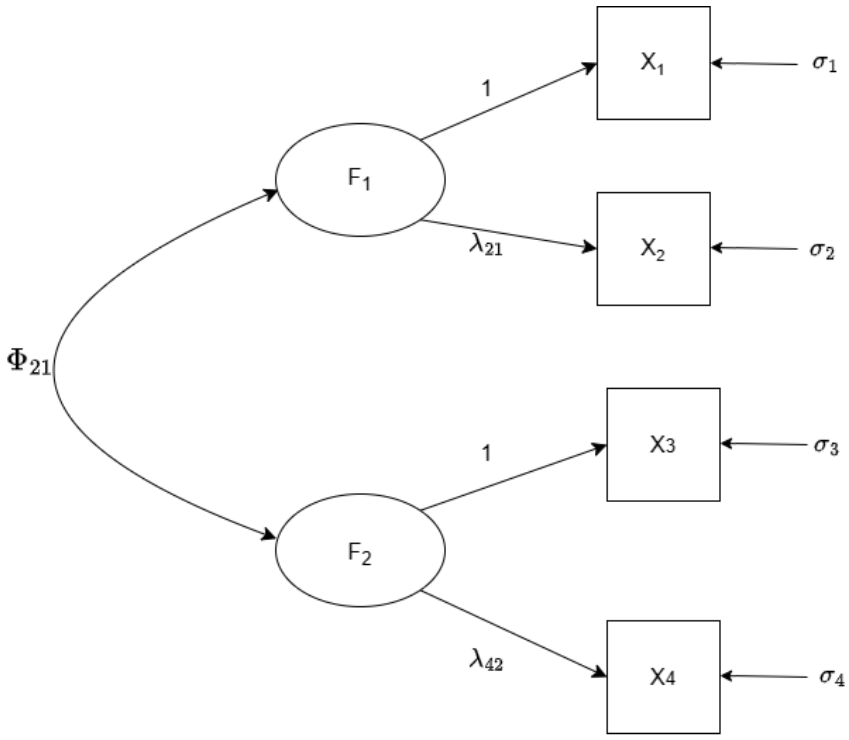

Figure 4.2: An illustration of the SEM path diagram (adapted from: (Hoyle 2012))

As an illustration, the path diagram in Figure 4.2 produces a set of four structural equations displayed in Equation 4.1. The latent, endogenous variable in the path diagram is defined by the $\mathbf{F}_i$ variables, whereas the observable variables are represented by the variable, $\mathbf{X}_i$. The factor loadings or regression weights of the observable variables on the latent factors, are represented by the λ_i. These parameters can be fixed, in this case, some parameters were fixed to 1 so as to the scale for the factors. The regression weight, $\mathbf{\Phi}_i$ represents the covariation between the latent variables F_1 and F_2 and the unspecified measurement error terms are represented by σ_i. The mathematical representation of the observed sample variance-covariance matrix in terms of unknown parameters is demonstrated by Equation 4.1. Thus, the observed variables are represented as a function of the latent factor and an unspecified measurement error.

$$\begin{aligned} X_1 &= 1F_1 + \sigma_1 \\ X_2 &= \lambda_{21}F_1 + \sigma_2 \\ X_3 &= 1F_2 + \sigma_3 \\ X_4 &= \lambda_{42}F_2 + \sigma_4. \end{aligned} \tag{4.1}$$

It is also possible to express Equation 4.1 in matrix form and subsequently in matrix notation

as depicted in Equations 4.2 and 4.3 respectively:

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \lambda_{21} & 0 \\ 0 & 1 \\ 0 & \lambda_{42} \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} + \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \end{pmatrix}, \tag{4.2}$$

$$\mathbf{X} = \mathbf{\Lambda}_X \mathbf{F} + \mathbf{\Sigma}, \tag{4.3}$$

where $\mathbf{X}$ is the vector of observed variables, $\mathbf{\Lambda}_X$, the matrix of factor loading or path coefficients and, $\mathbf{\Sigma}$, the vector of measurement error terms. The next step is to derive $\mathbf{S}$, the observed sample variance-covariance matrix. The matrix (Equation 4.4) shows the lower triangle of the sample variance-covariance matrix, $\mathbf{S}$:

$$\mathbf{S} = \begin{pmatrix} Var(X_1) & - & - & - \\ Cov(X_2, X_1) & Var(X_2) & - & - \\ Cov(X_3, X_1) & Cov(X_3, X_2) & Var(X_3) & - \\ Cov(X_4, X_1) & Cov(X_4, X_2) & Cov(X_4, X_3) & Var(X_4) \end{pmatrix}, \tag{4.4}$$

Assuming that the latent components and measurement errors are independent, the sample variance-covariances can be written in terms of the model parameters based on structural equations using covariance algebra. For example, Equation 4.5 expresses the observed variance parameter, $Var(X_1)$, in terms of its corresponding model parameter, which is the variance of the latent factor, F_1, and σ_1, the error term. The covariance, $Cov(\ F_1, \sigma_1)$ equates to zero since the factor, F_1 and the measurement random error, σ_1 are uncorrelated, hence:

$$\begin{aligned} Var(X_1) &= Var(\ 1F_1\ +\ \sigma_1) \\ &= Var(\ 1F_1) + Var(\sigma_1), + 2Cov(\ 1F_1, \sigma_1), \\ &= Var(F_1) + Var(\sigma_1). \end{aligned} \tag{4.5}$$

$$\begin{aligned} Cov(X_3, X_1) &= Cov(\ 1F_2\ +\ \sigma_3,\ 1F_1\ + \sigma_1) \\ Cov(aX + bY, cW + dV) &= ac\ Cov(X, W) + ad\, Cov(X, V) + bc\, Cov(Y, W) + bd\, Cov(Y, V), \\ &= Cov(F_2, F_1) + Cov(F_2, \sigma_1) + Cov(\sigma_3, F_1) + Cov(\sigma_3, \sigma_1), \\ &= \Phi_{21}. \end{aligned} \tag{4.6}$$

The Equation 4.6 demonstrates that the expression of $Cov(X_3, X_1)$ is equivalent to the model parameter, $\mathbf{\Phi_{21}}$. This is the latent variables and measurement error terms are independent, the

covariances, $Cov(F_2, \sigma_1)$, $Cov(\sigma_3, \sigma_1)$ and $Cov(\sigma_3, F_1)$ are zero. The Equation 4.7 shows the resultant lower triangle of the model-implied variance-covariance matrix represented by $\mathbf{\Sigma}(\mathbf{\Theta})$, where $\mathbf{\Theta}$ denotes all the unknown model parameters.

$$\mathbf{\Sigma}(\mathbf{\Theta}) = \begin{pmatrix} Var(F_1) + Var(\sigma_1) & - & - & - \\ \lambda_{21} Var(F_1) & \lambda_{21}^2 Var(F_1) + Var(\sigma_2) & - & - \\ \Phi_{21} & \lambda_{21}\Phi_{21} & Var(F_2) + Var(\sigma_3) & - \\ \lambda_{42}\Phi_{21} & \lambda_{21}\lambda_{42}\Phi_{21} & \lambda_{42} Var(F_2) & \lambda_{42}^2 Var(F_2) + Var(\sigma_4) \end{pmatrix}. \tag{4.7}$$

Given that $\hat{\mathbf{\Theta}}$ is the model parameter estimation; the goal is to find the estimated model-implied variance-covariance matrix, $\mathbf{\Sigma}(\hat{\mathbf{\Theta}})$ that minimizes the discrepancy with $\mathbf{S}$, the sample covariance matrix. Suppose $\hat{\mathbf{\Theta}}_\mathbf{0}$ is the initial random set of parameters values, the intermediate model-implied variance-covariance matrix can then be calculated by substituting the unknown parameters with the randomly chosen set of parameters values, $\hat{\mathbf{\Theta}}_\mathbf{0}$ to get $\mathbf{\Sigma}(\hat{\mathbf{\Theta}}_\mathbf{0})$. The discrepancy between the observed variance-covariance matrix, $\mathbf{S}$ and the estimated model, $\mathbf{\Sigma}(\hat{\mathbf{\Theta}}_\mathbf{0})$ produces new set of parameters, $\hat{\Theta}_1$. The initial free parameters are then replaced by the new set of free parameters, which in turn creates the new estimated model-implied variance-covariance matrix, $\mathbf{\Sigma}(\hat{\mathbf{\Theta}}_\mathbf{1})$. The computation process is repeated until the differences between adjacent parameter estimates are negligible. After several iterations, the process is said to have converged if the fitting function value obtained in Equation 4.9 cannot be minimized any further (Hoyle 2012).

4.2.5 Evaluation of Model Fit

Evaluating the overall model fit involves determining how closely the model estimates match the observed parameters (Hoyle 2012). Researchers use many tests and fit indices to achieve this goal. However, all model evaluation tests and fit indices are classified as either absolute fit indices or comparative fit indices. Absolute fit indices are functions that provide the test statistic, whereas comparative fit indices evaluate model fit improvements by comparing to a standard baseline model that is logically justified (Hoyle 2012). Although all comparative fit indices are goodness of fit tests, an absolute fit index might be a goodness of fit or badness of fit test, where the test scores declines as the model fit improves (Hoyle 2012).

4.2.5.1 Chi-square (χ^2) or the Likelihood Ratio Test

The main goal of the Chi-square (χ^2) or the likelihood ratio test is to test the null hypothesis that the model-implied covariance matrix is the same as the population covariance matrix (Hoyle 2012). Thus, assuming the observed variables have a multivariate normal distribution, the maximum likelihood estimator is unbiased, asymptotically consistent, and normally distributed, the test statistic, T for the model fit is calculated as shown in Equation 4.8:

$$T = (N-1) \times \mathcal{F}, \tag{4.8}$$

where the N is the sample size and $\mathcal{F}$ is the minimal value of $\hat{F}$, the convergent model generated from the discrepancy fit function for the maximum likelihood, as illustrated in Equation 4.9:

$$\hat{F} = \log|\mathbf{\Sigma}(\hat{\mathbf{\Theta}})| + tr|\mathbf{S}\mathbf{\Sigma}(\hat{\mathbf{\Theta}})^{-1}| - \log|\mathbf{S}| - p, \tag{4.9}$$

where p is the number of observed variables and tr is the trace of the matrix. The test statistic, T follows χ^2 distribution with degrees of freedom equal to the number of unique variances and covariances minus the number of estimated model parameters, under the assumption of multivariate normalcy and a sufficiently large sample size.
Thus,

$$T_{ML} \sim \chi^2_{(df)},$$

where

$$df = \frac{p(p+1)}{2} - t,$$

where p is the number of observed variables being modeled and t is the number of estimated model parameters. The null hypothesis, $\mathbf{\Sigma}(\mathbf{\Theta}) = \mathbf{\Sigma}$ is rejected when the test statistic exceeds the critical values at the given degrees of freedom and the Type 1 error rate, $\alpha = 0.05$ (Hoyle 2012). According to Hoyle (2012), the χ^2 test is the most commonly used method for checking model fitness, despite potentially concealing poor fits and producing less accurate results when small sample sizes are used. An $\chi^2 \backslash df$ ratio test is an alternative ad hoc absolute index fit test that utilizes the ratio between the χ^2 and its associated degrees of freedom. The premise of the $\chi^2 \backslash df$ ratio test is that the expected value of the χ^2 of a correct model approaches the degrees of freedom. Thus, a model is considered a good fit when the ratio test value is less than five (Hoyle 2012). An advantage of using the ratio test is that it penalizes complex models with a large number of parameters as additional parameters reduce the degrees of freedom and hence increase the value of the $\chi^2 \backslash df$ ratio test (Hoyle 2012).

4.2.5.2 Standardized Root Mean Square Residual

The Root Mean Square Residual (RMR) is an absolute fit index that takes the square root of the average squared residuals. In this case, the residuals are defined as the difference between the observed and model-implied variances-covariances. The RMR is a badness-of-fit index that approaches zero with better model fit. However, the scaling can be difficult to discern as it diverges from zero. The RMR has a tendency to be large for covariance matrices with big elements as compared to matrices with smaller elements (Hoyle 2012). The Standardized Root Mean Square Residual (SRMR) was introduced to address this comparison problem. For the SRMR, the residuals are transformed residuals into standard metrics. The SRMR converts the residuals into a standardized metric, where each standardized residual becomes a proportion of the estimated element of **S**. This allows meaningful comparison across models fit to different datasets (Hoyle 2012). The SRMR like the RMR is a badness-of-fit index with a minimum of zero when the model fits perfectly. A model that poorly fits the data is indicated by a SRMR of 1, meaning that the residuals are as large as the elements of **S** that are being estimated.

4.2.5.3 Root Mean Square Error Approximation

The Root Mean Square Error of Approximation (RMSEA) develops a fit index with the non-centrality parameter of the χ^2 distribution. It follows that since the test statistic, $T = (N-1)\times\mathcal{F}$ follows a central χ^2 distribution under the null hypothesis, RMSEA follows asymptotically non-central χ^2 distribution under the alternate hypothesis. Thus, the non-centrality parameter (λ) depends on how poorly the model fits and can be used to create a model fit index (Hoyle 2012). The non-centrality parameter (λ) can be estimated as shown in Equation 4.10 :

$$\hat{\lambda} = (\chi^2 - df)/(N-1), \tag{4.10}$$

Equation 4.11 depicts the estimated $\hat{\lambda}$ bounded at zero to prevent unrealistic negative values of the estimate hence, the bounded λ estimate, $\hat{\lambda}_n$ then would be :

$$\hat{\lambda}_N = max((\chi^2 - df, 0))/(N-1), \tag{4.11}$$

where the the subscript N indicates that $\hat{\lambda}_N$ has been normed to maintain its non-negative value. Similar to the RMR, the RMSEA is a badness-to-fit model bounded at zero. Moreover, the index performs badly and underestimates the model fit with fewer degrees of freedom and smaller samples less than $N = 200$ (Hoyle 2012, Kline 2023). Most SEM software tests retain a confidence interval and a null hypothesis of the true value of $RMSEA \leq$ .05 (Kaplan 2008).

4.2.5.4 Comparative Fit Test

Other alternatives to the χ^2 test consist of comparative fit indices that reflect improvements of the specified model compared to a standard baseline independent model (Hoyle 2012). For instance, the Bentler Comparative Fit Index (CFI) is an incremental fit index goodness of fit statistic, bounded between zero and one (Kline 2023). The CFI compares the amount of departure from close fit for the hypothesized model against that of the baseline model. The CFI formula is shown below:

$$CFI = \frac{max(\chi_0^2 - df_0, 0) - max(\chi_k^2 - df_k, 0)}{max(\chi_0^2 - df_0, 0)}, \tag{4.12}$$

where χ_0^2 and df_0 are the χ^2 test and the degrees of freedom for the baseline model and χ_k^2 and df_k are the are the χ^2 test and the degrees of freedom for the hypothesized model. CFI is limited to the maximum theoretical value of 1. The CFI is a more efficient estimator (lower standard error) due to its truncated distribution, which eliminates values that the population index cannot take on.

4.2.5.5 Tucker Lewis Index (TLI) or Non Normed Fit Index (NNFI)

The TLI or NNFI seeks to control for the df_k and the df_0 from both the hypothesized and baseline models. Although the TLI is related to the CFI, the NNFI imposes much greater relative penalty for complex models than the CFI. The TLI test statistic is formulated by the Equation 4.13 below:

$$TLI = \frac{\chi_0^2/df_0 - \chi_k^2/df_k}{\chi_0^2/df_0 - 1}. \tag{4.13}$$

The TLI statistic is a goodness of fit that is bound between 0 and 1, although it can be negative when the model is extremely misspecified and exceeds 1, indicating a well fitting model (Hoyle 2012).

4.2.5.6 Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC)

A tool for selecting models, the Akaike Information Criterion (AIC) gauges the expected discrepancy between the actual model and the hypothesized model. The model with the smallest AIC is chosen as since it has the least predicted deviations from the true model (Hoyle 2012). When calculating the complexity of the model, the AIC considers the number of free parameters. The AIC requires simple computations and works well with large sample sizes. However, because it ignores the influence of sample size on model selection, it favors complex models when the sample size is small (Hoyle 2012). The BIC is a large-sample approximation of Bayesian model

selection. The model with the smallest BIC is selected. Unlike the AIC, as the sample size rises, the impact of extra parameters on model selection decreases (Hoyle 2012).

