

UNIVERSITY FOR DEVELOPMENT STUDIES

ENHANCED HIDDEN MARKOV MODELS (HMMs) FOR REAL-TIME FRAUD
DETECTION IN ELECTRONIC BANKING

ABUKARI ABDUL AZIZ DANAA



UNIVERSITY FOR DEVELOPMENT STUDIES

FACULTY OF MATHEMATICAL SCIENCES

ENHANCED HIDDEN MARKOV MODELS (HMMs) FOR REAL-TIME FRAUD
DETECTION IN ELECTRONIC BANKING

BY

ABUKARI ABDUL AZIZ DANAA (UDS/DCM/0001/18)

MSc.(Information Technology)

DISSERTATION SUBMITTED TO THE DEPARTMENT OF MATHEMATICS,
FACULTY OF MATHEMATICAL SCIENCES, UNIVERSITY FOR DEVELOPMENT
STUDIES IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
AWARD OF PhD COMPUTATIONAL MATHEMATICS

MARCH, 2022



DECLARATION

Student

I hereby declare that this thesis is the result of my own original work and that no part of it has been presented for another degree in this University or elsewhere:

Candidate's Signature:..... Date:.....

Candidate's Name: Abukari Abdul Aziz Danaa

Supervisors

I hereby declare that the preparation and presentation of this thesis was supervised in accordance with the guidelines on supervision of thesis laid down by the University for Development Studies.

Principal Supervisor's Signature:..... Date:.....

Name: Prof. Mohammed Ibrahim Daabo

Co-Supervisor's Signature:..... Date:.....

Name: Dr. Alhassan Abdul-Barik



ABSTRACT

Hidden Markov Models (HMMs) has become increasingly popular in the last few decades due to its very rich mathematical structure and therefore forming the theoretical basis for use in a wide range of real-life applications such as in speech and image recognition, motion analysis in videos, bio-informatics among others. However, an effective optimization of the parameters of these Models for enhanced performance has remained computationally challenging and there is no generally agreed method that can guarantee best performance within reasonable computing time. Another significant challenge with the application of Machine learning algorithms to anomaly/fraud detection is the high number of false positives and negatives especially in the presence of highly class-imbalanced data sets. Designing an accurate efficient real-time Fraud Detection System (FDS) that is low on false positives and negatives but detects fraudulent activities effectively is essential. In this research, a hybrid algorithm comprising the Particle Swarm Optimization (PSO), Baum-Welch (BW), and Genetic algorithms (GA) is proposed and implemented for optimizing the parameters of HMMs. A framework based on HMMs, modified Density Based Spatial Clustering of Applications with Noise (DBSCAN) and Synthetic Minority Oversampling Technique (SMOTE) is also implemented to effectively detect real-time electronic Banking fraud. An enhanced multi-layer HMM is proposed and implemented to further reduce the false positives and negative rates. Simulation results demonstrates that, the proposed hybrid optimization algorithm overcomes the weaknesses of the slow convergence of PSO whilst enabling the BW to achieve a global optimal solution. It also improves the performance of the GA by reducing its search space for optimal performance. Using highly imbalanced datasets, the proposed system performed relatively better when compared to some common approaches in literature in terms of precision, recall, F1-Scores and convergence rates. An improved multi-layer HMM proposed by the study also performed better with enhanced training and detection times as compared to other techniques widely used in literature.



ACKNOWLEDGMENT

I will forever be grateful to my lovely wife, Adam Rafatu Kande, without whose support, this journey could not have been that successful. I also say a very big thank you to my mother, Mma Adamu, for instilling in me, discipline, hardwork and sacrifice right from the beginning which played a critical role in the successful completion of this programme. I would also like to express sincere gratitude to my supervisors, Prof. Mohammed Ibrahim Daabo and Dr. Alhassan Abdul-Barik for their guidance, understanding, and patience in shaping and adding value to this research work.

I would like to thank all staff of the Mathematics and Computer Science Departments, C. K. Tadam University of Technology and Applied Science (CKT-UTAS), formerly the Navrongo campus of University for Development Studies for their emmense support during the programme. Last but not least, I thank all my colleague students in the Department for their suggestions and comments that contributed to the enhancement of this research.



DEDICATION

I dedicate this thesis to the memory of my Late Father, Issifu Abukari, and to my lovely kids; Mufida, Ridwan, Hamdan and Shakir.



TABLE OF CONTENTS



DECLARATION	i
ABSTRACT	ii
ACKNOWLEDGMENT	iii
DEDICATION	iv
LIST OF TABLES	x
LIST OF FIGURES	xii
CHAPTER 1 INTRODUCTION	1
1.1 Background of the Study	1
1.1.1 Electronic Banking	2
1.1.2 Electronic Banking Platforms	3
1.1.3 Electronic Banking Architecture	5
1.1.4 Types of Intrusions in Electronic Financial Transactions	7
1.1.5 Intrusion Detection Approaches	9
1.1.6 Fraud Detection using Machine learning algorithms	11
1.1.7 Fraud Detection using HMMs	12
1.1.8 Classification related Problems	16
1.1.9 Machine Learning Model Evaluation	19
1.1.10 Hidden Markov Models	19
1.1.11 HMM Notation	21
1.2 Problem Statement	24

1.3	Objectives of the Study	26
1.4	Research Questions	27
1.5	Significance of the Study	27
1.6	Scope of the Study	28
1.7	Organization of the Thesis	29

CHAPTER 2 LITERATURE REVIEW 30

2.1	Introduction	30
2.2	Optimizing the parameters of HMMs	30
2.3	Detecting real-time electronic banking fraud on highly imbalanced datasets using HMMs	34
2.4	Reducing False Positives in Real-time Fraud Detection using Multi-Layer HMMs	38
2.5	Chapter Summary	42

CHAPTER 3 METHODOLOGY 44

3.1	Introduction	44
3.2	An improved Hybrid Algorithm for optimizing the parameters of HMMs	44
3.2.1	The Baum-Welch Algorithm	45
3.2.2	Particle Swarm Optimization Algorithm	48
3.2.3	Genetic Algorithm	51
3.2.4	Existing Hybrid Algorithms that combines GA, PSO and BW Algorithms for optimizing the Parameters of HMMs	52
3.2.5	The Proposed Hybrid Optimization Algorithm	56
3.3	Detecting Real-Time Electronic Banking Fraud on highly imbalanced datasets using HMMs	59
3.3.1	Dataset	59
3.3.2	Data Pre-Processing	60
3.3.3	Identifying Transaction Profile of Customers using a Modified DB-SCAN Clustering Algorithm	61
3.3.4	Scalling the proposed HMMs	64



3.3.5	Training the proposed HMMs	66
3.3.6	Fraud Detection	68
3.3.7	Evaluation Metrics	69
3.4	Reducing False Positives in Real-Time Fraud Detection Using Improved Multi-Layer HMMs	70
3.4.1	Data Pre-processing	70
3.4.2	Structure of the proposed Improved Multi-layer HMM	70
3.4.3	Training the Proposed Improved Multilayer-HMM	73
3.5	Chapter Summary	75

CHAPTER 4 RESULTS AND DISCUSSIONS 76

4.1	Introduction	76
4.2	Implementing the improved Hybrid Algorithm for optimizing the parameters of HMMs	76
4.3	Detecting Real-Time Electronic Banking Fraud on highly imbalanced dataset using HMMs	79
4.3.1	Precision Comparison	80
4.3.2	Recall Rates Comparison	80
4.3.3	F1-Measure Comparison	80
4.3.4	Learning and AUC-ROC Curves	82
4.4	Reducing False Positives in Real-Time Fraud Detection Using an Improved Multi-Layer HMM	84
4.4.1	Confusion Matrix	85
4.4.2	Precision, Recall and F1-score and ROC Curve Plot	86
4.4.3	Computational efficiency of the Models	87
4.5	Chapter Summary	89

CHAPTER 5 SUMMARY, CONCLUSIONS AND RECOMMENDATIONS 90

5.1	Introduction	90
5.2	Summary of Major Research Findings	90



5.2.1	An improved Hybrid Algorithm for optimizing the parameters of HMMs	90
5.2.2	Detecting Real-Time Electronic Banking Fraud on highly imbalanced datasets using HMMs	91
5.2.3	Reducing False Positives in Real-Time Fraud Detection Using an Improved Multi-Layer HMMs	92
5.3	Conclusion	93
5.3.1	Implementing the improved Hybrid Algorithm for optimizing the parameters of HMMs	93
5.3.2	Detecting Real-Time Electronic Banking Fraud on highly imbalanced datasets using the proposed HMMs	94
5.3.3	Reducing False Positives in Real-Time Fraud Detection Using the Improved Multi-Layer HMMS proposed by the study	94
5.4	Recommendations for future research	95

