



UNIVERSITY OF NAIROBI
SCHOOL OF COMPUTING AND INFORMATICS

TITLE:
REAL TIME FRAUD DETECTION SYSTEM FOR MOBILE BANKING: BASED ON
EXPERIENTIAL PARADIGM

BY:
JAMES OCHIENG OMOLLO
P58/77093/2012


SUPERVISOR:
PROF. ELISHA OPIYO

A RESEARCH PROPOSAL SUBMITTED TO THE SCHOOL OF COMPUTING AND
INFORMATICS IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
AWARD OF THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE OF
UNIVERSITY OF NAIROBI.

October 2020.

DECLARATION

I James Ochieng Omollo declare that this project proposal is my original work and has not been submitted for the award of a degree in any other university. \

Signed: 

Date: 28th June 2021

James Ochieng Omollo

P58/77093/2012

This proposal report has been submitted for examination with my approval as the university supervisor.

Signed:

Date: 08th July 2021



Prof. Elisha Opiyo

Senior Lecturer,

School Of Computing and Informatics,

University of Nairobi, Kenya

DEDICATION

This project proposal is dedicated to my entire family. You have always been at my side during times of need and your constant encouragements have made me achieve this far.

ACKNOWLEDGEMENTS

My deepest gratitude is, first, due to the infinite God, who through inimitable wisdom established creative systems and immutable laws through which man could aim to pursue his highest aspirations and fulfil his deepest potential. Secondly, special thanks go to my supervisor Dr. Elisha Opiyo, for providing unlimited, invaluable and active guidance throughout the study. His immense command and knowledge of the subject matter enabled me to shape this research proposal to the product that it is now. Thirdly, with sincere gratitude and appreciation, I would also like to acknowledge my dear wife, Maximilah and my entire family, for showing the patience and understanding during the most intensive and extensive phases of this research work. May your vision and horizon ever expand to match your limitless potential. Finally, I owe my gratitude to a number of people who in one way or another contributed towards completion of this work especially my fellow colleagues at work and students.

ABSTRACT

The current banking industry is characterized by hyper-competition driven by technological innovations that revolve around provision of ubiquitous access to banking services especially through mobile banking. Proliferation of mobile phones in Kenya acts as a substrate for the increased adoption of mobile banking in Kenya. Frauds perpetrated through mobile banking platforms have become prevalent eroding the hard-earned profits by banks. This research therefore was aimed at developing a case-based reasoning framework that would do real time fraud detection in mobile banking. Case-based reasoning problem solving technique which makes use of prior knowledge and specific problem scenarios (cases) to solve new problems by identifying similar past problem episodes and applying them to the new problem situations. The research employed an incremental prototyping model in which the overall architectural design was done upfront but the detailed design and developments of the subcomponents were done in incremental manner. The research used a four-step approach for building the Case Based Reasoning engine which included features calibration, case stabilization, and implementation and finally the evaluation process. The research relied on both primary and secondary data to collect the past fraud incidences to build a reference case library. The research design was in form of interviews done to the target population comprising of individuals drawn from the bank's risk, forensics, digital channels support and information systems security. The Case Based Reasoning algorithm implemented incorporated a threshold retrieval mechanism combined with K-Nearest Neighbor algorithm. The system prototype was built and trained using a data set of 120 transactions with system evaluation done in three iterations of 40 transactions in every iteration revealing an average classification accuracy of 84.17%.

LIST OF FIGURES

Figure 1:	Conceptual Model for Case Based Reasoning Real Time Fraud Detection System.....	31
Figure 4:	Case Based Reasoning System Research Approach	33
Figure 5:	Incremental Prototyping Model	36
Figure 6:	High Level Architectural Design for Real Time Fraud Detection System.....	43
Figure 7:	Attribute Weighting Matrix Definition	48
Figure 8:	Fraud Score Matrix Definition	48
Figure 9:	Sample view of Case Library	50
Figure 10:	Sample view of Mobile Transactions & Associated Classes.....	53
Figure 11:	CBR Real Time Fraud Detection System Context Diagram	54

LIST OF ABBREVIATIONS

CBR	Case Based Reasoning
CBS	Core Banking System
FDS	Fraud Detection System
PIN	Personal Identification Number
AI	Artificial Intelligence
ML	Machine learning
MNOs	Mobile Network Operators
SIM	Subscriber Identity Module
CBK	Central Bank of Kenya
STK	Sim Tool Kit
IP	Internet Protocol addresses
USSD	Unstructured Supplementary Service Data
GSM	Global System for Mobile
C2B	Customer to Bank
B2C	Bank to Customer
ESB	Enterprise Service Bus
KNN	K-Nearest Neighbour

TABLE OF CONTENTS

DECLARATION	i
DEDICATION	ii
ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
LIST OF FIGURES	v
LIST OF ABBREVIATIONS	vi
CHAPTER ONE: INTRODUCTION	1
1.1 Background of the Study	1
1.2 Problem Statement.....	3
1.3 Project Goal	3
1.4 Objectives	3
1.5 Justification.....	4
1.6 Scope of the study.....	4
CHAPTER TWO: LITERATURE REVIEW	6
2.1 Introduction.....	6
2.2 Fraud Detection.....	6
2.3 The Concept of Mobile Banking.....	7
2.4 Mobile Banking Fraud in Kenya.....	10
2.5 Fraud Detection for Mobile Banking.....	12
2.6 Theoretical Review of Fraud Detection Techniques & Algorithms	13
2.6.1 Case Based Reasoning	13
2.6.2 Decision Trees	15
2.6.3 Clustering Techniques	15
2.7 Challenges and Opportunities in Fraud Detection	16
2.8 Conceptual Model.....	19
2.9 Research Gap	20
CHAPTER THREE: RESEARCH METHODOLOGY	21
3.1 Introduction.....	21
3.2 Research Approach.....	21
3.3 System Development Methodology.....	23
3.4 Requirements Engineering.....	24

3.5	Study Population.....	25
3.6	Sample and Sampling Technique.....	25
3.7	Research Instruments	26
3.8	Data Collection Procedure	26
3.9	Conclusion	26
	CHAPTER FOUR: SYSTEM ANALYSIS, DESIGN & IMPLEMENTATION.....	28
4.1	Introduction.....	28
4.2	System Analysis.....	28
4.2.1	Review of Existing Fraud Management Systems and Frameworks	28
4.2.2	Functional Requirements	28
4.2.3	Non-Functional Requirements	29
4.2.4	Features Augmentation & Calibration	30
4.2.5	System Design	31
4.2.6	Database Design.....	32
4.2.7	Activity Diagram	35
4.2.8	System Development & Implementation.....	36
4.3	Contextual Application of the CBR Real Time Fraud Detection System.....	41
	CHAPTER FIVE: SYSTEM EVALUATION AND ACHIEVEMENTS.....	44
5.1	Introduction.....	44
5.2	Performance evaluation measures.....	44
5.3	Experimental Results and Discussion.....	46
5.4	Discussions on the Objective’s Achievement.....	49
	CHAPTER SIX: CONCLUSION AND RECOMMENDATIONS	52
6.1	Introduction.....	52
6.2	Conclusion	52
6.3	Recommendations.....	53
	REFERENCES.....	54
	APPENDICES:.....	58
	APPENDIX 1: RESEARCH PLAN AND TIMETABLE	58
	APPENDIX 2: LIST OF REQUIRED RESOURCES	59
	APPENDIX 3: BUDGET ESTIMATES.....	59
	APPENDIX 4: QUESTIONNAIRE SURVEY.....	60
	APPENDIX 5: SAMPLE SOURCE CODES	64

CHAPTER ONE: INTRODUCTION

This chapter gives an introduction to mobile banking and the need for real time fraud detection within mobile banking & electronic payments space. It introduces the study by outlining the core thematic areas.

1.1 Background of the Study

According to the latest data from Kenya Bankers Association, mobile banking penetration has reached staggering 68% penetration amongst the Kenyan population. There is an increase in Mobile banking fraud with the expansion of mobile technology and the drive by financial institutions to offer ease of access and convenience of banking and other financial services (Deloitte, 2018). As different financial institutions continue to deploy financial services through the mobile & e-payments space, fraudsters are also getting increasingly complex and adaptive and continue to devise means and ways to reap out of these new ecosystems. This increase in the fraudulent transactions results into huge financial losses and puts the institutions at risk in terms of their reputation and customer confidence. The financial industry recognizes this problem but has not put in sufficient measures and tools to pre-empt such attempts or at least realize the acts just before they are committed (Bolton, 2002).

Fraud prevention is where you try to deter fraud before it happens, according to Boyer (2018). Detection and prevention of fraud is a continuous and cyclic process that involves tracking, detection, decisions, case management and learning and feeding of the learnings into the system. Firms should aim to continuously learn from fraud events and integrate the findings into future processes of monitoring and detection (Bradley,2019). This calls for an organizational approach to the fraud analytics life cycle. Personal Identification Numbers (PINs), passwords, watermarks, amongst others are everyday examples of this. These are precautions taken before fraud happens, but fraud prevention is not a flawless affair since the passwords and PINs of people can be stolen and credit cards can be skimmed to get details to perpetuate fraud. This underscores the primacy of fraud detection (Bradley, 2019). In his study of financial institutions in the Middle East, Boyer (2018) claimed that fraud detection operates reactively where prevention fails. Therefore, there is need for continuous development and enhancements of fraud detection systems because once a fraudster determines fraud detection system exists; they may try to find new ways to beat it.

Organizations should also bear in mind that new fraudsters will come along and there is a risk that both new and already used techniques will be used and therefore one does not eliminate previous methods of detection but rather have to expand the old one. Although the safest way to minimize fraud is to through prevention, fraudsters are resilient and can typically find ways to bypass such steps. Therefore, detecting fraud is necessary once the process of prevention has failed (Gunasegaran, 2018). Elmousalami (2014) found that fraud detection is a problem of prediction and its aim is to optimize the right prediction and preserve incorrect predictions at a reasonable cost level. Studies have shown that data mining has achieved greater performance using Artificial Intelligence (AI) techniques than conventional statistical approaches for developing prediction models (Paheding, 2019).

To support such analysis and classification problems, AI techniques, especially rule-based expert systems, case-based reasoning systems and machine learning (ML) techniques such as neural networks, have been used. As a consequence, the constructs of the models used in statistical methods are pretty basic, easy to understand and appear to underfit the data, whereas models acquired in machine learning techniques are typically very complex, difficult to describe and tend to overfit the data. In fact, the trade-off between the plausibility and parsimony of a model is underfit and over fit of the data, where explanatory power leads to high prediction accuracy and parsimony typically ensures generalizability and interpretability of the model (Elmousalami, 2014).

This research therefore presents an experiential paradigm based on Case Based Reasoning (CBR) system that does fraud analysis in Mobile banking system and flags fraudulent transactions in real time. The system has a self-learning case library with predictive models built through learning from this data, and the capacity to adapt past predictions to help classify current transactions behavioral patterns as either fraudulent or not fraudulent in real time based on a weighting matrix of the transaction attributes.

1.2 Problem Statement

Due to the very high penetration of mobile phones among the Kenyan population, provision of banking service through Mobile phones has proven to be necessary for any banking institution to keep abreast with the increased competition and to increase its financial inclusivity to the larger population by growing its customer base (Mwangi and Njuguna, 2009). The other side of this is that with this increased mobile banking service penetration comes various risks associated with high velocity of money in the mobile space. The matter is even complicated further because most of the mobile banking systems are now integrated with the Mobile Network Operators (MNOs) in a bid to extend their niche markets and offer convenience of banking to their customers. Innovative fraudsters who are continuously researching on the exploitable vulnerabilities of these systems continue to thrive. The vulnerabilities exploited ranges from social engineering; weak data interchange security and encapsulation mechanisms, compromised Personal Identification Numbers (PINs) and passwords, Subscriber Identity Module (SIM) Swaps etc. To be able to safeguard their reputation and maintain customer confidence, it is therefore an obligation on the banking sector to institute mechanisms not only be able to prevent fraud but also be able to detect (where prevention fails) fraudulent activities and cripple them before substantial damage is caused. Unfortunately, the Kenyan banks have lagged behind in putting in the right tools to assist them in such and as at current continue to lose large sums of money to fraudsters.

1.3 Project Goal

The goal of this project is to build a Case Based Reasoning Engine for real time fraud detection of mobile banking transactions.

1.4 Objectives

- i) Investigate and document past incidences of mobile banking fraud
- ii) Establish relevant attributes and features that are useful for classifying transactions as fraudulent and not fraudulent.
- iii) Design a high-level architecture of Real Time Fraud Detection System
- iv) Develop a Case Based Reasoning Engine prototype for real time fraud detection
- v) Evaluate the Case Based Reasoning Engine prototype with new test cases

1.5 Justification

Fraud costs the financial sector about \$80 billion annually, according to research done by Technology.org (2013) for reported fraud cases. Experience can be quite expensive for fraud victims and even lead to identity theft, which can take quite a long time to fix. For banks and other financial institutions, lack of sound fraud management controls can expose the bank to non-compliance with regulations, can create reputational risks and can potentially lead to huge fines by the regulator (Central Bank of Kenya). When fraud reaches unprecedented levels, it can cause deregistration of the bank or the bank service by the regulator in order to protect the customers. Mejia (2019) in his study of anomaly detection in banking showed that instead of singling out specific types of transactions, the fraud solutions should analyze historical transaction data to build a model that can detect fraudulent patterns for future use cases. This model, otherwise called stream computing is used in in which large number of financial transactions are processed and evaluated in real time. A fraud score is calculated for each transaction, which depicts the likelihood of a transaction being fraudulent while minimizing false alarms by analyzing the connection between potentially fraudulent transactions and actual fraud. The model is adapted to data on mobile transactions and then constantly revised to cover new forms of fraud.

The Case Based Reasoning engine is developed into the architecture of the mobile banking system and a case resolution framework generated using the current case library and retrained as circumstances change, forming an automated system that enables the organization to identify fraud before it occurs. A bank may take preventive steps to alert a customer via his or her cell phone or trigger additional measures to authorize a transaction by detecting legitimate transactions that have a high likelihood of being accompanied by a fraudulent transaction. Although the future cannot be predicted by machine learning techniques and stream computing paradigms, they allow financial institutions to make intelligent decisions and intervene to prevent fraud before it occurs.

1.6 Scope of the study

This is a research was carried out in KCB Bank Kenya Limited at its headquarters in Nairobi Kenya whose respondents were drawn from various departments that deal with fraud and risks within the bank. The research study and implementation involved collecting past mobile banking fraud cases for the last five years (from 2013 to 2018), calibrating the significant features and

parameters and implementation of a case-based reasoning system that was later evaluated and the resultant system recommended to piloting in a banking setup. The CBR system implementation was through incremental development model in which case an initial high-level design would be done, and then different system subcomponents low level design and development done in iterative manner and later integrated together into one whole.

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

This chapter explains the concept of fraud detection, examines different types of mobile banking fraud and presents the significance of real time fraud detection in the banking industry. Also presents the previous work related to CBR implementations, gives the theoretical background of the mechanics of case-based reasoning systems, analyses different fraud detection techniques and finally presents the conceptual model of the proposed CBR system.

2.2 Fraud Detection

Several literatures describe fraud as deliberate deceit, such as when a person makes false claims, conceals or omits material evidence, leading to injury to another (Fitch, 2006). Without the violence, it is the exploitation of an institution that inevitably leads to legal consequences (Phua et al., 2005). It can also be referred to as fraudulent deception or the use of false statements to achieve an unfair advantage, according to Bolton and Hand (2002). Jeffery Lehman, (2004) defined fraud as portrayal of a result, whether by words or actions, by inaccurate or deceptive allegations, or by concealment of what should have been declared that deceives and is meant to mislead another in order for the person to act on it or his legal injury.

Fraud was divided into three major categories by Silverstone and Davia (2005). These are those frauds that have been revealed and are widely known; those that have been found but not yet made public by institutions; and those that have not been identified at all. Approximately 20% of the total frauds belong to the revealed fraud group. The advanced reasons for this are that most frauds are either inadvertently discovered or independent auditors do not proactively audit for fraud detection. The other supporting fact is that most companies without internal personnel cannot audit for fraud proactively or if they can, their internal internal auditors do not have sufficient knowledge or experience to proactively detect fraud. Finally, the other fact us that most of the companies' internal controls may also be inadequate to aid in fraud prevention (Albrecht, 2004).

Given the lack of physical presence of consumers in the realm of electronic banking (e-banking), it is very paramount that financial and monetary institutions consider identification of customers seeking these services. Maybe it can be argued that the need to consider the identity of individuals

is the key constraint to the provision of more expansive banking services. In the sense of e-banking services, this concern is the most important factor in the prevalence of fraud, which is growing due to the growth of e-banking. In order to detect the activities of fraudsters, financial and monetary institutions are extremely striving for the pace needed. Due to its indirect impact on customer service in these institutions and the reduced operating costs as a provider of valid and efficient financial services, this topic is of considerable importance (Chartered Institute of Management Accountants, 2008).