4.2.6 Respecification

The researcher might have to respecify the model if the model evaluation process produces negative results that contradict the defined model. Respecification comprises finding the sources of misspecification in the fixed and free parameters of the defined model and going back to the fitting processes of identification, estimate, and evaluation of model fit. Finding significant residuals in the residual matrix is one manual respecification technique (Hoyle 2012). In an effort to enhance model fit, the researcher may frequently try to alter or modify the model's structure by examining the covariance residuals and revising the plausibility of the alternative structures to an empirically based model change may be necessary to achieve this (Hoyle 2012). But findings from empirically based change can be deceptive. Therefore, models derived in this manner must be applied to new data to determine if they can be replicated. According to Kline (2023), there are two areas to focus on during the respecification process. The first area of inspection involves the indicators. It is conceivable for an indicator to lack a significant pattern coefficient for the factor it is intended to measure (Kline 2023). The issue can be resolved by giving an alternative factor for that indicator. Examining the residuals can also help discover the alternative component to which the indicator may be moved (Kline 2023). The second area of inquiry concerns potential adjustments to the factors. The researcher may have incorrectly provided the number of factors. For instance, a large number of associated components indicates low discriminant validity. Poor convergent validity within sets of indicators for the same factors indicates that the model contains too few factors. As a result, altering the number of factors might have a significant impact on the model's performance (Kline 2023).

4.2.7 Interpretation and Reporting

The final stage is the interpretation and reporting of the results. The main goals are to establish the degree of the model's uniqueness, interpret the basic model, and the meaning of the identified parameters (Hoyle 2012). The structural equation model can be graphically represented as path diagrams which are clear and efficient ways to represent complex multivariate relationships (Hoyle 2012).

4.3 Anomaly Detection

As mentioned in Section 2.10.2, outliers or anomalous observations in financial transactions are usually associated with suspicious financial activity since they differ significantly from the rest of the data collection (Hawkins 1980, Sudjianto et al. 2010). This section focuses on introducing the concept of anomaly detection and developing the Isolation Forest (IF) and the Local Outlier Factor (LOF) algorithms as anomaly detection algorithms. Furthermore, Section 4.6 explains the rationale behind the proposed ensemble Local Outlier Factor-Isolation Forest (LOF-IF) algorithm and how the model would be implemented to detect suspicious transactions in migrant remittances.

4.3.1 Distance and Density Outlier Detection

Anomalies can be found by utilizing density and distance techniques, as was briefly discussed in Section 2.10.2. A concise way to describe an outlier is as "few and far" from other observations (Liu et al. 2008, Sudjianto et al. 2010). The foundation of distance-based outlier detection relies on the neighborhood of points or the k- nearest point neighbors (Angiulli et al. 2005). The further a point is from other points, the more it is considered an outlier. Thus, distance-based outliers are defined as a fraction of data objects with a greater distance from the data center (Angiulli et al. 2005). Ramaswamy et al. (2000) proposed a simple and intuitive distance-based definition for outliers, which states that a point, p in a data set is an outlier if less than k points in the data set has a distance of d or less from p. The distance-based outlier detection algorithm calculates and compares the Euclidean distance between the data points. However, the algorithms differ in how this distance is evaluated, as there is no consensus on which sets of points should be used for the distance comparison and how the distance of a collection of points should be evaluated (Mehrotra et al. 2017).

Density-based outlier detection algorithms identify outlying observations based on the density distribution of data points (Zhang & Yang 2023). The guiding premise for density-based outlier algorithms is that data points located in a region of low-density data points as compared to their nearest neighbors are considered outliers. Thus, the density-based model inspects the density of a point's locality and compares it with the density of its associated neighbors (Mehrotra et al. 2017). Although distance and density-based outlier algorithms can effectively detect outliers, they have substantially high computational and storage costs.Distance and density-based outlier models require distance and density measures to be individually computed, which is laborious and time-consuming, particularly for big datasets. Furthermore, direct distances and density estimates are less robust when density variations are present in the data (Mehrotra et al. 2017).

4.3.2 Binary Search Trees

Before introducing the Isolation Forest algorithm, it is important to understand binary search trees, which are the essential building blocks of any Isolation Forest algorithm. A tree is defined by Goodrich et al. (2014) as an abstract data structure that arranges elements hierarchically. According to Garnier & Taylor (2009), trees are linked graphs without any cycles, which means loops or edges connected by the same vertices do not exist. Therefore, a tree, **T** is defined as a set of oval/rectangular nodes storing elements such that each element v of **T** is different from the root node and has a unique parent node (Goodrich et al. 2014). The root node is the special non-empty node without a preceding parent node. It is also crucial to note that an edge of the tree **T** is defined as the connection between the parent and child nodes. A sequence of nodes is called a path. The length of a unique path from its root node to the leaf nodes is known as the level or depth of a tree and the tree height is the maximum level of its nodes (Garnier & Taylor 2009). Figure 4.3 illustrates the tree structure usually depicted with oval or rectangular nodes connected with straight lines or edges in a hierarchical pattern.

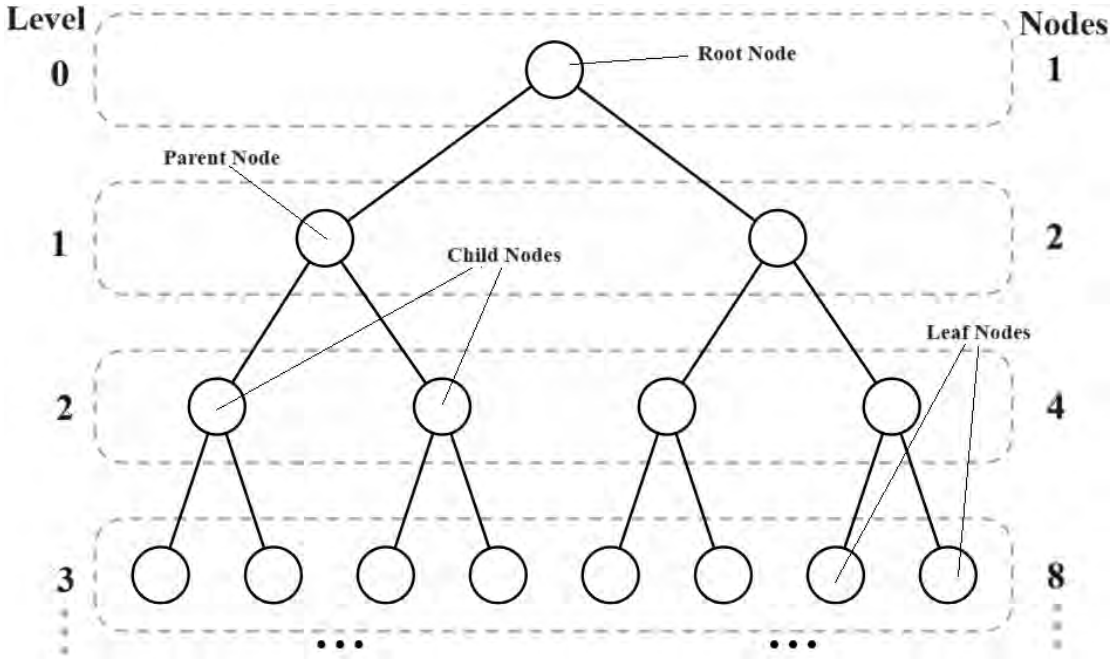

Figure 4.3: An illustration of a binary tree (adapted from: Goodrich et al. (2014))

Now, having the definition of a tree, binary trees are a subset of ordered trees in which each node has at most two child nodes, commonly labeled as the left or right child (Goodrich et al. 2014). The term *ordered* refers to retaining a meaningful linear order among each child node.

4.3.2.1 Properties of Binary Trees

Given the set of all the nodes of a tree $\mathbf{T}$, at a depth of d as shown in Figure 4.3, then the maximum number of nodes at level d is equivalent to 2^d nodes. Thus, the first property of binary trees is that the number of nodes in each tree grows exponentially as the tree levels increase (Goodrich et al. 2014). Now suppose a non-empty binary tree $\mathbf{T}_1$ with n denoting the number of nodes and $\mathbf{n}_E$, $\mathbf{n}_I$ denote the number of external and internal nodes respectively, then $\mathbf{h}$, the height of $\mathbf{T}_1$ which is the number of edges in the longest path from the root to the leaf. Then $\mathbf{T}_1$ has following properties shown below:

$$T_1 = \begin{cases} h + 1 \leq n \leq 2^{h+1} - 1 \\ 1 \leq n_E \leq 2^h \\ h \leq n_I \leq 2^h - 1 \\ \log(n+1) - 1 \leq h \leq n - 1. \end{cases} \tag{4.14}$$

An important application of binary trees arises in binary search trees. Binary search trees are ordered binary trees in which nodal values in the left sub-trees are smaller than the nodal values of the right subtree. Thus, let $\mathbf{S}$ be a set of ordered unique values such as a set of integers, then a binary search tree for the $\mathbf{S}$ would be defined as a binary tree $\mathbf{T}$ such that for each position of $\mathbf{T}$, elements in the left subtree are less than elements stared in the right subtree (Goodrich et al. 2014). Binary search trees are useful in finding values by traversing down the path length starting from the root node. At each position, the search value is compared to the element value stored at the parent node. If the search value is less than the element value, the search proceeds to the left path length in the left subtree. However, if the search value exceeds the element value, the search proceeds the right path length in the right subtree. The search terminates successfully when the search value equals the element value or unsuccessfully when the search ends with an empty set.

4.4 Isolation Forest

Isolation Forests (IF), or *iForest*, developed by Liu et al. (2008), are an ensemble of random binary trees that detect anomalies. These anomalies are incidences with short average path lengths on the Isolation Trees; hence, they are identified as being closer to the root node than a normal point (Liu et al. 2008). Anomaly detection with an IF algorithm is a two-step approach that involves training and evaluation. The training stage generates trees from sub-samples of a specified dataset, while the evaluation stage passes test data instances to compute an anomalous score for each instance. During training, each Isolation Tree is constructed via recursive partitioning without replacing any sub-samples in the dataset, yielding a collection of distinct trees. In the evaluation stage, path length is obtained by counting the number of edges traversed through the Isolation Tree from the root node to the leaf node, and an anomalous score, s is computed (Liu et al. 2008). A a crucial feature of the IF algorithm is it does not require distance-density computations, relied on by most existing methods. This eliminates the major computing expense associated with estimating distance and allows for scalability in large datasets Liu et al. (2008).

4.4.1 Isolation Tree

To understand the Isolation Forest, it is necessary to investigate each component separately. Suppose an Isolation Tree $\mathbf{T}$, has a node, t where t can either be an external node or an internal node with exactly two child nodes (t_l, t_r). A test of an attribute $\mathbf{q}$, with a split value of $\mathbf{p}$ that divides the data points into $t_l \; if \; \mathbf{q} \leq \mathbf{p}$ and t_r otherwise (Liu et al. 2008). Then, for a sample dataset, $\mathbf{X} = \{x_1, ..., x_n\}$, an Isolation Tree is constructed by recursively dividing the dataset, $\mathbf{X}$ with randomly chosen attributes of $\mathbf{q}$ until the tree reaches a desired height, or all the data has been partitioned making the Isolation Tree a proper binary tree with each node having at most two child nodes as explained in the Section 4.3.2. Each instance is separated at the external node of a fully grown tree, hence, the number of n internal nodes is $n-1$ and the total number of nodes is $2n-1$. To quantify the degree of anomaly, the IF sorts data points according to their path lengths (Liu et al. 2008). Now, since the Isolation Tree shares an identical structure to a binary search tree, the estimation of the average height of an external node is equivalent to an unsuccessful search in a binary search tree. The average path length of an unsuccessful search in a binary search tree is given as:

$$c(n) = \begin{cases} 2H(n-1) - 2(\frac{n-1}{n}) \; for \; n > 2 \\ 1 \; for \; n = 2 \\ 0 \; Otherwise, \end{cases} \tag{4.15}$$

where $H(i)$ is the Euler's constant harmonic number used to normalize the comparison of the anomaly score. The Euler's number approximates to $\ln(i)+\gamma$, where the mathematical constant, γ is derived as follows:

$$\gamma = \lim_{n\to\infty} \left(-\log n + \sum_{k=1}^{n} \frac{1}{k} \right). \tag{4.16}$$

We normalise $h(x)$ using $c(n)$,the average of $h(x)$ for a given n. The anomaly score s of an instance x is defined as follows:

$$s(x,n) = 2^{-\frac{E[h(x)]}{c(n)}}, \tag{4.17}$$

and $E(h(x))$ is the expectation or average of $h(x)$ from a collection of iTrees. If the data instances have an s value close to 1, it is identified as an anomaly otherwise, data points that have an anomaly score, $s \leq 0.5$ would be regarded as a normal data point.

4.4.1.1 Properties of an Isolation Tree

An Isolation Forest is an ensemble of Isolation Trees that identify anomalies as points with shorter than average path lengths. Each of the numerous trees is "specialised" to detect distinct anomalies Liu et al. (2008). IF methods perform best with partial or small sub-sampled datasets, unlike existing models that require large datasets for training. Large datasets hinder the IF model's capacity to clearly identify anomalies due to the presence of too many normal cases. This increases the amount of superfluous divisions to isolate each anomaly. Furthermore, evaluating these trees with higher path lengths makes detecting abnormalities more challenging. Therefore, a major aspect of IF algorithms is to generate partial models by sub-sampling. Sub-sampling provides controlled sample size, allowing for improved isolation of outliers. As a result, each Isolation Tree trained on several sub-samples datasets becomes "specialised" in recognising distinct sets of anomalies (Liu et al. 2008).

Figure 4.4 illustrates how sub-sampling affects data instances and improves the isolation of outliers.

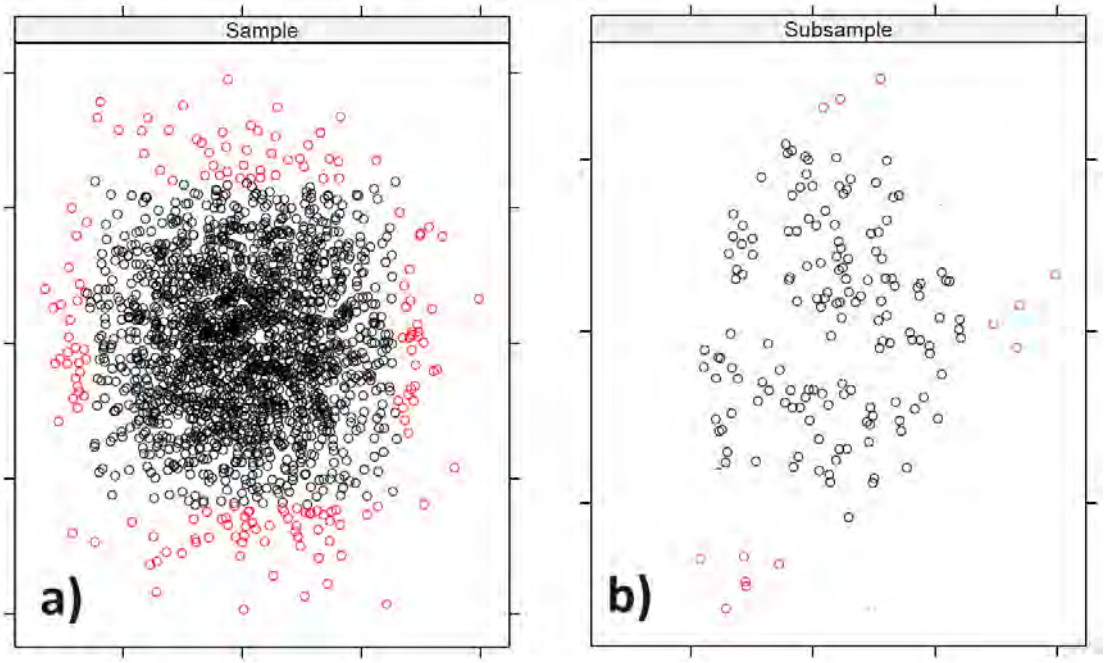

Figure 4.4: The effects of sub-sampling size. Black circles depict the normal points while red circles are the anomalous data points

The original dataset before sub-sampling as shown in Figure 4.4(a) has two anomaly clusters closely packed with the normal points. Reducing the sample size makes the cluster anomalies more distinct and easily identifiable as portrayed in Figure 4.4(b).The normal instances surrounding the two anomaly clusters have been cleared out, and the size of anomaly clusters becomes smaller and easier to identify. Liu et al. (2008) study empirically found that setting $\phi = 2^8 = 256$ is sufficient for detecting outliers in various data settings. Furthermore, the number of trees t, which controls the size of the iForest, converges at $t = 100$.