APPENDICES 110

APPENDIX A SAMPLE PYTHON SOURCE CODES 110

A.1	Clustering Algorithms	110
A.2	Fraud Detection	112
A.3	Defining the structure of HMM	113
A.4	The Hybrid Optimization Algorithm	114
A.5	Generating a Learning Curve for our models	121

APPENDIX B SAMPLE SIMULATION RESULTS 123

B.1	Fraudulent and genuine transactions Dispersion over time in our sample Dataset	123
B.2	Fraudulent and genuine transactions dispersed over amount in our sample Dataset	124
B.3	Values of $P(O \lambda)$ for the various transaction profiles for various hidden number of states for BW, PSO, GA and BWPSO	125



B.4 Values of $P(O|\lambda)$ for the various transaction profiles and various hidden number of states for BWGA, GAPSO, and the Proposed Algorithm 126

B.5 Time taken to optimize the HMMs by various algorithm for all Transaction profiles 127

B.6 Precision, Recall and F1-Scores of the Various Algorithms/Approaches for various Number of Hidden States 128

Publications and Conferences attended **129**



LIST OF TABLES

1.1	A comparison of some Fraud/Intrusion Detection Techniques	17
3.1	Transaction mix representing the various transaction profiles of customers .	58
3.2	Details of the Paysim Dataset adopted for the Research	59
4.1	Average values of $P(O/\lambda)$ for a low-amount, Low-frequency transaction profile for all values of N	77
4.2	Confusion Matrix for the various number of hidden states	85
4.3	Precision, Recall and F1-scores obtained by the various Models	86
4.4	Training and Detection times (secs) obtained by the various models for all the possible values of Hidden states	88





LIST OF FIGURES

1.1	Electronic Banking Architecture	7
1.2	A general framework of a misuse/signature detection system	10
1.3	A logical flow of anomaly detection approach	11
1.4	A general framework of a Fraud Detection System using a trained profile. .	13
1.5	Categorization of various Fraud detection methods	16
1.6	Basic structure of a HMM	20
3.1	A diagramatic representation of the movement of particles using PSO algorithm	50
3.2	Structure of GA	51
3.3	Flow chart of a GA-PSO hybrid optimization algorithm	53
3.4	A flow of the Proposed Hybrid Optimization Algorithm	56
3.5	Flowchart of the detection phase of the proposed system	69
3.6	A flow of the proposed Improved Multi-layer HMM	71
4.1	Values of $P(O/\lambda)$ for a Low-Amount, Low-Frequency transaction profile for all values of N	78
4.2	Values of $P(O \lambda)$ for the various transaction profiles for various hidden number of states	78
4.3	Time taken to optimize the HMMs by the various algorithm for all Transaction profiles	79
4.4	Comparison of the precision of the four (4) different approaches for various number of hidden states	81
4.5	Recall rates of the four (4) approaches for different number of Hidden states	81

4.6	F1 Sores of the four (4) approaches for different number of Hidden states . . .	82
4.7	Learning curve for approaches where the Modified DBSCAN is adopted . . .	83
4.8	Learning curve for approaches where K-Means is adopted	83
4.9	AUC-ROC Curve obtained for the various approaches	84
4.10	ROC Curve Plot of the models for various number of Hidden States	87
4.11	Average training and detection times (sec) obtained by the various models	88



LIST OF ABBREVIATIONS

ICT	Information and Communication Technology
ATMs	Automated Teller Machine
K-NN	k-nearest neighbors
HMMs	Hidden Markov Models
ROCs	Receiver operating characteristic
RUC	Area under Curve
BW	Baum-Welch
PSO	Particle Swarm Optimization
GA	Genetic Algorithm
TS	Tabu Search
DBSCAN	Density-based spatial clustering of applications with noise
SMOTE	Synthetic Minority Over-sampling TEchnique
FP	False Positive
TP	True Positive
FN	False Negative
TN	True Negative





AI	Artificial Intelligence
ICPA	Image Click Point Authentication
TPR	True Positive Rate
FPR	False Positive Rate
FDS	Fraud Detection System
ADLs	Activities of Daily Livings
ELM	Extreme Learning Machine
MLHMM	Multiple-layer Hidden Markov Model
SLHMM	Single-layer Hidden Markov Model
E-Banking	Electronic Banking
KNN	K-Nearest Neighbor
SMS	Short Message Service

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

Electronic banking has redefined the way banking is conducted across the globe over the last few decades and the use of electronic payments platforms has continued to experience significant growth. It allows customers a 24-hour access to their accounts with the ability to transfer funds, perform on-line payments and apply for loans and other financial products virtually (Asare & Sakoe, 2015). Unfortunately, fraud cases relating to cyber-crime perpetrated through electronic banking resulted in an actual loss of almost GH¢1 billion and therefore presents a unique challenge to individuals and financial institutions that offer those services (Bank of Ghana, 2020).

According to Barrios (2013), any conscious or deliberate action by an individual or group of persons purposely to alter the truth or facts for personal gain is known as fraud. NGGL (2013) however defined fraud as an act of deception or forgery planned by unauthorized individuals or group of persons with the main ambition to change the facts for monetary gain.

To detect and prevent fraudulent transactions, a variety of fraud prevention tools such as transaction authorization in real-time, codes for real-time transaction tracking, alerts for transaction confirmation among others are employed by various financial institutions (Kar et al., 2016). However, according to Vaidya & Mohod (2013), due to the ever-changing nature of hackers and intruders, it is quite possible to bypass these security measures with time to commit fraud which has serious financial implications for individuals and financial institutions.



According to Pérez et al. (2007), it is very necessary to maintain the reliability and efficiency of E-banking platforms through the implementation of robust and adaptive technologies that has great potentials to detect and prevent fraud effectively especially in real-time. It has also been established that, the more time and efforts financial institutions put in place to detect and prevent fraudulent activities, so does the approaches employed by these fraudsters gets sophisticated (Kovach & Ruggiero, 2011).

1.1.1 Electronic Banking

According to Aupetit et al. (2007), the financial sector is one of those most affected by the ever-evolving nature of Information and Communications Technology (ICT). It has transformed beyond just the withdrawal and deposits of cash whether personally or through the use of checks to enabling customers perform even the most sophisticated transactions such as cash transfer and the payment for goods and services at their convenience.

According to Sulaiman & AbdelKarim (2019), ICT has redefined the approach adopted by most organizations in their way of conducting business resulting in the tremendous expansion in its customer base coupled with reduced cost of performing transactions with better quality. Customers are also able to personalize their banking transactions through self-service options available, thereby improving communication and customer experience (Ghamri, 2017). It is significant to note that, the competitive advantages gained by many financial institutions through the use of ICTs which can be seen in the reduction of operating costs and increased profits. When the financial industry is efficient and effective, its significant impact of the socio-economic development of the nation at large cannot be underrated (Zhang, 2009).

Aggelis (2006) opined that, due to the ever-changing globalization, the financial industry needs to constantly be given the needed attention to also grow accordingly, which has a potential to transform it from the well-known traditional banking system to a more robust distribution channel. Fortunately, the adoption of E-banking in Ghana is fast increasing and care must be taking to incorporate appropriate security measures to protect it.



Almost all financial institutions in Ghana and across the world have implemented at least some form of E-banking services in order not to lose existing and potential customers to competitors (Xiang et al., 2019). The delivery of traditional Banking services and products to customers electronically are automated through the adoption of E-Banking platforms where access to transaction accounts and interacting with the customers are simplified and enhanced (Sangita & Madhuri, 2015).

Customers can deposit and withdraw funds through Automated Teller Machines (ATMs), conduct banking transactions through the internet and Mobile phones, use Debit and credit cards to purchase and make payment for goods and services among others at their own convenience (Dandash et al., 2008). According to Adedoyin (2018), efficiency, availability and reliability are some of the benefits of adopting E-banking from the customers perspective since the services are provided on a 24/7 basis.

Although the deployment of E-Banking services is complicated and usually results in an increased operation cost, Devi (2018) insisted that, financial institutions has no option than to make conscious efforts in its adoption. He added that, this will result in enhanced operational efficiency and effectiveness so as to gain competitive advantage whilst providing excellent service as well establish everlasting stronger bonds with existing and potential customers.

1.1.2 Electronic Banking Platforms

Depending on the availability of a particular transaction type/service, customers may employ among others the following Electronic banking platforms (Gupta et al., 2017).

- **Automated Teller Machines (ATM)**

According to Diadiushkin et al. (2019), ATMs are terminals that enables registered customers have access to their bank accounts balances, make cash withdrawals and deposit and as well transfer funds among accounts. Certain types of ATM Cards allows customers to perform transactions with any ATM machine even if they have no accounts with that particular financial institution at a relatively higher service fee.



