MODELLING A MULTI-CHANNEL MESSAGING FRAMEWORK: A MACHINE LEARNING APPROACH

by

OLUSOLA OLUSEUN SALAMI

submitted in accordance with the requirements for
the degree of

DOCTOR OF PHILOSOPHY

in the subject

COMPUTER SCIENCE

at the

UNIVERSITY OF SOUTH AFRICA

SUPERVISOR: Prof E Mnkandla

09th January 2023

DECLARATION

Name: Olusola Oluseun Salami

Student Number: 49920847

Degree: PhD Computer Science

Exact wording of the title of the thesis as appearing on the electronic copy submitted for examination:

Modelling A Multi-Channel Messaging Framework: A Machine Learning Approach

I declare that the above thesis is my own work and that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references.

I further declare that I submitted the thesis to originality checking software and that it falls within the accepted requirements for originality.

I further declare that I have not previously submitted this work, or part of it, for examination at Unisa for another qualification or at any other higher education institution.

(The thesis will not be examined unless this statement has been submitted.)

09-Jan-2023

SIGNATURE DATE

ACKNOWLEDGEMENTS

Special thanks to Almighty God for the strength, knowledge and understanding that he bestowed on me his creation during this study. In addition, many thanks and gratitude to my amiable supervisor Prof. Ernest Mnkandla for his advice, support and guidance on this journey, in this emerging area of machine learning and artificial intelligence. My appreciation also goes to my wife Oyenike and children (Olusolafunmi Francis, Oluwadara Anne-Marie and Oluwaseyi Zoe) for their support and time taken in family way to progress this study, I will make this up with all of you. I also thank my Parents, Surv. & Dr. (Mrs.) Y.A Salami (FNIS) for their indelible believe in education for all your children (Dr. (Mrs.) Abiola Olubiyi, Olugbenga, Abiodun Salami and Oluwafolakemi Adebayo) and most especially the sacrifice for the family. To Dr. Olalekan Samuel Ogunleye for your support and encouragement all through my study. To Dr. Debo Owoseni and Oluwaseun Mary (BHM), for your encouragement assistance in reviewing my papers including insights and inputs on the flow which were adopted. Sincere thanks to Dr. Funmilade Faniyi for his valuable external perspective on this research area as it applies to behavioural learning and adaptability. Without all of you and your valuable contributions this study might not have been possible.

ABSTRACT

Multi-channel messaging (MCM) systems have been implemented to integrate heterogeneous messaging channels for message delivery to Financial Services Institutions (FSIs) customers worldwide. However, implementations utilizing machine learning techniques to determine channel availability, dynamic assignment, and monitoring of customer patterns are being explored with newer technological advancements. Such approaches are used to investigate how Integrated multi-channel messaging can be implemented using machine learning algorithms to enable effective and efficient dynamic channel selection and integration methods.

This research explored and investigated the various machine learning algorithms for optimal channel selection. The study delved into applying these algorithms and their use to channel selection, including the end-user context, focusing on the multi-armed bandit (MAB) problem, Tug of War and Upper Confidence Bound algorithm in providing a novel approach to solving this problem.

The study presented a framework that fully integrates different web channels (Facebook, WhatsApp, Instagram IM, SMS, and Email) with a decision-making module and machine learning for the model to learn customer patterns over time. This framework uses a minimal memory and computation capability which employs simple learning procedures from the machine learning algorithms while relying on the message acknowledgement feedback from each channel. This work also describes a software architecture to support this and evaluate its effectiveness in an MCM enabled customer alert system currently used by financial service institutions. The efficacy of the proposed solution was evaluated using a simulation-based performance analysis method.

The designed framework took advantage of simple machine learning algorithms and the inherent flexibility of integrating heterogeneous channels for efficient resource utilization when messages are transmitted. Furthermore, an Enterprise Service Bus (ESB) assigns weights dynamically to channels that customer use frequently with the use of the Upper Confidence Bound algorithm. This method allows flexibility to realise requirements for geolocation sensing, scheduling schemes and user mobility. The framework can choose channels dynamically, incorporating a machine learning module for message delivery patterns and supporting an integration layer via a common data channel-agnostic to each channel integrated. Lessons learnt from this design should be further refined to motivate future work in this area

KEY TERMS:

Machine Learning; Artificial Intelligence; Datasets; Multi-Channel Messaging System; Messaging Channels; Machine Learning Algorithms; Upper Confidence Bound Algorithm; Tug Of War Algorithm; Multi-Armed Bandits Problem; Design Science Research; Enterprise Service Bus; Message Producers; Message Consumers; Financial Services Institutions; Customer Alert Messaging System

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF LISTINGS

LIST OF APPENDIXES

LIST OF PUBLICATIONS FROM THIS THESIS

1. Salami, O, & Mnkandla, E (2020). A Machine-Learning-Based Channel Selection Model for A Multi-Channel Messaging Customer Alert System. Conference of the South African Institute of Computer Scientists and Information Technologists, Cape Town, 2.

2. Salami, O, & Mnkandla, E (2021). Towards A Machine Learning Enabled Multi-Channel Messaging Framework for Financial Service Institutions: Preliminary Investigations. International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems, Durban, KwaZulu Natal, 8.

3. Salami, O, & Mnkandla, E (2022). A Design For A Machine-Learning-Enabled Multi-Channel Messaging Framework for Financial Service Institutions. International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems, Durban, KwaZulu Natal, 8.

LIST OF ABBREVIATIONS

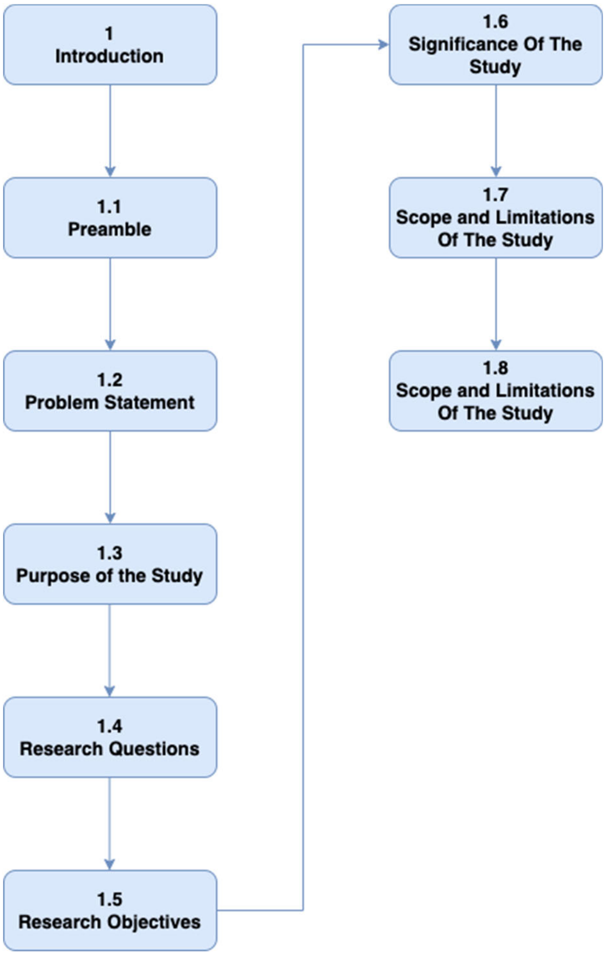
Acronym or Abbreviation	Description
3GPP	3rd Generation Partnership Project
ANT	Actor-Network Theory
API	Application Programming Interface
AI	Artificial Intelligence
BCNF	Boyce-Codd Normal Form
BVN	Bank Verification Number
CR	Carriage Return
DB	Digital Banking
DNN	Deep Neural Nets
DSR	Design Science Research
EAI	Enterprise Application Integration
EDI	Electronic Data Interchange
EMAIL	Electronic Mail
E2E	End To End Encryption
ESB	Enterprise Service Bus
ERM	Entity Relationship Model
FSIs	Financial Services Institutions
GDPR	General Data Protection Regulation
GPU	Graphics Processing Units
ICT	Information and Communication Technology
IoT	Internet of Things
IS	Information Systems
HEED	Hybrid, Energy-Efficient and Distributed Clustering approach

Acronym or Abbreviation	**Description**
HTTP	Hypertext Transfer Protocol
IM	Instant Messaging
IMF	Internet Message Format
IoT	Internet of Things
IMBC	IoT Message-Based Communication
IS	Information Systems
IT	Information Technology
JSON	JavaScript Object Notation
LF	Line Feed
ILL	Iterated Logarithm Law
LwM2M	Lightweight Machine to Machine.
MAB	Multi-Armed Bandit
MCM	Multi-Channel Messaging
MDP	Markov Decision Process
ML	Machine Learning
MIME	Multi-Purpose Internet Mail Extension
MMS	Multimedia Messaging Service
MMAP	Mobile Message Access Protocol
OTP	One-Time Password
REST or RESTFUL	Representational State Transfer
SCM	Single Channel Messaging
SIP	Session Initiation Protocol

Acronym or Abbreviation	Description
SMS	Short Messaging Service
SMTP	Simple Mail Transport Protocol
SMPP	Short Message Peer to Peer Protocol
SMSC	Short Message Service Centre
SOAP	Simple Object Access Protocol
TAM	Technology Acceptance Model
TNB	The Nigeria Bank
TOW	Tug of War
TP-User Data	Transfer Layer Protocol User Data
TRA	Theory of Reasoned Action
UCBA	Upper Confidence Bound Algorithm
TUS-ASCII	The United States American Standard Code for Information Interchange
USSD	Unstructured Supplementary Service Data
URI	Uniform Resource Identifier
PIN	Personal Identification Number
XLST	Extensible Stylesheet Language Transformations
XFD	XML Form Definition
X2JS	XML to JavaScript
XML	Extensible Markup Language
WSDL	Web Services Description Language

Acronym or Abbreviation	Description
4IR	Fourth Industrial Revolution

1 CHAPTER 1 INTRODUCTION



1.1 PREAMBLE

Messaging systems provide a platform for information sharing between organizations and their customers. Technological advancements such as machine learning (ML), the Internet of Things (IoT), the Fourth Industrial Revolution (4IR), and others provide financial service institutions (FSIs) with opportunities to connect with their customers in more intelligent ways. The Multi-Channel Messaging (MCM) system allows the seamless integration of different channels of communications within a single platform for different channels (Salami & Mtsweni, 2019). The various channels of messaging communication offered by Mobile Service Providers include email service, Short Messaging Service (SMS), Social media platforms (e.g., Telegram, Twitter Direct Message, Facebook Messenger and WhatsApp Instant Messenger). These channels allow FSIs to send and receive relevant information to or from their customers respectively. The ability of FSIs to implement the sending of messaging information timeously and via the channel available can determine their success or failure.

Liang et al. (2011) implemented an MCM system tagged Integrated Multi-Channel Messaging Model (IM^3) system with email and SMS integration. IM^3 utilized a module for channel assignment and decision making using message priority for channel selection. Khan and Siddique (2004) implemented an MCM solution using web service open standards in a retailer and credit card context. The core of their system implemented a static channel selection module, a service gateway, and a service integration bus with no preference for customer preference or message priority. The significant limitations of these systems are the use of customers' profiled choice to select the channel for sending the message and the lack of a self-learning mechanism to determine the channels the customer is most receptive to.

Dynamic channel selection has motivated the author to propose a ML-enabled channel assignment algorithm for making decisions regarding the most optimal channel for a Multi-Channel Customer Alert System. The decision-making module integrates heterogeneous web channels using an Enterprise Service Bus (ESB) layer with its Application Programming Interface (API). The framework utilizes ML algorithms to determine channel availability, dynamic assignment and monitor customer patterns. The advantage or s over the existing MCM systems is the ability to choose channels dynamically. This framework also learns the models for message delivery and support an integration layer via a common data channel-agnostic to each channel integrated. This dynamic learning framework ensures that multiple homogenous channels can share the same message structure with the use of an agnostic data sharing format.

The customers of FSIs are becoming technologically savvy and rely on channels that provide instant messaging. In response, FSIs are required to offer a bouquet of messaging channels to enable competitive options to satisfy their customers' expectations by presenting channel options that can be selected based on each consumer's preference. Due to customers' increased use of cyberspace and the Internet, a corresponding surge has been observed in cybercrime targeting FSIs and banking solutions (Schaffer et al., 2018). Customers need to be informed about operations and transactions on their accounts at all times, wherever they are, and in a timeous manner. Schaffer et al. (2018) further opined that cybersecurity incidents have increased over the previous year's owing to ease of access to technological platforms.

Such incidents can result in customers losing confidence in organizations that provide financial services because they fear financial loss resulting from security breaches or unknown threats to the infrastructure of these organizations. To further explain the severity of these security breaches, Muncaster (2015, p. 1) asserts that *"In fact, financial services firms are reportedly hit by security incidents a staggering 300 times more frequently than businesses in other industries."*

Customers typically subscribe to different alerts on their accounts to notify them of any real-time transactions taking place in their accounts (Caflisch et al., 2020). These notifications may include requests to access the account via an account password change, online banking, remote or out-of-profile login to mobile or online banking, promotional alerts, product information, and communicating new features or marketing material to customers. Customers base their willingness to be contacted by FSIs on the priority of delivery or the kind of message when interacting with the FSI (Chiu et al., 2004). Thus, in the case of an FSI customer who connects to the banking platform from a location not recognized by the system, the system can confirm the customer's identification using artificial intelligence (AI) via the communication channel specified for delivering information of this nature.

Omarini (2013) defined a channel as the medium through which FSIs can reach their customers and vice versa (inbound *or* outbound). This process includes the use of web-based technologies such as the Internet, email, Facebook, WhatsApp, Instagram messenger and non-web-based technologies such as call centres, SMS and digital TV (Alhassan, 2020). Individual channels are better suited to meeting specific and contextual user requirements. FSIs are quite innovative and randomly adopting innovative technologies due to the nature of their operations and services.

Selecting the most efficient or optimal channel is difficult. Ma et al. (2019) noted several studies that proposed various methods to determine the best channel available to transmit information. Zhu et al. (2016) proposed a design for the channel selection problem using a Multi-Armed Bandit (MAB) problem. Although algorithms such as Ɛ-greedy and Upper Confidence Bounds (UCB) were used to implement channels dynamically, drawbacks with overheads in channel selection were also reported. Ma et al. (2019) implemented an ML assigment in a multi-channel system using the Tug-of-War (TOW) model. Ma et al. (2019) have proposed the use of the TOW dynamic based channel selection or assignment algorithm. The TOW model implements a simple learning procedure with homogenous channels that utilize an acknowledged frame for learning procedure using minimal memory and computation capability that is suitable for MCM environments where channel variables change randomly. The designed framework expands and maximizes this approach with an ESB layer to manage the heterogeneous integration of channels to achieve a quick balance between the exploration-exploitation dilemma for channel selection.

According to Cioffi et al. (2020), recent advances in AI and its branch, ML, have created various opportunities and consequently raised users' expectations. AI's application extends to multiple technological areas such as medicine, gaming platforms, banking and other business (Cioffi et al., 2020). The new normal involves the use of ML and AI techniques in financial business applications using chatbots (Media Labs, 2017). Khavya (2018) described banking chatbots as AI-enabled software developed to solve daily banking operations tasks. These chatbots can understand customers' queries and respond or resolve them in real-time mode.

ML offers automation (Hendrik, 2020). AI-ML systems can provide virtual assistants with greater precision than humans and the capability to improve the translation and interpretation abilities of the human language. AI chatbots have also changed the landscape of the information technology (IT) ecosystem. These chatbots can solve business tasks, determine the best flights by cost and route, personal fitness diet and specialized training, automated hotel booking and banking tasks (Shabbir & Anwer 2018).

This study looks at the problems and challenges encountered in a MCM system currently used by FSIs. While proposing a framework for a multi-channel system that implement the selection and learning module of an ML channel, the study focuses on the dynamic channel selection approach and integration methods to make the system effective and efficient for use with an ESB. Section 1.2 discusses the problem statement in a MCM system. Subsequent sections (i.e., sections 1.3-1.7) focus on the purpose of the study, research questions, research objectives, significance of the study, and scope and limitations of the study.

1.2 PROBLEM STATEMENT

In the present era, the importance of using customer transactional alert systems by FSIs cannot be overstated. In the early 1980s, businesses contacted customers via telephone (landlines) and post office mails only, according to Universal Postal Union (2010). Subsequently, the use of SMS in banking evolved with emails, and this resulted in quicker and more efficient communication channels (Adagunodo & Bamidele, 2007). Web 2.0 technologies, such as social media messaging channels, is commonly used by customers and it offers better integration options between third-party systems (Ayodele & Babajide, 2015).

This integration layer allows the configuration of message types, levels of importance and priority across all channels (MBAMA, 2018).

However, current implementation of MCM systems that incorporate different messaging channels embedded for information delivery to customers is limited to using customer's pre-profiled data to select or learn a channel for message delivery. This method is entirely manual, irrespective of the message type, importance or priority level and, most importantly, the customer's most receptive channel for message delivery. Hence, a need exist to research and propose a framework for a ML and channel selection module embedded within the system to allow dynamic channel selection and ML using exploitation and exploratory approach for channel selection. Channel selection remains an integral part of an MCM system. Hence, it is imperative to implement a dynamic and self-aware system with ML capabilities to instantly determine available channels for customers (Kim et al., 2010; Ma et al., 2019).

1.3 THE PURPOSE OF THE STUDY

The research study designed an ML framework for enhancing MCM platform for FSIs for the seamless integration of heterogeneous channels of communication (e.g., SMS, VoIP, IM, USSD, App-based, and others) to the benefit of their customers. ML algorithms can be used to learn the availability of a channel, reliability of the channel and the most optimal channel to be selected at an instance of time for message delivery. The use of a self-aware MCM system as shown in Figure 1.1 would position FSIs for future advantages. The issues to be addressed include, but not limited to, message type support, message prioritization and message delivery feedback and channel selection methods. An ESB in the framework manages viable and alternative channels for message delivery to customers.

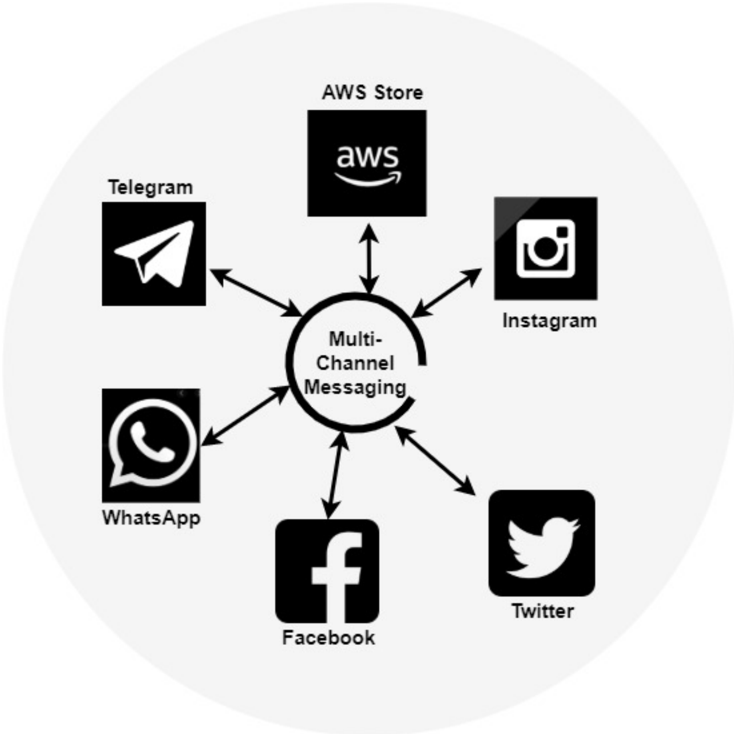


Figure 1.1: Multi-Channel Messaging System Architecture. Source **(Aaron, 2018)**

1.4 RESEARCH QUESTIONS

In alignment with the research problem, this exploratory study seeks to answer the following research question:

How can integrated multi-channel messaging be implemented using machine learning algorithms to enable effective and efficient dynamic channel selection and integration methods?

Furthermore, the main research question is broken down into the following research sub-questions:

a) *What are the general requirements of a machine learning-enabled multi-channel messaging platform?*
b) *What machine learning algorithms can be used to seamlessly determine the effective channel(s) to use for message delivery on an MCM platform?*
c) *What problems an efficient dynamic channel selection machine learning MCM platform solve for Financial Services Institutions using MCM for Customer Alert Systems?*

d) *How can MCM platforms be enhanced with machine learning algorithms to enable dynamic learning capacity of the channel selection module?*

1.5 RESEARCH OBJECTIVES

The main objective of this research study proposes a framework that addresses the challenge of dynamic channel assignment in a MCM alert messaging system while using ML to manage channel selection and routing of messages to customers. This main objective be fulfilled through the achievement of the following sub-objectives:

a) *To identify the general requirements of machine learning-enabled multi-channel messaging platform. This objective be achieved by reviewing existing MCM systems and eliciting requirements for a machine learning-enabled messaging platform.*
b) *To explore and evaluate the best machine learning algorithm that can be used to seamlessly determine the efficient channel or s to use for message delivery on an MCM platform. This objective be implemented through the review of existing ML algorithms, development and testing of ML algorithms used for channel selection.*
c) *To investigate the problems that an efficient dynamic channel selection machine learning MCM platform solve for Financial Services Institutions using MCM for Customer Alert Systems. This objective will be implemented by identifying the value-added services that will be created for FSIs using machine learning MCM for Customer Alert Systems.*
d) *To understand and evaluate the benefits of the application of machine learning in the channel selection module of an MCM platform. This objective will be realised by implementing a suitable machine learning algorithm in the channel selection module of the MCM platform.*

Figure 1.2 illustrates the link or s between the research question and objectives.

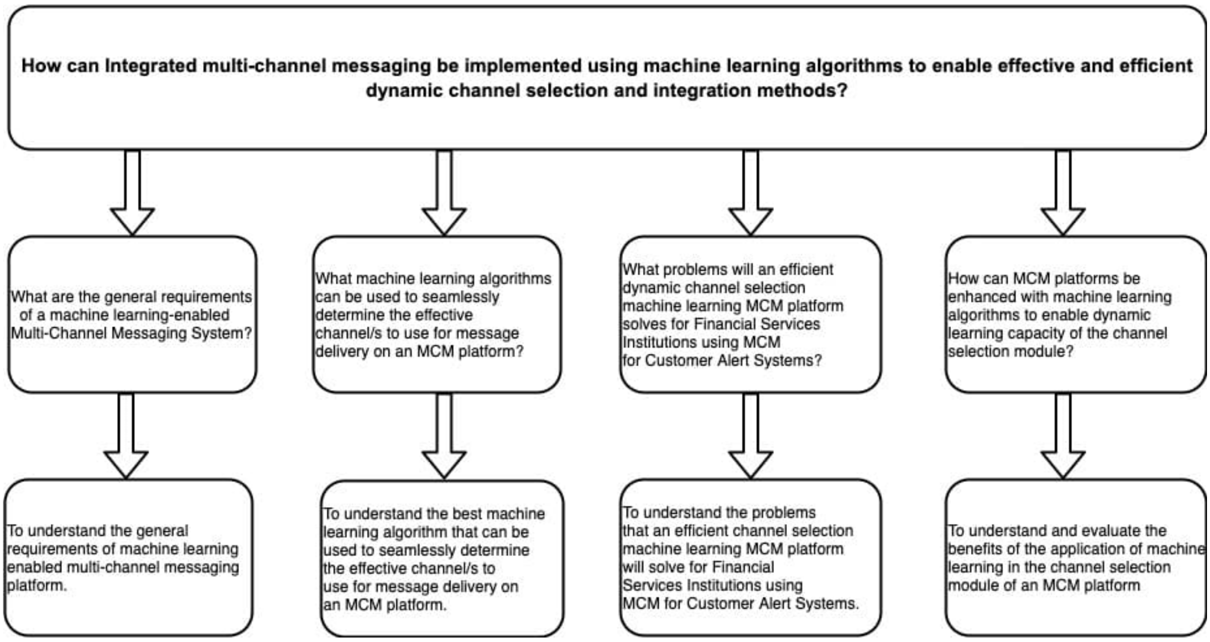


Figure 1.2: The link between research questions and research objectives

1.6 SIGNIFICANCE OF THE STUDY

This research study provides FSIs with a framework adapted to develop an ML-enabled customer alert messaging system that could provide seamless communication to their customers using ML capabilities. The following benefits could be derived from the implementation of the framework:

- **Automated and knowledge-based channel selection and management:** FSIs will be better equipped to automatically identify the channel(s) that customers are most receptive to. This approach will help the FSIs make cost and technological projections to ensure business continuity with ML-AI's power in cognitive channel selection.
- **Challenge and drives Innovation:** A well-implemented ML-enabled Customer Alert Multi-Channel system may expedite competition and drive innovation amongst the key market players in the financial services industry, thereby promoting an enhanced user experience.

This ML-enabled framework can help to optimize and streamline the implementation and maintenance of such systems. The study presents concepts and analyses the features of an ML-enabled multi-channel messaging system, focusing on the customer alert system used by

FSIs, all the while considering those elements vital to a successful implementation of the framework.

1.7 SCOPE AND LIMITATIONS OF THE STUDY

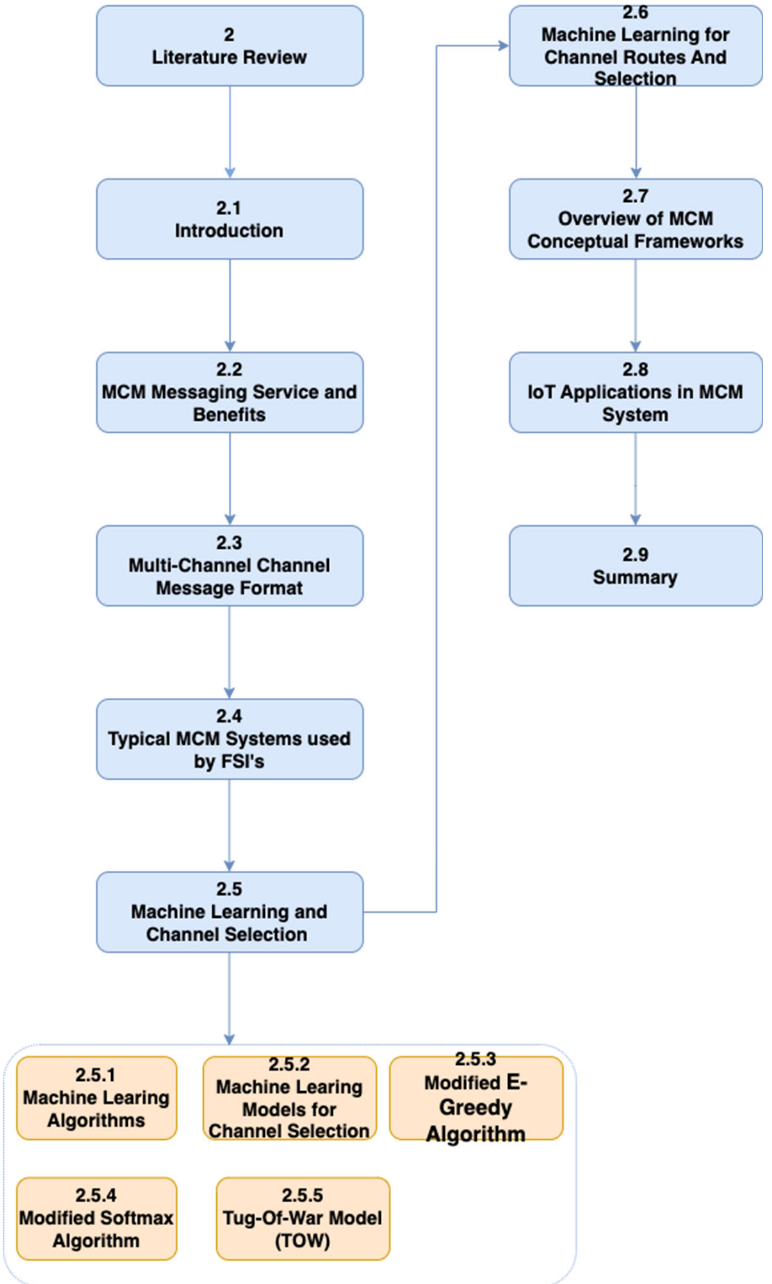
This study's scope is aligned and limited to the use of ML techniques in a Customer Alert System used by FSIs and channels known and available at the time of this study.

The General Data Protection Regulation (GDPR) laws on data access and confidentiality restrict FSIs to allow an in-depth review of the MCM applications hosted in their environments (Office, 2018). This study could have been enriched further if there was easy access to such customer-related data and usage patterns.

1.8 SUMMARY

In this chapter, we evaluated the concept of a ML-enabled multi-channel messaging platform. Our research goals and objectives towards delivering an ML-enabled MCM system were also defined. The next chapter presents the literature review conducted in this study.

2 CHAPTER 2 LITERATURE REVIEW



2.1 INTRODUCTION

According to Mbama (2018), a significant change in the retail financial services landscape has affected the way FSIs generate revenue and profits. Therefore, FSIs have had to provide seamless customer experience while transacting with the introduction of in-branch customer experience or social media channel interactions with FSI agents. How customers feel about using each channel for various transactions is critical for addressing the customer's experience with the service provider. The improvement in customer experiences by FSIs can determine the customer's future behaviour, affecting key business factors such as increased profitability and reduced service costs to the institutions involved. Chow (2017) opined that FSIs have access to big data about their customers, marketing, operations, accounting, and economic activities. However, FSIs need to integrate multiple channels for effective communication because most of them have implemented MCM features in their customer messaging platforms (Omarini, 2013).

This literature review chapter discusses some of the fundamental topics on ML and its application in multi-channel customer alert systems used by FSIs. This chapter provides an in-depth review of the benefits of ML and AI. These include the use of ML in channel selection for message delivery to FSI customers. The various algorithms available, including their strengths and weaknesses, were explored. Furthermore, journal articles published since 2010 in the area of ML was reviewed. This chapter assists in answering the following two sub-questions.

- *What are the general requirements of a machine learning-enabled multi-channel messaging platform?*
- *What machine learning algorithms can be used to seamlessly determine the effective channel or s to use for message delivery on an MCM platform?*

2.2 MULTI-CHANNEL MESSAGING SERVICE BENEFITS AND CONCEPTS

Effective management of the different communication channels made available to customers by businesses is critical in the new age. FSIs need to understand the detailed pattern for each customer segment. Such an approach enables the FSIs to identify the channels that would be effective and efficient to meet the customer's needs. Multiple-online channel capabilities may facilitate sophisticated customer segmentation and close customer experience gaps (MBAMA, 2018). As a multi-channel messaging system promotes sales to FSI customers across the various boundaries; it supports advertising and enables commerce via the various integrated channels (Aaron, 2018).

Customer's perception of a service offering is primarily dependent on the channel of delivery when it is implemented correctly. Financial services organizations can enhance their business portfolios by using the platforms of the organizations to alert customers to the various options that are available to meet their needs (Omarini, 2013). The impact of messages sent to customers should be maximized by ensuring that the customer's preferred channel is used at any time. These preferred channels can be pre-determined using ML analytics to ensure message delivery via the customer's appropriate messaging channel is based on analysed patterns (Amazon Web Services, 2020).

ML and AI methods have improved over the years, and have significantly improved in many areas of application and domains such as banking application chatbots, facial recognition applications, speech recognition, pattern recognition and dynamic ML (Munkhdalai et al., 2019). These AI and dynamic ML capabilities further ensure that channels are selected efficiently and effectively with integrations capabilities using a channel integration module. The concepts described in this section enabled us to answer the following research question:

"*How can integrated multi-channel messaging can be implemented using machine learning algorithms to enable effective and efficient dynamic channel selection and integration methods.*"

2.3 MULTI-CHANNEL CHANNEL MESSAGE FORMATS

Every channel has a unique messaging format when using heterogeneous systems (Oniga et al., 2020). To answer the sub-research question "*What are the general requirements of a machine learning-enabled multi-channel messaging platform?*", This study explored and investigated the channels available, including the format and protocols they support. The investigated channels are as follows:

1. **Short Messaging Service (SMS)**: SMS messaging formats are well documented in Request For Comments (RFC) 5724[1]. Brown et al. (2007) noted that SMS technology was created out of the Global System for Mobile Communications (GSMC) standard. SMS are limited to a maximum of 160 characters with 7-bit encoding. The default character sets for SMS are defined in the Third Generation Partnership Project (3GPP) when the full character set is exceeded. It is possible to extend it with other schemes such as 8 or 16-bit encoding. SMS is widely used by FSIs to deliver One Time Passwords (OTP) and to communicate customer-related information in real-time. SMS messages have support for plain text format only; it has no support for rich text or multimedia messages (Adagunodo & Bamidele, 2007).

[1] https://www.ietf.org/rfc/rfc5724.txt

2. **Multimedia Message Service (MMS)**: MMS has support for both plain text and multimedia content. MMS extends the functionality and capability of SMS, and it supports more than 160 characters of text, including rich media such as voice, graphics and video (Hasan et al., 2010). MMS shares the exact technical specifications as email with support for Multipurpose Internet Mail Extensions (MIME).
3. **Electronic Mail (email):** Email message is based on the Internet Message Format (IMF). IMF format relies on the Simple Mail Transfer Protocol (SMTP) for sending and receiving email messages (Koymans & Scheerder, 2008). IMF is well documented in the Request For Comments Standards (RFC) 5322[2]. Email supports plain text, multimedia, video, audio and attachments specified in the MIME standards. MIME format also allows support for in-line attachment of data within the body of the email message. FSIs use email communication to send communications emails, OTP's, Two Factor Authentication, password reset messages and secured key exchanges to customers on their platforms (Anitha et al., 2018).
4. **Voice Over Internet Protocol (VOIP)**: Voice Over IP allows users to make telephone audio and video calls via the Transmission Control Protocol or Internet Protocol (TCP or IP) using an application (Fayyaz, 2016). Jalendry and Verma (2015) explained that calls via the Internet use a signalling protocol that facilitates communication between the network components while using signalling for session establishment. The critical role of session establishment is further discussed below:
    - **Session Establishment:** allows the callee to have the ability to accept, reject or forward a call.
    - **User Location**: extracts and displays the geolocation of the caller to the callee.
    - **Call Participant Management:** responsible for allowing multiple parties to join or leave an existing call session.
    - **Session Negotiator**: manages the set of properties necessary to set up a call.

    VOIP is implemented on WhatsApp call, Skype and Microsoft Teams calls and others. FSIs use these channels for transactional banking with their customers (DeAngelis, 2019). Owing to the global COVID-19 pandemic, Deloitte (2020) noted an increase in the adoption of video collaboration tools by organizations.

[2] https://tools.ietf.org/html/rfc5322

5. **Twitter Direct Message (Twitter DM)**: Twitter messages support plain text and multimedia, including video and audio available on the web and mobile platforms. Twitter DM currently supports JavaScript Object Notation (JSON) format to store and retrieve messages (Kateb & Kalita, 2015). FSIs use this channel for automated chatbot and direct message delivery to customers on their platforms.
6. **Facebook Messenger**: Messenger on Facebook uses the traditional restful services via an Application Programming Interface for message exchange between users. Messenger has support for Bots implementation in use by FSIs for transaction management and inquiry (Media Labs, 2017). Facebook also implements JSON object format for message retrieval and submission on its platform and supports rich text, multimedia, audio, video and plain text. Messenger is available on the web and mobile platforms.
7. **WhatsApp**: WhatsApp Messenger is available on desktop and mobile devices. It implements Extensible Messaging and Presence Protocol (XMPP) with Extensible Mark-up Language XML format for its message format. WhatsApp is a freeware platform that supports End To End Encryption (E2E) of information exchange between users (Udenze, 2020). WhatsApp also supports chatbot's development used by FSIs to engage the customer on their inquiry or transaction status. Appendix E depicts the formats supported by each channel.

This section explored the various messaging formats and protocols they support for message sending. Some channels share similar features such as support for plain text messages, multimedia messages, audio, and video, while others do not. One key feature that stands out is that each channel has its standardized implementation framework. FSIs use each channel depending on their relevance to reach their customers based on need or engagement criteria.

2.4 TYPICAL MULTI-CHANNEL MESSAGING SYSTEM USED BY FSIS

The current transaction alert system implemented by most financial institutions has been achieved with the multi-channel approach of SMS, emails and social media channels integrated into a customer alert transaction system (Cortiñas et al., 2010). This study reviews the publicly available applications deployed on Android Play Store and Apple iOS Store by FSIs. The components of the customer MCM system listed below details the design of the existing MCM system in use by FSIs (Jeroen van Disseldorp, 2017):

- MCM-Alert Profiling System
- MCM-Alert Polling System
- Integrated Middleware Channel or Enterprise Service Bus for Decision Making

Figure 2.1 shows the interdependencies of the 3 systems integrated into the FSI's core systems in a customer transaction alert system.

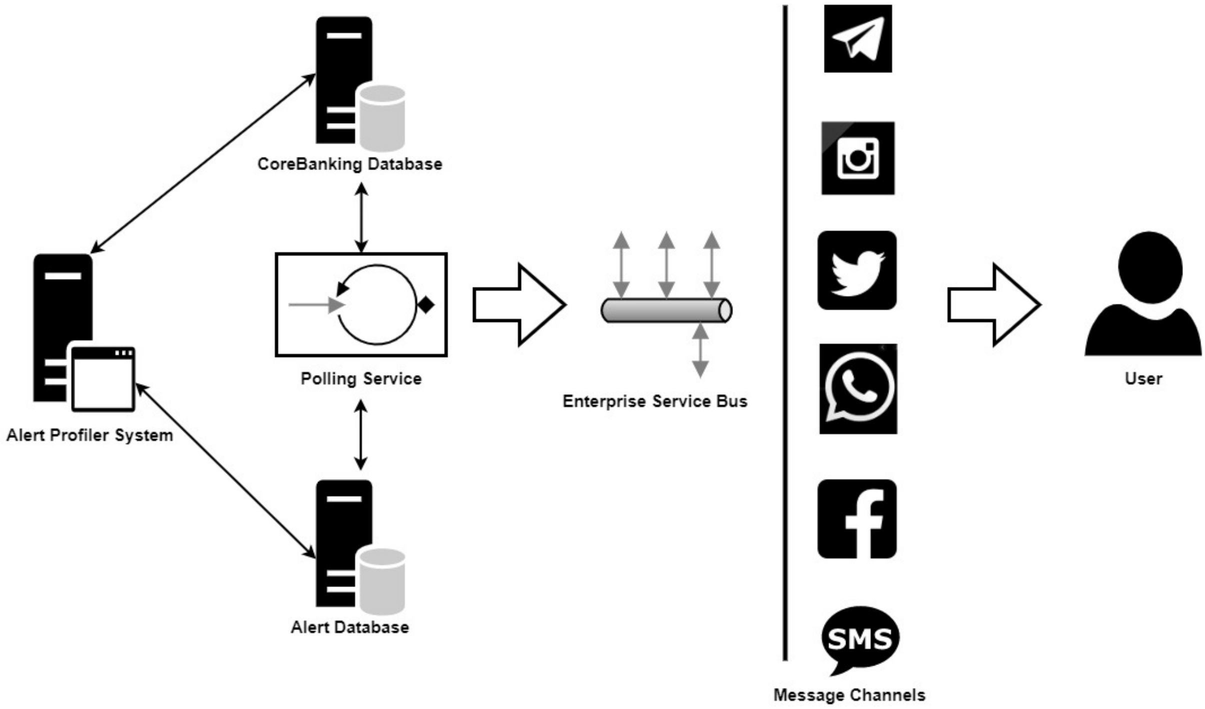


Figure 2.1: Typical Current Customer Alert Framework Used by the Applications Source: (Salami & Mtsweni, 2016)

The individual descriptions of the 3 main components are as follows:

a) **MCM-Alert profiler system:** Hosted as a GUI-web application deployed on-premises for FSI branches to set up each customers preference. Customer's data is retrieved using the primary account number in the database. This account number is linked to the customer's preferred messaging channel. Channels supported include SMS, email, Twitter Direct Message, Facebook Messenger, and WhatsApp. Other preferences the user has access to includes the following such as masking of the account number in messages and balance display. Once the customer's details are validated as correct, the system stores the information on the customer alert database (Kabari & Baah, 2015).

b) **MCM-Alert Polling Services:** This is implemented as a service that polls both the core systems and the customer alert database. The polling service runs at intervals to monitor events happening on each customer account in real-time and creates an entry in the database for each record.

c) **Integrated Middleware or ESB:** The ESB manages interactions between the MCM-Alert profiler and Polling services. It controls the channels connected and ensures message translation and transformation between the channels. The ESB layer also manages failure and retrials in the case of channel-connectivity failure.

The MCM system depends on the seamless integration of the components described above. Each component interacts with each other to deliver the actual message to the customer. To implement a framework for an efficient and effective dynamic channel selection, MCM enables the customer alert system to be enhanced with ML algorithms embedded within the system to coordinate the ESB layer activities for dynamic channel management and selection. This dynamic module integrated with the system's core ensures channel availability and learning using a simple learning procedure.

2.5 MACHINE LEARNING AND CHANNEL SELECTION

Machine Learning (ML) has increasingly become popular across the years with proven success in learning complicated models and patterns in systems (Chow, 2017). ML supports creating models that can immediately recognize patterns, classify input data into different categories, and make predictions (Rahimi et al., 2020). ML and AI have continuously been an evolving research area. This research aims to integrate ML algorithms for channel selection in a customer transaction alert system (Ma et al., 2019). Shalev-Shwartz and Ben-David (2014) explored the linkage between ML and ML algorithms. Futhermore, the authors documented that training data sets are required as inputs for the ML algorithms used to generate an output for other downstream computing outputs. ML enables computer systems to learn directly from training datasets and experience. Vansh (2019) listed several factors for selecting a suitable algorithm for a specific problem. These factors are discussed in detail in Section 2.5.1.

2.5.1 Machine Learning Algorithms

Pappad and Francesco (2019) proclaimed that ML algorithms can be used for business decisions that may directly or indirectly affect individuals, as in the case of a customer loan application. Selecting a suitable ML algorithm has proven to be a challenge, and this elicited some factors that can assist with choosing an ML algorithm (Vansh, 2019). These machine learning approach includes type of algorithm, parameterisation, memory size, overfitting tendency, time of learning and time of predicting (Vansh, 2019).

These machine learning techniques and features are described in more detail as follows:

A. **Regression**: A technique used to predict a dependent variable in a set of independent variables.
B. **Classification**: Classification ensures that the mapping functions derived from the inputs are aligned with the output variables.
C. **Clustering**: This is defined as similar data points grouped together in a distribution of data sets. These grouped data points are similar and different from other data groups in the same distribution.
D. **Parametrization:** ML algorithms use parameters to develop historical training data sets for the model to be used.
E. **Memory Size:** This is defined as the space needed to store datasets and other associated variables. Deep Neural Networks application are memory intensive due to the calculation involved with various algorithms.
F. **Overfitting Tendency:** Overfitting tendencies is usually due to data discrepancy in the complexity of the models used.
G. **Time for Learning:** This is the length of time used in training the model with the available dataset. The time for learning is highly dependent on the size of the dataset
H. **Time for Predicting**: Prediction time varies with the size if the testing dataset and the prediction algorithm selected.

The classification of each algorithm is summarised in Table 2.1.

Table 2.1: Machine Learning Algorithms Selections Criteria: **Source (Vansh, 2019)**

Algorithm	Type	Parametrization	Memory Size	Overfitting Tendency	Time for Learning	Time for Predicting
Linear Regression	Regression	Weak	Small	Low	Weak	Weak

Algorithm	Type	Parametrization	Memory Size	Overfitting Tendency	Time for Learning	Time for Predicting
Logistic Regression	Regression & Classification	Simple	Small	Low	Weak	Weak
Decision Tree	Regression & Classification	Simple or Intuitive	Large	Very High	Weak	Weak
Random Forest	Regression & Classification	Simple or Intuitive	Very Large	Average	Costly	Costly
Boosting	Regression & Classification	Simple or Intuitive	Very Large	Average	Costly	Weak
Naïve Bayes	Classification	No Parameters	Small	Low	Weak	Weak
SVM	Classification	Not Intuitive	Small	Average	Costly	Weak
Neural Networks	Classification	Not Intuitive	Inter	Average	Costly	Weak
K-Means	Clustering	Simple or Intuitive	Large	High	Weak	Weak

ML algorithm selection is essential for implementing a framework that can implement an AI or ML-enabled Customer Alert Multi-Channel Messaging System. Channel selection and availability algorithms are loaded into memory and run using the host resources. This execution must be optimal to achieve excellent results. One of the most critical elements of the Customer Alert ML-enabled system is channel reuse based on past successful deliveries to customers using a specific channel. This aspect is achieved via algorithms with optimized regression and parametrization techniques that use previous data to make real-time decisions. It is essential to select a mix of algorithms to achieve the desired result since each algorithm has its strengths and weakness, as indicated in Table 2.1. An implementation between linear regression and neural networks may be most desirable for this framework owing to the complementary nature of their strengths and weaknesses.

Furthermore, these algorithms have become standardized across various implementation models that can be simulated and modelled against real-life problems. These machine learning algorithms are explored further in subsequent sections.

2.5.2 Machine Learning Models for Channel Selection

This study explores the use of a simple ML-based Multi-Armed Bandit (MAB) problem (Kim et al., 2010; Ma et al., 2019) as it applies toward channel selection and the ability to derive the optimal solution to the channel selection problem with high probabilities. MAB is expressed as a set of outcomes such as rewards or losses for the player expressed in Bernoulli distributions, and the player receives a value of reward of failure at an instance of time (Belzner & Gabor, 2016). An activity or task is used to identify the optimal arm while maximizing its payoff at the same time. Wang et al. (2018) declared that the player may select an option set of arms at each time opportunity to play. Each option of the arm, if selected, can result in a reward or loss for the player. Niculescu-Mizil (2009) elicited further that a player's goal endeavours to maximize his compensation (exploration and exploitation) from the options available for selection. Lai and Robbins (1985) explained the trade-offs for the player selection policy's exploration-exploitation approach.

Herbert (1952) described MAB as an ML problem based on the situation faced by a player who desires to earn maximum reward from multiple slot machines. The MAB problem serves to detect using infinite trials the machine slot that the player should select to maximize his reward amount at an instance of time. There is a general assumption that the player has no previous knowledge about the outcome of any machines or slots. Each player naturally tries as many options as possible and estimates each machine's reward, which gives him the best rewards (Thompson, 1933). The player has to ensure that he makes a trade-off while striking a balance between exploitation and exploration strategy, as suggested by Kim et al. (2010). Ma et al. (2019) is of the view that solving this MAB trade-off is essential and further presented algorithms such as the Ɛ-greedy, Modified SoftMax Algorithm and Tug of War (TOW) Model.

2.5.3 Modified Ɛ-greedy Algorithm

There are different algorithms designed to address the MAB problem. The Ɛ-greedy algorithm ensures that a player displays a greedy action when selecting an option but with probability Ɛ that the player pick an activity at random. Selecting this action ensures that the player has a simple way of balancing exploration and exploitation (Kim et al., 2010). Gutowski et al. (2019) reported that the objectives to be fulfilled by the players include maximizing the significant rewards earned while minimizing the loss (regrets). However, in this study, the "average accuracy rate" documented by Kim et al. (2010) was used because of an interest in the short-term behaviour of the player and not a logarithmic approach for long term behaviour of the player. A player can make a random selection between options A or B, with the probability Ɛ otherwise, he/she selects a greedy action based on the estimates given in Eq. 2.1 and 2.2 expressed as 1- Ɛ. Selecting a greedy action translates to a player choosing option A, if $Q_A > Q_B$ or selects option B if $Q_A < Q_B$.

Ɛ: *random selection* and 1 – Ɛ: *greedy action based on estimates*

$$Q_A(t) = \frac{[number\ of\ rewards\ from\ A]}{[number\ of\ A\ selections]} \qquad \textbf{Eq. 2.1}$$

$$Q_B(t) = \frac{[number\ of\ rewards\ from\ B]}{[number\ of\ B\ selections]}$$

probabilities P_A and P_B are the estimates for the reward probabilities.

$$\varepsilon(t)\frac{1}{1+\tau .t} \qquad \textbf{Eq. 2.2}$$

where $\boldsymbol{\tau}$ is defined as the parameter that controls the decay rate.

This study uses a time-dependent probability Ɛ(t) expressed in Eq. 2.3 (Kim et al., 2010) In this case, the player makes a random selection of an arm with the probability Ɛ or implements a greedy action based on estimated values of (Q_1, Q_2,………Q_M) using a probability 1 – Ɛ with $Q_K(t)$ (K=1,2, …,m) specified as follows:

$$Q_k(t) = \frac{[number\ of\ rewards\ from\ k]}{[number\ of\ k\ selections]} \quad \textbf{Eq. 2.3}$$

The Modified Ɛ-greedy algorithm provides a starting base for the research to provide a platform for the exploitation and exploration approach. However, this algorithm has been extended further to accommodate time dependency requirements owing to the heterogeneous nature of the channels being considered. The parameters needed for their integration include an agnostic message layer and an integrated decision-making module with dynamic learning capabilities.

2.5.4 Modified SoftMax Algorithm

The modified SoftMax algorithm, which is also referred to as multinomial logistic regression, is a generalization of logistic regression that allows the handling of multiple classes and is quite useful for neural networks where non-binary classification is not needed. According to Gao and Pavel (2017), the modified SoftMax ensures a player can balance the opportunities available for the exploitation and exploration techniques within the MAB problem. This technique guarantees the context of decision-making while ensuring that every strategy in a player's possession has a chance of being explored, unlike some other algorithm such as Ɛ-greedy (Sutton & Barto, 2011).

In Figure 2.2 the players choose some strategy and play the game and receive payoffs. The players then use some learning rule to convert the payoffs into scores independently (Gao & Pavel, 2017a). In the end, each player uses the modified SoftMax algorithm to select the next strategy in that cycle to enhance his rewards.

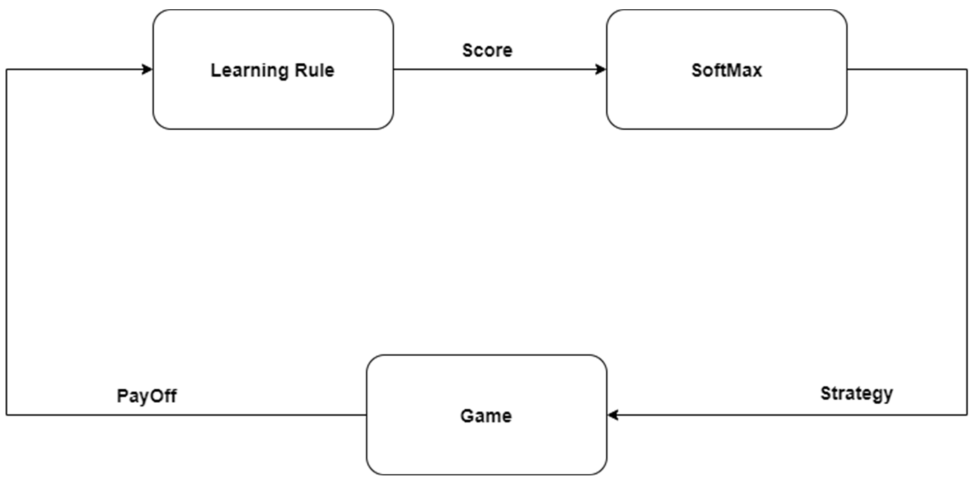


Figure 2.2: Modified SoftMax Algorithm Explorative and Exploration Strategy Source: (Sutton & Barto, 2011)

The modified SoftMax algorithm is represented by the Boltzmann distributions depicted in Equations 2.4 and 2.5 (Kim et al., 2010; Sutton & Barto, 2011; Young & Shmuel, 2014), where the player has the probability of choosing A or B. This probability is expressed as $P_A(t)$ or $P_B(t)$ as follows:

$$P_A(t) = \frac{\exp\,[\beta\,.P_a(t)]}{\exp[\beta\,.P_a(t)] + \exp\,[\beta\,.P_b(t)]} \quad \textbf{Eq. 2.4}$$

$$P_b(t) = \frac{\exp\,[\beta\,.P_b(t)]}{\exp[\beta\,.P_a(t)] + \exp\,[\beta\,.P_b(t)]}$$

Eq. 2.5

where Q_A and Q_B are represented by Eq. 2.1 and 2.2, respectively. β is similar to Ɛ in the Ɛ-greedy algorithm, β is further revised to a value that depends on time for this study as follows:

$$\beta(t) = \tau\,.\,t \quad \textbf{Eq. 2.6}$$

Whereas, if β is evaluated to 0 it was considered as a random selection, and if β tends towards ∞ it was equivalent to a greedy action. The Modified-SoftMax algorithm is used to enhance the MAB problem further. The probability of choosing an arm k, P_k (t) (k=1,2…….m) is expressed as follows:

$$P_k(t) = \frac{\exp\,[\beta\,.Q_k(t)]}{\sum_{j=1}^{m} \exp\,[\beta\,.Q_j(t)]} \quad \textbf{Eq. 2.7}$$

The Modified SoftMax algorithm was used to expand the exploitation and exploration approach for this research work. This algorithm extends the MAB problem in the context of the downstream decision-making module for the designed framework. The Boltzmann factor approach also enhances the context of the decision-making process of the framework.