4.4.2 Training and Evaluation of an Isolation Forest

When training an IF algorithm, individual Isolation Trees are constructed by repeatedly partitioning the training data instances until all data points are isolated or a specific height has been reached. It is important to note that the tree height limit, l is set by the sub-sampling size, ϕ as shown in Equation 4.18

$$l = ceiling(log_2(\phi)), \tag{4.18}$$

which is approximately the average or expected tree height. Data instances with shorter-than-average path lengths are more likely to be outliers of interest. The main advantage of the IF

algorithm is in its ability to identify anomalies without partitioning the full dataset, thus, building models using a small sample size. At the end of the training process, a collection of Isolation Trees is returned for the evaluation stage.

In the evaluation stage, using the R software, a single path length, $h(x)$ is derived by counting the number of edges, e from the root node to the termination node as the instance, x traverses through an Isolation Tree. When $h(x)$ is obtained from each tree of the ensemble, the expected path length is computed and subsequently the anomaly score, $s(x, \phi)$ is produced using Equation 4.17.

4.5 Local Outlier Factor

The LOF model is a density based, unsupervised outlier detection algorithm that compares the local density of a point to the local density of its k-nearest neighbours (Breunig et al. 2000, Johannesen et al. 2023). Though IF models are capable of efficiently identifying global outliers in the dataset, they perform poorly in detecting local outliers. Local outliers refer to how isolated a data point is relative to its surrounding neighborhood, hence a restricted neighborhood is considered of each object (Breunig et al. 2000). In other words, the global view of outliers may hold under certain conditions but is unsatisfactory for cases that involve clusters with varying densities as shown in Figure 4.5.

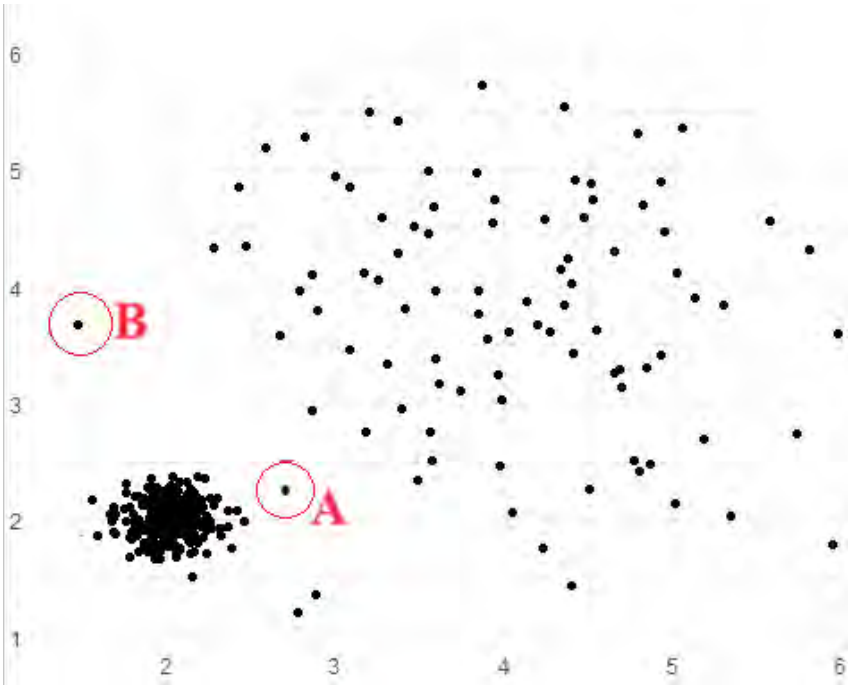

Figure 4.5: An illustration showing global and local outliers in a dataset with varying cluster densities

By definition, point **B** in Figure 4.5 is considered a global outlier as it is well isolated from the entire data. However, due to the uneven density distribution of the data points, point **A** is referred to as a local outlier relative to its proximity to the smaller denser cluster (Breunig et al. 2000). Despite the aforementioned problem, the LOF algorithm can detect outliers in data clusters with varying densities. The LOF algorithm borrows from the k-Nearest Neighbour (kNN) algorithm. It examines the uniqueness of each data instance based on its distance from its k-nearest neighbours. The LOF method can find outliers regardless of data distribution, as it makes no assumptions about distributions. The key idea is that the density around an outlier object differs greatly from that of its neighbours. Furthermore, LOF is considered an unsupervised outlier detection method, making it applicable for the dataset used in this research Auskalnis et al. (2018). Before defining the LOF algorithm, it is important to define the k-distance, k-distance neighborhood, the reachability distance, and the local reachability density of an object.

4.5.1 k-Distance Neighbourhood

Let $d(p, O))$ denote the distance between two points, p and O in a dataset, D be calculated in the Euclidean space as:

$$d(p, O) = \left(\Sigma_{i=1}^{n}(p_i - O_i)^2\right)^{\frac{1}{2}}. \tag{4.19}$$

For a positive integer, k, the k-distance of p is the distances between p and the farthest neighbor data point O, satisfying the following conditions:

- At least, for k data points, $\hat{O} \in D|p$ maintains $d(p, \hat{O}) \leq d(p, O)$;
- At most, $k-1$ data points, $\hat{O} \in D|p$ maintains $d(p, \hat{O}) < d(p, O)$.

Thus, given the k-distance of p, the k-distance neighborhood of an object p contains in its radius every object whose distance from p is not greater than the k-distance, hence:

$$N_{k-distance(p)}(P) \; or \; N_k(p) = \{q \in D \backslash (p) | d(p, q) \leq k - distance(p)\}$$

. The objects q is k-nearest neighbor of p (Breunig et al. 2000, Alghushairy et al. 2020).

4.5.2 Reachability Distance

The reachability distance ($reach-dist$) of an object p with regard to the data point O, is defined as:

$$reach - dist_k(p, O) = max\{k - distance(O), d(p, O)\} \tag{4.20}$$

Figure 4.6 shows an illustration of reachability distance with $k = 4$.

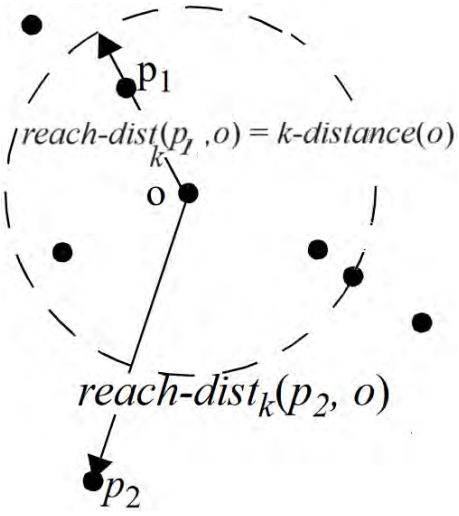

Figure 4.6: An illustration of the reachability distance with k=4 (adapted from: Breunig et al. (2000))

When the object p is significantly far from O, the reachability distance is measured as the actual distance between the data points, p_2 and O, depicting the distance $reach - dist_k(p_2, O)$. However, if the object p is sufficiently close to O or located within the k-distance radius, then the k-distance of O becomes the reachability distance (Breunig et al. 2000, Alghushairy et al. 2020).

4.5.3 Local Reachability Density

The concept of density is defined by two parameters: $MinPts$ - the number of nearest neighbors used in defining the local neighborhood of the object or k nearest objects and the volume parameter. These are crucial in determining the density threshold for a density-based clustering algorithm. Therefore, to detect density-based outliers, it is necessary to compare the different cluster densities. Breunig et al. (2000) used the N_{MinPts} which is the nearest neighborhood of p and the $reach - dist_{MinPts}(p, O)$ for $O \in N_{MinPts}(p)$ as a measure of the volume to determine the density in the neighborhood of an object p.

Thus, the local reachability density of object p is defined as:

$$lrd_{MinPts(p)} = \left[\sum_{O \in N_{MinPts}(p)} \left(\frac{reach - dist_{MinPts(p,O)}}{|N_{MinPts}|} \right) \right]^{-1} . \tag{4.21}$$

The local reachability density of an object p is the inverse of its average reachability distance based on $MinPts$ nearest neighbours (Breunig et al. 2000, Johannesen et al. 2023). A high local reachability density suggests a dense neighbourhood, while a low local reachability density denotes a sparse neighbourhood (Johannesen et al. 2023). Furthermore, when the total of the reachability distances approaches zero, it is important to note that the local density becomes infinite. Thus, the local outlier factor of p is the average of the ratio between the local reachability density of p and its $MinPts$ - nearest neighbors and is defined as:

$$LOF_{MinPts}(p) = \frac{1}{|N_{MinPts}(p)|} \left(\sum_{n \in N_{MinPts}(p)} \frac{lrd_{MinPts(O)}}{lrd_{MinPts(p)}} \right) . \tag{4.22}$$

The outlier factor of the data point p represents the extent to which p is an outlier. It is the average of the ratio of the local reachability density of p and the p's nearest neighbors, $MinPts$. If the local outlier factor for a data point exceeds 1, then it is considered an outlier (Breunig et al. 2000).

4.6 Proposed Outlier Detection Algorithm

The anomaly detection algorithm used for the detection of suspicious migrant remittances is an ensemble model developed from the combination of the Isolation Forest and the Local Outlier Factor (LOF-IF). As highlighted in Section 4.5, the Isolation Forest algorithm is sensitive to detecting global or extreme outliers but performs poorly in identifying local outliers in data clusters of varying densities. In contrast, the LOF algorithm proves to be robust in detecting such local outliers at the expense of a high computational cost when compared to the IF. Combining the two algorithms simultaneously improves the ensemble model's performance and lowers the time complexity (Cheng et al. 2019). This ensemble algorithm has been previously utilized in existing literature. Wang & Xu (2019) combine the IF and the LOF to improve the detection of anomalies found in concrete mixtures. The paper proposed an IF algorithm based on a sliding window technique for the Local Outlier Factor. The sliding window creates a window size data storage that stores the data points computed from the IF model. The LOF algorithm then uses a threshold to calculate the outlier score from the input data obtained from the sliding window. Data points exceeding the threshold value would be considered outliers. Instead of using the

sliding window technique, the proposed ensemble algorithm implemented in this study makes use of the pruning technique implemented by Cheng et al. (2019) as illustrated in Figure 4.7. This

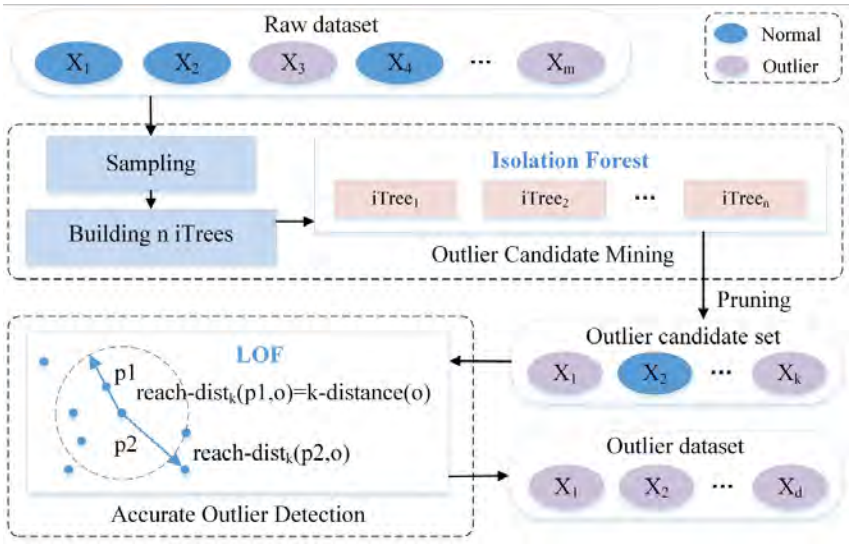

Figure 4.7: An illustration of the LOF-IF algorithm workflow (adapted from: Cheng et al. (2019)

approach involves three stages: mining, pruning, and detection (Cheng et al. 2019). The first stage involves building and implementing the IF. The IF algorithm lowers a huge raw dataset by recognizing and isolating global outliers from the rest, resulting in an outlier candidate set. Finally, using the LOF algorithm, the local outlier factor value for each data point in the outlier candidate set is determined and the top n points with the highest LOF values will be selected. Suppose a dataset $D = \{d_1, ...d_n\}$ has n samples and d_i is an attribute in D : $d_i = \{x_1, ..., x_n\}$ where x_j is the value of an attribute in d_i. The outlier coefficient of an attribute C_{d_i} is defined as:

$$C_{d_i} = \sqrt{\frac{(x_j - \bar{x})^2}{n\bar{x}^2}}, \tag{4.23}$$

where $\bar{x}$ is the mean of the d_i attribute and C_{d_i} is the dispersion measure of attribute d_i. After calculating the individual C_{d_i} and obtaining the outlier vector, $\mathbf{D}_C = \{C_{d_1}, C_{d_2}, ..., C_{d_n}\}$, the trim threshold, Θ_D which represents the ratio of outliers present in the dataset is determined by Equation 4.24:

$$\Theta_D = \frac{\alpha Top\_m(D_C)}{m}, \tag{4.24}$$

where α is the adjustment factor and Top_m is the number of top outliers retained and refers to the m values with a large dispersion coefficient after sorting.

4.7 Model Assessment

It is critical to evaluate the model's performance and ensure that it adequately represents the data. According to Hastie et al. (2009), model performance assessment measures the quality of the selected model. It is vital to analyse the performance of the SEM and LOF-IF models.

Model validation can be generalised as separating the dataset into two parts: the training and testing data. The training data is used to create and improve the model. In contrast, the testing data would be used to determine how well the model performed in the presence of previously unseen data. The model's prediction error or cost is calculated using the mean sum of squares (MSE), which is calculated as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{m_t}(y_i - \hat{f}(X_i))^2, \tag{4.25}$$

where y_i is the ith response variable and $\hat{f}(X_i)$ is the prediction function at the ith instant (Witten & James 2013). Small values of the MSE indicate that the model-predicted responses are significantly close to the true responses in the dataset. The primary interest of model validation is in the accuracy of the model predictions when applied to unseen testing data (Witten & James 2013). Similarly, cross-validation is defined as a process of assessing the ability of predictive models to generalize real-world data. In other words, cross-validation assesses a model's prediction capability (Berrar 2019, Hastie et al. 2009).

4.8 Cross Validating Classification Problems

The application of cross-validation can also be extended to classification algorithms that possess a qualitative response variable (Witten & James 2013). For such cases, the number of misclassifications is used to determine the error rate, which is

$$CV_n = \frac{1}{n}\sum I(y_i \neq \hat{y}_i). \tag{4.26}$$

The process is repeated until all data folds are utilized as test data and an average Mean Sum of Error Squares or the cost function, is calculated. According to a similar study by John & Naaz (2019), model accuracy measures how well a model predicts outcomes and is the primary evaluation metric for classification tasks for supervised datasets. However, it can be misleading for imbalanced or skewed datasets. Thus, model precision, recall, and F_1 score would be determined. Precision rate is the ratio of correctly predicted true positive observations and

the positive observations, hence :

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}. \tag{4.27}$$

High precision indicates the model is performing well in avoiding false positives.On the other hand, recall rate is the ratio of the predicted true positives and the total positive predictions, which is:

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negative}. \tag{4.28}$$

Recall measures the model's ability to detect positive observations, thus a high recall indicates the model's effectiveness in detecting suspicious cases.

The F_β measure is a trade-off between the precision and recall metrics Taha & Hanbury (2015). F_β measure is defined as:

$$F_\beta\ score = \frac{(\beta^2 + 1)\ Precision \times\ Recall}{\beta^2\ Precision + Recall}. \tag{4.29}$$

When the precision and recall scores are equally important, that is $\beta = 1$, then the F_β measure score becomes a F_1 score which is also known as the harmonic mean (Taha & Hanbury 2015). The F_1 score is given by

$$F_1\ score = \frac{2\ (Precision \times\ Recall)}{Precision + Recall}. \tag{4.30}$$

Another key measure to consider is sensitivity, specificity, and p-value. Sensitivity rate refers to the amount of successfully predicted positive records, whereas specificity refers to the number of correctly anticipated negative records (Swift 2020 sensitivity). These settings determine the model's ability to accurately recognise suspicious and legitimate transactions. The p-value is the probability of detecting a value of an extreme test statistic than the observed value if the null hypothesis is true (Bebchuk & Wittes 2012). The p-values are important since they indicate confidence in the model's performance. A smaller p-value indicates greater confidence in the model. Traditional cross-validation techniques apply to supervised models where the presence of a dependent variable allows testing for model accuracy and predictive power of the developed model. However, the lack of ground truth data makes it challenging to determine accuracy metrics for unsupervised machine learning models. As a result, model stability and resilience must take precedence over precision and accuracy.