2.3 The Concept of Mobile Banking

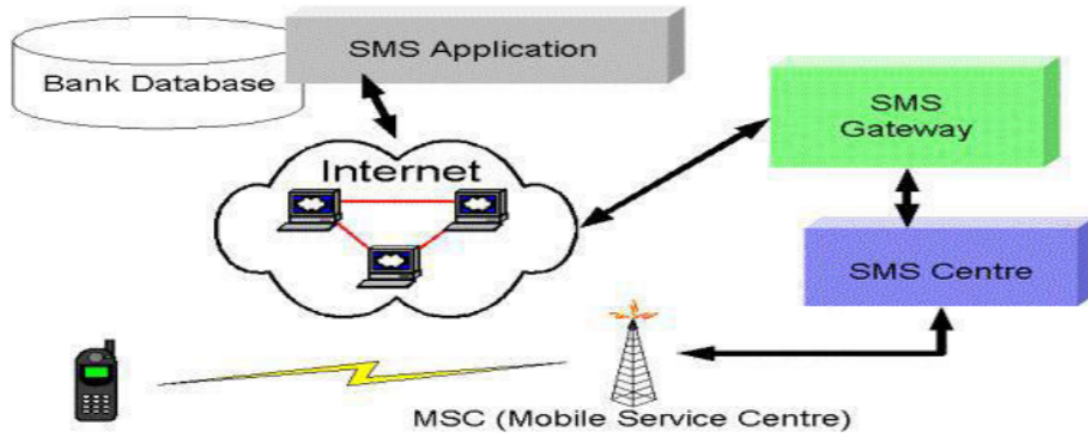
Mobile banking is a subset of electronic banking, which enables access of banking services and facilities using electronic mobile devices such as mobile phones and PDAs (Olweny & Shiphoh, 2011). Although various, and at times competing, labels, and definitions have been used when discussing the provision of financial services through mobile phone networks, this study uses the increasingly popular term “mobile money” to refer to the convergence of mobile telephone and financial services. Mobile banking (M-banking) entails the use of a mobile phone or other mobile device to conduct a financial transaction belonging to a customer’s account, according to Kigen (2010). M-banking refers to the provision and use of banking and financial services such as account balance checks, funds transfers, bill payments, loan applications amongst others with the use of mobile telecommunication devices, according to Kingoo (2011).

Mobile banking has revolutionized money transfer and payments is transferred in third world and now it is destined to deliver more advanced banking services that could make a significant transformations in the peoples' lives. An array of services can be offered by this mode of banking including alerting customers of any updates and transactions on their account via their mobile phones (Kigen, 2010). On their mobile phones, people receive brief messages reminding them of their recent activities in their bank accounts. Mobile banking services can be carried out via SMS, WAP, GPRS, 3G, USSD, and SIM toolkits. Most of these mobile banking services can technically be implemented using a number of different channels as discussed in the following sections.

2.3.1 SMS – Short Messaging Service

To allow mobile banking, SMS uses the common text-messaging protocol. This works by sending a text message with a service instruction to a pre-specified mobile phone number to which the

client required the information. The bank then responds with an SMS providing the relevant details. Additionally, there are a few cases where mobile banking services are availed to the customers using the SMS based utility (Otaïr & Mohammed, 2012).



2.3.2 Mobile Application Clients

For implementing robust and complex mobile banking functionalities such as trading in bonds and shares, mobile apps are the most appropriate. There could be flexibility in their design and architecture that enables configurations to deliver any desired complexities of the user interfaces of any mobile handset. Furthermore, mobile apps can enable the deployment of a very robust and secure communications platform. In order to use the mobile apps, the customers need to download and install the apps into their mobile phone. The mobile devices should be able to support one of the many operating systems including Android or Windows or Apple iOS to use the mobile apps, the customers need to download and install the apps into their mobile phone. The mobile devices should be able to support one of the many operating systems including Android or Windows or Apple iOS.

2.3.3 SIM Application Toolkit (STK)

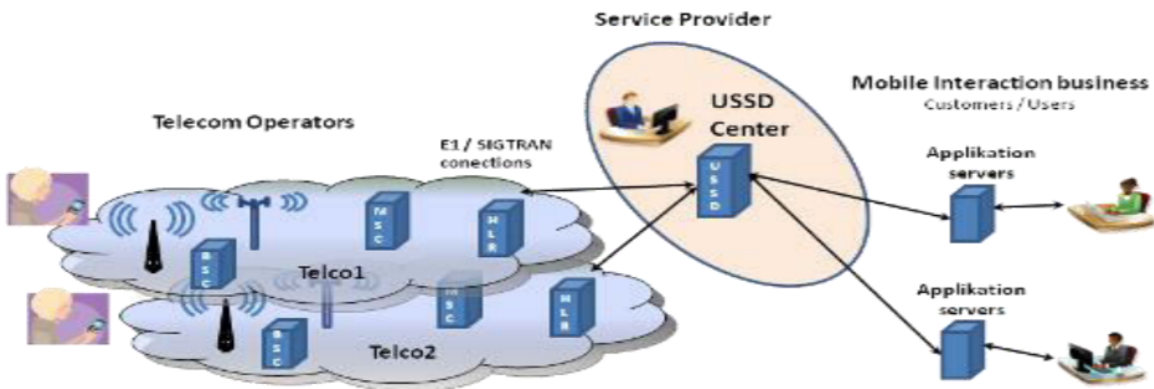
The SIM Application Toolkit (STK) is a GSM system standard that allows the SIM to perform actions meant to deliver different banking services. The SIM Application Toolkit includes a number of functions that are coded into the SIM card and that determine how the SIM communicates with the external end points and can trigger instructions independent of the mobile

handset and the network. This helps the SIM to establish an interactive exchange between a network application and the end user and access or control access to the network.

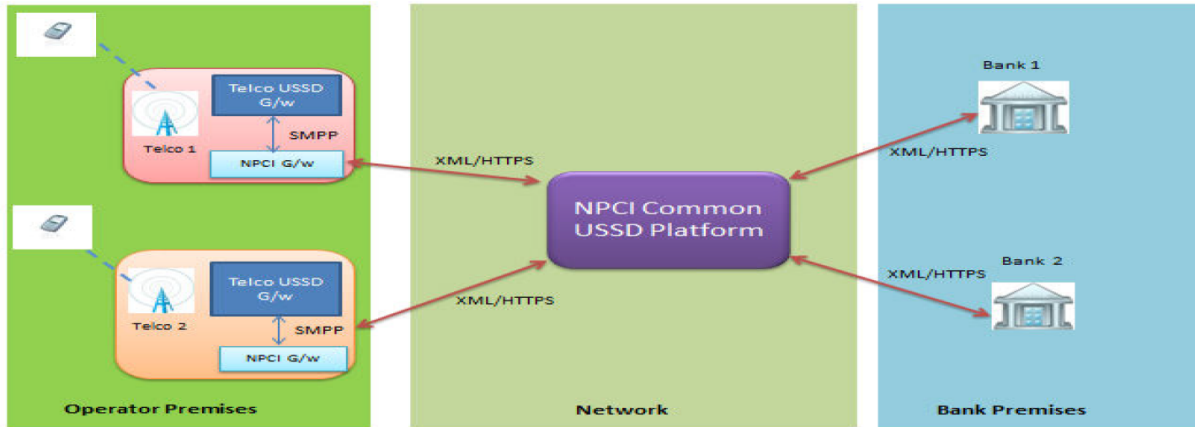
STK has been used for many environments, such as mobile banking, where a menu-based approach is needed. The majority of Kenyan telecoms, including Safaricom and Airtel, have deployed and integrated their mobile banking services with their partner banks through STK.

2.3.4 Unstructured Supplementary Services Data (USSD)

Unstructured Supplementary Service Data (USSD) is a communication protocol used between cellular phones and computer networks of various service providers. USSD can also be used for callback services, WAP surfing, menu-based data and financial services, geographical-based content services as well as part of mobile phone network configuration. USSD messages which can be up to 82 alphanumeric characters in length establish a real-time connection during a USSD session, unlike Short Message Service (SMS) messages. The session remains open, enabling a data sequence to be shared in duplex mode making USSD more responsive than SMS based services.



Usually, USSD operations are triggered by punching a short code between hashes, such as *522#, which would immediately return the response message and potentially show a menu with a number of choices to pick. USSD tends to be one of the most effective means of financial inclusion for mobile banking services.



In today's developing economies, the exponential growth of mobile financial services (MFS) is probably the most significant catalyst to growing financial inclusion. It has enabled access for the growing formerly unbanked sections of the society to affordable and efficient financial services. Innovative mobile money services such as M-Pesa have evolved into significant financial services in Kenya and Tanzania, transacting millions of dollars annually. Sadly, the MFS has become a gateway for fraud and other illegal activity.

2.4 Mobile Banking Fraud in Kenya

The major contributors of mobile money fraud consist of the maturity of mobile money systems, cultural problems, weak or non-standard frameworks and procedures, lack of enforcement of compliance (Mudiri, 2012) and any new service deployments not thoroughly considered, for instance the postpaid system in which the customers are allowed to enjoy the services but billed later (Merritt, 2010).

It's given that every payment system has some intrinsic flaws that could promote fraud when exploited. Rapidity is one of the leading determinants of fraud since the transaction velocity in mobile phone enabled transaction infrastructure is quite higher than cash. Therefore, rapidity is a greater risk factor for mobile banking services than for cash.

In the absence of robust internal controls, this can be an excellent mechanism for perpetrators to defraud financial institutions. Another avenue that can be exploited by fraudsters is a case where they open multiple accounts to move fraud money quickly into these accounts without notice after which they churn the money out through interbank transfers.

Bankruptcy fraud

One of the most complex kinds of fraud to foresee is bankruptcy fraud. However, there are some techniques and approaches that can be used in its prevention. Bankruptcy fraud entails where a customer applies for and uses a mobile loan while in insolvent state and hence cannot pay back the loan. Even though the customer can be given a demand by the bank to pay, he will be regarded as bankrupt and hence the bank will not be able to recover their debts forcing the bank to write off losses. The surest mechanism to prevent such fraud is through pre-validation checks with credit reference bureaus to confirm whether the customer has ever been listed for any bad debts in the past. This will give the bank a barometer to determine whether to grant the loan or not based on the bank's risk appetite.

Theft fraud/counterfeit fraud

This entails a case where a fraudster steals a customer's mobile phone plus the mobile banking PIN or STK PIN and uses them to perform illegal transactions on the customers' accounts before the fraud is realized and the mobile banking access blocked by the bank.

Application fraud

Application fraud entails identity fraud where a fraudster registers for his/her SIM card using false identity. In the realm of mobile banking, these criminals use this false information to open a mobile banking account and continue transact without being realized. As a direct consequent of his mobile banking activities through the account and enhanced credit limit, the fraudster proceeds and take a huge loan from the bank and the vanish after withdrawing the loan. The bank's only defense for such cases enforcement of stringent Know Your Customer (KYC) process and frameworks to ensure that they have the correct and integral details of all the customers they are dealing with. Its therefore important for the banks to do their own KYC and not rely on any third-party KYC information which can end up being false or inaccurate.

Behavioral fraud

This type of fraud occurs when details of legitimate SIM card holder are fraudulently obtained for instance through social engineering where the legitimate mobile banking account holder is duped into revealing his/her PIN number and also gives out his bio data. The transactions are made by fraudster through the SIM card as if they are made by the SIM card owner. This fraud can be also perpetrated when a fraudster gets hold of your National Id card and has your phone and/or mobile

banking details. At one point, the fraudster will do a SIM swap and use your phone to withdraw all your money from the bank account or borrow the maximum loan possible after which he/she dumps the SIM card. Behavioral fraud can be detected by implementing use of IMSI when registering the SIM card.

2.5 Fraud Detection for Mobile Banking

Pavel and Binkley (2007) illustrate that authentication detection technique as one critical pillar in any security system. Authentication is the process of verifying the identity of users, applications, or devices before giving them access to sensitive data or systems. Today's authentication schemes range from a simple user ID and password to multi-factor approaches that include smart cards, PINs, 20 mobile devices.

Pavel and Binkley (2007) posits that the mechanism of authentication detection in any security framework is a very important pillar. Authentication refers to the mechanism by which users, programs, or devices are verified for identity before they are granted access to sensitive information or systems. The nowadays authentication schemes vary from a basic user ID and password to multi-factor methods that involve smart cards, mobile devices and PINs. The choice of the authentication mechanism to be used could depend on the degree of security that an organization wants to offer to its customers, the cost of implementation and support and the target class or segment of customers. While digital revolution has been credited for major transformations on the financial services industry, it is also facilitating new forms of banking fraud due to the transition from the traditional branch-based to multi-channel service offerings which opens up new set of systems vulnerabilities where customers become the weakest links in the chain. It is important to remember that the understanding of online security threats by consumers is often weak which makes them to be easily duped into revealing sensitive information to fraudsters.

The fraudsters have unending appetite for exploitation of digital channels due to that fact that there are huge volumes of digital transactions processed in real time through the digital channels. This is because the immense volume of digital transactions makes the conventional manual methods of fraud tracking and detection unable to detect or report frauds due lack of capacity or the speed to

keep pace with the velocity of transactions. The exponential advancement of mobile fraud is revealing shortcomings in the safeguards of banks. Although banks make investments in providing their customers with the real-time digital banking services, there is no adequate investment or allocation of sufficient capital to keep their digital ecosystems secure. Consequently, many banks fail to identify fraudulent transactions until they are completed due to lack of effective anti-fraud mechanisms and therefore ultimately losing huge amounts of money fraudulently. This challenge is even more pronounced in smaller banking institutions due to constrained resources.

Financial services on mobile phone includes deposits, withdrawals from mobile money wallet, peer to peer transfer, pay for goods and services and m-banking services. The penetration and diffusion of mobile banking in Kenya and across the world has been phenomenal. For example, in Kenya, M-PESA, which was launched in 2007 has been the leading serving offering which has transformed Kenyan economy and many lives. Currently the service has over 25Million customers. As at the end of 2013, M-PESA accounted for 43% of the Kenya's GDP with over 237 million person to person transactions. The telecoms industry alone estimates fraud losses to be around 2-3% of all the mobile money revenues. Mobile banking fraud types include phishing which involves sensitive personal information being obtained by a fraudster usually through social engineering, illegal SIM swaps by use of fake identity documents or collusion with internal staff, Identity theft, Advance Fee scams which entails duping of subscribers to send money fraudsters, intrusion of mobile banking systems through cyber-attacks and denial of service attacks.

2.6 Theoretical Review of Fraud Detection Techniques & Algorithms

Principally, fraud detection is a data classification problem which can take several approaches of either supervised, reinforced or unsupervised.

2.6.1 Case Based Reasoning

Case-based reasoning (CBR) is an artificial intelligence paradigm that utilizes comparisons and similarities with previously solved situations (Nilsson, 1998). Case Based Reasoning incorporates both problem-solving and learning mechanics and has evolved into one of the most prominently applied disciplines of artificial intelligence in current history. CBR is based on the assumption that challenges continue to recur, so that new challenges often have some similarities with previously resolved ones and, hence, past remedies can be applicable to the challenge at hand. CBR is called

a lazy learning methodology when applied to classification problems and, most precisely, instance-based learning, where it utilizes these training instances in the neighborhood of the problem situation to ascertain its class rather than generating abstract representation of these set of training examples (Craw, 2006)

In the object-oriented paradigm, recent knowledge representations and encoding mechanisms for memory models include frames or classes. CBR was initially assumed as a memory model to aid in recalling past circumstances. However, CBR has become a problem-solving technique that could be used in a broad variety of applications, such as design, scheduling, configuration, and diagnosis assessments as well as for information acquisition and representations. Without reflecting specific or deep contextual information, knowledge is expressed by coherent chunks often referred to as cases. CBR offers many information repositories that can be used for application domain processing. The cases and their contents, the language used to define and index cases, the measure of similarities for matching cases, and the modifications or transformations of solutions can be stored in these information containers (Lenz et al., 1998).

Therefore, the fundamental concept of CBR is to adapt past problem solutions to new problems. In CBR, the descriptions of the past solutions which are in form of cases, are stored for in a database for future retrieval and adaptation when new instances with similar attributes is encountered (Glez-Peña et al. 2009). In this context, CBR Systems should be able to learn from transaction trends and respond to new fraud patterns as they evolve. A CBR framework attempts to find a similar case when presented with a new problem situation. There are many algorithms used in CBR systems for classification purposes, but the most common one is the nearest neighbour matching algorithm. By automatically changing and updating weighting steps, a CBR system can maximize the accuracy of its classification and can use several techniques to boost its final accuracy of its prediction.

CBR is useful in a realm with a huge range of instances, with the capacity to work with incomplete or noisy data, can be applied in a hybrid approach to be efficient, scalable, simple to update and manage. CBR systems have a range of merits over other artificial intelligent techniques, because

they provide significant assurance and model accuracy measures, needs minimal to no direct acquisition of expert information, they can be updated and retained with ease, describe the logic behind verdict clearly, are versatile and resilient to missing or noisy data and can take and process on noisy data with acceptable levels of accuracy.