2.5.5 Upper Confidence Bound Action (UCB)

UCB is a novel ML approach with Reinforcement Learning. This approach has found its use in the MAB problem. According to Jamieson et al. (2014), the UCB approach is used to determine the arm with the highest mean in a MAB game with a fixed confidence setting and a short number of total samples. The approach cannot be improved since the number of samples necessary to determine the best arm is within a constant factor of a lower bound based on the iterated logarithm law (LIL) (Jamieson et al., 2014).

Consider a Baby Robot that has become disoriented in a mall, and it must use Reinforcement Learning to assist it find its way back to its mother. However, before it can even start hunting for the mother, it must refuel using a series of power outlets each of which provides a slightly different amount of charge.

The Baby Robot has now entered a charging room with five separate power outlets. Each of these sockets gives a slightly different amount of charge back. The aim is to charge Baby Robot as quickly as possible; therefore, it must choose the best socket and use it until charging is complete. This is the same as the MAB problem, however, instead of seeking for the biggest reward, the Baby Robot must look for the most charge from a power outlet.

Figure 2.3: Baby Robot using Reinforcement Learning Approach (MAB)

In the problem stated above, the Baby Robot must get the highest charge in the shortest possible time. In the modified Ɛ-greedy algorithm approach, the baby robot needs to keep exploring the set of all actions long after gathering enough knowledge to know which of these acts are undesirable to take (Kim et al., 2010). Rather than merely undertaking exploration by selecting an arbitrary action with a fixed probability, the UCB algorithm alters its exploration-exploitation balance as it learns more about the environment. The UCB algorithm shifts from being primarily focused on exploration, where the least tried activities are preferred, to being focused on exploitation, where the action with the biggest estimated return is chosen (Zhang et al., 2019).

Through the utilisation of uncertainty in action-value estimations, UCB action selection strikes a balance between exploration and exploitation. When a sample set of rewards is used, there is an inherent uncertainty in the accuracy of the action-value estimations. Due to this, UCB makes use of the uncertainty in the estimates in order to promote exploration.

So therefore, a UCB (*A_t*), for selected action at a time step '*t*', is expressed in Eq. 2.8 as:

$$A_t \quad = \arg\max(a) \left[Q_t\ (a) + \sqrt[c]{\frac{logt}{N_t}}\right] \qquad \textbf{Eq. 2.8}$$

where:

- Q_t(a) is the estimated value of action 'a' at time step 't'.
- N_t(a) is the number of times that action 'a' has been selected prior to time 't'.
- 'c' is a confidence value that controls the level of exploration.

In line with the MAB approach, the UCB algorithm is distinctly separated into the following two main parts:

- **Exploitation**: Where Q_t(a) depicts the exploitation part of the equation above. UCB uses the principle of "**optimism in the face of uncertainty**", which essentially means that if you are unsure about the best course of action, you must take the one that seems to be the best at the time. By focusing only on this component of the equation, the selected action will be that which presently has the highest estimated payoff.
- **Exploration**: The exploration component is added in the second half of the equation, and the hyper-parameter *"c"* determines how much exploration is added. The uncertainty of the action's reward estimate is essentially measured by this component of the equation. The exploration strategy gives the agent the option to choose the action that will yield the greatest immediate payoff. Within the suggested system, purely greedy action selection may result in undesirable behaviour of the system.

The exploration and exploitation approach with UCB is further discussed below:

- $N_t(a)$ will be low if an activity has not been attempted very frequently or at all. As a result, there will be a lot of uncertainty and ultimately increasing likelihood that this course of action will be chosen. The player gains confidence in an action's estimate with each action. It is less likely that this course of action will be chosen as a result of investigation in this scenario since $N_t(a)$ increases and the uncertainty term drops (although it may still be selected as the action with the highest value owing to the exploitation term).
- The numerator's log function causes the uncertainty term to increase slowly when no action is chosen. However, because the growth in $N_t(a)$ is linear, the uncertainty will quickly decrease each time the action is chosen. Owing to the uncertainty in the estimates of their benefits, the exploration term will therefore be longer for actions that have been chosen infrequently.
- The exploration term rapidly reduces over time (because *log n or n* approaches zero as '*n*' approaches infinity), leading to the eventual selection of actions solely based on the exploitation term.

2.5.6 Tug-of-War Model

Kim et al. (2010) and Ma et al. (2019) described the TOW model as a unique ML method with extended use for the MAB problem. Kim et al. (2016) explored the MAB problem efficiently with fluid dynamics in cylinders. Furthermore, it was stated that the TOW model provides significant efficiency than other algorithms, for example, the Modified-SoftMax algorithm and the Ɛ -greed algorithm (Kim et al., 2016). Ma et al. (2019) implemented this approach to cognitively select channels for communications in massive Internet of Things (IoT), which is depicted in Figure 2.4 with the volume conservation law.

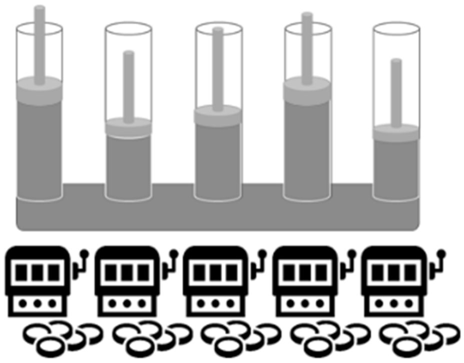


Figure 2.4: TOW Dynamics In Incompressible Branched Cylinders (Ma et al., 2019)

Kim et al. (2016) demonstrated TOW concept with fluid in the tube. The TOW method ensures that the fluids' density remains constant when the velocity of the fluid changes, as shown in Figure 2.4. X_k is equivalent to the movement of the arm $\underline{k}$ from an original position. The value of the parameter is expressed as $k \in$ Arm_1, Arm_2,…. Arm_K. Q_k ($k \in$ $Branch_1$, $Branch_2$,….$Branch_K$) and is represented and calculated as follows:

$$Q_k(t) = \Delta Q_k(t) + \propto Q_k(t)(t-1), (0 < \propto < 1) \quad \textbf{Eq. 2.9}$$

where $\Delta Q_k(t)$ is expressed as +1 (reward) or ω (failure) according to the selection outcome, the value ω is defined as the weighting parameter.

The parameter α controls how previous information can be quickly forgotten. The change in value X_k is derived from the difference in the following formula:

$$X_k(t+1) = Q_k(t) - \frac{1}{N-1} \sum_{i \neq k} Q_i(t) + os$$

Eq. 2.10

The fluctuation the fluid is subjected to is expressed as random value osc. When machine k is played each time t (Kim et al., 2016), +1 is added to Xk(t) if the player receives a reward. Otherwise, −ω is added to Xk(t). Following these actions, the interface's levels shift in line with the volume conservation law, and this law determines the next action taken by the player. According to Ma et al. (2019), TOW dynamics support an efficient cognitive search method while maximizing that channel selection is as correct as possible.

The TOW algorithm or UCB algorithm complements and enhances the Modified SoftMax and Modified Ɛ-greedy algorithms by providing a suitable option for dynamic channel selection, which is the core of the framework designed in this research. TOW model further implements simple learning using acknowledgement messages received from successful messages delivered by each channel to users while providing minimal memory and computation capabilities for a heterogeneous MCM system. The UCB interval was used in this study to determine the channel available for message delivery.

In this section, an in-depth review of ML algorithms used for channel selection was conducted to answer the following research question:

What machine learning algorithms can be used to seamlessly determine the effective channel or s to use for message delivery on an MCM platform?

At the centre of each ML algorithm are data inputs, analysis, decision-making, and ultimately a favourable outcome. Fundamentally, in each algorithm, the channel selection logic is modelled around the MAB problem with reward or loss outcome at an instance of time. The Modified SoftMax and Modified Ɛ-greedy introduced the concepts for exploration and exploitation critical for channel selection and management, with limitations on acknowledging successful messages and time dependency. The TOW algorithm enhances these limitations with robust dynamic channel selection and the message acknowledgement approach needed to design the framework.

2.6 MACHINE LEARNING FOR CHANNEL ROUTES AND SELECTION

In this section, the focus is on the application of ML to channel route availability and selection in a Multi-Channel Messaging system framework. ML establishes a framework to facilitate the system's reconfiguration, including its application to a wide range of applications (Alsheikh et al., 2014). Learning techniques use the outcomes obtained from the environment's perception and the reconfigurability of the interfaces involved to enhance the available resources. In other words, messaging channels can optimally adapt to the communication environment through learning (Zhou et al., 2018). The machine learning relationship between the entity and the environment is depicted in Figure 2.5.

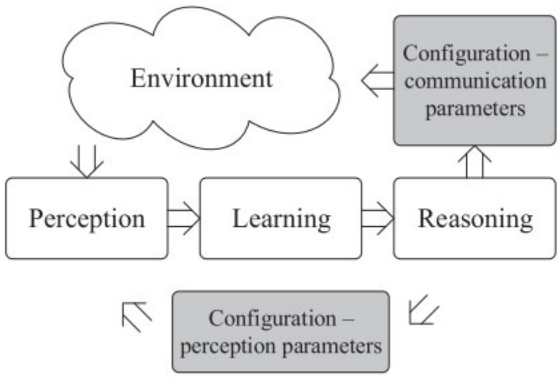


Figure 2.5: Machine Learning In Intelligent Communications (Zhou et al., 2018)

Bishop (2006) opined that ML algorithms learn to achieve a specific task (T) based on a unique learning that produces an experience (E), which, by utilising prior experience (E), improves the task's performance as measured by the performance metric (P). The parameters T, E and P, determine the outcome of the learning at any instance. ML provides the platform for developing algorithms that give computers the ability to learn. Nasteski (2017) asserted that learning means identifying statistical regularities or other patterns of data. ML algorithms are developed to reflect and adopt a human's approach to learning. ML techniques are further divided into 3 categories, namely Supervised, Unsupervised and Reinforcement Learning. The catergories are decribed in detail below.

a) **Supervised Learning**: Supervised learning see Figure 2.6 completes tasks by learning from any external activity. Also, every training example includes a pair of input data and expected output, and the aim is to develop a function that correctly predicts output for every kind of input (Zhou et al., 2018). Supervised learning methods have been used to collect and analyse various data (Nasteski, 2017). The algorithm produces a function that maps inputs to the desired output. The classification problem is one of the standard formulations of the supervised learning task: the learner must learn to approximate a function's action and then map a vector into one of many groups by looking at several input-output learnt over time.

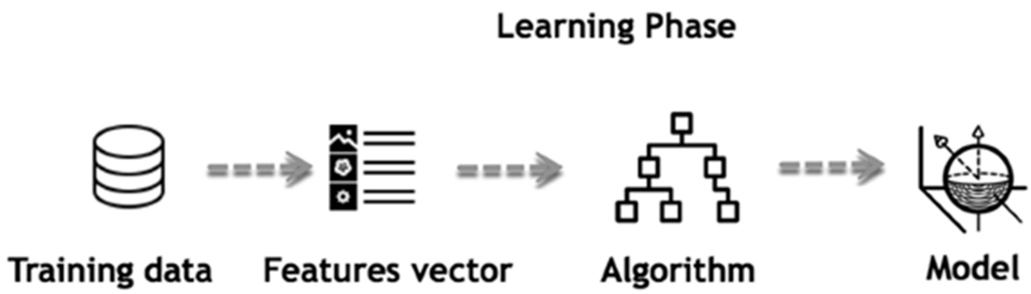


Figure 2.6: Supervised Learning Model

The learning process in a standard ML model which is categorized into two, namely: training and testing phases (Singh et al., 1999). The training phase consists of collecting samples of training data as input with attributes that are learned by a learning algorithm to create a learning model. In the testing phase, the learning model uses the execution engine to make predictions for the target output.

b) **Unsupervised Learning**: Unsupervised learning algorithms see Figure 2.7 typically consider that there are no labelled samples. The primary objective is to determine the hidden composition of the input data. Unsupervised learning uses clusters. Observations in the same clusters are more comparable, and observations in separate clusters are less similar (Zhou et al., 2018). Clustering has a range of uses in smart multi-channel selection applications. Unsupervised learning methods proposed by Pinheiro et al. (2020) have been noted to take advantage of a large quantity of unlabelled data.

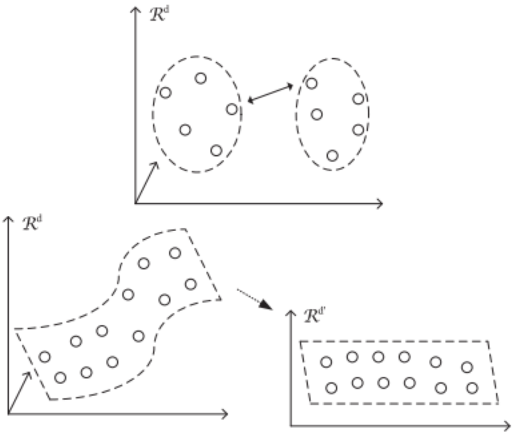


Figure 2.7: Unsupervised Learning Model (Zhou et al., 2018)

Younis and Fahmy (2004) are predisposed to a hybrid, energy-efficient, distributed clustering approach (HEED) for ad hoc network sensor groups when the allocation of channels is fixed. The topology management algorithm suggested by Chen et al. (2007) solves network creation in a cognitive radio context. The algorithm improved the settings of the cluster to react to changes in the network or environment. A clustered, spectrum-aware clustering technology was developed by Zhang et al. (2011) to classify energy-efficient clusters and reduce interference in the cognitive radio sensing networks to primary users to efficiently aggregate source information under energy constraints. The objective was to define clusters to decrease communication capacity and minimize squared distances between the cluster centre and each node.

c) **Reinforcement Learning**: Reinforcement learning see Figure 2.8 enables an agent to learn or perform specific tasks through interaction in a complex environment with negligible failure. A stark difference from supervised learning allows the agent to get feedback by merely engaging with the environment and learning itself, which makes the concept of Reinforcement Learning very useful in situations where decision making must happen with uncertainty. Reinforcement learning is quite beneficial when knowledge about the environment is limited. An agent tries to learn and relate to its operational environment with considerable uncertainty like Multi-Channel selection and learning in a Customer Alert System (Zhou et al., 2018). At any time, the agent aims to optimize rewards by exploring and exploiting its operational environment.

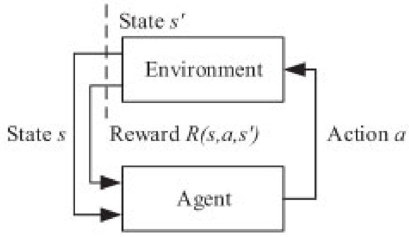


Figure 2.8: Reinforcement Learning Model (Zhou et al., 2018)

Berthold et al. (2008) proposed the Markov Decision Process (MDP), a mathematical formula used to model decision making under uncertainty. Modelling a problem with MDP involves the following factors: space (S), the action space (A), the transition probabilities (T) and the reward functions (R). The state-space S defines all the channels implemented in the MCM scenario. The action space A determines channel detection, message transmission on a current channel and out-of-channel detection, which is responsible for selecting a new channel when the existing one is unavailable for message delivery.

State transition only occurs if the parameter for switching to another channel is specified in the system configuration. The reward function R(s, a, s) is defined by transmitting a message in the current channel, and the second variable is the reward when performing channel switching. The third variable is the reward for switching successfully to the different channel due to lags in the communication framework of the MCM system. Zhou et al. (2018) concluded that the MAB problem is a simplified version of the full MDP where the agent's environment has many states, and new states rely primarily on previous condition actions.

Wang et al. (2018) asserted that the MAB and Multi-Player MAB are robust in channel selection in a multi-channel communication system. These methods apply to environments where channel conditions such as selection and availability must be "explored" and "exploited". Table 2.2 summarizes the ML models discussed so far.

Table 2.2: Comparison Of Machine Learning Algorithms

Category	Learning Algorithms	Characteristics	Applications
Supervised Learning	Logistics or SVM-Linear	Linear Separable Input easy to train	Spectrum Sensing Channel Selection
	Bayesian Net or HMM	Statistical models, interdependent outputs	Channel Estimation, Spectrum Sensing
Unsupervised Learning	HEED	Clusters adapt to data fully Bayesian	Network Clustering
Reinforcement Learning	Multi-Arm Bandit	Learning in stateless environment exploitation and exploration	Channel Selection

Jiang et al. (2014) have concluded that Multi-Channel Selection and Sensing are modelled as an Indian Buffet Game, where the secondary users are customers, and the main channels are depicted in the restaurant as several dishes. A cooperative approach was used to estimate the channel states with Bayesian learning to resolve the multi-channel sensing problem, typically using Supervised Learning techniques. In this study, it is expedient to explore the implementation using a hybrid of Supervised Learning (mapping of message endpoints as inputs to the delivery channels as outputs) (Zhou et al., 2018) and Reinforcement Learning using the MAB approach for channel selection (Wang et al., 2018). A hybrid Supervised-Reinforcement learning approach offers the following benefits:

- Channel selection feature of Supervised learning with statistical models mapped to interdependent outputs.
- Supervised learning supports the creation of labelled inputs from message producers to the delivery channels within the framework.
- The ML-enabled MCM framework is a stateless system that continually needs to learn about its environment, hence modelled as a MAB problem supported by Reinforcement learning.

The channel selection logic depends largely on balancing exploitation and exploration with Reinforcement learning.

2.7 OVERVIEW OF MCM CONCEPTUAL FRAMEWORKS

In this study, the author reviewed existing Multi-Channel Messaging system implementations. The research focused on customer transactional alert messaging system implemented with multiple heterogeneous channels for message delivery (e.g., email, Twitter DM, SMS, Facebook and WhatsApp) (Salami & Mtsweni, 2016). This system can only choose one channel to deliver the transaction messages to the customer's device based on a customer's pre-selected channel. Each MCM channel is directly integrated into an Enterprise Service Bus (ESB) layer, which provides message translation (see Figure 2.9.

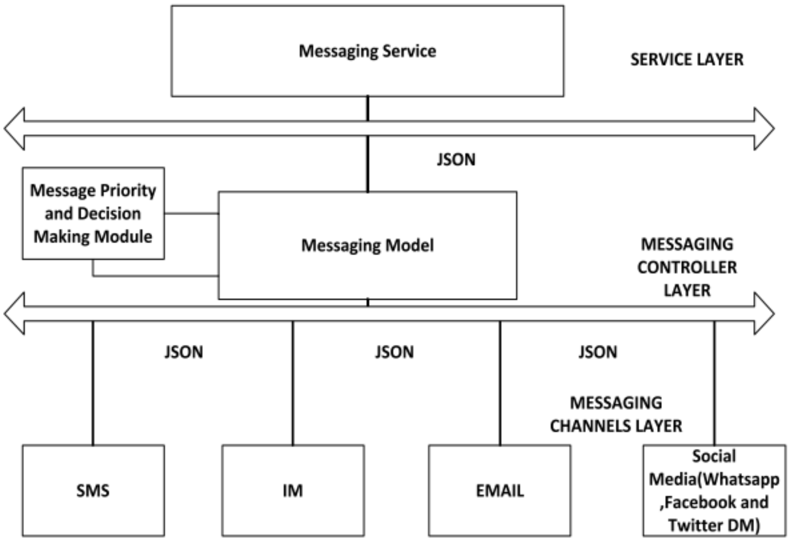


Figure 2.9: MCM Customer Alert System Model With ESB Integration (Salami & Mtsweni, 2016)

At time *t*, a scheduler randomly polls the messaging service queue to access transaction alert messages created by customers within an instance of time. Once a message is received on the queue. The message is translated to a standard agnostic data type JavaScript Object Notation (JSON), a lightweight data format. JSON is used to move messages around and within the platform since each channel implementation accepts different message formats. This structure is further converted to each channel messaging data format at the point of delivery.

The ESB uses the message priority and the feedback of the decision module to determine the channel to relay the message delivery. The existing MCM model decision making module solely uses the preferred or pre-set messaging channel chosen by each customer. There is feedback from the channel output that indicates whether the message is delivered or not. In the case of the non-delivery of messages, the system re-routes the message to the next available channel in the order of manual preference set by the customer.

Liang et al. (2011) implemented an integrated MCM model that combines different communication channels within the same integrated messaging layer. The model utilized message translation languages to manage information sharing between the channels with XML and XLST. The model can determine the urgency of the message and a message-transformation module that utilizes XLST or XML to adapt the message structure to the service layers, namely SMS, email and instant messaging. XLST or XML incurs overheads in data transformation and translation due to redundancy issues in XML or XLST schemas.

Pankowski and Pilka (2009) used cyclic XML for the normalization of XML schemas. These dependencies are in the form of XML Form Definition Format (XFDs) (Xu & Özsoyoglu, 2005). Performance overheads encountered in XML transformation were reduced using tree pattern analysis. Furthermore, Pankowski and Pilka (2009) reported that the XML schemas lack relational data relationships, hence the need to normalize the XML with Boyce-Codd Normal Form (BCNF) redundancy in the document schema.

Lim et al. (2008) designed a framework with a communication middleware to emphasise how the integrated layers communicates with each other. Channel access management and information exchange were handled using socket wrappers. Message transformation and handling were primarily based on the Simple Object Access Protocol (SOAP). However, the authors also noted a decrease in performance arising from the verbose nature of the XML, basically because SOAP or XML primarily relies on XML schemas for defining the Web Service Definition Language (WSDL). Kafeza et al. (2006) developed an alert system with a middleware based on a process scheduler that is primarily responsible for transforming and sending messages to the users. The alert system also incorporated a logic for scheduling and parallel execution of the system processes. This parallel execution ensures that the instantiation of multiple actions within the system can be executed simultaneously without interfering with each other.

In all the frameworks reviewed, channel intercommunication was standard across the board, including channel selection and management, which was handled manually. This standardised approach necessitates automation of the channel selection logic that ML can provide seamlessly.

2.8 INTERNET OF THINGS APPLICATIONS IN MCM SYSTEMS

IoT involves an interconnected network of physical objects. In today's world, integration is critical. Peterson et al. (2010) stated that an MCM integrated customer model could provide customer satisfaction and meaningful return on investment (ROI). This approach requires a deep understanding of customer needs and behaviours as well as a robust information technology (IT) architecture that supports the overall customer relationship management (CRM) strategy. Not only is the world wide web a computer network, but it has also transformed into a network of devices of all sorts and sizes (see Figure 2.10). These include cars, mobile devices, smartwatches, printers, and security alarm systems all linked while exchanging information based on stipulated protocols to achieve smart communication and interactions (Patel et al., 2016). IoT models manages the digital and physical world in synergy. IoT includes a wide range of applications and protocols which make each implementation special (Oniga et al., 2020).

Figure 2.10: An illustration of the Internet of Things(IoT) (Patel et al., 2016)

Oniga et al. (2020) proposed an IoT Message-Based Communication (IMBC) protocol to achieve IoT network management transparency. This approach improved the earlier framework implemented with OMA Lightweight Machine to Machine (LwM2M) protocol specified by Alliance (n.d.). LwM2M coordinates and manages devices in an IoT network, including data transfer. Oniga et al. (2020) stated that the limitation of LwM2M in managing heterogeneous devices and messaging channels are in the use of Client End Point Names that enforces device identification on the network. IMBC advanced over LwM2M by replacing the need for IP or LwM2M object translators with the use of service bootstrapping which eliminated the need to predefine Client Endpoint Names. Multi-Channel messaging systems utilize a common agnostic data sharing payload service JSON for communication supported by IMBC. The message format is defined as a service represented in a fixed-length data format (Oniga et al., 2020). Fixed-length data formats are stored in hexadecimal numbers of 4-bytes described as a floating-point see Figure 2.11.

Service identifier	Payload data


```
{
"20": {
        "service": "temperature",
        "title": "Temperature",
        "description": "Temperature value in degrees Celsius",
        "type": "float",
        "lenght": "4",
        "unit": "°C",
    }
}
```

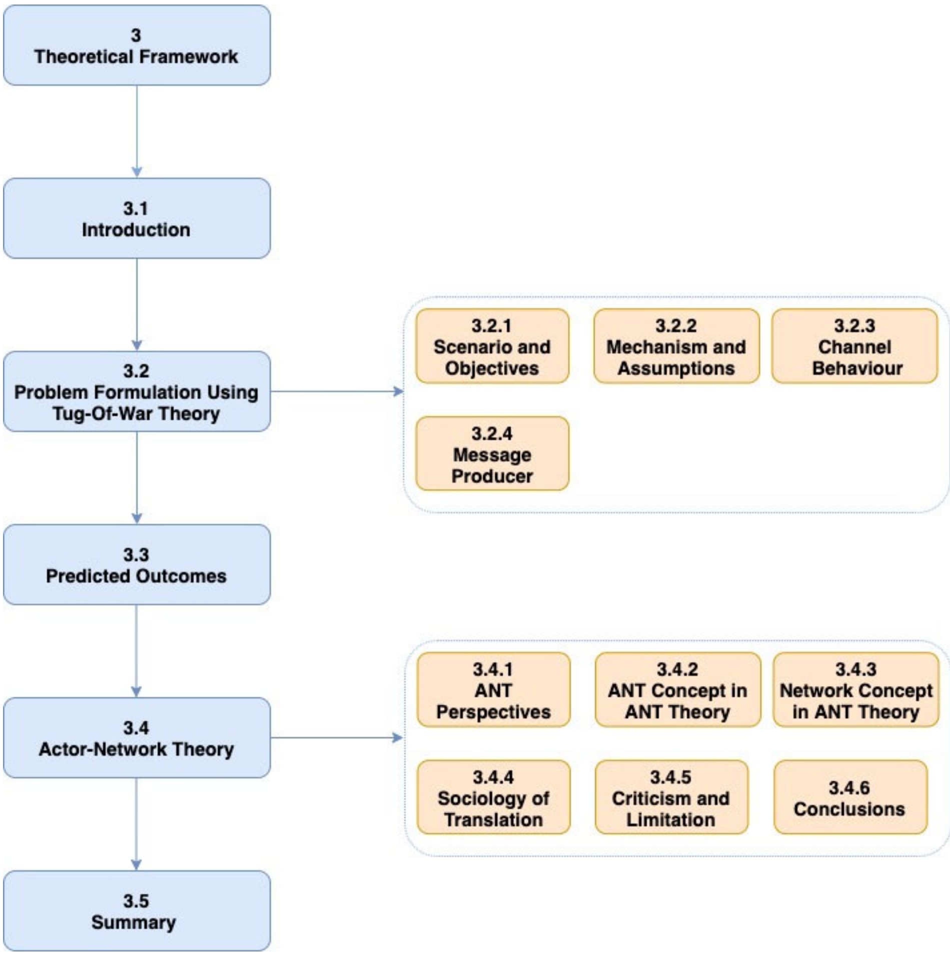

Figure 2.11: JSON Fixed Length Data Format (Oniga et al., 2020)

IoT MBC provides network management transparency, which is a critical element in the integration of disparate channels. Channel coordination and management provided by LwM2M offers a rich idea on how data transfer takes place between the message producers and message consumers or channels. However, it IoT MBC has limitations in managing heterogeneous channels. Overall, IoT MBC offers the ML MCM enabled systems to use fixed data formats in JSON structure so as to provide a standard agnostic data sharing payload between the heterogeneous channels used within the framework.

2.9 SUMMARY

This chapter provided an overview of machine learning to channel selection in a customer alert messaging system. The chapter also identified the requirements for implementing a machine learning-enabled MCM system. Furthermore, the chapter reviewed extant literature on Machine Learning algorithms suitable for channel selection and management. The challenges with existing MCM implementations were used as the foundational requirements for the proposed ML Enabled MCM system framework. Finally, after discussing the state of the application of IoT in the MCM system, various Machine Learning algorithms for Channel selection and management were explored. The next chapter presents the theoretical framework underpinning the research work.

3 CHAPTER 3 THEORETICAL FRAMEWORK



3.1 INTRODUCTION

This chapter discusses the theoretical framework that supports this research study. In academic research, theoretical framework plays three significant roles (Eisenhardt, 1989). First, it is a broad guide to the pattern and data collection approach. Second, being a share of an incremental collection of data and interpretation phase that ultimately leads to concept refinement, with all possibilities taken into account, and ultimately, to research the underlying MCM problem and propose an ML-enabled MCM system. ICT research studies have benefited from theoretical models, which have provided insightful information for potential information system design and development. The adoption of a machine learning (ML) customer alert system by Financial services institutions (FSIs) is both a social and organizational activity (Caflisch et al., 2020). Therefore, the purpose of this research is to make use of pre-existing theoretical frameworks that aid in understanding the constituent parts of multi-channel messaging systems, human activities, technological adoption and organizational deployment architecture of FSIs, and the adoption of the technology by the customers of FSIs. To this end, the Actor-Network (ANT) and Tug-of-War (TOW) theory are the frameworks used for this research analysis and the results presented later in this dissertation. The core principles of ANT and TOW are discussed in the subsequent sections. It should be highlighted that when these two theories are applied in this research, there are no conflicts or contradictions. This study does not seek to compare and contrast ANT and TOW, but rather to demonstrate the significance and complimentary nature of both theories for this research endeavour.

3.2 PROBLEM FORMULATION USING TUG-OF-WAR THEORY

3.2.1 Scenario And Objectives

As discussed in Section 2.5.6 the TOW model for channel selection and allocation of message resources is the simplest account of the interactions between the individual message producers (Customer Alert Messaging System) and channels available to send homogenous messages to the customer through the platform at an instance of time (Ma et al., 2019). The TOW channel selection mechanism is similar to the Multi-Armed Bandit (MAB) problem that relies on UCB values to determine the channel with the most rewards or availability. The mechanism also accounts for message producers that compete on channel availability to provide a homogenous message to a group of disparate message channels integrated into a single platform. For the types of computational channel resource allocation problems being investigated in this study, the messages are considered homogenous since the channels do not necessarily care about the producer of the message being sent to them. The messages sent from each message producer is translated using a data-agnostic layer, making them usable for each channel.

However, unlike other channel selection ML algorithms discussed in Section 2.5, this thesis focuses on achieving a particular set of outcomes for channel resources allocation in a given scenario. The approach selected in this study is informed by the dynamic channel selection and allocation mechanism which the system developer seeks to achieve. An artificial situation for message creation, translation and transmission is then created for the channel allocation scenario under the assumptions of the MAB framework.

First, consider a scenario comprising a set of message producing nodes in an MCM customer alert system, P. Each member produces varied messages ß, that can be translated and transmitted. The members of P are assumed to be operating under the MAB principle. Subsequently, consider a group of message consumers or channels, C, each member of which aims to consume and transmit the message ß, at an instant of time based on its availability. If p_i is a node in P, and c_j is a node in C, n_{ij} is used to determine the number of messages provided by p_i to c_j, the total number of ß provided at an instant of time also defined as the message *throughput,* t_{pi}, is defined as:

$$t_{pi} = \sum_{j=1}^{|C|} n_{ij} \qquad \textbf{Eq. 3.1}$$

The first objective to be realised from this setup is to be able to provide at an instance of time a balanced throughput within the system that ensures that each message producing node in P has an available channel C to handle and transmit the message from a group of disparate channels integrated for message transmission. In this case, a channel resource allocation and selection configuration given the provision of ß by the nodes in P at a given instant of time can be expressed using a vector T_s $<t_{s1}, t_{s2}, \ldots.., t_{sm}>$, where $m = |P|$. Measuring the throughput of the system is essential at this point, as well as the main objective of ensuring effective and efficient channel selection or allocation using the TOW approach in a fully decentralised fashion with no control from a central controller within the system with an ML method of the channel (Ma et al., 2019). Zhou et al. (2018) discussed the need for Reinforcement Learning which is essential to keep the state of success or failure of each channel's response. When a message is sent via the system by a message Producer P at each time t (Kim et al., 2016), +1 is added to throughput t_{pi} if the channel C can accept and transmit the message termed as a reward to the message producer. Otherwise, −ω is added to t_{pi} and thus defined in Eq. 3.2 and 3.3 respectively:

$$\textbf{Reward} = \sum_{j=1}^{|C|} n_{ij} + 1 \qquad \textbf{Eq. 3.2}$$

$$\textbf{Loss} = \sum_{j=1}^{|C|} n_{ij} - \omega \qquad \textbf{Eq. 3.3}$$

3.2.2 Mechanism and Assumptions

A TOW channel selection mechanism is used to decide how messages ß are routed between message producer nodes to messaging channel nodes in an ML-enabled MCM customer alert system. At an instant of time, a message producing node $p_i \in P$ via a broadcast mechanism sends out available messages. Each user making use of a resource, a channel, in this scenario, has the choice of accepting any of the message ß, only if they are not busy processing other messages in the queue, overloaded or completely unavailable (Oshima et al., 2020). The system iterates with channels that have shown their reliability over time using the previously-stored reward values (Zhou et al., 2018).

At this stage, the networked environment and other external factors are not considered because as it may complicate the understanding of the actual behaviour of the model under study. The following simplifying assumptions are made with this in mind (Zhou et al., 2018):

1. The system is initialised and operates synchronously in discrete time steps.
2. Each message producer produces one message ß per time-step.
3. The actual creation and production of ß may be regarded as instantaneous in that it does not unnecessarily interfere with the mechanism of the model.
4. Each message producer has the capacity to create messages ß and make the messages available to all the channels C available at each instance of time.
5. There is uniform network connectivity within the platform.

The first two assumptions are required at this abstract stage to understand and analyse the underlying system. However, their use has not been fully investigated. There does not appear to be any apparent reason for them to alter the underlying behaviour being demonstrated. Assumptions 3 and 4 represent the exploitation and exploration strategies explained in the related work by Gao and Pavel (2017), which are outlined in Section 0. Finally, the assumption 5 model's network conditions are described by Wolski et al. (2001) in the G-Commerce system.

At each time step, each channel, if available to receive messages, may receive any number of messages ß from any number or message producers in P, however, this is subject to the constraint that the total messages produced and transmitted per time-step are exactly one message (as per assumption 2). If no messages are produced by P, the channels may instead be in a waiting state with no messages to process and in a ready or waiting state. These constraints mean, therefore that $\sum_{i=1}^{|P|} n_{ij} \in \{+1|-w\}$ applies to all $c_j \in C$.

3.2.3 Channel Behaviour

Both the message producers and channels accrue a reward or loss for the interactions within the MCM system. This is deemed to be the value they associate with delivering messages sent to them successfully for channels. From a channel's perspective, if a message producer transmits a message that would not lead to failure in delivery at that instant of time, then the operation is described as *acceptable*. P_{cj} is used to denote the subset of P, which contains exactly those message producers whose messages were *acceptable* by channel C_j. Channels will not receive messages when they are not available because they are busy or out of service. The possibility of complex channel selection strategies means that there will not be a direct mapping between message producers and channels at any time (Berthold et al., 2008). Berthold et al. (2008) described the channel behaviour scenario by using the Markov Decision Process in which channels are represented as factors in space with reward functions. Channel states are determined from previous interactions within the system in line with the MAB problem (Zhou et al., 2018).

3.2.4 Message producer Behaviour

Message producers receive a reward based on the response received from a channel that delivers the message successfully. The reward of each message producer P_i is defined as R_{pi}. In its simplest form, the reward from the transmission of message ß is described as:

$$R_{pi} = \sum_{j=1}^{|C|} t_{pi}^{ß}\ n_{ij}, \qquad \textbf{Eq. 3.4}$$

Or alternatively as:

$$R_{pi} = t_{pi}^{ß} \text{ X } t_{pi} \qquad \textbf{Eq. 3.5}$$

This shows clearly that a message producer that wishes to maximize its reward would aim to quickly check the status of the channel and its previous successful delivery history and promptly decides to either engage and not to wait endlessly for an unresponsive channel. As seen from the channel’s behaviour, throughput for message producers is determined mainly by the channel’s performance. The section that follows investigates the trade-off of the exploration and exploitation strategy by message producers.

3.3 PREDICTED OUTCOMES

One of the motivation factors for employing the exploration and exploitation strategy TOW model is that it drives the ML-enabled MCM system towards equilibrium. It is at this stage of equilibrium that the MCM system is stable in the long term. This steady state, which ensures that message producers can connect and transmits messages optimally to the available channels, is referred to in this thesis as the outcome of channel resource allocation and selection. However, in realistic cases, abrupt changes to the system such as the addition of new channels or integrations may result in instability or a restart of the learning process; this hot addition of channels scenario is not considered in this study.

The model described in this chapter is a generalised version of the TOW model (Kim et al., 2016). As discussed in Chapter 2, the classic TOW modelled around the MAB problem consists of a player and a machine. Each machine has two slots A and B, and a player has an opportunity to play any of the slots at a time. A reward is added to a player by incrementing the estimator with +1, while a sort of loss expressed is as -w and the estimator is defined as Q_a. In contrast to other methods of calculating the probability of winning at a slot machine, the model uses the TOW learning approach with the volume conservation principle to update all estimators similar to the concepts of displacements arms in imaginary volume-conserving objects, and this value will either appreciate or depreciate with rewards or losses, respectively (Oshima et al., 2020).

This thesis uses the channel selection technique modelled on the MAB problem. As stated in the earlier section, TOW algorithm was used to provide a solution to the MAB problem. Figure 3.1 depicts the concept of the channel selection strategy. The message producers, capable of accessing the various performance of the channel cj, where j is the selected channel. c_j can be any indexes of the performance of the individual messaging channel c_j $(j \in \{1,2 \dots . n\})$.

Based on the TOW algorithm, the channel selection process determines variables such as throughput, delay, or other metrics of each channel. The TOW algorithm then assesses whether the reward or the loss is to be added by evaluating the obtained performance of channel j. If there is a reward in a transmission by a message producer, the TOW algorithm updates the estimator as $Q_i + 1$; otherwise, it updates the estimator as $Q_i - w$. The algorithm in the ML-enabled multi-channel system is described as follows:

1. Initiates monitoring of the performance of each channel in the MCM system c_j.
2. Updates Q_i, T_{pi}, and all Rewards and Losses based on the observation of the channels c_j as stated in Eq. 3.2 and 3.3.
3. Selects a channel c_j with greatest T_{pi}.
4. Observes the selected channel's performance and determine whether a loss or reward is to be awarded to it.
5. Go back to step 2 and continue processing.

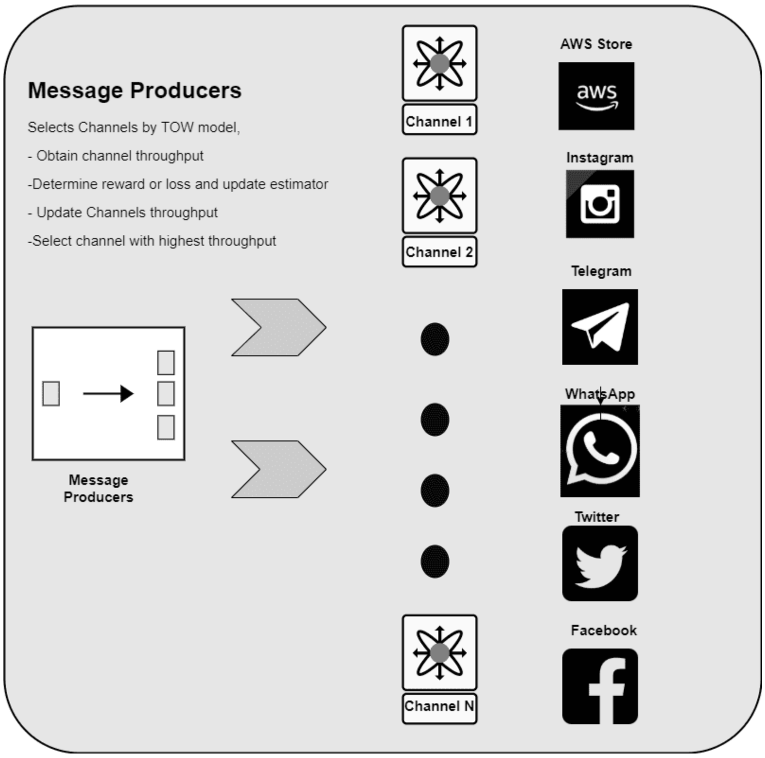


Figure 3.1: Machine Learning Enabled Channel Selection MCM framework based on MAB algorithm: Source (Kim et al., 2016)

Overall, the channel selection and assignment challenge were modelled as a MAB problem to offer a solution that is both effective and efficient in channel allocation using ML techniques.

3.4 ACTOR-NETWORK THEORY

The use of the TOW model to understand how a consumer alert system can be implemented using an ML framework (i.e., the channel selection module) provides a solid foundation to satisfying the research questions raised in this research work. ANT is examined in this section as a prospective supplemental theory employed to further analyse the case under investigation.

One of the key reasons for selecting ANT as an analysis tool in this research study is that it provides new insights into ML and channel selection dynamics. The ANT approach made it possible to study mobile devices, enterprise service bus, agnostic data formats, and impact on the network and their interactions. Additionally, this research indicates that deploying technology systems has immense technical and social benefits. To get a better understanding of the elements needed to put the ML-enabled framework in place, there is a need to bridge the gap between societal needs and technology. The success or failure of IS implementation is not driven by inherent qualities of the technology or social contexts (such as other actors and user behaviours), as is the case with conventional theories. Instead, the relationships that exist and are generated between the technology and IS deployment's success or failure is determined on the factors surrounding it; most specifically, the technical and social actors. ANT focuses on associations rather than individual properties. It is worth noting that ANT has its language and terminology, which may appear to be comparable to those of other theories such as systems theory at first glance; however, on closer inspection the language and terminology in ANT have entirely different interpretations. For example, ANT theory has its own "definition" of concepts such as, "*Blackbox*", "*Channel*" and "Translation of *data*".

ANT originates from the studies of science, technology and society (Callon, 1986, 1991; Callon & Latour, 1981; Latour, 1987) to investigate the role and complexity of various actors within the system. Latour (1987) argues that science and technology should be examined in action and that the focus should be on the dynamics of their interaction rather than the strength of their connection. As a result, the author introduced As a method of analysis, ANT provides the a formidable framework for studying relationships between the network entities (Latour, 1987). In IS research, ANT has been utilized to evaluate the failure and success of technological innovation within organizations. Subsequently, researchers have proved the efficacy of ANT in IS research, including the Electricité de France's creation and deployment of an electric vehicle (Callon, 1986). The ANT theory was utilised in organizational analysis (McLean & Hassard, 2004) and accounting systems (Lowe, 2001), as well as in the design of Aramis, a ground-breaking and innovative method of public transportation (Latour, 1996). Using ANT principles the effect of standards on Electronic Data Interchange systems (EDI) was reviewed by Monteiro and Hanseth (1996). Similarly, research on how networks (that makes use of channels) emerge and become stable may lead to the enhancement of channel selection logic with machine learning.

3.4.1 Actor-Network Theory Perspectives

ANT consists of a group of different actors linked by common interests. The ANT method represents an excellent theory in science sociology that aims to explain and interpret technological and social changes (Tatnall & Gilding, 1999). ANT theorises and elucidates more than just the non-technical and technical factors; it symbiotically describes actor relationship between human and non-human entities (Callon, 1991).

ANT's focal point shows the perspective of the social world and the relationship in an heterogenous network between objects and people and the process translation and service negotiation between them (Callon, 1986). These heterogeneous networks are similar to the disparate channel integration and selection in the MCM system which is at the heart of this study. ANT theory focuses on heterogeneous networks and drills down into the impact of identity within the system (Latour, 1996). Latour (1996) asserts that several organizational concepts working concurrently together demonstrates heterogeneity. The combination allows for a balance of specified and recognised distinct interests and values to be pursued (Latour, 1996).

Monteiro (2000) argues that ANT distinguishes clearly between social reductionism and technological determinism. The ANT theory disagrees with the idea that technology drives social development, which maintains that both technical and social developments drive social progress. ANT theory classifies both human and non-human entities within a heterogenous framework in the same way. To illustrate this with a concrete example, consider how our daily interactions are impacted by a variety of elements, such as political, social, historical, and technological factors. For example, when a transaction is conducted at the bank, and the customer receives a message alert, the customer is affected by the SMS mobile application for SMS and GSM network (Monteiro, 2000). To understand the phenomena of utilising a telephone messaging application, several elements must be evaluated simultaneously.

Law (1999) discusses the three methodological principles used to equitably describe actors (human and non-human) within a heterogeneous network critical to implementing a heterogenous multi-channel messaging system. The relationship between ANT and principles of free association, agnosticism and generalized symmetry are illustrated in *Figure 3.2*.

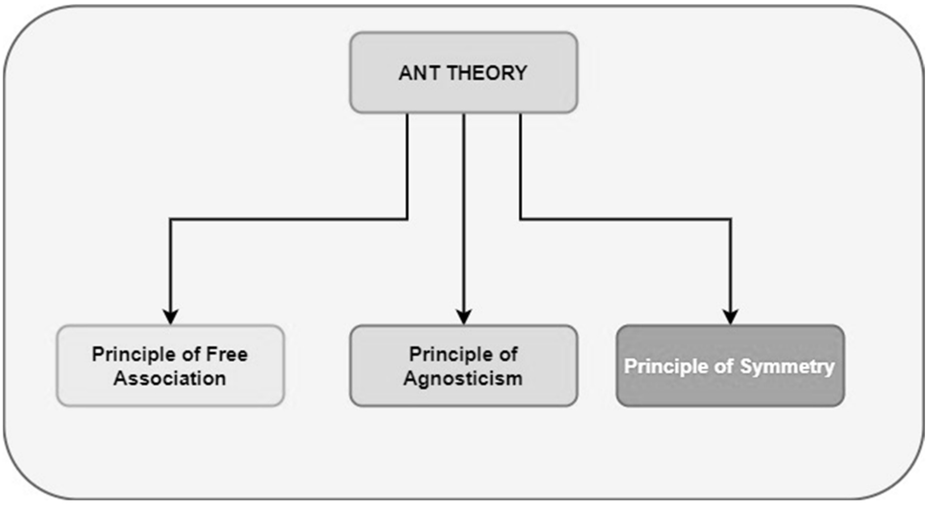


Figure 3.2: ANT Theory and related principles (Law, 1992)

These three methodological principles are explained below as described by Law (1992):

- **The Principle of Free Association**: Based on this principle the researcher should not dwell on the previously documented differences between the natural and social networks because there is no delineation between the two. However, it is possible in future for the natural and social networks to be separated. In the case of heterogenous multi-channel systems using ML for channel selection, message producers can share and exchange data to provide rich messaging templates which the channels can use. The concept of free association makes it possible to add new message producers with or without dependence on the other to ensure the system's integrity at all times.

- **The Principle of Agnosticism**: According to this principle, the researcher should avoid criticising or censoring any actor human or otherwise. According to the ANT principle, neutrality must be maintained by agents by both non-human or human actors. This ANT principle is critical to the ML-enabled MCM system because it allows the researcher to fully consider the implementation of a standard data-agnostic layer that is reusable by all message producers to ensure that the messages are usable by the disparate channels. This principle was considered during the implementation and evaluation of the proposed model.

- **Principle of Symmetry**: This ANT principle requires researchers to use abstract and neutral terminology to clarify the contradictory opinions of all participants in the same terms. This rule prohibits changing domains when moving from the technical to the social aspects of the problem at hand. It denies giving either technology or social actors any favourable analytical position. Furthermore, this concept allows the delineation of the disparities between the features of each message producer in the MCM system as well as the heterogeneous channels implemented. All the entities within the ML enable systems are actors playing their part to complement the system's overall effectiveness in a symbiotic and symmetric way.

In addition Callon (1991, p.61) asserts that: "*ANT was established to examine circumstances in which it is difficult to differentiate between human and non-human actors, as well as scenarios in which actors have variety of forms and levels of competency*." The way in which actors (human and non-human) are handled is well suited for the development and advancement of information technology systems i.e., understanding the problem of "***How machine learning algorithms can be used to enhance the channel selection logic of a multi-channel messaging system***" used by FSIs to provide cutting edge services to their customers.

Like the TOW theory, ANT identifies the collaboration between the organization and the actors. ANT, on the other hand, lays equal attention on actors that are non-human in nature (i.e., the software related artefacts such as message producers, message translators and messaging channels integrated within the system) and further describes the interaction between the human or individuals in the organization and the non-human actors to provide concrete support to the network through shared interests.

3.4.2 Actor Concept in ANT Theory

Langlais (2006) argues that actor-network is created by the relationships of the entities which form the network, namely attachments and actors. Langlais (2006) explains that actors are usually added to a significant network of attachments. An actor can be defined as an object that can act or has been granted to perform an activity by other actors. Any item that can originate from the source of action is an actor (Latour, 1996). Law (1992) described actors using effects they create within a heterogenous network and their interactions as in the case of the ML-enabled MCM system. These effects created by actors produces social agents aware of their environments and can relate based on other actors' reactions within the network. The concept of black-boxing describes each actor within an actor-network as a grouping of other actors or a part of an actor network themselves. Hence, each message producer, message translator, message router and messaging channel form a mesh of networks to produce the desired result of transmitting messages to the customers of the FSIs.

Monteiro (2000) explains that an actor is usually a network in infinity. Hence a researcher needs to drill down and focus on studying the network under investigation to understand the actors involved and the relationship between them; for example, the relationships between the message producers and heterogeneous channels. That being the case, Law (1992) argues that human-related activities like acting, reading, working, eating are embedded in a network beyond the human body itself. According to Monteiro, (2000) in identifying the size or form of an actor, the researcher conducting the study is limited based on the practises of each actor in the network. Based on ANT terminology, an actor is a product of complex relationships between artefacts and humans, and an actor is always a network (Law ,1992). Understanding the inter-relationship between the various actors in the ML-enabled MCM system enabled the researcher to implement an effective and efficient channel selection strategy that is critical for the system's operation.

3.4.3 Network Concept in ANT Theory

Actor-Networks are created by the interaction between the human and non-human participants in a network (McLean & Hassard, 2004). Translation of data from one form to the other is critical in implementing an ML-enabled MCM system. Message producers create messages that require transmission to the customers, hence the need for a standard data-agnostic layer that allows translation from one part of the system to the other to allow reusability. The translated messages are relayed to the selected channels for transmission to the customers. These interactions are enabled via the conversion of preferences and the network's addition of new actors. Association between disparate actors with initial non-aligned interests is termed translation of interests. Latour (2013) described the variables of these types of networks as being *“immutable mobile*”. Immutable mobile means that when these variables are randomly changed or rearranged in time and space, they remain steady and unaffected (Latour, 2013; Tatnall & Gilding, 1999). Let us consider, an immutable device such as a smart television when it shows assertive interrelated elements (such as near field communication support, internet connectivity, remote desktop functionality, video streaming application, and Bluetooth). The Irreversibility properties displayed by these networks allow them to endure through space and time without the need to change these properties (Walsham, 1997).

Network instability is a critical concept in ANT, including “time” on the network once formed (Law, 1992). Network instability can be due to the addition of alteration of alliances, introduction of new actors and the exit of actors within the existing network (Callon, 1986). Recursively, a network develops and reproduces itself. The effect of an addition or removal of a new message producer or channels needs to be measured, and it is critical to the overall effective ness of the system. The channel stability feature of ANTs allows for an in-depth review of this challenge. The term "network" does not refer to a static entity but rather to a dynamic alliance of players. Network durability in ANT is assured at each intersection that represents the convergence of multiple networks. Latour (1987) clarifies further that the actor networks are created and replicated continually and do not define the technical and social nature of networks; instead, the submission is on the weaker or stronger nature of the network association produced.

Power is an essential aspect of ANT. Power is created within the network when created or reproduced between the actors (Latour, 1990). ML enables learning and relearning with the training of nodes. Power is established when a message producer is locked continually to a previously selected channel for message delivery. The aspect of power sharing by actors is further explained in the next section with the concept of sociology of translation.

3.4.4 Sociology of Translation

The process of creating an actor-network is referred to as translation which is established when creating the ML-enabled MCM system. Translation is an essential concept in ANT, and it occurs between message producers and messaging channels during the design and development phase of the ML-enabled MCM system. Invariably, ANT also connotes “the sociology of translation”. Latour (1990) assets that the social and technical links between humans and other species are inextricably linked, hence there is no social assessment of existence. Actor-networks are built on relationships between human and non-human actors. The manifestations between these actor-networks are primarily called “translations” (Latour, 1987, 1990). Actors interact based on object infusion controlled by a program or controller script (Arkich, 1992). In the case of an ML-enabled system, the program is modelled as a MAB problem using the TOW algorithm. There are 4 stages in translation (Callon, 1991), namely the Problematisation stage, the Interessement stage, the Enrolment stage and the Mobilisation stage (Figure 3.3). A cycle of repetition of these stages is the case of failure at a stage.