4.9 Traditional Rule-Based Methods

In the absence of a response variable and a lack of cross-validation techniques for unsupervised datasets, a traditional Rule-Based Method (consisting of a set of rules and thresholds) was developed and implemented to classify suspicious transactions in the dataset and generate a reference response variable. Traditional rule-based systems use fixed rules and criteria to detect suspicious transactions (Jose-de Jesus et al. 2021). They have proven to be simple and easy to deploy, but also tend generate a large number of false positives since they are unable to dynamically adapt to changes in criminal behavior (Jose-de Jesus et al. 2021). The rules and thresholds for this study will be based on the standards set by the FAFT FAFT (2021), and FIC (FIC 2019). It is crucial to compare the proposed model results to those obtained using traditional Rule-Based approach as it is indicative of the model's adherence to the stipulated financial guidelines provided by FAFT and FIC. Furthermore, the goal is to create a base model that uses predefined standards to detect suspicious transactions, preferably those related to terrorist financing. The standard result would be used to assess the LOF-IF algorithm performance. Since there is a limited dataset, only the most relevant guidelines are considered. The traditional Rule-Based Method was developed using decision trees as they closely resemble the logic of a rule-based decision system.

4.9.1 Decision Trees

Decision trees can be described as sequential models that logically combine a sequence of tests (Kotsiantis 2013). Finding a model that predicts the value of a target-dependent variable from several input variables forms the core of the decision tree algorithm (Kotsiantis 2013). Similar to binary trees explained in Section 4.3.2, decisions trees are built by dividing the original dataset into subsets. Each split partitions the sample into two or more sections, with each subset of the partition with one or more classes in it. A set of splitting rules based on classification features forms the basis of the splitting criteria (Kotsiantis 2013). The process is performed recursively on each derived subset, a technique known as recursive partitioning. The process is complete when the subset at a node contains all of the same target variable values or when splitting no longer adds value to the predictions.

4.9.1.1 Gini Impurity

The Gini impurity is one of the proprties employed of decision tree models. When an element of a set is randomly selected and labelled independently based on the distribution of labels in the set, the Gini impurity quantifies the risk of the element being incorrectly labelled (Kotsiantis 2013). For a set of items with k classes, the probability of choosing an item with a label i is

p_i, and the probability of miscategorizing that item is $\sum_{k \neq i} p_k = 1 - p_i$. The Gini impurity is computed by summing pairwise products of these probabilities for each class label (Kotsiantis 2013):

$$I_G(p) = 1 - \sum_{i=1}^{J} p_i^2. \tag{4.31}$$

4.9.1.2 Information Gain

Another important property of decision trees is the information gain. Kotsiantis (2013) defines information gain as the amount of knowledge learned about one random variable given another random variable. Given that the entropy of a random variable is the average quantity of information or uncertainty associated with the variable's potential outcomes. The expected information gain is the reduction in information entropy, H from a prior state to a state that requires some information.

$$IG(T, a) = H(T) - H(T|a), \tag{4.32}$$

where $H(T|a)$ is the conditional entropy of T given the value of the attribute a (Kotsiantis 2013).

4.10 Chapter Summary

This chapter discussed the research methodology applied. Section 4.2 explored the steps in developing and implementing structural equation models. Section 4.3 discussed the concept of anomaly detection and its properties. Lastly, model validation and assessment are explored and discussed in Section 4.7.

Chapter 5

Results

5.1 Chapter Introduction

This chapter presents the results obtained from the statistical application of the methodology described in Chapter 4. Section 5.2 provides the model conceptualization and analysis of the SEM results. Section 5.3 presents the model performance results and analysis of the proposed ensemble LOF-IF algorithm. Finally, the chapter ends with a findings summary in Section 5.4.

5.2 Structural Equation Model

The structural equation model was developed from the available dataset using R programming. The variables used are mainly related to the recipient and their country of origin. The model conceptualization for this study as explained in Chapter 4 is shown in the section below.

5.2.1 Model Conceptualization

There are a lot of factors that can be considered when assessing complex relationships. These include measures of political instability, government corruption, gross domestic product, death per capita due to terrorism, and the human inequality index to mention a few. Nonetheless, the model framework used to develop the SEM stems from the Agnew (2010) general strain theory. The theory states that the risk of terrorism is determined by the collective strain experienced by a community. Therefore, three latent variables - economic, geographical, and social factors have been identified as factors that are indicative of societal collective strain. The proposed SEM model conceptualization and formulation is in line with the findings of the United Nations Development Programme survey reports on the spread of violent extremism (UNDP 2017, 2023).

5.2.2 Economic Factors

The UNDP (2017) survey report shows the relevance of economic factors as significant indicators of terrorism risks. The report expressed concerns associated with an upbringing in a poverty-stricken environment. Unemployment is noted as a source of frustration and the main reason youths join violent extremist groups. Furthermore, the report also claims that the need for employment is the most frequent reason cited by 80% of the respondents joining an extremist organization after their first contact. Based on UNDP (2017) and UNDP (2023) reports, the lack of economic inclusion in the community increases the risk of terrorist activity. Economic inclusion can be measured by the recipient's access to financial services such as bank accounts, loans, and savings accounts.

5.2.3 Geographical Factors

The UNDP (2017) and UNDP (2023) reports also identify geographical factors as major collective strains influencing particular individuals' participation in terrorist activities. The report claims that the "accident of geography" or childhood experiences linked to marginalized and peripheral regions, shape an individual's global perception and vulnerabilities. However, lack of parental involvement is cited as a critical factor that correlates to childhood dissatisfaction and participation in violent extremism. The measurement variables in the dataset include the categorical variable place of character, which denotes whether the recipient resides in a rural or urban setup. The accessibility to basic amenities, water, and electricity were also included as measurement variables for the geographical latent variable.

5.2.4 Social Factors

Lastly, social inequalities factor contribute to the spread of violent terrorism (UNDP 2017, 2023). These include wealth and income differences, educational differences, digital and technological exclusion, and housing inequalities. Education is a key factor in predicting an individual's likelihood of being recruited by a terrorist organization. The UNDP (2017) notes individuals deprived of basic education and literacy were more likely to join an extremist group. Higher levels of education strengthen an individual's resolve against extremist ideologies. Although 51% of the study's respondents noted religion as a reason for joining a terrorist organization yet most of the respondents had little to no understanding of religious texts (UNDP 2017, 2023). Among the dataset variables, the recipient's education, the recipient's age, and the recipient's household size are the measurement variables for the social factor latent variable.

5.2.5 Model Estimation and Fit

The resultant path diagram is shown in Figure 5.1:

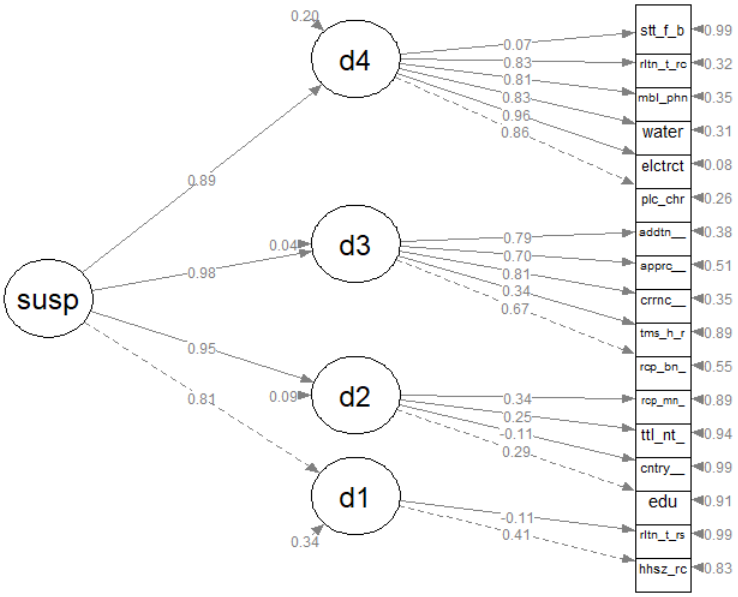

Figure 5.1: The path diagram for the Structural Equation Model

The latent factor - **susp** shown in the path diagram, depicts the suspicion level of terrorism associated with the recipient transaction. This latent variable is directly influenced by four endogenous variables $\mathbf{d_1}, \mathbf{d_2}, \mathbf{d_3}, \mathbf{d_4}$, which represent the socio-economic factors as explained in the model conceptualization. The latent variables $\mathbf{d_1}$ and $\mathbf{d_2}$ represent the socio-economic factors measured by the independent exogenous variables: the recipient household size, education level, total income earned, relationship to the respondent and country of study. The economic latent factor variable, $\mathbf{d_3}$ has exogenous measurement variables: recipient bank account, the frequency of remittances, the currency of the remittances received, and additional costs associated with the financial transfer. These exogenous variables measure the economic latent factor strain as it determines the recipient's access to financial services. The latent factor $\mathbf{d_4}$ indicates the geographical factors contain measurement variables which are place of character, (rural/ urban), and access to basic amenities (electricity and water).

5.2.6 Model Interpretation and Reporting

Table 5.1 summarizes the SEM model fitness results as discussed in Chapter 4.

Table 5.1: Table of SEM results

	Model Test User Model:		Model Test Baseline Model:	
	Standard	Scaled	Standard	Scaled
Test Statistic	1004.794	159.313	8684.918	832.912
Degree of freedom	115	115	136	136
p-value (Chi-square)	0.000	0.004	0.000	0.000
Scaling correction factor		14.583		11.997
Shift parameter		90.413		
Comparative Fit Index (CFI)		0.896		0.936
Tucker-Lewis Index (TLI)		0.877		0.925
Robust Comparative Fit Index (CFI)		0.923		0.909
Robust Tucker-Lewis Index (TLI)				
Loglikelihood and Information Criteria:				
Loglikelihood user model (H_0)	-39547.819	-39547.819	-39547.819	-39547.819
Loglikelihood unrestricted model (H_0)	-39045.423	-39045.423	-39045.423	-39045.423
Akaike (AIC)	79171.639	79171.639	79171.639	79171.639
Bayesian (BIC)	79361.304	79361.304	79361.304	79361.304
Sample-size adjusted Bayesian (SABIC)	79240.607	79240.607	79240.607	79240.607
Root Mean Square Error of Approximation:				
RMSEA	0.084	0.084	0.084	0.084
90 Percent confidence interval - Lower	0.080	0.011	0.080	0.011
90 Percent confidence interval - Upper	0.089	0.026	0.089	0.026
p-value $H_0 : RMSEA \leq 0.050$	0.000	1.000	0.000	1.000
p-value $H_0 : RMSEA \geq 0.080$	0.935	0.000	0.935	0.000
Robust RMSEA		0.072		0.072
90 Percent confidence interval - Lower		0.042		0.042
90 Percent confidence interval - Upper		0.098		0.098
p-value $H_0 : RMSEA \leq 0.050$		0.103		0.103
p-value $H_0 : RMSEA \geq 0.080$		0.320		0.320
Standardized Root Mean Square Residual:				
SRMR	0.055	0.055	0.055	0.055

After 138 iterations, the model converged to a theoretical model fit. The results displayed in Table 5.1 show a chi-square test statistic of 159.31 with 115 degrees of freedom indicating an over-identified model. The low p-value suggests there is little evidence to suggest the model does not properly fit the data. The goodness of fit indices, Comparative Fit Index - 0.896 and a Tucker-Lewis Index - 0.877 are significantly high and well above 85%, suggesting a fairly good model fit. Moreover, the Root Mean Square Error of Approximation, RMSEA is 0.084 which is significantly low, depicting a good model fit. The Standardized Root Mean Square Residual (SRMR) also indicates a good model fit.

There is a high positive correlation between socio-economic latent variables $\mathbf{d_1}$ and $\mathbf{d_2}$ and the measurement variables, recipient household size (0.41), education (0.29), the number of times the recipient receives their remittances, and the recipient's main occupation (0.34). However, all the residual terms for socio-economic predictors are very high, which suggests the predictor variables are poor indicators for the socio-economic latent variables, $\mathbf{d_1}$ and $\mathbf{d_2}$. The geographical and economic latent factors $\mathbf{d_3}$ and $\mathbf{d_4}$ have a positive direct influence on the level of terrorism risk with factor loadings (0.98 for $\mathbf{d_3}$ and 0.89 for $\mathbf{d_4}$).

All indicator variables for these latent variables also have significant positive factor loadings except for the measurement variable - state of birth. In addition, the small residual value indicates that much of the variation is explained by the measurement variable. However, only the state of birth measurement variable has a weak factor loading of 0.07 and a high residual value of 0.99, indicating an unsuitable variable that has a weak contribution to the model variation. In addition, the results show the log-likelihood measurement of the user-specified and the unrestricted model. The lower negative number suggests the user-specified model is a better fit for the data. The above model also depicts the Root Mean Square of Approximation as 0.084, suggesting the model fits the data with a 90% confidence interval of (0.080: 0.089). The SRMR of 0.055 indicates a good model fit. The results show that when geographical and economic latent factors are significant, there is a high chance that migrant remittance can be misappropriated for terrorist financing.

5.3 Model Results

The proposed LOF-IF algorithm was successfully implemented, and the cross-validation results were presented in this section. The sample size for the IF method was set to 256, with a k number of nearest neighbours of 5. Due to the dataset's small size, the local outlier factor values were determined on the outlier candidate set without the process of pruning, as indicated in Section 4.6. Figure 5.2 shows the results of implementing three models (LOF-IF, SEM, and Traditional Ruled-Based) using R programming. The R code for all of the models is included in the Appendix.

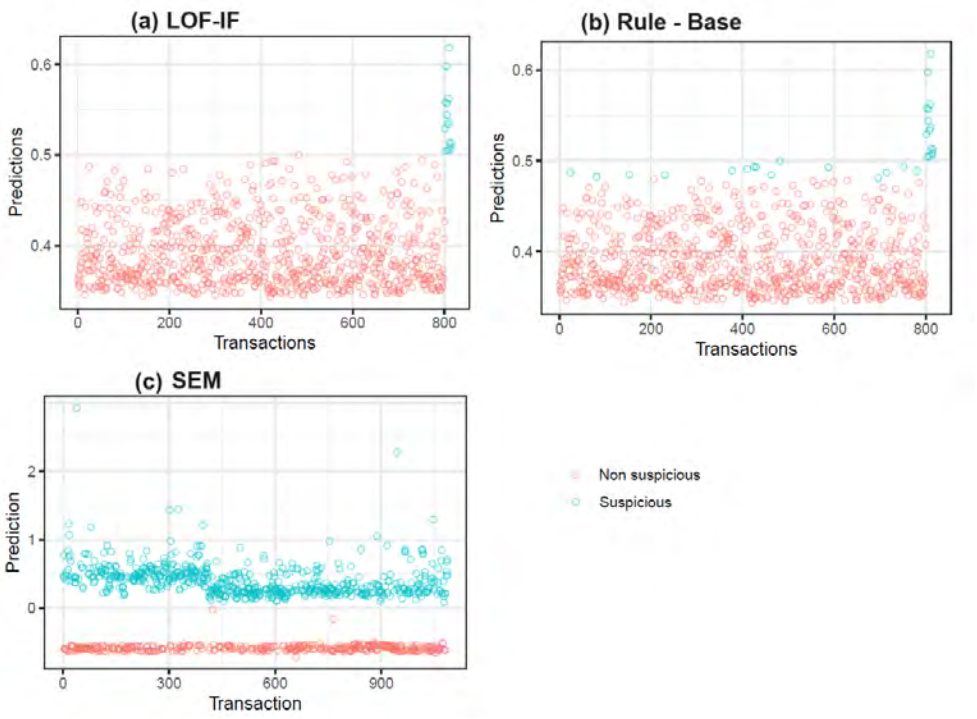

Figure 5.2: The scatter plots results obtained from the implementation of the (a) LOF-IF, (b) Rule-Based model and (c) SEM

After running all three models and plotting their predictions against the individual transactions, Figure 5.2 shows intriguing patterns in the scatter plots. The LOF-IF model identified the bulk of transactions as normal transactions. Only a tiny blue cluster of points in Figure 5.2(a) were detected as suspicious transactions. However, the SEM model in Figure 5.2(c) detected a substantial high number of suspicious transactions which is suggestive of a poor-performing model, likely to generate an unacceptable number of false positives. As expected, the traditional Rule-Based Model was more conservative and detected more suspicious transactions than the LOF-IF model. This is attributed to classification mostly done on transaction attributes. Moreover, this suggests the ensemble model is capable of detecting suspicious transactions as defined by the FIC and FAFT standard. Table 5.2 shows a summary of the classification results obtained from

all the three models.