2.6.2 Decision Trees

A decision tree logic defines a similarity tree which is recursively modelled with the tree nodes are labeled with attribute names, the edges are labeled with attribute properties that meet certain conditions and 'the leaves' that comprise a significance factor that is determined as the ratio of the total number of transactions that fulfill these condition(s) to the total number of transactions that are considered legitimate base on their behavioral patterns (Kokkinaki, 1997). The benefit of the decision tree technique is that it is simple to implement, to comprehend and to display. The criteria for verifying each transaction one by one is however a drawback of this method. None the less, trees with similarities have produced fairly accurate results. In his pursuit to create an intrusion detection method for another form of fraud, Fan et al. (2001) also focused on decision trees and in particular, on an inductive decision tree.

An extensive comparison of decision-tree-based data-mining methods applicable to binomial classification issues was made by Derrig and Francis (2008). A significant drawback of the decision tree is that the likelihood of classification as a valid or false claim is not created and therefore it can make distinction between claims in the same classification. Decision tree algorithms have also been accused of not checking previous rules when establishing new rules (Zopounidis and Dimitras, 1998), however there is no proof that this would decrease its predictive or classification accuracy. Additionally, their construction is susceptible to slight adjustments in the training dataset (Sudjianto et al., 2010).

2.6.3 Clustering Techniques

Clustering is one of the unsupervised learning processes in which instances are grouped into distinct sets called clusters. These clusters (Mehrdad, 2018) are often homogeneous. Within a cluster, the instances bear a clear similarity to each other, although those from different clusters vary. Two clustering strategies for behavioural fraud were proposed by Bolton & Hand (2002). The peer groups analysis is a methodology that allows for the recognition of entities that behave

differently from others at different moments, as much as they previously exhibited the same behaviours. They then mark those instances as potentially fraudulent which are subject to review by fraud analysts. The premise of the study of the peer group is that if instances exhibit the same behavior for a given amount of time and then one instance behaves substantially differently, then it is worthy to take note of this instance. Breakpoint analysis uses a different technique. The assumption is that the account has to be investigated if a shift in card use is identified on an individual basis. In other words, the break-point analysis would classify suspicious activity dependent on the transactions of a single card. Potentially malicious conduct signals are an unexpected transaction for a large amount, and a high frequency of use.

2.7 Challenges and Opportunities in Fraud Detection

For a financial institution, an efficient and affective fraud management system is quite necessary failure to which the institutions would be vulnerable to financial, reputational and punitive risks. Instituting an efficient and effective fraud solution provides an institution with a competitive advantage since it bolsters customer trust and experience in the institution and its systems. Additionally, deploying the right fraud management solution provides an institution with tremendous benefits including cost containment and risks reduction, boosting customer loyalty and fostering innovation. Therefore, an efficient and effective fraud detection solution should resolve the following challenges:

Imbalanced and Incomplete Datasets

The imbalanced dataset is one of the important aspects in fraud detection systems. Any fraud detection systems therefore need real data order to guarantee accurate and reliable outcome even though access such data is often subject to confidentiality and regulatory laws and hence is often a challenge. They usually do data anonymization out before making it public, which might need to misclassifications if the process is not done correctly. Hence, the process of extraction, transformation and loading must not lose the desired data attributes that would be essential in classification and prediction of fraudulent transactions.

Transaction Diversity

Another major challenge facing the fraud detection systems is the diverse nature of the transactions that they are required to analyze and classify as either fraudulent or otherwise. There are cases

where the transaction attributes for normal transactions can resemble an abnormally based on the classification's algorithms and the past fraud patterns and hence leading to false positives. This can at times lead to a compromise that sometimes can cost a few actual fraudulent cases for the banks. It is therefore important to implement the system in such a way that the transaction attribute weights, and the thresholds can always be adjusted based on the new fraud patterns and nature of the transactions.

Transaction Velocity & Big Data

Implementation of a real time fraud detection systems is often a challenge given the huge volumes of data and the velocity of data flowing through the digital channels. There is drastic upsurge of transactions per unit time for which the fraud detections must respond due and increase in the complexity of the fraud detection system. The speed and accuracy of fraud detection therefore becomes paramount in the design of the fraud detection systems.

Emerging New Patterns of Fraud

In the last decade, the technology revolution has seen not only a rise in the adoption of technology by the society, but also a growth in the misuse of technology. When technology progresses and advanced fraud detection and prevention methods emerge, the systems evolve using advanced fraudulent activity efficiency techniques to preserve balance. Due to the advent of new techniques and technologies, breaking this balance has been one of the challenging processes. Statistical or data mining techniques were used in the classical detection mechanisms, whereas more refinement leading to machine learning and heuristic methods is necessary in the present environment. It also needs speedy solutions due to the real-time nature of the problems, which could be the greatest challenge.

Fear of False Positives

In any fraud detection systems, one of the key concerns is misclassification. While true negatives tend to be costly, heavier losses are often caused by misclassifications leading to false positives. A false positive is a case where the system classifies a non-fraudulent transaction as fraudulent. If unchecked, this could cost the organization reputational damage and erode customer confidence.

Need for online real time prediction

The high transaction velocity is a real challenge for the design of real time fraud detection systems. Most of the fraud detection systems often operate high latencies simply due to the complexity of

their designs and the runtimes of their complex algorithms. The design of these systems must therefore take this into account and balance accuracy and speed of detection. The faster a fraud is detected the better for the financial institution otherwise the slow detection speeds can lead to enormous financial damage before the fraudulent activities are detected and stopped.

Balancing Priorities of Competitive Offering, Customer demands and security

In the implementation of the fraud detection systems, the financial institution is often required to balance between competitive offerings, customer demands and security of customer data. In order to stay competitive, the bank must strive to offer faster transaction processing speeds while ensuring that the customer is protected against any data breaches of their data by putting in place appropriate security controls and fraud detection systems.

Need to Implement Self-Learning Algorithms

Models and predictive engines for fraud detection need continuous modifications and improvements. The model needs to rapidly assimilate the knowledge once a fraud methodology is discovered and begin searching for the next vulnerability that a possible fraudster can exploit. Since the whole game is always focused on the ability to deter fraud, the strain to be ahead of the fraudsters is paramount. It is important to build fraud detection models and engines using machine learning algorithms that can learn from the positive classifications they create and thus evolve and enhance their functioning, hence minimizing false positives. The sharing of outcomes amongst algorithms also assists in triangulating outcomes.

2.8 Conceptual Model

The figure below shows the proposal implementation architecture of the Case Based Reasoning System for Real-time Fraud Detection in Mobile Banking.

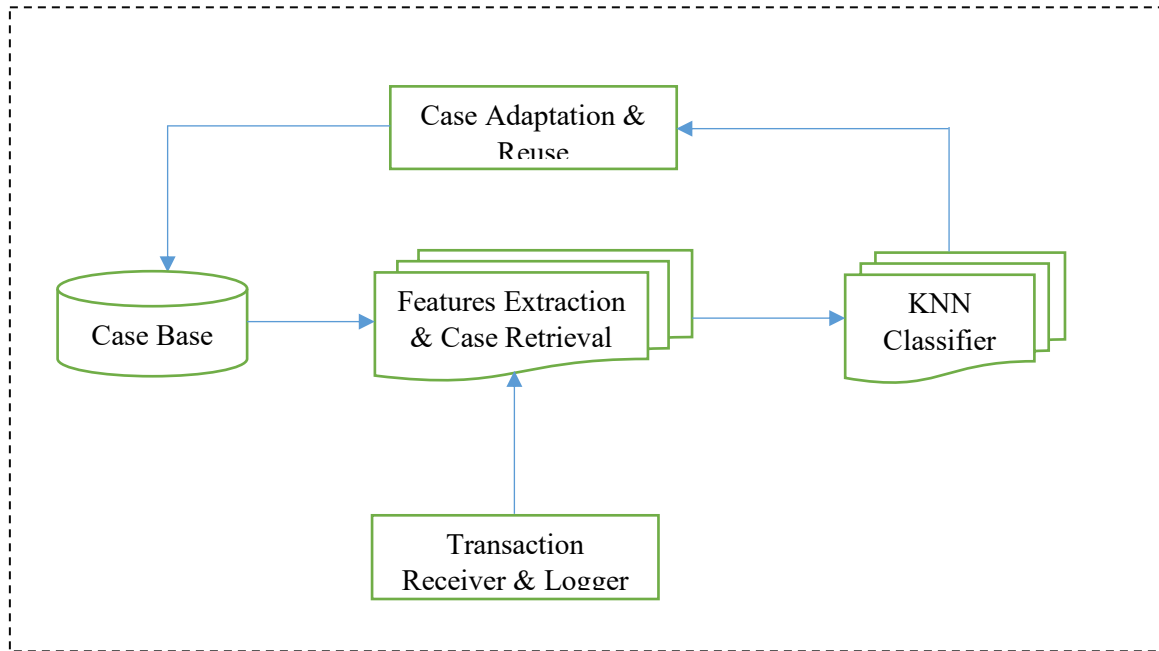


Figure 1: Conceptual Model for Case Based Reasoning Real Time Fraud Detection System

The Case Base: - This function as the repository of prior cases. The cases are indexed (as a key within the database) so that they can be quickly retrieved when necessary. A case contains the past episodes of mobile banking transactions with the diagnostic attributes and a classification as either fraudulent or non-fraudulent.

Transaction Receiver & Logger: – This is the component that listens to transaction invocations from the mobile banking system and logs the transaction into the processing database.

Features Extraction & Case Retrieval - This is the module responsible for the extracting the transaction diagnostic attributes and uses an appropriate algorithm to select set of past cases using weighted threshold retrieval algorithm.

KNN Classifier: – This is the engine that uses the set of retrieved cases and applies KNN similarity measure on the retrieved cases and classified the transaction as fraudulent or non-fraudulent.

Case Adaptation & Reuse: - This is the module meant to adapt the retrieved cases to attempt to correctly classify the new case for instances where the retrieved case cannot be used to correctly classify the current case. The case adapter can either do this through substitution by replacing values of the retrieved case with the new values appropriate to the new case or through transformation in which case the system alters the retrieved case by adding, deleting or replacing parts of the retrieved case in an attempt to correctly classify the new instance or alternatively using specialized heuristic knowledge to repair the retrieved case. Where the retrieved solutions is considered sufficient to classify the current instance, the past case is reused for the classification.

2.9 Research Gap

Most banks in Africa have remained behind in their technological deployment to the extent of managing all the associated security risks. This is largely attributable to management decisions, which in essence does not elevate the seriousness of cyber security risks, and consequently frauds that might be perpetuated through such loopholes. The systems deployed therefore employed weak security architectures and hence remains vulnerable. In addition, there is not management pursuit to deploy full-scale security and fraud monitoring operations center with the requisite tool and expertise to help the banks proactively combat fraud and security breaches. For cases where a subset of such tools are deployed which only offers after the fact information in which case the criminals have already accomplished their target objectives.

CHAPTER THREE: RESEARCH METHODOLOGY

3.1 Introduction

This chapter contains case-based reasoning system development methodology, requirements gathering, research approach, research instruments, study population and data collection procedure.

3.2 Research Approach

The end to end approach to this research project entailed the below steps:

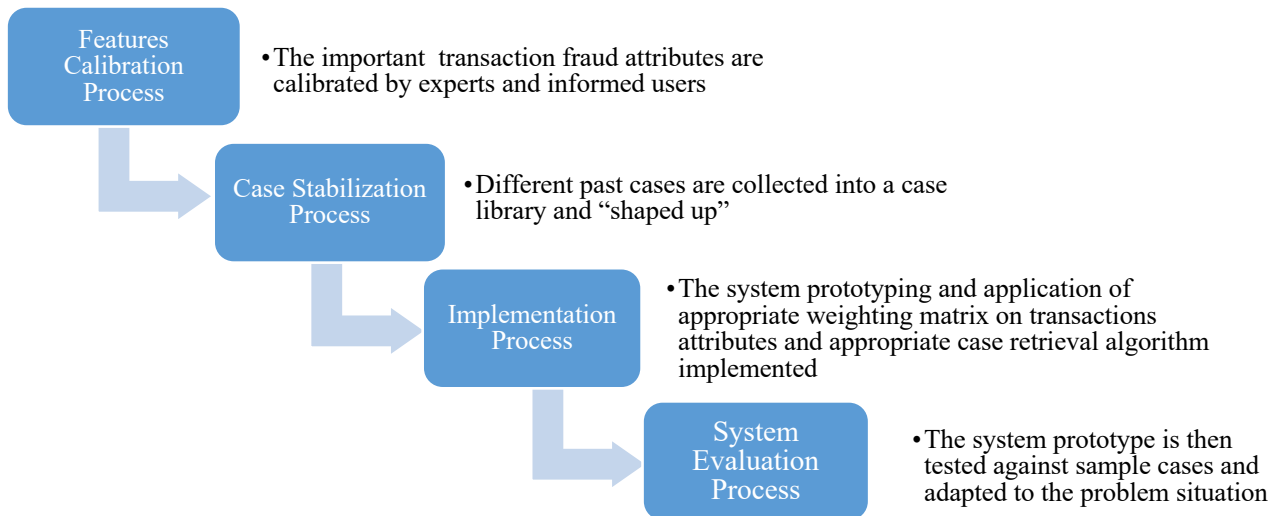


Figure 4: Case Based Reasoning System Research Approach

i) Transaction Features Augmentation & Calibration process

This involved the definition of the features or case descriptors that marks different mobile transaction as fraudulent or non-fraudulent. These features were collected by circulating a questionnaire to various risk and technology teams in the bank. The resulting features were then redistributed to allow the teams to alter, adjust or remove any features they thought were unacceptable, and once all the teams concurred on the final version of the features set, the process

was repeated. The method enabled business users and technocrats to evaluate and agree on both the exact attributes to be used for the classification of these cases, their number and types.

ii) Case Base Stabilization Process

After the transaction features calibration was validated, further cases based on the agreed set were collected over a five years period. The goal of the case stabilization was to gather appropriate number of cases that would have a good coverage of the problem domain of mobile banking fraud. Generally, the number of cases obtained is determined by the problem area in question, the parameters being used and the metadata of the cases. A reference case library was then developed. Thereafter, the case library was evolved through continues cross validation and testing. In this process, the case base was evaluated and revalidated when the tenth the target number of cases was reached during this testing exercise. In addition, some fictitious cases were also introduced that informed further review and refinement of the transaction features and weights. The paramount goal of both feature calibration and case stabilization was to reduces the risks associated with prototyping risk and knowledge engineering which is one of the main risk factors in the design and development of any knowledge-based systems (Gammack et al 1985).

iii) System Implementation Process

After the completion of the case stabilization exercise and reaching the target number of cases, a case library was established, and the system is therefore ready to be used. Based on the training set and moderations done by the fraud experts, each transaction attribute was assigned relative weights. The most similar case(s) were adapted to the current instances for situations where there were no exact matches for the problem under investigation. In order to allow the system to learn, the new instance for which a solution has been discovered was verified and stored for future adaptation. The system was again put through another phase of case stabilization for any new problem situations which had new points or attributes that the system had not been subjected to before. The process of case adaptation was both system-guided and user-guided. User-guided adaptation is where the fraud experts assign similar cases to the current problem situation based on their judgement as informed by the transaction attributes and their relative weights coupled by their experience in the problem domain. For system-guided adaptation, the system processes the current problem situation by searching through the case library and retrieving the most similar cases based on the transaction features and relative weights and adapts to the current instance.

iv) System Evaluation Process

The evaluation of the system was done iteratively through validation (black box testing) and verification (white box testing). The white box testing is concerned with the system's performance and its internal mechanics (Terano 1994). In this study, this entailed the calculation of the time and cost dimensions of transaction processing and classification (In other words the time and cost of case retrieval and adaptation). The verification process was done through testing of sampled cases. For instance, randomly selecting cases from the case base and run them through the system and then comparing the outcomes with their actual results in the case base. As converse to white box testing, black box testing is usually more concerned by how the user interprets the system's feedback and behaviour and the effect of the system in the context or institution where its being applied. A different viewpoint on the evaluation of CBR systems, however, takes into account not only intrinsic factors as influenced by its design and architecture (i.e. system accuracy), but also extrinsic factors such as user acceptance or behavior based on the result of feedback from the system (Althoff 1996). System evaluation and testing using both white box and black box approach were done as guided by a set of test cases that were developed. The outcome of each test case was recorded, and summaries analyzed by experts in order to deduce whether the system had adapted to the extent that it can be relied upon for real time fraud detection. Testing was done in 3 cycles, as the system was increasingly adapted for fraud detection accuracy in which case the case attributes weighting matrix were adjusted in the process.

3.3 System Development Methodology

The system design methodology adopted was incremental prototyping in which requirement engineering, both high level and low-level architecture and design is first done before the system is developed and tested. The design, development and testing were done on in incremental manner as system features are accomplished and confirmed in every iteration until the final product is achieved. The system is marked as complete after fulfilling all the functional and nonfunctional requirements as defined during the requirements engineering phase of the system implementation.