- **SMS-based banking**

Ali et al. (2019) explained that, this type of service allows customers to transact business through SMS-messaging on their mobile devices. Financial institutions are allowed to send customized messages to some targeted group of customers to update them on the current status of their accounts especially if there is an un-usual transaction.

- **Direct Deposit**

This type of service allows a customer to give formal consent for some types of deposits for recurring personal benefits such as those relating to social security among others to be performed on a particular account. Likewise, authorization can be given for some debits such as electricity and water bills, mortgages, insurance premiums among others that occur within definite time intervals on a specified account. Although customers possess the ability to discontinue any of these deductions at any point in time, care must be taken before authorizing recurring withdrawals especially with unfamiliar companies. Since the funds in a customer bank's account could be improperly withdrawn, financial Institutions needs to periodically monitor customers transactions to ensure the right recurring payments are made (Bignell, 2006) .

- **Tele-Banking**

According to Kumari et al. (2014), registered customers under this type of E-banking platform are usually required to dial a particular short code to verify and activate the service. Some means of verification include demanding answers to some secrete questions and pin codes meant to provide enough security to the users and their transactions. Payment for goods and services, balance inquiry, cheque book and card request among others are services available on this platform.

- **Debit/Credit card transactions**

This type of service allows a customer to perform transactions with either a debit or credit card where the transactions can take place online, physically, or through a phone call. Customers always needs to have enough balance in their account before



initiating a transaction since most of these transactions are instant. Keeping an eye on an account as a customer regarding the amounts and types on transactions made and when they were authorized is one of the ways to ensure that fraudulent transactions are detected and prevented. In some cases the financial institution will send you an authorization code through your mobile number or email to enable you complete such a transaction (Alghamdi, 2016).

- **E-Zwich**

E-zwich is the brand name for the National Switch and Smart card payment system which aims at improving accessibility to banking and retail services. The platform also offers Deposit taking financial institutions a secure platform that enables them to interoperate thereby enabling cardholders to perform banking and retail transactions at the outlets of other financial institutions with E-Zwich enabled systems. E-zwich cardholders are exposed to a large group of banks and their branches where transactions can be performed and it is currently the only card in Ghana that provides the convenience of nationwide access as well greater control over transactions for cardholders, retail merchants and other corporate users (Asare & Sakoe, 2015).

1.1.3 Electronic Banking Architecture

Generally, a banking network is equipped with communications that are needed between internal devices(nodes) and those that are employed when interacting with external nodes. For the electronic transactions that are confined to the bank, the internal network is protected by a core firewall that is usually connected with a dedicated router.

The server zone deploys different application servers, database servers, storage devices, network analysis tools among others. The demilitarized zone deploys a number of servers that are being invoked by external entities through the Internet. For security reasons, sensitive information and bank's internal data are never deployed inside the demilitarized zone. Transaction platforms such as the Automated Teller Machines(ATMs) are then established inside the branch offices and also at external premises.



A branch office and the server zone is connected through a private, secured point-to-point connection, which is shown as the data link provider network between the branch router and the core router. The transaction platform's channel established inside a branch is connected with the server zone through this secured point-to-point connection. As illustrated in Figure 1.1, access to a financial institution's E-Banking services by a registered customer is hosted by a web server provided by the web service provider's network is made possible using any internet-enabled device. Usually, with the use of various web browsers such as google chrome, Internet explorer, Moxila Firefox among others, customers are able to access the Web sites of financial institutions to transact business (Maruatona et al., 2012).

According to Rambola et al. (2018), with the use of a secured point-to-point network connection, the financial institution's network is configured appropriately to connect a web server providers network. He also added that, the privacy and security details of sensitive information transmitted through these public medium needs to be highly maintained.



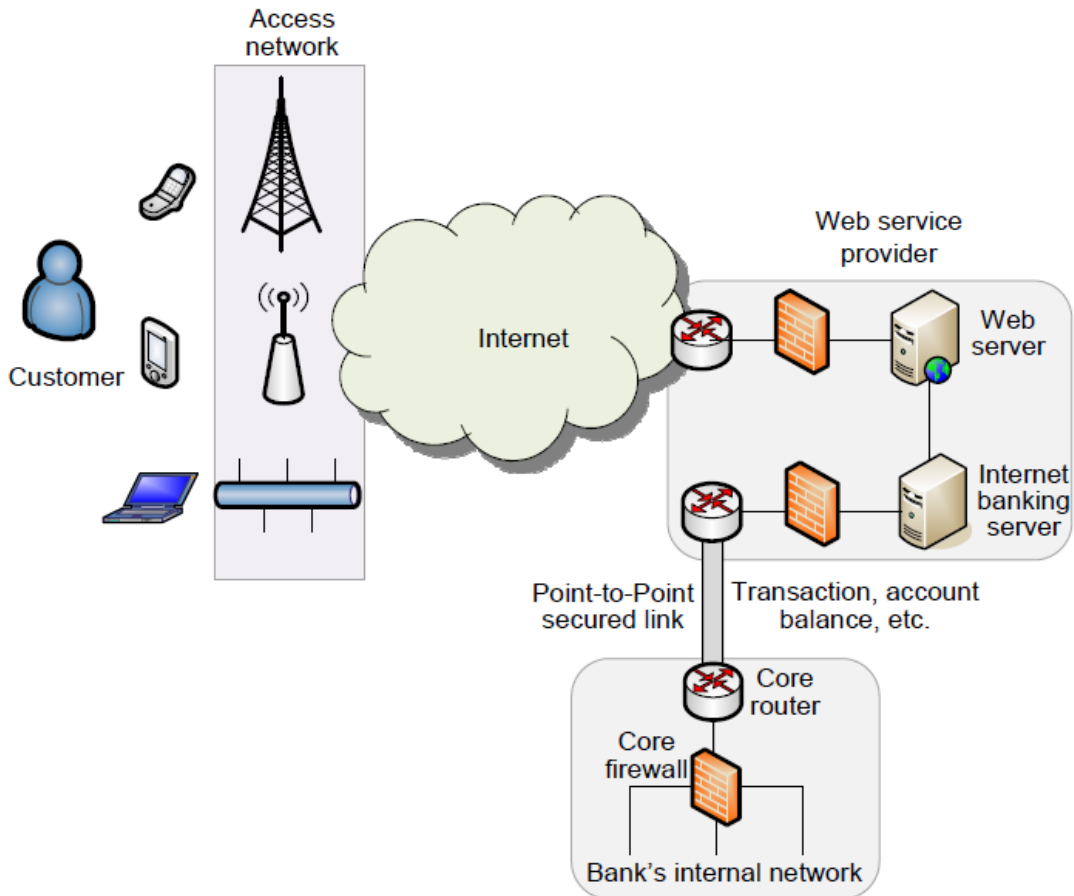


Figure 1.1: Electronic Banking Architecture

Note: Adopted from Islam & Mahfuz (2014), <https://www.researchgate.net/publication/28933466>

Adoption-of-e-banking-in-Bangladesh-Evolution-status-and-prospects/figures?lo=1



1.1.4 Types of Intrusions in Electronic Financial Transactions

The under-listed are some common types of attacks/intrusions with regards Electronic banking and E-commerce transactions.

a. Phishing

Phishing usually employs fake email messages pretending to represent one's financial institution and demanding sensitive information from users and directing them to fake and harmful sites. In some cases, although genuine web site addresses are displayed; the data provided by users into a pop-up window will be sent unknowingly to the intruders/attackers (Khraisat et al., 2019).

b. **Pharming**

According to Ali et al. (2019), redirecting traffic on a particular website to another one hosted by fraudsters by installing malicious codes on a victim's personal computer in order to have illegal access to valuable information is what Pharming seeks to achieve. Without users consent and knowledge, they are re-directed to fraudulent Web sites where they enter vital and sensitive information. It is also vital to note that pharming usually targets larger numbers of innocent computer users since victims need not give any authorization before the act. For instance, some codes may be sent in an email that modifies a local host to convert universal resource locator into the number strings that the computer uses to access Web sites files on user's personal computer. Even if genuine internet addresses are provided or affected bookmark entries are clicked using a compromised device, users are at risk of being redirected to malicious Web sites.

c. **Man-in-the-middle attack (MitM)**

According to Ahmadian Ramaki et al. (2018), Man-in-the-middle attack refers to an attack where a virus on a victim's computer modifies any legal communication between a financial institution and its clients. These illegally obtained sensitive information is then used to performing fraudulent transactions on behalf of the user.

d. **Man-in-the –browser attack (MitB)**

Man-in-the-browser attack usually infects a user's browser with a malware and attackers then perform actions on behalf of users when they log into their accounts.