Table 5.2: Classification results of migrant remittances with different algorithms

	LOF-IF	SEM	Rule-Based Model
Normal Transactions	1074	613	1057
Suspicious Transactions	16	474	30

Using the traditional Rule-Based model's predictions as an estimated response variable, it was possible to perform a 10 fold cross validation of the LOF-IF model. The training data constituted 75% of the dataset and the remaining data instances formed elements of the testing data. Tables 5.3 shows the confusion matrix of the LOF-IF algorithm.

Table 5.3: Confusion matrix of the LOF-IF algorithm

	Reference	
Prediction	Normal Transactions	Suspicious Transactions
Normal Transactions	268	1
Suspicious Transactions	0	2

Table 5.4 show the results obtained from the 10-fold cross-validation of the LOF-IF and the Rule-Based models using R programming code.

Table 5.4: The results obtained from the k-fold cross-validation for the LOF-IF algorithm and the Rule-Based Model

	LOF-IF	Rule-Based Model
Accuracy	0.9963	09890
95% Confidence Interval	(0.9796, 0.9999)	
p-value [Acc >NIR]	0.1975	0.0000
Sensitivity	1.0000	0.9954
Specificity	0.6667	0.9643
Pos Pred Value	0.9963	0.9908
Neg Pred Value	1.000	0.9818
Recall Rate	0.9909	0.9643
F_1 Score	0.9954	0.9931

It is equally important to evaluate the results of the Rule-Based model. As stated in Section 4.9, the model is based on a decision tree algorithm. The model had five variables at first as attributes. However, after running the model in R, three attributes were pruned from the model. Thus, the remaining attributed were: the migrant's total net income (*total_net_income*) and the total value of remittances sent within 12 months (*total_value_remitmoney*). The *total_net_income* variable is a categorical variable that grouped migrant's salaries and wages earned into nine different classes in ascending order. The numeric variable, *total_value_remitmoney* depicted the actual value of remittances sent within 12 months. The FIC currency transaction report standard recommends any transaction above R50 000.00 which is equivalent to US$ 2 500 must be reported. For the Rule Based model, the currency transaction threshold of US$ 5000 was implemented. Figure 5.3 depicts the decision tree for the Rule-Based model.

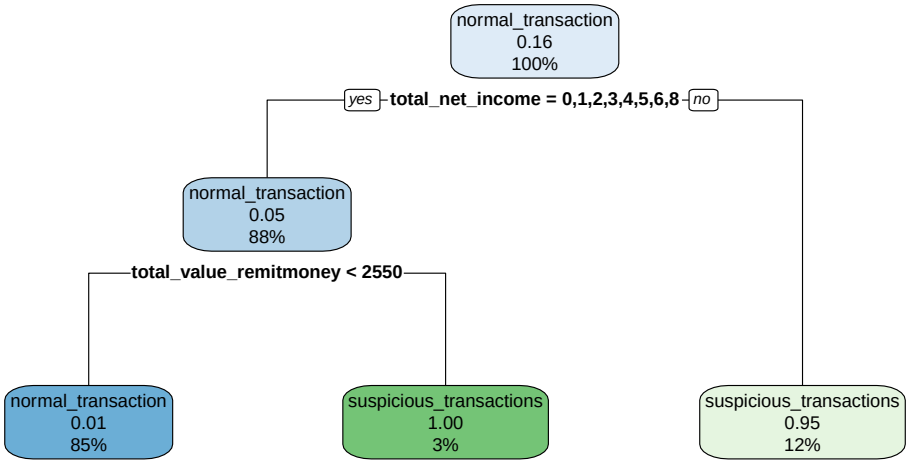

Figure 5.3: The Rule-Based model's decision tree

5.3.1 Results Summary

The LOF-IF algorithm performed remarkably well, successfully classifying most of the unseen test data and producing a recall rate of 0.9909 and an F_1 score of 0.9954. This shows that the proposed model can identify suspicious transactions as described by the Financial Intelligence Centre and the Financial Action Task Force (FIC 2019, FAFT 2021). In addition, the very high recall rates of both models indicate the model's effectiveness in detecting suspicious cases. It is also worth noting that the results obtained in this study align with existing studies that employed

the LOF-IF algorithm (Cheng et al. 2019, John & Naaz 2019). Comparing the ensemble model's performance with the Rule-Based Model results, both models have an identical performance. They both have a very positive prediction values or precision rates indicating both models can detect false positive effectively which is an ideal model trait. However, the p-value of 0.1975 for the LOF-IF model is unfavorable. Nonetheless, the p-value is expected to improve as more data is made available.

5.4 Chapter Summary

In this chapter, both the SEM model and the LOF-IF algorithm were successfully developed as described in the methodology. In evaluating the results, the LOF-IF algorithm demonstrated exceptional performance. With over 40% of the transactions being categorized as suspicious, the SEM model retained a significant proportion of suspicious transactions. This suggests that the SEM model has a large rate of false positives, which runs counter to the expectations of an imbalanced dataset.

Chapter 6

Conclusions and Recommendations

6.1 Conclusion

This study has provided ample evidence of the substantial contribution that migrant remittance transfers make to the socioeconomic development of low-income earning nations, as well as the associated risks of money laundering and terrorist financing. For these reasons, it is critical to maintain stringent monitoring and promptly detect any suspicious financial transfers. The conventional rule-based manual techniques have become ineffective, inflexible, and prone to false alarms. This project determined whether unsupervised machine learning approaches could effectively be used to identify TF and suspicious financial transactions in migrant remittances.

The proposed machine learning approaches achieved this goal by modeling latent features related with terrorism finance and employing an outlier detection strategy. As a result, a SEM model and an ensemble LOF-IF algorithm were selected as appropriate models. The results obtained from the SEM were consistent with the expectations of the general theory of collective strain reported in Agnew (2010). However, when it came to detecting suspicious transactions, the model performed below expectations as it labeled a large number of transactions as suspicious, indicating a high false positive rate. This can be attributed to the presences of outliers in the dataset which is a violation of one of the main assumptions for the SEM models as discussed in Section 4.2.

Finally, the outlier detection algorithm was created using an ensemble of two machine learning algorithms: Isolation Forest (IF) and Local Outlier Factor (LOF). Since the dataset was unsupervised, a traditional Rule-Based model was developed using the norms and recommendations established by FAFT and FIC. The Rule-Based model utilized a simple decision tree model as it closely resembles the decision-making process in a rule-based system. During cross-validation, the model performed remarkably well, correctly detecting over 90% of the predicted suspicious transactions. Moreover, when compared to the Rule-Based model, both models have approximately identical performance and results, indicating that the LOF-IF model's capability of detecting suspicious financial transactions as defined by the FIC and FAFT standard.

6.2 Recommendations

The access and availability of large financial datasets has proven to be a significant challenge, hence this study proposes the need to prioritize access to big data to simulate a more realistic transaction environment. The proposed outlier detection model was modeled on a small dataset of 1087 observations. Reduced barriers to financial datasets will enable access to a large training dataset and improve the model's performance. An important recommendation to consider, concerns the handling of unsupervised learning datasets, particularly for cross-validating unsupervised machine learning models. The majority of cross-validation approaches are mostly applicable to supervised learning models, where determining accuracy and precision is the primary objectives. However, such characteristics are worthless if the dataset is missing a response variable. Thus, it is critical to provide cross-validation procedures for unsupervised learning models that assess the model's stability and consistency. Although Perry (2009) addresses this issue and demonstrates the significance of cross-validating unsupervised models, further work is required to address this matter. Another recommendation the use of lasso regression techniques on SEM models to penalize and exclude weak features and create an efficient parsimonious model. Jacobucci et al. (2016) developed a regularized SEM applies ridge and lasso regression methods to structural equations. This introduces penalties into the model parameters, allowing for greater flexibility in model formulation. In addition, more work needs to be done in examining and reporting on the model interactions between the latent components and their influence (Wei et al. 2004).

6.3 Future Work

The evolving financial landscape with new payment methods such as virtual currencies and the blockchain, provides opportunities for terrorist financiers to evade financial regulators. The FAFT (2020) report also states the ability to quickly transact across borders not only facilitates criminals to electronically acquire, move, and store assets but also disguises the origin and destination of these funds. The resultant models developed in this study can be implemented in such automated environments and act as a bridge between law enforcement agencies and financial institutions. There is a need for further research into the implementation of outlier detection algorithms to monitor virtual financial transactions and devising appropriate outlier detection models for detecting digital terrorist financing. The introduction of deep learning techniques in detecting financial crimes also presents opportunities for further research. There is a need for further work into the implementation of deep learning algorithms to monitor and detect terrorist financing.

References

Abdullah, D. M. & Abdulazeez, A. M. (2021), 'Machine Learning Applications based on SVM Classification: A Review', *Qubahan Academic Journal* **1**(2), 81–90.

Abraham, A. (2005), 'Artificial Neural Networks', *Handbook of Measuring System Design* pp. 901–908.

Adams Jr, R. H. & Cuecuecha, A. (2010), 'Remittances, Household Expenditure and Investment in Guatemala', *World Development* **38**(11), 1626–1641.

Agnew, R. (2010), 'A General Strain Theory of Terrorism', *Theoretical Criminology* **14**(2), 131–153.

Ajzen, I. (1991), 'The Theory of Planned Behavior', *Organizational Behavior and Human Decision Processes* **50**(2), 179–211.

Akram, A. A., Chowdhury, S. & Mobarak, A. M. (2017), 'Effects of Emigration on Rural Labor Markets', pp. 1–34.

Al-Suwaidi, N. & Nobanee, H. (2020), 'Anti-Money Laundering And Anti-Terrorism Financing: A Survey of the Existing Literature And a Future Research Agenda', *Journal of Money Laundering Control* **24**(2), 396–426.

Alexandre, C. R. & Balsa, J. (2023), 'Incorporating machine learning and a risk-based strategy in an anti-money laundering multiagent system', *Expert Systems with Applications* **217**, 119500.

Alghushairy, O., Alsini, R., Soule, T. & Ma, X. (2020), 'A Review of Local Outlier Factor Algorithms For Outlier Detection in Big Data Streams', *Big Data and Cognitive Computing* **5**(1), 1.

Angiulli, F., Basta, S. & Pizzuti, C. (2005), 'Distance-Based Detection and Prediction of Outliers', *IEEE Transactions on Knowledge and Data Engineering* **18**(2), 145–160.

Arbib, M. A. (2003), *The Handbook of Brain Theory and Neural Networks*, MIT Press.

Auskalnis, J., Paulauskas, N. & Baskys, A. (2018), 'Application of Local Outlier Factor Algorithm to Detect Anomalies in Computer Network', *Elektronika ir Elektrotechnika* **24**(3), 96–99.

Barbrook-Johnson, P. & Penn, A. S. (2022), Bayesian Belief Networks, *in* 'Systems Mapping: How to Build and Use Causal Models of Systems', Springer, pp. 97–112.

Barton, D. C. & Barton, P. J. (2006), 'Statistical Analysis/Psychometric Modeling: Understanding and Influencing Societal Vulnerabilities to Terrorism', *Connections* **5**(3), 105–114.

Basit, M. S., Khan, A., Farooq, O., Khan, Y. U. & Shameem, M. (2022), Handling Imbalanced and Overlapped Medical Datasets: A Comparative Study, *in* '2022 5th International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT)', IEEE, pp. 1–7.

Bebchuk, J. & Wittes, J. (2012), 'Fundamentals of Biostatistics', *Clinical Trials in Neurology* p. 28.

Beeres, R., Bertrand, R. & Bollen, M. (2017), 'Profiling Terrorists: Using Statistics to Fight Terrorism', *Netherlands Annual Review of Military Studies 2017: Winning Without Killing: The Strategic and Operational Utility of Non-Kinetic Capabilities in Crises* pp. 221–235.

Berrar, D. (2019), 'Cross-Validation', pp. 542–545.

Biersteker, T. J., Eckert, S. E. & Passas, N. (2008), *Countering The Financing of Terrorism*, Routledge London.

Biyase, M. (2012), 'The Relationship Between Poverty and Remittances in South Africa', *Strategies to Overcome Poverty and Inequality, Towards Carnegie III, University of Cape Town* **3**.

Bollen, K. A., Bauer, D. J., Christ, S. L. & Edwards, M. C. (2010), 'Overview of Structural Equation Models and Recent Extensions', *Statistics in the Social Sciences: Current Methodological Developments* pp. 37–79.

Bolshibayeva, A. K., Rakhmetulayeva, S. & Kulbayeva, A. K. (2023), Machine Learning Methods to Detect Terrorist Financing, *in* 'DTESI (workshops, short papers)'.

Bolton, R. J. & Hand, D. J. (2001), 'Unsupervised Profiling Methods for Fraud Detection', *Credit Scoring and Credit Control VII* .

Borum, R. (2004), *Psychology of Terrorism*, University of South Florida.

Boyd, K. A. (2016), 'Modeling Terrorist Attacks: Assessing Statistical Models to Evaluate Domestic and Ideologically International Attacks', *Studies in Conflict & Terrorism* **39**(7-8), 712–748.

Breunig, M. M., Kriegel, H.-P., Ng, R. T. & Sander, J. (2000), 'LOF: Identifying Density-Based Local Outliers', *SIGMOD Rec.* **29**(2), 93–104.

Chalapathy, R. & Chawla, S. (2019), 'Deep Learning For Anomaly Detection: A Survey', *arXiv preprint arXiv:1901.03407* .

Chandola, V., Banerjee, A. & Kumar, V. (2009), 'Anomaly Detection: A Survey', *ACM Computing Surveys (CSUR)* **41**(3), 1–58.

Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. (2002), 'SMOTE: Synthetic Minority Over-Sampling Technique', *Journal of Artificial Intelligence Research* **16**, 321–357.

Chen, X. & Yuille, A. L. (2004), Detecting and Reading Text in Natural Scenes, *in* 'Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.', Vol. 2, IEEE.

Cheng, Z., Zou, C. & Dong, J. (2019), Outlier Detection using Isolation Forest and Local Outlier Factor, *in* 'Proceedings of the Conference on Research in Adaptive and Convergent Systems', pp. 161–168.

Civelek, M. E. (2018), *Essentials of Structural Equation Modeling*, Zea Books.

Colladon, A. F. & Remondi, E. (2017), 'Using Social Network Analysis To Prevent Money Laundering', *Expert Systems with Applications* **67**, 49–58.

Cortes, C. & Vapnik, V. (1995), 'Support-Vector Networks', *Machine Learning* **20**, 273–297.

Crenshaw, M. (2000), 'The Psychology of Terrorism: An Agenda for the 21st Century', *Political Psychology* **21**(2), 405–420.

Cunningham, P., Cord, M. & Delany, S. J. (2008), Supervised Learning, *in* 'Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval', Springer, pp. 21–49. **URL:** *https://doi.org/10.1007/978-3-540-75171-7_2*

Dilip, R., Vandana, C., Eung, J.-K., Nyasha, K. & Baran, P. (2023), 'Remittances Remain Resilient But Are Slowing', *Migration and Development Brief* **01**(38), 1–33.

Dumitrescu, E.-I., Hué, S. & Hurlin, C. (2021), 'Machine Learning or Econometrics for Credit Scoring: Let's Get the Best of Both Worlds', pp. 1–43.

Durner, T. & Shetret, L. (2015), 'Understanding Bank De-risking and Its Effects on Financial Inclusion: An Exploratory Study', *Global Center on Cooperative Security* **1**, 1–56.

Egmont (2011), 'Enterprise-wide STR Sharing: Issues and Approaches', *Egmont* .
URL: *https://cafiu.nbc.gov.kh/egmonts_group/1.STR_Sharing_White_Paper_Feb_2011.pdf*

Ekici, N. & Akdogan, H. (2020), 'Structural Equation Modeling of Terrorism Perception', *Perspectives on Terrorism* **14**(5), 63–76.