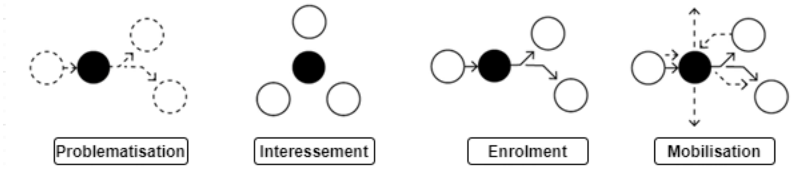


Figure 3.3 Translation Stages (Callon, 1991)

The problematisation Stage defines key actors within the ML-enabled MCM system, it searches for other actors or message producers or channels with attributes similar to theirs and establishes identities based on interest using the channel selection algorithm implemented. These major actors define the challenges and solutions within the MCM system and assign roles and identities to other network actors. Interessement is the second stage of translation. Other message producers seek to convince others that the interests created by key message producers are genuine and aligned with their interests to form new alliances. This collaboration enables message producers to share data and align their interests using a common data-agnostic data layer for message translation and transmission.

The third stage, defined as Enrolment, comprises assigning unique responsibilities to each message producer in the actor-network (MCM system). Key actors within the system utilise various negotiation strategies to convince other message producers to align in the newly created network and perform their roles effectively. Mobilisation is the last stage of translation that ensures that key message producers deploy different strategies to ensure that aligned message producers can communicate and work harmoniously and in agreement within the ML-enabled MCM system. Stability within an actor-network is achieved when ally message producers are mobilized successfully (Callon, 1991).

3.4.5 Criticisms and Limitations of Actor-Network Theory

The ANT theory has faced criticism over its narrow focus on local actors and neglect for macro-social structures by not creating a visible relationship between the global and local actors, in the case of this study, message producers external to the system. Walsham (1997) identified about 4 criticisms focused on ANT in research studies. These limitations of the ANT theory include its symmetrical classification of actors (human and non-human), which is, the description of power rather than its application. However, proponents of the ANT theory, both at macro and micro stages, that may be evaluated and synchronised using the same technical tools (Latour, 1990). Walsham (1997) defines structuration theory as the levels of analysis that are directly related between micro and macro actors and therefore provides a model of the structure and social action to solve these problems. Latour (2013) opined that the variations between the actors and network are two sides of the same subject.

As far as Callon & Latour (1981) are concerned, utilising the same analysis framework for micro and macro-actors is appropriate. Such a strategy guarantees the scalability of the actor-network paradigm. This inherently means that a single actor network element may be expanded into a new full actor network and vice-versa, subsequently the entire actor network may be transformed to become a single element within the network or the other (Monteiro, 2000). Boucaut (2001) argues that the claims that either micro or macro actors have priority over each other are misleading. Other critics also claimed that the symmetric nature in which ANT theory threatens actors by placing human and non-human actors on the same level is problematic, and thus contend that all actors are not equal and others have a more substantial influence than others (Mutch, 2002). ANT theory neglects emotions and therefore insinuates that human actors play a significant role when interacting with technology (Mutch, 2002). Knights and Murray (1995) argue and criticized the ANT theory owing to neglect for global power influence and the effects power has on the formation of the network. Latour (2013) disagrees with this argument claiming that theorists who criticize ANT focus solely on the disparities of social interactions in the network. The differences between the social and technical effects may be considered separately when focusing on both to provide outcomes. Either machines or humans can be affected by social relations and vice-versa (Law, 1999). As documented by Walsham (1997) arguments about political and moral issues should be supported by strong empirical data, while nothing that ANT contributes to both.

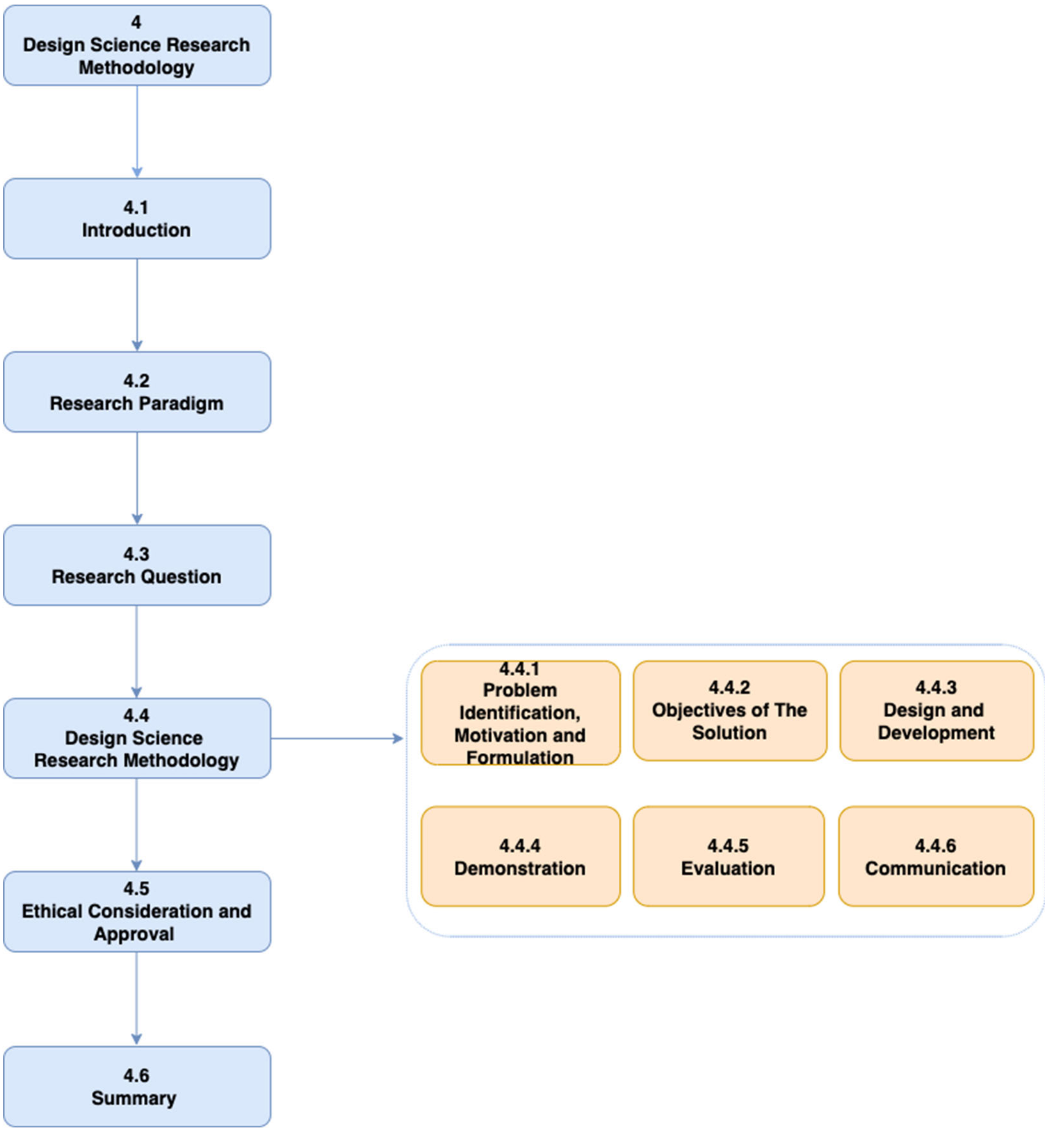
3.4.6 Conclusions of Actor-Network Theory

ANT theory provides a platform for case studies and artefact implementation. It also offers a more varied and profound analytical tool for thinking and designing new solutions while understanding the relationship between the inter-connected parties. An ML-enabled multi-channel customer alert system cannot operate in isolation; instead, effective and efficient collaboration is needed between the channels and the subsystems within and outside the system’s control. ANT thrives on the relationship between disparate actors (channels, message producers, message translators) and ensures coordination (Tatnall, 2014). ANT provides a platform to explore system complexities at the heart of information systems research. The contradictions noted by critics of ANT have been resolved by understanding that both a network's technical and social components are directly related (Langlais, 2006). Tatnall and Gilding (1999) concluded that the ANT is useful in IS research that requires the interactions between the social, technological, and political actors that are essentially important and in the business use of technology. This business use of technology will allow FSIs to benefit from using the ML-enabled customer alert systems.

3.5 SUMMARY

This chapter started by discussing the TOW model and MAB problem for adoption and implementation in this study. The choice of the TOW model approach is due to its novel approach to channel selection and management using the MAB approach. ANT’s examination of the relationships between human and non-humans and establishing the relationship between technology and individuals gave it a more robust platform that supplements the TOW’s approach. ANT’s tenets sit solely in the centre of the interaction between the individuals and technology platforms. The ANT theory was chosen to complement the study because ANT supports identification of relationships between each entity within the ML-enabled MCM system. This complementary nature of ANT and TOW theories offers a complete analysis of how FSIs can implement IT artefacts, development and deployment strategies. These two separate theoretical views enabled this study to understand the technological and social context of the implementation of an ML-enabled customer alert system. The next chapter discusses the research methodology employed in this study.

4 CHAPTER 4 DESIGN SCIENCE RESEARCH METHODOLOGY



4.1 INTRODUCTION

In 2, a review was conducted to understand the requirements of a machine learning (ML) enabled Multi-Channel Messaging (MCM) system and its use in customer alert system used by Financial Service Institutions (FSIs). The literature review was conducted methodically to understand the various elements necessary to create a preliminary design of the ML-enabled MCM model from the existing body of work with enhancements in the channel selection module. As attested by Avison et al. (1999), research is conducted by following different research methods that are based on the research context, questions and objectives. Choosing a research methodology is therefore dependent on its suitability of the research problem under investigation. Therefore, this chapter presents the argument for the philosophical approach underpinning the study of how an integrated MCM can be implemented using ML algorithms to enable effective and efficient dynamic channel selection and integration methods in the context of a customer alert system used by FSIs. Furthermore, an explanation of the research questions, design and strategies and evaluation method to be utilised was discussed. The chapter concludes with the ethical approval obtained prior to conducting the study from the University overseeing this research. In this chapter, the motivation and for the suggested technique were examined in detail. The next section describes various research paradigms, including the one adopted in this research study.

4.2 RESEARCH PARADIGM

The choice of a research paradigm to use for research is a critical step within the research stage. Oates (2006) is of the view that various methodologies can be applied to investigate the problem under investigation. Ontology, epistemology, axiology, and methodology are the dimensions that distinguish each research paradigm. These paradigmatic aspects have an impact on how research is conducted, and knowledge is constructed. Investigating these research paradigms is necessary for determining a good paradigm in which to conduct this research. As far as Wilson (2014) is concerned, ontology focuses on acceptable knowledge while focusing on the nature of reality and how the researcher thinks about the world. Epistemology connotes the nature of knowledge and how the researcher conceives the surrounding environment for the research. Epistemology describes the nature of knowledge within a specific paradigm (Wilson, 2014). Bell and Bryman (2007) contend that, for the research in a given paradigm to be credible, it should be conducted in accordance with what that paradigm accepts as the norm. Axiology is known as the study of nature of value. The focus of axiology is primarily on the role that the researcher's perception plays in the study (Wilson, 2014). Research approach and strategy are referred to as methodology. In general, methodology is concerned with the entire strategy for conducting research (Bell & Bryman, 2007). This involves all aspects of the investigation, from data gathering and analysis to theoretical application. When deciding which paradigm to use to conduct research, it is imperative to consider these dimensions and select a paradigm that is most appropriate for the study.

Interpretivist and positivist research, qualitative and quantitative research, critical realism, pragmatist, and confirmatory and exploratory research are some of the dichotomous paradigms and views that have underpinned research methodologies in science (Creswell et al., 2007). A paradigm serves the following multiple purposes as stated by Creswell (2009):

1. Creates standards for research tools such as instruments, technique, and data collection.
2. Produces models that enable researchers to address challenges.
3. Provides procedures, philosophies, and strategies to consider when similar research challenges arise in the future and
4. Guides professionals by identifying key issues that are applicable to any field of study.

Myers (1997) explored the following philosophical views: that the basis of positivist research is that a researcher uses techniques that are already available for testing. Myers (1997) asserts that the positivist research technique features quantifiable characteristics describing the phenomena and genuineness of the problem under investigation irrespective of the observer(s) and the strategies employed. This basically indicates that the researcher is impartial toward the study hypotheses and the phenomena (Orlikowski & Baroudi, 1991). Positivists proclaim that knowledge can only be true, false, or meaningless. Positivists adhere to the traditional research system and is based on the concept that every effect has a cause that must be determined (Creswell, 2003). Positivist approaches are quantitative in nature where scientific enquiry is important and observation is through the search for empirical evidence (Gray, 2014). Gaaloul and Molnar (2014) opines that research needs to validate the results generated from experiments while ensuring that the results produced align with sufficiently verifiable literature.

Interpretivism research approach is a philosophical approach modelled primarily in the assumption that gives no preference to predetermined variables thereby compelling the researcher to study and understand the context of the research. Interpretive research creates a method to understand the context of the research from a specific perspective and is motivated by the context of the research (Myers, 1997). Interpretivism emphasises the impact of social and cultural elements of an individual. In the context of the socio-cultural background, this viewpoint focuses on people's thoughts and ideas. The interpretivist paradigm requires the researcher to take an active involvement in the study since a comprehensive perspective of the participants and their behaviours, thoughts, and meanings is required.

People see critical realism research as a form of social criticism in which the limiting and isolating conditions are examined (Klein & Myers, 1999). Critical realism research is a philosophical perspective that holds that independence can only be accomplished by addressing the historical, cultural, socioeconomic, and political reasons of social inequities and injustices that people have created or perpetuate (Myers, 1997).

Pragmatism emphasises the importance of exploring phenomena with the most effective methods (Creswell, 2009). The basic objective of pragmatism is to conduct research from a practical standpoint, in which knowledge is continually questioned and understood rather than constant. Consequently, pragmatism involves some researcher interaction and subjectivity, especially when generating conclusions based on participant responses and decisions (Myers, 1997); hence, pragmatism is not bound or limited by any one ideology. The scientific philosophy known as pragmatism holds that ideas are only worth considering if they can be put into practise. Pragmatists recognise that there are many potential interpretations of the world and research methods, that no single perspective can ever offer a whole picture, and that there may be multiple realities (Saunders et al., 2007). The various research philosophies are summarised in Table 4.1.

Table 4.1: Research Philosophy adapted from Wilson (2014)

Paradigm	Research Approach	Ontology	Axiology	Research Strategy
Positivism	Deductive	Objective	Value-free	Quantitative
Interpretivism	Inductive	Subjective	Biased	Qualitative
Pragmatism	Deductive or Inductive	Objective or Subjective	Value-free or Biased	Quantitative and or Qualitative

This study inclines to the pragmatism philosophical approach due to it support for the use of experiments. Pragmatism has been adopted for this research work based on the reasons listed below:

- The aim of this study is to understand how an ML framework can be used to enhance MCM platform used by FSIs for the seamless integration of heterogeneous channels of communication (e.g., SMS, VoIP, IM, USSD, App-based, etc.) to the benefit of their customers. This developed framework and the artefact usefulness was simulated in line with the experimental approach that the pragmatist approach supports.

- There are no reasons to support or refute any hypothesis. In line with pragmatism, there is no ontological distinction between facts and values or between practical and theoretical reasoning (Oates, 2006).
- Support for multiple research methodologies (Qualitative and Design Science Research methodology discussed in subsequent sections) (Saunders et al., 2012) see Table 4.1

The methodologies selected have been chosen based on their ability to provide optimal solutions to the challenges raised. This study made use of the mixed research methods, these includes the process based, pragmatist and design science research methodology discussed in Section 4.4

4.3 RESEARCH QUESTIONS

The primary goal of this research is to determine how FSIs may apply current breakthroughs in machine learning (ML) and artificial intelligence (AI) to their multi-channel customer alert system to increase the coverage of the services provided by FSI's. Roode (1993) asserts that researchers must keep in mind the inherent social aspect of computing and solution delivery when developing research topics. The research questions asked in this research study allowed the researcher to elicit requirements for the ML-enabled MCM system within the scope of these questions while leveraging on the actor-network theory (ANT) to identify all elements within the ecosystem. This is because computer science is a field of study in which researchers studies technology, and information systems and their integration within organizations to derive technological benefits (Roode,1993). A problem statement, which is usually represented as a question or questions, is always the starting point for a research endeavour. Using the process-based research framework, the researcher would have to ask many research questions to investigate various parts of the problem statement. The research questions may not be related in the same way, and the uniqueness of each issued statement will determine which questions are pertinent and in what order they should be asked. The research questions for the problem statement of the research study were chosen from the list of four generic research questions illustrated in Figure 4.1 (Roode,1993).

	What is?	
How does?	Research study problem statement Teaching situation Information System development	Why is?
	How should?	

Figure 4.1: Research Framework (Process-based) Source: Roode (1993).

In this research work, the research questions including 4 subsidiary questions were formulated:

"How can integrated multi-channel messaging be implemented using machine learning algorithms to enable effective and efficient dynamic channel selection and integration methods?"

The following sub-questions listed below were derived from the main research question:

a) *What are the general requirements of a machine learning-enabled multi-channel messaging platform?*
b) *What machine learning algorithms can be used to seamlessly determine the effective channel or s to use for message delivery on an MCM platform?*
c) *What problems will an efficient dynamic channel selection machine learning MCM platform solve for Financial Services Institutions using MCM for Customer Alert Systems?*
d) *How can MCM platforms be enhanced with machine learning algorithms to enable dynamic and learning capacity of the channel selection module?*

The study used a pragmatist approach to analyse the interaction between actors (i.e., ANT) in the design and development of the IT artefact for an ML-enabled MCM system using Roode's (1993) approach, namely: "How should? And What is?" within the contexts of the research questions. The study also highlights the challenges in linking IT business strategy, the implementation of ML-enabled multi-channel customer alert system by FSIs and the transformational impact on the organization that adopts it. As a result, this study provides a deeper understanding of how FSIs may leverage ML and AI technologies to enhance the delivery and accessibility of sophisticated financial messaging platforms to their customers.

In summary, the research questions were evaluated from two different but related levels i.e. (macro and micro) (Boucaut, 2001). On one hand, the macro level was applied to analyse and address how a well formulated IT strategy (ANT approach) that leads to technology adoption (such as ML, AI enabled channel selection) was used to enhance service within the organization that deploys it. On the other hand, the micro-level was utilized to evaluate the IT strategy implementation using the organizational standpoint.

4.4 DESIGN SCIENCE RESEARCH METHODOLOGY

Methodologies are used to organise a study by laying out the steps, activities, and tools to be used for research, and to provide a way of evaluating that research. Because DSRM is the overarching paradigm in this thesis, a methodology based on this paradigm is required. Design Science Research (DSR) methodology mainly focuses on the identification of a specific problem in an organization environment. Sein et al. (2011) assets that DSR methodology is used to build and evaluate prescriptive design knowledge in an organizational environment while addressing two apparently diverse challenges. This is done by responding to a problematic situation in a specific organizational environment through intervention and evaluation of the problem and building an information technology (IT) artefact that provides solution to the class of problems within an organization.

This research focused on understanding the current limitations of multi-channel messaging implementations, which currently use pre-selected channels for message delivery without ML or AI in the channel selection logic. Since ML-enabled multi-channel message delivery system is, in principle, an organizational issue, the DSR methodology was chosen to provide an approach to deliver a solution to the limitations of base MCM systems. The requirements for the solution was derived solely from the pre-selected messaging channels problem in the existing MCM for transmitting messages to FSI customers.

The DSR methodology see Figure 4.2 effectively handles disparate problems that are encountered within an organization and ensures that an effective solution is proffered to address the issue. The main aim of the DSR methodology is to ensure the implementation of an artefact that provides a solution for an organization based on the design activities. Existing knowledge repositories provide a guide for the research and establishes the critical foundation and processes for the conducting the research (i.e., relevance, design and rigor cycle) (Adikari et al., 2009).

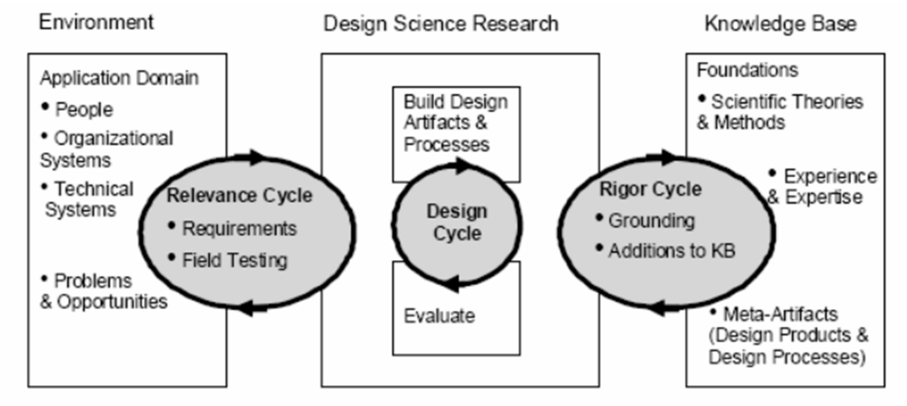


Figure 4.2: DSR Cycles Source (Adikari et al., 2009)

This study applied the Design Science Research (DSR) methodology as recommended by Hevner et al. (2004). DSR stipulates methods that enable the building of innovative IT artefacts within the organizational setting while defining the procedures for the build, test and run phases of solution design and development (Oates 2006). The requirements can be categorised based on a number of factors. The untested features or conditions for the proposed system are collectively referred to as the raw requirements. The most significant aspect of this phase is determining the goals that must be met. Security, performance, security , and reliability are all key examples of non-functional requirements. These non-functional needs are frequently important in system evaluation. According to Peffers et al. (2007) the DSR Methodology consist of the stages listed below:

- Stage (i) *Identify Problem and Motivate*: this includes the identification of a research problem. Before moving on to the next stage of the DSR technique, the problem that motivated the study must be methodically researched and analysed. This step includes articulating the criteria used to review the artefact developed.
- Stage (ii) *Define Objectives of the Solution*: this step defines the objectives of the solution explicitly, while ensuring that there are no misconceptions about problems the artefact solves when developed.
- Stage (iii) *Design and Development of Artefact*: this stage involves the design and implementation of the proposed artefact.
- Stage (iv) *Demonstration:* this involves showcasing the developed artefact to establish if the design requirements are met.

- Stage (v) *Evaluation*: the artefact's use is assessed based on its performance after the demonstration stage. Evaluation can be conducted using various methods which include quantitative, use-case analysis and prototyping of the artefact.
- Stage (vi) *Communication*: this entails the use of suitable channels to publicise the developed artefact at various forums or channels such as journals, book chapters, and conferences by showcasing the design process, review and evaluation process and, most importantly, seeking feedback from the audience on how to improve the artefact.

This study focuses on the problem-centred initiation approach to solve the problem detailed in subsequent sections to build the design for the ML-enabled MCM system. Similar methods supported by DSR include objective centred solution, design and development plan and observing a solution approach. DSR is different from other traditional methods of research methodology due its support for learning while building an artefact (Alturki et al., 2013). The problem-centred approach was explored in this study due to the need to be able to elicit the requirements for the implementation and design of the dynamic ML-enabled MCM system for FSIs. The main stages of DSR are presented diagrammatically in Figure 4.3 (Peffers et al., 2007)

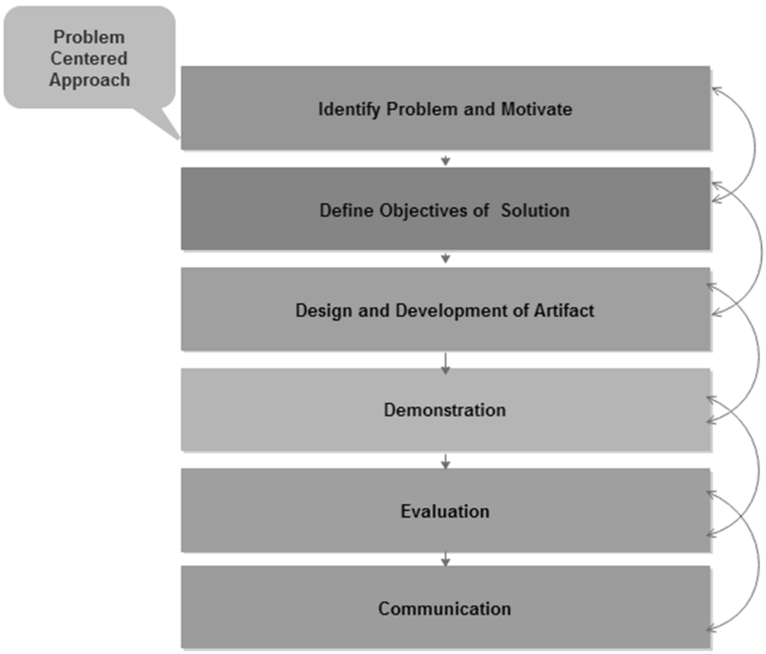


Figure 4.3: Design Science Research Stages (Peffers et al., 2007)

4.4.1 Problem identification, motivation and formulation

The problem identification stage of DSR stage is an important stage required to be used for dimensioning the scope of the work in order to proffer a suitable solution to the research problem. This problem limited customers to the use of a pre-selected channel for message delivery in the customer alert systems used by FSIs. Further interests were also driven by the limited studies found in the field of MCM. Based on the researcher's experience working with FSIs for more than a decade, the problem in this case was the pre-selected channel for delivery of messages in the customer alert solution used by FSIs. Therefore, the motivation is to create a ML enabled multi-channel messaging system (with features which could facilitate the option of using different channels and over time can learn the channels which the customers are more reachable than the other) and then implement this in an integrated system of which the requirements had been derived qualitatively. Section 2.7 discussed the existing conceptual frameworks used by MCM system which currently lacks an ML-enabled channel selection module and other features that include message channel prioritization and message channel delivery routing as well as AI self-learning using the TOW approach.

4.4.2 Objectives of the solution

At this stage of the research, the requirements elicitation for the implementation for the ML-enabled MCM system were reviewed and defined. The ML-enabled MCM system was proposed to address some of the challenges inherent to the MCM model by providing a system that implements channels selection logic with AI embedded and deployed in an efficient manner to allow self-learning and channel optimisation. This research proposes an ML-enabled MCM system that has in-built capabilities for automatic message routing to available channels and message prioritization which provides customers with an opportunity to receive messages in a prioritised and consistent way from a variety of services available. The derived theory and literature review findings show that existing MCM has limited support for channel prioritization and selection using ML. The proposed system builds onto the existing framework by incorporating ML algorithms into its channel selection module which is the main contribution of this thesis.

4.4.3 Design and development

This DSR stage focuses on the design of the initial prototype of the IT artefact and subsequent development. In this stage, the researcher proposes a model for the ML-enabled customer alert system in multi-channel mode for the delivery of messages in a customer alert system.

A detailed requirement gathering and elicitation was conducted on the existing pre-selected multi-channel platforms used in an financial service institution as a customer alert solution with a technical review. Additionally, the author reviewed multi-channel message implementations which use publicly available applications hosted on mobile applications stores and FSI banking websites. A systematic review of the messaging platforms for FSI MCM applications available on Google Play and the Apple store was conducted (these stores being the most popular mobile application stores which host the financial mobile applications used by FSI mobile users). According to Roode, (1993), an approach was used to validate analytical inferences from texts and the website sections describing the use of each solution. In line with Walsham (1997a), this study considered the analytical constructs of each actor in the ML-enabled MCM system to determine their relationship in accordance with ANT principles. The association between the actors including their interest in maintaining the system without a change to their inherent properties were also considered (Latour, 2013; Tatnall & Gilding, 1999). The research work targeted websites which documented multi-channel messaging system, and their implementations. In addition, the research work adopted these targeted websites approach for publicly deployed applications on mobile application stores used by FSIs. Identifying mobile applications and websites among the wide number of financial institutions that manage or develop these mobile applications using the pre-selected channel MCM model can be challenging and is largely dependent on the approach utilised and how the review feedback is processed or managed. In accordance with the principles expressed by Etikan et al. (2016), the former enabled the researcher to select mobile applications and websites which provided the researcher with information relevant to this study, as well as ensuring that researcher selected major FSIs that used MCM to implement their applications. It was presumed that the selected FSIs were involved directly in the creation and or evaluation of the messaging solution under consideration. The researcher may not have had the opportunity to probe FSIs which knew about the matter under investigation if random sampling were used for the selection (Palys, 2008). Etikan et al. (2016) found that entities selected using purposive sampling were selected intentionally because of the attributes they contain in the area of research. The requirements for the MCM system derived from the systemic review was used to develop the artefact.

In the testimony of Abowd et al. (2011), gathering requirements is an effective strategy for determining the features that needs to be incorporated in a software artefact and used by users of a proposed system. In ANT, the relationship between each entity within the system needs to be clearly identified together with their roles (Langlais, 2006). Further to this, Monteiro

(2000) expounded that the effects of these interactions is critical to establish the interrelationship between the ecosystem of actors in the network. Tan (2016) declares that obtaining requirements is essential for a system designer, regardless of the settings in which it occurs. Tan (2016) emphasised the need for the requirements elicitation procedure to have the following elements:

- What the new system artefact must do and the challenges it proffers solution to.
- The features which the proposed system "MUST" have.
- An understanding that requirements may evolve as the research progresses from analysis to design and finally to implementation.

As professed by Kothari et al. (2014), qualitative content analysis is a method of analysing and finding significant themes in observation notes once a review has been completed. When using qualitative content analysis, the themes must be examined to identify qualities that effectively describe the problem statement. The primary themes of the research must be selected and documented to produce a broad criteria that will guide the creation of the new system. Krippendorff (1989) explains that qualitative content analysis may be utilised to establish reproducible and accurate conclusions in both the data and context domains. Silhavy et al. (2011) have pointed out that faulty requirements gathering could be a major issue during the development of a software artefact. On the account of the above-mentioned matters, the researcher proposed a qualitative context analysis of existing mobile applications and websites hosted publicly by FSIs that use the pre-selected MCM model to implement their customer transaction alert systems.

ANT suggests that technology adoption entails actions done at the organization level. This adoption level can improve both the perceived ease of use and ease of usefulness (Venkatesh & Bala, 2008). Since the proposed artefact is not an open-source system and is custom built, it is quite necessary to ascertain the limitation of pre-selected channels MCM system. This approach enabled the researcher to elicit and gather the requirements necessary to implement the model.

The design and development stage was carried out in an agile and iterative way, as pontificated by Hevner et al. (2004). In the first iteration, software development techniques and practices were drawn from existing implementation of MCM systems. The primary ML-enabled MCM system was designed based on the existing MCM conceptual framework described in Section 2.7 thereby giving the resulting methodology greater qualities and capabilities than the existing model. The enhancement of the methodology was a result of the introduction of load balancing between the disparate channels, TOW machine learning algorithm introduced in the channel selection and assignment logic. The DSR methodology by Peffers et al. (2007) was designed in a manner similar to the approach used in this thesis for the ML-enabled MCM model design. The fundamental method and practises of Peffers et al. (2007) were derived from existing DSR approaches, thus giving the methodology a solid foundation. The first sprint in developing the ML-enabled MCM system was focused primarily on acquiring the model design. This primary model was initially presented in a seminar paper and feedback on the channel selection from the participants were used to refine the methodology further. This body of work is detailed in Salami and Mnkandla (2020). Subsequently, to ensure that a more effective and efficient model is developed, the methodology was refined by applying feedback from the participants on the channel selection logic in the design cycle and process of building the model. The resulting model was presented to an audience at an international peer-reviewed conference. The resulting work is detailed in Salami and Mnkandla (2021).

In the second sprint of the design cycle, TOW algorithm was implemented in the model to improve the channel selection logic. This ML algorithm enhanced the system and provided a better alternative to existing models based on literature review that had revealed that most MCM implementation do not use ML algorithms for channel selection and management. The result of this stage was an adapted ML-enabled MCM artefact. The outcome was analysed further based on the research objectives of this study.

4.4.4 Demonstration

This stage of DSR defines the process of reviewing the developed artefact and applying the learning to a solve a wider range of problems. Sein et al. (2011) have emphasised that the demonstration stage involves more than only *solving* the current MCM preselected channel system problem, it also encompasses a *concerted reflection* on the main problem and the theories involved in the design. These proposed refinements to the design included software enhancement and patches to the newly constructed ML-enabled MCM prototype as well as significant changes to the structure, model and meta-functionalities, as deemed fit, while aligning the solution to the research objectives and ensuring that the goals are being addressed. Conceptual models aid in understanding a complex domain, and are useful for implementing a design idea (Robinson et al., 2016). Conceptual models consist of real aspects of the proposed system and represent most of the technical features of the framework. They illustrate relationships and linkages in a clear and explicit manner, and, according to Wand and Webers (2002), have the following three distinct purposes:

- Enable the technical and business analyst to grasp the intricacies of the problem domain.
- Enable a pathway for the design process to create a prototype with a set of inputs.
- The final requirements distilled from the raw requirements can be documented for future reference.

The research simulated the conceptual model linking the interactions and relationships between the entities within the ML-enabled MCM system in a controlled environment by contriving the model and testing it with messages while checking the throughput for delivery of the messages via the various channels integrated.

4.4.5 Evaluation

This stage encompasses the comprehensive evaluation of the ML-enabled MCM system through the use-*case approach*. The analysis of the insights gained from the evaluation of the artefact was used for the refinements of the system in each development iterations to improve the system. The method of classifying scenarios to record interactions between the system developed and related actors, is referred to as use-case modelling (Tiwari et al., 2019). Software development projects are built using high-quality requirement engineering approach that includes identifying and analysing scenarios from the specification. These scenarios can be recorded in various ways, such as prototypes, textual format and documentation during the requirements elicitation stage. Use-cases are a prominent method of capturing a software system's functional requirements (Mutch, 2002). The use-case research-based evaluation is consistent with the collection of qualitative data. Rather than performing full-scale randomized user evaluations, use-case research-based evaluation has been known to enable researchers to detect faults related to the system by making observations of the system's reaction to changes in the variables (Jacobson et al., 2016). The study generated user test conditions through use-case scenarios to evaluate the capabilities of the developed ML-enabled MCM model after the framework has been designed. The user test scenarios served as a step-by-step check for evaluating the model and its anticipated (and unanticipated) behaviours (Conway, 2016). The user tests were created directly from the requirements of the system and the research objectives to validate the model. Each outcome of the test was recorded against their expected test conditions. The model passed and met the exit criteria, since the overall tests passed in cycles 1 and 2 with no defects recorded in cycle 2.

4.4.6 Communication

The communication stage of DSR documents the knowledge discovered during the implementation and adds knowledge to both ML and messaging environments. In addition, the ML-enabled MCM problem in this thesis was communicated and the importance of the ML-enabled MCM system was presented at two peer reviewed international conferences. The design process, evaluation process research outputs was shared with the conference audience. The seminar paper presented at a conference was used to present the proposal to design and develop the ML-enabled MCM system while the second conference paper was used to present the preliminary design of the model. The audience at both seminars provided feedback that assisted in the design of the artefact. An international peer reviewed conference was used to present and discuss the preliminary ML approach using TOW algorithm for the channel selection module. This were discussed in both Salami and Mnkandla (2020) and Salami and Mnkandla (2021).

Figure 4.4 summarises the activities of DSR methodology as used in this thesis. Chapter 1 focused on defining the scope, research problem and motivation for the study, Chapter 2 presented a detailed literature review of MCM models and ML framework, and thus providing a refinement to the research problem and the elicited requirements for the ML-enabled MCM model. Chapter 4 detailed the DSR methodology and the pragmatist approach used in the design and development of the ML-enabled MCM model in this study.

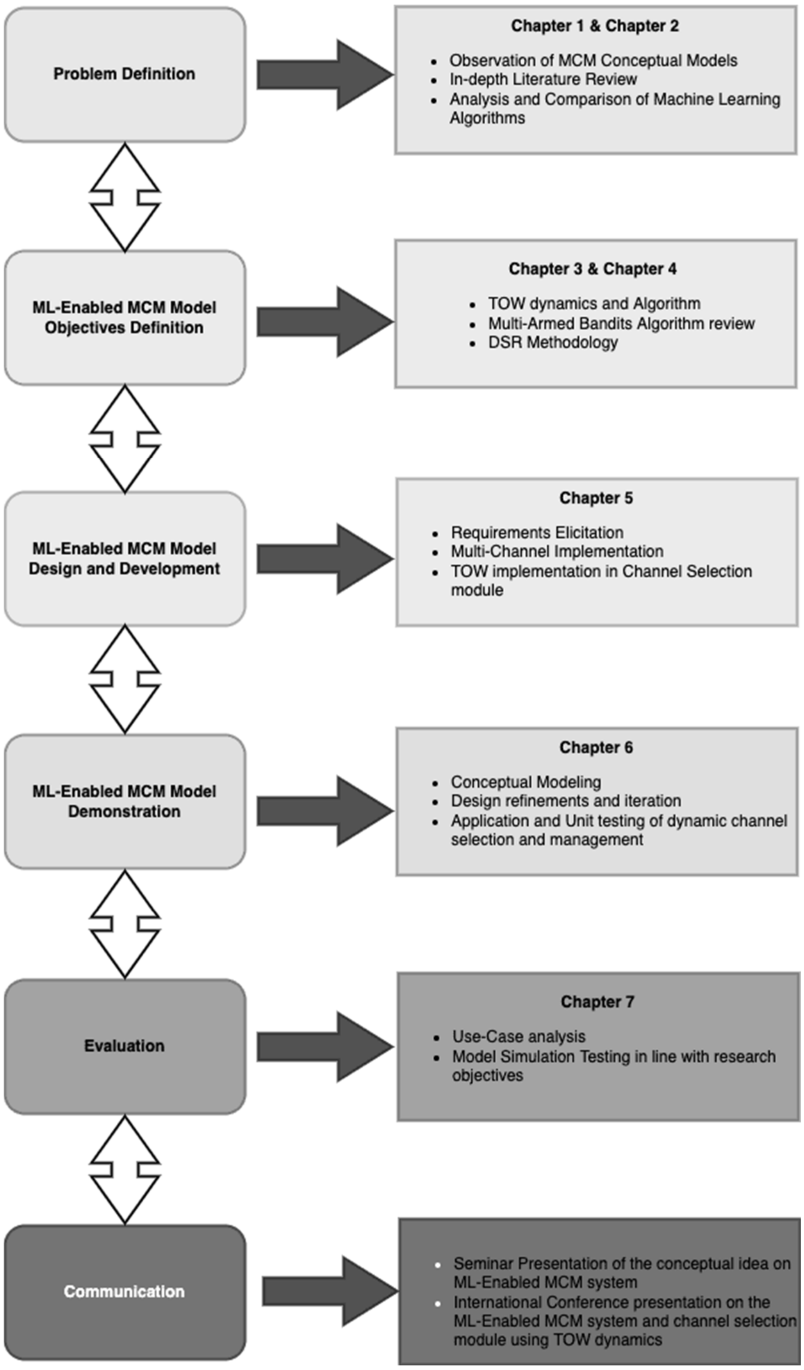


Figure 4.4: Using DSR Methodology Design to design the ML-enabled MCM model (Peffers et al., 2007)

4.5 ETHICAL CONSIDERATION AND APPROVAL

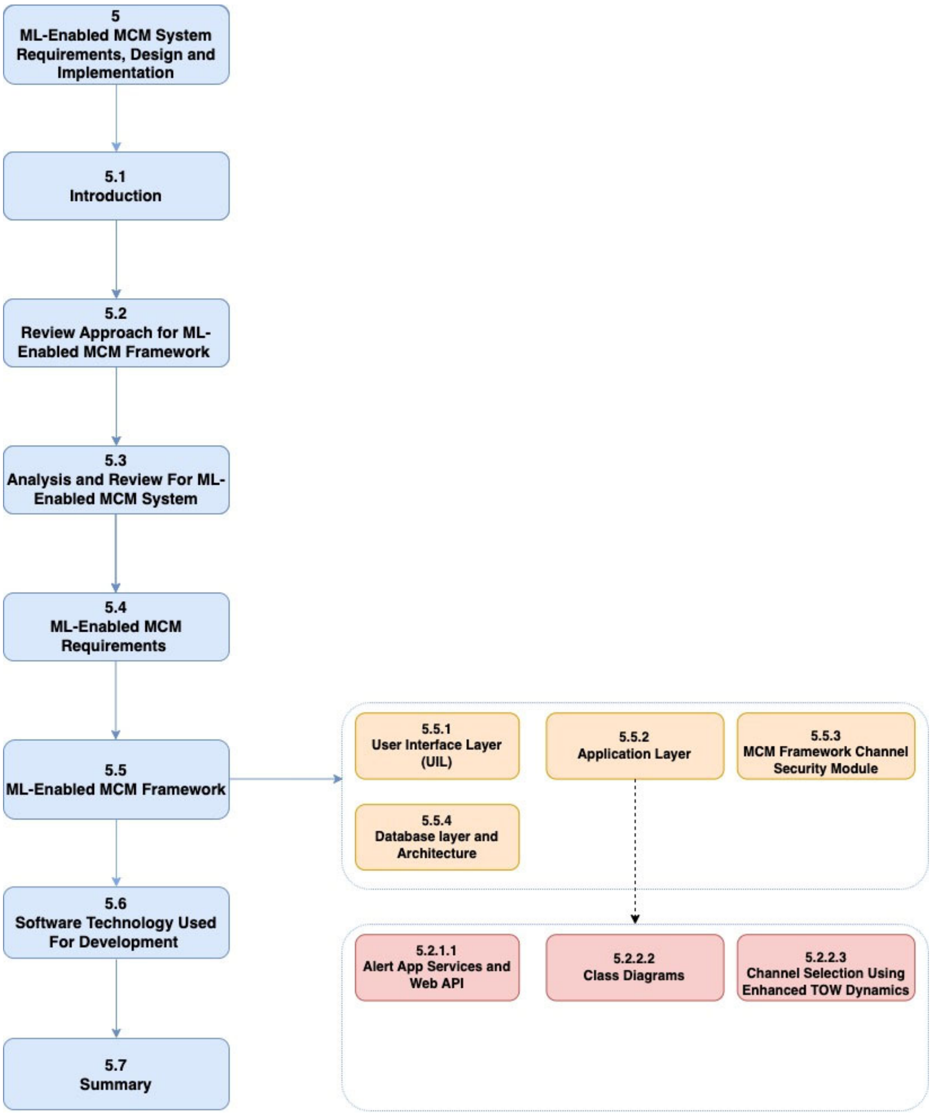
On the authority of Creswell (2009), researchers must protect study participants, promote research integrity, build trust with participants, defend against misbehaviour and impropriety that could reflect negatively on their organizations, and deal with emerging difficulties. It is the researcher's responsibility to guarantee that ethical standards are followed. In view of this, the relevant University authority was approached to request permission to undertake this study in line with the University's regulations and guidelines. To this end, an approval was granted (see Appendix A) before commencing the research work. This approval aims to guarantee that the study maintained research integrity, which includes a variety of good research principles and ethics, such as reliability, intellectual integrity, fairness, and safeguarding of the non-human participants in the study (Leedy & Ormrod, 2015). In addition to the numerous ethical concerns raised in the previous paragraphs, the researcher also ensured that the following requirements were in line with the approval:

- The study was conducted according to the guideline outlined in the approved ethical application.
- The ideals and concepts outlined in the UNISA Policy on Research Ethics are adhered to throughout the research thesis.
- The research study complied with all relevant national legislation, institutional guidelines , professional codes of conduct, and scientific standards relevant to this research, namely: *"The Protection of Personal Information Act, no 4 of 2013; the Children's Act no 38 of 2005; and the National Health Act, no 61 of 2003"*

4.6 SUMMARY

This chapter highlighted and discussed the research methodology adopted for this study. The research questions, research outline and methodology were reviewed in depth with a focus on Design Science Research Methodology which is most suitable for the artefact design. Furthermore, in-depth focus was placed on the requirements gathering of existing MCM models and the importance of using conceptual modelling to design and develop the ML-enabled MCM model using TOW algorithm in its channel selection module. The importance of use-case analysis, as it pertains to this study for the evaluation of the proposed artefacts, was also discussed. The next chapter analyses and reviews existing MCM framework with the elicitation of requirements for machine learning implementation in its channel selection module.

5 CHAPTER 5 ML-ENABLED MCM SYSTEM REQUIREMENTS, DESIGN AND IMPLEMENTATION



5.1 INTRODUCTION

In 4, a research methodology was adopted to guide this study. The two philosophical paradigms chosen were the Design Science Research (DSR) and pragmatist paradigms. The pragmatism approach offered the researcher the ability to conduct experiments under varying conditions with a view to evaluate the results of each scenario being tested to elicit requirements. Prior to conducting experiments, requirements pertaining to each instance of test to be performed must be well established. These set of requirements were derived from the existing multi-channel messaging (MCM) implementations deployed publicly by Financial Services Institutions (FSIs) on Apple iOS and Google Play store. Both Apple iOS and Google Plays store are the popular stores for hosting mobile applications. DSR was adopted for the design of the machine learning (ML)-enabled MCM artefact in a Customer Alert System used by FSIs to improve the channel selection logic module and the efficiency of the channel delivery system using ML algorithms in an organizational setting (Peffers et al., 2007). The organizational settings in this case, are the FSIs whose MCM systems are not using ML in their channel delivery selection logic.

The research methodology steps documented in Section 4.4 were strictly adhered to and this chapter presents an analysis of the MCM systems identified from literature and are hosted by FSIs on popular mobile application stores, with the view of eliciting and creating requirements for the ML-enabled MCM system and its application in FSIs customer alert system. The DSR methodology adapted from Peffers et al. (2007) in 4 provided a firm scientific base for this study while also enabling an implementation template for ensuring that detailed work and attention was given to the design and evaluation of the artefact produced by the study (Hevner et al., 2004). The chapter considers the channel selection behaviour identified in Sub-Section 3.2.3 and the ML algorithms discussed in Section 2.5.1 to be used for deriving the framework for building the ML-enabled MCM artefact.

Requirements for the ML-enabled MCM system were drawn mainly from the literature and from the review of existing MCM hosted by FSIs on popular public mobile stores (Apple iOS and Google Play store). The literature review conducted ensured compliance with the rigour cycle of the DSR methodology (Hevner et al., 2004) meant to situate the ML-enabled MCM artefact within the contemporary literature (Romero et al., 2020). DSR methodology allows the simultaneous evolution of both the design process and the artefact development while undergoing the design and evaluation process. This validation and review process built into the DSR methodology drives the evolution of both the MCM knowledge base and the development of the ML-enabled MCM artefact.

In addition, this chapter discusses a suitable design for meeting the requirements for the ML-enabled MCM artefact. Therefore, it is important to refer to the main research question of this research study, namely: "*How can integrated multi-channel messaging be implemented using machine learning algorithms to enable effective and efficient dynamic channel selection and integration methods?*". This requires the design and development of an artefact that will be used to examine the use of ML algorithms in the channel selection module of an MCM system.

However, as part of the design science research methodology, a conceptual framework was created during the process of designing and developing the solution. Zhu (2005) defined software architecture and design as an entity that has to be implemented.

The ML-enabled MCM framework was designed iteratively in line with the design science research (DSR) process of: (i) Problem identification, motivation and formulation; (ii) Objectives of the solution; (iii) Design and development; (iv) Demonstration; (v) Evaluation; and (vi) Communication of Results (Peffers et al., 2007). The benefit of the final artefact derived from the iterative development process was presented while inviting suggestions and feedback from attendees of an international peer-reviewed computing conference via an abstract paper presentation. Another presentation of the preliminary framework for the ML-enabled MCM framework was conducted at a peer-reviewed international conference on big data and advanced computing and data communications systems. Feedback received was used to enhance and build the corresponding artefact.

Subsequently, a review of the existing MCM implementations used by FSIs was conducted in Section 5.3 to understand and elicit more requirements for the artefact. This review was conducted on existing MCM platforms used by FSIs. This method was used to prove the use-case and validation of the artefact while using existing industry applications hosted on Google play and iOS stores. The chapter further discusses the DSR methodology and its application to the design and development of the artefact

Finally, the design and development of the ML enabled MCM artefact which investigates the use of ML algorithms in the channel selection module of customer alert systems used by FSIs was also outlined in this chapter.

5.2 REVIEW APPROACH FOR ML-ENABLED MCM SYSTEM

To understand MCM system implementation, there is a need to study how FSIs have implemented the MCM model in their current applications (see Section 2.4). The researcher identified and selected MCM applications, which are implemented by FSIs and deployed on Google Play and iOS Apple stores. These applications were then reviewed to determine their weaknesses and strengths, as well as to provide solution to the first research question, regarding the general understanding of the requirements which govern a ML-enabled MCM system. MCM term used henceforward implies a multi-channel messaging system whose channel selection system does not utilize ML algorithms in its channel module or an MCM system that uses pre-selected channels for message delivery.

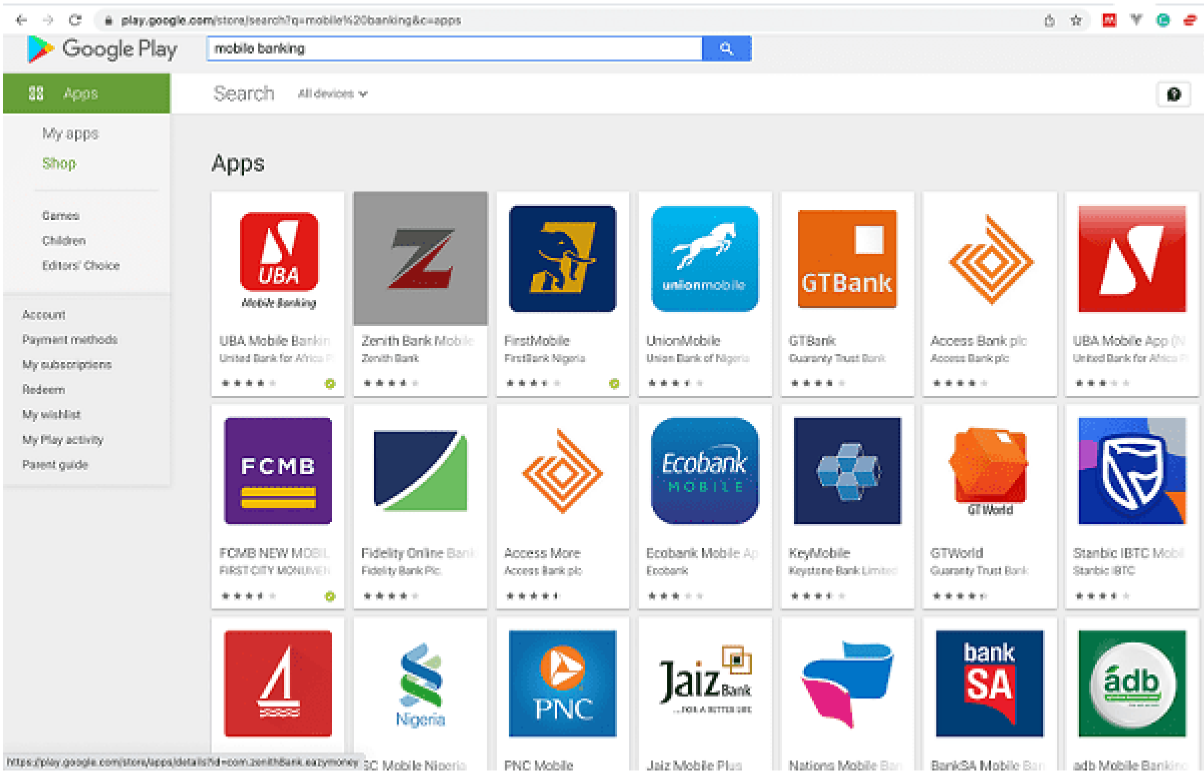
The pre-selected channel selection module problem of the current MCM system, as explained in Section 2.4, was aligned with the adapted problem definition stage of DSR, as documented by Peffers et al. (2007). Furthermore, the study reviewed publicly hosted financial services applications on the Android Google Play and iOS Apple stores, the public platforms where most FSIs host their mobile applications. Owing to confidentiality concerns and business competition amongst FSIs, the researcher opted for this approach because it was not practical to visit each FSI and conduct assessments of in-house hosted applications the FSIs had implemented based on the pre-selected channel module.

5.3 ANALYSIS AND REVIEW OF EXISTING MCM SYSTEMS

The knowledge base for this study comes from MCM systems that have already been implemented by FSIs. The literature and review of existing MCM systems hosted on public stores also assisted in determining how to develop the proposed ML-enabled MCM system (Hevner et al., 2004). Existing MCM systems, on the other hand, laid the foundation for the artefact under design. It is therefore essential, that this research searches these knowledge base on a regular basis to see if there are any updates or new applications and materials that have been added to these important sources. This way, both the design process and the artefact under development can stay up to date in line with the DSR methodology.