This type of attack takes over a user's session at his/her blind side where malicious actions can be performed. (Abomhara & Køien, 2015).

e. **Spyware and Viruses**

According to Mehra (2015), attackers may also employ spyware to steal online banking credentials by illegally obtaining it while it is transmitted between user's computer and other websites. It is usually executed when users are deceived into installing malicious applications on their machines. Viruses on the other hand allows for an unauthorized access to a victim files by searching for information considered to be of value.



- f. **Card skimming** This type of attack usually occurs when there is an unauthorized copying and recording of sensitive data such as PIN code information from credit and debit cards. In some cases, counterfeit cards may even be manufactured to bear these illegally acquired details with the main intent to perform illegal transactions on behalf of the cardholder (Lopez-Rojas, 2014).

1.1.5 Intrusion Detection Approaches

Intrusion detection can generally be categorized into two (2) main approaches as detailed below.

a. **Misuse Detection**

In this approach, history of previously known intrusions (signatures of attacks) extracted from data sources such as user commands, audit events, system calls, network packets, keystrokes, user transaction history among others are utilized. Misuse Detection is good at detecting known intrusions and a potential of providing ways to detect different versions of the same kind of intrusions if the extracted intrusion profile is well generalized to match with the new versions (Panigrahi et al., 2009).

Figure 1.2 depicts the general structure of a typical misuse detection system. Possible matches between current user transactions and available fraudulent transactions are identified using some machine learning algorithms. In cases where exact matches are detected, the transaction is considered fraudulent and an appropriate alarm is raised. Further investigation may be initiated automatically to confirm whether the transaction is genuine or otherwise before updating the intrusion signature database with the newly detected case (Sultana et al., 2012).



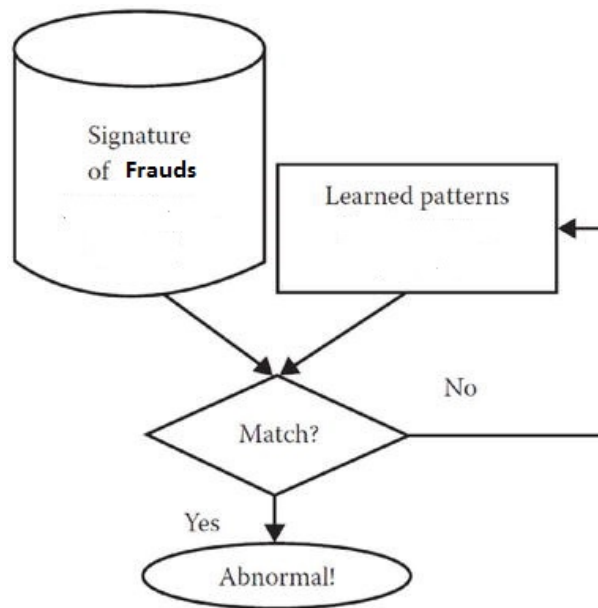


Figure 1.2: A general framework of a misuse/signature detection system

Note: Adopted from Sultana et al. (2012), <https://www.researchgate.net/publication/261317660-An-improved-Hidden-Markov-Model-for-anomaly-detection-using-frequent-common-patterns>

b. **Anomaly detection**

According to Grewal et al. (2020), in anomaly detection, normal customer transaction profiles are maintained through training a model. An incoming transaction is suspected fraudulent if it is not accepted by the trained model with a significant high probability. Although it is possible to extract all forms of a user's normal behavior in a dataset for a current problem, they usually change over time and a good model needs to take that into consideration.

A diagrammatic representation of an anomaly detection system which generally consists of five (5) steps is depicted in Figure 1.3. In the last two stages, the identification of mismatches which are deemed fraudulent transactions using dissimilarity detection techniques is executed and then an appropriate security alert is issued.



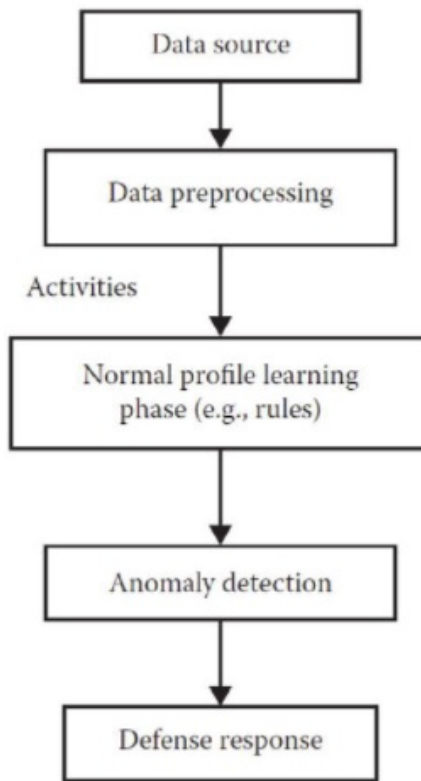


Figure 1.3: A logical flow of anomaly detection approach

Note: Adopted from Grewal et al. (2020), <https://pubmed.ncbi.nlm.nih.gov/32020103/>

1.1.6 Fraud Detection using Machine learning algorithms

The field of machine learning involves the design and implementation of algorithms that learns from data and past experiences. It encompasses many disciplines such as data mining, statistics and information theory etc. typically sharing some algorithms or approaches to solve problems relating to pattern recognition and prediction, diagnosis, planning, controlling among others (Wang et al., 2020).

Ahmadian Ramaki et al. (2018) outlined three (3) basic steps necessary to perform in machine learning-based intrusion/anomaly detection, namely, training, testing and analysis. The training step involves profiling the normal behavior of users whilst the testing step compares current activities/transactions with the learned behavior. In the analysis step however, the test results is evaluated to report significant deviations if any exists.

Three (3) different learning approaches for intrusion/anomaly detection are also outlined by Prasad et al. (2009) depending on the availability of labeled data. Supervised



learning algorithms such as k-nearest neighbor, decision trees, Naive Bayes etc. are trained with normal and abnormal data with labels in order to detect anomalies.

Unsupervised learning on the other hand is used to train models with the normal and abnormal data without labels aimed at determining possible anomalies without prior knowledge of the data with the assumption that, anomalies will appear as outliers. Unsupervised learning algorithms include among others Singular Value Decomposition, K-NN (k nearest neighbors), HMMs, K-means clustering (Khare et al., 2018). According to Xiang et al. (2019), in Semi-supervised learning however, labeled data is provided to the model and serves as a starting point for its training and testing. The model is then capable of accepting and classifying a broad range of unlabeled data it receives.

1.1.7 Fraud Detection using HMMs

Figure 1.4 depicts the general framework of a Fraud Detection System using a trained profile. In the detection process, Rieke et al. (2013) outlined the following three (3) main steps.

- The parameters for creating the model is defined where the inputs are data gathered from various users' transaction which may contain both fraudulent and genuine cases. Due to the large number of features, it is always useful to employ appropriate feature selection techniques to determine the right number and types of features to incorporate in the model for optimal performance.
- Normal behaved profiles which are used to train the Model from which the fraud detection system is created and has a potential of classifying an incoming transaction as normal or otherwise is constructed.
- In the final step, an incoming transaction is evaluated using the trained model from the previous step. Any significant deviation between the observed sequence and a normally behaved profile is considered abnormal by the model and the transaction is declined.