Elnahass, M., Marie, M. & Elgammal, M. (2022), 'Terrorist Attacks and Bank Financial Stability: Evidence from MENA Economies', *Review of Quantitative Finance and Accounting* **59**(1), 383–427.

Ezell, B. C., Bennett, S. P., Von Winterfeldt, D., Sokolowski, J. & Collins, A. J. (2010), 'Probabilistic Risk Analysis and Terrorism Risk', *Risk Analysis: An International Journal* **30**(4), 575–589.

FAFT (2013), 'The Role of Hawala and Other Similar Service Providers in Money Laundering and Terrorist Financing', *Financial Action Task Force (FAFT)-Paris* .

FAFT (2015), 'Emerging Terrorist Financing Risks', *Financial Action Task Force Report* .

FAFT (2020), 'Money Laundering and Terrorist Financing. Red flag Indicators Associated with Virtual Assets', *Finacial Action Task Force. Paris* .

FAFT (2021), 'Anti-Money Laundering And Counter-Terrorist Financing Measures -South Africa ', *Financial Action Task Force (FAFT), - Fourth Round Mutual Evaluation Report* **01**(4), 19–90.

FAFT (2023), 'Crowdfunding for Terrorism Financing', *Financial Action Task Force* .

FIC (2019), 'Guidance Note 4B: On Reporting of Suspicious and Unusual Transactions and Activities to the Financial Intelligence Centre', *Financial Intelligence Centre Manual - Fourth Round Mutual Evaluation Report* .

FIC (2020), 'Guidance Note 6A: On Terrorist Financing and Terrorist Property Reporting Obligations in Terms of Section 28A of the Financial Intelligence Center Act, 2001', *Financial Intelligence Centre Manual - Fourth Round Mutual Evaluation Report* .

FIC, . A. R. (2023), 'Financial Intelligence Centre Annual Report', p. 40.

FIC-NAMIBIA (2023), 'Financial Intelligence Centre Annual Report', pp. 30–32.

Flach, P. (2012), *Machine Learning: The Art and Science of Algorithms That Make Sense of Data*, Cambridge University Press.

Foster, D. P. & Stine, R. A. (2004), 'Variable Selection in Data Mining: Building A Predictive Model for Bankruptcy', *Journal of the American Statistical Association* **99**(466), 303–313.

Freeman, M. (2011), 'The Sources of Terrorist Financing: Theory and Typology', *Studies in Conflict & Terrorism* **34**(6), 461–475.

Freeman, M. & Ruehsen, M. (2013), 'Terrorism Financing Methods: An Overview', *Perspectives On Terrorism* **7**(4), 5–26.

Friedman, N., Geiger, D. & Goldszmidt, M. (1997), 'Bayesian Network Classifiers', *Machine Learning* **29**, 131–163.

Gao, S., Xu, D., Wang, H. & Green, P. (2009), 'Knowledge-Based Anti-Money Laundering: A Software Agent Bank Application', *Journal of Knowledge Management* **13**(2), 63–75.

Gao, Z. (2009), Application of Cluster-Based Local Outlier Factor Algorithm in Anti-Money Laundering, *in* '2009 International Conference on Management and Service Science', Institute of Electrical and Electronics Engineers, pp. 1–4.

Garnier, R. & Taylor, J. (2009), *Discrete Mathematics: Proofs, Structures and Applications*, CRC press.

Goodrich, M., Tamassia, R. & Goldwasser, M. (2014), *Data Structures and Algorithms in Python*, Wiley.

Graeme, H., Aron, H. & Tankiso, M. (2022), 'SA CRISIS: How R6bn got from Spaza Shops to African Terrorists', *Sunday Times* .
URL: *https://www.timeslive.co.za/sunday-times-daily/news/2022-05-08-sas-is-crisis-how-r6bn-got-from-spaza-shops-to-african-terrorists/*

Hastie, T., Tibshirani, R., Friedman, J. H. & Friedman, J. H. (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Vol. 2, Springer Publication.

Hawkins, D. M. (1980), *Identification of Outliers*, Vol. 11, Springer.

Hilal, W., Gadsden, S. A. & Yawney, J. (2022), 'Financial Fraud: a Review of Anomaly Detection Techniques and Recent Advances', *Expert Systems With Applications* **193**, 116429.

Hoyle, R. H. (2012), *Handbook of Structural Equation Modeling*, Guilford Press.

Huang, Z. (1998), 'Extensions to the K-Means Algorithm for Clustering Large Data Sets With Categorical Values', *Data Mining and Knowledge Discovery* **2**(3), 283–304.

Huang, Z. & Ng, M. K. (2003), 'A Note on K-modes Clustering', *Journal of Classification* **20**(2), 257.

Islam, M., Chen, G. & Jin, S. (2019), 'An Overview of Neural Network', *American Journal of Neural Networks and Applications* **5**(1), 7–11.

Jacobucci, R., Grimm, K. J. & McArdle, J. J. (2016), 'Regularized Structural Equation Modeling', *Structural Equation Modeling: A Multidisciplinary Journal* **23**(4), 555–566.

Jenatabadi, H. S. (2015), 'A Critical Story About Sample Size, Outliers, and Normality Criteria in Structural Equation Modelling ', *Outliers, and Normality Criteria in Structural Equation Modelling - April, 2015* .

Jennings, N. R. (2000), 'On Agent-Based Software Engineering', *Artificial Intelligence* **117**(2), 277–296.

Johannesen, N. J., Kolhe, M. L. & Goodwin, M. (2023), Vertical Approach Anomaly Detection Using Local Outlier Factor, *in* 'Power Systems Cybersecurity: Methods, Concepts, and Best Practices', Springer, pp. 297–310.

John, H. & Naaz, S. (2019), 'Credit Card Fraud Detection using Local Outlier Factor and Isolation Forest', *International. Journal of Computer. Science. Engineering* **7**(4), 1060–1064.

Johnson, R. A. & Wichern, D. W. (2007), *Applied Multivariate Statistical Analysis*, Prentice Hall Upper Saddle River, NJ.

Jose-de Jesus, R.-S., Maria-Jesus, S.-V. & Mariadel Mar, C. M. (2021), 'Money Laundering and Terrorism Financing Detection Using Neural Networks and an Abnormality Indicator', *Expert Systems with Applications* **169**, 114470.

Jost, P. M. & Sandhu, H. S. (2000), 'The Hawala Alternative Remittance System and its Role in Money Laundering'.

Jung, S. (2013), 'Structural Equation Modeling with Small Sample Sizes using Two-stage Ridge Least-Squares Estimation', *Behavior Research Methods* **45**, 75–81.

Kaplan, D. (2008), *Structural Equation Modeling: Foundations and Extensions*, Vol. 10, SAGE publications.

Kapp-Joswig, J.-O. F. & Keller, B. G. (2022), 'Clustering–Basic Concepts and Methods', *arXiv preprint arXiv:2212.01248* pp. 1–59.

Kaufman, L. & Rousseeuw, P. J. (2009), *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons.

Keatinge, T. & Keen, F. (2020), 'A Sharper Image: Advancing A Risk-Based Response to Terrorist Financing', *Royal United Services Institute - Occasional Paper* **01**.

Kempen, A. (2023), 'Not Our Color of Choice: Implications of the grey-listing on South Africa', *Servamus Community-Based Safety and Security Magazine* **116**(4), 23–25.

Khan, N. S., Larik, A. S., Rajput, Q. & Haider, S. (2013), 'A Bayesian Approach for Suspicious Financial Activity Reporting', *International Journal of Computers and Applications* **35**(4), 181–187.

Kline, R. B. (1998), 'Structural Equation Modeling', *New York: Guilford* .

Kline, R. B. (2023), *Principles and Practice of Structural Equation Modeling*, Guilford Publications.

Kotsiantis, S. B. (2013), 'Decision Trees: A Recent Overview', *Artificial Intelligence Review* **39**, 261–283.

Kulatilleke, G. K. (2022), 'Credit Card Fraud Detection-Classifier Selection Strategy', *ArXiv Preprint ArXiv:2208.11900* pp. 1–17.

Laqueur, W. (2003), *No End to War: Terrorism in the Twenty-First Century*, Bloomsbury Publishing.

Leonov, S., Yarovenko, H., Boiko, A. & Dotsenko, T. (2019), 'Information System for Monitoring Banking Transactions Related to Money Laundering', *International Conference on Monitoring, Modeling and Management of Emergent Economy* **2422**(8), 297–307.
URL: *https://api.semanticscholar.org/CorpusID:190494964*

Liu, F. T., Ting, K. M. & Zhou, Z.-H. (2008), Isolation Forest, *in* '2008 8th Institute of Electrical and Electronics Engineers International Conference on Data Mining', Institute of Electrical and Electronics Engineers, pp. 413–422.

Lucas, R. E. B. (1987), 'Emigration to South Africa's Mines', *The American Economic Review* **77**(3), 313–330.

Major, J. A. (2002), 'Advanced Techniques For Modeling Terrorism Risk', *The Journal of Risk Finance* **4**(1), 15–24.

Mandhare, H. C. & Idate, S. (2017), A Comparative Study of Cluster-based Outlier Detection, Distance-based Outlier Detection and Density-based Outlier Detection Techniques, *in* '2017 International Conference on Intelligent Computing and Control Systems (ICICCS)', Institute of Electrical and Electronics Engineers, pp. 931–935.

Markou, M. & Singh, S. (2003), 'Novelty Detection: A Review Part 1: Statistical Approaches', *Signal Processing* **83**(12), 2481–2497.

Mehrotra, K. G., Mohan, C. K. & Huang, H. (2017), *Anomaly Detection Principles and Algorithms*, Springer.

Mercer, L. C. (1990), 'Fraud Detection via Regression Analysis', *Computers & Security* **9**(4), 331–338.

Mukhtar, A. (2018), 'Money Laundering, Terror Financing, and FAFT: Implications For Pakistan', *Journal of Current Affairs* **3**(1), 27–56.

Nivette, A., Eisner, M. & Ribeaud, D. (2017), 'Developmental Predictors of Violent Extremist Attitudes: A Test of General Strain Theory', *Journal of Research in Crime and Delinquency* **54**(6), 755–790.

OECD (2019), 'Money Laundering and Terrorist Financing Awareness Handbook for Tax Examiners and Tax Auditors', *Organisation for Economic Co-operation and Development* .

Pearl, J. (2012), 'The Causal Foundations of Structural Equation Modeling', *Handbook of structural equation modeling* pp. 68–91.

Perry, P. O. (2009), 'Cross-Validation for Unsupervised Learning', *arXiv preprint arXiv:0909.3052* pp. 1–140.

Posel, D. & Casale, D. (2006), 'Internal Labour Migration and Household Poverty in Post-Apartheid South Africa', *H. Bhorat and R. Kanbur (2006) Poverty and Policy in Post-Apartheid South Africa. HSRC Press: Pretoria* .

Qu, Y., Chen, X., Li, F., Yang, F., Ji, J. & Li, L. (2020), 'Empirical Evaluation on the Impact of Class Overlap for EEG-based Early Epileptic Seizure Detection', *IEEE Access* **8**, 180328–180340.

Ramaswamy, S., Rastogi, R. & Shim, K. (2000), Efficient Algorithms for Mining Outliers from Large Datasets, *in* 'Proceedings of the 2000 ACM SIGMOD international conference on Management of data', pp. 427–438.

Rangongo, T. (2022), 'South Africas Threat of Being Grey-Listed by Financial Crimes Watchdog Growing Bigger Than Ever', *Money Marketing* **2022**(9), 2–3.

Rapoport, H. & Docquier, F. (2006), 'The Economics of Migrants' Remittances', *Handbook of the Economics of Giving, Altruism, and Reciprocity* **2**, 1135–1198.

Ratha, D. (2003), 'Workers Remittances: An Important And Stable Source of External Development Finance', *Global Development Finance* **9**, 157–174.

Ratha, D. (2013), 'The Impact of Remittances on Economic Growth and Poverty Reduction', *Policy Brief* **8**(1), 1–13.

Raza, S. & Haider, S. (2011), 'Suspicious Activity Reporting Using Dynamic Bayesian Networks', *Procedia Computer Science* **3**, 987–991.

Rokach, L. & Maimon, O. (2005), 'Clustering Methods', *Data Mining and Knowledge Discovery Handbook* pp. 321–352.

Romaniuk, P. (2014), 'The State of the Art on the Financing of Terrorism', *The RUSI Journal* **159**(2), 6–17.

Said, A. B., Hadjidj, R. & Foufou, S. (2017), 'Cluster Validity Index Based on Jeffrey Divergence', *Pattern Analysis and Applications* **20**, 21–31.

Samantha Maitland, A., Raymond Choo, K.-K. & Liu, L. (2011), 'An Analysis of Money Laundering and Terrorism Financing Typologies', *Journal of Money Laundering Control* **15**(1), 85–111.

Sánchez-Rebollo, C., Puente, C., Palacios, R., Piriz, C., Fuentes, J. P. & Jarauta, J. (2019), 'Detection of Jihadism in Social Networks using Big Data Techniques Supported by Graphs and Fuzzy Clustering', *Complexity* **2019**.

Savage, D., Wang, Q., Chou, P. L., Zhang, X. & Yu, X. (2016), 'Detection of Money Laundering Groups Using Supervised Learning in Networks', *ArXiv/abs/1608.00708* .
URL: *https://api.semanticscholar.org/CorpusID:16895665*

Shen, I.-L., Docquier, F. & Rapoport, H. (2010), 'Remittances and Inequality: A Dynamic Migration Model', *The Journal of Economic Inequality* **8**, 197–220.

Shokry, A. M., Rizka, M. A. & Labib, N. M. (2020), Counter Terrorism Finance By Detecting Money Laundering Hidden Networks Using Unsupervised Machine Learning Algorithm, *in* 'International Conferences ICT, Society, and Human Beings', pp. 89–97.

Sinaga, K. P. & Yang, M. (2020), 'Unsupervised K-means Clustering Algorithm', *Institute of Electrical and Electronics Engineers Access* **8**, 80716–80727.

Sudjianto, A., Yuan, M., Kern, D., Nair, S., Zhang, A. & Cela-Díaz, F. (2010), 'Statistical Methods For Fighting Financial Crimes', *Technometrics* **52**(1), 5–19.

Taha, A. A. & Hanbury, A. (2015), 'Metrics for Evaluating 3D Medical Image Segmentation: Analysis, Selection, and Tool', *BMC Medical Imaging* **15**, 1–28.

Teichmann, F. M. (2019), 'Recent Trends in Money Laundering and Terrorism Financing', *Journal of Financial Regulation and Compliance* **27**(1), 2–12.

Tiwari, A. (2022), Supervised Learning: From Theory to Applications, *in* 'Artificial Intelligence and Machine Learning for EDGE Computing', Elsevier, pp. 23–32.

UNDP (2017), 'Journey to Extremism in Africa', *United Nations Development Programme Special Report* **01**(1), 1–84.

UNDP (2023), 'Journey to Extremism in Africa: Pathways to Recruitment and Disengagemnt', *United Nations Development Programme Special Report* **02**(2), 1–84.

Wang, X. & Xu, Y. (2019), An Improved Index for Clustering Validation Based on Silhouette Index and Calinski-Harabasz Index, *in* 'IOP Conference Series: Materials Science and Engineering', Vol. 569, IOP Publishing, p. 52024.

Wei, M., Mallinckrodt, B., Russell, D. W. & Abraham, W. T. (2004), 'Maladaptive Perfectionism as a Mediator and Moderator between Adult Attachment and Depressive Mood', *Journal of Counseling Psychology* **51**(2), 201.

Wierzchoń, S. T. & Kłopotek, M. A. (2018), *Modern Algorithms of Cluster Analysis*, Vol. 34, Springer.

Witten, D. & James, G. (2013), *An Introduction to Statistical Learning with Applications in R*, Springer Publications.

World Bank (2021), 'A Draft Framework For Money Laundering/ Terrorism Financing Risk Assessment a Remittance Corridor', *World Bank Report Paper* .

World Bank Database Repository (2015), `https://drive.google.com/uc?export=download&id=191kP2Rs7zKYJnnJWk9q9Fv7VgejvWBFT`. Accessed: 2022-09-30.

Yang, D. (2011), 'Migrant Remittances', *Journal of Economic Perspectives* **25**(3), 129–152.