To sufficiently address the research question "*How can integrated multi-channel messaging be implemented using machine learning algorithms to enable effective and efficient dynamic channel selection and integration methods?*", the study used the problem definition approach, adapted from Peffers et al. (2007). The problems under investigation (pre-selected messaging channel, lack of ML algorithm in channel selection, message prioritization, lack of self-learning) were described and a design solution artefact (ML-enabled multichannel messaging system platform) was determined as a solution. These applications were further reviewed by downloading them from the application store. The registration module was reviewed to understand how the communication channel selected was used to send messages during user registration and transaction creation process.

Section 2.2 presented the features of the existing MCM system as well as its drawbacks to assist in the elicitation of the requirements of the ML-enabled MCM artefact. To choose the mobile banking applications that are publicly hosted on the mobile application stores, the researcher logged on to the Android Google Play store (same applications exists on Apple iOS store) and used the keyword "mobile banking" to search for applications hosted by FSIs in this category. It is noteworthy that the researcher was in the United Kingdom at the time of the conducting of the experiment with a view to understand delivery of messages outside the location of the selected FSIs located in Nigeria and to test customer out of profile message delivery. As articulated by Etikan et al. (2016), 6 mobile banking applications and 2 web banking solutions were selected purposively from the mobile application store. The researcher considered only applications that have been downloaded more than 1,000,000 times and developed in Nigeria to be included in the review and to aid in the conceptualization of the ML-enabled MCM framework. These applications were designed and hosted by Nigerian FSIs and were selected based on the mobile telephone number registered in Nigeria to receive messages or notifications on SMS, WhatsApp, Telegram and Facebook channels. It is important to note that these apps are available and used all over the world. The sample applications from the Google Play store that were used in the review are shown in Figure 5.1



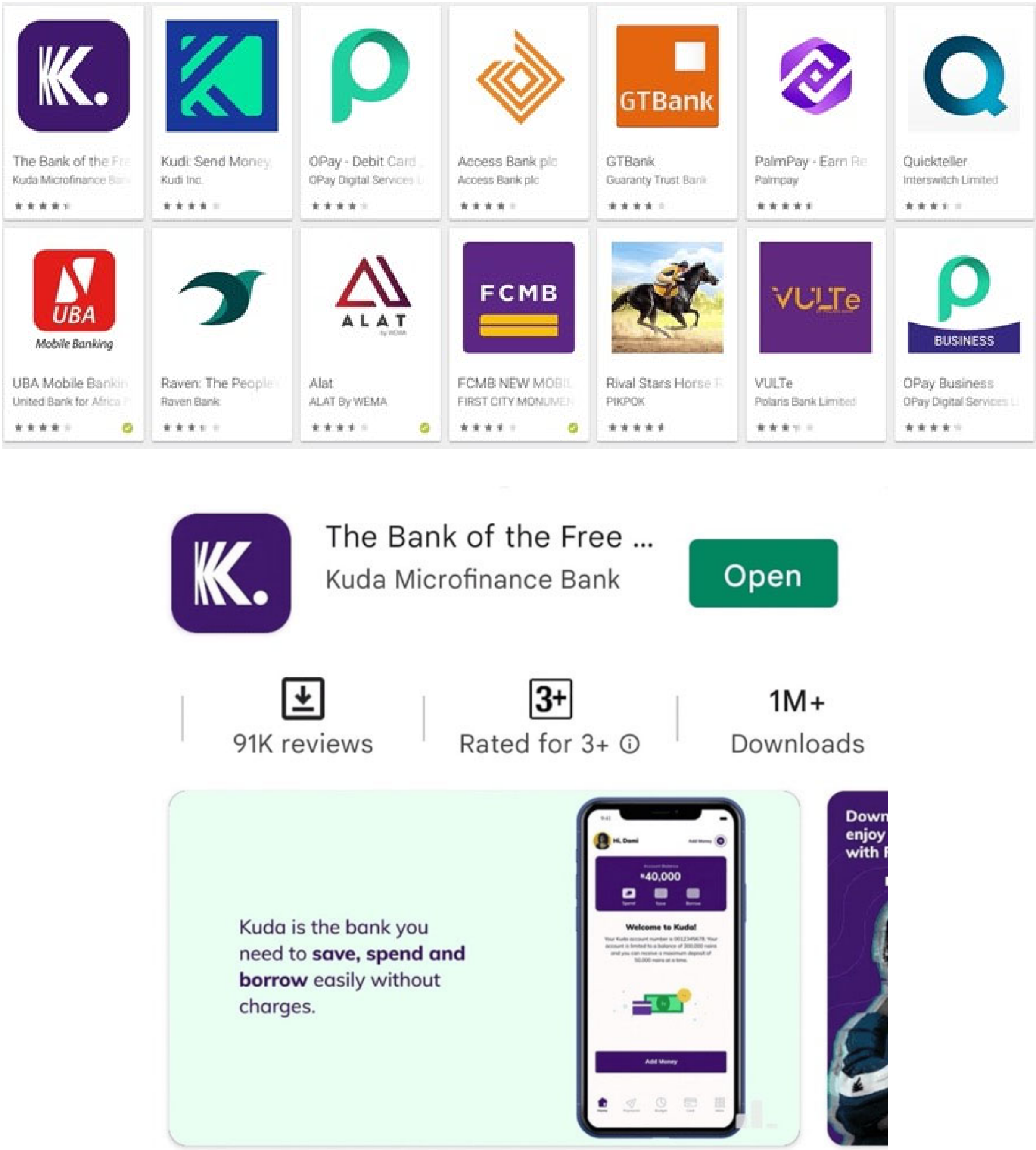


Figure 5.1: Selected Mobile Applications (Source: Google Play Store)

In line with the pragmatist approach, the researcher experimented the registration process flow by downloading Kuda Bank Mobile application from Android Google Play store. The details required for registration includes the following: email address, telephone number, bank verification number (BVN), date of birth (DOB), age, location and pre-selected notification preferences. For some other applications, only the use of a customer primary account number or existing credit or debit card was required for registration because any of this information can be used to retrieve existing customer data from the country’s centralized customer registration database. Feedback from the reviews is documented in Table 5.1.

Table 5.1: Review of MCM Applications Hosted by FSIs (Source: Google Play Store)

S or No	Name of Application	Features	Channel Priority	Number of Downloads	Channels Supported	Application Type
1.	Kuda MFB URL: https://play.google.com/store/apps/details?id=com.kudabank.app https://kuda.com	• Self-Onboarding • Virtual Cards • Bills Payment • 2FA Authentication • Credit or Debit Card Request	No	1,000,000+	SMS Email WhatsApp Telegram Facebook	Mobile App
2.	FirstMobile URL: https://play.google.com or store/apps/details?id=com.firstbank.firstmobile	• Creation of Accounts • QR Card Payment • Bills Payment • Token Synchronisation	No	5,000,000+	SMS Email WhatsApp Telegram	Mobile App

	www.firstbanknigeria.com	• Credit or Debit Card Request			Facebook	
3.	Alat Mobile URL: https://play.google.com/store/apps/details?id=com.wemabank.alat.prod https://alat.ng	• Signup • Loans • Transfer and Payments • Savings and goals • Virtual Dollar Card	No	1,000,000+	SMS Email WhatsApp Telegram Facebook	Mobile App
4.	UBA Mobile URL: https://play.google.com or store or apps details?id=com.uba.vericash https://www.ubagroup.com	• Leo AI chatbot • Biometric(facial or fingerprint) Authorization • ATM Branch Locator • Transfers	No	5,000,000+	SMS Email WhatsApp Telegram Facebook	Mobile App

		• Banking Services on Self Service				
5.	Access Mobile URL: https://play.google.com/store/apps /details?id=com.accessbank.accessbankapp https://www.accessbankplc.com	• Intra or Interbank Transfers and Payments • Bill Payments • Cable TV • Stop or Cancel or Order Cheques • Bank Branch Locator	No	1,000,000+	SMS Email WhatsApp Telegram Facebook	Mobile App
6.	Stanbic IBTC Mobile URL: https://play.google.com/store /apps/details?id=com.stanbicibtc.mobile https://www.stanbicibtc.com	• Voice Banking • Investment • Trade • Transfers and Payments • Insurance Module	No	1,000,000+	SMS Email WhatsApp Telegram Facebook	Mobile App

7.	TajBank URL: https://ibank.tajbank.com/root/login	• Inter or Intra Bank Transfers • Self-registration • Password for transaction authorisation. • Dashboard personalisation. • Utility bills payment. • Airtime recharge. • ATM or Branch locator.	No	N or A	SMS Email WhatsApp Telegram Facebook	Web App
8.	Fidelity Bank URL: https://online.fidelitybank.ng/#login	• Self-registration • Customizable Dashboards • Bill payments • ATM Branch Locator	No	N or A	SMS Email WhatsApp Telegram	Web App

		• Intra or Inter Bank transfers			Facebook	

Eight (8) applications were reviewed in total, with specific features noted for MCM implementation in the registration and transactional features offered by the applications. All the applications reviewed have implemented MCM support as per channels supported and documented in Table 5.1. The researcher noted that all the applications offered similar service features with different product names. In addition, during the registration or transaction process, messages are sent to customers via pre-selected messaging channels without considering previous deliveries, customer location or preferences owing to lack of ML in the message channel selection logic.

The registration process flow for the Kuda Mobile Banking application is documented in Figure 5.1. After downloading and installing the application the researcher clicked the “Start Button” to commence the registration process. Thereafter, an option to enable in-app notifications was presented and accepted. Subsequently, in the flow, the researcher inputted in the required fields the name, address, telephone number, email address and BVN. A one-time password (OTP) code was sent to confirm the creation of the account via email in the preregistration. However, after the registration, another OTP code was sent via SMS to confirm successful registration on the platform. No option was provided to capture any other social media channel addresses during the registration process. The KUDA application automatically pre-enrolled the researcher’s account for SMS and email delivery. Upon logging into the application settings, no option was made available to automate other message delivery channels. Since the researcher was roaming the telephone number used for the message delivery significant cost was incurred and there was a delivery delay in the registration process owing to network connection issues.

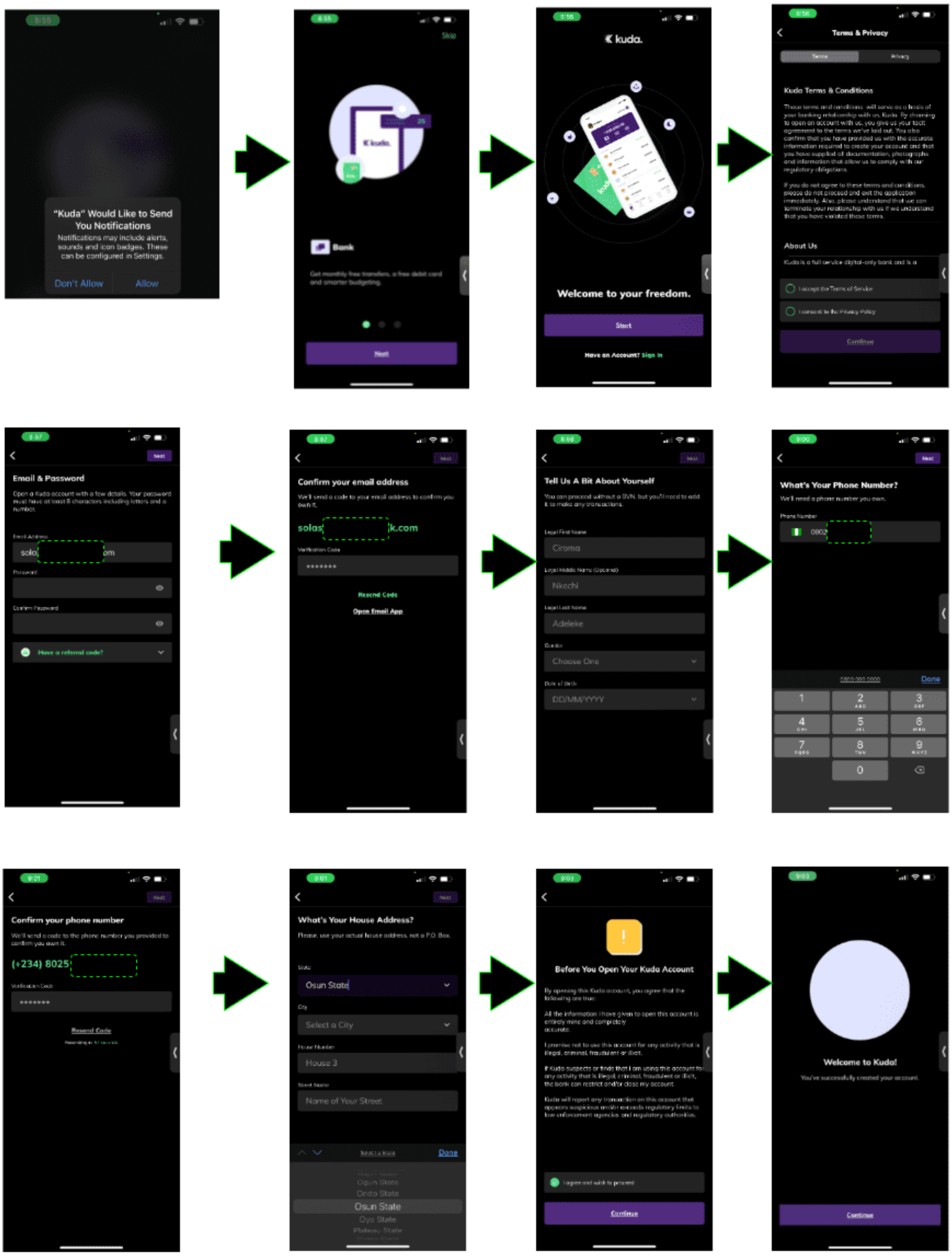


Figure 5.2: Kuda Bank Mobile Application Registration Flow

5.4 ML-ENABLED MCM REQUIREMENTS

Requirements are qualities or features that an artefact under development must possess (Garg et al., 2015). The artefact being designed is evaluated in line with DSR methodology to assess the usability of the ML-enabled MCM artefact being developed (Hevner et al., 2004). Requirements are solutions (or artefact) objectives in the context of the DSR approach. The objective of this research is to develop an ML-enabled MCM system. This chapter discusses both the requirements and system objectives and subsequently the design of the artefact. These requirements were elicited from the review in Section 2.4 and 5.3 to establish the objectives for the artefact under design which is comprised of the following layers:

- **Channel Service Layer**: Channel service layer enables web service support and allows channel transaction management and quality of service in terms of availability of each integrated channel. This layer also manages routing of messages between the disparate channels within the MCM system (Ganesh et al., 2004).
- **Channel Gateway Layer**: This layer handles the business logic of channel selection using ML algorithms, security, and customer data management in a secured manner. This layer also parses agnostic data messages exchanged between the channels to ensure they are not compromised by stripping malicious content from the message structure (Khan & Siddique, 2004).
- **Channel Service Integration Bus Layer**: The integration bus layer maintains integration of downstream services and web service orchestration. This layer also manages channel sequencing, message routing and overall coordination between the other channels and services within the MCM system. Message translation, exception handling and channel meta-data are also maintained at this layer to ensure that, if there is fault, the system can resume from its previous state of the business operation.
- **Channel Decision Making and Learning Layer**: This layer implements the channel decision and learning module with the use of ML and is linked with the integration bus layer. The ML algorithm implemented at this layer is in line with TOW dynamics discussed in Subsection 2.5.6 and channel selection in Section 2.6. Messages are routed to delivery channels based on urgency, message content, and channel availability. Using the enhanced TOW algorithm, this layer learns the reward (successful message delivery) or loss (failed message delivery) and applies this learning in routing messages to each channel on the platform. Message delivery, channel availability status and reliability are determined and computed within this layer.

- **Channel Message Transformation Layer**: The layer responsible for message translation within the module. Due to the integration of MCM channels (SMS, email, WhatsApp, Telegram, Twitter DM, and Facebook Messenger), each channel has its own message implementation (see Appendix E). This layer also implements a lightweight data agnostic messaging format JavaScript Object Notation (JSON) used to send messages within the system. This format is easily adapted to each channel's specification while maintaining the message structure and integrity.

Figure 5.3 depicts the relationship between the major service components necessary to implement an ML-enabled multi-channel messaging system as applicable in customer service alert system used by FSIs. Each service component is built with resilience and adaptability to recover from failures due to the tightly coupled nature of the overall artefact. Furthermore, the MoSCow approach (Must Haves, Should Haves, Could Haves and Won't Haves) was, as touted by Waters (2009), applied to classify these features as "Must Haves" in the design of the ML-enabled MCM artefact due to their criticality for the smooth operation of the system.

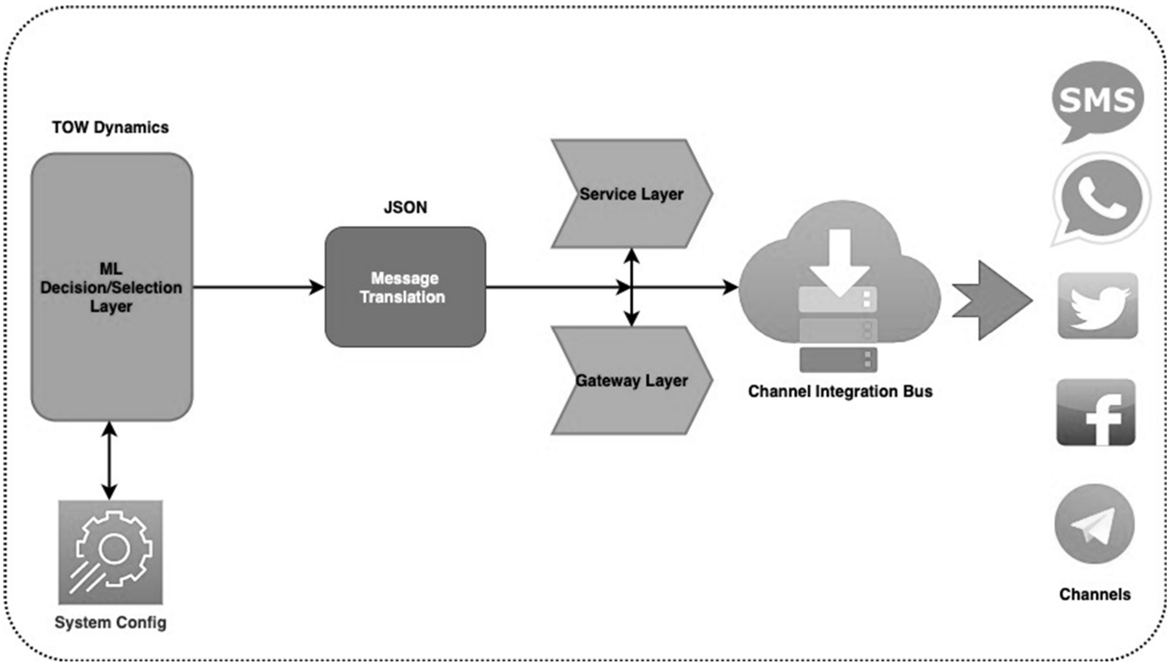


Figure 5.3: ML-Enabled MCM System Requirements

5.5 ML-ENABLED MCM FRAMEWORK

The ML-enabled MCM framework is implemented in a customer transaction alert system as a 3-tier application. The first tier of the application consists of a presentation layer or user interface layer. The second tier is the application layer which consist of data translation, message service bus, message routing, Application Programming Interface (API) layer, machine learning and channel selection logic. The third tier is the database layer storing information about system configuration and channel state management. Figure 5.4 presents the architectural design of the artefact.

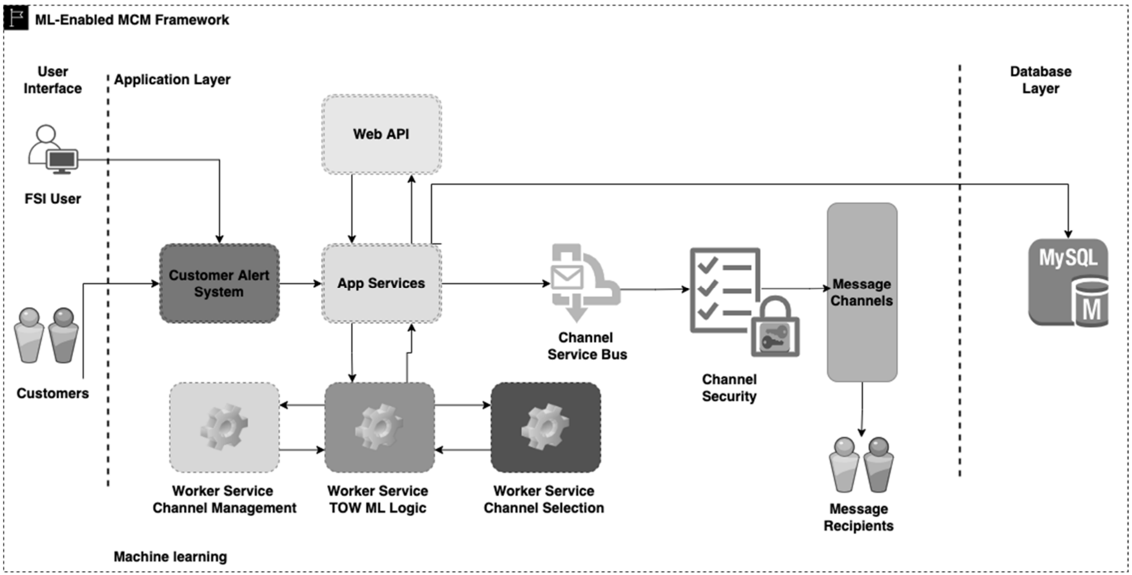


Figure 5.4: ML-Enabled MCM Architecture

The user interface setup and management interface allow FSI designated agents to log in to a web-based interface to profile a customer on the platform. The application allows the user to assign message channels to customers based on initial preference, manage customer information and setup of each channel's profile. For example, a typical setup requires the FSI user to capture a unique customer identifier from the FSIs core banking platforms and then link this to a telephone number for SMS, Telegram, WhatsApp message delivery, email address for email message delivery, twitter handle for Twitter direct message and Facebook identity for Facebook messenger delivery. The user interface allows the FSI agent to manage the customer life cycle, generate relevant reports and to monitor the system performance during operation.

The application layer handles and processes all the requests from the web-based interface. This layer is further broken down into an API managed layer that exposes secured services via a Representational State Transfer (REST) interface specification for third party integration; worker services module executing as background services responsible for channel state management, ML and channel selection logic; and, an integration with a customised message service bus module responsible for the integration of the support messaging channels, such as SMS, email, Facebook, Twitter, and Telegram. The REST web service layer allows customers to self-provision the channels from their mobile or web banking applications by selecting channels of their own choice. An overview of how the system's three main layers interact is provided in Figure 5.4. The next section presents detailed description of each layer's implementation.

5.5.1 User Interface Layer

FSI user's login via a secured web module with authentication to the customer alert web system (see Figure 5.5). The web system prototype allows the users to profile and customers using a predefined bank customer identifier registered on their core banking system. The customer information such as name, address, and age are retrieved from the existing database. The FSI user uses the web interface screen to populate messaging channel details requested by the customer. The messaging channels available include SMS or WhatsApp or Telegram, which requires the use of a registered mobile telephone number, an email address, Twitter DM, Facebook messenger identity and Instagram user identity.

Apart from the user's telephone number and email address, none of the other channels are, as shown in Figure 5.6) mandatory on the system. Customers also have a self-service option powered by Web API's that exposes the functionality of the customer alert system allowing them to provision these channels for message delivery.

The Web API's allows 3rd party applications such as internet banking and mobile banking solutions which customers are already using to connect to platforms seamlessly. Customers may choose any channel they wish to be alerted on for their transactions at any time. However, the system uses its intelligence via ML to understand and prioritise each customers preference based on usage and delivery data over a period of use.

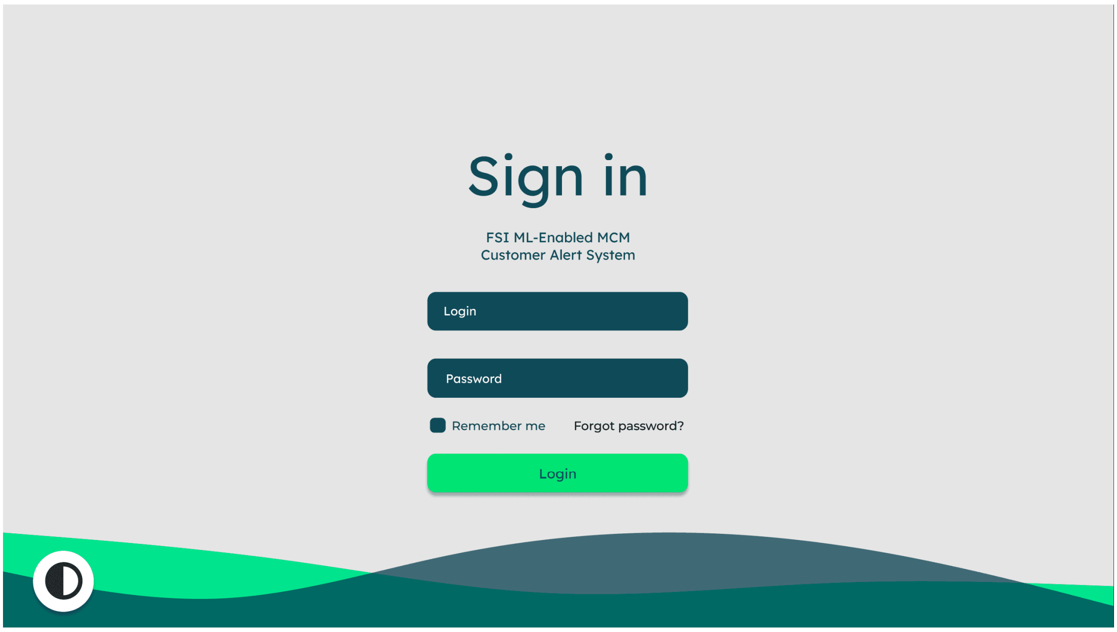


Figure 5.5: UI Login Screen to FSI Customer Alert System

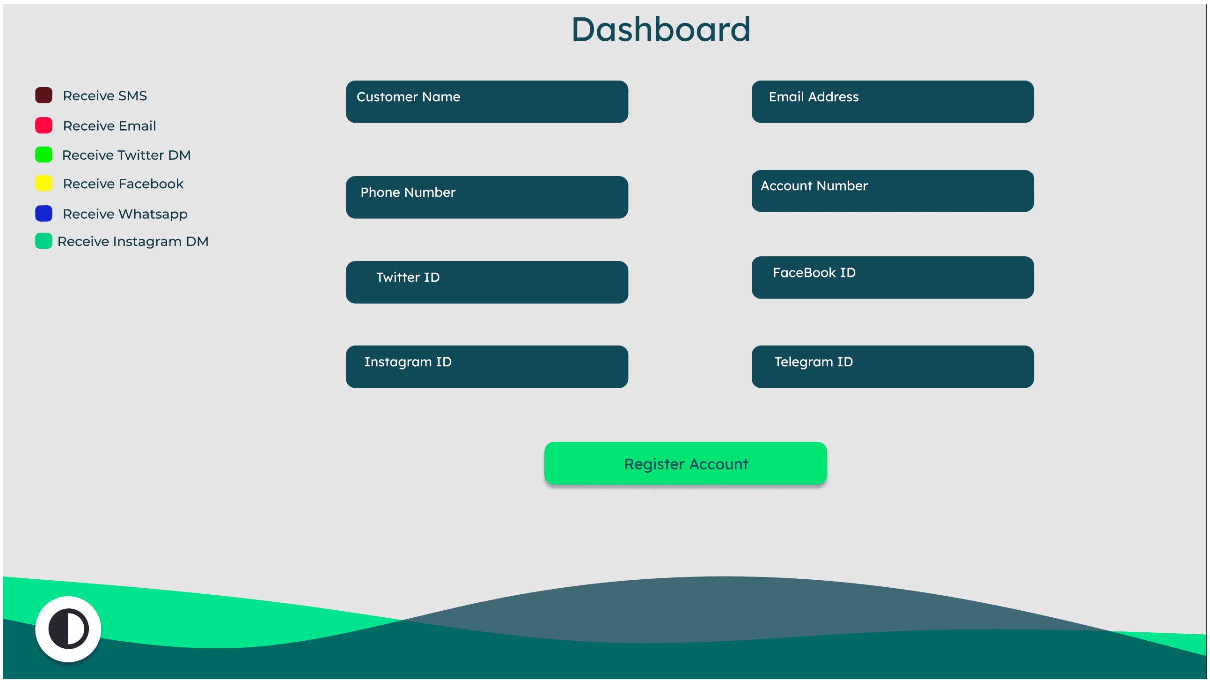


Figure 5.6: UI Login Screen to Setup Customer for Message Channel Delivery

Figure 5.7 presents the system sequence diagram for customer onboarding by an FSI user or customer user. From the normal flow of events in Figure 5.7, the user has captured or selected the message delivery channels available and submitted to the customer alert system via Web

API. The inputs are validated by the customer alert system and, if successful, the customer configuration is stored on the database.

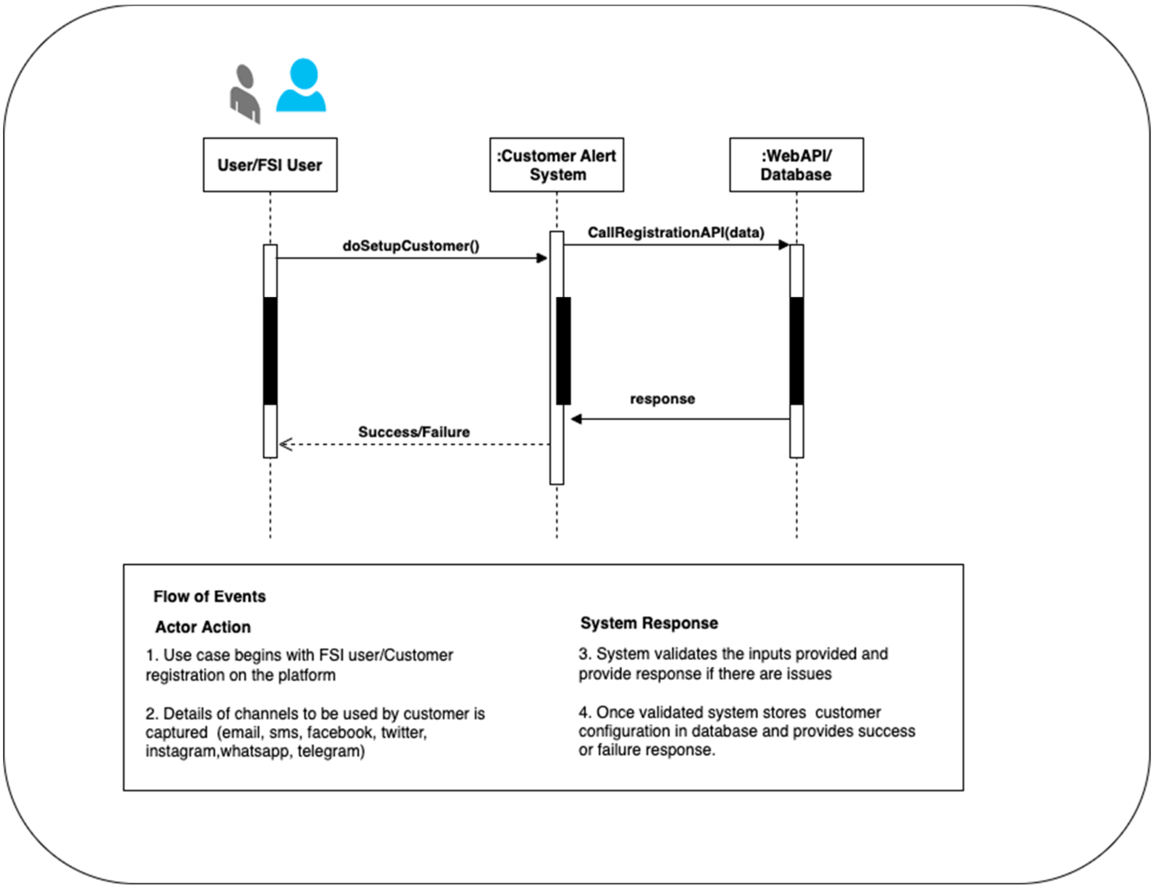


Figure 5.7: Sequence Diagram for Customer Onboarding

5.5.2 Application Layer

The application layer of the ML-enabled MCM framework consist of 3 main components, namely: the alert application services and Web API; ML module; and channel service bus.

5.5.2.1 Alert Application Services and Web API

The alert web services layer was implemented with C#.NET core programming language, and the ASP.NET core is an open source platform (Jatinder, 2020). The .net core run time is fast in execution and provides native out-of-the box support for Web API development. The .net core platform is a cross platform and can thus seamlessly operate on Windows, Linux, Unix or any other operating system. C# also supports a wide range of applications such as ML, IOT, Desktop, Gaming, Cloud and AI. The application services layer is responsible for parsing input data from the user interface (UI) layer. Input data such as name, customer account number, address, selected messaging channels are validated prior to acceptance for storage in the database layer. The application layer also exposes the validation and application logic via REST API services.

The Web API module exposes application services such as *createCustomer(customerData), manageCustomer(customer), getCustomer(customer)* to allow third party application integration and also to separate concerns between each layer of the framework.

5.5.2.2 Class Diagrams for Alert Application Services and Web API

Class diagrams are used in object-oriented programming design to describe an artefact's framework, the characteristics, functional methods and sub-routines. As highlighted in sub-Section 5.5.2.1, the ML-enabled MCM system includes source codes written in C# ASP.NET core object-oriented platform. C# is the programming language of choice for the ML-enabled MCM system for this reason, as well as other reasons listed in Section 5.5.2.1 of the chapter. The primary classes required are *createCustomer, manageCustomer, and sendMessageClass, and channelManagementTOW* (responsible for channels states and learning), all of which include characteristics and methods to perform the required tasks. *channelManagementTOW* is a class that has characteristics and methods that relate to the management of each channel, ML and states management. Following the display of the class diagrams in Listing 5.1 - Listing 5.8, each method is given a brief description of its responsibilities in the framework.

sendMessageClass()
#Attributes stringChannel: String stringCustomerId: String stringServerIP: String stringServerID : String stringAppSenderID: String stringToDate: String stringSenderName: String stringSubjectId: String boolStatus : bool strChannelAvailability: int

#Methods + setChannel (strChannelID: String) + setCustomerId (strCustomerId : String) + setServerId (_strServer : String) + setToDate(strTodate : string) + setSenderId(str: SenderId) + getSenderId () : String + getChannelsStatus(strCustomerId : String) : List + getChannel(status : bool): List + sendMessageToCustomer (strCustomerId: String): bool

Listing 5.1 Class Diagram of SendMessageClass

createCustomer(),manageCustomer(),
#Attributes strCustomerId: String strCustomerName: String strCreateDate: String strFacebookId: String strInstagramId: String strWhatsappId: String isActive: Bool strPhoneNumber: String strTelegramId: String int ChannelCount : int
#Methods + AddCustomer (getCustomer(): String) + ManageCustomer(getCustomer(): String) + setCustomerId(strCustomerId : String) + setChannel(strCustomerId:String) + getCustomer(strCustomerId : String) : String + getCustomerId (strCustomerId: String) :String + setChannels(strCustomerId:String) + getChannels(strCustomerId : String) : List

Listing 5.2: Class Diagram of createCustomer,manageCustomer

channelManagementTOW ()
#Attributes strCustomerId: String strChannel: String lstChannels: List isActive: Bool
#Methods + setCustomerId(strCustomerId: String) + setChannel(strCustomerId: String) + getChannelsList(strCustomerId : String) : List + getCustomerId (strCustomerId: String) :String + executeTOW(strCustomerId: String) : String + ChannelStatus(strCustomerId: String) : List

Listing 5.3: Class Diagram of ChannelManagementTOW

Listing 5.4- 5.8 shows the class definitions for the methods defined the sendEmailclass, channelManagementTOW and createCustomer or manageCustomer classes.

Class Contract Name	executeTOW(strCustomerId: String)
Use Case	Execute TOW Machine Learning
Responsibilities	Run TOW ML algorithm to manage channels availability and status
Exceptions	None
Pre-Conditions	Customer is set up with at least one channel for message delivery
Post-Conditions	The channels available to deliver message for each customer is returned

Listing 5.4: Specification of ExecuteTOW Class Method

Class Contract Name	ChannelStatus(strCustomerId: String)
Use Case	Get the status of channels for a customer
Responsibilities	Retrieve the status on available for message delivery at an instance of time
Exceptions	None
Pre-Conditions	Customer is set up with at least one channel for message delivery and TOW algorithm has been executed to manage channel state
Post-Conditions	The list of channels available to deliver messages to a customer

Listing 5.5: Specification of ChannelStatus Method

Class Contract Name	sendMessageToCustomer(strCustomerId: String)
Use Case	Send a Message to a customer
Responsibilities	Send message to a customer on a selected channel available
Exceptions	None
Pre-Conditions	Customer is setup with at least one channel for message delivery and TOW algorithm has been executed to manage channel state

Post-Conditions	Message sending status, retry another channel if message sending failed

Listing 5.6: Specification of sendMessageToCustomer Method

Class Contract Name	AddCustomer (getCustomer(): String)
Use Case	Create a customer on the platform
Responsibilities	Create customer and assign channels based on initial preference of the customer
Exceptions	None
Pre-Conditions	Customer is set up with at least one channel for message delivery based
Post-Conditions	Customer is created successfully, and messaging channels assigned

Listing 5.7: Specification of AddCustomer Method

Class Contract Name	ManageCustomer (getCustomer(): String)
Use Case	Manage existing customer on the platform
Responsibilities	Modify existing customer data and channel preferences
Exceptions	None
Pre-Conditions	Customer is existing and setup with at least one channel for message delivery based
Post-Conditions	Customer data is updated successfully and

Listing 5.8: Specification of ManageCustomer Method

5.5.2.3 Channel Selection using Enhanced TOW Dynamics

The logic of operation of the TOW machine learning algorithm outlined in Section 2.5.6 is in line with the approach adopted by Ma et al. (2019). The TOW algorithm was enhanced by applying it to the selection of disparate channels co-located in a multi-channel messaging system of a customer alert system used by FSIs. Once set up on the system, during operation, the customer is mandated to pre-select at least one messaging channel to be used for message delivery. Sending a message to a customer can be initiated by the FSI for messages such as general information, emergency information, product, and market advertising; alternatively, the message can be self-initiated as in the case of customer-initiated inquiry or transactions, OTP delivery and balance statements.

To understand the operation of TOW channel algorithm in an ML-enabled MCM system, a messaging scenario where a customer has pre-selected message channels such as SMS and email and WhatsApp is considered. A message producing node *P*, produces an OTP request due to a transaction initiated by a customer (see sample message ß in Figure 5.8).

From {SenderId}

Dear {Salutation},
Please use PIN {XXXX} to complete your
transaction for NGN100.00 with FSI Card ending
{Card}. If you did not request a PIN,
Please call us.
Terms & Conditions apply

Figure 5.8: Sample message by a Message Producer

The parameter *{SenderId}* is the FSI sender ID which is a name the customer is familiar with for business operations within each FSI. The *{PIN}* and *{Card}* parameters are the unique identifiers generated for the customer to identify the customer's own ATM card and secure a PIN to be delivered to securely complete the transaction. In principle, the system is not limited to a single message producer because each node is bound and under operation of the multi-armed bandit (MAB) principle. Listing 5.9 shows a typical configuration of a message producer.

Config.xml

```
1  <?xml version="1.0" encoding="UTF-8" ?>
2  <definitions name="ConfigureMessageProducers"
3  targetNamespace="http://namespaces.mcm.producers.com"
4  xmlns:es="http://www.snowboard-info.com/mcm.wsdl"
5  xmlns:esxsd="http://schemas.snowboard-info.com/mcm.xsd"
6  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns="http://schemas.xmlsoap.org/wsdl/">
7  <!-- omitted types section with content model schema info -->
8  <message name="MessageProducer">
9  <part name="body" element="esxsd:MessageProducer"/>
10 </message>
11 <message name="Producer">
12 <part name="body" element="esxsdMessageProducerResponse"/>
13 </message>
14 <portType name="PushMessageProducerPortType">
15 <operation name="MessageProducer">
16 <input message="es:PushMessage"/>
17 <output message="es:PushMessageProducer"/>
18 <fault message="es:PushMessageFault"/>
19 </operation>
20 </portType>
21 <service name="MessageProducer">
22 <documentation>MCM.com Web service for MCM Message Producer</documentation>
23 <Endpoint name="MCMPort" binding="es:netTcpRelayBindingChannels"/>
24 <soap:address
25 location="http://www.mcm.com/mutlichannelmeessagingsystem"
26 port="8082"/>
27 </service>
28 </definitions>
29
```

Listing 5.9: MCM Message Producer Configuration

Tags <service name="MessageProducer"> (as per line 15 in Listing 5.9) defines the service name to be called by the application service client, <soap:operation> specifies the service contract and operation that can be performed by the calling application client. The service clients can pass data defined in the contract schema located and published in “http://schemas.mcm.com or MessageProducers.xsd”.

The web service descriptor language WSDL specifies a fault message contact specified in <faultmessage=”es:MessageProducerFault> (see line 18 in Listing 5.9) which enables the calling client to be notified of any error in the service. The service contract <outputmessage=”es:PushMessageProducer> (see line 17 in Listing 5.9) provides a success status to the calling client in when the message is successfully pushed.

Subsequently, the group of message consumers or channels *C*, which is integrated and available to transmit messages being managed by the ML-enabled system, includes SMS, email, WhatsApp, Telegram, Instagram and Facebook. The channel configuration for these channels is depicted in Listing 5.10.

appsettings.json

https://json.schemastore.org/appsettings.json

```
{
    "ConnectionStrings": {
        "DefaultConnection": "DataSource=MCMapp.db"
    },
    "Logging": {
        "LogLevel": {
            "Default": "Warning"
        }
    },
    "AllowedHosts": "*",
    "NoOfThreads": 5,
    "Channels": [
        {
            "ChannelId": "Twitter",
            "isActive": true
        },
        {
            "ChannelId": "SMS",
            "isActive": true
        },
        {
            "ChannelId": "Email",
            "isActive": true
        },
        {
            "ChannelId": "Facebook",
            "isActive": true
        },
        {
            "ChannelId": "WhatsApp",
            "isActive": true
        },
        {
            "ChannelId": "Telegram",
            "isActive": true
        }
    ]
}
```

Listing 5.10: Channel Configuration And Database Settings

The customer preference is filtered automatically based on customer selected choice eliminating the need to use other channels such as Twitter, Facebook, and Telegram. Any of the customers pre-selected channels (i.e., SMS, email and WhatsApp) is ready and available by configuration to accept and transmit the message ß at an instance of time. This was defined and expressed earlier as the throughput value t_{pi} responsible for determining the total number of messages processed at an instant of time by the channels t_{pi}. The formula $t_{pi} = \sum_{j=1}^{|C|} n_{ij}$ n_{ij} is used to determine the number of messages provided by p_i to c_j. The TOW algorithm using Upper Confidence Bound (UCB) exploitation and exploration provides a balanced throughput within the system by ensuring that each message producing node *P* has a channel available from the 3 channels pre-selected by the customer to deliver the message at an instance of time. TOW guarantees channel selection assignment using a fully decentralized operational method. Hence, when a message is delivered successfully, +1 is added to the throughput t_{pi} if the channel C can accept and transmit the message termed as a reward to the message producer. Otherwise, −ω is added to t_{pi} and defined as follows:

Reward = $\sum_{i=1}^{|C|} n_{ij}$ + 1 and Loss = $\sum_{i=1}^{|C|} n_{ij}$ −ω.

The channel selection worker service determines the channel to be used to transmit the message and relies on the TOW or UCB channel selection mechanism to decide how a message ß is routed from the message producer to the messaging channel node. Each channel, in this case, can accept some of the message ß if the channel is available to process the message. The system iterates with channels that have shown their reliability over time using the previously stored reward values. Some assumptions must be made about the behaviour of the system such as (i) The system must be initialized and operate in discrete time steps. (ii) Each message must be sent one step at a time (the actual creation of messages can also work instantaneously without interfering with the mechanism of the framework), and lastly (iii) the model must maintain a uniform network connectivity. As shown in Table 5.2, the SMS, email and WhatsApp channels at this stage are in the waiting state. This is based on the assumption that the system is operationalised the reward and loss values are set according to Table 5.2 to maintain the initial equilibrium of the TOW channel selection logic.

Table 5.2: Channel State In Initialised State

S or No	Channel	Reward	Loss
1.	SMS	+1	N or A
2.	Email	+1	N or A
3.	WhatsApp	+1	N or A

In this initialised state, based on the configuration and order the TOW-worker service returns the SMS channel to be used to deliver the message to the customer. As earlier stated, message channels receive a reward response based on whether the message sent was delivered successfully, and this is expressed as R_{pi} = $\sum_{j=1}^{|C|} t_{pi}^{ß} n_{ij}$. Let us assume further that the SMS network is not available or busy processing another message. The TOW-worker service kickstart its exploration and exploitation strategy by taking the next available channel in that order (in this case the email channel). This service kickstart is based on the assumption that the channel in question is available for message delivery (i.e., the email channel reward estimator is incremented by +1). Table 5.3 shows the state of the channels in the processing state.

Table 5.3: Channel State in Processing State

S or No	Channel	Prev. Reward	Prev. Loss	T_{pi}
1.	SMS	+1	-ω	+1
2.	Email	+1	N or A	+2
3.	WhatsApp	+1	N or A	+1

At this stage, if the customer initiates another transaction that generates a new message for transmission, the channel selection worker service initiates a new call to the TOW worker service to get the channel ready for delivery. In this case, let us assume that the network issue with the SMS channel has been resolved and the channel can process the message. The TOW worker service evaluates the channel availability state using the values of the channel state listed in Table 5.3 and decides to use its internal logic. Prior to this, the TOW worker service initiates a self-assessment to determine the performance of each channel in the system. This approach allows the system to update the reward and losses state for each channel based on the observation of the channels and their current state. In line with the TOW ML framework, the channel with the highest T_{pi} would be selected. In this case, channel with the highest T_{pi} is the email channel because the T_{pi} of the e-mail is 2 see Table 5.4. The email channel is only be returned after further evaluation has been performed on its network availability, otherwise the next channel, that is, WhatsApp, is evaluated and returned. The final T_{pi} for email channel is shown in Figure 5.9.

Table 5.4: Channel State in Final State

YS or No	Channel	Prev. Reward	Prev. Loss	T_{pi}
1.	SMS	+1	-ω	+1
2.	Email	+1	N or A	+3
3.	WhatsApp	+1	N or A	+1

The exploration and exploitation balance which provides the MAB trade-off is achieved by the TOW worker service through ensuring that each channel state is re-evaluated based on the last processing states (see Figure 5.10).

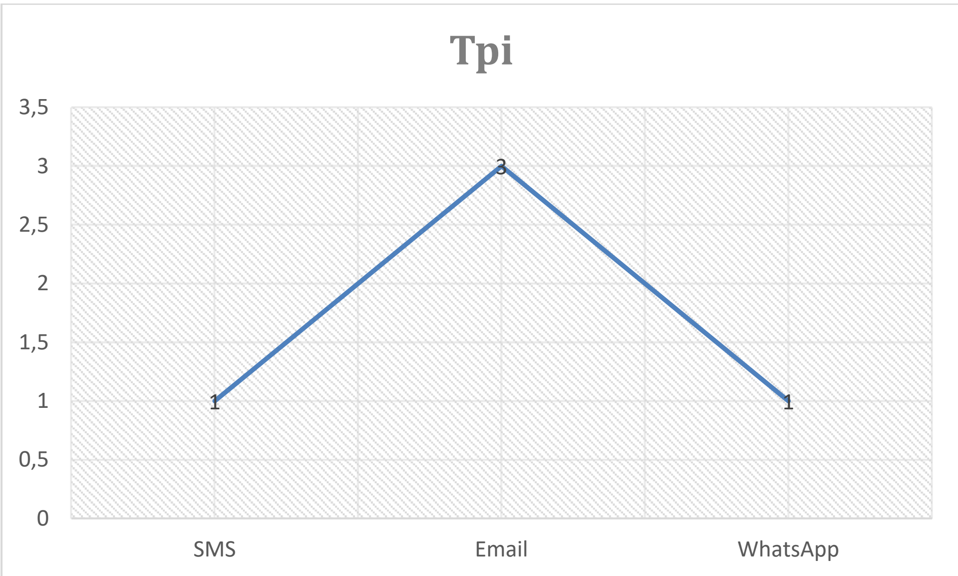


Figure 5.9: TOW calculated Tpi for email Channel

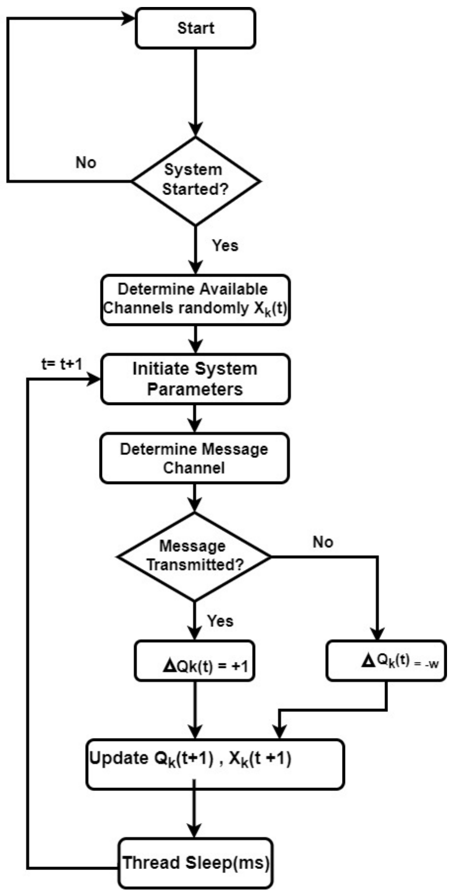


Figure 5.10: ML-enabled System Flowchart

The class diagrams in Listing 5.11 and Listing 5.12 describe each method and their responsibilities.

TOWAlgorithm
#Methods +getChannels()
#Methods + configureChannel (strChannelId: List<>) + returnChannel(strChannel: String)

Listing 5.11 Class Diagram of TOW Algorithm

RunTowdynamics()
#Attributes listChannel: List<> stringCustomerId: String stringToDate: Date boolStatus : bool stringChannelAvailability: int
#Methods + configureChannel (strChannelId: List<>) + returnChannel(strChannel: String)

Listing 5.12 Class Diagram of SendMessageClass

The implementation code snippet and logic for the TOW or UCB framework is show in Figure 5.11.