According to Amalina et al. (2020), it is very crucial to select features that will increase the efficiency of developed models since some parameters in the profile data can result



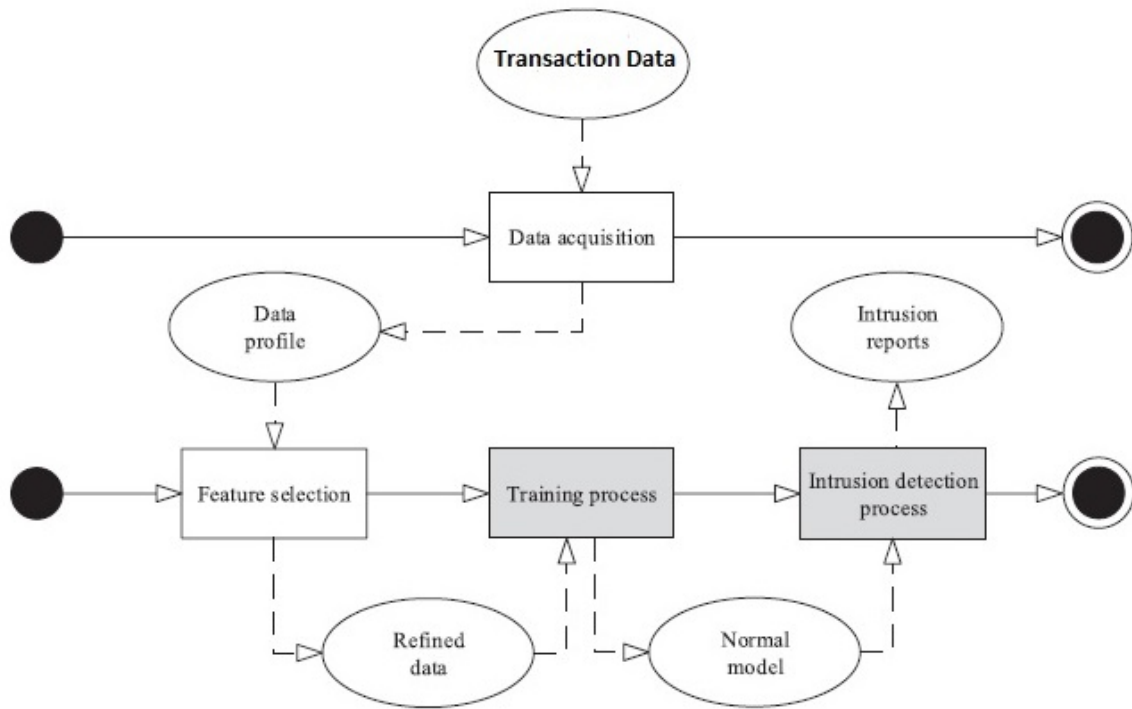


Figure 1.4: A general framework of a Fraud Detection System using a trained profile. in increased training time with decreased detection rates especially in real-time intrusion detection.

Reducing the number of attributes usually at the selection and pre-processing stage using the right data is appropriate in building an effective FDS using Data mining techniques. Without significantly altering/removing the initial features, the purpose of feature selection is to select a subset of the current features by modifying the initial features using some clustering principal component analysis techniques (Calvo-Zaragoza et al., 2019).

Outlined below are some feature selection approaches in the design of an effective Fraud Detection System.

- **The filtering approach:** The number and types of attributes to include are done without taking the specific classifier into consideration and priority is given to features with greater separation. (Moerland, 1996).
- **The wrapper approach:** Taking the precision defined by a particular classifier into consideration, the criteria for selecting attributes such as the sequential forward selection and sequential backward selection techniques among others are employed. This implies that, different groups of attributes can be chosen based on the particular classification approach adopted (Alimolaei, 2016).



- **The Embedded Approach:** According to Thiruvadi & Patel (2011), the embedded approach refers to the case where the selection of features happens alongside the creation of the classifier such as in the case of a using decision tree.

A categorization of various Fraud/Intrusion Detection techniques including HMMs is presented in Figure 1.5 and their associated benefits and challenges enumerated in Table 1.1.

1. **Statistical-Based methods (SBM):** This approach makes use of observations, which could be customers' transactions in fraud detection systems used to construct profiles to reflect normal users' transaction behavior. These profiles can be created using parameters such as transaction amounts, transaction frequency, transaction time gaps among others (Correa Bahnsen et al., 2016).
2. **Knowledge-based methods (KBM):** Also known as expert based-methods, Kang (2019) outlined the following general steps to adopt in fraud detection:
 - Identifying the various classes of the training data and the corresponding attributes to use.
 - A set of classification rules are formulated by taking the identified classes and attributes into consideration.
 - According to the specified rules, incoming transactions are classified as fraudulent or otherwise.
3. **Data mining-Based Methods (DMM):** According to Guan et al. (2017), using this method allows users to create a pattern of observed normal transactions and then uses a model to analyze and classify newly observed case. A proper labeling of the transaction data to create the behavioral model is essential for creating good Data mining-Based models.

The focus of Data mining which falls under the general umbrella of Machine learning is the appropriate recognition of useful patterns in data sets. It comprises a combination of techniques such as machine learning algorithms, statistics, and perhaps database systems (Phua et al., 2010). However, Helode et al. (2017) also added that,



extracting relevant information from significant large datasets by adopting various techniques to refine them for further usage is what Data Mining seeks to achieve and generally include the following classes of tasks;

- **Outlier Detection:** This refers to an identity of unusual pattern in a dataset that are of interest and requires an in-depth investigation to likely reveal more related facts for consumption (Atwi et al., 2011).
- **Learning Association Rules:** This type of Data mining identifies relationships among several attributes of interest. For instance, an online business interested in establishing relationship between customers and their shopping preferences. Such useful information can form the basis to develop customized and targeted products to customers to meet their needs (Pérez et al., 2007).
- **Clustering:** According to Pérez et al. (2007), Clustering techniques are employed to identify the connections that exists among various groups with varying shapes and to associate various data-points. It is also essential to note that, some clustering techniques have the ability to filter outliers to produce more accurate results. To be able to dynamically classify a new data point, it is usually compared to the centroid of each cluster to determine the minimum distance in order to classify that particular transaction.
- **Classification:** If the target is to generalize known hidden structures within a dataset in order to identify new datapoints, then employing classification methods will be most appropriate. For example, the act of classifying an incoming electronic mail as genuine or spam can be considered an act of classification (Xiaoguo et al., 2018).
- **Regression:** According to Lookingbill & Urban (2005), ideal relationship that exist among dependent and independent attributes in a datasets can be achieved by employing regression analysis. For example, if there is a suspicion that the total debit amount on a bank account within a week is related to the income of the account-holder, then Regression analysis would be a good method to establish such.



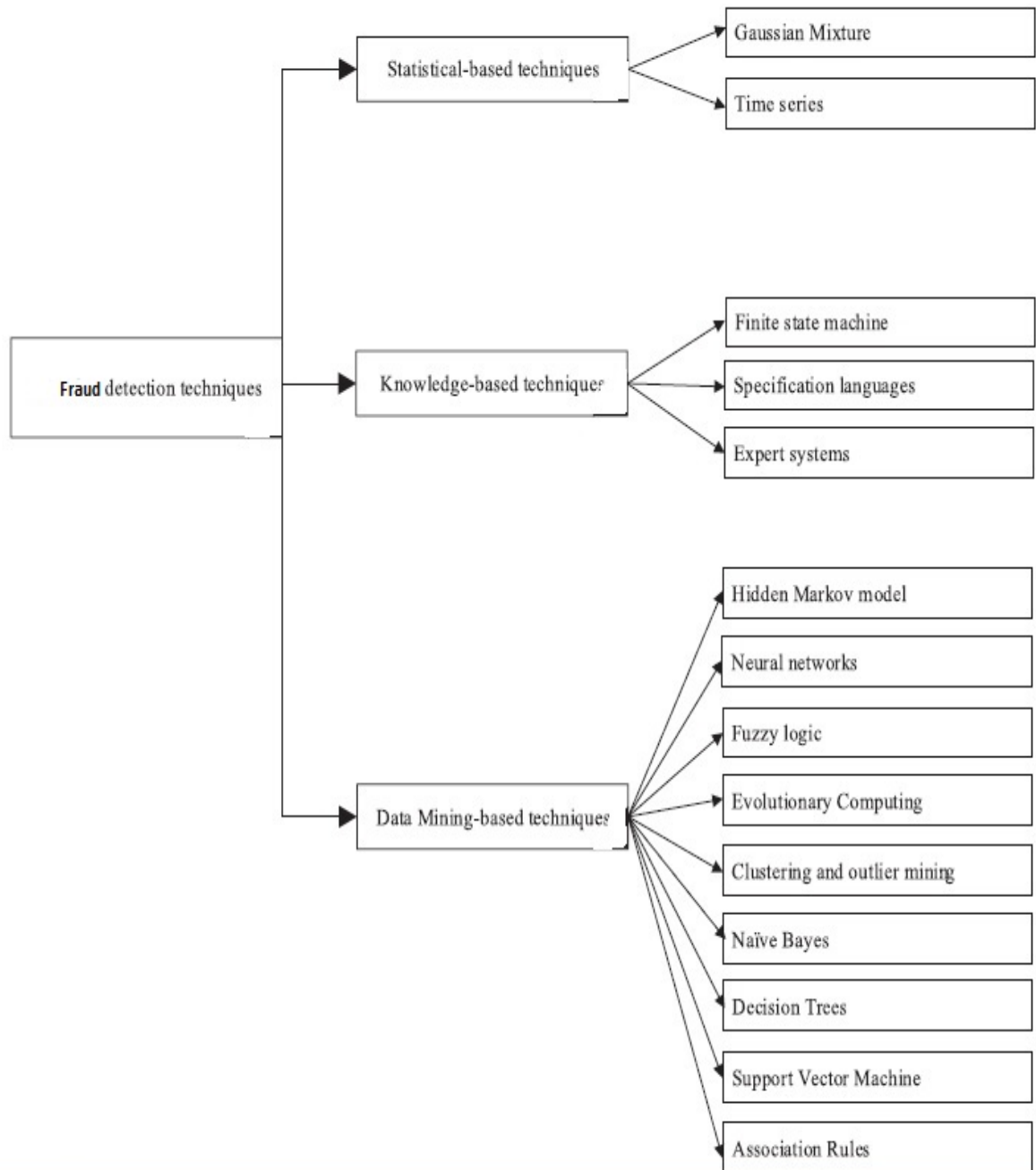


Figure 1.5: Categorization of various Fraud detection methods

Note: Adopted from Ahmadian Ramaki et al. (2018), <https://dl.acm.org/doi/abs/10.1002/sam.11>