In this paradigm, the development of the system is was decomposed into several of sub-systems or modules, each of which was designed and developed and tested independently.



Figure 5: Incremental Prototyping Model

The general model of the prototype was designed and implemented and then the other intricate components were added incrementally step by step until the objectives of the system were met. This system implementation was therefore taking a top down approach in which entailed overall system overview and coming up with the actual sub-components to be implemented. Each sub-component was then further broken down in terms of design and functionalities to be achieved. Once these base elements were recognized then built as system modules. Once the individual modules were completed, they were integrated and merged together to make the entire system.

The rationale for selecting this methodology was that the approach generally reduced the development costs and time. For each level of development, there was an expected output within a specific timeframe. This was evaluated against the individual level deliverables as well as the overall objectives of the proposed prototype. The whole process involved establishing the requirements specification and determining the structural design of the prototype.

3.4 Requirements Engineering

The process of requirements specifications, review and analysis started by interviews and gathering data on what characterized fraud cases in the banking industry for the last five years. The major functional groups interviewed included information risk, operational risks and compliance risks departments, the technology teams supporting digital channels especially mobile banking platforms, the departments in charge of forensics investigations of fraud cases and the technology departments in charge of Technology Risks and Security. These groups were chosen

since they could give clear illustrations and documentations in regard to past mobile banking fraud episodes in the bank and to calibrate the parametric definitions of these cases to enable researcher to know specific features and attributes of fraudulent transactions. The said technology teams also shared the non-functional requirements that the system needs to fulfil. Since the proposed architectural design requires that all mobile banking transactions must pass through the CBR Fraud Detection system that would give of the fraud score, it was paramount that this does not affect the system performance and the envisaged customer experience for the mobile banking services. The output of this phase was a requirements specifications document that was ultimately reviewed across all the relevant stakeholders before signing off.

3.5 Study Population

A study population is defined by Lavrakas (2008) as any finite or infinite set of individual entities. A population applies to all objects in any area of investigation, according to Zikmund et al. (2010), and is referred as the “universe”. The study population consisted of data of past episodes of fraudulent transactions from KCB Bank Kenya. The selection of KCB Bank Kenya was informed by the fact that it’s the largest bank in East Africa by asset base and was one of the banks that was driving one of the highest mobile transaction volumes and hence by all probabilities, the propensity of occurrence of fraudulent transactions were high in addition to the fact that it was one of the banks which had the largest number of active customers in the banking industry. Additionally, the bank employees drawn from Risk, Forensics, IT Security and Digital Channels teams were interviewed since those teams were had the knowledge and information regarding the frauds that happened via the banks’ digital channels and could give a clear account and descriptions of various past fraud episodes.

3.6 Sample and Sampling Technique

To classify the sample units, the research used a purposeful sampling technique. Lavrakaz (2008) states that a purposeful sample is a form of non-probability sample, often referred to as a judgmental or expert sample. A purposeful sample's primary objective is to generate a sample that can reasonably be believed to be representative of the general population. This is also done by applying the populations expert's knowledge to pick a sample of elements that constitute a cross-section in a non-random manner. The sample units chosen consisted of the top 15 Mobile Banking transactions which had been classified as fraudulent and 15 transactions classified as non-

fraudulent in the previous year proceeding the year of this research. This constituted a sample of 30 transactions since we targeted to have a case library of around 300 transactions. The sample was therefore 10% of the target population. Gall and Borg (2007), agree that a ten per cent sample is adequate for a descriptive study. We then analyzed and generated data in form of transaction attributes and weighting matrix that were used to segregate fraudulent transactions from non-fraudulent ones. This was done in conjunction with the feedback from questionnaires that were distributed among various departments dealing with information risk, technology security and forensics within the bank using simple random sampling.

3.7 Research Instruments

The research used questionnaires to collect qualitative data about the fraudulent transaction attributes for review, as previously stated. This was further confirmed by the analysis of outcome of the secondary data collection and study. A survey questionnaire was used in this research because it offered an unobtrusive and economical data collection instrument (Zikmund, Babin, Carr, & Griffin, 2010). Secondary data consisting of the past episodes of frauds from KCB Bank Kenya was also collected from a cross section of sources including bank's forensic reports, bank's information systems audit reports and central bank's antifraud investigation reports. The aim of the approach of using both primary and secondary data to address the same research goals was to enhance interpretive coherence and bolster the communicative and pragmatic reliability of the findings of the study.

3.8 Data Collection Procedure

Data from both primary and secondary sources were collected with primary data obtained from questionnaires and secondary data from bank forensic and audit reports, including reports from the anti-fraud investigation unit of the Central Bank of Kenya (CBK). In order to obtain reliable results and improve the accuracy of the data obtained, the questionnaires were administered to the respondents by the investigator.

3.9 Conclusion

For far too long, many financial institutions have suffered from fraud schemes some of which they become aware much later after the events have occurred leaving them to write off a lot of losses. The current processes that these firms institute which relies on post analysis of system logs and reconciliation of the critical transactional accounts and are not adequate to fully combat or detect fraud in real time especially given the transaction volumes that characterize mobile banking

systems and the velocity of money thereof. Therefore, there was need to have a pragmatic approach and develop a system that would bring this close to a reality.

This project was therefore aimed at building a real time case-based reasoning system that relied on a knowledge base of cases – a problem situation and solution pair, that would be used to parametrically associate new transactions to determine whether these transactions are characteristically fraudulent or not. The system building approach enabled the system to be matured incrementally and then taken through a rigorous testing process including adapting new problems and updating the case library.

The completed system is expected to be rolled out to appointed banks to use it to assist in the detection fraud in the mobile banking space and hence be able to greatly reduce losses due to fraud going forward. The institutions would be required to use the experts and those personnel who understand the fraud behaviour of customers to continuously update the case library so as to increase the efficiency and the effectiveness of the system as fraudsters craft new schemes of fraud in a bit to circumvent the system.

CHAPTER FOUR: SYSTEM ANALYSIS, DESIGN & IMPLEMENTATION

4.1 Introduction

This chapter contains case-based reasoning prototype analysis, the prototype design and implementation using a set of technology tools and artifacts.

4.2 System Analysis

Requirements engineering phase revealed the following minimal functional and non-functional requirements that the system needs to fulfil to be able to meet the research objectives as outlined in this project.

4.2.1 Review of Existing Fraud Management Systems and Frameworks

The bank relies on an assortment of rule-based system definitions, custom system reports, system monitoring and system logs to be able to detect fraud. Most of the fraud detection and interventions happens after the fact which means that the bank acts to prevent further loss of funds after some frauds have already been successfully committed and funds already lost. Therefore, besides the fact that fraud detection not real time, the bank's action is also not automated and is only manually triggered through blacklisting of the potentially fraudulent customer profiles to prevent them from performing subsequent transactions.

4.2.2 Functional Requirements

These are the requirements describe how the system should behave under various conditions. They generally entail system functions and features. The functional requirements of the system to be implemented were as follows.

- i) Customer Profile Management - The system needs to implement a functionality for managing customer profiles including customer registration for mobile banking customers and ability to update customer profile. Customers important bio data such as customers names, identification numbers including National Identification or passport numbers, Gender, Date of Birth, Phone Number (MSISDN) and the IMSI ID for the current phone number should be captured and updated when necessary.
- ii) Transaction Authentication & Processing – The system should be able to authenticate all the transaction requests using the customers mobile phone number and Personal Identification Number (PIN). The system should also be able to process the transaction successfully and effect the necessity accounting entries upon meeting all the validation checks.

- iii) Messaging – The system should have ability to define message constructs for different transaction statuses and should be able to generate appropriate messages to be sent to the customers for statuses.
- iv) User Management – This includes ability to register users and various user groups, define the authorization levels and permissions for the users.
- v) Real Time Fraud Detection – This is the main functionality on the system in which case all transaction requests that meets a defined fraud score based on the transaction weighting attributes and the Euclidean metrics with reference to fraudulent cases in the cast base should be rejected.
- vi) Case Adaptation & Reuse – Being the last stages of the CBR Cycle, the system should provide a mechanism to allow for the adaptation and reuse of the fraud cases based on the verdict of the system and of the fraud experts.

4.2.3 Non-Functional Requirements

These are requirements that defines the operational or technical capabilities and constraints that should be built into the system for it to fulfil its functional requirements to the degree that is desired. They are sometimes referred to as system quality attributes. These are emergent properties that contributes towards meeting the overall functional requirements of a system.

- i) Performance – The system throughput should be of a minimum of 10 transactions per second with each transaction response time under 2 minutes.
- ii) Robustness – The system should be fault tolerant, able to handle and recover from faults. It should withstand stress and process large amounts of data in the case base without compromising on the transaction speeds and response times.
- iii) Reliability - The system should be highly reliable to guarantee transaction atomicity so that there are not transactions that are left in a transient state. The fraud score and Euclidean distance computations should be beyond precision with very consistency in its fraud detection with near zero-defect rate. The system should implement proper error and exception handling to be able to report any errors and exceptions correctly and timely.
- iv) Security – The integrations between different sub systems should be via secure protocols with public private key infrastructure implementations. The user credentials should also be stored in a secure hashed or irreversibly encrypted format. There system should also implement

proper user session management such that users' sessions are expired within 2 minutes of idle time.

- v) Maintainability - The system should be built in modularized and microservices fashion such that extending new functionalities should be easy and less disruptive. The components, design and modularized code implementation should also be reusable.
- vi) Portability - The system should be platform and operating system agnostic. It should be able to run on the common hardware platforms and operating systems like windows, Unix or linux.

4.2.4 Features Augmentation & Calibration

After questionnaire feedback from the fraud experts within the bank, the following transaction features were identified to be the key determinants of whether a transaction would be classified as fraudulent or otherwise. Analysis of the feedback data set from the questionnaire was confirmed to have internal consistency by calculating the Cronbach's alpha coefficient which were all greater than 0.70. The below table shows the summary of the calibrated features and comments and justifications for their relevance to the objectives of the study. The feedback from the fraud experts also guided on the diagnostic significance of each attribute since not all the attributes were considered to have the same weights towards the determination of a transaction to be classified as fraudulent or otherwise.

Domain	Attributes	Comment
Customer Bio Data	Gender, Age	- Younger persons up to 40 years are more likely to engage in fraud than the older populations.
Customer Behavioral Attributes	Account Opening timestamp, Mobile Banking Registration timestamp, Mobile Banking PIN change timestamp, Customer's Mobile Phone SIM Swap timestamp	- All these were all considered in relation to the date and time the customer did the transaction. - There was a tendency of fraud being committed within a short span after the said activities happens in the system.
Transaction Related Attributes	Transaction Amount, Transaction Type,	- Most of the fraudsters transact amounts that almost burst the ceiling

	Transaction Date	of allowed transaction thresholds per transaction or per day - Most of the fraudulent transactions are debits to other banks or to Mobile wallets
--	------------------	--

4.2.5 System Design

High Level Architectural Design

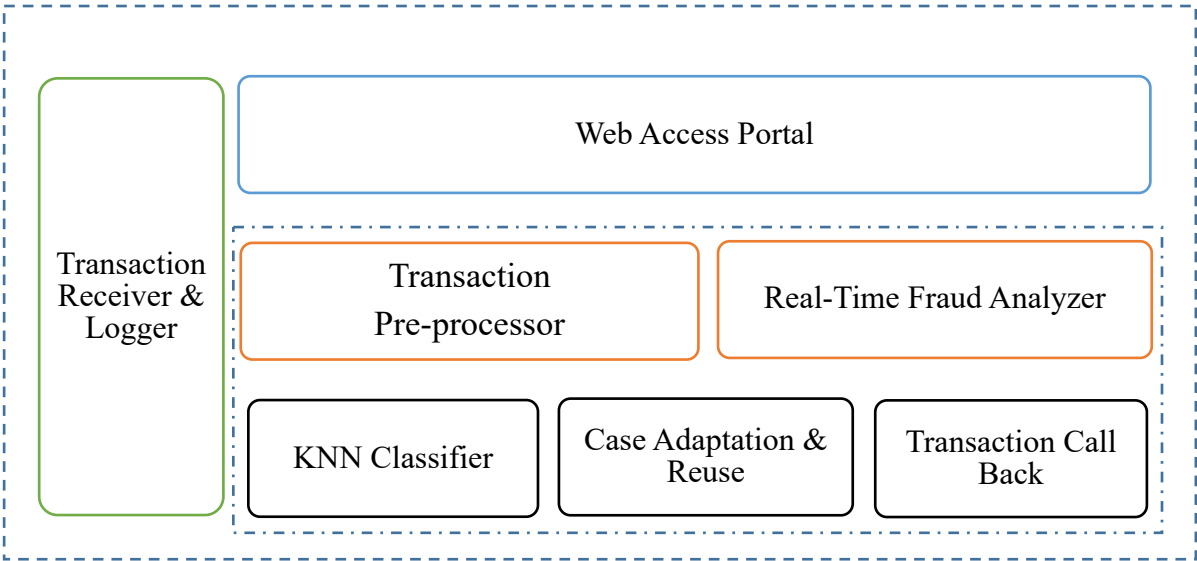


Figure 6: High Level Architectural Design for Real Time Fraud Detection System

Transactions Receiver & Logger: - This is a web service interface that is being invoked by the mobile banking system to pass on the transactions information. The interface receives the transaction and logs them into a processing queue.

Transaction Pre-Processor: - This is the sub-component that retrieves any new transactions from the processing queue and does the high-level validation of the transaction including web service payload parameter validation and the confirmation of all the mandatory transaction attributes.

Fraud Analyzer: - This is the sub-component that extracts the transaction attribute features scores based on a scoring matrix and computes the transaction aggregate score. The component also performs threshold retrieval of the all transactions whose score are within defined search region on the case base based on the calculated aggregate scores.

KNN Classifier: - This is the sub-component that applies the KNN algorithm based on a predefined K-value and a Euclidian distance metric to classify appropriately the transaction as either fraudulent or non-fraudulent. For cases where the classifier cannot appropriately do the classification, the case is marked for adaptation.

Case Adaptation & Reuse: - This is the sub-component that is meant to adapt the retrieved cases to attempt to classify the new case for instances where the retrieved case cannot be used to correctly classify the current case. The case adaptation mechanism implemented was through transformation in which case the system altered the retrieved case through addition, deletion or replacement of some parts of the retrieved cases to enable correct classification of the new instance using heuristic knowledge as supplied by the fraud experts. Where the retrieved cases were considered sufficient to classify the current instance, the system reused the past cases for classification.

Transaction Call Back: - This is the component that invokes a call back to the mobile banking system to either proceed to process the transaction for negative classification and to reject the transaction for positive classification.

Web Access Portal: - This is a web portal that provides a console for performing any system set ups and for viewing and monitoring transactions.

4.2.6 Database Design

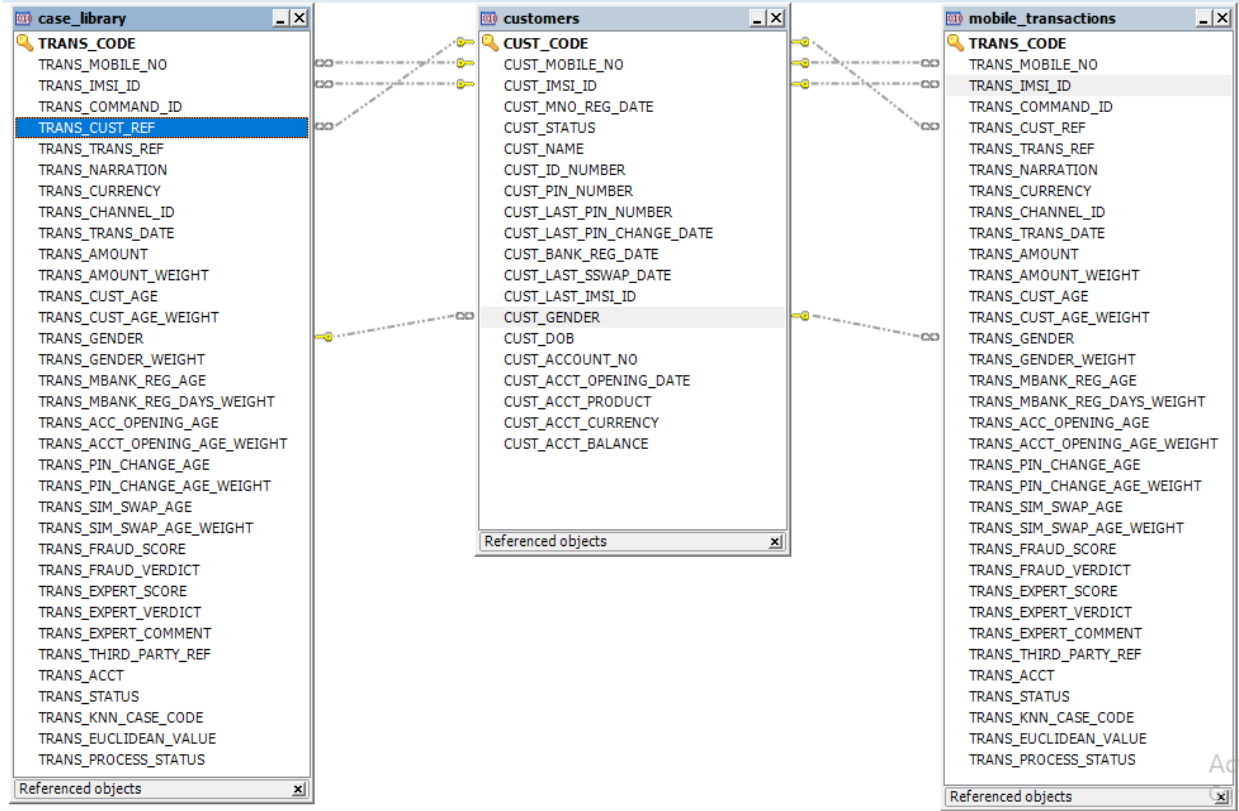
The above information from the questionnaire feedback was used to inform the kind of data which was collected from the past episodes of mobile banking fraud. A set of 30 sample data records from the past transaction were harvested & anonymized/depersonalized, 15 of which consisted of transactions which were rightly classified as fraudulent while the other 15 transactions were classified as non-fraudulent.