Startup.cs

Startup ▸ Excecute(IApplicationBuilder RunTowDynamics)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Builder;
using System.Buffers;
using System.Collections;
using System.Numerics
using System.IO;

namespace Application
{
    public class Startup
    {
        public void Excecute(IApplicationBuilder RunTowDynamics)
        {

            int maxTime = int.MaxValue;
            int wakeTime = 1;
            int thread = 1000;
            List<int> ChannelStatus = 0
            int w = 1;

            while (wakeTime <= maxTime)
            {
               List<Channels> Determine(CallRandom(string channel));

               IntiateSystemParams();

                try
                {
                    while (Channels.Isaccessible)
                    {

                        Sleep(thread);

                        PreformChannelChecks();

                        if (IsChannel.Idle)
                        {

                            bool messageStatus SendMessage();
                            EvaluateChannelStatus();

                            if (messageStatus)
                            {
                                //Message sent via Channel Successfully

                                ChannelStatus = ChannelStatus + 1;
                            }
                            else
```

Figure 5.11: TOW or UCB Channel Selection Implementation Logic

The code snippet for the frontend design of the application is show in Figure 5.12:

```
import { useState, useEffect } from "react";
import Wrapper from "../assets/wrappers/RegisterPage";
import { FormRow, Logo, Alert } from "../components";

import { useAppContext } from "../context/appContext";

const initialState = {
  name: "",
  email: "",
  password: "",
  facebook: "",
  twitter:"",
  sms:"",
  email:"",
  telegram:"",
  phone:"",
  accountNumber:"",
 };


function MCMRegister() {
  const [values, setValues] = useState(initialState);

  const { isLoading, showAlert } = useAppContext();

  const handleChange = (e) => {
    console.log(e.target);
  };

  const onSubmit = (e) => {
    e.preventDefault();
    console.log(e.target);
  };

  const toggleMember = () => {
    setValues({ ...values, isMember: !values.isMember });
  };

  return (
    <Wrapper className="full-page">
      <form className="form" onSubmit={onSubmit}>
        <Logo />

        <h3>{values.isMember ? "Login" : "Subscribe"}</h3>
        {showAlert && <Alert />}

        {/* Account Number field */}

        <FormRow
          type="number"
          name="accountNumber"
          value={values.accountNumber}
```

Figure 5.12: Frontend ReactJS Code for Registration and Subscription ML-enabled MCM Platform

5.5.3 MCM Framework Channel Security Module

The proposed ML-enabled MCM system was designed to be a secured system considering the base of usage by FSIs. Security is critical and essential for FSI customers as any breach within the system can cause reputational issues for FSI, and massive loss of personal and financial information for customers; hence the need to provide a level of guarantee for messages being produced and transmitted on the platform in a secured and efficient manner. According to a report by ACSC (2016), high cybersecurity crimes and incidents have been increasing steadily over the years and the adoption of a preventive measures to curb the issue is preferable than impending consequences for the customers. In view of this, a security module was built into the design to cater for and provide assurances that messages generated and sent to the customers are originated from the ML-enabled platform. The current design leverages on HTTPS or SSL with public or private key security implementation for communication between the users and the user interface platforms. The communication between third party applications and the Web API module uses JSON Web bearer tokens. Furthermore, once a message is ready to be delivered and a channel is chosen, the system follows the security implementation of each channel selected to deliver the message securely to the customer. The channel security module verifies the JSON bearer token in the message header before message transmission. In any case, if the bearer token is invalid or has been tampered with, the message will be dropped, and the system administrator is notified of a service breach. The Table 5.5 shows the security protocol implementation for each channel.

Table 5.5: Channel Security Protocol

S or No	Channel	Security Implementation	Bearer Token Support
1.	SMS		Yes
2.	Email		Yes
3.	WhatsApp		Yes
4	Telegram		Yes
5.	Facebook		Yes
6	Twitter DM		Yes

5.5.4 Database Layer and Architecture for the ML-Enabled MCM Framework

The ML-enabled MCM customer transaction alert system uses MySQL database to store customer channel provision data, channel management configuration and channel state. The main entities in the database management system are customers, channels, channel state, messages, fsiusers, fsirole and logs. A customer can subscribe to multiple channels and transact to produce multiple messages on the platform. A customer can have multiple account or registrations; hence, the customer identity number is the unique field in the customer entity. The channel entity persists the channel supported in the platform with a unique code representing an identity for each channel and their description. The channel state entity persists the channel state values after each run of the TOW dynamics algorithm storing the values for the reward or losses for each channel. All message producers registered in the configuration create and persist messages into the messages table with a unique identity. The status of the transmission of each message is also stored for reference and traceability maintenance within the system. Lastly, the logs entity stores the application performance and states generated from the standard logger class. The log values stored includes states such as *INFO* for information-based logs which contain general performance, operation and health of the system, and *ERROR* for errors encountered during the operation of the system. The information about the *ERROR* states allows further investigation into the behaviour of the system by the designer to be used to further evaluate and improve the overall system design in line with the DSR methodology. The relationship amongst the database objects is illustrated in Figure 5.13.

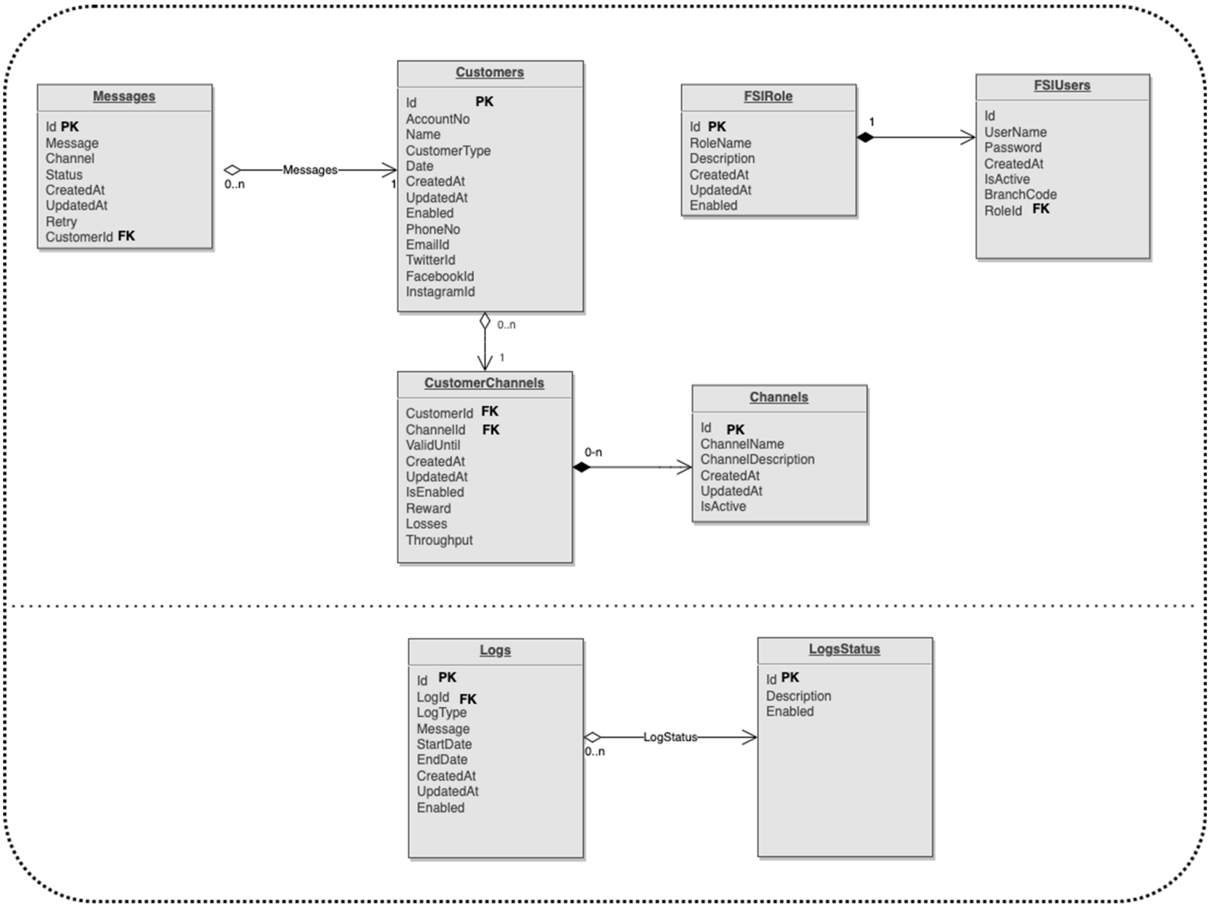


Figure 5.13: Entity Relationship Diagram Between Tables used by ML-Enabled MCM Framework

When the ML-enabled MCM framework is initiated for the first time, the database is started and thereafter seeded with pre-set data for the set of channels supported on the platform (see Figure 5.14).

```sql
---- Seed Create for Channels Table ------
INSERT INTO ML_ENABLED_MCM.Channels (Id,ChannelName,ChannelDescription,CreatedAt,UpdatedAt,isActive)
VALUES ('1','SMS','Short Messaging Service','2022-04-02 01:01:00','2022-04-02 01:01:00','true');

INSERT INTO ML_ENABLED_MCM.Channels (Id,ChannelName,ChannelDescription,CreatedAt,UpdatedAt,isActive)
VALUES ('2','Email','Email Messaging Service','2022-04-02 01:01:00','2022-04-02 01:01:00','true');

INSERT INTO ML_ENABLED_MCM.Channels (Id,ChannelName,ChannelDescription,CreatedAt,UpdatedAt,isActive)
VALUES ('3','Twitter','Twitter Direct Messaging','2022-04-02 01:01:00','2022-04-02 01:01:00','true');

INSERT INTO ML_ENABLED_MCM.Channels (Id,ChannelName,ChannelDescription,CreatedAt,UpdatedAt,isActive)
VALUES ('4','Facebook','Facebook Direct Messaging','2022-04-02 01:01:00','2022-04-02 01:01:00','true');

INSERT INTO ML_ENABLED_MCM.Channels (Id,ChannelName,ChannelDescription,CreatedAt,UpdatedAt,isActive)
VALUES ('5','Instagram','Instagram Direct Messaging','2022-04-02 01:01:00','2022-04-02 01:01:00','true');

INSERT INTO ML_ENABLED_MCM.Channels (Id,ChannelName,ChannelDescription,CreatedAt,UpdatedAt,isActive)
VALUES ('6','Whatsapp','Whatsapp Instant Messaging','2022-04-02 01:01:00','2022-04-02 01:01:00','true');

INSERT INTO ML_ENABLED_MCM.Channels (Id,ChannelName,ChannelDescription,CreatedAt,UpdatedAt,isActive)
VALUES ('7','Telegam','Telegram Instant Messaging','2022-04-02 01:01:00','2022-04-02 01:01:00','true');



---- Seed Create for FSIRole Table ------
INSERT INTO ML_ENABLED_MCM.FSIRole (Id,RoleName,Description,CreatedAt,UpdatedAt,Enabled)
VALUES ('1','Initiator','Creator','2022-04-02 01:01:00','2022-04-02 01:01:00','true');

INSERT INTO ML_ENABLED_MCM.FSIRole (Id,RoleName,Description,CreatedAt,UpdatedAt,Enabled)
VALUES ('2','Verifier','Supervisor','2022-04-02 01:01:00','2022-04-02 01:01:00','true');

INSERT INTO ML_ENABLED_MCM.FSIRole (Id,RoleName,Description,CreatedAt,UpdatedAt,Enabled)
VALUES ('3','Approver','Approver','2022-04-02 01:01:00','2022-04-02 01:01:00','true');

INSERT INTO ML_ENABLED_MCM.FSIRole (Id,RoleName,Description,CreatedAt,UpdatedAt,Enabled)
VALUES ('4','Inquiry','General Inquiry','2022-04-02 01:01:00','2022-04-02 01:01:00','true');
```

Figure 5.14: Seed data for initialising the database

Once the database is seeded with data. The FSI users can log in to provision a customer. The customer can also use alternate web channels via web API integration to provision the account by subscribing to at least two messaging channels presented from the list of the channels in the database table. The channel configurations provisioned are stored in the customer Channels table with their status and the reward or loss values updated subsequently when the TOW algorithm is executed. A typical use case will be when an FSI customer initiates a transaction from their account and a message is generated. These message are stored in the message table and the ML-enabled MCM logic would subsequently come into force to deliver the message to the customer through the channels profiled for the customer. After the TOW dynamics logic is executed, the MCM system selects the channel that is most reliable while considering the reward or loss table fields and thereafter select the channel that has greater reward at that instant of time. The code depicted in Figure 5.15 is used to retrieve available channels at the point of customer provisioning and subsequent calls to retrieve the channels available.

```
#############--- Code Segment ----#####################

get_Channels_Query = "SELECT a.channelName, b.customerId FROM CustomerChannels a, Channels b WHERE b.customerId = {$customerId} and a.IsEnabled=true";

# Get list of customerChannels
        private CustomerChannels getCustomerChannels()
        {
            SqlCommandBuilder sqlCommandBuilder = null;
            SqlDataAdapter sqlDataAdapter = null;
            SqlConnection sqlConnection = new SqlConnection();
            DataSet dataSet = new DataSet();

            try
                {
                    SqlConnection sqlConnection1 = sqlConnection;
                    sqlConnection1.ConnectionString = utils.AlertDTCnnString;
                    sqlConnection1.Open();
                    sqlConnection1 = null;
                    sqlDataAdapter = new SqlDataAdapter(utils.RunQuery, sqlConnection);
                    sqlCommandBuilder = new SqlCommandBuilder(sqlDataAdapter);
                    sqlDataAdapter.Fill(dataSet, "MCMCustomerChannels");

                    CustomerChannels customerchannel = new CustomerChannels();

                    for (long i = num3; i <= count; i++)
                    {
                        CustomerChannels.ChannelId = dataSet.Tables["MCMCustomerChannels"].Rows[checked((int)i)]["channelName"].ToString();
                        CustomerChannels.customerId = dataSet.Tables["MCMCustomerChannels"].Rows[checked((int)i)]["customerId"].ToString();
                    }

                }
            catch (Exception ex)
                {
                    Log.Error(ex.Message + " " + "SQL Server Error");
                }
            }
            finally
            {
                sqlConnection.Close();
                sqlConnection = null;
                sqlDataAdapter.Dispose();
                sqlDataAdapter = null;
                dataSet.Dispose();
                dataSet = null;
                sqlCommandBuilder.Dispose();
                sqlCommandBuilder = null;
            }
            return CustomerChannels;
        }
```

Figure 5.15: SQL Connection And Code To Get Channels List

5.6 SOFTWARE TECHNOLOGY USED FOR DEVELOPMENT

Advancements in technology and software development has led to the availability of several choice high level programming language choices such as Java, JavaScript, Python, C# .Net Core, Visual Basic, R, Scala, Go, Ruby, SQL. Each of these programming languages has their own strengths and weaknesses depending on the intent of use and applicability. For the ML-enabled MCM system customer, the researcher chose C# .NET core due to its powerful capabilities in implementing mathematical algorithms and ML libraries, relative ease of integration with popular databases such as MySQL, and rich support for REACT JS (a popular JavaScript frontend web interface program). Furthermore, C# .NET core has native support for web API's and native nugget packages with extensive support from integration and logging. C# .NET core is a full object-oriented programming language and the researcher's experience on working with multiple projects stacks turned out to be useful for using the features provided by C#. The ML-enabled MCM framework uses MySQL database, a robust open-source database management system with great capabilities for implementing enterprise-grade systems used by major corporations around the world.

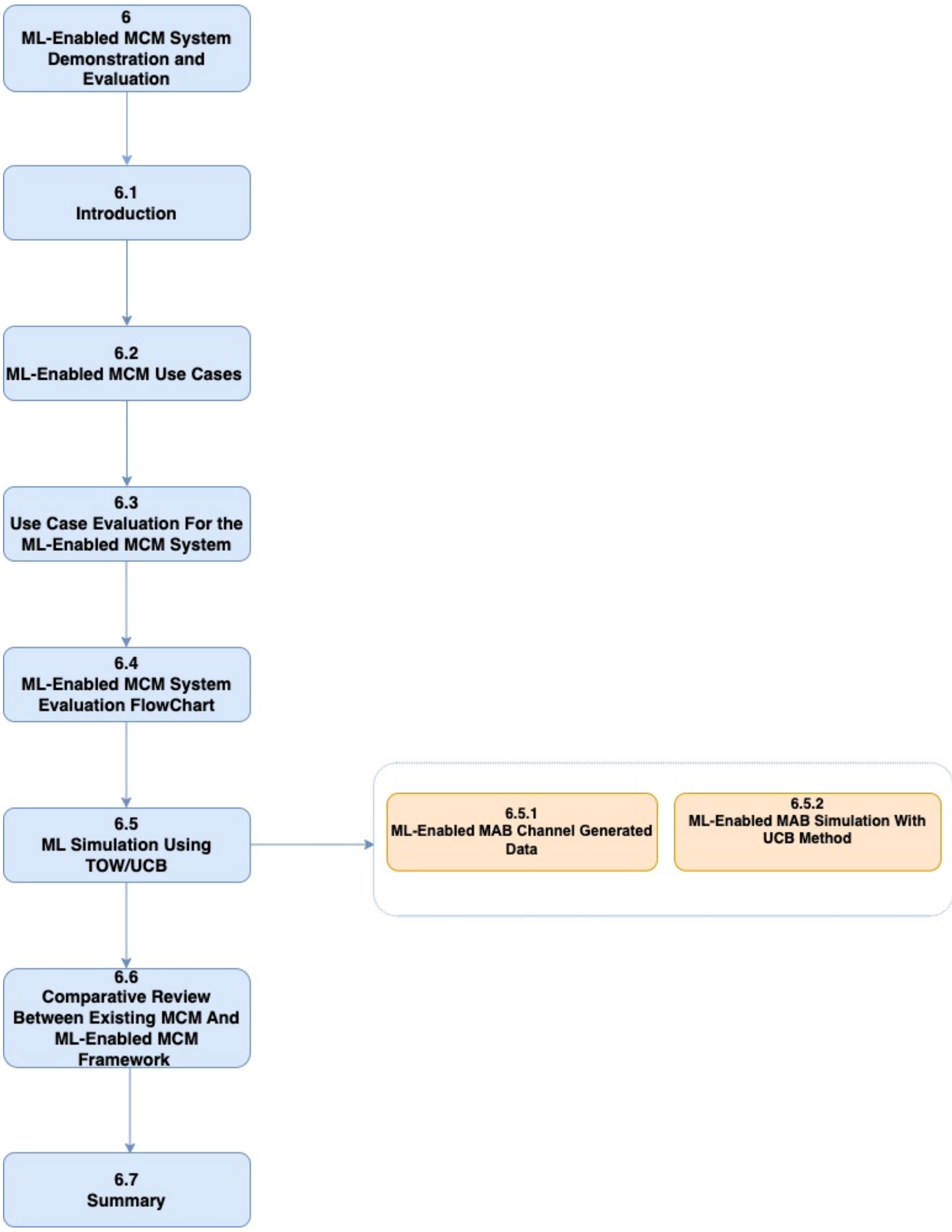
5.7 SUMMARY

This chapter presented a review of the current features in the existing MCM systems. Subsequently, a review of publicly deployed MCM systems hosted on mobile application stores was conducted. Following the Design Science Methodology approach (Peffers et al., 2007), this chapter aided the definition of the objectives to be fulfilled by the Machine Learning Enable Multi-Channel Messaging System. The review of the existing MCM systems hosted on public application stores provide the researcher with insights on what the current systems offer. The review conducted while the customer was registering on the Kuda MFB application with SMS and Email messaging channels revealed gaps. These gaps are as a result of the using pre-selected messaging channels in the MCM system which form the requirements for the ML-enabled MCM framework. Further reviews on similar mobile and web application conducted also served to perfect the elicitation of the requirements. Delivering a working artefact for the ML-enabled MCM system requires agility in design, and is evaluated in line with DSR principles (Hevner et al., 2004). These principles include continuous improvement based on feedback into the design, progress measurement and solution refinement.

Furthermore, the background for the development and integration of the ML-enabled MCM system was discussed extensively. As earlier stated, Salami and Mnkandla (2020, 2021) have written about the approach for the ML-enabled MCM system as applied in a customer alert system used by FSIs and have proposed creating a prototype-based solution. The chapter laid out the architectural design principles used, and the choice of technology selected for the implementation which covers the user interface layer, the application layer, and the database layer in line with the key development layers for a modern application. The chapter also detailed the components of the user interface layer, the alert application, and the support for integrating the disparate messaging channels such as WhatsApp, SMS, email, Twitter DM, Facebook DM and Instagram DM via a message service bus. The design of the solution was presented with class diagrams, sequence diagrams with the application database ERM structure including the scripts. The TOW algorithm was also implemented using C# .NET core code with enhancement on the channel selection management and selection features.

Finally, the chapter was concluded with the choice of C# .NET core used as the programming language for the implementation of the solution due to its native support for ML, web front-end application and back-end implementation. The next chapter discusses the demonstration and evaluation of the designed artefact.

6 CHAPTER 6 ML-ENABLED MCM SYSTEM DEMONSTRATION AND EVALUATION



6.1 INTRODUCTION

In Chapter 5, the ML-enabled MCM system artefact was designed and developed. The DSR methodology was adopted for the design and implementation artefact. Subsequently, a workable artefact was produced using C# .Net core programming language. The application and webservice layer provided a platform for the implementation of the TOW dynamics algorithm, in addition the web framework for the application presentation layer was developed using REACTJS and MySQL database powered the backend of the solution see Section 5.5. Additionally, in Section 5.5, the architecture underpinning the implementation of the ML-enabled MCM system was presented. This architecture was derived from the existing MCM system knowledge base and requirements elicited to enhance the framework with ML using the TOW algorithm.

This chapter presents the demonstration and evaluation of the ML-enabled system to determine it applicability and extent of use by FSIs. The ML-enabled MCM system was designed to be implementable and resource conservative in terms of efforts needed to develop the proposed architecture in any of the high-level programming languages, including a flexible choice of database as preferred by the implementation or development team. Overall, the ML-enabled MCM artefact is meant to revolutionize messaging systems used by FSIs by providing alternative and reliable services to their customers. This phase of demonstration and evaluation of the artefact is meant to provide an assurance on the utility of the system. The main objective is to establish and fulfil the aim of this research work through the development of an integrated platform that allows seamless integration of disparate messaging channels using dynamic channel selection and assignment in a customer alert messaging system used by FSIs.

In line with the DSR methodology adopted in this study, both the demonstration and evaluation process are conducted concurrently. Owing to the practicality of the adoption of the proposed artefact for use by FSIs, the demonstration and evaluation of the system adopted the following approach: examining how well the artefact functions in relation to the objectives it was designed to achieve (simulation), a use case evaluation method was performed to evaluate the artefact's usefulness and effectiveness based on the investigation by Jacobson et al. (2016). The use case evaluation (UCE) is consistent with the collection of qualitative data. Instead of performing full-scale experimental user evaluations, it has been known to enable researchers to uncover faults associated with the system by gathering data on the system's reaction to changes while tweaking the artefacts parameters (Bumblauskas et al., 2020;

Ellram, 1996; Hansen et al., 2014; Hornbæk et al., 2007). In this study, both methods of evaluation were used to ensure the validation of the intended objectives of designing and implementing the solution.

Evaluation is typically an iterative process that begins with the artefact's design stage. Mental evaluations of the components occur while the researcher considers which components to combine to make the artefact (Vaishnavi & Kuechler, 2004). Evaluation and validations were performed on each iteration to validate the design process.

The initial rigorous evaluation of the ML-enabled MCM system was conducted in Section 2.5, where the various algorithms for the implementation of ML and channel selection techniques were discussed. In Subsection 2.5.6, the TOW algorithm used for the channel selection and route management was elicited to lay the foundation for the implementation of the MAB approach using Reinforcement Learning. This was done because of the uncertainty in surrounding the selection of a channel at a time for message translation. The reward value defined as the Upper Confidence Bound (UCB) value is determined by the channel with the highest T_{pi} which determines if a channel will be selected at a time see Section 3.3.

The final rigorous evaluation of the ML-enabled MCM was detailed in Section 6.3 and subsequent sections whereby the requirements for the design of the artefact were elicited and reviewed to propose the architecture for the design. Only those requirements which fully satisfied the research objectives were adopted and incorporated in the design. The channel reward system was based on the TOW or MAB dynamics approach. As recommended by Hasegawa et al. (2020) and Kim et al. (2010a), the value 1 was adopted as the acceptable value for a channel reward that translates into a successful transmission of a message; by the same token, -w or 0 expresses a loss when the channel is busy or not available to process or transmit messages.

Overall, the use case evaluation approach determined from the elicited requirements and demonstration of use via simulation of the MAB or UCB model approach using Python language was adopted to validate the utility of the designed artefact. The Python programming language was selected by the researcher for the simulation owing to its robust and extensive capabilities for data visualization, analysis and support for ML. The math, histogram, graph plotting libraries enabled the implementation of the MAB or UCB algorithm using native functions. In addition, Python APIs for ML are readily available for third party applications to integrate effortlessly.

6.2 ML-ENABLED MCM SYSTEM USE CASES

A use case is a breakdown of every feasible user interaction with a platform, gadget, or software artefact (Bumblauskas et al., 2020). Use cases explain how a system's design respond to demands from its end-user or actor. These actors could be human beings or other systems capable of interacting or providing input into the system (Jacobson et al., 2016). Use cases help to structure every part of the various functional requirements and is also used to determine the scope of the project. Each use case is specified using the details below (Bumblauskas et al., 2020):

- The goal of each use case
- Determining whether the actor is a human or another system
- Preconditions, or the state the system needs to be in for the use case to occur
- The regular series of steps the system will take; also called normal flow of events
- Alternative paths the system could take to behave in a certain way
- Postconditions or actions the system takes at the end of the use case or the various states the system could be in after the use case concludes

Furthermore, use case evaluation provides the following benefits:

- **Summary and Planning Skeleton for Evaluation**: Use cases provide an overview of what the developed artefact will deliver to all stakeholders. Use cases provide a planning framework to assist teams in prioritising, estimating timing, and carrying out predefined actions.
- **Requirements Context**: Use case provides detailed context about the requirements used for the design to ensure stakeholders are aligned on the output of the design, that is, what the system will and will not do.
- **Addresses Specific Design Issues**: Use cases provide answers to peculiar problems that system designers may have in evaluating a system performance. The use case method guarantees that all queries about concerns or potential scenarios are handled seamlessly.

The use case evaluation for the implementation of the ML-enabled MCM system is shown in Figure 6.1. The recognised role actors are depicted outside the system border, which is represented by the large solid-line rectangle. The use cases identified for each actor are represented with an ellipse and arrows showing directions and relationships between each use case.

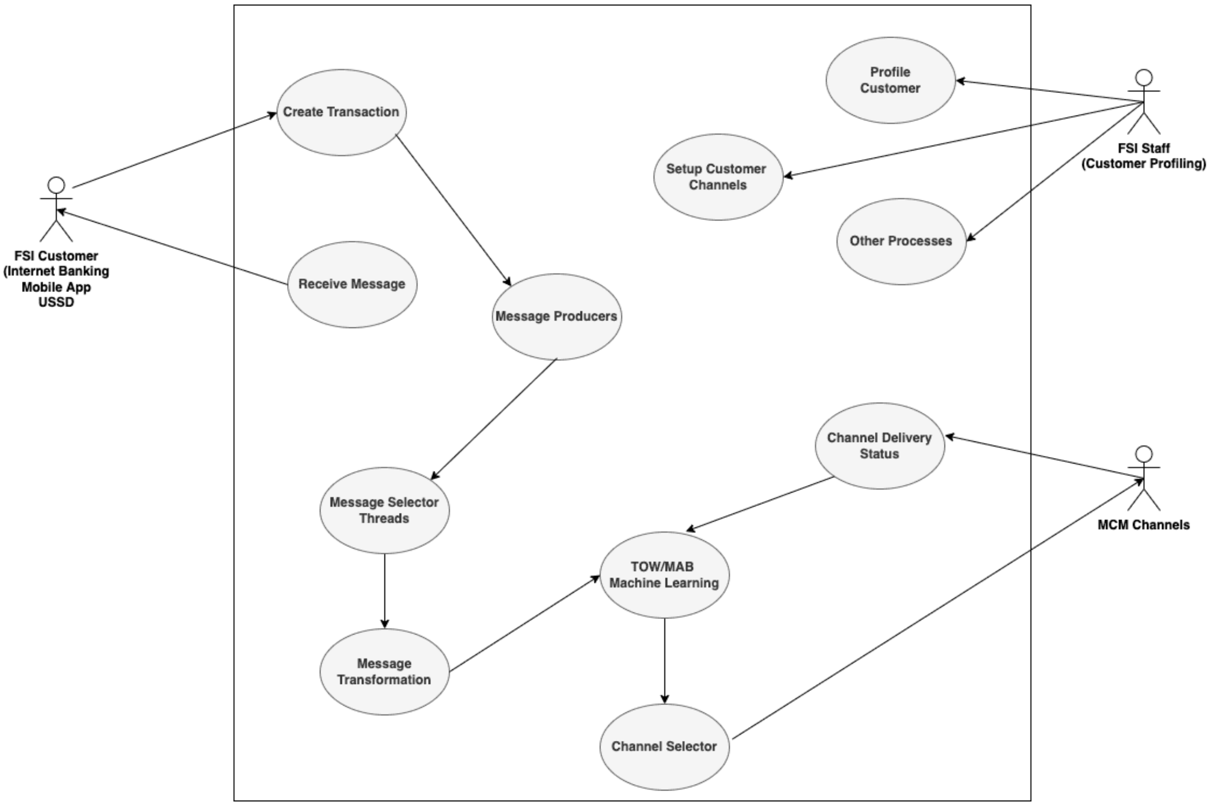


Figure 6.1: ML-enabled MCM Evaluation Use Case Diagram

Based on the complex interactions that occur within the ML-enabled MCM system boundary, each actor is expressed outside the system. In this case, the FSI customer initiates a transaction and receives a transaction alert message from any of the applications available for use. Preconditions to this event are that the FSI Staff must have used the customer alert profiling system to set up the customer and messaging channels available such as WhatsApp, SMS, email, Telegram, Twitter DM and Facebook Messenger. Each message is handled by a message selector thread and passed over to the message transformation module to be converted into an agnostic data format JavaScript Object Notation (JSON). This is to ensure that the message can be easily translated for use by each integrated channel. The channel selector module is integrated into the TOW or MAB ML-algorithm to check and select the channel available for delivery from the bouquet of integrated channels. TOW or MAB ML-algorithm computes the reward for each channel based on the current and past rewards expressed as the channel with the highest upper confidence value. The message is then routed to the selected channel, which confirms the delivery to the FSI customer via the customer's mobile phone terminal. A channel delivery report of the delivery status, which is either successful (increments) or unsuccessful (decrements) expresses the reward for the channel respectfully.

6.3 USE CASE EVALUATION FOR THE ML-ENABLED MCM SYSTEM

To evaluate the critical features of the ML-enabled MCM system, the study focused on two main use cases to evaluate the artefact, namely: "**FSI User Profile Customer on ML-Enabled MCM Alert Application**" and "**User Requests for OTP Pin to Complete a Transaction**". These identified use cases enabled the research to focus on the critical parts of the ML-enabled MCM system, that is, channel prioritization, assignment, ML, message delivery and customer profiling. The use cases 3 main use cases are described in Table 6.1 and Table 6.2 and Figure 6.2 and Figure 6.3 .

Table 6.1: Profile Customer on ML-Enabled MCM Alert Application

Use Case: Profile Customer on ML-Enabled Alert Application	
Role players	FSI Staff, ML-Enabled MCM System, MCM channels
Pre-conditions	1. Interface to access message delivery channels is available. 2. ML-Enabled MCM alert application is available. 3. Customer to be profiled has an account with the FSI. 4. Connectivity to channels and network are available and optimal. 5. FSI staff has a profile for setup on the MCM alert platform.
Flow of Events	1. FSI staff logs on to the ML-Enabled MCM application. 2. Search or select a customer account to set up. 3. Choose from a list of the message delivery channels for message delivery, as specified by the customer on the alert application document or form, submits the request. 4. If the staff receives a "Failed setup" response from the alert application: a. Staff logs out the application. b. Staff has the option to retry the application. 5. Staff receives a "Successful registration" response. 6. Profiled customer record is shown on the application dashboard.
Post-conditions	1. The staff has successfully completed the setup of the customer profile.

	2. The ML-enabled MCM alert system has updated its records to reflect that the user has been successfully set up.

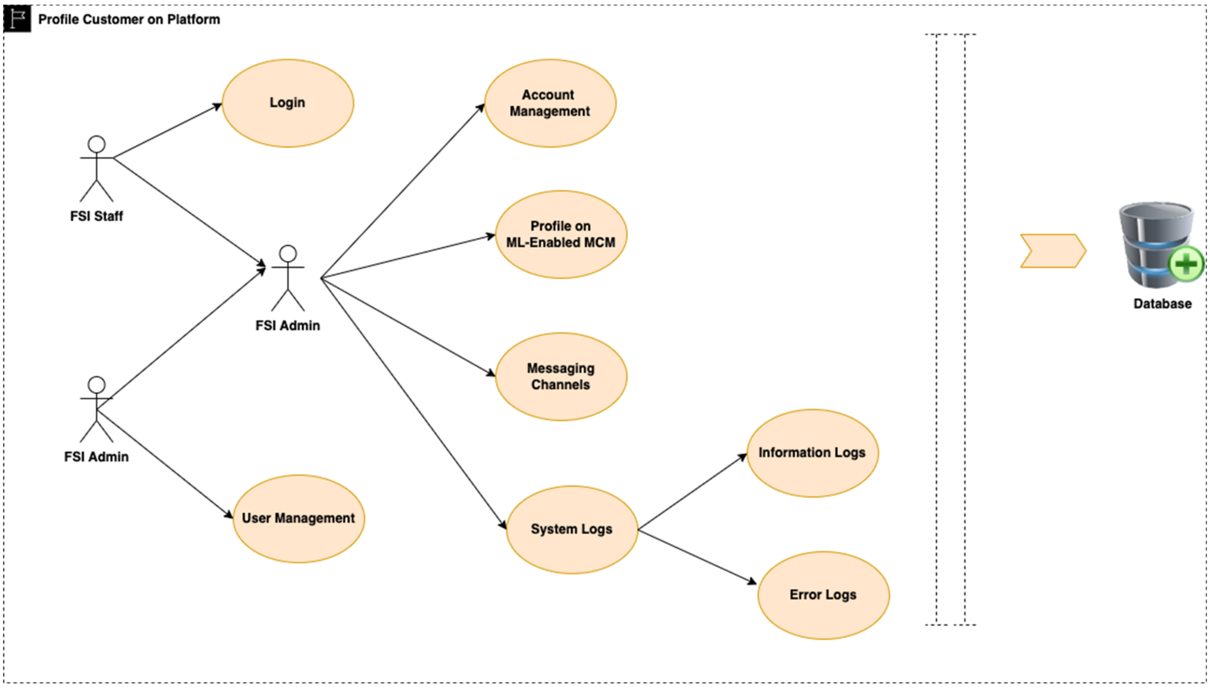


Figure 6.2: Customer Profiling Use Case

Table 6.2: Customer Requests for OTP PIN to Complete a Transaction

Use Case: Customer Request OTP Pin	
Role players	Customer, OTP System, ML-Enabled MCM system, MCM channels
Pre-conditions	1. OTP service is available. 2. Customer has access to OTP platform. 3. ML-Enabled MCM alert application is available. 4. TOW or MAB algorithm service is running. 5. Customer has been profiled on the MCM alert application by FSI Staff.
Flow of Events	1. Customer logs on to OTP platform. 2. Customer requests for OTP from Service. 3. If the Customer received a "Failed OTP request" response from the application:

	a. User exits the application. b. Customer can retry the application. 4. Customer receives an “OTP Response” on any of his preferred channels determined by TOW or MAB algorithm.
Post-conditions	1. The customer has successfully completed the process of requesting for OTP. 2. The customer receives a message on the channel auto determined as available by the TOW or MAB machine learning algorithm 3. The ML-Enabled MCM alert system updated its records to reflect that the OTP has been delivered successfully.

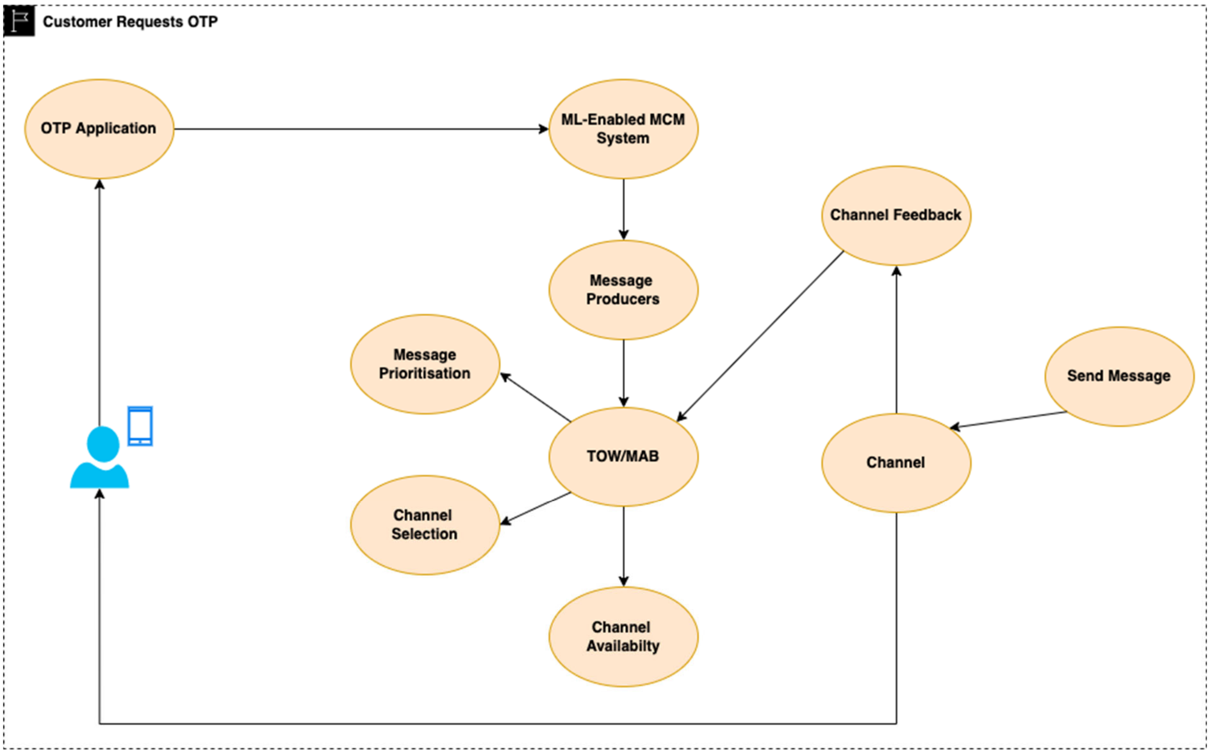


Figure 6.3: Customer Requests OTP Use Case

6.4 ML-ENABLED MCM SYSTEM EVALUATION FLOWCHART

To evaluate the use case that has the impact on the operation of the ML-enabled MCM system, the use case “**User Requests for OTP Pin to Complete a Transaction**” was derived to test the critical path of execution using an activity diagram as shown in Figure 6.4

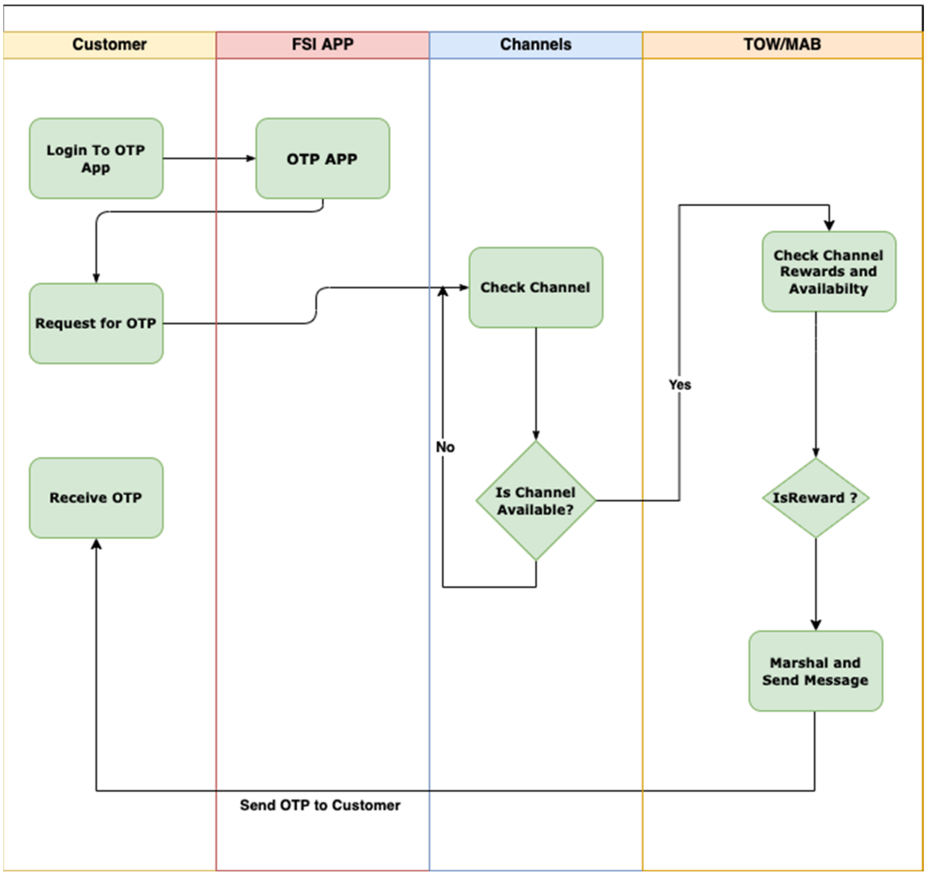


Figure 6.4: Activity Diagram for User Requesting for OTP Use Case

The customer logs on to the FSI OTP application platform and selects the option to request for an OTP to complete a transaction securely. The OTP application validates the customer's request and generates a secure value to be sent to the customer using any of the registered messaging channels. The secured value is processed by an available message producer which converts the message into data agnostic format (JSON). The channel selection module loads and retrieves all available messaging channels available or profiled for the customer and sends a request to the ML module to check the availability and reliability of the channels using TOW or MAB ML-algorithm. The channel with highest reward (i.e., highest UCB value) is selected to transmit the message. The converted JSON-formatted-secured message is marshalled using a serializer to send as an input into the selected channel. Once the message is delivered successfully to the customer's terminal, the TOW or MAB algorithm is executed to update the reward and loss status of all the channel.

Assumptions for the use case evaluation:

- Customer terminal is available and network connectivity is available to receive messages
- TOW or MAB algorithm returned SMS channel
- FSI OTP platform is available
- The customer has been profiled on the platform by FSI staff
- The ML-enabled MCM platform is available and running.

Figure 6.5 shows the state of the application, message delivery and the database before and after values for the channels at an instance of time.

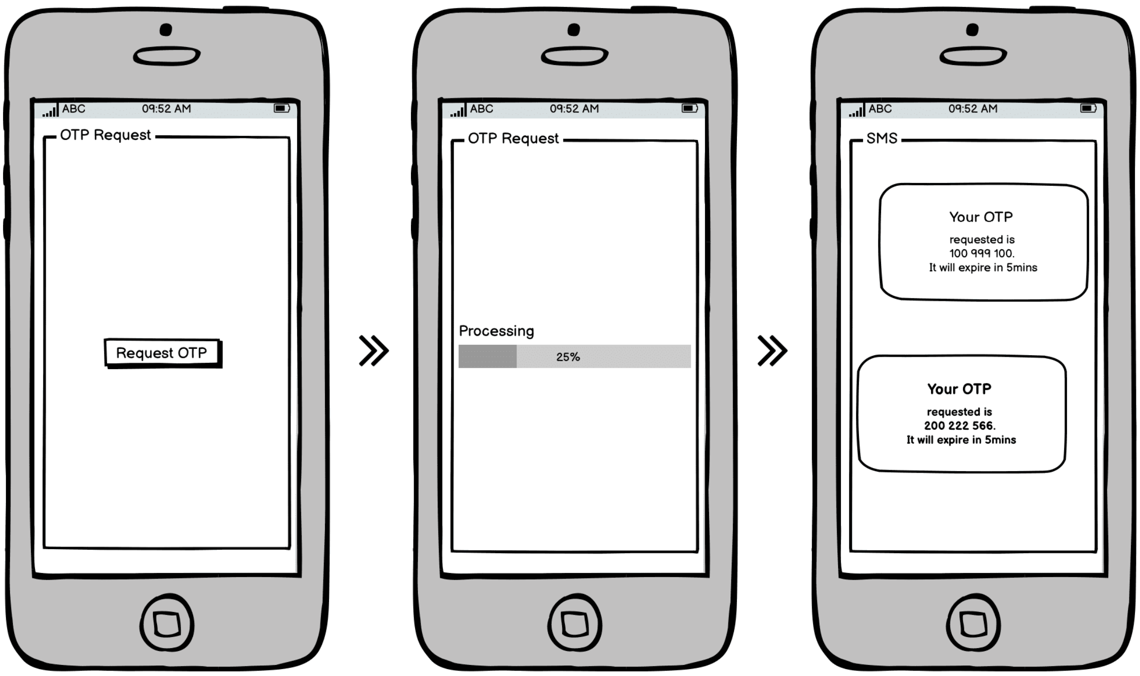


Figure 6.5: Customer Requests For OTP

6.5 MAB SIMULATION USING TOW OR UCB FOR THE ML-ENABLED MCM SYSTEM

Reinforcement learning algorithm was discussed in Section 2.6. In addition to this, Reinforcement Learning application in line with the MAB principles was examined in Section 3.2 with focus on its implementation using the TOW algorithm. The artefact or decision-maker in Reinforcement Learning generates training and subsequent learning data by directly interfacing with the world. Rather of being explicitly instructed on acceptable behaviour, the artefact must understand the consequences of its choices by trial and error (François-Lavet et al., 2018).

Decision making in Reinforcement Learning is well expressed using the MAB Problem due to uncertainty expressed by k-armed bandits. In the MAB Problem, an artefact or decision maker must select one of k-possible actions and be rewarded accordingly. Basic Reinforcement Learning ideas including rewards, timesteps, and values in line with MAB principles.

To simulate the ML enabled MCM system where the system at an instant of time chooses the channel that is available to transmit a message to the customer using both the exploitation and exploration approach, the ML-enabled MCM channel selection logic either has to stick to a channel from the start which may or not be available for message delivery (random selection) or rely on a common strategy referred to as the UCB (Zhang et al., 2019). The UCB method (see Section 0) enables the channel selection according to its potentials for delivery using the calculated upper confidence interval value while balancing this out with how uncertain its measurement is. The UCB formula is expressed below in Eq. 6.1:

$$A_t \quad = \arg\max a \left[Q_t\,(a) + \sqrt[c]{\frac{log t}{N_t}} \right]$$

Eq. 6.1

where:

- t = is the time (or channel selection round) we are currently at
- a = the action or channel selected by the ML-enabled MCM system (in this case the channel chosen to transmit the message to the customer)
- N_t = the number of times an action was chosen prior to the time *t*
- $Q_t(a)$ = average reward of action or channel *a* prior to the time *t*

- c = control number that controls the degree of exploration, c must be a number greater than 0
- Log t = the natural logarithm of time *t*

The square root in the algorithm indicates the uncertainty of choosing a channel at an instant of time *t* by choosing an action *a*. The likelihood that an action would be taken increases if it is not chosen because its uncertainty increases as *t* increases N_t (exploration). Both numerators increase if an action is selected. Due to the natural *log*, the numerators decrease over time, while the denominator stays constant, thus reducing the uncertainty. At each round, the channel with the highest UCB is chosen. To simulate this, the researcher has used the Python framework to implement and demonstrate this.

6.5.1 ML-Enabled MAB Channel Generated Data

To implement this case of ML-enabled MCM system using the MAB algorithm and UCB metho, the Python framework was used to demonstrate this Reinforcement Learning problem. The Python library frameworks (i.e., math, random, seaborn, pandas, NumPy, matplotlib) were imported to facilitate the execution and implementation of the TOW or UCB algorithm. The code was executed using the Anaconda Integrated Development Environment hosting the Jupyter notebook containing the codes (see Figure 6.6)

Import the Necessary Library

```
In [4]: import math
        import random
        import seaborn as sns
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        plt.style.use('seaborn-talk')
        %matplotlib inline
```

Figure 6.6: Anaconda IDE And Import Of The Necessary Python Library

The next step is to create the 6-column dataset. The first column of the dataset contains the customer or recipient of the messages, followed by the channel (such as Email, Facebook, Instagram, SMS and Telegram). 5 channels were implemented just for this simulation and can be varied as needed. The value represents the rewards for each channel and user combination in the dataset. Each dataset has a mean value between 95 and 105 and a standard deviation of 5 or 10, which corresponds to a normal distribution. Twenty-thousand samples for each channel were produced for this simulation. The NumPy random normal *(np.random.normal)* function was used to create a NumPy array that contains the normally distributed data. The parameter is expressed as*: np.random.normal (loc, scale, size)*. The *loc* parameter is used to control the mean of the function, scale parameter is used to control the value of standard deviation of the normal distribution, and the size is used to control the size and shape of the output (see the input code and output data in Figure 6.7). For system optimisation and execution, the data is exported to the system disk for storage and retrieval of the datasets.

In [5]:

```
"""
The NumPy random normal (np.random.normal) function is used to create a NumPy
array that contains normally distributed data.
The parameter is expresses as this: np.random.normal(loc, scale, size).
the loc parameter is is used to control the mean of the function, scale parameter is used to
control the standard deviation of the normal distribution, and the size is used to control the
size and shape of the output
"""
channels = list(range(0,20000))
email = np.random.normal(100,10,20000)
facebook = np.random.normal(105,5,20000)
instagram = np.random.normal(95,10,20000)
sms = np.random.normal(100,5,20000)
telegram = np.random.normal(105,5, 20000)
dataset = pd.DataFrame({"Channels": channels,"Email":email,"Facebook":facebook,"Instagram":instagram,"SMS":sms, "Telegr
dataset = dataset.reset_index(drop=True)
dataset.head(60)
```

Out[5]:

	Channels	Email	Facebook	Instagram	SMS	Telegram
0	0	110.574124	102.380594	92.976697	108.309306	98.582928
1	1	90.275802	111.926479	90.962935	93.180035	95.854354
2	2	84.686064	106.323009	87.566436	94.508324	104.797952
3	3	105.353938	102.717723	84.421040	103.069773	108.975010
4	4	101.482113	106.201012	76.020470	98.346240	102.136331
5	5	100.923784	111.614776	85.179066	98.441410	111.542032
6	6	99.989743	105.107748	93.380854	89.260057	106.802856
7	7	84.736718	106.201186	80.947906	98.268089	105.925655
8	8	105.673985	101.955221	87.462336	97.000371	104.506118
9	9	110.191007	103.593052	92.636053	107.767240	104.674975
10	10	116.157719	108.106886	105.974730	96.220612	106.401266
11	11	103.110234	107.988328	91.959914	98.195981	105.519073
12	12	108.841443	108.358779	95.474627	100.260733	104.104740
13	13	98.023219	107.287746	98.931774	103.074437	100.926650

19975	19975	107.836711	96.580660	77.782744	95.698949	111.098866
19976	19976	98.123692	107.630534	103.733111	101.748719	107.358377
19977	19977	109.729181	101.294600	92.446435	99.804171	104.358578
19978	19978	105.874150	98.321551	98.364289	101.385007	103.264489
19979	19979	102.902144	104.240550	112.737603	95.619038	114.626994
19980	19980	106.628948	102.241580	78.042872	92.951112	104.347851
19981	19981	99.480672	109.543076	100.826749	97.477861	104.959905
19982	19982	110.885257	102.675416	87.221976	96.924972	103.083445
19983	19983	101.887863	107.263122	85.308776	96.972216	113.568577
19984	19984	102.251585	106.377190	112.141571	93.712513	111.801931
19985	19985	94.721816	102.431367	99.919010	98.258485	108.006845
19986	19986	97.810984	102.629825	107.888347	92.831715	100.845008
19987	19987	96.364687	102.940484	89.659878	104.626953	98.833776
19988	19988	94.412906	104.303374	90.245159	97.151660	106.894360
19989	19989	95.429087	106.876362	84.861047	91.042072	104.000350
19990	19990	101.825774	99.579510	87.569138	97.194641	100.491155
19991	19991	95.454588	98.120506	113.313040	83.461868	103.304825
19992	19992	94.202387	112.496089	83.860304	106.172663	108.664611
19993	19993	104.934371	101.520825	97.243219	107.945002	102.033578
19994	19994	93.098697	96.392554	90.267226	98.363755	115.207023
19995	19995	113.493988	101.203141	93.481492	101.679008	112.753841
19996	19996	104.394990	108.832846	104.439012	104.330933	99.893448
19997	19997	100.918580	110.928106	105.714075	93.941088	110.718275
19998	19998	103.402967	105.562424	100.605577	97.464669	109.322138
19999	19999	102.710870	108.312164	90.912022	101.808338	100.342120

```
In [30]: dataset.to_csv('/Users/olusolasalami/Documents/PhD/datasets_for_Channels/dataset.csv', index=False)
```

Figure 6.7: Generate the Normal Distributed Channel Data using NumPy Function

The average reward for each channel is displayed using a dataset in a box plot shown in Figure 6.8.

```
In [18]: sns.boxplot(x="variable", y="value", data=pd.melt(dataset[['Email','Facebook','Instagram','SMS', 'Telegram']]))
         plt.title("Distribution of Rewards by Channels")

Out[18]: Text(0.5, 1.0, 'Distribution of Rewards by Channels')
```

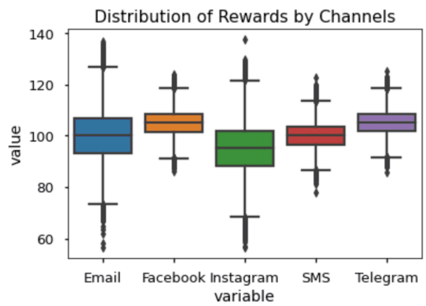


Figure 6.8: Average Reward for Each Channel

Based on the average reward plotted above for each channel, the best channel to choose to transmit the message to the customer at this instance of time is the Facebook Channel.

Subsequently, the UCB MAB method was utilised to implement and simulate what the ML-enabled artefact would choose.

6.5.2 ML-Enabled MAB Simulation with UCB Method

Simulating the approach requires going over each round of the MAB Problem using the UCB channel selection method, in the following order (i) execute an action (choose and send a message through a channel), (ii) observe the results, and (iii) pick a message again to send via the channel. Subsequently in each run, most effective channel was chosen to send the message. The UCB variables are initialised in Python to solve the problem. Variable *N* stores the round or time of execution, *d* stores the total number of messaging channels implemented (a total of 5 channels for this simulation), *Qt_a a*nd *Nt_a* are the maximizing actions initialized to 0 while *c* is initialized to 1, a value that controls the degree of exploration of each channel. The cumulative sum of rewards for a particular channel and other Python helper variables UCB_CHOSEN is used to store the cumulative rewards value, and OPTIMAL cumulative rewards and RANDONMLY store a history of randomly selected action rewards (see Figure 6.9).