1.1.8 Classification related Problems

According to Ko et al. (2012), classification is an instance of a more general pattern recognition problem where an out value is assigned to every corresponding input value. However, when real-value output are rather assigned to various input values, the process is seen rather as regression and sequence identifying focuses on relating classes to each



Table 1.1: A comparison of some Fraud/Intrusion Detection Techniques

	SBM	KBM	DMM
Advantages			
Independence from previous knowledge of normal behavior	yes	no	no
Ability to identify fraudulent activities with high precision	yes	no	yes
Requires large volume of data for training	no	no	no
Possess a High degree of scalability and flexibility	no	yes	yes
Ability to produce significant low level of FPs	no	yes	no
Ability to incorporate adaptive learning for increased performance	no	yes	yes
Ability to present intrusion patterns graphically	no	no	yes
Ability to detect fraudulent transactions which are unknown	no	no	yes
Disadvantages			
Capability of a fraudster to re-train a model	yes	no	yes
High level of complexity in terms of Configuration	yes	no	no
High complexity in terms of model training time	no	yes	yes
Depends solely on normal trained profiles to detect fraud	no	yes	no

member on a chain of values (Achituve et al., 2019).

Another category of classification algorithms are termed probability classifiers that employ statistical inference methods in the determination of appropriate class for a given dataset. The difference between probability and non-probability classifiers is that, the latter returns only single "best" classes whilst the other return the probability that an instance is a member of each possible class. In general, the class with the highest probability is chosen as the best.



Outlined below are the advantages of probability over non-probability classifiers. (Panigrahi et al., 2009):

1. A category of classifiers known as weighted classifier have the ability to produce a confidence value based to the parameters you desire.
2. Their ability to hold back if the confidence level in choosing a particular output is statistically insignificant.
3. Due to the output, specifically, definite Probabilistic classifiers can perform better when combined with other larger machine learning tasks to avoid the problem of error propagation.

One of the earliest works on classification using statistical approaches was carried out by Andrade et al. (2006) using a 2-grouped problems and discovered a rule for associating an observation to existing groups. Their work is based on the assumption that there exist a multivariable normal distribution for the data values in each of the individual groups. With the assumption that the classification rule is approximately linear, they also considered problems with more than two groups (Görnitz et al., 2015).

According to Ahmadian Ramaki et al. (2018), it is possible to allow for non-linear classifiers with the expansion of multi-variable normal distributions by driving a variety of several classification rules on different points which are fitted usually with a new observation having some form of connection to the group where the corrected viewing distance is the smallest. Bayesian classification usually possess the strong ability to factor in all the necessary useful information about the relative sizes of populations related to various clusters in the total population.

These methods are computationally intensive, and approximations for Bayesian group rules were developed in the days leading up to the development of Monte Carlo calculations of Markov series (Hoang et al., 2003). According to Philip et al. (2018), Several Bayesian methods have much to do with calculating the probability of an observation sequence of belonging to a particular group and can be seen as the output of conducting data analysis that does not just match a particular group to a new observation but also carries a more in-depth information.



Classification problems have been widely adopted to solve problems related to Fraud detection, Medical imaging and image analysis, Optical character recognition, Video tracking, Computer Vision, Pattern recognition, Internet search engines, Statistical natural language processing among others (Ghamri, 2017).

1.1.9 Machine Learning Model Evaluation

According to Vipparla et al. (2021), it has been established that there is no particular single classifier that works best for all problems. However, the number and type of attributes selected highly influence the performance of a particular classifier. The performance however of various classifiers can actually be compared based on certain standards and characteristics and selecting the best classifier for a given machine learning problem is key (Zamini & Hasheminejad, 2019).

It is also necessary to consider evaluation metrics such as accuracy, precision, recall and F1-score as well as resource consumption among others when establishing the efficiency of machine learning models/algorithms. Recently, ROC-AUC curves which evaluates the relationships between True and False positive rates have been widely adopted to determine the performance of machine learning algorithms (Qin et al., 2000).

1.1.10 Hidden Markov Models

HMMs are machine learning algorithms consisting of a number of hidden states and observable outputs for modelling probability distributions over sequences of observations (Rabiner, 1989). In simpler Markov models (such as Markov chains), the state is directly visible to the observer and thus the state transition probability is the only parameter.

According to Cao et al. (2019), the hidden state layer is a stable Markov chain influenced by the initial state probability vector π and the state transition probabilities. The Observable output layer is decided from the emission symbol probabilities which is derived from the observed symbols of each hidden state.



Figure 1.6 depicts the structure of a HMM where $X_i, (i = 1, 2, 3...G)$ represents the various hidden states. Above the dashed line is the markov process in which transition from a state depends only on the current state and then by the hidden state transition Matrix, A . It is important to note that the emission symbols represented by $\sigma_i, (i = 1, 2, 3...H)$ are what is observable but are statistically related to the hidden states using the Emission Transition Matrix, B (Prakash & Chandrasekar, 2012).

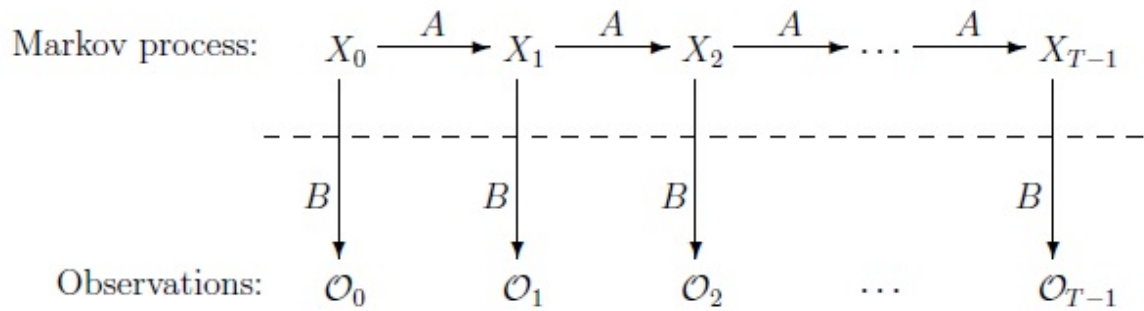


Figure 1.6: Basic structure of a HMM

Where;

- X denotes the possible Hidden States
- Y represents the various Emission symbols
- A denotes the State transition probability Matrix
- B represent the Emission probability Matrix

It is also observed that, transitions among states are governed by the state transition Matrix, A , and sum of all possible transition probabilities from a state must equate to one(1). Likewise, from a particular state, observation symbols are emitted based on the emission transition Matrix B and all possible emission probabilities from a particular state must also sum up to one(1) (Moin et al., 2015). According to Ghahramani (2001), the Initial probability vector, π together with the Transition probability matrix determines the state transition probabilities of the Hidden state layer.