Based on the nature of this data, the data base design consisted of the database objects as below:

Table	Description
CUSTOMERS	Contains the Bio Data of mobile banking customers
CASE_LIBRARY	Contains the past episodes of mobile banking transaction classified as fraudulent or non-fraudulent
MOBILE_TRANSACTIONS	Entry table for all mobile transaction as performed by customers from time to time
ATTRIBUTE_WEIGHTING_MATRIX	Contains the key attributes and their discretized weights to be used to generate a fraud score for each mobile transaction.
FRAUD_SCORE_MATRIX	Contains the definitions of the ranges of fraud scores to help classify transactions appropriately.
MESSAGE_TEMPLATES	Definitions of the various message constructs to be used to form messages to be send to customer to inform them on the status of their transaction.
MESSAGES	Contains the actual messages as constructed by the system and queued to be send to the customers.

Entity Relationship Diagram (ERD)

The diagrams below the entity relationships for all the database tables.



weighting_matrix
WM_CODE
WM_ATTRIBUTE
WM_MIN_VALUE
WM_MAX_VALUE
WM_THRESHOLD_WEIGHT
WM_ADAPTATION_NO
WM_ADAPTATION_DATE
WM_STATUS
WM_DATE
WM_MVARIABLE_VALUE

fraud_score_matrix
FSM_CODE
FSM_FLOOR_VALUE
FSM_CEILING_VALUE
FSM_LABEL
FSM_COMMENT

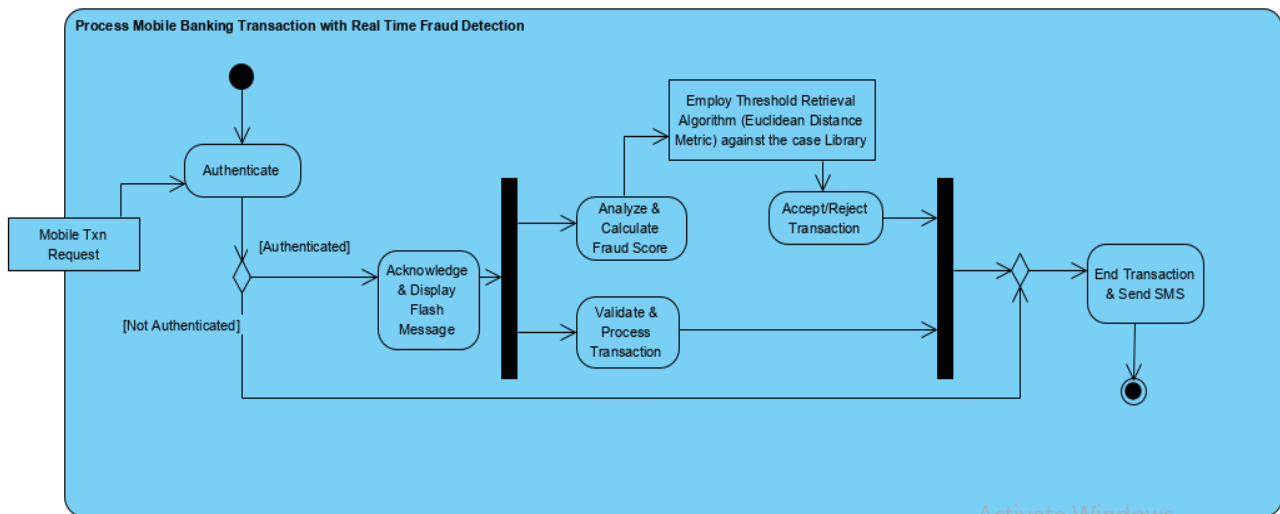
message_templates
MST_CODE
MST_MT_CODE
MST_NAME
MST_TEMPLATE
MST_STATUS
MST_DATE
MST_MODE

Referenced objects
messages (MSG_MST_CODE)

messages
MSG_CODE
MSG_MST_CODE
MSG_MESSAGE
MSG_SOURCE
MSG_DEST
MSG_STATUS
MSG_PRIORITY

4.2.7 Activity Diagram

The activity diagram below depicts the flow of customer mobile transaction request till its end state which can either be approved or rejected state based on the decision from the fraud analyzer. The transaction request is first authenticated based on the mobile number and associated PIN number stored on the bank's mobile banking platform. On successful authentication, the request is admitted and queued for processing otherwise the request is rejected, and appropriate SMS sent to the customer. In the processing queue, the transaction is picked by a pre-processor engine which performs the necessary validation on the transaction against other business rules including maximum transaction amount per transaction per day per transaction type, whether there are other financial transactions in queue and account balance validation. In parallel, the real time fraud analyzer engine also picks this transaction and calculates a fraud score based on the customer profile and the transaction attributes using the defined attributes weighting. The fraud analyzer then calculates the Euclidean distance between the current transaction fraud score and the fraudulent transactions in the case base. Based on the Euclidean distance metric, transaction is labeled appropriately as either fraudulent or non-fraudulent. If the transaction is frequent, it's rejected, and rejection SMS send to the customer otherwise the transaction is approved, and appropriate accounting entries applied.



4.2.8 System Development & Implementation

The necessary database objects including tables and procedures were created. The system developed was a 3-tier consisting of MySQL database, the background processing java demons and a front-end web application build on Oracle ADF framework and deployed on Oracle WebLogic server.

The data for the past mobile fraud episodes were loaded into the case library table using database script and indexed appropriately. Using the feedback from the fraud experts, the attribute weighting definitions based on their diagnostic significance and the transaction fraud score ranges as derived from the past classifications were fed into the system through the web interface as below:

Attribute	Min Value	Max Value	Weight
ACCT_OPENING_AGE_DAYS	0	7	0.3
ACCT_OPENING_AGE_DAYS	8	100000	0.2
AGE	18	40	0.3
AGE	41	65	0.2
AGE	66	120	0.1
GENDER	MALE	MALE	0.2
GENDER	FEMALE	FEMALE	0.1
MBANKING_REG_AGE_DAYS	0	7	0.3
MBANKING_REG_AGE_DAYS	8	1000000	0.1
PIN_CHANGE_AGE_DAYS	0	7	0.4
PIN_CHANGE_AGE_DAYS	8	1000000	0.2
SIM_SWAP_AGE_DAYS	0	7	0.4
SIM_SWAP_AGE_DAYS	8	1000000	0.2
TRANS_AMOUNT	69000	1000000	0.2
TRANS_AMOUNT	1	68999	0.1

Figure 7: Attribute Weighting Matrix Definition

Floor Value	Ceiling Value	Fraud Label	Comments
0	1.19	NEGATIVE	NEGATIVE
1.2	3	POSITIVE	POSITIVE

Figure 8: Fraud Score Matrix Definition

Data Preprocessing

Most machine learning paradigms basically utilize statistical relevance of the input data for their inference. It therefore means that for successful machine learning outcomes, the data being used to create the prediction model must be relevant and less noisy. It is therefore important to preprocess such data, in as much as the process may take quite some time to complete. The main activities that data preprocessing involves include: data cleaning, data integration, extraction of attributes and selection (Han J, Kamber M, 2006).

For this study, a lot of preprocessing of the case library data was undertaken to clean it up and remove any irrelevant information from the data set before the data was made fit for use. The mobile transactions data was collected from an enterprise data warehouse spanning for a period of five years. The data was a lot and contained some irrelevant attributes such as branch codes, mobile banking menu profile codes, and outstanding mobile loan amounts amongst others. Such attributes which were irrelevant for the purpose of this study were weeded out from the data set using database PLSQL scripts. Additionally, the data was also formatted into csv files in readiness for loading into the target database.

Case Stabilization and Indexing

After loading the case library data, the next step was to analyze the data to identify the fraud patterns and correctly label the fraudulent and non-fraudulent transactions. This was done with the help of the fraud experts. Additionally, as the recalibration of the data sets was being done, the transaction attribute weighting matrix was correspondingly also being adjusted appropriately based on the original deduction and classification of the transactions. At the end, all the data was correctly labeled and the final correct attribute weighting and fraud score matrices were arrived at at least based on the sample data loaded into the case library. These matrices are the ones that were target to be used in the simulation of the real-time fraud detection during the system experimentation and evaluation stage. At the system evaluation stage, there would still be a chance for readjustment of the attribute weighting and the score matrices based on the simulated transaction data. The diagram below shows a sample screen shot of the case library from the application web console.

Trans Reference	Customer No	Currency	Trans Date	Amount	Account No	Fraud Score	Fraud Verdict	Status	transAmountWeight	transGenderWeight	transGender
TR16195LDGHJ	8425855	KES	13-Jul-2016	26	158280571	31	1	BLOCKED	2	4	MALE
TR16195YXWPG	8889419	KES	13-Jul-2016	12000	165146303	29	1	BLOCKED	2	2	FEMALE
TR161970GM06	5165452	KES	13-Jul-2016	1000	171182707	31	1	BLOCKED	2	4	MALE
TR161970SNLF	9749238	KES	13-Jul-2016	70000	168295490	32	1	BLOCKED	5	2	FEMALE
TR161958M2GK	6624915	KES	13-Jul-2016	70000	168257440	34	1	BLOCKED	5	4	MALE
TR16195X9KY7	2692319	KES	13-Jul-2016	50	158269438	29	1	BLOCKED	2	2	FEMALE
TR16197NHKWW	8223866	KES	13-Jul-2016	229	160953545	31	1	BLOCKED	2	4	MALE
TR16197VVBYS	1273334	KES	13-Jul-2016	10000	170804798	29	1	BLOCKED	2	2	FEMALE
TR16197MB7XC	6370481	KES	13-Jul-2016	12000	169558096	31	1	BLOCKED	2	4	MALE
TR16197R0LHT	8979953	KES	13-Jul-2016	1128	159592046	29	1	BLOCKED	2	2	FEMALE
TR16195P2ZHX	1475416	KES	13-Jul-2016	10000	164465847	17	0	APPROVED	2	4	MALE
TR161962570J	4425386	KES	13-Jul-2016	2350	157665089	15	0	APPROVED	2	2	FEMALE
TR1619707HRF	4425386	KES	07-Mar-2014	5000	167519468	15	0	APPROVED	2	2	FEMALE
TR16197LT4WP	2746470	KES	07-Mar-2014	100	158775805	15	0	APPROVED	2	2	FEMALE
TR1619751711	6708097	KES	07-Mar-2014	70000	169669807	20	0	APPROVED	5	4	MALE
TR161975QWZB	9100777	KES	07-Mar-2014	70000	169660249	18	0	APPROVED	5	2	FEMALE
TR16196CMBPS	5536826	KES	07-Mar-2014	5000	166970522	17	0	APPROVED	2	4	MALE
TR161955QDZH	2885316	KES	07-Mar-2014	1000	166435961	15	0	APPROVED	2	2	FEMALE
TR16195N7WTS	4618848	KES	07-Mar-2014	450	157070034	17	0	APPROVED	2	4	MALE
TR16197Z6L8	10333803	KES	07-Mar-2014	300	157047490	15	0	APPROVED	2	2	FEMALE
TR16197MRHHC	332605	KES	07-Mar-2014	70000	169353133	18	0	APPROVED	5	2	FEMALE
TR16182CM06L	9334679	KES	07-Mar-2014	40000	172568951	15	0	APPROVED	2	2	FEMALE
TR16197QZFL	736262	KES	07-Mar-2014	50000	169190730	17	0	APPROVED	2	4	MALE
TR16195T82G5	6704717	KES	07-Mar-2014	40000	165793457	15	0	APPROVED	2	2	FEMALE
TR16197T36HK	6432424	KES	07-Mar-2014	47	157031241	17	0	APPROVED	2	4	MALE

Figure 9: Sample view of Case Library

CBR Real Time Fraud Detection Engine

The CBR Fraud Analyzer and predictor was developed as a multi-threaded engine using Java JPA that employs the use of executor services using the producer-consumer principle. The multithreading concept of up to 100 concurrent threads was meant to achieve transaction parallelism and concurrency. This enables the system to process up to 100 transactions per second and hence able to efficiently handle the sheer velocity and volumes of mobile banking transaction thereby not compromising on the transaction lifecycle speeds and therefore customer experience. Additionally, the architecture of the implementation is such that all the customer requests are received and acknowledged immediately but the requests are processed asynchronously hence no effect on the customer experience. The fraud analyzer producer listens for any mobile transaction requests, immediately picks and puts them into a shared memory structure, a linked blocking queue. The fraud analyzer consumer engine works by picking any requests present in the queue and first does the necessary validations including confirming sufficiency of funds in the customer’s account, posting restrictions, and maximum transaction limits and daily transaction thresholds. Upon passing the validations, the consumer then uses a threshold retrieval to select the ten nearest neighbours based on an aggregate fraud score of the transaction attributes after which it applies a

standard CBR classification technique to classify the transaction as either fraudulent or non-fraudulent. It classifies new instances by retrieving similar cases from the case library using k-nearest neighbour (kNN) algorithm. The kNN works by defining how the case is represented in the case base and the associated similarity function which basically utilizes algorithms for transaction attributes selection based on specific weighting matrix. Weighted Euclidean distance metrics are performed to measure the similarity between a new instance and previously encountered cases.

$$d = |Z - X| = \sqrt{\sum_{i=1}^m w_i |Z_i - X_i|^2}$$

Where:

w_i is the weight of transaction attribute i ,

Z is the new instance,

X is the retrieved case from the case base,

m is the number of transaction attributes in each instance, and

i is an individual transaction attribute from 1 to m .

This algorithm works by selecting K neighbouring cases for the new instances and classifies the new instance as fraudulent or non-fraudulent based on the class of its nearest neighbours. Therefore to assign a transaction class K to any new transaction Z_i , the new transaction is compared to all the cases in the case library using the similarity measure d .

The similarity between two instances is defined as a weighted average of transaction attribute similarities, since different attributes are of different diagnostic importance. In this case the consumer calculates the transaction attribute (vector) weights based on the defined weighting matrix to get an aggregate weighted score which it then used to retrieve K most similar cases as an ordered list.

The resultant similarity value is therefore calculated by using the weights of each transaction attribute (v_i).

Therefore, the Euclidian distance for each pair of transaction attributes $[v_i^z, v_i^x]$ is calculated whose resulting value is assigned a weighting (w_i) which represents the relevance of the corresponding transaction in the overall similarity computation.

$$Sim(Z, D_x) = \sum_{j=1} w_i * Sim_i(\overline{v_i^z}, \overline{v_i^x})$$

To arrive at the solution for a new instance (Z), the system applies a weighted voting schema against the retrieved cases to arrive at one of the similar cases to be takes as the solution to the problem situation. Alternatively, the system selects the case with the least Euclidean distance if $K=1$. Using a scoring function:

$$score(p_i) = \sum sim(Z, S_x) \forall x | S_x = p_i$$

Consequently, the solution to the new instance is:

$$p_i = argmax\{score(p_i), i = 1, \dots, k\}$$

Where k is the value of the instance of the nearest neighbour assigned by the kNN algorithm during case retrieval. The diagram below gives a view of sample test transactions after the CBR Detecting Engine processing and labeling of those transactions.