```
In [11]: #Initialize the UCB Variables
N = len(dataset.index)        # stores the round or time
d = 5                   # number of channels implemented in the ML-Enabled MCM system
Qt_a = 0
Nt_a = np.zeros(d)      #number of times action a has been selected prior to time T
                        #If Nt(a) = 0, then a is considered to be a maximizing action.
c = 1                   # c is a number greater than 0 that controls the degree of exploration
sum_rewards = np.zeros(d) # this the cumulative sum of reward for a particular channel
#python helper variables to assist in performing data analysis
hist_t = [] #holds the natural log of each round
hist_achieved_rewards = [] #this variable holds the history of the UCB CHOSEN cumulative rewards
hist_best_possible_rewards = [] #this variable holds the history of OPTIMAL cumulative rewards
hist_random_choice_rewards = [] #this variable holds the history of RANDONMLY selected actions rewards
```

```
In [62]: #loop through number of rounds #t = time
for t in range(0,N):
    UCB_Values = np.zeros(d) #an array holding the ucb values. the MAX will always be selected
    action_selected = 0
    for a in range(0, d):
        if (Nt_a[a] > 0):
            ln_t = math.log(t) #natural logarithm of t
            hist_t.append(ln_t) #to plot natural log of t
            #calculate the UCB
            Qt_a = sum_rewards[a]/Nt_a[a]
            ucb_value = Qt_a + c*(ln_t/Nt_a[a])
            UCB_Values[a] = ucb_value
        #if ucb_value equals zero, choose this value as the maximum. Cant divide by negative
        elif (Nt_a[a] == 0):
            UCB_Values[a] = 1e500 #make the value large
    #select the max UCB value
    action_selected = np.argmax(UCB_Values)
    #update Values as of round t
    Nt_a[action_selected] += 1
    reward = dataset.values[t, action_selected+1]
    sum_rewards[action_selected] += reward
```

Figure 6.9: UCB Variables Initialization And Execution

The calculation expressed as "*sum rewards [action selected]* += *reward*" was added to enable further analysis at each round of execution. The values that correspond to the reward for each channel selected was accumulated. By default, it is expected that the UCB algorithm performs better than just choosing a random channel that may not be available for message transmission (see Figure 6.10). In this case, the random channel selection returned a value of 106.934 while UCB algorithm returned 108.00, thus performing more accurately than random selection.

The variable N_{t_a} stores the number of times each an action was chosen, and as shown in Figure 6.9, the channel Facebook is the best channel because its distribution has the highest average. According to Figure 6.10, when each action is plotted against each channel selected for message transmission, the Facebook channel becomes the best channel selected.

```
In [63]:    #perform analysis on the Algorithm
            r_ = dataset.values[t,[1,2,3,4,5]] #get all rewards for time t to a vector
            r_best = r_[np.argmax(r_)]      #select the best action

            pick_random = random.randrange(d) #choose an action randomly
            r_random = r_[pick_random] #np.random.choice(r_) #select reward for random action
            if len(hist_achieved_rewards)>0:
                hist_achieved_rewards.append(hist_achieved_rewards[-1]+reward)
                hist_best_possible_rewards.append(hist_best_possible_rewards[-1]+r_best)
                hist_random_choice_rewards.append(hist_random_choice_rewards[-1]+r_random)
            else:
                hist_achieved_rewards.append(reward)
                hist_best_possible_rewards.append(r_best)
                hist_random_choice_rewards.append(r_random)

In [76]: print("Reward if we choose a channel randonmly {0}".format(hist_random_choice_rewards[-1]))
         print("Reward of the UCB method selecting a channel {0}".format(hist_achieved_rewards[-1]))

         Reward if we choose a channel randonmly 106.93175975858809
         Reward of the UCB method selecting a channel 108.00740572479013

In [65]: plt.bar(['Email','Facebook','Instagram','SMS', 'Telegram'],Nt_a)
         plt.title("Number of times each Message Channel was Selected")

Out[65]: Text(0.5, 1.0, 'Number of times each Message Channel was Selected')
```

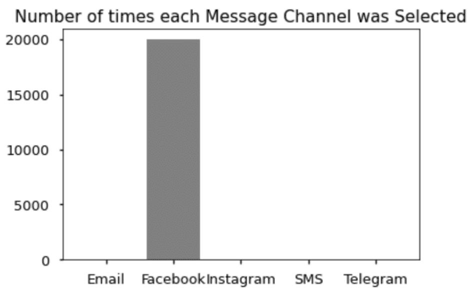


Figure 6.10: Random Channel Selection Vs UCB Algorithm Channel Selection

Furthermore, to explain the reward estimators expressed as 1 (reward) or 0 (loss) according to the selection outcome, the python binarizer module was used on the dataset to set the 0 or 1 loss or reward values for each channel based on a threshold of 99.99. Values less than or equal to the threshold are marked as loss (0) for the channel which means the channel is not available for message transmission while values above the threshold means the channel is available for message transmission and rewarded (1). The data pre-processing with the binarized data is shown in Figure 6.11.

Data Preprocessing

```
In [77]: ## Import the Libraries
         # x <= 99.9 = 0 else x > 99.9 = 1
         import pandas as pd
         from numpy import set_printoptions
         from sklearn.preprocessing import Binarizer

         #Load dataset and create headings for the dataset
         Binarydataset = pd.read_csv('/Users/olusolasalami/Documents/PhD/datasets_for_Channels/dataset1.csv')
         array = Binarydataset.values

         inputdata = array[:, 0:5]
         outputdata = array[:, 5]

         #Apply Preprocessing Transform
         binarizer = Binarizer(threshold = 99.9).fit(inputdata)
         newbinarizeddata = binarizer.transform(inputdata)

         set_printoptions(precision = 2)
         print(newbinarizeddata[0:5, :])

         newbinarizedinputdata = pd.DataFrame(newbinarizeddata)

         binarizedcsvfile = newbinarizedinputdata.to_csv('/Users/olusolasalami/Documents/PhD/datasets_for_Channels/processedData