1.1.11 HMM Notation

The application of HMMs ranges from speech and image recognition, intrusion/fraud detection to motion/action analysis in videos among others and is generally characterized by the following; (Bhusari & Patil, 2011);

1. Observation sequence length denoted by S
2. Total number of Hidden states denoted by G
3. Total number of emission symbols denoted by H
4. Unique hidden states in the model denoted by $X = X_0; X_1, \dots, X_{G-1}$
5. A set of possible emission symbols represented as $d = 0, 1, 2, \dots, H - 1$
6. A Hidden State transition probabilities Matrix denoted by A
7. An emission probability matrix represented as B
8. An initial state probability vector denoted by π
9. An Observation sequence denoted by $\sigma = \sigma_0, \sigma_1, \dots, \sigma_{S-1}$

The State transition probability matrix denoted as, $A = [a_{ij}]$ is defined as in Eqn.(1.1) and Eqn.(1.2).

$$a_{ij} = P(q_{t+1} = X_i | q_t = X_j), \quad 1 \leq j, i \leq G, t = 1, 2 \dots T - 1 \quad (1.1)$$

$$\text{Also } \sum_{i=1}^N a_{ij} = 1, \text{ where } 1 \leq j \leq G \quad (1.2)$$

The emission probability matrix denoted by $B = [b_i(d)]$ is defined as in Eqn.(1.3) and Eqn.(1.4):

$$b_i(d) = P(V_d = q_t | X_i = q_t), 1 \leq j \leq G, 1 \leq d \leq H \quad (1.3)$$

$$\text{Also } \sum_{d=1}^H b_i(d) = 1, \quad \text{and } 1 \leq i \leq G \quad (1.4)$$



An initial probability vector denoted as $\pi = [\pi_j]$ is defined as in Eqn.(1.5):

$$\pi_j = P(q_1 = X_j), \quad \text{such that,} \quad \sum_{j=1}^N \pi_j = 1 \quad (1.5)$$

According to Rabiner (1989), the ability of HMMs to solve real-life problems depends on whether they can address these three fundamental problems;

1. The evaluation problem

Given the observation sequence, $\sigma = \sigma_0, \sigma_1, \dots, \sigma_{S-1}$ and the HMM, $\lambda = (A, B, \pi)$, the task is to compute the probability of acceptance of the observation sequence by the HMM, denoted as $P(\sigma|\lambda)$?

2. The decoding problem

Given the observation sequence, $\sigma = \sigma_0, \sigma_1, \dots, \sigma_{S-1}$ and the HMM, $\lambda = (A, B, \pi)$, how can we achieve the optimal state sequence $X = X_0; X_1, \dots, X_{G-1}$ for creating $\sigma = \sigma_0, \sigma_1, \dots, \sigma_{S-1}$?

3. The training problem

The main objective of the training problem is estimating the parameters of the HMM in order to best represent the training data for a specific application.

The increasing use of HMMs over the last several years can be attributed to their high level mathematical structure and strong theoretical basis, enabling them to be used in a wide range of real-life applications (Panigrahi et al., 2009).

An effective optimization of the parameters of these Models for enhanced performance has remained computationally challenging. Several algorithms including the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Baum-Welch (BW), Tabu-Search (TS) and their hybrids have been proposed and implemented to optimize the parameters of these models for optimal performance. Unfortunately, none of these algorithms has proven to always guarantee the best results within reasonable training and detection times (Thatphithakkul & Kanokphara, 2004).



Although the BW algorithm likely guarantees at least a local optimal solution, its high level of sensitivity to preliminary estimates remains a major challenge (Fengqin & Changhai, 2008). The ability of GA to handle effectively several points in a population at a time provides it the capability to avoid being trapped in local optimal solutions. However, when the number of possible solutions (chromosomes) which are exposed to the genetic operators and mutation is large, there is most likely to be an exponential increase in the search space which leads to performance degradation (Haikuan et al., 2019). According to Hassan et al. (2005), PSO algorithm has emerged as a new training algorithm for HMMs based on its simplicity and robust optimization capacity requiring small number of parameters and correspondingly lower number of iterations but has a relatively low convergence speed.

The most common approach in designing Intrusion/Fraud Detection Systems using HMMs is where a single HMM is trained and maintained for each user based on normal user transactions. The forward and backward algorithms are then employed to determine whether or not a particular transaction is genuine or not when the Model is used to compute its probability and compare it to a predefined threshold (Ahmadian Ramaki et al., 2018). According to Abdallah et al. (2016), increased learning time resulting in reduced system performance is one of the significant challenges with this approach making it not ideal for applications that aims to process data sequence online in real-time (Abdallah et al., 2016).

Commonly used at the Host Level are other improved approaches that involves the use of a number of HMMs at different layers with a main focus on reducing false positives and negatives by improving the testing process for a retrieved observation sequence (Ahmadian Ramaki et al., 2018). According to Srivastava et al. (2008), employing multiple HMMs at different layers is capable of reducing False Positive Rate whilst enhancing detection of both known and unforeseen intrusions. During the construction of such models, reasonable sizes of observation sequences such as transaction amounts or frequency of transactions in the case of electronic banking fraud detection are used together with a Database that contains normal transaction sequences of users with varying lengths.

The state transition, observation emission and initial probabilities of the HMM are



then accurately initialized using customers transaction profiles. A comparison between a newly received transaction sequence and valid transaction sequences in the database are made in order to classify a case as genuine or otherwise (Mor et al., 2021).

1.2 Problem Statement

The use of Electronic banking platforms has continued to experience significant growth and has redefined the way banking and Electronic Commerce is conducted across the world (Devi, 2018). On the other hand, fraudulent Electronic banking and E-commerce activities are becoming more sophisticated and challenging leading to massive financial losses. Effective and efficient detection of Electronic banking fraud is therefore regarded as one of the major challenges to all financial institutions, and is an increasing cause for concern (Asare & Sakoe, 2015). According to the Bank of Ghana 2020 banking industry fraud report, fraud cases relating to cyber-crime perpetrated through electronic banking and mobile banking platforms accounts for the highest value of attempted fraud amounting to GH¢1.0 billion in 2020 as against GH¢115.51 million in 2019 (Bank of Ghana, 2020). An effective approach based on Machine learning for real-time fraud detection in electronic banking is the use of HMMs. However, an effective optimization of the parameters of these Models for optimal performance has remained a great challenge to researchers and there is no generally agreed technique that can always guarantee best performance at all times. (Kwong et al., 2001).

Various algorithms such as the BW, PSO, GA, TS and their hybrids have been proposed in literature to perform this task (Shi et al., 2005). The BW algorithm is an Expectation-Maximization algorithm that re-estimates HMM parameters by calculating the backward and forward probabilities of sequences in each iteration. However, this algorithm likely suffers from slow convergence and very sensitive to preliminary estimates (Chang et al., 2018). According to Chen et al. (2004), TS is a heuristic global search method where possible solutions to a given problem is encoded into an array which is a concatenation of parameters of the model. If the solution is not in the Tabu list, the algorithm explores the solution space by way of traveling the neighbors of the contemporary solution. The execution of this algorithm is extremely time-consuming when the



range of hidden states is large. In GA, when the number of possible solutions, known as chromosomes which are uncovered to the genetic operators and mutation is large, there is most likely to be an exponential increase in the search space which leads to a poor performance of the algorithm (Nikraves et al., 2018). According to Haikuan et al. (2019), the PSO algorithm has a simple and robust optimization capacity, but suffers from premature convergence as with other evolutionary algorithms due to variety loss.

Majority of the research in the area of HMM-based fraud detection in Electronic banking focuses only on payments to merchants for goods and services. Transaction amounts are mostly taken as observation symbols and the types of items purchased considered as the hidden states of the proposed HMMs. In related studies conducted by Kovach & Ruggiero (2011), Wei et al. (2013), Sangita & Madhuri (2015), Carminati et al. (2015), Xiaoguo et al. (2018) and Achituve et al. (2019), techniques such as Neural Network , Bayesian Network , Dempster-Shafer theory, Support Vector Machine among others are employed which incorporated other forms of electronic banking options such as remote funds transfers and deposits. However, all these proposed techniques perform classification based on a single transaction while relying on domain-expert features without considering a sequence of transactions to make a decision hence producing high levels of false positives. Additionally, the highly imbalanced nature of banking datasets typical for fraud detection is also another serious challenge when applying Machine learning algorithms to fraud detection. Achieving significant low number of FPs while detecting fraudulent transactions more effectively is the concern of numerous researchers in this field.

In this research therefore, a hybrid algorithm inspired by the BW, GA and PSO algorithms is developed and implemented for optimizing the parameters of HMMs. An improved HMM-based Fraud Detection System (FDS) that incorporates both debit and credit transactions in electronic Banking and the various transaction types, transaction amounts and the frequency of transactions is also proposed and implemented. To determine the transaction profile of customers, a modified Density-based Spatial Clustering of Applications with Noise (DBSCAN) which is capable of discovering clusters of different shapes and sizes from a large amount of data containing noise and outliers was employed. The Synthetic Minority Oversampling Technique (SMOTE) is also employed to handle



the imbalanced class problem typical of Electronic banking datasets.

To significantly reduce the number of False positives, the study further proposes and implements an enhanced multi-layer HMM where the parameters of a single HMM are optimized using various subgroups of the training data which are eventually combined to produce the final Model. For a new transaction to be classified as fraudulent or otherwise, possible matches are identified in a list of normally behaved sequences of transactions in a database and then its probability of occurrence is then computed by the HMM.