Mobile Transactions											
New Edit Delete											
View Detach											
Trans Reference	Customer No	Currency	Trans Date	Amount	Account No	Status	Fraud Score	Fraud Verdict	Amount Weight	Age Weight	Gender Weight
TR16195LDGHJ	8425855	KES	28-May-2020	30026	158280571	APPROVED	17	0	2	4	4
TR16195YXWPG	8889419	KES	28-May-2020	42000	165146303	APPROVED	15	0	2	4	2
TR161970GM06	5165452	KES	28-May-2020	31000	171182707	APPROVED	17	0	2	4	4
TR16197QSNLF	9749238	KES	28-May-2020	70000	168295490	APPROVED	18	0	5	4	2
TR161958M2GK	6624915	KES	28-May-2020	70000	168257440	APPROVED	20	0	5	4	4
TR16195X9KY7	2692319	KES	28-May-2020	30050	158269438	APPROVED	15	0	2	4	2
TR16197N4KWW	8223866	KES	28-May-2020	30229	160953545	APPROVED	17	0	2	4	4
TR16197VTBY5	1273334	KES	28-May-2020	40000	170804798	APPROVED	15	0	2	4	2
TR16197MB7XC	6370481	KES	28-May-2020	42000	169558096	APPROVED	17	0	2	4	4
TR16197R0LHT	8979953	KES	28-May-2020	31128	159592046	APPROVED	15	0	2	4	2
TR16195P2ZNX	1475416	KES	28-May-2020	40000	164465847	APPROVED	17	0	2	4	4
TR161962570J	4425386	KES	28-May-2020	32350	157665089	APPROVED	15	0	2	4	2
TR16197D7HRF	4425386	KES	28-May-2020	35000	167519468	APPROVED	15	0	2	4	2
TR16197LT4WP	2746470	KES	28-May-2020	30100	158775805	APPROVED	15	0	2	4	2
TR16197S1711	6708097	KES	28-May-2020	70000	169669807	APPROVED	20	0	5	4	4
TR161975QWZB	9100777	KES	28-May-2020	70000	169660249	APPROVED	18	0	5	4	2
TR16196CMBP5	5536826	KES	28-May-2020	35000	166970522	APPROVED	17	0	2	4	4
TR16195SQDHz	2885316	KES	28-May-2020	31000	166435961	APPROVED	15	0	2	4	2
TR16195N7WTS	4618848	KES	28-May-2020	30450	157070034	APPROVED	17	0	2	4	4
TR16197VZ6L8	10333803	KES	28-May-2020	30300	157047490	APPROVED	15	0	2	4	2
TR16197MRHHC	332605	KES	28-May-2020	70000	169353133	APPROVED	18	0	5	5	2
TR16182CM06L	9334679	KES	28-May-2020	70000	172568951	APPROVED	18	0	5	4	2
TR16197QZFL	736262	KES	28-May-2020	70000	169190730	APPROVED	20	0	5	4	4
TR16195T82GS	6704717	KES	28-May-2020	70000	165793457	APPROVED	18	0	5	4	2
TR16197T36HK	6432424	KES	28-May-2020	30047	157031241	APPROVED	17	0	2	4	2

Figure 10: Sample view of Mobile Transactions & Associated Classes

The last two stages in the Case Based Reasoning cycle of revision and retention is performed by the fraud and forensics experts after inspecting the system fraud detection capability and classification precision from the retrieved cases and considering the transaction in view. The experts can perform parametric adjustments on the weighting and fraud scoring matrices after which such cases can be retain in the case base for future fraud detection.

4.3 Contextual Application of the CBR Real Time Fraud Detection System

To make use of the developed Real Time Fraud Detection system in the industry, the would have to be integration through an integration service bus in such a way that any mobile transaction is routed to the Fraud Detection System to be processes and give a verdict before the transaction cab be processed and approves by the mobile banking and then subsequently routed to the core banking system. The transaction shall be processed asynchronously such that the request is received and acknowledged by the mobile banking system and an acknowledged SMS sent to the customer awaiting the full transaction cycle processing after which the customer will get a final transaction rejection or approval SMS for the transactions classified by the fraud detection system as fraud or non-fraudulent respectively. The diagram below shows the context diagram depicting all the interrelated systems and the interaction to the Real Time Fraud Detection system.

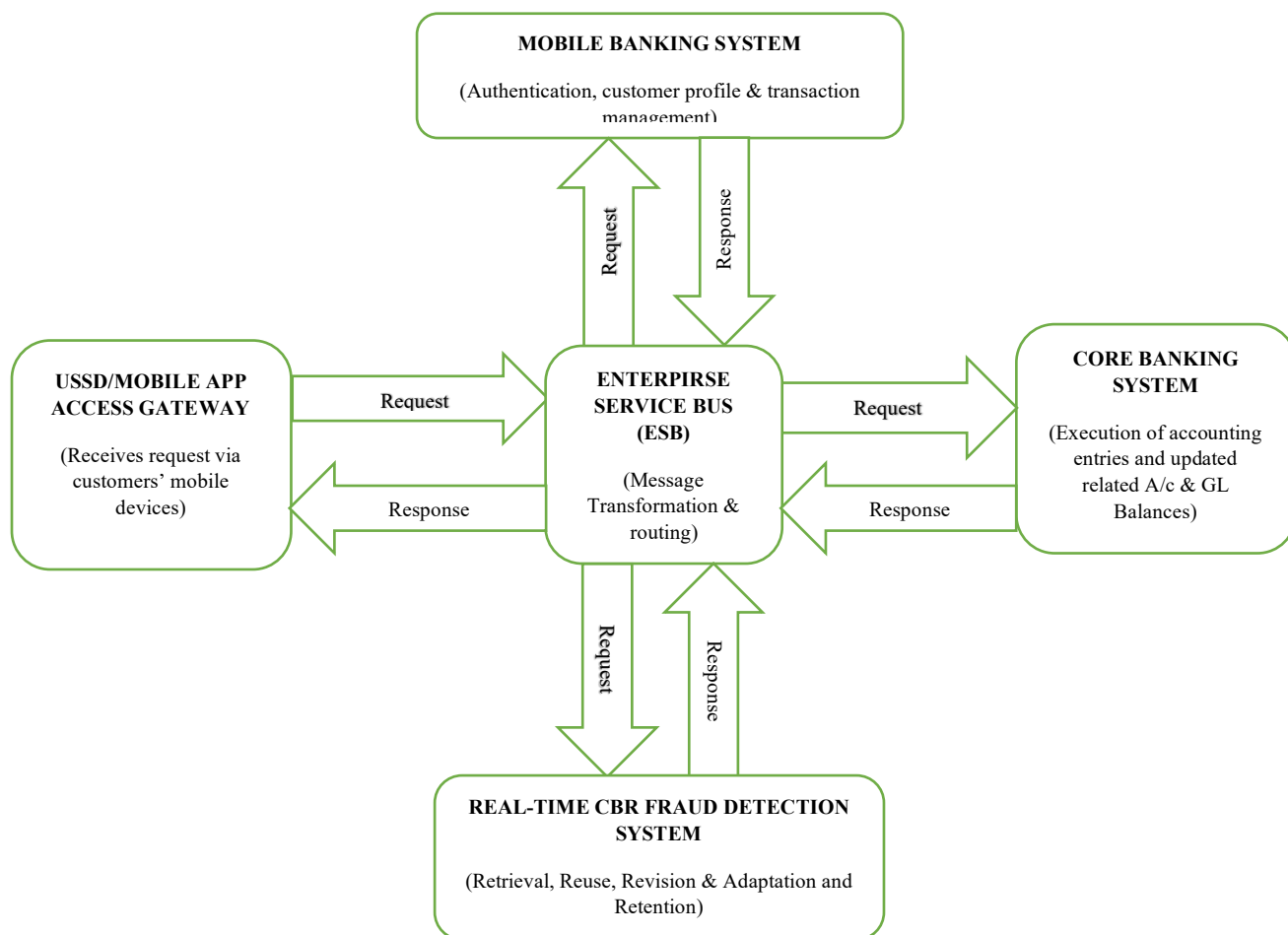


Figure 11: CBR Real Time Fraud Detection System Context Diagram

USSD/Mobile App Access Gateway

The Unstructured Supplementary Service Data (USSD) is a Global System for Mobile (GSM) communication technology used to send messages between a mobile phone and network application program. Banks publish their services via telco’s specific USSD codes via which customers can transact once they are authenticated into the target systems – in this case mobile banking systems. The customers can also access the same set of banking services deployed and published on mobile apps across different mobile operating systems including Android, windows and IOS phones.

Mobile Banking System

The Mobile Banking System hosts all mobile banking customer profiles and authorized transacting accounts. It also has customer authentication information like PINs and manages customer access sessions to all deployed services. This is the central system that the customers interact with and is where all the financial services definitions (services packages) are done per group of customers such that there can be a group of customers who have access to certain bank services which differs from another set of customers.

The Core Banking System (CBS)

The core banking system is where the GL entries are booked for every transaction. All transaction done on the mobile banking platform are translated into accounting entries which are completed in the CBS and customer account balances updated accordingly.

The Enterprise Service Bus

The Enterprise Service Bus (ESB) is an orchestration layer in the ecosystem. It acts the message transformation and routing middleware such that all transaction message interchanges that happen in the ecosystem happen via the ESB. The ESB has the capability to receive messages in various protocols including SOAP, REST (JSON), and ISO, interprets and uses the business and technical rules to transform the message into message structures and protocols acceptable by the target systems.

The CBR Real-Time Fraud Detection System

The is the system that would receive and intercept all transactions requests just before authorization to determine whether the transactions are fraudulent or not. For each transaction, the Mobile Banking system would invoke the Fraud Detection System to get a score and determine whether it would authorize a transaction or not. It harbours a case library of the past fraud incidences and uses that as reference to retrieve similar cases or adapt to new cases as and when requests are made. The system shall have an agreed weighting matrix of all transaction attributes of diagnostic importance, a configuration which can always be varied as the system gets adapted to the environment.

CHAPTER FIVE: SYSTEM EVALUATION AND ACHIEVEMENTS

5.1 Introduction

This chapter contains a review of the performance evaluation measures, the results and performance measures of the testing cycles, the discussions of the results and the extent of achievements of the objectives of the study as per the results obtained.

5.2 Performance evaluation measures

There are different measures used to determine or evaluate the performance of classifiers. The measurements of precision, accuracy, and recall were used in this research. The accuracy of the classifier, refers to the ratio of the correctly classified test instances to the total number of instances being evaluated and is typically calculated to determine the classifier's overall performance as follows:

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test samples}}.$$

The performance of each class in a dataset is evaluated by determining the precision and recall measures. Precision (or positive predictive value) is the percentage of relevant retrieved cases, while recall (or sensitivity) is the percentage of retrieved relevant cases. By calculating the following, these measures can be obtained:

1. **True positive (TP):** This denotes the total number of correctly classified episodes or samples of a specific class. In this research, this will denote the number of transactions correctly classified as fraudulent.
2. **True negative (TN):** This denotes the total number of correctly classified episodes or samples not belonging to the specific class. In this research, this will denote the number of transactions correctly classified as non-fraudulent.
3. **False positive (FP):** This denotes the total number of episodes or samples incorrectly assigned to the specific class. In this research, this will denote the number of transactions incorrectly classified as fraudulent.

4. **False negative (FN):** This denotes the total number of episodes or samples incorrectly assigned to another class. In this research, this will denote the number of transactions incorrectly classified as non-fraudulent.

The precision and recall of a multi-class classification system are defined by,

$$\text{Average Precision} = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FP_i},$$

$$\text{Average Recall} = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FN_i},$$

Where:

- N - Is the number of classes (in this case, 2 classes i.e. Fraudulent and Non-Fraudulent);
- TP_i - Is the number of true positive for class i ;
- FN_i - Is the number of false negative for class i and
- FP_i - Is the number of false positive for class i .

The confusion matrix can always be used to derive these performance measures. The confusion matrix shows a matrix of the predicted and actual classifications. The matrix is $n \times n$, where n is the number of classes.

Generally, the structure of confusion matrix for multi-class classification is given by,

	Predicted Class			
	Classified as c_1	Classified as c_{12}	...	Classified as c_{1n}
Actual Class c_1	c_{11}	c_{12}	...	c_{1n}
Actual Class c_2	c_{21}	c_{22}	...	c_{2n}
...
Actual Class c_n	c_{n1}	c_{n2}	...	c_{nn}

This matrix reports the number of false positives, false negatives, true positives, and true negatives which are defined through elements of the confusion matrix as follows,

$$\begin{aligned}
TP_i &= c_{ii} \\
FP_i &= \sum_{k=1}^N c_{ki} - TP_i \\
FN_i &= \sum_{k=1}^N c_{ik} - TP_i \\
TN_i &= \sum_{k=1}^N \sum_{f=1}^N c_{kf} - TP_i - FP_i - FN_i
\end{aligned}$$

Accuracy, precision and recall measures will be calculated for the KNN classifier using Euclidian distance similarity metric for all the test datasets.

5.3 Experimental Results and Discussion

More generally, the k nearest neighbours classifier maps any feature vector to the pattern class that appears most frequently among the k nearest neighbors (Kulkarni et al., 1998). The nearest neighbor is determined by distance metric function. The k nearest neighbours classifiers performance mainly depends on three factors: k value of the number of neighbors, distance metric and decision rule. One of the determinants of the performance of the KNN classifier is the choice of K. As the rule of the thumb, choosing the value of K is $K = \sqrt{n}$, where n stands for the number of samples in your training dataset. For large case bases, this can lead to computational complexity and hence is prohibitively expensive. But in practice, K should be large enough to minimize the classification error rate but not too large to avoid over-smoothed boundaries, small enough so that only nearby samples are included but not too small to avoid noisy decision boundaries.

Heuristic techniques such as cross-validation can be used in selecting a good value for k (Khuman, 2012) but if k is even, it would be necessary to define an auxiliary procedure to handle ties. However, Cover and Hart (1967) observed that for some distributions, $k = 1$ is optimal for all numbers, such that in nearest neighbor algorithm, the nearest neighbor label of X is assigned. Indeed, Devroye et al. (1996) argued further that, when the expected posteriori probabilities of error is small, there is little advantage in choosing k larger than 3. Therefore, generally, the k value is a relatively small integer. The cross validation and system evaluation process was done jointly

with the selected information risk analysts and forensic experts. The process of cross-validation with a selected data set from the universal data set to be used for testing the KNN classifier. Using a universal data set of 300 transactions, 60% of the data set was apportioned for cross validation with 40% of the data sets apportioned for testing and evaluation of the model. The 60% and 40% data set split was done through a random procedure. As part of cross validation, an approach to smoothen the decision rule, reduce the computational complexity and fasten the classification algorithm was further defined. This smoothening approach made reference to Kim and Park (1986), who proposed a fast nearest neighbor finding algorithm based on the ordered lists of the training samples of each projection axis. The ordered partition contained two properties, one is ordering to bound the search region and the other is partitioning to reject the unwanted samples without actual distance computations. Similarly, researchers have proposed partitioning the space into regions (Knorr and Ng, 1999; Ramaswamy et al., 2000) which allows for fast determination of nearest neighbors. Therefore, during the cross validation, the research introduced a precursor process for threshold retrieval based on weighted aggregate transaction fraud scores to restrict the nearest neighbor within a partitioned search region as subsequently explained in the forgoing sections. The search region was varied from 30 to 10 nearest case instances with each retrieval iteration decrementing this value by one. Working from the initial rule of thumb where $K = \text{sqrt}(180) = 13.4$, and as the final step of cross validation to predict the label for every instance in the validation set, the research used integer values of K from 1 to 13, noting what value of K gives the best performance on the validation set. Therefore, the resultant K value picked was 1 which gave the best classification accuracy of 87.40% with a threshold retrieval search region of 10 instances, while restricting the search region) whereas a K value of 13 gave the lowest classification accuracy of 46.60%, with a threshold retrieval search region of 30 instances.

During the testing phase, a transaction data set of 120 transactions were tested in three cycles. Each test cycle having a batch of 40 transactions. In each iteration, the data sets were normalized (using min-max normalization) to ensure conformity to Gaussian distribution in regard to each transaction attribute critical to classification the transactions. Each data set consisted of the transaction reference, customer mobile number, SIM card IMSI Id, Transaction type, customer reference, currency, transaction date and amount. The rest of the other critical attributes including the last SIM Swap Age, Customer Age, last PIN Number Change Age, Account Opening Age and Mobile

Banking Registration Age were derivatives generated by the Real Time Fraud Analyzer as a precursor process before the application of the KNN algorithm for classification of the transactions. The table below shows the summary of the performance of the Real Time Fraud Detection Engine for the three cycle of tests.

PERFORMANCE								
	TOTAL	TP	TN	FP	FN	ACCURACY	AVG. PRECISION	AVG. RECALL
<i>ITERATION 1</i>	40	4	31	2	3	87.50%	33.33%	28.57%
<i>ITERATION 2</i>	40	7	25	5	3	80.00%	29.17%	35.00%
<i>ITERATION 3</i>	40	5	29	4	2	85.00%	27.78%	35.71%
<i>Cumulative Average</i>						84.17%	30.09%	33.10%

Table 8: Results and Performance of The Fraud Detection Engine

The Confusion matrix for the tested data sets is a depicted below.