         [[0. 1. 1. 0. 1.]
          [0. 1. 1. 1. 1.]
          [0. 0. 1. 1. 1.]
          [0. 1. 1. 0. 1.]
          [0. 1. 1. 1. 1.]]
```

[[0. 1. 1. 0. 1.] **Email**

[0. 1. 1. 1. 1.] **Facebook**

[0. 0. 1. 1. 1.] **Instagram**

[0. 1. 1. 0. 1.] **SMS**

[0. 1. 1. 1. 1.]] **Telegram**

Figure 6.11: Binarized Data Showing Facebook As The Best Channel For Message Transmission

6.6 COMPARATIVE REVIEW BETWEEN EXISTING MCM AND ML-ENABLED MCM FRAMEWORK

To review the differences and similarities between the traditional MCM system used by FSIs and ML-enabled MCM framework, it is important to bring into context the review conducted in Section 5.3 that led to the elicitation of the requirements for the ML-enabled MCM requirements in Section 5.4 (Hevner et al., 2004). The identified gaps in the review included the following, (i) pre-selected messaging channel, (ii) lack of ML algorithm in channel selection, message prioritization, and (iii) lack of self-learning was identified in the Kuda Mobile application reviewed (Etikan et al., 2016). This review provided an opportunity to fully grasp the benefits provided by the new approach of channel selection using ML techniques and, most importantly Reinforcement Learning, to manage the uncertainties of channel delivery response and availability that is non-existent in the implementation of MCM used by FSIs.

In addition to the review, it is pertinent to evaluate the benefits and values that the ML-enabled MCM system offers in relation to other investigations conducted on MCM Systems, ML and channel selection. MCM enables the integration of various channels into a single system. These platforms enable users to select from a variety of communication channels. Furthermore, the system's design allows for disparate channels to coexist and transmit similar types of information to the receivers. In the present era, businesses must effectively handle the various communication channels available to customers using ML techniques to automate the choice of customer preferences based on past delivery or channel availability.

Khan and Siddique (2004) implemented an MCM system in the banking industry with webservices open standards which allowed publish-and-subscribe features in the retail and credit card products. Customers and users were able to interact across different channels because to this service. However, the implementation was carried out by connecting directly to each different channel via the service integration bus, without a need for implementation of the message nature, urgency, or prioritization. The core of the system consists of a channel service selection layer (see Figure 6.12) that manages channel selection based on customers' pre-selected preferences. This is a major limiting factor when compared to the benefits of using ML channel selection with the TOW or UCB algorithm approach presented in this study.

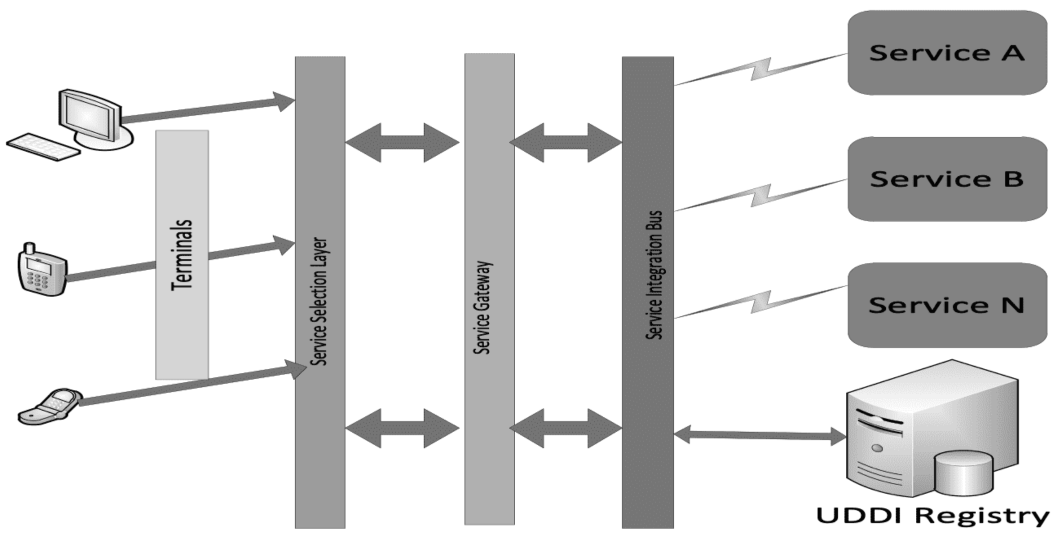


Figure 6.12: Webservices Based MCM System (Khan & Siddique, 2004)

Liang et al. (2011) designed the Integrated Multi-Channel Messaging Model (IM^3) which supports multiple channels (i.e., SMS, email, IM in Figure 6.13). The core of the system uses a general-purpose message format using XSLT (Extensible Style Language Transformation) to manage message translation between each channel with low efficiency owing to redundancy overhead with XML transformation. As shown in Table 6.3, the channel decision making module uses pre-classified values for each channel which must be pre-selected by the user of the system.

Table 6.3: Channel Message Urgency (Liang et al., 2011)

Degree	Code Name	Channel
Normal	1	email
Urgent	2	IM
Extremely Urgent	3	SMS (IM, email, SMS)

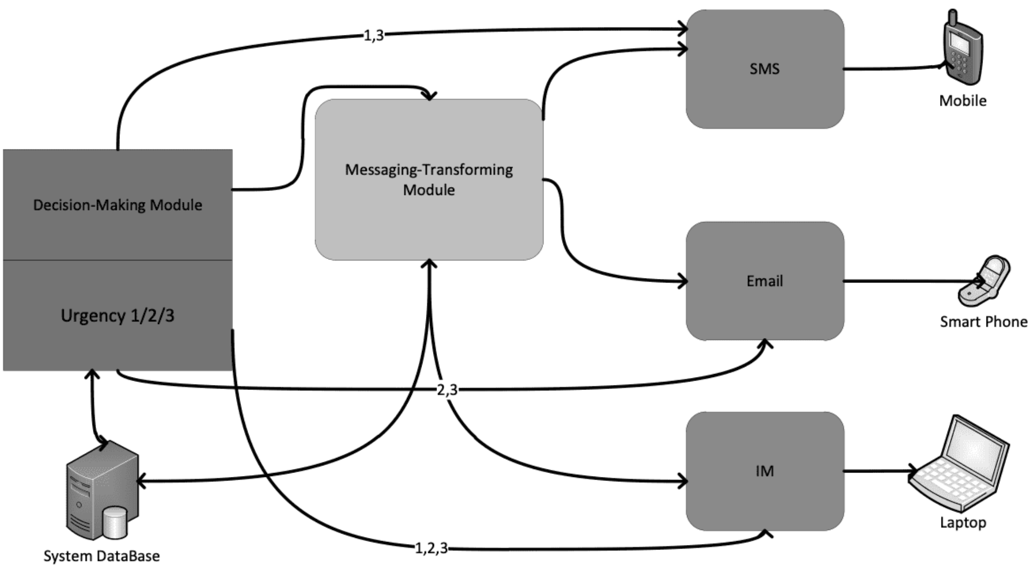


Figure 6.13: Integrated Multi-Channel Messaging Model (IM3) (Liang et al., 2011)

The foundation of this study improved on the work conducted by Kim et al. (2010) and Ma et al. (2019) with respect to the application of TOW dynamics to massive IoT channel selection which focused on hardware channel selection across various spectrums. The study relied mainly on the TOW algorithm which, according to the results of the above-mentioned authors, has roughly same results as the UCB compliant algorithm. Unlike previous algorithms for assessing the likelihood of obtaining a reward for each machine, the UCB-tuned algorithm is regarded as the best parameter-free algorithm. The authors applied TOW dynamics as a novel learning mechanism that is comparable to simultaneously updating both estimates in line with the volume conservation law. Owing to the unique nature of this study, which uses software messaging channels, in this study the researcher used a combination of both the TOW dynamics and the UCB for the channel selection module. The UCB algorithm complements the channel selection approach which was demonstrated in the simulation by offering an intuitive method that is aligned with Reinforcement Learning used to accurately calculate the reward estimators for each channel. Table 6.4 summarises the benefits and limitations of the related MCM works.

Table 6.4: Summary Of The Related MCM Works

SN	Related Work	Benefits	Limitations	ML-enabled MCM Benefits
1.	Liang et al. (2011) proposed and MCM system with tightly coupled decision making module	Decision making module with support for disparate channels Message translation module Message urgency	Overheads in XML or XLST translation. Tight coupling of the MCM components. Lack of automation in decision making Static message urgency assigned to each channel	The ML-enabled MCM system offers dynamic channel selection and message routing. The message translation layer uses JSON data agnostic layer for message sharing

SN	Related Work	Benefits	Limitations	ML-enabled MCM Benefits
				between various channels
2.	Khan and Siddique (2004) used point-to-point integration for MCM system that relies on the use of web services	Web services implementation offers third party integration Channel service registry	Lack of message translation module due to point-to-point integration using service integration. Lack of channel or message prioritization module	The ML-enabled MCM system was implemented with an ESB to connect all the disparate channels for ease of integration.
3.	Kim et al. (2010 and Ma et al. (2019) implemented channel using TOW methods in a homogenous massive IoT system	Implemented TOW dynamics for channel selection. Reward estimators determine channel availability	Homogenous hardware channel implementation (Single channel) Lack of the use of UCB integration with TOW dynamics to compute reward estimators	Disparate Multi-Channel Integration (Software channels). TOW or UCB integration to compute the channel rewards estimators.

A summary of the findings is shown in Table 6.5.

Table 6.5: Comparison Between Existing MCM & ML-enabled MCM Framework

Category	MCM	ML-enabled MCM
Message Sending Module	Yes	Yes
Message Producers	No	Yes
Integration of Disparate Channels	Yes	Yes
Support for 3rd Party App Integration	Yes	Yes
Determine Message Urgency using ML	No	Yes
Message Priority Module using TOW dynamics	No	Yes
Hot swapping of Channels	No	Yes
Support for Agnostic Data translation	Yes	Yes
Use customer's pre-profiled messaging channel	Yes	No

6.7 SUMMARY

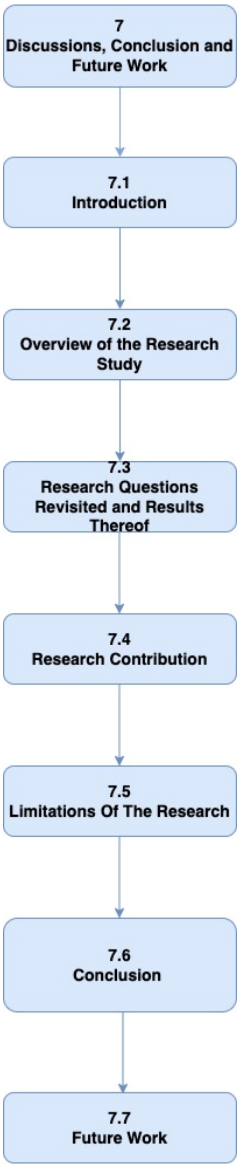
Chapter 0 has provided a full discussion on the demonstration and evaluation of the ML-enabled MCM artefact in a customer alert system used by FSIs using the use case evaluation (UCE) and simulation approach. The simulation of the artefact using Python to implement the TOW or UCB algorithm in Section 6.5 demonstrated that ML and most importantly Reinforcement Learning are directly applicable to the channel selection mechanism of a customer alert system with embedded MCM. Both the use case evaluation and the simulation offered an insight into the implementation of and offers strong proof relating to the efficacy of using the MAB or TOW for channel selection.

The use case evaluation method evaluates two important use cases to test the flow of the normal operation as customer of a FSI would do. The evaluation offered insight into the steps of operation and the final output that results in the message delivery to the customer terminal.

The MAB or UCB algorithm using Reinforcement Learning provided the implementation approach for channel selection in the ML-enabled MCM system by allowing the decision-making module of the solution to learn the channel's state and availability through their historical availability data and then make an informed decision rather than via assumption using trial and error. Furthermore, the Multi-Armed Bandit technique enabled the selection of the channels while relying on the foundation knowledge base of the k-armed bandits in which case a reward (1) or loss (0) value is awarded to each channel (Oshima et al., 2020). The reward value over time expressed as the action-value estimate for each channel is further used to determine how reliable each channel has been over-time during the operation of the artefact. The Upper Confidence Bound Algorithm enables the artefact to select the channel with the highest upper-bound value and further learns about other disparate messaging channels integrated on the platform.

In conclusion, the ML-enabled MCM system examination further revealed that the MAB or UCB technique meets the expectations of Reinforcement Learning as the binarized output and the UCB results showed the application of ML to the channel selection problem with the desired results during operation. The details of the implementation can be used by software developers to design the system using the designed framework. The next chapter provides a summary of the study including the conclusion and recommendations for future studies to enhance the field of ML and AI in messaging platforms.

7 CHAPTER 7 DISCUSSIONS, FUTURE WORK AND CONCLUSION



7.1 INTRODUCTION

This chapter covers the research findings, its implications, and suggestions for resolving some of the problems that emerged from the findings. The study's research questions and including the research objectives are also reviewed in relation to the study's findings. An overview of what this study has covered and accomplished is provided in the conclusion section based on the research outcomes. This chapter presents the research study's contributions as well as its limitations. The chapter closes with a discussion of potential future works in the messaging space using machine learning (ML) dynamics and conclusion on the benefits of the designed framework.

The aim of this research was to examine and investigate how integrated multi-channel messaging (MCM) systems can be implemented with the use of ML algorithms to enable effective and efficient dynamic channel selection. In this regard, the research study adopted the Tug-of-War (TOW) algorithm for channel selection based primarily on the Multi-Armed Bandits (MAB) problem. The study implemented the Upper Confidence Bound (UCB) algorithm to complement the TOW ML algorithm to overcome the challenges of randomised channel management, selection, scalability and learning in the MCM customer alert system used by Financial Services Institutions (FSIs). This chapter is divided into 7 main sections. The first section introduces the chapter, the second contains the overview of the research study, subsequently the third section revisits the research questions in relation to the contribution of other studies. The fourth section discusses this study's contribution to the field of ML, which includes the practical, methodological, theoretical and the general contributions of the study. The fifth section discusses the limitations of the study, and the sixth section makes inputs that can be considered in the future by other researchers. Lastly, the chapter closes with the achievements of the research in the seventh section.

7.2 OVERVIEW OF THE RESEARCH STUDY

In Chapter 1, the nature of the research problem was described, setting the context for the study. A significant number of FSIs across Africa have implemented Multi-channel Service Messaging service to provide messaging services and delivery to their customer base using the available customer channels with the inherent objective of improving service delivery on their platforms. This objective has given the customer options to monitor and view their transactions online and in real time. Furthermore, this objective has in some cases protected customers of FSIs from targeted fraud or security breaches with the use of OTP or MFA services to authenticate their identity with the FSIs. However, lack of a ML framework embedded within the MCM implementation by FSIs has forced customers to use pre-selected messaging channels for which they want to be alerted at an instance of time. Some studies have implemented dynamic channel selection using hardware channels. This study did not find any study specifically that has implemented ML or artificial intelligence (AI) using software messaging channels including both traditional channels such as SMS, email or social media channels such as Telegram, Facebook, WhatsApp, and Twitter DM.

Chapter 1 presented the research project and the background of the study. This covered the introduction to the problem and the relevance of ML-enabled framework for enhancing channel messaging selection, which emphasises the significance of creating a framework that can serve as a guide for improving the quality of dynamic message delivery in customer alert systems used by FSIs. Furthermore, the research questions were formulated, and the significance of the study was articulated.

The benefits of using ML-enabled MCM framework was revealed through the detailed literature review in Chapter 2. It was discovered that disparate channel integration using a common data agnostic method was necessary to share the same message data between the different channels. The review examined the different channel messaging formats used to establish how the integration can be implemented. This led the researcher to review existing MCM implementation by FSIs to understand the requirements for the framework that uses ML capabilities. Various ML algorithms that support decision making and learning were also reviewed with specific focus on Reinforcement Learning algorithm because it offered the study the benefit of both internal and external learning capabilities. The algorithms reviewed includes the modified Ɛ-greedy, the modified SoftMax, UCB and TOW, where the last two algorithms formed the pivot for the framework design owing to the leverage they offer for payoffs and rewards systems.

The review laid a foundation for the theoretical approach selected which used both TOW and Actor-Network theory (ANT) theories to build a firm foundation for the use of the Upper Confidence Bound (UCB) for channel selection in the framework. The chapter ventured into the ML and channel selection algorithms with specific reference to their benefits and the works highlighted by other researchers to situate the problem as a Reinforcement Learning use case. Also, the topic of the MCM conceptual framework was reviewed, and this allowed the researcher to understand the impact of message priority, message prioritization and integration via an Enterprise Service Bus (ESB). Existing MCM system implementations including the integrated MCM model and a communication middleware system based on MCM were examined. Also, the interconnection of systems using IoT devices with MCM systems was reviewed in-line with ANT theory to examine the actors within the framework including the support for messaging protocols such as IoT Message-Based Communication (IMBC) and Lightweight Machine to Machine (LwM2M) protocol. This highlighted the need to use a variable messaging payload for disparate message channels rather than fixed messaging models in MCM systems. The base requirements for the ML-enabled system were unpacked in Chapter Two.

In Chapter 3, the foundational theories underpinning the study were discussed, the TOW and ANT, were examined. The chapter examined the relationships between message producers and message channel behaviours. The chapter uncovered similarities between heterogenous networks supported by ANT and disparate channel integrations in the ML-enabled MCM framework implementation. In addition, the review demonstrated the limits of ANT and TOW and the relationship between both theories.

Chapter 4 outlined the research methodology underpinning this research work, which includes the pragmatist approach, Design Science Research (DSR) and process-based research approach. The pragmatist philosophical approach was chosen owing to its support for experimentation. The DSR methodology offers a framework for problem identification and the ability to proffer solutions using predefined steps that are iterative in nature. This approach ensured the process produced a quality artefact which is very important for the framework being designed. The chapter discussed in detail the DSR methodology and concluded with the ethical approval gotten from the university which serves as the terms of reference for the research work.

In Chapter 5 the researcher explored, in line with the DSR problem identification phase, the possibility of gathering the requirements that are necessary for the implementation of the ML-enabled MCM framework. These set of requirements were derived from the review of existing MCM applications hosted publicly by FSIs on Google Play store, Apple iOS stores and web platforms. This requirement elicitation led to discovery of the importance of components such as channel discovery, gateway layer, decision and learning module using ML and message transformation. In addition, the development and design of the ML-enabled artefact was presented. The architecture underpinning the implementation was presented showing all the components required for development of the artefact in line with the ANT via the identification of the actors within the system. The user interface was designed and relationships between the application layer, classes and channel selection logic using TOW dynamics were discussed. The system flow chart including the suggested class interface implementation was also presented to guide system designers. The chapter concluded with ideas on proposing the design for the framework using these requirements and the selection for high level programming language C# used for the implementation in this study and the database backend selected for hosting the application data.

In Chapter 6, the review of the ML-enabled framework was provided. A review of the system was conducted using the use case evaluation approach, selected to evaluate the framework to allow the researcher to evaluate the usefulness of the artefact. Furthermore, the researcher discussed the evaluation's feedback and simulated the performance of the designed framework using the TOW or UCB approach. Through a simulation with Python, a ML programming language enabled the researcher to use data to evaluate the system and understand how the artefact can improve the efficiency of message delivery channels to FSI customers using the system.

7.3 RESEARCH QUESTIONS REVISITED AND RESULTS THEREOF

In Chapter 1 the research questions formulated in this research work were presented. The questions were posed by the researcher to focus on how FSIs can integrate ML in their customer alert messaging systems to improve the delivery of message delivery services to their customers. The following sections discuss the study's findings per research question.

7.3.1 How can integrated multi-channel messaging be implemented using machine learning algorithms to enable effective and efficient dynamic channel selection and integration methods?"

This is the primary research question of this study on which the proposed ML-enabled MCM framework was designed and built. The question was posed to understand how FSIs can then leverage the recent advances in messaging technology to offer superior services to their customers given that traditional electronic and social media messaging platforms (particularly SMS, email, WhatsApp, Telegram, Facebook and Twitter DM) have become the go to messaging channels for many people around the world. The question was asked by the researcher to provide a solution that addresses the challenges confronting FSIs with the use of customer alert systems with MCM framework which confines the customers to use pre-selected channels for message delivery.

Based on the findings in this research, an ML-enabled MCM system can be implemented by embedding Reinforcement Learning into the channel selection module of the system. A user layer is required and serves as an entry point into the system for FSI customers to interact with the system and realise the benefits. In addition, a ML layer that makes use of a web API module that exposes the logic of the application to other third-party systems. The ML layer also provides the integration of heterogenous messaging channels in a single system with ML features. This addresses the challenge of heterogenous messaging channels which makes use of different messaging formats both for traditional and social media messaging channels that were used in the study. A channel integration bus is also required to allow the integration of the channels and also supports the use of an agnostic data sharing message format (JSON) to enable channels to share the same message structure within the system. Furthermore, the

ML layer used must implement message producers that enables the system to accept input messages from multiple sources. In addition, the core of the system should include a decision module, based on the design in this study this module leverages user channel management worker services. This was implemented with the TOW or UCB machine learning worker services responsible for learning, calculating channel payoffs (reward or losses) and the worker service for channel selection that manages and coordinates dynamic channel selection within the framework. Also, the database layer is essential to store the configuration and manage the overall application and learning data used by the system. Overall, the UCB algorithm eliminates the need for randomly selecting channels since the channel throughput is calculated using the algorithm during the framework operation which provides an efficient and effective method for channel selection.

The framework design and result align with the earlier results from the works by Kim et al. (2010) and Ma et al. (2019), which use TOW dynamics and its application in homogenous IoT hardware channel selection as well as that of Hasegawa et al. (2020), which created a framework for a distributed channel selection solution using the MAB. However, this study provided an enhancement with the integration of heterogenous messaging channels. The MAB problem provided an approach for the solution design for the channel selection problem using the reward and loss concept while TOW allows the framework to randomly choose the channels with significant payoffs. This study provided an approach that eliminated the need for random channel selection by using the UCB algorithm that is different from the use of non-locally correlated search agents method (Kim et al., 2010a).The use of the UCB action selection within the framework provided a balance between the exploration and exploitation approach, as shown in the simulation results, by leveraging the uncertainty in action-value estimates (Auer et al., 2002). This occurs with the use of a sampling set of rewards (Lai & Robbins, 1985). Due to the action-value estimates inherent uncertainty, the framework uses the UCB uncertainty to its advantage to promote channel exploration with decentralized elimination (Féraud et al., 2019).

7.3.2 What are the general requirements of a machine learning-enabled multi-channel messaging platform?

The general requirements specify which features the ML-enabled MCM artefact must have and how those features must function. This aided in the definition of test criteria with the use case evaluation conducted, which is necessary for verification and validation of the designed framework. Requirements gathering occurs when creating a new artefact as well as modifying or changing an existing product. This further allowed the design efforts to be quantified and bound, therefore this question partially defines the implementation scope of the study.

The elicitation of requirements using the publicly-hosted MCM applications by FSIs selected for this study revealed the need to design a framework that satisfies the following design requirements:

- **Channel Service Layer**: This requirement provides web service support, channel service availability, management, and quality of the channel service. This requirement also enables the routing of messaging data between the heterogenous channels in the framework.
- **Channel Gateway Layer**: This requirement allows the framework to implement business logic using the ML algorithm (i.e., TOW or UCB) for channel selection. This layer also provides message assurance and can detect security issues in messages being shared within the platform by the channels.
- **Channel Service Integration Layer**: This layer captures the requirements for the channel service meta-data since the framework supports heterogenous and disparate messaging channels. Other requirements fulfilled are message translation and exception handling within the framework which allows the framework to resume its current state of operation in case of service downtime.
- **Channel Decision and Learning Layer**: This requirement is the core of the framework design that implements the TOW or UCB. Message delivery, channel availability status and reliability are determined and computed within this layer. With the use of TOW or UCB algorithm this layer learns the reward (successful message delivery) or loss (failed message delivery) and applies this learning in routing messages to be sent to each channel on the platform without random channel selection.
- **Channel Transformation Layer:** Owing to the support of multiple messaging channels which use different messaging formats, this requirement provides a method that allows the use of a transformation layer using JSON and agnostic messaging format to make sharing of messaging data seamless between channels.

The channel service layer and channel transformation layer design follows the design approach by Ganesh et al. (2004), which supports routing of messages between channels, this is required to support sharing of information between the channels. However, Ganesh et al. (2004) supports and provided a framework that allows agents to choose channel availability uniformly and independently at random. The framework proposed in this study provides an enhancement with the use of TOW or UCB algorithm at the channel decision and learning layer which leverages exploitation and exploration approach thus eliminating the need for random selection (Auer et al., 2002; Féraud et al., 2019; Ma et al., 2019).

7.3.3 What machine learning algorithms can be used to seamlessly determine the effective channel or s to use for message delivery on an MCM platform?

This question was asked to understand and choose ML algorithms that support selection and choice. Effective and efficient channel selection is at the core of the proposal for the design of the ML-enabled MCM system including a learning feature that can use past experiences to determine the state for future operations.

The results of the in-depth literature review revealed that this class of solution can only be provided by algorithms that were built and premised on Reinforcement Learning. This led to the researcher focussing on Reinforcement Learning-machine learning algorithms such as (i) the modified Ɛ-greedy algorithm, (ii) the modified SoftMax algorithm (iii) UBC algorithm, and (iv) the TOW algorithm. This review allowed the researcher to understand what each algorithm offers and how applicable they were in solving problems with disparate messaging channels in a customer alert system. The literature review also revealed the need for the framework to use an ML algorithm that enables an agent to learn or perform channel selection with interaction in a complex environment with minimal failure (Zhou et al., 2018). This led to the choice of TOW or UCB because it provided a reward or loss approach used for the assessment of the reliability of each integrated channel (Hasegawa et al., 2020).

The simulation of the artefact with Python ML framework with TOW revealed that channel selection leverages random selection while relying on the payoff values in line with the framework proposed by Ma et al. (2019) in a study that used homogenous IoT hardware channel selection. The TOW ML framework also works in a decentralized fashion leveraging on actions that only depend on their observed operational history (reward or loss) in the past (Sankararaman et al., 2019). The research study established that while TOW dynamics provided the platform for the reward system based on the MAB approach, the UCB algorithm further complements and improves channel selection using a cumulative approach for each channels status eliminating the need for the random channel selection approach that TOW supports as shown in the results of the study based on the action estimates for each channel. The combination of these two algorithms TOW or UCB presented betters result than the frameworks implemented by Ma et al. (2019) and Sankararaman et al. (2019) which relied on random channel selection. This study, on the other hand, used the exploitation and exploration approach to provide efficient channel selection. These results are well documented in Section 6.5.

7.3.4 What problems will an efficient dynamic channel selection machine learning MCM platform solve for Financial Services Institutions using MCM for Customer Alert Systems?

This question was posed to determine the value that dynamic channel selection using ML provides for FSIs that implement the framework in their MCM Customer Alert Systems.

Based on the review of existing MCM system and the simulation conducted on the designed framework, the author identified the following problems that can be solved:

- Pre-selected messaging channels which force the FSI customer to choose a message delivery channel upfront.
- Lack of channel service integration layer that handles the integration of channels and sharing of messages using a data agnostic messaging framework
- Lack of a ML and decision-making module that uses Reinforcement Learning to determine channel service availability and message delivery.
- Channel selection logic that used random approach for channel selection.

These set of problems are applicable to a plethora of solutions offered by FSIs that transmit messages to their customers such as Internet Banking, Product websites and service application BOTs (ExadelTeam, 2022). Dynamic channel selection proposed in this study is in line with Reinforcement Learning where the channel state and data is updated after each operation (Alsheikh et al., 2014).

7.3.5 How can MCM platforms be enhanced with machine learning algorithms to enable dynamic learning capacity of the channel selection module?

This objective of this research question was to determine how existing MCM systems can be enhanced with ML to provide dynamic channel selection.

To meet this objective, the study proposed a framework that introduced Reinforcement Learning algorithm (TOW or UCB) which uses an exploitation and exploratory approach for channel selection. The framework was designed in Chapter 5 of this study. The artefact was refined and evaluated (through the use-case evaluation) and design validated (through the simulation method) and the results were presented and discussed in Chapter 6 . The results of the evaluation and simulation were discussed in sections 6.3 and 6.5. The results revealed the following key findings that can be used to enhance existing MCM platforms with dynamic learning:

- The need to introduce and implement ML in the MCM channel selection module.
- The need to use Reinforcement Learning-machine learning using the TOW or UCB approach that provides an efficient payoff calculation for channel selection within the system (Alsheikh et al., 2014; Ma et al., 2019; Wang et al., 2018).
- The need for a database layer that can store information about each channel's operation and channel service configuration.
- The need for message producer configurations that allows efficient setup and configuration for messaging agents on the system. This allows message producers to be hot swappable during business operations hence providing reliability to the ML-enabled MCM system.

7.4 RESEARCH CONTRIBUTION

This research contributes immensely to body of knowledge (theoretically and empirically) in the field of ML and digital messaging which aims to enhance the automation and message service delivery by FSIs. The study employs a novel technique to improve the dynamic selection of messaging channels in a customer alert system used by FSIs, in which a framework was developed to help mitigate the challenges of pre-selected message delivery channels used by customers in MCM systems. The designed framework is an appropriate solution for this study. As a result, the following contributions were made by this research study:

7.4.1 Practical Contribution of the Research

The practical contribution of this study is based on the following findings: In the use case study and the simulation of the ML-enabled system, real-world problems such as message deliverability and channel self-learning were identified and solutions using TOW or UCB were demonstrated. Furthermore, factors that contribute to the successful design and implementation of a ML-enabled multi-channel messaging system were identified (exploitation and exploration methodology or balance), thereby improving the quality of the message delivery framework. These were broadly covered in academic research publications that stemmed from this research.

Furthermore, the objective of this study was to investigate and proffer solutions to how FSIs can use ML technology in their MCM customer alert implementations to improve the delivery of messages timeously to their customers. In the review for the requirements using publicly-hosted MCM applications by FSIs, it was evident that there was no proper framework to follow when it came to implementing ML-enabled MCM systems, which constrained customers to use pre-selected messaging channels and thereby under-utilisation of the technologies available in the contemporary to improve message service delivery. The detailed work on the designed framework is described in Chapter 6, which is essentially the study's practical contribution. It is acknowledged and accepted that, FSI stakeholders, decision-makers and software designers cannot be expected to directly understand the entirety of the research to appreciate its practicability in real life system design. However, given that the focus of this work is research based (especially academic research), one can undoubtedly make this claim without providing evidence of the practical benefit. It does, however, imply that there would be a need for a concerted effort to "implement" the work that has been conducted into a form that would be usable by FSI stakeholders to provide guidance to these s stakeholders based on this research work. However, this is entirely outside the scope of this research. Other practical contributions that this work provides include the following:

- An innovative method of channel resource allocation in an automated way that allows message producers select available channels, powered by the TOW or UCB Reinforcement Learning algorithm.
- A centralised and integrated layer using a channel service bus; this approach provided a method to achieve disparate channel management and integration across a decentralised network of channel-resource providing nodes in the framework.
- A configuration method for specifying appropriate system parameters for each messaging channel to guarantee stability within the system and to further allow transparency when a channel selected is not available to process messages from a message producer.
- An approach for software messaging channel availability using ML approach that is capable of obtaining the channel throughput. The throughput is determined for each channel using the upper confidence bound algorithm to determine reward and loss attributes for each channel resource in the framework.

The proposed method was also assessed for its capacity to yield the desired channel availability and resolution of allocation constraints. This approach also eliminates the need to randomly select or pre-select channels for customers on the platform. Nevertheless, the TOW or UBC algorithms do not rule out the use of more sophisticated cognitive or other ML approaches since they continue to evolve. Most importantly, this thesis shows that, under the assumptions of agnostic message data sharing between disparate channels, the channel reward or loss aggregate using the throughput values are sufficient to produce the desired channel allocation, learning and assignment within the framework.

7.4.2 Methodological Contribution of the Research

The combination of TOW and ANT is the study's primary methodological contribution. This was accomplished through the classification and identification of each component with the framework (from ANT) and ML capacities using the MAB approach (from TOW) during use case evaluation. This was accomplished in both Chapters 3 and 6 . Another methodological contribution is the way the researcher implemented the simulation of the framework using the UCB approach and then highlighted the results of the simulation using the Python framework.

7.4.3 Theoretical Contribution of The Research

The TOW and ANT were the two theories used in this research study. These theories and use of case evaluation aided the researcher's understanding of the roles of message producers, message channels, TOW algorithm, UCB algorithm used to implement the ML-enabled MCM customer alert system. Furthermore, it contributes to a better understanding of how FSIs can leverage ML frameworks to improve message service delivery to their customers. Theoretically, the research contribution can be found in the framework developed in Chapter 5.

7.4.4 General Contribution of the Research

The general contribution of this study is evident in the use case and the possibility of enabling FSI customers to receive messages using dynamic messaging channel selection. This feature of dynamic channel selection eliminates the need for customers to pre-select messaging channels and further improves their interaction with FSIs and their service delivery. The study also aids in the integration of disparate messaging channels in a single platform enhanced with ML to provide dynamic selection of channels and learning module for tracking delivery of messages to customers. The study is anticipated to provide researchers and FSIs with background information on how Reinforcement Learning algorithms can improve the delivery of messages using disparate channels.

7.5 LIMITATIONS OF THE RESEARCH

The recommendations for improving the selection of channels using dynamic ML methods in a customer alert MCM system used by FSIs have been presented in this study. The process resulted in the development and proposal of a framework for addressing the problems with pre-selected channels and integration of disparate messaging channels. The requirement elicitation for the framework relied on publicly hosted applications by FSIs on the Google

Playstore and iOS stores. The review of these publicly hosted applications limited the researcher's direct access to the FSIs since this could have provided a better platform to interact on other features that could enhance the framework design. Another limiting factor of this research study was the access restriction to FSIs on the African continent mainly due to the location of the researcher and understanding of the technology and business terrain. Lastly, the data of 20,000 samples for each channel were produced for the simulation of this study and used due to computing capacity of the machines used for the implementation, it would be interesting to see larger samples being used in the future when technology and capacity for processing increases.

7.6 CONCLUSION

This study has presented a new method for dynamic messaging channel selection using the TOW or UCB ML algorithms that was demonstrated in an ML-enabled MCM customer alert system used by FSIs. This result has demonstrated that an existing MCM system can be enhanced with ML to offer an ML-enabled MCM system. The designed ML-enabled framework achieved this objective of dynamic channel selection using the TOW or UCB reinforcement ML algorithm that provided a reward and loss values used to determine channel availability. Ultimately, the designed framework promotes improved methods of deriving the availability of the disparate channels integrated using the computed throughput value for each channel. Additionally, the study contributed an effective and efficient evaluation of the exploration and exploitation features of the MAB base algorithm which eliminated the use of random channel selection and applicable for use in the ML-enabled MCM customer alert systems used by FSIs.

The researcher hopes that the findings presented in this thesis will help researchers to better understand how to design and integrate ML into messaging system that can deal with dynamic message channel selection. As shown in the roadmaps for ML channel selection and AI by Ma et al.(2019), Alsheikh et al. (2014), Hendrik (2020) and Kim et al. (2010a), there are still a lot of open challenges such as (i) The distributed and light weight message exchange techniques, (ii) Reinforcement Machine learning algorithms, efficient data-clustering patterns and (iii) Adoption of ML in resource allocation and management in disparate messaging-channels and (iv) Enhanced reinforcement machine learning systems that are resilient, scalable, adaptive and reliable. The researcher strongly believes that the work presented in this study provides a platform for future research that help solve some of these of challenges and advance the field of messaging and ML or AI. Parts of the research findings were also published at internationally renowned academic conferences with feedback incorporated into the design of the framework.

7.7 FUTURE WORK

The dynamic channel selection and learning feature using TOW or UCB algorithms was discussed in this study. Subsequent work should examine additional ML procedures such as the modified Ɛ-greedy algorithm and SoftMax methods, which are covered in sections 2.5.3 and 0. Research can be conducted on how the Ɛ-greedy approach and multinomial logistic regression can be used for channel selection, learning, and calculating payoffs. In earlier studies, ML techniques that incorporate logistic regression were used (Gao & Pavel, 2017b; Kim et al., 2010). The model presented in this thesis is intended to be abstract and generic such that it is applicable to all classes of systems that exhibit their characteristics, that is, channel resource allocation and delivery learning. For the approach to be applicable to ML computing systems, cloud computing, and reinforcement machine learning computing, the design or extension of the framework to incorporate the model is a crucial next step. Future studies should investigate how these techniques relate to better calculations for each channel’s payoff and learning ability of the framework. The advantages of using these methods should be reviewed to determine the accuracy of the predictions produced by existing reinforcement machine learning algorithms.

REFERENCES

Aaron, O. (2018). *Multi-Channel Marketing: Definition, Data, and a Strategy to Sell Anywhere*. https://www.shopify.com/enterprise/multi-channel-marketing

Abowd, G., Foley, J., Gromala, D., Mynatt, E., Pierce, J., Potts, C., Shaw, C., & Stasko, J. (2011). *Requirements Gathering & Task Analysis Project Part 1 : Big Picture*.

ACSC. (2016). Threat Report. *Australian Cyber Security Center*, *1*, 40.

Adagunodo, E. R., & Bamidele, O. (2007). SMS Banking Services: A 21 st Century Innovation in Banking Technology. In *Issues in Informing Science and Information Technology* (Vol. 4). http://www.bankislam.com

Adikari, S., McDonald, C., & Campbell, J. (2009). Little design up-front: A design science approach to integrating usability into agile requirements engineering. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5610 LNCS*(PART 1), 549–558. https://doi.org/10.1007/978-3-642-02574-7_62

Alhassan, G. S. (2020). *E-governance for sustainable development in Ghana: Issues and prospects.* i–93.

Alliance, O. M. (n.d.). *Lightweight Machine to Machine Technical Specification: Core*. Retrieved November 28, 2020, from https://www.omaspecworks.org/about/intellectual-property-rights/.

Alsheikh, M., Lin, S., Niyato, D., & Tan, H. P. (2014). Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Communications Surveys and Tutorials*, *16*(4), 1996–2018. https://doi.org/10.1109/COMST.2014.2320099

Alturki, A., Gable, G. G., & Bandara, W. (2013). The design science research roadmap: In progress evaluation. *Proceedings - Pacific Asia Conference on Information Systems, PACIS 2013*.

Amazon Web Services. (2020). *Multi-Channel Messaging System*. AWS. https://aws.amazon.com/pinpoint/customer-engagement/multi-channel-messaging/

Anitha, A., Varalakhshmi, M., Mary Mekala, A., Subashanthini, & Thilagavathy, M. (2018).

Secured cloud banking transactions using two-way verification process. *International Journal of Civil Engineering and Technology*, *9*(1), 531–540.

Arkich, M. (1992). The De-scription of Technical Objects. In *Shaping technology/building society: studies in sociotechnical change* (pp. 205–224). In Bijker, W.E. and Law, J., Eds., Shaping Technologies/ Building Society. Studies in Sociotechnical Change, The MIT Press, Cambridge/London, - References - Scientific Research Publishing. https://www.scirp.org/(S(vtj3fa45qm1ean45vvffcz55))/reference/ReferencesPapers.aspx?ReferenceID=1252509

Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, *47*(2–3), 235–256. https://doi.org/10.1023/A:1013689704352

Avison, D. E., Lau, F., Myers, M. D., & Nielsen, P. A. (1999). Action research. *Communications of the ACM*, *42*(1), 94–97. https://doi.org/10.1145/291469.291479

Ayodele, O. M., & Babajide, O. (2015). *Assessment of Use of Social Media in Real Estate Transactions in Lagos Property Market*. *1*(2), 63–68.

Bell, E., & Bryman, A. (2007). The ethics of management research: An exploratory content analysis. *British Journal of Management*, *18*(1), 63–77. https://doi.org/10.1111/j.1467-8551.2006.00487.x

Belzner, L., & Gabor, T. (2016). QoS-Aware multi-Armed bandits. *Proceedings - IEEE 1st International Workshops on Foundations and Applications of Self-Systems*, 118–119. https://doi.org/10.1109/FAS-W.2016.36

Berthold, U., Fu, F., Van Der Schaar, M., & Jondral, F. (2008). *Detection of Spectral Resources in Cognitive Radios Using Reinforcement Learning*.

Bishop, C. (2006). Pattern recognition and machine learning. In *Pattern recognition and machine learning* (Vol. 9, Issue 4, pp. 257–261). https://doi.org/10.1109/TIT.1963.1057854

Boucaut, R. (2001). Understanding workplace bullying: A practical application of Giddens' Structuration Theory. *International Education Journal*, *2*(4), 65–73.

Brown, J., Shipman, B., & Vetter, R. (2007). SMS: The short message service. *Computer*, *40*(12), 106–110. https://doi.org/10.1109/MC.2007.440

Bumblauskas, D., Mann, A., Dugan, B., & Rittmer, J. (2020). A blockchain use case in food distribution: Do you know where your food has been? *International Journal of Information Management*, *52*, 102008. https://doi.org/10.1016/J.IJINFOMGT.2019.09.004

Caflisch, A., Grubb, M. D., Kelly, D., Nieboer, J., & Osborne, M. (2020). Sending Out an SMS: The Impact of Automatically Enrolling Consumers Into Overdraft Alerts. *SSRN Electronic Journal*, *May*. https://doi.org/10.2139/ssrn.3538527

Callon, M. (1986). The Sociology of an Actor-Network: The Case of the Electric Vehicle. In *Mapping the Dynamics of Science and Technology* (pp. 19–34). Palgrave Macmillan UK. https://doi.org/10.1007/978-1-349-07408-2_2

Callon, M. (1991). Techno-economic Networks and Irreversibility. *The Sociological Review*, *38*(1_suppl), 132–161. https://doi.org/10.1111/j.1467-954x.1990.tb03351.x

Callon, M., & Latour, B. (1981). LInscrewing the big Leviathan: how actors macro-structure reality and how sociologists help them to do so. In *In K. Knorr-Cetina & A. V. Cicourel (eds.)*. Routledge & Kegan Paul, London. https://doi.org/10.1111/j.1572-0241.1995.tb08084.x

Chen, T., Zhang, H., Maggio, G. M., & Chlamtac, I. (2007). Topology management in CogMesh: A cluster-based cognitive radio mesh network. *IEEE International Conference on Communications*, 6516–6521. https://doi.org/10.1109/ICC.2007.1078

Chiu, D. K. W., Kwok, B. W. C., Wong, R. L. S., Cheung, S. C., Kafeza, E., & Kafeza, M. (2004). *Alerts for Healthcare Process and Data Integration* +. *00*(C), 1–10.

Chow, J. C. K. (2017). *Analysis of Financial Credit Risk Using Machine Learning*. *April*. https://doi.org/10.13140/RG.2.2.30242.53449

Cioffi, R., Travaglioni, M., Piscitelli, G., Petrillo, A., & De Felice, F. (2020). Artificial intelligence and machine learning applications in smart production: Progress, trends, and directions. *Sustainability (Switzerland)*, *12*(2), 492. https://doi.org/10.3390/su12020492

Conway, A. (2016). *User Acceptance Testing (UAT)*. Bugwolf.

Cortiñas, M., Chocarro, R., & Villanueva, M. L. (2010). Understanding multi-channel banking customers. *Journal of Business Research*, *63*(11), 1215–1221. https://doi.org/10.1016/j.jbusres.2009.10.020

Creswell, J. W. (2003). *creswell_Research Mthods_Qual_Quant Mixed Methods Approaches.pdf* (2nd Editio). Sage Publications Ltd London EC1Y 1SP.

Creswell, J. W. (2009). Research Design: Qualitative, Quantitive, and Mixed Methods Approaches. In *Sage Publications* (3rd Editio, Vol. 20, Issue 2). Sage Publications Ltd London EC1Y 1SP. https://doi.org/10.1080/14675980902922143

Creswell, J. W., Hanson, W. E., Clark Plano, V. L., & Morales, A. (2007). Qualitative Research Designs: Selection and Implementation. *The Counseling Psychologist*, *35*(2), 236–264. https://doi.org/10.1177/0011000006287390

DeAngelis, A. (2019). *6 Reasons To Implement VoIP for Financial Institutions*. https://blog.votacall.com/voip-for-financial-institutions

Deloitte. (2020). Remote Collaboration Facing the challenges of COVID-19. *Deloitte*, *March*, 1–13. https://www2.deloitte.com/content/dam/Deloitte/de/Documents/human-capital/Remote-Collaboration-COVID-19.pdf

Eisenhardt, K. M. (1989). Building Theories from Case Study Research Published by : Academy of Management Stable. *The Academy of Management Review*, *14*(4), 532–550.

Ellram, L. M. (1996). The use of the case study method in logistics research. *Journal of Business Logistics*, *17*(2), 93–138.

Etikan, I., Musa, S. A., & Alkassim, S. R. (2016). Comparison of Convenience Sampling and Purposive Sampling. *American Journal of Theoretical and Applied Statistics*, *5*(1), 1. https://doi.org/10.11648/j.ajtas.20160501.11

ExadelTeam. (2022). *Machine Learning in Banking and Finance | Exadel*. https://exadel.com/news/how-machine-learning-is-used-in-finance-and-banking

Fayyaz, Y. (2016). The Evaluation of Voice-over Internet Protocol (VoIP) by means of Trixbox. *International Journal of Natural and Engineering Sciences*, *10*(December), 33–41. https://www.researchgate.net/publication/313101329_The_Evaluation_of_Voice-over_Internet_Protocol_VoIP_by_means_of_Trixbox

Féraud, R., Alami, R., & Laroche, R. (2019). Decentralized exploration in multi-armed bandits. *36th International Conference on Machine Learning, ICML 2019*, *2019-June*(May), 3368–3379.

François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., Pineau, J., Brain, G., & -Delft, B. (2018). *An Introduction to Deep Reinforcement Learning*. *11*(3–4), 1–140. https://doi.org/10.1561/2200000071

Gaaloul, K., & Molnar, W. (2014). Research methodologies in enterprise engineering: Insights from a workshop. *Proceedings - 2014 International Workshop on Advanced Information Systems for Enterprises, IWAISE 2014*, 58–64. https://doi.org/10.1109/IWAISE.2014.8

Ganesh, J., Srinivas, P., & Moitra, D. (2004). Web Services and Multi-Channel Integration : A Proposed Framework. *Proceedings of the IEEE International Conference on Web Services (ICWS'04)*, 8.

Gao, B., & Pavel, L. (2017a). *On the Properties of the Softmax Function with Application in Game Theory and Reinforcement Learning*. *August*. http://arxiv.org/abs/1704.00805

Gao, B., & Pavel, L. (2017b). *On the Properties of the Softmax Function with Application in Game Theory and Reinforcement Learning*. 1–10. http://arxiv.org/abs/1704.00805

Garg, K., Sureka, A., & Varma, V. (2015). A case study on teaching software engineering concepts using a case-based learning environment. *CEUR Workshop Proceedings*, *1519*, 71–78. https://www.iiit.ac.in/

Gray, D. E. (2014). Theoretical perspectives and research methodologies. In *Doing research in the real world*. Sage Publications Ltd London EC1Y 1SP. http://www.uk.sagepub.com/books/Book239646#tabview=toc

Gutowski, N., Amghar, T., Camp, O., & Chhel, F. (2019). *CONTEXTUAL MULTI-ARMED BANDITS Global Versus Individual Accuracy Track on Recommender Systems*. *September*.

Hansen, P., Järvelin, A., Eriksson, G., & Karlgren, J. (2014). A use case framework for information access evaluation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8830*, 6–22. https://doi.org/10.1007/978-3-319-12511-4_2/COVER/

Hasan, M., Haron, N., & Md Yazid, N. S. (2010). Development of multimedia messaging service (MMS)-based examination results system. *Proceedings of the 9th WSEAS International Conference on Applications of Computer Engineering, ACE '10*, *January*, 157–163.

Hasegawa, S., Kim, S., Shoji, Y., & Hasegawa, M. (2020). *Performance Evaluation of Machine Learning Based Channel Selection Algorithm Implemented on IoT Sensor Devices in Coexisting IoT Networks*.

Hendrik, F. (2020). *Artificial Intelligence and Sustainability*. https://www.future-customer.com/artificial-intelligence-and-sustainability/

Herbert, R. (1952). Some Aspects of the Sequential Design of Experiments. *Bulletin of the American Mathematical Society*, *58*(5), 527–535. https://doi.org/10.1090/S0002-9904-1952-09620-8

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science In Information Systems Research. *MIS Quarterly*, *28*(1), 75–105.

Hornbæk, K., Høegh, R. T., Pedersen, M. B., & Stage, J. (2007). Use case evaluation (UCE): A method for early usability evaluation in software development. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4662 LNCS*(PART 1), 578–591. https://doi.org/10.1007/978-3-540-74796-3_58

Jacobson, I., Spence, I., & Kerr, B. (2016). Use-case 2.0. *Communications of the ACM*, *59*(5), 61–69. https://doi.org/10.1145/2890778

Jalendry, S., & Verma, S. (2015). A Detail Review on Voice over Internet Protocol (VoIP). *International Journal of Engineering Trends and Technology*, *23*(4), 161–166. https://doi.org/10.14445/22315381/ijett-v23p232

Jamieson, K., Malloy, M., Nowak, R., & Bubeck, S. (2014). Lil' UCB: An optimal exploration algorithm for multi-armed bandits. *Journal of Machine Learning Research*, *35*(1964), 423–439.

Jatinder, S. (2020). *Advantages of .Net Core. There was increasing demand of software… | by Jatinder Singh | Medium*. https://techjatinder.medium.com/advantages-of-net-core-b605aa76fbb8

Jeroen van Disseldorp. (2017). *Real-time Financial Alerts at Rabobank with Apache Kafka's Streams API*. https://www.confluent.io/blog/real-time-financial-alerts-rabobank-apache-kafkas-streams-api/

Kabari, L. G., & Baah, B. (2015). An Enhanced Data Validation System for a Relational DataBase. *International Journal of Advanced Research in Computer Science*, *2*(6), 27–35. https://doi.org/10.26483/ijarcs.v3i5.1322

Kafeza, E., Chiu, D. K. W., & Karlapalem, K. (2006). Improving the Response Time of Business Processes: An Alert-Based Analytical Approach. In *System Sciences, 2006. HICSS '06. Proceedings of the 39th Annual Hawaii International Conference on* (Vol. 2, pp. 30a-30a). https://doi.org/10.1109/HICSS.2006.218

Kateb, F., & Kalita, J. (2015). Classifying Short Text in Social Media: Twitter as Case Study. *International Journal of Computer Applications*, *111*(9), 1–12. https://doi.org/10.5120/19563-1321

Khan, M. T., & Siddique, A. (2004). Multi-channel integration framework for web services-based business. *Engineering, Sciences and Technology*, 1–4. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1564788

Khavya, K. (2018). *Banking Bot*. *7*, 56–59.

Kim, S.-J., Aono, M., & Hara, M. (2010a). Tug-of-war model for multi-armed bandit problem. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *6079 LNCS*, 69–80. https://doi.org/10.1007/978-3-642-13523-1_10

Kim, S.-J., Aono, M., & Hara, M. (2010b). Tug-of-war model for the two-bandit problem: Nonlocally-correlated parallel exploration via resource conservation. *BioSystems*, *101*(1), 29–36. https://doi.org/10.1016/j.biosystems.2010.04.002

Kim, S.-J., Naruse, M., & Aono, M. (2016). Harnessing the Computational Power of Fluids for Optimization of Collective Decision Making. *Philosophies*, *1*(3), 245–260. https://doi.org/10.3390/philosophies1030245

Kim, S. J., Aono, M., & Hara, M. (2010). Tug-of-war model for the two-bandit problem:

Nonlocally-correlated parallel exploration via resource conservation. *BioSystems*, *101*(1), 29–36. https://doi.org/10.1016/j.biosystems.2010.04.002

Klein, H., & Myers, M. (1999). A SET OF PRINCIPLES FOR CONDUCTING AND EVALUATING INTERPRETIVE FIELD STUDIES IN INFORMATION SYSTEMS 1. *MIS Quarterly*, *23*(1), 67–97.

Knights, D., & Murray, F. (1995). *Managers Divided: Organization Politics and Information Technology Management*. John Wiley Series in Information Systems. https://www.wiley.com/en-us/Managers+Divided%3A+Organization+Politics+and+Information+Technology+Management-p-9780471935865

Kothari, C., Kumar, R., & Uusitalo, O. (2014). Research Methodology. In *New Age International*. https://doi.org/http://196.29.172.66:8080/jspui/bitstream/123456789/2574/1/Research%20Methodology.pdf

Koymans, C. P. J., & Scheerder, J. (2008). Email. In *Handbook of Network and System Administration* (Issue January). https://doi.org/10.1016/B978-044452198-9.50009-4

Krippendorff, K. (1989). *Content Analysis*. *1*, 403–407. http://repository.upenn.edu/asc_papers%0Ahttp://repository.upenn.edu/asc_papers/226

Lai, T. L., & Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, *6*(1), 4–22. https://doi.org/10.1016/0196-8858(85)90002-8

Langlais, R. (2006). Reassembling the Social: An Introduction to Actor-Network-Theory. *Science & Technology Studies*, *19*(1), 93–100. https://doi.org/10.23987/sts.55207

Latour, B. (1987). Science in Action: How to Follow Scientists and Engineers through Society. In *Havard University Press, Cambridge* (Vol. 29, Issue 4). https://doi.org/10.2307/3105094

Latour, B. (1990). Technology is Society Made Durable. *The Sociological Review*, *38*(1_suppl), 103–131. https://doi.org/10.1111/j.1467-954x.1990.tb03350.x

Latour, B. (1996). *Social Theory and the Study of Computerized Work Sites* (pp. 295–307). Springer, Boston, MA. https://doi.org/10.1007/978-0-387-34872-8_18

Latour, B. (2013). Pandora's Hope. In *Journal of Chemical Information and Modeling* (Vol. 53, Issue 9).

Law, J. (1992). Notes on the theory of the actor-network: Ordering, strategy, and heterogeneity. *Systems Practice*, *5*(4), 379–393. https://doi.org/10.1007/BF01059830

Law, J. (1999). After Ant: Complexity, Naming and Topology. *The Sociological Review*, *47*(1_suppl), 1–14. https://doi.org/10.1111/j.1467-954x.1999.tb03479.x

Leedy, P., & Ormrod, J. (2015). *Practical Research Planning and Design* (11th ed.). Pearson Education Limited. www.pearsonglobaleditions.com

Liang, F., Liu, S., Meng, X., & Yang, C. (2011). An integrated multi-channel messaging model supporting for business collaboration. *Proceedings of the 2011 15th International Conference on Computer Supported Cooperative Work in Design*, 532–537. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5960123

Lim, M., Nijdam, N., & Magnenat-Thalmann, N. (2008). A general collaborative platform for mobile multi-user applications. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, 1346–1353.

Lowe, A. (2001). After ANT - An illustrative discussion of the implications for qualitative accounting case research. In *Accounting, Auditing & Accountability Journal* (Vol. 14, Issue 3, pp. 327–351). https://doi.org/10.1108/EUM0000000005519

Ma, J., Nagatsuma, T., Kim, S.-J., & Hasegawa, M. (2019). A Machine-Learning-Based Channel Assignment Algorithm for IoT. *1st International Conference on Artificial Intelligence in Information and Communication, ICAIIC 2019*, 467–472. https://doi.org/10.1109/ICAIIC.2019.8669028

Mbama, C. (2018). *Digital banking, customer experience and bank financial performance : UK customers' perceptions*. *36*, 230–255.

McLean, C., & Hassard, J. (2004). Symmetrical absence/symmetrical absurdity: Critical notes on the production of actor-network accounts. In *Journal of Management Studies* (Vol. 41, Issue 3, pp. 493–519). https://doi.org/10.1111/j.1467-6486.2004.00442.x

Media Labs. (2017). *Using Chat Bots in the Banking Industry*.

Monteiro, E. (2000). Actor-network theory and information infrastructure. *From Control to Drift. The Dynamics of Corporate Information Infrastructure*, *January*, 148–171.

Monteiro, E., & Hanseth, O. (1996). Social shaping of information infrastructure: on being specific about the technology. In J. I. Orlikowski, W. J., Walsham, G., Jones M.R. & DeGross (Ed.), *Proceedings of the IFIP WG8.2 Working Conference*. Cambridge University London and Chapman & Hall.

Muncaster, P. (2015). *Finance Hit by 300 Times More Attacks Than Other Industries*.

Munkhdalai, L., Munkhdalai, T., Namsrai, O. E., Lee, J. Y., & Ryu, K. H. (2019). An empirical comparison of machine-learning methods on bank client credit assessments. *Sustainability (Switzerland)*, *11*(3), 1–23. https://doi.org/10.3390/su11030699

Mutch, A. (2002). Actors and networks or agents and structures: Towards a realist view of information systems. *Organization*, *9*(3), 477–496. https://doi.org/10.1177/135050840293013

Myers, M. D. (1997). Qualitative research in information systems. *MIS Quarterly: Management Information Systems*, *21*(2), 241–242. https://doi.org/10.2307/249422

Nasteski, V. (2017). An overview of the supervised machine learning methods. *Horizons.B*, *4*, 51–62. https://doi.org/10.20544/horizons.b.04.1.17.p05

Niculescu-Mizil, A. (2009). Multi-armed bandits with betting. *COLT 2009 Workshop*, *April*, 133–138.

Oates, B. J. (2006). *Researching Information Systems and Computing* (First Edit). Sage Publications Ltd London EC1Y 1SP.

Office, I. C. (2018). *Guide to the General Data Protection Regulation (GDPR)*.

Omarini, A. (2013). Multichannel distribution in banking: Customers perspectives and theoretical frameworks to increase user acceptance of a multiplatform banking business. *Banks and Bank Systems*, *8*(1), 78–96.

Oniga, B., Denis, L., Dadarlat, V., & Munteanu, A. (2020). Message-based communication for heterogeneous internet of things systems. *Sensors (Switzerland)*, *20*(3). https://doi.org/10.3390/s20030861

Orlikowski, W., & Baroudi, J. (1991). Studying information technology in organizations: Research approaches and assumptions. *Information Systems Research*, *2*(1), 1–28. https://doi.org/10.1287/isre.2.1.1

Oshima, K., Onishi, T., Kim, S.-J., Ma, J., & Hasegawa, M. (2020). Efficient wireless network selection by using multi-armed bandit algorithm for mobile terminals. *Nonlinear Theory and Its Applications, IEICE*, *11*(1), 68–77. https://doi.org/10.1587/nolta.11.68

Palys, T. (2008). *Purposive sampling. In L. M. Given (Ed.)* (Vol. 2).

Pankowski, T., & Pilka, T. (2009). Transformation of XML Data into XML Normal Form. *Informatica*, *33*(4), 417–430.

Pappad, R., & Francesco, P. (2019). *Discriminazione negli algoritmi di apprendimento*. *January*.

Patel, K. K., Patel, S. M., & Scholar, P. G. (2016). Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges. *International Journal of Engineering Science and Computing*, *6*(5), 1–10. https://doi.org/10.4010/2016.1482

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Peterson, M., Gröne, F., Kammer, K., & Kirscheneder, J. (2010). Multi-channel customer management: Delighting consumers, driving efficiency. *Journal of Direct, Data and Digital Marketing Practice*, *12*(1), 10–15.

Pinheiro, P. O., Almahairi, A., Benmaleck, R. Y., Golemo, F., & Courville, A. (2020). *Unsupervised Learning of Dense Visual Representations*. *NeurIPS*, 1–14. http://arxiv.org/abs/2011.05499

Rahimi, N., Maynor, J., & Gupta, B. (2020). *Adversarial Machine Learning : Difficulties in Applying Machine Learning Existing Cybersecurity Systems*. *69*, 40–47.

Robinson, S., Arbez, G., Birta, L. G., Tolk, A., & Wagner, G. (2016). Conceptual modeling: Definition, purpose and benefits. *Proceedings - Winter Simulation Conference*, *2016-Febru*(December), 2812–2826. https://doi.org/10.1109/WSC.2015.7408386

Romero, M., Guédria, W., Panetto, H., & Barafort, B. (2020). Towards a Conceptual Framework for Smart Assessment in Organizations. *Preprints of the 21st IFAC World Congress (Virtual) Berlin, Germany*, *21*, 11090.

Roode, J. (1993). Implications for Teaching of a Process-Based Research Framework for Information Systems. *8th Annual Conference of the International Academy for Information Management,* 10.

Salami, O., & Mnkandla, E. (2020). A Machine-Learning-Based Channel Selection Model for A Multi-Channel Messaging Customer Alert System. *Conference of the South African Institute of Computer Scientists and Information Technologists, Cape Town*, 2.

Salami, O., & Mnkandla, E. (2021). Towards A Machine Learning Enabled Multi- Channel Messaging Framework for Financial Service Institutions : Preliminary Investigations. *International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems, Durban, Kwazulu Natal*, 8.

Salami, O., & Mtsweni, J. (2016a). *Towards A Context-Aware Multi-Channel Messaging Model for African Banks : Preliminary Investigations*. 1–9.

Salami, O., & Mtsweni, J. (2016b). Towards A Context-Aware Multi-Channel Messaging Model for African Banks: Preliminary Investigations. *Conference Proceedings*, 978–1. http://www.ist-africa.org/Conference2016

Salami, O., & Mtsweni, J. (2019). A Context-Aware Multi-Channel Messaging Framework for African Banks : Design and Implementation University of South Africa. *Icsim*, 10. https://doi.org/3305160.3305162

Sankararaman, A., Ganesh, A., & Shakkottai, S. (2019). Social Learning in Multi Agent Multi Armed Bandits. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, *3*(3), 1–35. https://doi.org/10.1145/3366701

Saunders, M., Lewis, P., & Thornhill, A. (2007). Research Methods for Buniess Students. In *Pearson*. https://www.researchgate.net/publication/330760964_Research_Methods_for_Business_Students_Chapter_4_Understanding_research_philosophy_and_approaches_to_theory_development

Saunders, M., Lewis, P., & Thornhill, A. (2012). Research Methods for Business Students Eights Edition Research Methods for Business Students. In *Research Methods for Business Students* (6th Editio). www.pearson.com/uk%0Ahttps://www.amazon.com/Research-Methods-for-Business-Students/dp/1292208783/ref=sr_1_2?dchild=1&qid=1614706531&refinements=p_27%3AAdrian+Thornhill+%2F+Philip+Lewis+%2F+Mark+N.+K.+Saunders&s=books&sr=1-2&text=Adrian+Thornhill+%2F+Phili

Schaffer, P. L., Velasquez, E., Fiorentino, N., Dwyer, K., Hamilton, A., & Barney, K. (2018). *THE IMPACT OF CYBERSECURITY INCIDENTS ON FINANCIAL INSTITUTIONS*.

Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011). RESEARCH ESSAY ACTION DESIGN RESEARCH. *MIS Quarterly Research Essay*, *35*(1), 37–56.

Shabbir, J., & Anwer, T. (2018). *Artificial Intelligence and its Role in Near Future*. *14*(8), 1–11. http://arxiv.org/abs/1804.01396

Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. In *Understanding Machine Learning: From Theory to Algorithms* (Vol. 9781107057). https://doi.org/10.1017/CBO9781107298019

Silhavy, R., Silhavy, P., & Prokopova, Z. (2011). Requirements gathering methods in system engineering. *Recent Researches in Automatic Control - 13th WSEAS International Conference on Automatic Control, Modelling and Simulation, ACMOS'11*, *May 2014*, 106–110.

Singh, Y., Bhatia, P. K., & Sangwan, O. (1999). A review of studies on machine learning techniques. *International Journal of Computer Science and Security*, *1*(1), 70–84.

Sutton, R. S., & Barto, A. G. (2011). An introduction to reinforcement learning. *Decision Theory Models for Applications in Artificial Intelligence: Concepts and Solutions*, 63–80. https://doi.org/10.4018/978-1-60960-165-2.ch004

Tan, Y. (2016). *Lecture 3*. 1–7.

Tatnall, A. (2014). Technological advancements and the impact of actor-network theory. In *Technological Advancements and the Impact of Actor-Network Theory*. IGI Global. https://doi.org/10.4018/978-1-4666-6126-4

Tatnall, A., & Gilding, A. (1999). Actor-Network Theory and Information Systems Research. *Proceedings of the 10th Australasian Conference on Information Systems*, *January 1999*, 955–966. https://doi.org/10.4018/jantti.2009062304

Thompson, W. R. (1933). On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika*, *25*(3/4), 285–294. https://doi.org/10.2307/2332286

Tiwari, S., Ameta, D., & Banerjee, A. (2019). An approach to identify use case scenarios from textual requirements specification. *PervasiveHealth: Pervasive Computing Technologies for Healthcare*. https://doi.org/10.1145/3299771.3299774

Udenze, S. (2020). *AWARENESS AND USE OF WHATSAPP FOR BANKING AND FINANCIAL*. *September*, 0–12. https://doi.org/10.13140/RG.2.2.21501.18404

Universal Postal Union. (2010). *ICTs , New Services and transformation of the Post*.

Vaishnavi, V., & Kuechler, B. (2004). *DESIGN SCIENCE RESEARCH IN INFORMATION SYSTEMS*. *39*(3), 541–564.

Vansh, J. (2019). *Machine Learning Algorithms*. *May*, 210–233. https://doi.org/10.4018/978-1-5225-7955-7.ch009

Venkatesh, V., & Bala, H. (2008). Technology Acceptance Model 3 and a Research Agenda on Interventions. *Journal of Decision Sciences Institute*, *39*(2), 273–315. https://www.mendeley.com/catalogue/technology-acceptance-model-3-research-agenda-interventions-2/

Walsham, G. (1997). Actor-Network Theory and IS Research: Current Status and Future Prospects. In *Information Systems and Qualitative Research* (pp. 466–480). Springer US. https://doi.org/10.1007/978-0-387-35309-8_23

Wand, Y., & Weber, R. (2002). Research Commentary: Information Systems and Conceptual Modeling - A Research Agenda. *Information Systems Research*, *13*(4), 363–376. https://doi.org/10.1287/isre.13.4.363.69

Wang, Z., Zhou, R., & Shen, C. (2018). Regional Multi-Armed Bandits with Partial Informativeness. *IEEE Transactions on Signal Processing*, *66*(21), 5705–5717. https://doi.org/10.1109/TSP.2018.2870383

Waters, K. (2009). Prioritization using MoSCoW. *All About Aile*, *1*(1). http://www.allaboutagile.com/prioritization-using-moscow/

Wilson, J. (2014). *Essentials of Business Research: A Guide to Doing Your Research Project* (J. Seaman (ed.); 2nd Editio). Sage Publications Ltd London EC1Y 1SP.

Wolski, R., Plank, J. S., Brevik, J., & Bryan, T. (2001). Analyzing market-based resource allocation strategies for the computational Grid. *International Journal of High Performance Computing Applications*, *15*(3), 258–281. https://doi.org/10.1177/109434200101500305

Xu, W., & Özsoyoglu, M. Z. (2005). Rewriting XPath Queries Using Materialized Views. *Proceedings of the 31st International Conference on Very Large Data Bases. VLDB Endowment*, 121–132.

Young, P., & Shmuel, Z. (2014). *Handbook of Game Theory*. Amsterdam: Elsevier, North-Holland.

Younis, O., & Fahmy, S. (2004). HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, *3*(4), 366–379. https://doi.org/10.1109/TMC.2004.41

Zhang, H., Zhang, Z., Dai, H., Yin, R., & Chen, X. (2011). Distributed spectrum-aware clustering in cognitive radio sensor networks. *GLOBECOM - IEEE Global Telecommunications Conference*, *December 2013*. https://doi.org/10.1109/GLOCOM.2011.6134296

Zhang, Y., Cai, P., Pan, C., & Zhang, S. (2019). Multi-Agent Deep Reinforcement Learning-Based Cooperative Spectrum Sensing with Upper Confidence Bound Exploration. *IEEE Access*, *7*, 118898–118906. https://doi.org/10.1109/ACCESS.2019.2937108

Zhou, X., Sun, M., Ye Li, G., & Juang, B.-H. (2018). *Intelligent Wireless Communications Enabled by Cognitive Radio and Machine Learning*.

Zhu, H. (2005). *Software Design Methodology: From Principles to Architectural Styles*. Butterworth-Heinemann.

Zhu, J., Song, Y., Jiang, D., & Song, H. (2016). Multi-armed bandit channel access scheme with cognitive radio technology in wireless sensor networks for the Internet of Things.

IEEE Access, *4*, 4609–4617. https://doi.org/10.1109/ACCESS.2016.2600633

APPENDICES

Appendix A: Letter Of Approval

UNISA COLLEGE OF SCIENCE, ENGINEERING AND TECHNOLOGY'S (CSET) ETHICS REVIEW COMMITTEE

24 August 2021

Dear Olusola Oluseun Salami

ERC Reference #: 2021/CSET/SOC/052
Name: Olusola Oluseun Salami
Student #: 49920847
Staff #:

Decision: Ethics Approval from 2021 to 2025 (No humans involved)

Researcher(s): Olusola Oluseun Salami
49920847@mylife.unisa.ac.za

Supervisor (s): Prof E Mnkandla
Mnkane@unisa.ac.za, 072-219-6927

Working title of research:

Modelling a Multi-Channel Messaging Framework: A Machine Learning Approach

Qualification: PhD in Computer Science

Thank you for the application for research ethics clearance by the Unisa College of Science, Engineering and Technology's (CSET) Ethics Review Committee for the above mentioned research. Ethics approval is granted for 5 years.

*The **negligible risk application** was expedited by the College of Science, Engineering and Technology's (CSET) Ethics Review Committee on 24 August 2021 in compliance with the Unisa Policy on Research Ethics and the Standard Operating Procedure on Research Ethics Risk Assessment. The decision will be tabled at the next Committee meeting for ratification.*

The proposed research may now commence with the provisions that:

1. The researcher will ensure that the research project adheres to the relevant guidelines set out in the Unisa COVID-19 position statement on research ethics attached.

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsimile: +27 12 429 4150
www.unisa.ac.za

2. The researcher(s) will ensure that the research project adheres to the values and principles expressed in the UNISA Policy on Research Ethics.
3. Any adverse circumstance arising in the undertaking of the research project that is relevant to the ethicality of the study should be communicated in writing to the College of Science, Engineering and Technology's (CSET) Ethics Review Committee.
4. The researcher(s) will conduct the study according to the methods and procedures set out in the approved application.
5. Any changes that can affect the study-related risks for the research participants, particularly in terms of assurances made with regards to the protection of participants' privacy and the confidentiality of the data, should be reported to the Committee in writing, accompanied by a progress report.
6. The researcher will ensure that the research project adheres to any applicable national legislation, professional codes of conduct, institutional guidelines and scientific standards relevant to the specific field of study. Adherence to the following South African legislation is important, if applicable: Protection of Personal Information Act, no 4 of 2013; Children's act no 38 of 2005 and the National Health Act, no 61 of 2003.
7. Only de-identified research data may be used for secondary research purposes in future on condition that the research objectives are similar to those of the original research. Secondary use of identifiable human research data require additional ethics clearance.
8. No field work activities may continue after the expiry date 31 December 2025. Submission of a completed research ethics progress report will constitute an application for renewal of Ethics Research Committee approval.

Note

The reference number 2021/CSET/SOC/052 should be clearly indicated on all forms of communication with the intended research participants, as well as with the Committee.

Yours sincerely,

RVorster

Mrs R Vorster
Deputy-Chair: Computer Science Computing Ethics Review Subcommittee
School of Computing
College of Science, Engineering and Technology (CSET)
E-mail: rvorster@unisa.ac.za

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsimile: +27 12 429 4150
www.unisa.ac.za

Tel: (011) 471-2208

Pp.

Prof. E Mnkandla
Director: School of Computing
College of Science Engineering and
Technology (CSET)
E-mail: mnkane@unisa.ac.za
Tel: (011) 670 9104

Prof. B Mamba
Executive Dean
College of Science Engineering and
Technology (CSET)
E-mail: mambabb@unisa.ac.za
Tel: (011) 670 9230

Appendix B: Publications

A Machine-Learning-Based Channel Selection Model for A Multi-Channel Messaging Customer Alert System

Salami Olusola

School of Computing, University of South Africa, Johannesburg, Gauteng, South Africa, 49920847@mylife.unisa.ac.za

Mnkandla Ernest

School of Computing, University of South Africa, Johannesburg, Gauteng, South Africa, mnkane@unisa.ac.za

ABSTRACT

Messaging plays a significant role in the sharing of information between Financial Service Institutions and their clients. Technological advancements such as machine learning have provided financial service institutions (FSI's) with the opportunity to reach out to customers in more intelligent ways. Multi-Channel Messaging system (MCM) allows the seamless integration of disparate channels of communications within a single platform for disparate channels (Salami & Mtsweni, 2019). The different modes of communication available, including Short Messaging Service (SMS), Instant Messaging (IM), e-mails and social media messaging platforms (including Twitter, Facebook and WhatsApp), allow businesses to communicate critical and relevant information to their customers. Obtaining information timeously and via the correct means may determine the level of success, or failure, of the business and/or customer venture. Liang *et al.* (2011) implemented an MCM system Integrated Multi-Channel Messaging Model (IM3) system with e-mail and SMS integration, the system relied heavily on a tightly coupled decision-making module for channel assignment using message priority for channel selection. Khan and Siddique (2004) implemented an MCM solution using a web service implementation in retailer/credit card context, using web service open standards. The core of their system implements a static channel selection module, a service gateway and a service integration bus with no preference for customer preference or message priority. The major limitations of these systems are the use of customers' profiled preference to select the channel to use to send the message and also the lack of a self-learning mechanism for determining the channels the customer is most receptive to. This has motivated us in this paper to propose a machine learning enabled channel assignment algorithm for making decision about the most optimal channel at an instance of time in a Multi-Channel Customer Alert System. The decision-making module will integrate heterogenous channels using an Enterprise Service Bus (ESB) layer and utilize machine learning algorithms to determine channel availability, dynamic assignment and monitor customer patterns. Whose advantage/s over the existing systems is the ability to choose channels dynamically , this model also learns the patterns for message delivery and supports an integration layer via a common data channel that is agnostic to each channel integrated this ensures that multiple homogenous channels can share the same message structure.