Yee, O. S., Sagadevan, S. & Malim, N. H. A. H. (2018), 'Credit Card Fraud Detection Using Machine Learning As Data Mining Technique', *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)* **10**(1-4), 23–27.

Yuan, C. & Yang, H. (2019), 'Research on k-value Selection Method of k-Means Clustering Algorithm', *Multidisciplinary Scientific Journal* **2**(2), 226–235.

Zakaria, M., Mabrouka, A. & Sarhan, S. (2014), 'Artificial Neural Network: A Brief Overview', *Neural Networks* **1**, 2.

Zhang, J. & Yang, Y. (2023), 'Density-Distance Outlier Detection Algorithm Based on Natural Neighborhood', *Axioms* **12**(5), 425.

Zhu, B., Baesens, B. & vanden Broucke, S. K. (2017), 'An Empirical Comparison of Techniques For the Class Imbalance Problem in Churn Prediction', *Information Sciences* **408**, 84–99.

Zimek, A., Schubert, E. & Kriegel, H.-P. (2012), 'A Survey on Unsupervised Outlier Detection in High-Dimensional Numerical Data', *Statistical Analysis and Data Mining: The ASA Data Science Journal* **5**(5), 363–387.

Zojaji, Z., Atani, R. E. & Monadjemi, A. H. (2016), 'A Survey of Credit Card Fraud Detection Techniques: Data and Technique Oriented Perspective', *arXiv preprint arXiv:1611.06439* .

Zubair, A. A.-Q., Oseni, U. A. & Yasin, N. M. (2015), 'Anti-Terrorism Financing Laws in Malaysia: Current Trends and Developments', *International Islamic University Malaysia Journal* **23**, 149.

Appendix

6.1 R Code

```
##########################
## Author: Stanley M Mbiva
## Last update: 12/04/2024
## This program presents the
## LOF-IF Outlier detection algorithm
##used to detect suspicious transactions
#in migrant remittance data
##########################
#loading the packages
set.seed(123)

ins_packages <- function(x){
  x <- as.character(match.call()[[2]])
  if (!require(x, character.only = TRUE)){
    install.packages(pkgs = x, repos ="http://cran.r-project.org")
    require(x, character.only = TRUE)
  }
}
ins_packages(rio); ins_packages(skimr); ins_packages(DT); ins_packages(tidyverse)
ins_packages(ggplot2); ins_packages(corrplot); ins_packages(psych);
ins_packages(dplyr)
ins_packages(ggcorrplot); ins_packages(caret); ins_packages(clustMixType);
ins_packages(FactorMinerR); ins_packages(mice); ins_packages(graphics);
ins_packages(mvtnorm);ins_packages(stats); ins_packages(utils);
ins_packages(doParallel);ins_packages(parallel);
ins_packages(foreach);ins_packages(missMDA); ins_packages(RcmdrMisc);
```

```
ins_packages(regsem);ins_packages(glmnet);ins_packages(lattice);
ins_packages(latticeExtra);ins_packages(tidyverse);ins_packages(Rlof);
ins_packages(readr); ins_packages(caret);ins_packages(clustMixType);
ins_packages(lavaanPlot);
ins_packages(missMDA);ins_packages(factoextra);ins_packages(anomaly);
ins_packages(isotree);ins_packages(IsolationForest);ins_packages(fBasics);
ins_packages(dplyr);ins_packages(dbplyr);ins_packages(psych);
ins_packages(Factoshiny);ins_packages(semPlot);ins_packages(DiagrammeR);
ins_packages(stringr);ins_packages(bpca);
ins_packages(factoextra);ins_packages(emmeans);ins_packages(RColorBrewer);
ins_packages(FactoMineR);ins_packages(gridExtra);ins_packages(ggiraphExtra);
ins_packages(adegenet);ins_packages(Hmisc);ins_packages(xtable);
ins_packages(knitr);ins_packages(ggfortify);ins_packages(lavaan);
ins_packages(regsem);
ins_packages(cvms)

#================================================
#Loading the data

mig_rem_df <- rio::import("migrant_rem_datamsv.csv")
head(mig_rem_df)
#mig_rem_df_imp <- rio::import("migrant_rem_data_nmv.csv")
# imputated zero as a place holder value

#Identifying the factor variables
mig_rem_df$B03 <- factor(mig_rem_df$B03, order = FALSE)
mig_rem_df$B04 <- factor(mig_rem_df$B04, order= FALSE)
mig_rem_df$B06 <- factor(mig_rem_df$B06, order = FALSE)
mig_rem_df$B07 <- factor(mig_rem_df$B07, order = FALSE)
mig_rem_df$B10 <- factor(mig_rem_df$B10, order = FALSE)
mig_rem_df$B11 <- factor(mig_rem_df$B11, order = FALSE)
mig_rem_df$B12 <- factor(mig_rem_df$B12, order = FALSE)
mig_rem_df$B15 <- factor(mig_rem_df$B15, order = FALSE)
mig_rem_df$C06 <- factor(mig_rem_df$C06, order = FALSE)
mig_rem_df$C09A <- factor(mig_rem_df$C09A , order = FALSE)
mig_rem_df$C09B <- factor(mig_rem_df$C09B, order = FALSE)
mig_rem_df$C09C <- factor(mig_rem_df$C09C, order = FALSE)
```

```
mig_rem_df$C10 <- factor(mig_rem_df$C10, order = FALSE)
mig_rem_df$C16 <- factor(mig_rem_df$C16,order = FALSE)
mig_rem_df$C12 <- factor(mig_rem_df$C12, order = FALSE)
mig_rem_df$C13 <- factor(mig_rem_df$C13, order = FALSE)
mig_rem_df$C18 <- factor(mig_rem_df$C18, order  = FALSE)
mig_rem_df$C19 <- factor(mig_rem_df$C19, order = FALSE)
mig_rem_df$C20 <- factor(mig_rem_df$C20, order = FALSE)
mig_rem_df$C21 <- factor(mig_rem_df$C21, order = FALSE)
mig_rem_df$C22 <- factor(mig_rem_df$C22, order = FALSE)
mig_rem_df$C25 <- factor(mig_rem_df$C25, order = FALSE)
mig_rem_df$C26 <- factor(mig_rem_df$C26, order = FALSE)
mig_rem_df$C28 <- factor(mig_rem_df$C28, order = FALSE)
mig_rem_df$C31 <- factor(mig_rem_df$C31, order = FALSE)
mig_rem_df$C35 <- factor(mig_rem_df$C35, order = FALSE)

# 1. Structural Model
model_1 <- '
  #measurement model
  d1  =~ hhsize_recip +relation_to_respondent
  d2  =~ edu  + country_of_study + total_net_income +recip_main_activity
#  d3  =~ relation_to_recip + state_of_birth
  d3  =~ recip_bank_account + times_he_remits
  +currency_received_money
  +approach_to_send+additional_costs_paid
  d4  =~ place_charact + electricity + water
  +mobile_phone + relation_to_recip + state_of_birth

  #latent measurement
  susp =~ d1 + d2 + d3+ d4
d1~~d1
d2~~d2
d3~~d3
d4~~d4'

fit_1 <- sem(model_1, data = num_data1, se = TRUE,sample.cov = NULL, estimator="MLMV")
summary(fit_1, standard = TRUE, fit.measure = TRUE)
```

```
semPaths(fit_1, what = "paths",whatLabels = "par", rotation = 2,
         layout = "tree", style = "lisrel",color = "white",
         curvature = 3, curve = 1, residuals = TRUE, nCharNodes = 7,
         shapeMan = "rectangle", sizeMan = 8, sizeMan2 = 5)

semPaths(fit_1,whatLabels="std",  rotation = 2, intercepts=FALSE, style="OpenMx",
                      curvature = 3, curve = 1, residuals = TRUE, nCharNodes = 7,
                      curveAdjacent = TRUE,title=TRUE, layout="tree2",curvePivot=TRUE)

#==========================================

pv <- parameterEstimates(fit_1,standardized=TRUE)

b <- paste(round(pv$est,3), ifelse(pv$pvalue <= 0.05, "**", "ns"), sep="")
b <- ifelse(b == "1NA", "1", b)
#------------------------------------------
#Make predictions
pred <- predict(fit_1)
hcdist <- pred[,5]
hcdist
mdist <- ifelse(hcdist <= 0, 0, 1)
mdist
prop.table(table(mdist))
# head(num_data1)
 graf <- histogram(~hcdist, type="count", ylab=" ",xlab=" ")
# graf
#hist(hcdist)

graf$panel.args.common$breaks
xbreak <- graf$panel.args.common$breaks
xbreak
xbreak[1] <- NA
xbreak <- round(xbreak, 0)

xbreak
update(graf, scales=list(x=list(at=xbreak, rot=45), y=list(at=seq(0,1600, by=100))),
 ylim=c(0,1200))
```

```
### Cross Validation

# head(num_data1)
# dim(num_data1)

num_data1[["Resp"]] <- mdist

# Training set
num_data1[["ID"]] <- paste(1:nrow(num_data1))
dat.train <- slice_sample(num_data1, prop=0.75)
dat.train$Resp <- as.factor(dat.train$Resp)

# Validation set
dat.test <- anti_join(num_data1, dat.train, by="ID")
dat.test$Resp<- as.factor(dat.test$Resp)

x.train <- as.data.frame(subset(dat.train, select=c(-ID, -Resp)))
y.train <- as.character(dat.train$Resp)
#-----------------------------------------
head(x.train)
head(y.train)

grid <- 10^seq(10, -2, length=100)

mod.lasso <- glmnet(x.train, y.train, alpha=1, lambda=grid)

mod.lasso$lambda

cv.out <- glmnet(x.train, y.train, alpha=1)
plot(cv.out)
bestlam <- min(cv.out$lambda)
bestlam
ggplot(cv.out)
x.test <- dat.test[,c(-47,-48)]
# x.test <- dat.test
```

```
lasso.pred <- predict(mod.lasso, s=bestlam, newx=as.matrix(x.test), type="class")

coefi <- predict(mod.lasso, s=bestlam, type="coefficients")
conf_matrix<-confusionMatrix(as.matrix(asso.pred, as.array(x.test$amount_sent_netcosts)))

pred <- ifelse(lasso.pred < 0.5, 0, 1)

table(pred, dat.test$Resp)

coef(mod.lasso, s=bestlam)
#-----------------------------------------
#Hierarchical clustering Ascendant

res_hcpc_clust <- HCPC(res_PCA_clus, kk = Inf, consol = TRUE, min =3, max= 10 )
plot(res_hcpc_clust)
#plot(res_hcpc_clust, axes = 5:4 )

# HC Output
res_hcpc_clust$call$t #shows the PCA results
res_hcpc_clust$data.clust #returns original data
res_hcpc_clust$desc.var #characterises the classes found interms of variables
res_hcpc_clust$desc.axes #
plot(res_hcpc_clust, axes = c(1,2,3,12))
res_hcpc_clust$desc.ind
# Determine the number of clusters (k) using the elbow method
wcss <- vector("double")  # Within-cluster sum of squares

for (i in 1:10) {
  kmeans_result <- kmeans(combined_data, centers = i)
  wcss[i] <- kmeans_result$tot.withinss
}

# Plot the elbow method graph
plot(1:10, wcss, type = "b", pch = 19, frame = FALSE, xlab = "Number of Clusters", ylab =

abline(v = which.min(wcss), col = "red", lty = 2)  # Vertical line at the elbow point
```

```
#==========================================
#kprototype

res_kproto_data <- cbind(res_MIPCA$res.imputePCA,res_nFAC$completeObs)
#joining the two data frames
res_kproto_clus<-kproto(res_kproto_data,
                        method = "gower",
                        k =  3,
                        iter.max = 100,
                        nstart = 1)
summary(res_kproto_clus)
#==========================================
#Isolation Forest
install.packages("caret")
install.packages("Rlof")
 install.packages("isotree")
 install.packages("solitude")

library(dplyr)
library(Rlof)
library(caret)
library(isotree)
library(solitude)
#-----------------------------------------------
#the isolation forest function
lof_isolation_forest<-function(data,k){
# Step 1: Construct an Isolation Forest
sample_size <-256

# Fit an isolation forest model
model <- isolation.forest(data, sample_size = 256,
                          max_depth = ceiling(log2(sample_size)),
                          seed = 24, ntrees = 500)
data$pred<-predict.isolation_forest(model, data, type = "score")#predictions
data$outlier<- as.factor(ifelse(data$pred > 0.5,
"suspicious_transactions","normal_transactions"))
out_data<- data[data$outlier=="suspicious_transactions",]
```

```
out_data_prun<- data[data$outlier=="normal_transactions",]
# Step 2:LOF values for outlier candidate
head(out_data)
#lof_model <- lof(outlier_candidates, k = 3)
out_data$lof_values<-  lof(out_data$pred, k =3)

out_data$Class<- as.factor(ifelse(out_data$lof_values< 1, "suspicious","normal"))

#combined data set

out_data_prun$Class[out_data_prun$outlier == "normal_transactions"]<-"normal"
out_data<-within(out_data, rm(lof_values))

final_output <-rbind(out_data_prun,out_data)


return(final_output)

}
#  top n points with the highest LOF values as target outliers
#n <-10
#top_outliers <- outlier_coefficients[order(-lof_model)[1:n], ]
#print(top_outliers)
##############################################################################
# Cross-validation of the model
# using the Isolation Forest and the Local Outlier Factor Model
data<-read.csv('migrant_rem_data_nmv.csv')

table(data$Class)
data$Class<- ifelse(data$total_value_remitgoods>1500, "suspicious","normal")

#data$Class[data$Class == 0]<-"normal"
#data$Class[data$Class == 1]<-"suspicious"

set.seed(1254)

train_index <- createDataPartition(data$total_value_remitgoods,
```

```
p =.70, list = FALSE, times= 1)
train_data <- data[train_index,]
test_data <- data[-train_index,]
table(train_data$Class)
# Create the train and test data frame

#Convert the outcome variable to type factor
train_data$Class<- as.factor(train_data$Class)
test_data$Class <- as.factor(test_data$Class)

#Specify cross validation
train_ctrl <- trainControl(method = "cv",
                           number = 10,
                           savePredictions = T,
                           verboseIter =TRUE,
                           summaryFunction = defaultSummary,
                           classProbs = TRUE)
#==========================================
# LOF-IF Ensemble
#set random set
# head(data)
# cv_data <- data_com %>% select_if(is.numeric)
# cv_data[is.na(cv_data)] = 0
# head(cv_data)
#cv_out_data_pred<- rbind(cv_out_data,cv_out_data_pruned)
# #==========================================
# #train_df$Suspicion <- factor(train_df$Suspicion, levels = c("suspicious_transactions",

set.seed(123)  # Set a seed for reproducibility
cv_model <- train(
  x = train_data,
  y = train_data$Class,  # Ad hoc response variable for unsupervised learning
  methodArgs = "lof_isolation_forest",  # Specify function
  trControl = train_ctrl,  # Cross-validation control object
  tuneList = list( k = 3)  # Model parameters

)
```

```
#==========================================
#output in the regression coefficients
summary(cv_model)#variable importance of the predictors
varImp(cv_model)

#Predict the outcome and create a confusion matrix function
predictions <-  predict(cv_model,newdata = test_data)
confusionMatrix(data = predictions, test_data$Class)
table(cv_model$pred$obs)
)
#==========================================
#Decision Tree model
# Install and load necessary packages
install.packages("rpart")
install.packages("rpart.plot")
install.packages("caret")
library(caret)
library(rpart)
library(rpart.plot)

# Load your dataset
set.seed(123)
cv_df<- read.csv("migrant_rem_data_nmv.csv")
cv_df[is.na(cv_df)] = 0
cv_df$Suspicion <-
ifelse(cv_df$total_value_remitmoney >2500 |cv_df$amount_sent_netcosts>1500,
"suspicious_transactions",ifelse(cv_df$main_job==6& cv_df$total_net_income== 7,
"suspicious_transactions",ifelse (cv_df$main_job==6 |cv_df$total_net_income== 9, "suspicio
"suspicious_transactions","normal_transaction"))))
head(cv_df)

# Preprocess the data
cv_df$main_job <- as.factor(cv_df$main_job )
cv_df$total_net_income <- as.factor(cv_df$total_net_income)
cv_df$Suspicion <- as.factor(cv_df$Suspicion)
```

```
# Split the data into training and testing sets
set.seed(123)

sample_index <- sample(seq_len(nrow(cv_df)), size = 0.75 * nrow(cv_df))
train_data <- cv_df[sample_index, ]
test_data <- cv_df[-sample_index, ]

# Build the decision tree model
tree_model <- rpart(Suspicion ~ amount_sent_netcosts + total_net_income + main_job +
total_value_remitmoney + times_he_remits, data = cv_df, method = "class")

# Visualize the decision tree
rpart.plot(tree_model)

## Evaluate the model
predictions <- predict(tree_model, test_data, type = "class")
confusion_matrix <- table(test_data$Suspicion, predictions)
print(confusion_matrix)
# Calculate accuracy
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
print(paste("Accuracy:", round(accuracy, 4)))
# Prune the tree
printcp(tree_model)
pruned_tree <- prune(tree_model,
cp = tree_model$cptable[which.min(tree_model$cptable[,"xerror"]), "CP"])

# Visualize the pruned tree
rpart.plot(pruned_tree)


\section{$k$-prototype}


#kprototype clusttering
Distance type: standard

Numeric predictors: 17
```

```
Categorical predictors: 26
Lambda: 463136

Number of Clusters: 3
Cluster sizes: 488 27 572
Within cluster error: 1485142438 1861866083 2788542823

Cluster prototypes:
       A03      C02      B05       B09      C07       C08      C11      C15      C23
1 4.317623 2.167426 20.15156  6.797131 2.913025  361.5046 36.55967 3.148969 19.81352
2 4.148148 2.222222 37.51852 18.962963 7.222222 6690.3704 48.02550 9.481481 61.48148
3 2.723776 2.011776 37.90427 19.982517 7.761160  883.7265 48.52028 6.063535 46.35664
         C24        C27       C29      C30      C32       C33       C34       C36
1   52.27992   32.51009  8.106557 12.25915 12.96203  65.45107  87.19811  527.1311
2 2401.10070 2822.20407 56.629630 27.47762 29.97682 205.30653 296.08451 1464.2756
3  156.65170   65.57586 16.040210 14.09900 16.49089  58.49097 140.95129  475.6055
B03 B04 B06
3   2   2
1   1   1
1   1   1
   B07 B10 B11 B12 B15 C06 C09A C09B C09C C10 C12 C13 C16 C18 C19 C20 1
1   1   4   2   1   1   2    4    1    1   5   4   8   1   1   1   1
2   1   6   1   4   7   2    1    1    1   5   6   4   1   1   1   1
3   1   4   1   4   4   2    4    1    1   6   4   8   1   1   1   1
          C35
1           2
2           2
3           2
> summary(res_kproto_clus)
A03
  Min. 1st Qu. Median     Mean 3rd Qu. Max.
1    1       3      4 4.317623       5   10
2    1       3      4 4.148148       5    8
3    1       1      2 2.723776       4   10

------------------------------------------------------------
C02
```

```
  Min. 1st Qu. Median     Mean 3rd Qu. Max.
1    1     1.0      2 2.167426       3    5
2    1     1.5      2 2.222222       3    4
3    1     1.0      2 2.011776       3    5

------------------------------------------------------------
B05
  Min. 1st Qu. Median     Mean 3rd Qu. Max.
1    0       6   15.5 20.15156      27   99
2    8      31   37.0 37.51852      51   60
3    1      29   35.0 37.90427      42   99

------------------------------------------------------------
B09
  Min. 1st Qu. Median      Mean 3rd Qu. Max.
1    0     0.0      0  6.797131      14   43
2    0    13.5     17 18.962963      26   57
3    0    14.0     19 19.982517      25   52

------------------------------------------------------------
C07
       Min.    1st Qu.   Median     Mean  3rd Qu. Max.
1 -4.996896 -0.1441185 1.000000 2.913025  3.00000   99
2  1.000000  3.0000000 6.000000 7.222222 12.00000   12
3 -3.165134  3.0000000 5.421798 7.761160 10.25568   99

------------------------------------------------------------
C08
       Min.   1st Qu.    Median      Mean   3rd Qu.  Max.
1 -304.8260  225.3432  303.1991  361.5046  450.8628  2000
2  840.0000 5000.0000 6000.0000 6690.3704 7100.0000 15000
3  -88.8625  500.0000  648.1498  883.7265 1000.0000  4500

------------------------------------------------------------
C11
  Min. 1st Qu.   Median     Mean 3rd Qu. Max.
1    8 30.8856 33.67348 36.55967    39.0   82
```

```
2   24 38.0000 49.00000 48.02550    56.5   72
3    1 38.0000 47.59138 48.52028    60.0   91

------------------------------------------------------------
C15
        Min.  1st Qu.   Median     Mean   3rd Qu. Max.
1 -0.1006985 1.732254 2.401422 3.148969  3.861151   20
2  0.0000000 4.000000 6.000000 9.481481 10.000000   70
3  0.0000000 3.790024 5.000000 6.063535  7.000000   34

------------------------------------------------------------
C23
  Min. 1st Qu. Median     Mean 3rd Qu. Max.
1    0       0      0 19.81352      60  422
2   49      61     63 61.48148      63   64
3    0      30     62 46.35664      63   71

------------------------------------------------------------
C24
   Min. 1st Qu. Median       Mean 3rd Qu.  Max.
1 0.000       0      0   52.27992   47.15  3300
2 0.119     550    990 2401.10070 3850.00 13200
3 0.000       0    110  156.65170  220.00  1980

------------------------------------------------------------
C27
  Min. 1st Qu. Median       Mean 3rd Qu.  Max.
1    0    0.00   0.00   32.51009       0  2730
2    0   85.18 633.25 2822.20407    2685 33000
3    0    0.00   0.00   65.57586      85  1800

------------------------------------------------------------
C29
  Min. 1st Qu. Median      Mean 3rd Qu. Max.
1    0     0.0      0  8.106557       1  480
2    0     1.5     24 56.629630      48  552
3    0     0.0      2 16.040210      24  504
```

```
------------------------------------------------------------
C30
       Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
1 -17.64378 8.759492 13.31711 12.25915 16.00559  75.0000
2 -11.00004 4.000000 18.33661 27.47762 40.50207 101.9277
3 -19.28969 8.000000 14.54152 14.09900 17.05459 100.0000

------------------------------------------------------------
C32
       Min.   1st Qu.   Median     Mean  3rd Qu.      Max.
1 -20.40345  9.312763 14.48101 12.96203 17.15846  36.90563
2 -15.75469 13.590048 22.16227 29.97682 39.10003 107.19653
3 -23.94260 10.383674 15.83486 16.49089 19.53423 200.00000

------------------------------------------------------------
C33
       Min.   1st Qu.    Median      Mean   3rd Qu. Max.
1   0.00000  48.50343  62.65861  65.45107  78.69896  635
2   0.00000 117.72619 149.47070 205.30653 287.16200  598
3 -12.79798  35.18907  51.28107  58.49097  66.35916  655

------------------------------------------------------------
C34
       Min.  1st Qu.   Median      Mean  3rd Qu. Max.
1 -290.0892 41.60364 100.7715  87.19811 132.6873  400
2 -265.1386  0.00000 193.5713 296.08451 408.4660 2000
3 -347.3552 61.30731 127.5803 140.95129 173.0596 1100

------------------------------------------------------------
C36
       Min.  1st Qu.    Median      Mean   3rd Qu. Max.
1  -105.610 297.7414  439.5658  527.1311  698.9950 7000
2 -1357.337 667.6817 1010.6761 1464.2756 1708.8861 5550
3 -1697.552 207.9618  413.3126  475.6055  625.5501 7000

------------------------------------------------------------
```