CONFUSION MATRIX			
		PREDICTED AS FRAUDULENT	PREDICTED AS NON-FRAUDULENT
ITERATION 1	ACTUAL FRAUDULENT	7	3
	ACTUAL NON-FRAUDULENT	33	31
ITERATION 2	ACTUAL FRAUDULENT	10	3
	ACTUAL NON-FRAUDULENT	30	25
ITERATION 3	ACTUAL FRAUDULENT	7	2
	ACTUAL NON-FRAUDULENT	33	29

Table 9: Confusion Matrix for Results and Performance of The Fraud Detection Engine

It's important to reiterate that the engine used seven transaction attributes each of which had different diagnostic importance, and hence each attribute was assigned weights ranging from 0.1 to 1 as was advised by the fraud experts and also derived from cross validation with the training

data sets where the attributes weights were continuously adjusted to achieve the least misclassification error. Zhao and Chen, 2016 confirmed that attribute Weighted KNN algorithm achieves better classification accuracy compared to non-weighted or instance weighted ones except for the algorithm runtime cost, which has been handled by the bounded search region. The Fraud Detection Engine did the classification in a two-step approach. In the first instance, the engine computed the aggregated sum of all weights of the seven attributes to arrive at an aggregated fraud score (FS). The engine then used threshold retrieval mechanism to retrieve maximum of ten cases from the case base within an interval between *FS-X and FS+X*, where the modifier **X** was given a value of 0.3. The decision to consider only up to ten cases from the case library was basically to limit the region of space for the subsequent application of the KNN classification algorithm with the main driver being to reduce the runtime cost thereby improving the efficiency and speed of the classifier. The modifier of 0.3 was arrived at after executing several training tests and cross-validations with the case base, a process which gave this modifier as the most optimal if the threshold retrieval query is to pick approximately 10 cases from the case base for consideration.

After the retrieval of the ten cases, the engine then calculated the Euclidean distance between the ten cases and the new instance for all the seven attributes after which the Euclidean distances obtained are ordered from the least to the largest. Since the engine is using KNN Algorithm where $K=1$, it picked the case with the least Euclidean distance and uses its label to classify the current instance.

From the analysis the Real Time Fraud Analyzer performed above average with an average accuracy of 84.17%. Precision and recall were however average. These measures can be improved by using large amounts of data to train the classifier before application to or testing with real data sets for classifications.

5.4 Discussions on the Objective's Achievement

In this section, we discuss the findings of the research and how it is related to objectives of the study.

Objective One

The first objective was to investigate and document past incidences of mobile banking fraud. The research achieved this through scanning through past forensic documents and fraud records and establishing the nature and characteristics of such frauds. The information gathering was aided by

the fraud and forensics experts. The fraudulent transactions were retrieved from the enterprise data warehouse where inspections of such transactions were done.

Objective Two

The second objective was to establish relevant attributes or features that are useful for classifying transactions as fraudulent and not fraudulent.

This was achieved through interviews and information collection via questionnaires from the fraud and forensic experts. Through this process, the research arrived at seven key transaction and customer behavioral attributes which were further validated through the inspection of the past fraud incidences during the case stabilization phase of the research.

Objective Three

The third objective develop a high-level architecture of Real Time Fraud Detection System.

A high-level solution architecture was developed taking cognizance of the reliability, the much-envisaged real-time fraud detection speed and accuracy, the desired transaction processing speed and atomicity, fault tolerance and desired scalability of the system. The architectural components comprised the right building blocks best fit to achieve the desired solution as per the functional and non-functional requirements.

Objective Four

The other objective was to develop a Case Based Reasoning Engine prototype for real time fraud detection.

From the architectural design and the requirements engineering output, a system prototype was developed. The prototype was able to accept a mobile banking transaction, do high level validations and perform fraud analysis using threshold retrieval and K nearest neighbour algorithm to classify a transaction as either fraudulent or non-fraudulent based a case base and a fraud scoring matrix.

Objective Five

The fifth objective was to populate the CBR Engine with cases of past fraudulent transactions to develop a case library (with a target threshold of cases) then test it with new test cases. Record test results and draw conclusions.

After transaction features calibration process, an appropriate data model was developed to inform the structure of the date of past fraud incidences to retrieved from the enterprise data warehouse. The retrieved data was then the loaded into the system's case base via SQL scripts and re-validated. Cross validation and system evaluation was done to measure the efficacy of the system, which was done in three iterations recording an acceptable classification accuracy of 84.17%. The information risk analysts and forensic experts also gave their affirmation of the effectiveness of the system being within acceptable thresholds.

From the discussions of the objectives above in terms of their achievement, it is clear that the research and the system prototype developed was able to achieve the main goal set for the study. The prototype was able to detect fraudulent transactions in real time with an acceptable performance accuracy without degraded transaction processing speeds.

CHAPTER SIX: CONCLUSION AND RECOMMENDATIONS

6.1 Introduction

This chapter captures the conclusion of this study as well as the areas that need further investigation in case-based reasoning real time fraud detection. It also further gives guidelines and framework for future implementation of such systems.

6.2 Conclusion

In Big Data, data science and machine learning, classification is a significant challenge. One of the oldest, easiest and most reliable algorithms for pattern classification and regression models is the K-nearest neighbor (KNN). In data mining, KNN has been listed as one of the top ten methods. One of the simplest and most widely used classifiers is the K-nearest neighbor (KNN) classifier, but its efficiency competes with the most complicated classifiers in the literature. This classifier fundamentally depends on the distance or similarity between the cases and the input samples.

Through the study, it was underscored that most financial institutions have not but in the right instrumentations and framework to proactively detect and deal with frauds leaving most of them vulnerable to the fraud stars and having to act after the fact hence having to contend with huge financial losses. This is one of the imperatives that drove this research to be able to institute a framework that would allow real time fraud detection. In the development of the prototype, it was clear that the resultant solution should not in any way affect the transaction processing time so as not to have any negative impact on the overall customer experience. The system was therefore design and developed with that in mind while ensuring that the classification accuracy of the system is maintained above the acceptance thresholds.

In this review, the performance (accuracy, precision and recall) of the threshold retrieval and KNN classifier has been evaluated using Euclidian distance measures, in an attempt to appropriately classify transactions as either fraudulent or non-fraudulent. The research modified the decision rule to include threshold retrieval mechanism so as to fasten the classification algorithm. After running several test iterations, classifier performance averaging at 84.17% was found to be at acceptable level. The performance of KNN classifier depended significantly on the retrieval mechanism used, and the choice of K. For example, the research found that found that using large

partitioned search regions for retrieval mechanism with large K values resulted into lower performance compared to smaller search regions and lower values of K.

6.3 Recommendations

With the background of this research study and the learnings, the following are the recommendations that should inform any future research studies of similar nature.

1. The study only picked on KNN classification algorithm with Euclidian distance choice of similarity measure. However, there are many other distances and similarity measures that are available in the machine learning that can be used and evaluated comparatively for optimal performance with and without noise.
2. Additionally, other than KNN Classification there are other machine learning algorithms including support vector machines, neural networks or KNN Algorithm coupled with Genetic Algorithm for cross validation can be evaluated so as to apply the most appropriate algorithms with maximum performance and high precision.
3. The study datasets of 300 might not be enough to draw significant conclusions in terms of the effectiveness of the classification algorithm, and therefore, there is a need to use larger datasets with sufficient training samples.
4. While there are significant dividends of using large data sets and case bases, it's important for the researcher to work out ways on how to minimize the algorithm runtime costs while minimizing misclassification error.
5. The training and testing of the prototype didn't include of noise data which could have been a good test to determine the performance of the algorithm. Future studies need to explore introduction of noisy data by replacing a certain percentage (in the range 10% - 90%) of the examples by completely random values in the attributes to evaluate the robustness of the classifier.
6. Only KNN classifier was implemented in this study, for studies where the choice still remains KNN Classifications, other variants of KNN such as Two-point-based binary search trees, Furthest-Pair-Based Decision Trees and Norm-Based Binary Search Trees may need to be investigated.

REFERENCES

- Aleskerov, E., Freisleben, B. & B Rao (1997). 'CARDWATCH: A Neural Network-Based Database Mining System for Credit Card Fraud Detection', Proc. of the IEEE/IAFE on Computational Intelligence for Financial Engineering, 220-226.
- A. Aamodt, E. Plaza (1994). Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, AI Communications. IOS Press, Vol. 7: 1, pp. 39-59.
- Akhilomen. John, (2013). "Data Mining Application for Cyber Credit-Card Fraud Detection System", Springer-Verlag Berlin Heidelberg, pp. 218–228
- Bentley, P., Kim, J., Jung. G. & J Choi (2000). Fuzzy Darwinian Detection of Credit Card Fraud, Proc. of 14th Annual Fall Symposium of the Korean Information Processing Society.
- Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. Decision Support Systems, 50 (3), 602-613.
- Bolton, R & Hand, D (2001). Unsupervised Profiling Methods for Fraud Detection. Credit scoring and Credit control VII
- Brause R., Langsdorf T. & M Hepp. (1999). Credit card fraud detection by adaptive neural data mining, Internal Report 7/99 (J. W. Goethe-University, Computer Science Department, Frankfurt, Germany).
- Brause, R., Langsdorf, T. & M Hepp. (1999). Neural Data Mining for Credit Card Fraud Detection, Proc. of 11th IEEE International Conference on Tools with Artificial Intelligence.
- Burge, P & Shawe-Taylor, J (2001). An unsupervised Neural, Network Approach to profiling the behaviour of Mobile Phone, Users for use in Fraud Detection. A journal of parallel and Distributed computing 61: 915-925.
- Chan, P., Fan, W. Prodromidis, A. & S Stolfo (1999). 'Distributed Data Mining in Credit Card Fraud Detection'. IEEE Intelligent Systems, 14; 67-74.

- Chan, P., Stolfo, S., Fan, D., Lee, W. & A Prodromidis (1997). Credit card fraud detection using meta learning: Issues and initial results, Working notes of AAAI Workshop on AI Approaches to Fraud Detection and Risk Management.
- Cover, T. and P. Hart, 1967. Nearest neighbor pattern classification. IEEE Trans. Inform. Theory, 13: 21-27.
- Devroye, L., L. Györfi and G. Lugosi, 1996. A Probabilistic Theory of Pattern Recognition. Springer Science and Business Media, New York, USA., ISBN-13: 9780387946184, Pages: 636.
- D. B. Leake (1996). Case-Based Reasoning: Experiences, Lessons and Future Directions, MIT Press, Cambridge, MA, USA.
- D.J. Hand. Discrimination and classification. Wiley series in probability and mathematical statistics: Applied probability and statistics. J. Wiley, 1981.
- Dorransoro, J. Ginel, F. Sanchez, C. & C Cruz. (1997). 'Neural Fraud Detection in Credit Card Operations'. IEEE Transactions on Neural Networks, 8; 827-834. 17
- Estevez, P., C. Held, and C. Perez (2006). Subscription fraud prevention in telecommunications using fuzzy rules and neural networks. Expert systems with Applications 31,337-344
- Fan, W. (2004) Systematic Data Selection to Mine Concept-Drifting Data Streams, Proc. of SIGKDD04; 128-137.
- Fan, W., Miller, M., Stolfo, S., Lee, W. & P Chan (2001). Using Artificial Anomalies to Detect Unknown and Known Network Intrusions, Proc. of ICDM01; 123-248.
- Gadi, Wang, Lago (2008). Comparison with Parametric Optimization in Credit Card Fraud Detection; IEEE;
- Ghosh, S. & Reilly, D. (1994). 'Credit Card Fraud Detection with a Neural-Network, Proc. of 27th Hawaii International Conference on Systems Science, 3; 621-630.
- Green, B & Choi J (1997). Assessing the Risk of Management Fraud through Neural Network Technology. Auditing 16(1): 14-28.

- G.J. McLachlan (1992). Discriminant analysis and statistical pattern recognition. Wiley Series in Probability and Statistics. Wiley.
- Leo Breiman (1984). Classification and Regression Trees. Wadsworth Publishing Co Inc.
- Murad, U & Pinkas G. (1999). Unsupervised Profiling for identifying superimposed Fraud. Proceedings of PKDD'99
- Richard J. Bolton, David J. Hand, and David J. H (2002). Statistical fraud detection: A review. *Statistical Science*, 17(3):235–249.
- Richard J Bolton and David J Hand (2001). Unsupervised profiling methods for fraud detection. *Credit Scoring and Credit Control VII*, pages 235–255.
- J.R. Quinlan (1993). C4.5: Programs for Machine Learning. C4.5 - programs for machine learning / J. Ross Quinlan. Morgan Kaufmann Publishers.
- S. Viaene, R.A. Derrig, and G. Dedene (2004). A case study of applying boosting naive bayes to claim fraud diagnosis. *Knowledge and Data Engineering, IEEE Transactions on*, 16(5):612–620.
- Kokkinaki, A. (1997). On Atypical Database Transactions: Identification of Probable Frauds using Machine Learning for User Profiling, Proc. of IEEE Knowledge and Data Engineering Exchange Workshop; 107-113.
- Kim, M. & Kim, T (2002). A Neural Classifier with Fraud Density Map for Effective Credit Card Fraud Detection, Proc. Of IDEAL 2002, 378-383
- Maes, S., Tuyls, K., Vanschoenwinkel, B. & B Manderick (2002). Credit Card Fraud Detection using Bayesian and Neural Networks, Proc. of the 1st International NAISO Congress on Neuro Fuzzy Technologies.
- Quah T. S, & Sriganesh M. (2008). 'Real-time credit card fraud using computational intelligence'. *Expert Systems with Application*, 35:4, 1721-1732.
- Wheeler, R. & Aitken, S. (2000). 'Multiple Algorithms for Fraud Detection'. *Knowledge-Based Systems*, 13; 93-99.

- Zaslavsky V. & Strizhak A. (2006). 'Credit card fraud detection using self-organizing maps'. *Information and Security*, 18; 48-63.
- Tue, Ren, Liu (2004); *Artificial Immune System for Fraud Detection*; IEEE, vol. 2, pp. 1407-1411,.
- Dasgupta, D., Ji, Z., & González, F. A. (2003). Artificial immune system (AIS) research in the last five years. In *IEEE Congress on Evolutionary Computation (1)* (pp. 123-130).
- Kolodner, J. (1993). *Case-based reasoning*. Morgan Kaufmann.
- Pozzolo. A, Boracchi. G, Caelen. O, Alippi. C, Bontempi. G, "Credit Card Fraud Detection and Concept-Drift Adaptation with Delayed Supervised Information"
- M. Sasirekha, Thaseen. Sumaiya, Banu. Saira, (2012). "A DEFENSE MECHANISM FOR CREDIT CARD FRAUD DETECTION", *International Journal on Cryptography and Information Security (IJCIS)*, pp. 89-100.
- Otair, Mohammed. (2012). *Mobile Banking Based On Stand-Alone Mobile Application Clients – A Suggested Mobile Banking Solution For Banks In Jordan-*. *International Journal of Advanced Research in Computer Science*. 3. 49-58.
- R. C. Schank, R. P. Abelson (1977). *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures*, L. Erlbaum, Hillsdale, NJ.
- Khuman, M.B., 2012. Classification of remote sensing data using KNN method. *J. Inform. Knowledge Res. Electron. Commun. Eng.*, 2: 817-821.
- Kulkarni, S.R., G. Lugosi and S.S. Venkatesh, 1998. Learning pattern classification-a survey. *IEEE Trans. Inform. Thoery*, 6: 2178-2206.
- Ming Zhao and Jingchao Chen, 2016. Improvement and Comparison of Weighted k Nearest Neighbors Classifiers for Model Selection. *Journal of Software Engineering*, 10: 109-118.
- Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., et al. (2008). Top 10 algorithms in data mining. *Knowledge and information systems*, 14 (1), 1 {37.

APPENDICES:

APPENDIX 1: RESEARCH PLAN AND TIMETABLE

ACTIVITY	NOV 2019	DEC 2019	JAN 2020	FEB 2020	MARCH 2020	APRIL 2020
✓ Problem Identification & Development of Concept Paper						
✓ Proposal writing & review ✓ Proposal M1 Presentation (Defense)						
✓ Requirements Gathering & Analysis						
✓ Prototype Design & Development						
✓ Project Progress Review & Presentation						
✓ Implementation & Evaluation						
✓ Finalization of Project Document						
✓ Final Presentation & Submission of research project report						

APPENDIX 2: LIST OF REQUIRED RESOURCES

1. MongoDB – NoSQL Database or MySQL Database
2. NetBeans 8.0 IDE for Java Development
3. Oracle JDeveloper 12c
4. 16GB RAM & 300GB HDD Laptop
5. Oracle Web Logic Server 12c

APPENDIX 3: BUDGET ESTIMATES

ITEM	QUANTITY	UNIT COST (KES)	TOTAL COST (KES)
Stationery	5 reams	700	3,500
Travelling			2,000
Internet Services	50GB Bundles		2,000
Phone calls Airtime (Safaricom)	2 Months Post Paid		2,000
Typing services	80 pages (5 copies)	10	4,000
Binding Services	80 pages (5 copies)	150	750
Photocopying	80 pages (5 copies)	3	1,200
Meals			2,000
Requirements Gathering & Analysis			30,000
Prototyping & Evaluation			40,000
Data Analysis & Interpretation			30,000
Total			117,450

APPENDIX 4: QUESTIONNAIRE SURVEY

TOPIC: REAL TIME FRAUD DETECTION FOR MOBILE BANKING: BASED ON EXPERIENTIAL PARADIGM

The questionnaire forms part of Master of Science in Computer Science study conducted under supervision of the School Of Computing & informatics of the University Of Nairobi (UON) and will be used for academic purposes ONLY. Your responses will be kept confidential and used as data for model assessment. The name of your organization will not be used. Your responses will not be published in any way that the organization or you can be identified.