Selecting the most efficient or optimal channel is difficult. Ma *et al.* (2019) noted that several studies have proposed various method to use in determining the best channel that is available to transmit information. Zhou *et al.* (2006) in their study discussed an assignment of channels in a static way based on the static-network topology assumption methods. Zhu *et al.* (2016) proposed a model for the channel selection problem using as Multi-Armed Bandit (MAB) problem and the study documented the use of algorithms such as Ɛ-greedy, Upper Confidence Bounds (UCB) algorithms to explore dynamical channel assignment while documenting in the study that this approach might have its own drawbacks with overheads in channel selection. Ma *et al*. (2019) elucidates a machine learning channel assignment in a multi-channel system using the Tug-of-War (TOW) model. Kim *et al*. (2010, 2015) and Ma *et al*. (2019) proposed in their studies the TOW dynamically based channel selection / assignment algorithm that implements simple learning procedure that only needs to receive an acknowledge frame for learning procedure with the use of minimal memory and computation capability which gives it an edge over the other algorithms for multi-channel environments where variables change quickly. Our model will expand and maximize this approach to get a quick balance between the exploration-exploitation dilemma for channel selection.

This study aims to examine the problems and challenges encountered in a multi-channel messaging system currently in use by financial service institutions while proposing a model for a multi-channel system that implements a machine learning channel's selection and learning module. The study will focus on the dynamic channel selection approach and integration methods that can make the system effective and efficient for use. Lessons learnt from the design will be further refined to motivate future work in this area.

CCS CONCEPTS

• Information Systems • Computer Science

KEYWORDS

Web Services, Multi-Channel Messaging Systems, Banking, Software, Framework, Platform, Machine Learning, Algorithm, Artificial Intelligence

ACKNOWLEDGMENTS

Special thanks to my supervisor Prof. Ernest Mnkandla for his support on this journey.

REFERENCES

[1] Khan, M. T. and Siddique, A. (2004) 'Multi-channel integration framework for web services-based business', *Engineering, Sciences and Technology, ...*, pp. 1–4. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1564788 (Accessed: 5 July 2014).

[2] Kim, S. J., Aono, M. and Hara, M. (2010) 'Tug-of-war model for the two-bandit problem: Nonlocally-correlated parallel exploration via resource conservation', *BioSystems*, 101(1), pp. 29–36. doi: 10.1016/j.biosystems.2010.04.002.

[3] Kim, S. J., Aono, M. and Nameda, E. (2015) 'Efficient decision-making by volume-conserving physical object', *New Journal of Physics*, 17(8). doi: 10.1088/1367-2630/17/8/083023.

[4] Liang, F. *et al.* (2011) 'An integrated multi-channel messaging model supporting for business collaboration', in *Proceedings of the 2011 15th International Conference on Computer Supported Cooperative Work in Design*, pp. 532–537. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5960123 (Accessed: 5 July 2014).

[5] Ma, J. *et al.* (2019) 'A Machine-Learning-Based Channel Assignment Algorithm for IoT', *1st*

International Conference on Artificial Intelligence in Information and Communication, ICAIIC 2019, pp. 467–472. doi: 10.1109/ICAIIC.2019.8669028.

[6] Salami, O. and Mtsweni, J. (2019) 'A Context-Aware Multi-Channel Messaging Framework for African Banks : Design and Implementation University of South Africa', *Icsim*, p. 10. doi: 3305160.3305162.

[7] Zhou, G. *et al.* (2006) 'MMSN: Multi-frequency media access control for wireless sensor networks', *Proceedings - IEEE INFOCOM*. doi: 10.1109/INFOCOM.2006.250.

[8] Zhu, J. *et al.* (2016) 'Multi-armed bandit channel access scheme with cognitive radio technology in wireless sensor networks for the Internet of Things', *IEEE Access*. IEEE, 4, pp. 4609–4617. doi: 10.1109/ACCESS.2016.2600633.

Towards A Machine Learning Enabled Multi-Channel Messaging Framework for Financial Service Institutions: Preliminary Investigations

Olusola Salami and Ernest Mnkandla
School of Computing
University of South Africa
Johannesburg, South Africa
49920847@mylife.unisa.ac.za, mnkane@unisa.ac.za

***Abstract*— Messaging is essential when organizations dealing with financial customers need to share information. Technological innovations, such as machine learning (ML), have provided financial service institutions (FSIs) with the ability to reach out to consumers in more intelligent ways. Multi-Channel Messaging System (MCM) in use by FSI's enables the seamless integration of disparate channels of communication within a single system. This problem has driven us to develop a machine learning-enabled channel assignment algorithm in the Multi-Channel Messaging System. The decision-making module would incorporate heterogeneous channels using an Enterprise Service Bus (ESB) layer and use machine learning algorithms to assess channel capacity, dynamic assignment, and customer trends. Our framework would extend and maximize this approach to achieve a fast balance between the exploration-exploitation dilemma for channel selection. This research explores the problems and challenges a multi-channel messaging system currently uses by financial services organizations while proposing a multi-channel framework that incorporates a machine learning channel selection and learning module. The analysis concentrates on the complex approach to channel selection and integration methods that can make the system effective and usable. The lessons learned from the design would be further refined to inspire future work in this field.**

Keywords—Multi-channel Messaging System, Financial Service Institutions, Single-Channel Messaging, Tug of War Dynamics, Software Oriented Architecture

I. Introduction

Messaging systems provide a platform for information sharing between organizations and their clients. Technological advancements such as machine learning, the Internet of Things (IoT), the Fourth Industrial Revolution (4IR), etc., have provided financial service institutions (FSI's) the many opportunities to connect with their customers intelligently. A Multi-Channel Messaging system (MCM) allows the seamless integration of different channels of communications within a single platform for different channels [1]. The various channels of messaging communication offered by Mobile Service Providers include Email service, Short Messaging Service (SMS), Social media platforms (such as Telegram, Twitter Direct Message, Facebook Messenger and WhatsApp Instant Messenger, to name a few). These channels allow FSI's to send relevant information to their clients. Their ability to transmit information timeously and via the correct channel can determine their business venture's success or failure.

Liang et al. [2] designed Integrated Multi-Channel Messaging Model (IM^3) system with email and SMS integration as an MCM system. IM^3 utilized a module for channel assignment and decision-making using message priority for channel selection. Khan and Siddique implemented an MCM solution using a web service implementation in retailer/credit card context, using web service open standards [3]. The core of their system implements a static channel selection module, a service gateway, and a service integration bus with no preference for customer preference or message priority. The significant limitations of these systems are the use of customers' profiled choice to select the channel to send the message and the lack of a self-learning mechanism to determine the channels the customer is most receptive to. Selecting the most efficient or optimal channel can be very difficult.

Several studies proposed various methods to determine the best channel available to transmit information [4]. Zhou et al., in their research, implemented a topology using the static-network approach [5]. Zhu et al. [6] proposed designing a solution for selecting multiple channels with a Multi-Armed Bandit (MAB) algorithm. This study used Upper Confidence Bounds (UCB) and Ɛ-greedy machine learning algorithms to implement channels dynamically while reporting in the study that this approach might have drawbacks with overheads in channel selection. Ma et al. [4] implemented a machine learning channel assignment in a multi-channel system using the Tug-of-War (TOW) model [7,8]. Other studies proposed the channel selection logic primarily based on TOW as in [4]. In MCM environments where channel variables change randomly, the TOW model implements a simple learning procedure with homogeneous channels that uses an established frame for learning procedure with minimum memory and computational resources. Therefore, our framework expands and maximizes the TOW approach with an Enterprise Service Bus layer to manage the heterogenous integration of channels to get a quick balance between the exploration-exploitation dilemma for channel selection.

Previous studies on dynamic channel selection have motivated us in this study to propose a machine learning-enabled channel assignment algorithm for making decisions about the most optimal channel at an instance of time in a Multi-Channel Customer Alert System. The decision-making module integrates heterogeneous web channels using an Enterprise Service Bus (ESB) layer with its Application Programming Interface (API). The framework utilizes machine learning algorithms to determine channel availability, dynamic assignment and monitor customer patterns. The use of machine learning gives an advantage over the existing systems in that it affords the ability to choose channels dynamically. This framework learns the models for message delivery and supports an integration layer via a common data channel-agnostic to each channel integrated; this ensures that multiple homogenous channels can share the same message structure.

Machine learning offers automation [9]. AI/ML systems can provide virtual assistants that can improve translation and interpretation abilities of the human language with greater precision than humans. AI Chatbots have also changed the IT ecosystem's landscape. These bots can solve business tasks, determine the best flights by cost and route, personal fitness diet and specialized training, automated hotel booking and banking tasks [10].

This paper looks at the problems and challenges encountered in a multi-channel messaging system currently used by financial service institutions. While proposing a preliminary framework for a multi-channel system that implements a machine learning channel's selection and learning module, the study focuses on the dynamic channel selection approach and integration methods that can make the system effective and efficient for use with an ESB.

II. Related Works or Literature Review

According to [11], there has been a significant change in the retail financial services landscape that has mostly changed the way FSI's generate revenue and profits. Therefore, they have also had to provide their customers with effective and efficient channels to transact their business, i.e. In-branch customer experience or social media channel interactions with FSI agents. How customers feel about using each channel for various business transactions is critical to addressing the customer's experience with the service provider. This method can determine the customer's future behaviour, affecting key business factors such as increased profitability and reduced service costs to the institutions involved. [12] opined that Financial Services Institutions have access to big data about their customer, marketing, operations, accounting, economic activities. Integrating multiple channels for effective communication is essential as most financial services institutions have implemented MCM features in their customer messaging platforms [13].

This literature review aims to introduce machine learning to multi-channel customer alert systems used by FSI's. This section provides a review of the benefits of machine learning, artificial intelligence. These include the potential benefits of using machine learning in channel selection for message delivery to FSI clients to answer the questions below:

- *What requirements are suitable for a machine learning-enabled multi-channel messaging platform?*
- *What machine learning algorithms can be used to seamlessly determine the effective channel/s to use for message delivery on an MCM platform?*

Effectively managing the different communication channels available to customers by businesses is critical in this new age. FSI's need to understand the detailed pattern for each customer segment. This approach will enable the FSI to identify the channels that would be effective and efficient to meet the client's needs. Multiple-Online channel capabilities may facilitate sophisticated customer segmentation and close gaps in customer experience [11]. Multi-channel messaging systems promote sales across boundaries to FSI customers; it supports advertising and enables commerce via the various integrated channels [14] see Fig. 1

Fig. 1. Multi-Channel Messaging System Architecture. Source [14]

Customer's perception of a service offering is primarily dependent on the channel of delivery when implemented correctly. Financial services organizations can enhance their business portfolios by providing the customers' different ways to alert them from their platform by providing various choices that may match their needs [13]. The impact of messages sent at any time to customers should be maximized by ensuring that the customer's preferred channel is used at any time. These preferred channels can be pre-determined using machine learning analytics to ensure message delivery via the customer's appropriate messaging channel based on patterns [15].

Machine learning and artificial intelligence methods have improved over the years. They have significantly improved in many areas of application and domains. These include banking application bots, facial recognition applications, speech recognition, pattern recognition and dynamic machine learning [16]. This artificial intelligence and dynamic machine learning capabilities further ensure that channels are selected efficiently and effectively with integrations capabilities using a channel integration module.

Every channel has a unique messaging format when using heterogeneous systems [17]. This study explored and investigated the channels available, including the format and protocols they support.

1) **Short Messaging Service (SMS)**: SMS messaging formats are well documented in Request For Comments

(RFC) 5724[1]. SMS technology was created out of the Global System for Mobile Communications (GSMC) standards [18]. SMS are limited to a maximum of 160 characters with 7-bit encoding. The default character sets for SMS are defined in the Third Generation Partnership Project (3GPP) when the entire character set is exceeded. It is possible to extend it with other schemes such as 8 or 16-bit encoding. SMS are widely used by FSI's to deliver One Time Passwords (OTP) and to communicate customer-related information online and in real-time. SMS messages have support for plain text format only; it has no support for rich text or multimedia messages [19].

2) **Multimedia Message Service (MMS)**: MMS has support for both plain text and multimedia content. MMS extends the functionality and capability of SMS, and it supports more than 160 characters of text, including rich media like voice, graphics and video [20]. MMS shares the same technical specifications as email with support for Multipurpose Internet Mail Extensions (MIME).

3) **Electronic Mail (Email)**: Email message is based on the Internet Message Format (IMF). IMF format relies on the Simple Mail Transfer Protocol (SMTP) for sending and receiving email messages [21]. IMF is well documented in the Request For Comments Standards (RFC) 5322[2]. Email supports plain text, multimedia, video, audio and attachments specified in the MIME standards. MIME format also allows support for in-line attachment of data within the body of the email message. FSI uses email communication to send communications emails, OTP's, Two Factor Authentication, password reset messages and secured key exchanges to customers on their platforms [22].

4) **Voice Over Internet Protocol (VOIP)**: Voice Over IP allows users to make telephone audio and video calls via the Transmission Control Protocol/Internet Protocol (TCP/IP) using an application [23]. [24] explained that calls via the Internet use a signalling protocol that facilitates communication between the network components while using signalling for session establishment. The critical role of session establishment is further discussed below:

- **Session Establishment**: allows the callee to have the ability to accept, reject or forward a call.
- **User Location**: extracts and displays the geolocation of the caller to the callee.
- **Call Participant Management**: responsible for allowing multiple parties to join or leave an existing call session.
- **Session Negotiator**: manages the set of properties necessary to set up a call.

VOIP is implemented on WhatsApp Call, Skype and Microsoft Teams calls etc. FSI's use these channels for transactional banking with their clients [25].

5) **Twitter Direct Message(Twitter DM)**: Twitter messages support plain text and multimedia, including video and audio available on the web and mobile platforms. Twitter DM currently supports JavaScript Object Notation (JSON) format to store and retrieve messages [26]. FSI uses this channel for automated bot chat and direct message delivery to clients on their platforms.

6) **Facebook Messenger**: Messenger on Facebook uses the traditional restful services via an Application Programming Interface for message exchange between users. Messenger supports Bots implementation in use by FSI's for transaction management and inquiry [27]. Facebook also implements JSON object format for message retrieval and submission on its platform and supports rich text, multimedia, audio, video and plain text. Messenger is available on the web and mobile platforms.

7) **WhatsApp:** WhatsApp Messenger is available on desktop and mobile devices. It implements Extensible Messaging and Presence Protocol (XMPP) with Extensible Mark-up Language XML format for its message format. WhatsApp is a freeware platform that supports End To End Encryption (E2E) of information exchange between users [28]. WhatsApp also supports bot's development used by FSI's to engage the customer on their inquiry or transaction status.

The transaction alert system in use by most financial institutions has been achieved with the multi-channel approach of SMS, emails and social media channels integrated into a customer alert transaction system [29]. This study will review publicly available applications deployed on Android Play-store and Apple iOS store by FSI's. The components of the customer MCM system listed below represent the design in detail [30]:

- MCM-Alert Profiling System: used for manually profiling each customers preference.
- MCM-Alert Polling System: The polling system responsible for checking new messages created on the platform. This module is responsible for dropping the messages in each channel's format
- Integrated Middleware Channel/Enterprise Service Bus: responsible for Decision Making and signalling on the platform.

Fig. 2 below shows the interdependencies between the three systems integrated with the financial services institutions' core systems in a customer transaction alert system.

[1] https://www.ietf.org/rfc/rfc5724.txt
[2] https://tools.ietf.org/html/rfc5322

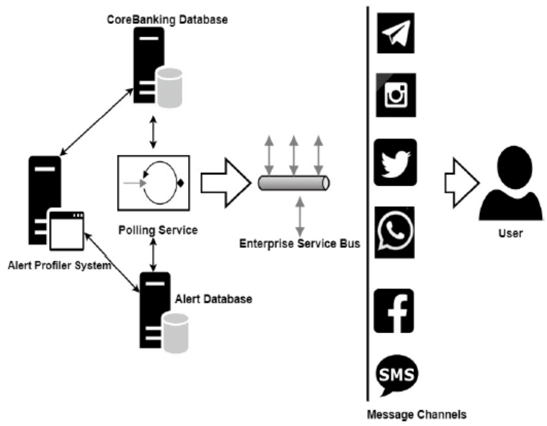


Fig. 2 Typical Current Customer Alert Framework Used by the Applications Source: [31]

This paper explores using a simple machine-learning-based problem Multi-Armed Bandit (MAB) problem [4] as it applies toward channel selection and the ability to derive the optimal solution to the channel selection problem with high probabilities. MAB is expressed as a set of outcomes such as rewards or losses for the player expressed in Bernoulli distributions, and the player will receive a value of reward of failure at an instance of time [32]. An activity or task is used to identify the optimal arm while maximizing its payoff at the same time. [33] documented that in a Multi-Armed Bandit game, the player may select an option set of arms at each time opportunity to play. Each option of the arm, if selected, can result in a reward or loss for the player. [34] elicited further in their work that the goal of a player aims to maximize his compensation (exploration and exploitation) from the options available for selection. Lai and Robbins [35] further explain the trade-offs for the player selection policy's exploration-exploitation approach.

Documented widely in their study, Herbert [36] discussed MAB as an ML problem based on the situation faced by a player that desires to earn maximum reward from multiple slot machines. MAB problem serves to detect using infinite trials the machine slot that should be selected by the player to maximize his reward amount at an instance of time. There is a general assumption that the player has no previous knowledge about any machine/slots' outcome. Each player, understandably, tries as many options as possible and calculates each device/console's payoff, which gives him the best rewards [37]. The player must ensure that they use trade-off while striking a balance between exploitation and exploration strategy, as suggested by [7]. Solving this MAB trade-off is essential as elicited by [4] and further presents algorithms such as Ɛ-greedy, Modified SoftMax Algorithm and Tug of War (TOW) Model.

Furthermore, we focused on applying machine learning to channel routes availability and selection in a Multi-Channel Messaging system framework. Machine learning establishes a framework to facilitate the system's reconfiguration, including its application to a wide range of applications [38]. Learning techniques use the outcomes obtained from the environment's perception and the reconfigurability of the interfaces involved to enhance the available resources. In other words, messaging channels can optimally adapt to the communication environment through learning [39]. The relationship is depicted in **Error! Reference source not found.**.

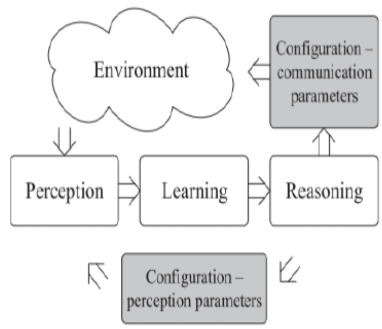


Fig. 3 Machine Learning In Intelligent Communications Source: [39]

ML techniques can be utilized to accomplish a task T associated with a particular experience E to improve the task's performance as measured by the performance metric P by leveraging previous experience E [40]. The parameters T, E and P, determine the outcome of the learning at any instance. Machine learning provides the platform for developing algorithms that give computers the ability to learn. [41] asserted that learning means identifying statistical regularities or other patterns of data. Machine learning algorithms are developed to reflect and adopt a human's approach to learning. Machine learning techniques are further divided into three categories Supervised, Unsupervised and Reinforcement Learning.

a) **Supervised Learning**: Supervised learning completes tasks by learning from any external activity. Besides, each training example contained a series of inputs and desired outcomes to create a mechanism that accurately predicts output for any combination of inputs. [39]. Supervised learning methods have been used to collect and analyse various data [41]. The algorithm produces a function that maps inputs to the desired output. The classification problem is one of the standard formulations of this approach task learning under supervised mode: the learner must learn to approximate a function's action and then map a function into one of many groups by looking at different input, and output data learnt over time. The learning process in a standard machine learning model is categorized into two: training and testing phase [42]. The learning phase consists of collecting samples of training data as input into which attributes are learned by a learning algorithm to create a learning model. In the testing phase, the learning model utilises the execution engine to make predictions for the target output. Labelled data is the result of the learning model that provides the predicted data.

b) **Unsupervised Learning**: This method considers unlabelled samples majorly. The primary objective is to determine the hidden composition of the series of input data. Unsupervised learning uses clusters; observations in the same clusters are more comparable, and observations in separate clusters are less similar [39]. Clustering has a range of uses in innovative multi-channel selection applications. Unsupervised learning methods proposed by [43] have been noted to take advantage of a large quantity of unlabeled data. Younis and Fahmy [44] asserted that the clustering approach suggested in the Hybrid, Energy-Efficient, Distributed Clustering (HEED) method for Ad hoc network sensor groups where the allocation of channels is fixed. The topology management algorithm suggested by [45] solves network creation in a cognitive radio context. The method improved the settings of the cluster to react to changes in the network or environment. A clustered, spectrum-aware clustering technology was developed by [46] to classify energy-efficient clusters and reduce interference in the cognitive radio sensing networks to primary users to effectively aggregate source data while adhering to energy constraints. The objective is to define clusters to decrease communication capacity and minimize squared distances between the cluster centre and each node.

c) **Reinforcement Learning**: Reinforcement learning enables an agent to learn or perform specific tasks through interaction in a complex environment with negligible failure. A stark difference from supervised learning allows the agent to get feedback by merely engaging with the environment and learning itself, making the concept of reinforcement learning very suitable in situations where channel-decision making has to happen with uncertainty. Reinforcement learning is quite beneficial when prior knowledge about the environment is limited. An agent tries to understand and relate to its operational surroundings with considerable uncertainty like Multi-Channel selection and learning in a Customer Alert System [39]. At any time, the agent aims to optimize rewards by exploring and exploiting its operational environment.

Jiang et al. [47] concluded that Multi-Channel Selection and Sensing could be modelled in the form of an Indian Buffet Game. The secondary users are clients, and the main channels are depicted in the restaurant as several dishes. A cooperative method was used to estimate the channel states with Bayesian-Learning to resolve the multi-channel sensing problem, typically using Supervised Learning techniques. Furthermore, In this research, it is expedient to explore the implementation using a hybrid of Supervised Learning (mapping of message endpoints as inputs to the delivery channels as outputs) [39] and Reinforcement Learning using the MAB approach for channel selection [48].

III. Methodology

This research makes use of the Design Science Research (DSR) paradigm [49]. DSR stipulates methods that enable the building of innovative Information Technology artefacts within the organizational setting while defining the procedures for the build, test and run phases of solution design and development [50].

DSR was defined as a methodology that specifies how Information technology artefacts are designed and developed within an organization [51]. The DSR methodology effectively handles disparate problems encountered within an organization and ensures that an effective solution is proffered to address the issue. Our investigation focuses on how an MCM system can be implemented with a dynamic machine learning algorithm for channel selection, a problem that aligns with DSR-problem solving paradigm and has sought to explore this approach for the problem to determine a viable solution.

IV. Preliminary Framework

We considered an existing Multi-Channel Messaging system model in a customer transactional alert messaging system implemented with multiple heterogeneous channels for message delivery, e.g. (Email, Twitter DM, SMS, Facebook and WhatsApp) [31]. This system can only choose one channel to deliver the transaction messages to the customer's device based on a pre-selected channel by the customer. Each MCM channel is directly integrated into an Enterprise Service Bus (ESB) layer, which provides message translation.

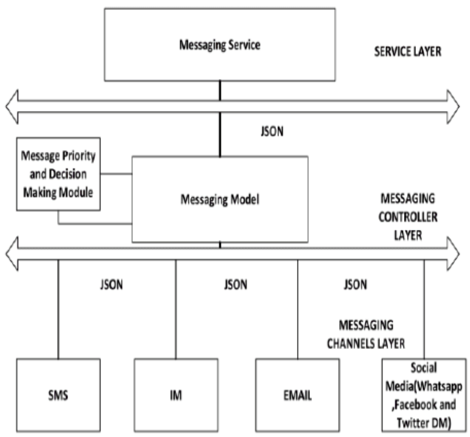


Fig. 4 MCM Framework with ESB Integration Source: [31]

At time *t*, a scheduler randomly polls the messaging service queue to get transaction alert messages done by customers within an instance of time. Once a message is received on the queue, the message is translated to a standard agnostic data type JavaScript Object Notation (JSON) which is a lightweight data format for moving messages around within the platform since each channel implementation accepts different types of message formats, this structure is further converted to each channels messaging data format at the point of delivery.

The ESB relies on the feedback from the message priority and decision module to determine the channel to relay the message to for delivery with this model while heavily relying on the customer preferred/pre-set messaging channel. There is feedback from the channel output that indicates if a message is delivered or not. In the case of non-delivery, the system re-routes the message to the next available channel in the order of manual preference set by the customer.

We designed an enhancement to the current MCM system implementation to enable the channel selection logic to use a machine learning method. The channel selection was modelled as MAB problem and further proposed using extended TOW methods for unsupervised learning and exploration due to its ability to provide optimal channel assignment to use at any time within the system see Fig. 5.

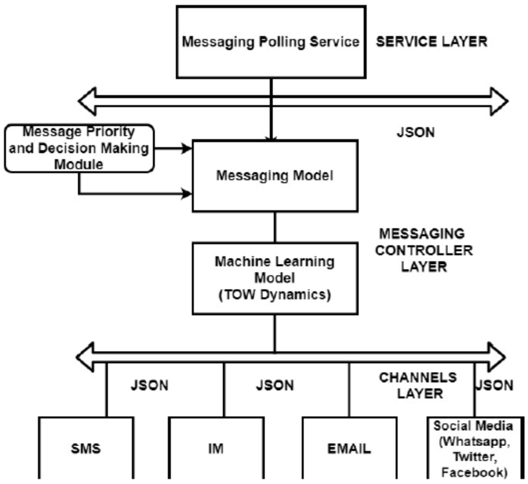


Fig. 5 Proposed MCM Channel Assignment Framework Using TOW Dynamics

The proposed channel assignment module using the TOW dynamics-based channel assignment method. This module is initiated by the Messaging Controller Layer and system parameters like its status and availability to receive new messages. The module polls the messaging channels available, and once availability is confirmed, it will try sending the message via the channel selected. In line with TOW dynamics, an addition operation is performed (for successful transmission and acknowledgement) +1 and a subtraction operation is decreased by −ω. The cycle of the process can be restarted once the initial actions have been completed see the flow chart in Fig. 6

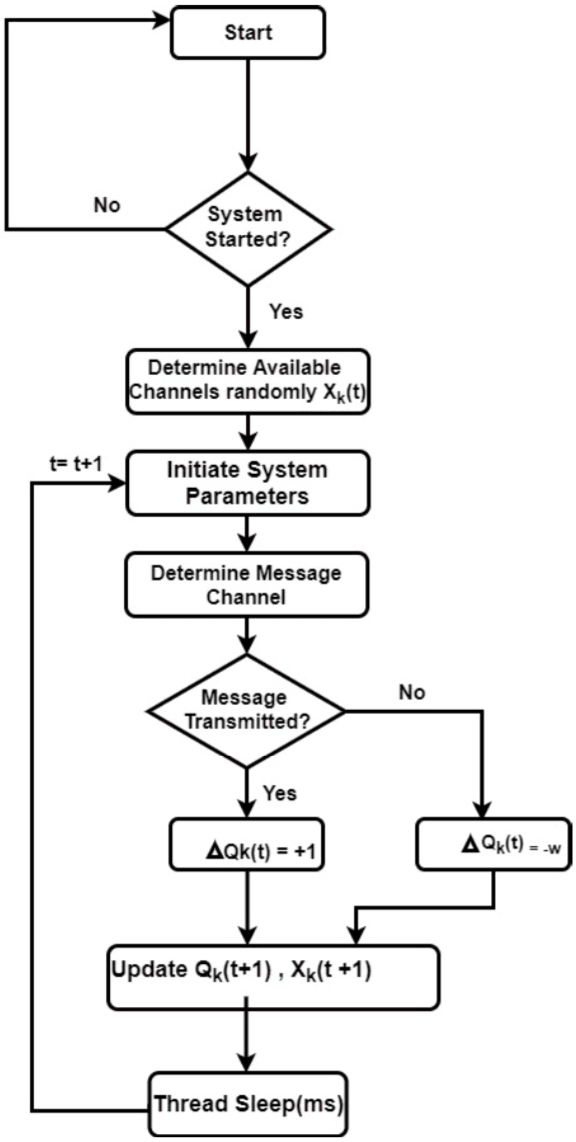


Fig. 6 Flowchart of the Proposed Framework

V. Conclusion

In this paper, we proposed and designed the application of Machine Learning in an MCM system. This model introduced Unsupervised Learning and TOW dynamics into channel selection, management, and behavioural learning. Financial Services Institutions can use this model to develop an ML-enabled MCM customer Transactional Alert System. The MCM system would be channel agnostic and would support both transactional and non-transactional messages.

At a more mature stage of our research, we plan to demonstrate this framework's use using a system prototype.

Feedback from this implementation would be refined further to enhance the framework. We focused on the channel selection algorithms and their impact on the efficiency of the system. The users would have a consistent experience across all communications platforms on the message delivery system due to this implementation.

Another area of concern in the design would be channel availability and how it would react under such conditions. Future research can improve some of the limitations, such as trade-offs between memory-intensive processes when executing channel selection algorithms and service negotiation algorithms between message producers and consumers, which would be considered in future iterations of this research framework.

ACKNOWLEDGMENT

Special thanks and appreciation to Oyenike and Oluwaseun, who assisted in reviewing the first draft of this paper and their insights and inputs on the flow which were adopted. Sincere thanks to Dr Funmilade Faniyi for his valuable external perspective on this research area as it applies to behavioural learning.

REFERENCES

[1] O. Salami and J. Mtsweni, "A Context-Aware Multi-Channel Messaging Framework for African Banks : Design and Implementation University of South Africa," Internation Conference on Software Engineering and Information Management, p. 10, 2019, doi: 3305160.3305162.

[2] F. Liang, S. Liu, X. Meng, and C. Yang, "An integrated multi-channel messaging model supporting for business collaboration," in Proceedings of the 2011 15th International Conference on Computer Supported Cooperative Work in Design, 2011, pp. 532–537.

[3] M. T. Khan and A. Siddique, "Multi-channel integration framework for web services-based business," Engineering, Science and Technology, pp. 1–4, 2004.

[4] J. Ma, T. Nagatsuma, S.-J. Kim, and M. Hasegawa, "A Machine-Learning-Based Channel Assignment Algorithm for IoT," 1st International Conference on Artificial Intelligence in Information and Communication, ICAIIC 2019, pp. 467–472, 2019, doi: 10.1109/ICAIIC.2019.8669028.

[5] G. Zhou, C. Huango, T. Yan, T. He, J. A. Stankovic, and T. F. Abdelzaher, "MMSN: Multi-frequency media access control for wireless sensor networks," Proceedings. - IEEE INFOCOM, 2006, doi: 10.1109/INFOCOM.2006.250.

[6] J. Zhu, Y. Song, D. Jiang, and H. Song, "Multi-armed bandit channel access scheme with cognitive radio technology in wireless sensor networks for the Internet of Things," IEEE Access, vol. 4, pp. 4609–4617, 2016, doi: 10.1109/ACCESS.2016.2600633.

[7] S.J. Kim, M. Aono, and M. Hara, "Tug-of-war model for multi-armed bandit problem," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 6079 LNCS, pp. 69–80, 2010, doi: 10.1007/978-3-642-13523-1_10.

[8] S. J. Kim, M. Aono, and E. Nameda, "Efficient decision-making by volume-conserving physical object," New Journal on Physics., vol. 17, no. 8, 2015, doi: 10.1088/1367-2630/17/8/083023.

[9] F. Hendrik, "Artificial Intelligence and Sustainability," 2020. [Online]. Available: https://www.future-customer.com/artificial-intelligence-and-sustainability/.

[10] J. Shabbir and T. Anwer, "Artificial Intelligence and its Role in Near Future," vol. 14, no. 8, pp. 1–11, 2018.

[11] C. Mbama, "Digital banking, customer experience and bank financial performance : UK customers' perceptions," vol. 36, pp. 230–255, 2018.

[12] J. C. K. Chow, "Analysis of Financial Credit Risk Using Machine Learning," no. April, 2017, doi: 10.13140/RG.2.2.30242.53449.

[13] A. Omarini, "Multichannel distribution in banking: Customers perspectives and theoretical frameworks to increase user acceptance of a multiplatform banking business," Banks and Bank Systems, vol. 8, no. 1, pp. 78–96, 2013.

[14] O. Aaron, "Multi-Channel Marketing: Definition, Data, and a Strategy to Sell Anywhere," 2018. [Online]. Available: https://www.shopify.com/enterprise/multi-channel-marketing. [Accessed: 14-Mar-2020].

[15] Amazon Web Services, "Multi-Channel Messaging System," AWS, 2020. [Online]. Available: https://aws.amazon.com/pinpoint/customer-engagement/multi-channel-messaging/. [Accessed: 14-Mar-2020].

[16] L. Munkhdalai, T. Munkhdalai, O. E. Namsrai, J. Y. Lee, and K. H. Ryu, "An empirical comparison of machine-learning methods on bank client credit assessments," Sustainability (Switzerland)., vol. 11, no. 3, pp. 1–23, 2019, doi: 10.3390/su11030699.

[17] B. Oniga, L. Denis, V. Dadarlat, and A. Munteanu, "Message-based communication for heterogeneous internet of things systems," Sensors (Switzerland), vol. 20, no. 3, 2020, doi: 10.3390/s20030861.

[18] J. Brown, B. Shipman, and R. Vetter, "SMS: The short message service," Computer Journal (LongBeach California)., vol. 40, no. 12, pp. 106–110, 2007, doi: 10.1109/MC.2007.440.

[19] E. R. Adagunodo and O. Bamidele, "SMS Banking Services: A 21 st Century Innovation in Banking Technology," 2007.

[20] M. Hasan, N. Haron, and N. S. Md Yazid, "Development of multimedia messaging service (MMS)-based examination results system," Proceedings of the 9th WSEAS International Conference on Applications of Computer Engineering, ACE '10, pp. 157–163, 2010.

[21] C. P. J. Koymans and J. Scheerder, Email, no. January. 2008.

[22] A. Anitha, M. Varalakhshmi, A. Mary Mekala, Subashanthini, and M. Thilagavathy, "Secured cloud banking transactions using two-way verification process," International Journal of Civil Engineering and Technology, vol. 9, no. 1, pp. 531–540, 2018.

[23] Y. Fayyaz, "The Evaluation of Voice-over Internet Protocol (VoIP) by means of Trixbox," International Journal of Natural and Engineering Sciences, vol. 10, no. December, pp. 33–41, 2016.

[24] S. Jalendry and S. Verma, "A Detail Review on Voice over Internet Protocol (VoIP)," International Journal of Engineering Trends and Technology, vol. 23, no. 4, pp. 161–166, 2015, doi: 10.14445/22315381/ijett-v23p232.

[25] A. DeAngelis, "6 Reasons To Implement VoIP for Financial Institutions," 2019. [Online]. Available: https://blog.votacall.com/voip-for-financial-institutions. [Accessed: 15-Nov-2020].

[26] F. Kateb and J. Kalita, "Classifying Short Text in Social Media: Twitter as Case Study," International Journal of Computer Applications, vol. 111, no. 9, pp. 1–12, 2015, doi: 10.5120/19563-1321.

[27] Media Labs, "Using Chat Bots in the Banking Industry," 2017.

[28] S. Udenze, "AWARENESS AND USE OF WHATSAPP FOR BANKING AND FINANCIAL," no. September, pp. 0–12, 2020, doi: 10.13140/RG.2.2.21501.18404.

[29] M. Cortiñas, R. Chocarro, and M. L. Villanueva, "Understanding multi-channel banking customers," Journal of Business Research., vol. 63, no. 11, pp. 1215–1221, 2010, doi: 10.1016/j.jbusres.2009.10.020.

[30] Jeroen van Disseldorp, "Real-time Financial Alerts at Rabobank with Apache Kafka's Streams API," 2017. [Online]. Available: https://www.confluent.io/blog/real-time-financial-alerts-rabobank-apache-kafkas-streams-api/. [Accessed: 26-Apr-2020].

[31] O. Salami and J. Mtsweni, "Towards A Context-Aware Multi-Channel Messaging Model for African Banks: Preliminary Investigations," IST Africa Conference Proceedings, pp. 978–1, 2016.

[32] L. Belzner and T. Gabor, "QoS-Aware multi-Armed bandits," Proceedings - IEEE 1st International Workshops on Foundations and Applications of Self-Systems, pp. 118–119, 2016, doi: 10.1109/FAS-W.2016.36.

[33] Y. Chen, W. Lu, X. Chen, L. Tang, F. Rao, Q. Wang and L. Zhang, "Location aware messaging-integrating LBS middleware and converged services," Proceedings of the 2005 IEEE International Conference on e-Business Engineering, pp. 419–426, 2005.

[34] A. Niculescu-Mizil, "Multi-armed bandits with betting," COLT 2009 Workshop, April, pp. 133–138, 2009.

[35] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," Advances in Applied Mathematics, vol. 6, no. 1, pp. 4–22, 1985, doi: 10.1016/0196-8858(85)90002-8.

[36] R. Herbert, "Some Aspects of the Sequential Design of Experiments," Bulletin of the American Mathematical Society., vol. 58, no. 5, pp. 527–535, 1952, doi: 10.1090/S0002-9904-1952-09620-8.

[37] W. R. Thompson, "On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples," Biometrika, vol. 25, no. 3/4, pp. 285–294, Mar. 1933, doi: 10.2307/2332286.

[38] M. Alsheikh, S. Lin, D. Niyato, and H. P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," IEEE Communications and Survey Tutorials, vol. 16, no. 4, pp. 1996–2018, 2014, doi: 10.1109/COMST.2014.2320099.

[39] X. Zhou, M. Sun, G. Ye Li, and B.-H. Juang, "Intelligent Wireless Communications Enabled by Cognitive Radio and Machine Learning," 2018.

[40] C. Bishop, "Pattern recognition and machine learning," in Pattern recognition and machine learning, vol. 9, no. 4, 2006, pp. 257–261.

[41] V. Nasteski, "An overview of the supervised machine learning methods," Horizons.B, vol. 4, pp. 51–62, 2017, doi: 10.20544/horizons.b.04.1.17.p05.

[42] Y. Singh, P. K. Bhatia, and O. Sangwan, "A review of studies on machine learning techniques," International Journal of Computer Science and Security, vol. 1, no. 1, pp. 70–84, 1999.

[43] P. O. Pinheiro, A. Almahairi, R. Y. Benmaleck, F. Golemo, and A. Courville, "Unsupervised Learning of Dense Visual Representations," no. NeurIPS, pp. 1–14, 2020.

[44] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," IEEE Transactions on Mobile Computing, vol. 3, no. 4, pp. 366–379, Oct. 2004, doi: 10.1109/TMC.2004.41.

[45] T. Chen, H. Zhang, G. M. Maggio, and I. Chlamtac, "Topology management in CogMesh: A cluster-based cognitive radio mesh network," IEEE International Conference on Communications, pp. 6516–6521, 2007, doi: 10.1109/ICC.2007.1078.

[46] H. Zhang, Z. Zhang, H. Dai, R. Yin, and X. Chen, "Distributed spectrum-aware clustering in cognitive radio sensor networks," GLOBECOM - IEEE Global Telecommunications. Conference, no. December 2013, 2011, doi: 10.1109/GLOCOM.2011.6134296.

[47] C. Jiang, Y. Chen, and K. J. R. Liu, "Multi-channel sensing and access game: Bayesian social learning with negative network externality," IEEE Transactions on Mobile Computing, vol. 13, no. 4, pp. 2176–2188, 2014, doi: 10.1109/TWC.2014.022014.131209.

[48] Z. Wang, R. Zhou, and C. Shen, "Regional Multi-Armed Bandits with Partial Informativeness," IEEE Transactions on Signal Processing, vol. 66, no. 21, pp. 5705–5717, 2018, doi: 10.1109/TSP.2018.2870383.

[49] A. R. Hevner, S. T. March, J. Park, and S. Ram, "DESIGN SCIENCE IN INFORMATION," vol. 28, no. 1, pp. 75–105, 2004.

[50] B. J. Oates, Researching Information Systems and Computing, First Edit. London: Sage Publications Ltd London EC1Y 1SP, 2006.

[51] M. K. Sein, O. Henfridsson, S. Purao, M. Rossi, and R. Lindgren, "RESEARCH ESSAY ACTION DESIGN RESEARCH," MIS Quarterly Research Essay, vol. 35, no. 1, pp. 37–56, 2011.

A Design For A Machine-Learning-Enabled Multi-Channel Messaging Framework for Financial Service Institutions

Olusola Salami and Ernest Mnkandla
School of Computing
University of South Africa
Johannesburg, South Africa
49920847@mylife.unisa.ac.za, mnkane@unisa.ac.za

***Abstract*— Financial Services Institutions (FSIs) utilize multi-channel messaging systems (MCM) to integrate diverse consumer communications channels into a unified system for the benefit of their customers. However, the multi-channel service has an extra overhead coordinating channel selection to identify which channel is available to send messages to customers because of the disparate nature of the integrated channels. Machine learning can be used to determine channels for communication in a customer alert system used by financial services institutions (FSIs). This work offers a machine-learning-enabled channel selection and management method based on Tug Of War (TOW) algorithm. The framework proposed will be based on the TOW dynamics algorithm. TOW uses a modified learning technique based on addition and subtraction's reward/loss principle. Furthermore, we demonstrate the efficacy of this strategy by prototyping the proposed framework in a customer alert system termed the ML-Enabled MCM system. Messaging channels such as SMS, Email, Twitter Direct Messages, Instagram Direct Messages, Telegram, and Facebook Direct Messages were integrated with an Enterprise Service Bus (ESB) responsible for channel management. The new design reveals that the machine learning-enabled MCM system can learn and adapt quickly in response to channel feedback and make quick decisions about which channels a customer can receive messages on while maintaining balance within the system. The proposed framework will be relevant to the event and message processing products offered by FSIs.**

Keywords*— *Channel Assignment, Messaging, Banking, Event Driven Systems, Service Oriented Architecture, TOW, Machine Learning, Decision Based Systems

I. Introduction

In the present era, the importance of using customer transactional alert systems by FSI's cannot be overstated in the financial services industry. In the early '80s, businesses only contacted customers via telephone (landlines) and post office mails[1]. Subsequently, the use of Emails [2] in banking evolved with SMS, which resulted in quicker and more efficient communication channels. Web 2.0 technologies, such as social media messaging channels, commonly used by customers [3], offer better integration options between Third-party systems. This integration layer allows the configuration of message types, levels of importance, and priority across all channels [4].

However, current implementations of MCM systems that incorporate different messaging channels embedded for information delivery to customers are limited to using customers' pre-profiled data to select or learn a channel for message delivery. This method is entirely manual, irrespective of the message type, importance or priority level and, most importantly, the customer's most receptive channel for message delivery. Hence, there is a need to research and propose a framework for a machine learning and channel selection module. This module will be embedded within the system to allow dynamic channel selection and machine learning using an exploitation and an exploratory approach for channel selection. Channel selection remains an integral part of an MCM system. Hence, it is imperative to implement a dynamic and self-aware system with machine learning capabilities to instantly determine available channels for FSI clients [5][6].

Omarini [7] defined a channel as the medium through which FSI's can reach their customers and vice versa (inbound *or* outbound). This process includes *web-based* technologies such as the Internet, email, Facebook, Telegram, WhatsApp, Blackberry messenger and non-web-based technologies such as call centres, SMS and digital TV [8]. Individual channels are better suited to meeting specific and contextual user requirements. Due to the nature of their operations and services, FSIs are extremely innovative and adopt new technology to delight their customers.

It is challenging to choose the most efficient or ideal messaging channel. Different studies have offered several ways for determining the optimal channel for transmitting information [6]. Zhou et al [9], used the static-network approach to implement a channel selection topology, their work proposed a design for the messaging channel selection problem a solution with the Multi-Armed Bandit (MAB) approach [10]. As part of the study, the researchers used algorithms such as UCB and Ɛ-greedy technique to provide channel selection dynamically and reported that the approach may have limitations with regards to channel selection overheads.The study in [6] designed a artificial intelligence/ML solution using channel assignment in an MCM system using the TOW approach [5][6].

On the other hand [11] proposed the TOW dynamic based channel selection/assignment algorithm in their studies. A basic learning learning procedure with homogeneous channels is implemented by the TOW model, which makes use of a frame received successfully from the learning module, The process requires low memory and computational capabilities, making it appropriate for MCM contexts in which channel variables fluctuate periodically. With the addition of a Service Bus layer to manage the heterogeneous integration of channels, our proposed design will broaden and optimise this approach in order to provide a quick an equilibrium between the exploration-exploitation dilemma in selecting available channels.

This paper examines financial service institutions' challenges and issues with the use of a MCM system. The research is focused on a dynamical channel decision strategy, machine learning with integration methods that could make a MCM system more functional with an ESB, as well as a framework that includes a learning module with artificial intelligence and machine learning.

II. Related Works

Machine learning (ML) has increasingly become popular with proven success in learning complicated models and patterns in systems [12]. Machine learning supports creating models that immediately recognize patterns, classify input data into different categories, and make predictions [13]. AI and machine learning are a constantly evolving technology research fields. This research aims to integrate machine learning algorithms for channel selection in a FSI transaction alert system used by their clients[6]. Shalev-Shwartz and Ben-David [14] explored the linkage between machine learning and machine learning algorithms. They further documented that training data sets are required as inputs for machine learning algorithms to generate an output for other downstream computing outputs. ML enables computer systems to learn directly from training datasets and experience. [15] opined several factors in selecting a suitable algorithm to address this challenge.

The MAB dilemma [5][6] is applicable to the channel assignment issue and the potential to develop the ideal result with a strong likelihood. MAB is defined as a set of possible outcomes for a player, described in Bernoulli distributions, and the participant will get a value of reward or loss at any point in time [16]. Lai and Robbins [19] in their work explains the trade-offs associated with the exploration-exploitation strategy of the player selection policy. An action is performed to find the ideal arm while also maximizing its reward. At each chance to play, the player may decide to choose an optional set of arms [17]. Each arm alternative, if picked, might lead to reward or a loss for the player. In their research Niculescu-Mizil opined that a player's objective is to increase his reward (exploration and exploitation) from the existing possibilities [18].

Based on a player's desire to improve his or her payout from the several slot machines, Herbert described MAB as a machine learning problem [20]. In solving the challenge, it is essential to identify the ideal machine slot for a player in order to boost his/her payoff at a given point in time. The player is assumed to have no prior knowledge of the result of any of the slots. In order to get the most out of the game, each player instinctively attempts as many possibilities and evaluates each machine's payout [21]. Hence, each player has to determine a trade-off between the exploitation and exploration strategy, as outlined in [5]. The TOW, enhanced and modified Ɛ-greedy, modified softMax algorithm are examples of algorithms that can help in resolving the MAB tradeoff, which is addressed in [6].

There are different machine learning algorithms designed to address the MAB problem discussed below:

1) **The Enhanced and Modified Ɛ-greedy Algorithm**: The Ɛ-greedy algorithm ensures that a player displays a greedy action when selecting an option but with probability Ɛ that the player will pick an activity at random in effect. This action ensures that the player has a simple way of balancing exploration and exploitation [22]. Gutowski et al [23] documented in their study that the objectives include maximizing the significant rewards earned while minimizing the loss (regrets). However, in this paper, the "average accuracy rate" as documented by [22] will be used based on interest in a short-term behaviour and not a logarithmic approach for long term behaviour. A player can make a random selection between two options(A/B), with a probability value Ɛ otherwise, referred to as a greedy-action with the estimates expressed in Equations 1 and (2) expressed as a probability of (1- Ɛ). A greedy-action means that the player chooses option A if ($Q_A > Q_B$) or selects option B if ($Q_A < Q_B$.)

Ɛ: *random selection* and 1 – Ɛ: *greedy action based on estimates*

$$Q_{(t)} = \frac{Number\ of\ Rewards\ from\ A}{Number\ of\ A\ Selections} \quad (1)$$

$$Q_{(t)} = \frac{Number\ of\ Rewards\ from\ B}{Number\ of\ B\ Selections} \quad (2)$$

Equations 1 and (2) defines the probalities for reward P_A and P_B. This study uses time-dependent probability Ɛ(t) expressed in (3)

$$\epsilon(t)\frac{1}{1+r.t} \quad (3)$$

τ is defined as the parameter that controls the decay rate.
In this case, the player makes a random selection of an arm expressed as probability (Ɛ) or implements the greedy-action based on estimated values of (Q_1, Q_2,………Q_M) using a probability 1 – Ɛ with $Q_K(t)$ ($K=_{1,2,\ \ldots,m}$) specified in the formula below as:

$$Q_{(t)} = \frac{Number\ of\ Rewards\ from\ k}{Number\ of\ k\ Selections} \quad (4)$$

2) **The Modified Softmax Algorithm**: The modified softmax algorithm, also referred to as multinomial logistic regression, is a generalization of logistic regression that allows handling multiple classes and is quite useful for neural networks where non-binary classification is not needed. According to Gao and Pavel [24], the modified SoftMax ensures a player can balance the opportunities available for the exploitation and exploration techniques within the MAB problem. This technique guarantees the context of decision-making while ensuring that every strategy in a player's possession has a chance of being explored, unlike some other logic implemented such as the enhanced Ɛ-greedy algorithm [25].

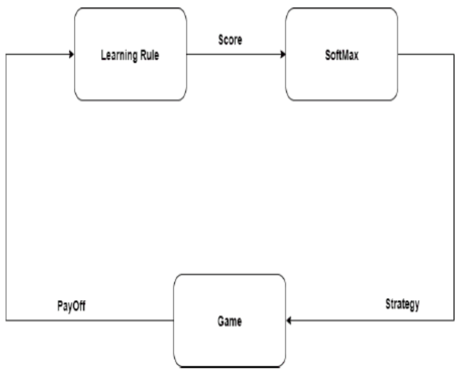


Fig. 1. The modified softmax logic for the explorative and exploration strategy: Source [25]

The TOW algorithm complements and enhances the Modified SoftMax and Modified Ɛ-greedy algorithms by providing a suitable option for dynamic channel selection, the core of the proposed framework in this research. TOW model further implements simple learning using acknowledgement messages received from successful messages delivered by each channel to users while providing minimal memory and computation capabilities for a heterogeneous MCM system.

Ma et al [5] [6] described the TOW model as a unique machine learning method with extended use for the MAB problem. Kim et al [26] explored the MAB problem efficiently with fluid dynamics in cylinders. They further stated that the model used by TOW provides a significant efficiency than other algorithms, for example, the the modified softmax algorithm and the enhanced Ɛ-greedy algorithm. Fig. 2 depicts the volume-conservation law, which was used by Ma et al [6] in their work to intelligently select communication channels in massive IoT.

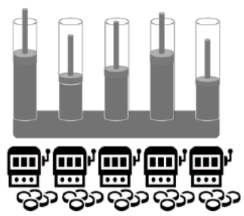


Fig. 2. The TOW Dynamics: Source [6]

Kim et al [26] demonstrated Tug of War with fluid in the tube. TOW ensures that the fluids' density remains constant as the fluids move with velocity, as shown above in Fig. 2. X_k is equivalent to the movement of the arm $\underline{k}$ from an original position. The value of the parameter is expressed as $k \in Arm_1, Arm_2, \ldots Arm_K$. $Q_k(k \in Branch_1, Branch_2, \ldots Branch_K)$ this is represented and calculated below:

$$Qk(t) = \Delta Qk(t) + Qk(t)(t+1), (0 < \alpha < 1) \quad (5)$$

Where the above is expressed as +1 (reward) or ω (failure) according to the selection outcome, the value ω is defined as the weighting parameter. The parameter α controls how previous information can be quickly forgotten. The change in value X_k is derived from the difference in the formula below in (6).

$$Xk(t+1) = Qk(t) - \frac{1}{N-1} \sum_{i \neq k} Qi(t) + osc \quad (6)$$

The fluctuation the fluid is subjected to is expressed as a random value *osc*. At the moment machine k is selected at time t [26], $X_{k(t)}$ is incremented by +1 denoting a reward for the player. Incase of a loss, then $-\omega$ is added to $X_{k(t)}$. Following these actions, the interface's levels shift in line with the volume conservation law, and this law determines the next step taken by the player. According to Ma et al [6], TOW dynamics supports an efficient cognitive search method while maximizing that channel selection is as correct as possible.

III. System Framework

Consider a machine learning-enabled MCM Customer Alert System integrated with an Enterprise Service Bus (ESB). The MCM system can send messages to multiple disparate message channels such as Whatsapp, Twitter Direct Messages, Instagram Direct Messages, Telegram, and Facebook Messenger. Still, it can only send messages to one channel at a time to a customer. Each channel is independent, with messages transmitted to customers' mobile terminals using an agnostic light-weight data format Javascript Object Notation (JSON). As a result, it is essential that the channel delivers feedback to ensure the ML-enabled MCM system's reliability and accessibility. The TOW algorithm, which manages the system state, availability, and selection, is at the heart of the ML-enabled MCM framework. When a customer completes a transaction, or an FSI has to communicate with a client, they send messages translated into JSON format and then sent through any of the various channels. During operation, core messaging processor/service workers receive messages and make an API request to the TOW worker service to discover which channel is accessible from the customer's pre-selected channels. Following a successful response, the TOW algorithm returns the channel available, and the data payload is handed over to the ESB. The ESB service also determines whether the channel is accessible or is currently busy processing another message. If the channel is busy, the ESB waits for a configurable interval before accessing it again. Suppose the channel successfully delivers the message payload to the customer's mobile terminals successfully. In that case, the ESB receives the acknowledgement message and advises the TOW worker service to update the channel state. Fig. 3 describes the flow.

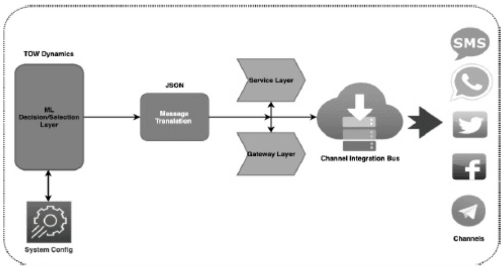


Fig. 3 System model using ESB integration in MCM System

IV. IMPLEMENTATION AND DESIGN RESULTS

Multi-Channel Messaging systems that integrate several disparate channels will benefit from a novel machine-learning-based assignment of channels using the TOW solution. Each messaging disparate messaging -channel uses the TOW algorithm channel selection technique to achieve successful delivery rates and availability.

First, let us consider a scenario comprising a set of message producing nodes in an MCM customer alert system, *P*. Each member produces varied messages ß, that can be translated and transmitted. The members of *P* are assumed to be operating under the MAB principle. Subsequently, consider a group of message consumers or channels, *C*, which attempts to receive and send the message ß to each of its members, at a time based on its availability. If p_i is a node in *P* and c_j is a node in *C*, n_{ij} is used to determine the number of messages provided by p_i to c_j. The total number of ß provided by at an instant of time, its message *throughput*, t_{pi}, is defined as:

$$t_{(pi)} = \sum_{j=1}^{|C|} n_{(ij)} \tag{7}$$

The first objective to be realized from this setup is to be able to provide at an instance of time a balanced throughput within the system that ensures that each message producing node in, *P*, has an available channel, *C*, to handle and transmit the message from a group of disparate channels integrated for message transmission. In this case, a channel resource allocation and selection configuration given the provision of ß by the nodes in *P* at a given instant of time can be expressed using a vector T_s $<t_{s1}, t_{s2}, \ldots, t_{sm}>$, where $m = |P|$. Measuring the throughput of the system is essential at this point, as well as the main objective of ensuring effective and efficient channel selection/allocation using the TOW approach in a fully decentralized fashion with no control from a central controller within the system with a machine-learning method of the channel [6]. Zhou et al [27] discussed the need for reinforcement learning essential to keep the state of success or failure of each channel's response. When a message is sent via the system by a message Producer *P* [26], the throughput t_{pi} is incremented by +1, hence the channel *C* can accept and transmit the message termed as a reward to the message producer. Otherwise, $-\omega$ is added to t_{pi} defined below as:

$$Reward = \sum_{j=1}^{|C|} n_{(ij)} + 1 \tag{8}$$

$$Loss = \sum_{j=1}^{|C|} n_{(ij)} - w \tag{9}$$

A tug-of-war channel selection mechanism is used to decide how messages ß are routed between message producer nodes to messaging channel nodes in an ML-enabled MCM customer alert system. At an instant of time, a message producing node, p_i summation of P, via a broadcast mechanism sends out available messages. Messages ß can be accepted or rejected by each resource handler, a channel in this case, only if they are not busy processing other messages in the queue, overloaded or completely unavailable [28]. The system iterates with channels that have shown their reliability over time using the previously-stored reward values [27].

The networked environment and other external factors are not considered at this phase because they would make evaluating and interpreting the framework underlying behaviour unnecessarily complicated. The following simplifying assumptions are made with this in mind [27]:

1. That the system is initialised and runs in discrete time steps synchronously,
2. every message producer produces one message ß, at each step of time,
3. the process of creating messages ß is immediate in nature, to the point that it does not interfere with the model's mechanism.
4. that each message producer has the capacity to create messages ß and make it available to all the channels *C* available in at each instance of time,
5. that there is uniform network connectivity within the platform.

The first two assumptions are required to understand and analyze the underlying system at this proposal stage. However, their use has not been thoroughly investigated. There appears to be no compelling reason for them to change their current behaviour. This paper's exploitation and exploration methods are described in assumptions above in (3) and (4), respectively [24]. Finally, the assumption (5) model's network conditions are described by [29] in the G-Commerce system.

At each time step, each channel, if available to receive messages, may receive any number of messages ß from any number or message producers in *P*, subject to the constrain that the total messages produced and transmitted per time-step are exactly one message (as per assumption 2). If there are no messages produced by *P*, the channels may instead be in a waiting state with no messages to process and in a ready/waiting state. In equation 10, these constraints mean, therefore that

$$\sum_{j=1}^{|p|} \in| \ +1| - w \ for \ all \ C_j \in \ of \ C \tag{10}$$

Both the message producers and channels accrue a reward or loss for the interactions within the MCM system. This is deemed to be the value they associate with delivering messages sent to them successfully for channels. From a channel's perspective, if a message producer transmits a message and it would not lead to failure in delivery at that instant of time, then the operation is described as acceptable. P_{cj} is used to describe the subsets of P, This comprises of message producers whose messages are deemed acceptable by channel C_j.

Channels will not receive messages when they are unavailable due to being busy or out of service. The possibility of complex channel selection strategies means that there will not be a direct mapping between message producers and channels at any time [30]. [30] described this with Markov Decision Process in which channels are represented as factors in space with reward functions. Channel states are determined from previous interactions within the system in line with the MAB problem [27].

Message producers receive a reward based on the response received from a channel that delivers the message successfully. The reward of each message producer P_j is defined as R_{pi}. In its simplest form, the reward from the transmission of message ß in (11) and (12):

$$R_{pi} = \sum_{j=1}^{|C|} t\frac{\beta}{pi}\, n_{ij} \tag{11}$$

Or alternatively

$$R_{pi} = \frac{\beta}{pi}\, Xt_{pi} \tag{12}$$

Equation 12 shows that a message producer that wishes to maximize its reward would aim to quickly check the status of the channel and its previous successful delivery history and promptly decide to either engage and not to wait endlessly for an unresponsive channel. As we have seen from the channel's behaviour, throughput for message producers is determined mainly by the channel's performance.

This paper proposed a technique of selecting messaging channels using the MAB problem while leveraging on the exploration and exploitation strategy. As described earlier, the TOW algorithm addressed the MAB problem. Fig. 4 depicts the concept of the channel selection strategy. The message producers that has the ability of accessing the performance of the channel cj, where j is the selected channel. c_j can be any indexes of the performance of the individual messaging channel c_j.

Based on the TOW algorithm. It determines variables such as, delay, throughput or other variables of each channel. The selection algorithm then accesses the possibility to chooce of the reward or the loss is to be added by evaluating the obtained performance of channel j . If there is a reward in a transmission by a message producer, it updates the estimator as *Q+1,* otherwise, it updates the estimator as *Qj - 1.* The algorithm in the machine learning-enabled multi-channel system is described below:

1. Initiate monitoring of the performance of each channel in the MCM system c_j. All channels performance are checked at least at an interval of time.
2. Notify and change the values of Q_j,T_{pi}, and all Rewards and Losses by observing the throughput performance of the channels C_j, by Eq. (10) and Eq. (11).
3. At an instant of time make a selection of a channel C_j, with the highest T_{pi}.
4. Examine the effectiveness of the chosen channel and decide whether to assign a reward or a loss value..
5. At this step revert to 2 and continue processing

Overall, the channel selection and assignment challenge was modelled as a MAB problem to offer a solution that is both effective and efficient in channel allocation using machine learning techniques.

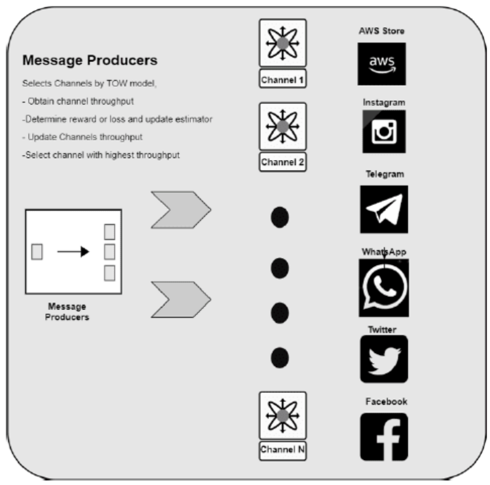


Fig. 4. Proposed ML-Enabled MCM Framework based on TOW Dynamics

The ML-enabled MCM framework is implemented in a customer transaction alert system is as a 3-tier application. The first tier of the solution consists of a presentation layer or user interface layer, the second tier is the application layer which consists of the application layer, data translation, message service bus, message routing, API layer, machine learning and channel selection logic and the third tier is the database layer that stores information about system configuration, channel state management.

Fig. 5 presents the architectural design of the artefact.

The user interface setup and management interface allow FSI designated agents to log in to a web-based interface to profile a customer on the platform. The application enables the user to assign message channels to customers based on initial preference, manage customer information and setup of each channel's profile. For example, a typical setup requires the user to capture a unique client identifier from the FSI's core banking platforms and then link this to a telephone number for SMS, Telegram, WhatsApp message delivery, email address for email message delivery, Twitter handle for Twitter direct message and Facebook identity for Facebook messenger delivery. The user interface allows the FSI agent to manage the customer life cycle, generate relevant reports and monitor the system performance during operation.

The application layer handles and processes all the requests from the web-based interface. This layer is further broken down into an API managed layer that exposes secured services via a REST interface specification for third party integration. The layer also includes worker services module executing as background services responsible for channel state management, machine learning and channel selection logic and an integration with a customized message service bus module responsible for the integration of the support messaging channels, such as SMS, Email, Facebook, Twitter, and Telegram.

Fig. 5 provides an overview of how the system's three main layers interact and the flowchart for execution in Fig. 6.

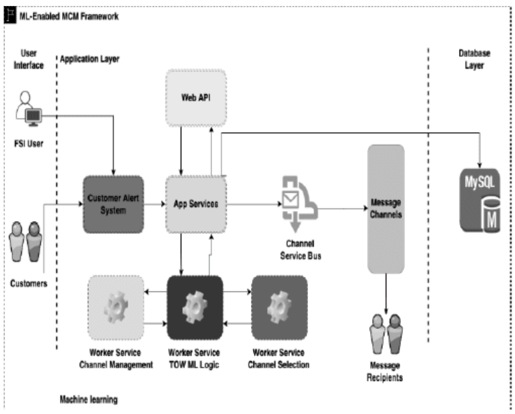


Fig. 5. ML-Enabled MCM Architecture in a Customer Alert System

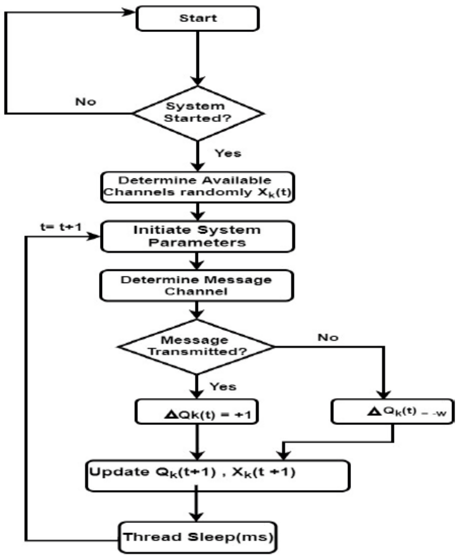


Fig. 6. Flowchart of the Proposed Framework

V. Conclusion

The design of a machine learning-enabled MCM-channel selection algorithm based on TOW-dynamics was proposed in this paper. This solution cab be used in a customer alert system used by FSIs to identify select communication channels available to their customers.

The proposed solution implements a simple-learning technique, that requires a success or failure message to understand the channel's availability status. The proposed TOW algorithm was modelled around the customer alert system utilized by financial services institutions. The focus was on the TOW algorithm channel selection algorithm and its effect on the MCM system's efficiency. Due to its implementation, FSI customers will have a consistent experience across all communication platforms. As further work in this research study, the next stage will delve more into the evaluation of the design.

Channel availability and how the system would respond under these circumstances are two other areas of concern in the design. Additional research can be conducted in the future to address other difficulties, such as memory-intensive operations in channel worker services and negotiation algorithms between the message producing and consuming nodes.

Acknowledgment

Our thanks and special gratitude to Oyenike, Birmingham City University and Oluwaseun for their contributions to this paper's review and revision process and ideas on the structure. In addition special thanks to Dr. Adebowale Owoseni, De Montfort University, Leicester City, who also provided an expert perspective on this topic related to machine learning is greatly appreciated.

References

[1] Universal Postal Union, ICTs , New Services and transformation of the Post. 2010.

[2] E. R. Adagunodo and O. Bamidele, "SMS Banking Services: A 21 st Century Innovation in Banking Technology," 2007. Accessed: May 09, 2020. [Online]. Available: http://www.bankislam.com.

[3] O. M. Ayodele and O. Babajide, "Assessment of Use of Social Media in Real Estate Transactions in Lagos Property Market," vol. 1, no. 2, pp. 63–68, 2015.

[4] C. Mbama, "Digital banking, customer experience and bank financial performance : UK customers' perceptions," vol. 36, pp. 230–255, 2018.

[5] S.-J. Kim, M. Aono, and M. Hara, "Tug-of-war model for the two-bandit problem: Nonlocally-correlated parallel exploration via resource conservation," BioSystems, vol. 101, no. 1, pp. 29–36, 2010, doi: 10.1016/j.biosystems.2010.04.002.

[6] J. Ma, T. Nagatsuma, S.-J. Kim, and M. Hasegawa, "A Machine-Learning-Based Channel Assignment Algorithm for IoT," 1st Int. Conf. Artif. Intell. Inf. Commun. ICAIIC 2019, pp. 467–472, 2019, doi: 10.1109/ICAIIC.2019.8669028.

[7] A. Omarini, "Multi-channel distribution in banking: Customers perspectives and theoretical frameworks to increase user acceptance of a multiplatform banking business," Banks Bank Syst., vol. 8, no. 1, pp. 78–96, 2013.

[8] P. Germanakos, G. Samaras, and E. Christodoulou, "Multi-channel Delivery of Services -- The Road from eGovernment to mGovernment : Further Technological Challenges and Implications," pp. 210–220, 2004.

[9] G. Zhou, C. Huango, T. Yan, T. He, J. A. Stankovic, and T. F. Abdelzaher, "MMSN: Multi-frequency media access control for wireless sensor networks," Proc. - IEEE INFOCOM, 2006, doi: 10.1109/INFOCOM.2006.250.

[10] J. Zhu, Y. Song, D. Jiang, and H. Song, "Multi-armed bandit channel access scheme with cognitive radio technology in wireless sensor networks for the Internet of Things," IEEE Access, vol. 4, pp. 4609–4617, 2016, doi: 10.1109/ACCESS.2016.2600633.

[11] S. J. Kim, M. Aono, and E. Nameda, "Efficient decision-making by volume-conserving physical object," New J. Phys., vol. 17, no. 8, 2015, doi: 10.1088/1367-2630/17/8/083023.

[12] J. C. K. Chow, "Analysis of Financial Credit Risk Using Machine Learning," no. April, 2017, doi: 10.13140/RG.2.2.30242.53449.

[13] N. Rahimi, J. Maynor, and B. Gupta, "Adversarial Machine Learning : Difficulties in Applying Machine Learning Existing Cybersecurity Systems," vol. 69, pp. 40–47, 2020.

[14] S. Shalev-Shwartz and S. Ben-David, Understanding machine learning: From theory to algorithms, vol. 9781107057. 2014.

[15] J. Vansh, "Machine Learning Algorithms," no. May, pp. 210–233, 2019, doi: 10.4018/978-1-5225-7955-7.ch009.

[16] L. Belzner and T. Gabor, "QoS-Aware multi-Armed bandits," Proc. - IEEE 1st Int. Work. Found. Appl. Self-Systems, pp. 118–119, 2016, doi: 10.1109/FAS-W.2016.36.

[17] Z. Wang, R. Zhou, and C. Shen, "Regional Multi-Armed Bandits with Partial Informativeness," IEEE Trans. Signal Process., vol. 66, no. 21, pp. 5705–5717, 2018, doi: 10.1109/TSP.2018.2870383.

[18] A. Niculescu-Mizil, "Multi-armed bandits with betting," COLT 2009 Work., no. April, pp. 133–138, 2009.

[19] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," Adv. Appl. Math., vol. 6, no. 1, pp. 4–22, 1985, doi: 10.1016/0196-8858(85)90002-8.

[20] R. Herbert, "Some Aspects of the Sequential Design of Experiments," Bull. Am. Math. Soc., vol. 58, no. 5, pp. 527–535, 1952, doi: 10.1090/S0002-9904-1952-09620-8.

[21] W. R. Thompson, "On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples," Biometrika, vol. 25, no. 3/4, pp. 285–294, Mar. 1933, doi: 10.2307/2332286.

[22] S.-J. Kim, M. Aono, and M. Hara, "Tug-of-war model for multi-armed bandit problem," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 6079 LNCS, pp. 69–80, 2010, doi: 10.1007/978-3-642-13523-1_10.

[23] N. Gutowski, T. Amghar, O. Camp, and F. Chhel, "CONTEXTUAL MULTI-ARMED BANDITS Global Versus Individual Accuracy Track on Recommender Systems," no. September, 2019.

[24] B. Gao and L. Pavel, "On the Properties of the Softmax Function with Application in Game Theory and Reinforcement Learning," no. August, 2017, [Online]. Available: http://arxiv.org/abs/1704.00805.

[25] R. S. Sutton and A. G. Barto, "An introduction to reinforcement learning," Decis. Theory Model. Appl. Artif. Intell. Concepts Solut., pp. 63–80, 2011, doi: 10.4018/978-1-60960-165-2.ch004.

[26] S.-J. Kim, M. Naruse, and M. Aono, "Harnessing the Computational Power of Fluids for Optimization of Collective Decision Making," Philosophies, vol. 1, no. 3, pp. 245–260, 2016, doi: 10.3390/philosophies1030245.

[27] X. Zhou, M. Sun, G. Ye Li, and B.-H. Juang, "Intelligent Wireless Communications Enabled by Cognitive Radio and Machine Learning," 2018.

[28] K. Oshima, T. Onishi, S.-J. Kim, J. Ma, and M. Hasegawa, "Efficient wireless network selection by using multi-armed bandit algorithm for mobile terminals," Nonlinear Theory Its Appl. IEICE, vol. 11, no. 1, pp. 68–77, 2020, doi: 10.1587/nolta.11.68.

[29] R. Wolski, J. S. Plank, J. Brevik, and T. Bryan, "Analyzing market-based resource allocation strategies for the computational Grid," Int. J. High Perform. Comput. Appl., vol. 15, no. 3, pp. 258–281, 2001, doi: 10.1177/109434200101500305.

[30] U. Berthold, F. Fu, M. Van Der Schaar, and F. Jondral, Detection of Spectral Resources in Cognitive Radios Using Reinforcement Learning. 2008.

Appendix C: Certificate of Language Editing

THEEDITINGDOCTOR(PTY)LTD

Principal Research Editor & Managing Director TJ Malefetse - PhD, MSc (Wits); BSc Hons (Essex, UK); HND (Kingston, UK)
COMPANY REGISTRATION NO.: 2017/053873/07; **ADDRESS:** 48 Acacia Glen, Trefnant Road, Ormonde, 2091, Republic of South Africa;
TEL: + 27 (0) 61 544 2641; **E-MAIL:** info@TheEditingDoctor.com **OR** Editing_Doctor@yahoo.com

EDITORIAL CERTIFICATE

This certificate serves to confirm that the manuscript listed below was edited by one or more professional scientific editors with a PhD. The University or the student's supervisor(s) can contact us for a copy of the edited document that was submitted to the author(s).

Manuscript Title:

Modelling a Multi-Channel Messaging Framework: A Machine Learning Approach

Authors:

Olusola Oluseun Salami

Date Issued:

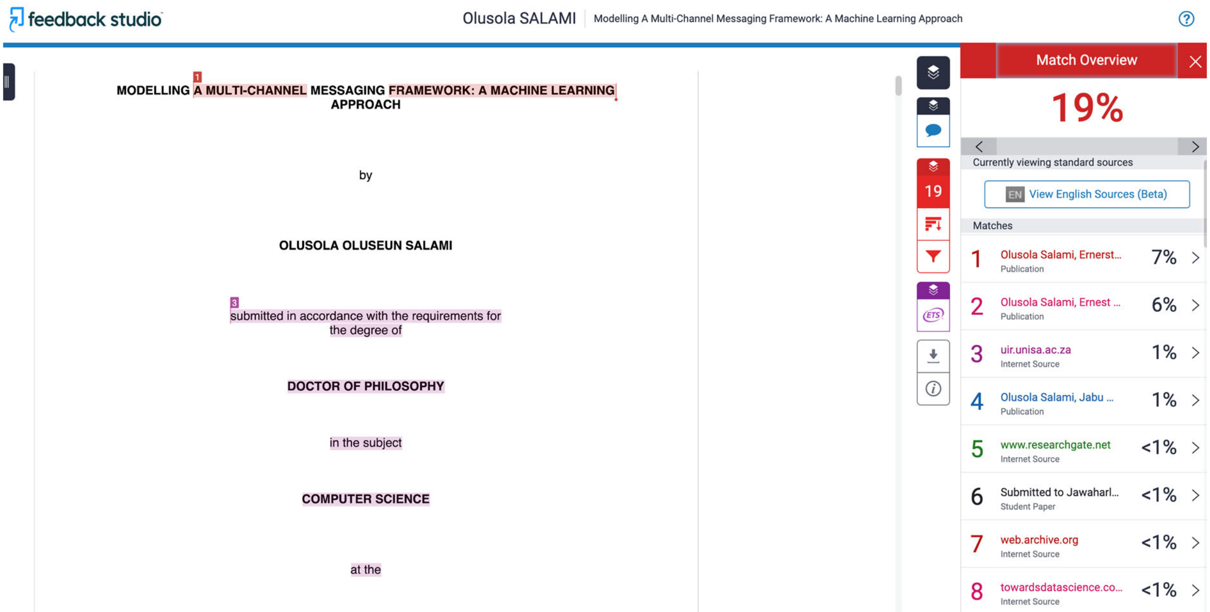
02 December 2022

Certificate Number:

11-092022 PVT

TheEditingDoctor(Pty)Ltd works with researchers from academic and research organizations in the fields of science, engineering and technology to get their work ready for examination and/or publication in internationally acclaimed English journals. Our areas of specialization include Editing (i.e., editing poor syntax, poor general structure, repetition, redundancy, contradiction, or inconsistency) and Rewriting (i.e., rewriting of phrases, sentences, paragraphs or sections to achieve better flow, more coherence clearer logic, and more appropriate vocabulary).

Appendix D: Turnitin Similarity Reports



Appendix E: Messaging Channels Format

Channel	Message Format

Twitter	{
	“message type “: {
	"message":"<message details>",
	"createdTime":"<Time Message was Created>",
	"Sent":"<Boolean Flag to know if message has been sent>",
	}
	}

Facebook	**<u>Request Message:</u>**
	curl -X POST -H "Content-Type: application or json" -d '{
	"message_type": "<Type_Of_Message>",
	"recipients": {
	"guuid": " {<Unique message ID>}"
	},
	"message": {
	"text": "<Message to be Sent>"
	}
	}'
	
	<u>Response Message:</u>
	{
	"recipient_id": "b4601c36-f508-4d12-bd99-4bb4b2cbeb33",
	"message_id": "05eff707-fcf8-46eb-a566-483f890ae241",
	“status”: “<true or false>”
	}

Channel	Message Format

WhatsApp	**<u>Request Message:</u>**
	
	curl -X POST $MSG_API_ENDPOINT \
	-H 'Authorization: Bearer <JWT_TOKEN>
	-H 'Content-Type: application or json' \
	-H 'Accept: application or json' \
	-d $'{
	"from": { "type": "whatsapp", "number": "'$WHATSAPPNUMBER'" },
	"to": { "type": "whatsapp", "number": "'$TONUMBER'" },
	"message": {
	"content": {
	"type": "text",
	"text": "<Message>"
	}
	}
	}'
	
	<u>Response Message:</u>
	{
	"recipient_id": "b4601c36-f508-4d12-bd99-4bb4b2cbeb33",
	"message_id": "05eff707-fcf8-46eb-a566-483f890ae241",
	“status”: “<true or false>”
	}

SMS	Plain Text
Email or MMS	Plain Text, Audio, Video formats, Attachments
VOIP	**<u>Session Establishment</u>** {

Channel	Message Format

	"id": " 6c5aeab0-e27e-4bd0-ab33-38973cf4a6be",
	"title": "place call to XXX-XXX-XXXX",
	"record": false,
	"steps": [
	{
	"id": " dff16e1a-61e2-43d1-8aca-5ba251c6ade5",
	"action": "call",
	"options": {
	"destination": "XXX-XXX-XXXX"
	},
	}
	],
	"createdAt": "2020-11-06T13:34:14Z",
	"updatedAt": "2020-11-06T13:34:14Z"
	}