B03

cluster	1	2	3	4	5	6	7	8	9	98	99
1	0.098	0.160	0.617	0.006	0.061	0.000	0.004	0.035	0.012	0.004	0.002
2	0.593	0.222	0.148	0.000	0.000	0.037	0.000	0.000	0.000	0.000	0.000
3	0.738	0.157	0.012	0.000	0.037	0.012	0.007	0.019	0.017	0.000	0.000

B04

cluster	1	2
1	0.412	0.588
2	0.704	0.296
3	0.673	0.327

B06

cluster	1	2	3	8	9
1	0.352	0.611	0.033	0.002	0.002
2	0.852	0.148	0.000	0.000	0.000
3	0.970	0.017	0.012	0.000	0.000

B07

cluster	1	2	3	4	5	6
1	0.770	0.016	0.006	0.014	0.014	0.000
2	0.296	0.037	0.111	0.037	0.037	0.000
3	0.313	0.024	0.014	0.063	0.026	0.009

	7	8	9	10	11	12
1	0.000	0.002	0.057	0.035	0.012	0.010
2	0.000	0.037	0.074	0.111	0.000	0.148
3	0.019	0.026	0.058	0.129	0.052	0.121

14	15	17	19	20	23
0.000	0.006	0.000	0.000	0.000	0.000

```
        0.000 0.000 0.000 0.000 0.000 0.000
        0.010 0.000 0.014 0.003 0.005 0.002

        24    25    26    28    29    30    31
       0.002 0.012 0.000 0.002 0.002 0.000 0.002
       0.000 0.037 0.000 0.000 0.000 0.000 0.000
       0.003 0.037 0.016 0.009 0.016 0.002 0.009

cluster    32    33    34    36    98    99
      1 0.004 0.010 0.000 0.000 0.018 0.002
      2 0.037 0.037 0.000 0.000 0.000 0.000
      3 0.007 0.000 0.002 0.002 0.009 0.000

-------------------------------------------------------------
B10

cluster     1     2     3     4     5     6     7     8     9
      1 0.168 0.129 0.109 0.273 0.195 0.041 0.066 0.018 0.002
      2 0.037 0.074 0.074 0.111 0.296 0.407 0.000 0.000 0.000
      3 0.016 0.042 0.058 0.344 0.311 0.163 0.063 0.003 0.000

-------------------------------------------------------------
B11

cluster     1     2     3     8     9
      1 0.043 0.869 0.033 0.027 0.029
      2 0.481 0.444 0.037 0.000 0.037
      3 0.579 0.388 0.026 0.007 0.000

-------------------------------------------------------------
B12

cluster     1     2     3     4     5     6     7     8    98    99
      1 0.715 0.047 0.029 0.092 0.018 0.002 0.004 0.043 0.020 0.029
      2 0.074 0.000 0.000 0.630 0.185 0.000 0.000 0.074 0.000 0.037
      3 0.098 0.150 0.068 0.432 0.100 0.010 0.016 0.107 0.009 0.010
```

B15

cluster	1	2	3	4	5	6	7	8	9
1	0.654	0.051	0.070	0.025	0.033	0.012	0.002	0.086	0.068
2	0.074	0.000	0.037	0.148	0.185	0.185	0.296	0.037	0.037
3	0.066	0.084	0.206	0.283	0.086	0.038	0.009	0.084	0.143

C06

cluster	1	2	8
1	0.107	0.889	0.004
2	0.444	0.556	0.000
3	0.170	0.811	0.019

C09A

cluster	1	2	3	4	5	6	8	9	10	11
1	0.119	0.000	0.004	0.791	0.004	0.059	0.000	0.006	0.008	0.008
2	0.407	0.000	0.111	0.296	0.000	0.074	0.000	0.000	0.074	0.037
3	0.253	0.003	0.007	0.586	0.012	0.080	0.002	0.007	0.026	0.023

C09B

cluster	1	2	3	4	5	6	10	11
1	0.764	0.002	0.000	0.027	0.004	0.197	0.004	0.002
2	0.593	0.000	0.111	0.037	0.000	0.222	0.037	0.000
3	0.615	0.002	0.005	0.077	0.012	0.260	0.019	0.009

C09C

cluster	1	2	3	4	5	6	10	11
1	0.955	0.002	0.002	0.002	0.002	0.023	0.012	0.002

```
      2 0.889 0.000 0.000 0.037 0.037 0.000 0.037 0.000
      3 0.937 0.000 0.000 0.023 0.002 0.021 0.007 0.010

-------------------------------------------------------------------
C10

cluster     2     3     5     6     7     8     9
      1 0.016 0.002 0.637 0.234 0.023 0.072 0.016
      2 0.111 0.000 0.407 0.333 0.074 0.000 0.074
      3 0.054 0.023 0.175 0.615 0.031 0.080 0.021

-------------------------------------------------------------------
C12

cluster     1     2     3     4     5     6     7     8     9
      1 0.012 0.016 0.027 0.783 0.086 0.027 0.016 0.027 0.006
      2 0.037 0.074 0.222 0.111 0.037 0.407 0.074 0.037 0.000
      3 0.212 0.047 0.080 0.414 0.122 0.059 0.037 0.026 0.002

-------------------------------------------------------------------
C13

cluster     1     2     3     4     5     6     7     8    98    99
      1 0.053 0.020 0.012 0.264 0.074 0.006 0.043 0.502 0.025 0.000
      2 0.037 0.000 0.000 0.333 0.259 0.037 0.222 0.111 0.000 0.000
      3 0.072 0.035 0.002 0.128 0.142 0.012 0.156 0.414 0.037 0.003

-------------------------------------------------------------------
C16

cluster     0     1     2     3     4     5     6     7     8     9
      1 0.000 0.834 0.008 0.006 0.006 0.004 0.000 0.002 0.002 0.055
      2 0.000 0.222 0.037 0.074 0.037 0.074 0.000 0.037 0.000 0.037
      3 0.002 0.559 0.031 0.007 0.026 0.009 0.005 0.010 0.007 0.047

      10    11    12    13    14    15    17    20    23    24    25    26    28    29
   1. 0.023 0.016 0.025 0.002 0.000 0.002 0.000 0.000 0.000 0.000 0.006 0.000 0.002 0.000
```

```
   2. 0.148 0.000 0.037 0.000 0.037 0.148 0.037 0.000 0.000 0.000 0.074 0.000 0.000 0.000
   3. 0.070 0.019 0.105 0.002 0.003 0.009 0.007 0.002 0.002 0.002 0.035 0.010 0.003 0.010

cluster    32    33    98    99
      1 0.000 0.000 0.002 0.002
      2 0.000 0.000 0.000 0.000
      3 0.007 0.005 0.000 0.000

------------------------------------------------------------------
C18

cluster     1     2     8     9
      1 0.980 0.018 0.002 0.000
      2 0.963 0.037 0.000 0.000
      3 0.960 0.035 0.002 0.003

------------------------------------------------------------------
C19

cluster     1     2     8
      1 0.984 0.014 0.002
      2 0.889 0.111 0.000
      3 0.962 0.037 0.002

------------------------------------------------------------------
C20

cluster     1     2     8
      1 0.957 0.039 0.004
      2 0.926 0.074 0.000
      3 0.885 0.108 0.007

------------------------------------------------------------------
C21

cluster     1     2     8
      1 0.949 0.043 0.008
```

```
      2 0.889 0.111 0.000
      3 0.802 0.194 0.003

------------------------------------------------------------
C22

cluster     1     2     8     9
      1 0.119 0.855 0.027 0.000
      2 0.704 0.296 0.000 0.000
      3 0.278 0.682 0.038 0.002

------------------------------------------------------------
C25

cluster     1     2     3     4     8     9
      1 0.006 0.943 0.045 0.002 0.004 0.000
      2 0.037 0.963 0.000 0.000 0.000 0.000
      3 0.005 0.944 0.044 0.000 0.005 0.002

------------------------------------------------------------
C26

cluster     1     2     3     4     8
      1 0.586 0.039 0.371 0.000 0.004
      2 0.333 0.296 0.333 0.000 0.037
      3 0.577 0.100 0.316 0.002 0.005

------------------------------------------------------------
C28

cluster     1     2     3     4     5     6     8     9    10    11
      1 0.088 0.000 0.004 0.777 0.008 0.090 0.000 0.006 0.016 0.010
      2 0.407 0.000 0.111 0.222 0.037 0.037 0.000 0.000 0.111 0.074
      3 0.212 0.002 0.007 0.568 0.014 0.128 0.002 0.007 0.028 0.033

------------------------------------------------------------
C31
```

cluster	0	1	2	8	9	15	20	60
1	0.002	0.107	0.869	0.020	0.000	0.002	0.000	0.000
2	0.037	0.222	0.630	0.074	0.000	0.000	0.000	0.037
3	0.002	0.177	0.778	0.037	0.003	0.002	0.002	0.000

C35

cluster	0	1	2	8
1	0.002	0.000	0.980	0.018
2	0.000	0.000	0.889	0.111
3	0.002	0.007	0.972	0.019