The purpose of the study is to determine the research and develop a prototype for real time fraud detection for mobile banking using case-based reasoning.

SECTION A: DEMOGRAPHIC INFORMATION

1. Please select the name of your gender
 - Male
 - Female

2. How long have you worked in KCB?
 - 0 - 1 Year

 - 1 - 5 Year

 - More than 5 Years

3. What's your highest level of education?
 - Undergraduate

 - Masters Degree

 - PhD Degree

4. What's your current position/role in KCB?

- Information Risk Analyst
- Cyber Security Analyst
- Digital Forensics Analyst
- Head Information Risk
- Head IT Security
- Head Forensics

5. How long have you worked in KCB at the above role/position?

- 0 - 1 Year
- 1 - 5 Year
- More than 5 Years

SECTION B: TRANSACTION ATTRIBUTES CALIBRATION

i) Using a scale of 1-5, where 1= Strongly Disagree (SD); 2= Disagree (D); 3=Neutral extent (N); 4=Agree (A); 5= Strongly Agree (SA), please indicate the extent to which you agree that the following features are key in determining whether a transaction can be considered fraudulent or not

	1	2	3	4	5	Comment/Justification
1. Transaction amount						
2. Customer's Age						
3. Customers Gender						

4. Mobile Banking Registration Date						
5. Mobile Banking PIN Reset Date						
6. SIM Swap Date						
7. Account Opening Date						
8. Transaction Type						

ii) Kindly indicate any other customer or transaction attribute that you think can determine whether a transaction can be considered fraudulent or not

.....

.....

.....

SECTION C: TRANSACTION ATTRIBUTES WEIGHTS

i) Using a scale of 1-5, where 1 indicates the smallest weights and 5 the greatest weight, please indicate the relative weighting of for the above transaction attributes towards contributing to a transaction being considered as fraudulent.

	1	2	3	4	5	Comment/Justification
1. Transaction amount						
2. Customer's Age						
3. Customers Gender						
4. Mobile Banking Registration Date						
5. Mobile Banking PIN Reset Date						
6. SIM Swap Date						

7. Account Opening Date						
8. Transaction Type						

ii) Kindly indicate any other customer or transaction attribute that might have been left out and its relative weighting

.....

.....

.....

APPENDIX 5: SAMPLE SOURCE CODES

RealTimeFraudAnalyzer.java

===

```
package com;

/**
 *
 * @author McOmollo
 */

import com.processor.MobileFraudAnalyzerConsumer;

import com.processor.MobileFraudAnalyzerProducer;

import java.util.concurrent.*;

import org.apache.log4j.Logger;

import org.apache.log4j.PropertyConfigurator;

public class RealTimeFraudAnalyzer {

    static Logger logger = Logger.getLogger(RealTimeFraudAnalyzer.class);

    /**
     * @param args the command line arguments
     */

    public static void main(String[] args) {
```

```

PropertyConfigurator.configure("C:\\Users\\McOmollo\\Documents\\NetBeansProjects\\RealTimeFraudAnalyzer\\src\\log4j.properties");

    logger.info(" -----Initialising Real Time Fraud Analyzer Engine-----");

    LinkedBlockingQueue sharedQ = new LinkedBlockingQueue(1000);
    LinkedBlockingQueue sharedMQ = new LinkedBlockingQueue(1000);

    ExecutorService eMobileFraudAnalyzerProducer = Executors.newSingleThreadExecutor();
    ExecutorService eMobileFraudAnalyzerConsumer = Executors.newFixedThreadPool(100);
    eMobileFraudAnalyzerProducer.execute(new MobileFraudAnalyzerProducer(sharedMQ));
    eMobileFraudAnalyzerConsumer.execute(new
MobileFraudAnalyzerConsumer(sharedMQ));

    logger.info("-----Real Time Fraud Analyzer Started -----");

    }
}
===

```

MobileFraudAnalyzerProducer.java

```
package com.processor;

import com.base.EngineBean;

import com.entities.MobileTransactions;

import java.util.List;

import java.util.Properties;

import java.util.concurrent.LinkedBlockingQueue;

import org.apache.log4j.Logger;

/**
 *
 * @author McOmollo
 */
public class MobileFraudAnalyzerProducer implements Runnable {

    LinkedBlockingQueue sharedQ;

    Properties prop;

    String hostIpAdress;

    int hostPortNo;

    static Logger logger = Logger.getLogger(MobileFraudAnalyzerProducer.class);

    public MobileFraudAnalyzerProducer(LinkedBlockingQueue sharedQ) {
```

```

    this.sharedQ = sharedQ;

    // this.prop = prop;
}

public String initialiseProcessTransactions() {

    EngineBean eb = new EngineBean();

    List<MobileTransactions> transList = eb.getUnProcessedMobileTrans("0");

    if (!transList.isEmpty()) {

        for (MobileTransactions tx : transList) {

            if (!this.sharedQ.contains(tx)) {

                try {

                    sharedQ.put(tx);

                    tx.setTransProcessStatus("1");

                    eb.update(tx);

                    logger.info(" Queuing Transaction..." + tx.getTransTransRef());

                } catch (InterruptedException ex) {

                    ex.printStackTrace();

                }

            }

        }

    }

}

return null;

}

```

```
@Override  
  
public void run() {  
    while (true) {  
        initialiseProcessTransactions();  
    }  
}  
}
```

==

MobileFraudAnalyzerConsumer.java

===

```
package com.processor;  
  
import com.base.EngineBean;  
import com.base.GlobalCC;  
import com.entities.AttributeWeightingMatrix;  
import com.entities.Customers;  
import com.entities.FraudScoreMatrix;  
import com.entities.MobileTransactions;  
import java.math.BigDecimal;  
import java.math.BigInteger;  
import java.text.SimpleDateFormat;  
import java.util.Date;
```

```

import java.util.List;

import java.util.Properties;

import java.util.concurrent.LinkedBlockingQueue;

import org.apache.log4j.Logger;

/**
 *
 * @author McOmollo
 */

public class MobileFraudAnalyzerConsumer implements Runnable {

    LinkedBlockingQueue sharedQ;

    Properties prop;

    String hostIpAdress;

    int hostPortNo;

    static Logger logger = Logger.getLogger(MobileFraudAnalyzerConsumer.class);

    public MobileFraudAnalyzerConsumer(LinkedBlockingQueue sharedQ) {

        this.sharedQ = sharedQ;

        // this.prop = prop;

    }

    public void processMobileTrans() {

        MobileTransactions mttx = null;

```

```

if (!sharedQ.isEmpty()) {
    try {
        mtx = (MobileTransactions) sharedQ.take();
        consumeMobileTrans(mtx);
    } catch (InterruptedException ex) {
        ex.printStackTrace();
        logger.error(ex.getMessage());
    } catch (Exception ex) {
        ex.printStackTrace();
        logger.error(ex.getMessage());
    }
}

}

}

public void consumeMobileTrans(MobileTransactions mtx) {

    try {
        BigInteger transAmt = mtx.getTransAmount();
        BigInteger transCustRef = mtx.getTransCustRef();
        String transCustMobileNo = mtx.getTransMobileNo();
        Date transDate = mtx.getTransTransDate();
        String transRef = mtx.getTransTransRef();
        logger.info(" Processing transaction..." + transRef);
    }
}

```

```
//String transIMSI = caseBase.getTransImsiId();

//Long transCode = caseBase.getTransCode();

EngineBean eb = new EngineBean();

Customers cust = eb.getCustomerDetails(transCustMobileNo);

if (cust != null) {

    String custName = cust.getCustName();

    String custGender = cust.getCustGender();

    String custIMSI = cust.getCustImsiId();

    Date mbankRegDate = cust.getCustBankRegDate();

    Date custPinChangeDate = cust.getCustLastPinChangeDate();

    Date custSIMSwapDate = cust.getCustLastSswapDate();

    Date custDOB = cust.getCustDob();

    Date custAccOpenDate = cust.getCustAcctOpeningDate();

    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

    //Date today = new Date();

    String transDateVal = sdf.format(transDate);

    String custDOBDate = sdf.format(custDOB);

    String mbankRegDateStr = sdf.format(mbankRegDate);

    String custPinChangeDateStr = sdf.format(custPinChangeDate);
```



```
String custSIMSwapDateStr = sdf.format(custSIMSwapDate);

String custAccOpenDateStr = sdf.format(custAccOpenDate);

GlobalCC gcc = new GlobalCC();

String custAgeYears = gcc.calculateDateDiff(custDOBDate, transDateVal, "YEARS");

String    mbankingRegAgeDays    =    gcc.calculateDateDiff(mbankRegDateStr,
transDateVal, "DAYS");

String    custPinChangeAgeDays    =    gcc.calculateDateDiff(custPinChangeDateStr,
transDateVal, "DAYS");

String    custSIMSwapAgeDays    =    gcc.calculateDateDiff(custSIMSwapDateStr,
transDateVal, "DAYS");

String    custAccOpenAgeDays    =    gcc.calculateDateDiff(custAccOpenDateStr,
transDateVal, "DAYS");
```

```
List<AttributeWeightingMatrix> matixList = eb.getWeigtingMatrix();
```

```
int custAgeWeight = 0;

int transGenderWeight = 0;

int transAmtWeight = 0;

int transMbankingRegAgeWeight = 0;

int transCustAccOpenAgeWeight = 0;

int transCustPinChangeAgeWeight = 0;

int transCustSIMSwapAgeWeight = 0;

String fraudLabel = "NEGATIVE";
```

```

BigInteger transFraudVerdit = new BigInteger("0");

BigInteger fraudScore = new BigInteger("0");

for (AttributeWeightingMatrix mx : matixList) {

    String wmAttribute = mx.getWmAttribute();

    String wmMinVal = mx.getWmMinValue();

    String wmMaxVal = mx.getWmMaxValue();

    String wmStatus = mx.getWmStatus();

    String wmWeight = mx.getWmThresholdWeight();

    String custAgeYrs = custAgeYears + "";

    mtx.setTransCustAge(new BigInteger(custAgeYears + ""));

    if (wmAttribute.equalsIgnoreCase("AGE")) {

        if (new Integer(custAgeYrs) >= new Integer(wmMinVal) && new
Integer(custAgeYrs) <= new Integer(wmMaxVal)) {

            custAgeWeight = new Integer(wmWeight);

            mtx.setTransCustAgeWeight(new BigInteger(custAgeWeight + ""));

        }

    }

    mtx.setTransGender(cust.getCustGender());

    if (wmAttribute.equalsIgnoreCase("GENDER")) {

        if (custGender.equalsIgnoreCase(wmMinVal)) {

            transGenderWeight = new Integer(wmWeight);

            mtx.setTransGenderWeight(new BigInteger(transGenderWeight + ""));

        }

    }

}

```

```

    }
}

if (wmAttribute.equalsIgnoreCase("TRANS_AMOUNT")) {
    int rs = transAmt.compareTo(new BigInteger(wmMinVal));
    int rs2 = transAmt.compareTo(new BigInteger(wmMaxVal));
    if (rs == 1 && rs2 == -1) {
        transAmtWeight = new Integer(wmWeight);
        mtx.setTransAmountWeight(new BigInteger(transAmtWeight + ""));
    }
}

mtx.setTransMbankRegAge(new BigInteger(mbankingRegAgeDays + ""));
if (wmAttribute.equalsIgnoreCase("MBANKING_REG_AGE_DAYS")) {
    if (new Integer(mbankingRegAgeDays) >= new Integer(wmMinVal) && new
Integer(mbankingRegAgeDays) <= new Integer(wmMaxVal)) {
        transMbankingRegAgeWeight = new Integer(wmWeight);
        mtx.setTransMbankRegDaysWeight(new
BigInteger(transMbankingRegAgeWeight + ""));
    }
}

mtx.setTransAccOpeningAge(new BigInteger(custAccOpenAgeDays));

```

```

    if (wmAttribute.equalsIgnoreCase("ACCT_OPENING_AGE_DAYS")) {
        if (new Integer(custAccOpenAgeDays) >= new Integer(wmMinVal) && new
Integer(custAccOpenAgeDays) <= new Integer(wmMaxVal)) {
            transCustAccOpenAgeWeight = new Integer(wmWeight);
            mtx.setTransAcctOpeningAgeWeight(new
BigInteger(transCustAccOpenAgeWeight + ""));
        }
    }

    mtx.setTransPinChangeAge(new BigInteger(custPinChangeAgeDays + ""));
    if (wmAttribute.equalsIgnoreCase("PIN_CHANGE_AGE_DAYS")) {
        if (new Integer(custPinChangeAgeDays) >= new Integer(wmMinVal) && new
Integer(custPinChangeAgeDays) <= new Integer(wmMaxVal)) {
            transCustPinChangeAgeWeight = new Integer(wmWeight);
            mtx.setTransPinChangeAgeWeight(new
BigInteger(transCustPinChangeAgeWeight + ""));
        }
    }

    mtx.setTransSimSwapAge(new BigInteger(custSIMSwapAgeDays + ""));
    if (wmAttribute.equalsIgnoreCase("SIM_SWAP_AGE_DAYS")) {
        if (new Integer(custSIMSwapAgeDays) >= new Integer(wmMinVal) && new
Integer(custSIMSwapAgeDays) <= new Integer(wmMaxVal)) {
            transCustSIMSwapAgeWeight = new Integer(wmWeight);

```

```

        mtx.setTransSimSwapAgeWeight(new
BigInteger(transCustSIMSwapAgeWeight + ""));

    }

}

}

mtx.setTransProcessStatus("2");

BigDecimal AggregatefraudScore = new BigDecimal(transGenderWeight).add(new
BigDecimal(transAmtWeight));

AggregatefraudScore = AggregatefraudScore.add(new
BigDecimal(transMbankingRegAgeWeight)).add(new
BigDecimal(transCustAccOpenAgeWeight));

AggregatefraudScore = AggregatefraudScore.add(new
BigDecimal(transCustPinChangeAgeWeight)).add(new
BigDecimal(transCustSIMSwapAgeWeight));

String transStatus = "APPROVED";

fraudScore = new BigInteger(AggregatefraudScore + "");

mtx.setTransFraudScore(fraudScore);

FraudScoreMatrix fsm = eb.getLabeledFraudScore(fraudScore);

if (fsm != null) {

    fraudLabel = fsm.getFsmLabel();

} else {

    System.out.println(" Fraud Score " + fraudScore + " for trans Ref=>" + transRef + "
Mobile No => " + transCustMobileNo + " is out of defined fraud score matrix...");

    logger.info(" Fraud Score " + fraudScore + " for trans Ref=>" + transRef + " Mobile
No => " + transCustMobileNo + " is out of defined fraud score matrix...");

```

```

    }

    if (fraudLabel.equalsIgnoreCase("POSITIVE")) {

        transFraudVerdit = new BigInteger("1");

        transStatus = "BLOCKED";

    }

    mtx.setTransStatus(transStatus);

    mtx.setTransFraudVerdict(transFraudVerdit);

    logger.info(" Transaction Attributes Weights: [ GENDER: " + transGenderWeight + "
]; [ AMOUNT: " + transAmtWeight + " ]");

    logger.info(" Transaction Attributes Weights: [ BANK REG AGE: " +
transMbankingRegAgeWeight + " ]; [ ACCT OPEN AGE: " + transCustAccOpenAgeWeight + "
]");

    logger.info(" Transaction Attributes Weights: [ PIN CHANGE AGE: " +
transCustPinChangeAgeWeight + " ]; [ SIM SWAP AGE: " + transCustSIMSwapAgeWeight + "
]");

    logger.info(" Fraud score for transaction " + transRef + " Is => " +
AggregatefraudScore);

    } else {

        System.out.println(" Transaction missing customer record...transRef=>" + transRef + "
Mobile No => " + transCustMobileNo);

        logger.info(" Transaction missing customer record...transRef=>" + transRef + " Mobile
No => " + transCustMobileNo);

        mtx.setTransProcessStatus("3");

        Customers cx = eb.getCustomerByCustCode(transCustRef);

```

```
        if (cx != null) {  
            mtx.setTransMobileNo(cx.getCustMobileNo());  
            mtx.setTransProcessStatus("0");  
        }  
    }  
    eb.update(mtx);  
} catch (Exception ex) {  
    ex.printStackTrace();  
}  
}  
  
@Override  
public void run() {  
    while (true) {  
        processMobileTrans();  
    }  
}  
}
```