1.3 Objectives of the Study

The main objective of the research is to design and implement enhanced HMMs for real-time Intrusion/Fraud detection in Electronic Banking. The specific objectives however are to:

1. Develop an improved hybrid algorithm for optimizing the parameters of HMMs inspired by the BW, GA and PSO algorithms;
2. Propose and implement an enhanced HMM-based framework for real-time fraud detection in Electronic banking and optimize their parameters using the hybrid algorithm as developed in Objective 1;
3. Apply the approach/framework as proposed in Objective 2 to detect real-time fraud on highly imbalanced data to determine its efficiency;
4. Extend the framework as proposed and implemented in objective 3 to an enhanced Multi-layer architecture in order to drastically reduce the rate of false positives in real-time fraud detection.



1.4 Research Questions

The following research questions are formulated in order to achieve the above stated objectives;

1. How can the optimization of HMMs parameters be enhanced through the use of improved hybrid optimization algorithms;
2. How can the proposed Hybrid Optimization Algorithm as proposed in (1) be used to train improved HMM-based framework for real-time fraud detection.
3. What is the applicability of the proposed HMM-based approach/framework be applied to detect real-time fraud on highly imbalanced data.
4. To what extent can the framework as proposed and implemented be extended to an enhanced Multi-layer architecture in order to reduce false positives in real-time fraud detection.

1.5 Significance of the Study

Electronic banking has become a powerful platform to attract and retain customers by financial services firms with the aim of promoting non-cash and alternative payment channels (Helode et al., 2017). According to the Ghana payment systems oversight annual report (2019), Electronic banking has witnessed a tremendous increase in the number of subscribers and transaction value over the last few years. The risk of cyber-attack and intrusions on electronic transactions is on an unprecedented rise resulting in a huge financial loss to financial institutions and individuals.

What raises a major cause for worry is the fact that, the increase in some of the fraud cases are largely due to customers' adoption of digitization and electronic modes of transaction. This is especially the case when the Covid-19 pandemic propelled people who, would have otherwise sought for in-person services, decided to go online in order to respect laid down safety protocols. The movement of transactions digitally and electronically though is positive for the economy, as it pushes the cash lite agenda, it turned out to



bite customers and the banking sector in general, given that there was higher exposure to fraud among electronic transactions.

Despite the various forms of fraud detection and prevention mechanisms adopted by financial institutions to limit drastically if not eliminate risk associated with various fraudulent activities, the situation is not getting any better. Designing an effective and precise fraud Detection system that produces significant low false positives and negatives but detects fraudulent activity effectively to complement the existing efforts of financial institutions is what this research seeks to achieve.

1.6 Scope of the Study

There are several other machine learning algorithms such as artificial Neural Network, Lazy learning, Support Vector Machines, Decision Trees, K-nearest neighbors among others which are employed by various researchers for detecting fraud/intrusions in areas such as credit card transactions, abnormal applications behavior, protecting privacy, internet malware, risk analysis and mitigation, and multistep attack detection and prediction among others (Prakash & Chandrasekar, 2015).

This research however focuses on developing and implementing HMMs for Intrusion detection in electronic banking transactions and therefore limited by the Markov property which assumes that, the conditional probability distribution of future states of the process (conditional on both past and present values) depends only upon the present state, not on the sequence of events that preceded it (Panigrahi et al., 2009).



1.7 Organization of the Thesis

In this chapter, the concepts of Intrusion/Fraud detection in electronic banking transactions. The Objectives, Motivation, Scope and significance of applying HMMs in real-time fraud detection are also outlined in this chapter. Related works in optimizing the parameters of HMMs, detecting real-time Fraud in Electronic Banking and as well developing multi-layer HMMs to reduce False Positives in Real-time Fraud detection with a focus on identifying the limitations of each approach are reviewed in Chapter 2. Related works in the application of HMMs in detecting fraud on highly imbalanced datasets is also presented. The chapter concludes by considering previous works done by other researchers to drastically reduce false positives in real-time fraud detection using enhanced versions of single-layer HMMs. The methodology adopted for the study relating to each of the objectives is outlined in Chapter Three. Using a public benchmark Electronic Banking Dataset, detailed experimental results and discussion to establish the efficiency of the proposed models/algorithms is presented in Chapter Four. Finally, the Summary of findings and conclusions based on the research objectives with some recommendations are outlined in Chapter Five.



CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter presents related works in the design and implementation of algorithms for optimizing the parameters of HMMs. The various techniques available in literature in the application of HMMs to detect real-time fraud in electronic banking on highly imbalanced datasets are also reviewed. The Chapter concludes by reviewing existing works on drastically reducing false positives whilst enhancing training and detection times in real-time intrusion/fraud detection. It is however significant to note that, the outcomes and lapses associated with each approach/technique in all these reviews is what this chapter seeks to unravel.

2.2 Optimizing the parameters of HMMs

The theoretical concept and practicability of HMMs in real-world problems ranging from the simplest to more sophisticated approaches was initially presented by Rabiner (1989). He successfully applied these models to handle problems in speech recognition where optimization of the parameters of these models was performed using the BW algorithm. It was however established from his simulation results that, the inability to escape from local optimal solutions coupled with high level of sensitivity to initial parameter estimates are the challenges encountered by the BW algorithm.

Slimane et al. (1996) proposed and applied GA and BW to optimize the parameters of HMMs. Instead of guessing the initial parameters, the GA is able to find good values for the initial probability vector, Transition and Emission probability matrices, and this initial parameters is optimized further using the BW algorithm. An appropriate genetic



operators and mutation mechanisms was introduced to maintain some constraints on these parameters. Simulations using artificial data revealed great advantages in using a GA with BW in terms of producing global optimal solutions as compared to using only the BW algorithm or GA. The slow convergence rate of their proposed approach was however a challenge.

Chau et al. (1997) rather employed only GA in training HMMs to process signals of speech whose properties transformed over time. A string of real numbers was used to represent the chromosomes comprising two parts composed by concatenating the Transition probability and emission probability rows of the Matrixes. The wheel selection scheme with single and double points crossovers was used until the population of next generation is produced in training their proposed HMM. Experimental results revealed that, HMMs optimised using GA performs better in terms of recognition rates but with lower convergence rate. This was attributed to the fact that, GA is able to accept a probabilistic changes and also able to achieve a global maximum.

Kwong et al. (2001) conducted a comparative analysis of the performance between using GA and BW algorithms in obtaining an optimized number of states and parameters of HMMs. Experimental results also revealed that, the GA required more training time but with better recognition rates as compared to the BW algorithm. It was also noted that, their proposed approach resulted in faster convergence rates as compared to the approaches adopted by Slimane et al. (1996) and Chau et al. (1997).

Thatphithakkul & Kanokphara (2004) applied TS, a technique capable of stepping back from a local optimum and producing global optimal solutions for optimizing the parameters of HMMs. To obtain better results than using BW and GA training, two-hundred (200) iterations of HMM with TS and baseline training were used. Simulation results proved that, when initial parameters are chosen randomly without carefully considering the transaction patterns of users , the proposed approach do not show an improvement over BW and GA.

Shi et al. (2005) presented a variable population-length GA (VPGA) by means of introducing the “demise possibility” inspired by the natural features of the population variable size. In their proposed technique, each character is assigned a death probability



in line with its living generations after the execution of the traditional GA. Primarily based on the variable population-length GA and the PSO algorithms, they also proposed a GA and PSO hybrid algorithm (PGHA). Simulated results for the optimization of nonlinear features proved that the PGHA algorithm is better than the PSO and GA in terms of producing global optimal solutions although more training time is required.

The PSO algorithm was also employed by Xue et al. (2007) to optimize the parameters of HMMs in speech recognition. As compared to the BW algorithm which was proposed and implemented by Rabiner (1989) which are hill-climbing and hardly to escape local optimal solutions, their proposed method was formulated in a way to discover global solutions or at worse better local optimal solutions. Simulation results confirmed their proposed approach is most effective comparatively to using the BW and GA-HMM training methods although the selection of initial parameters needs considerable attention.

A hybrid of PSO and BW algorithms for optimizing the parameters of HMMs was proposed by Fengqin & Changhai (2008). The positions of the newly discovered particles from the PSO algorithm are locally improved using the BW Algorithm. In their proposed approach, a specific unit correspond to a model where the training data contains a number of data sequences and the points which are not precisely known. Simulation results revealed that, their proposed hybrid algorithm overcame the slow convergence rate of the PSO algorithm and also aided the BW algorithm to achieve a global solution making it superior to the BW algorithm in terms of recognition ability. There was however a performance degradation when the number of Hidden states and observation symbols increases.

In order to improve the convergence rate in optimizing the parameters of HMMs, Yang et al. (2008) combined PSO and BW algorithms to obtain a hybridized set of rules and compared its performance to a hybrid algorithm obtained from merging GA and BW algorithms. They performed a number of iterations using the BW algorithm and the various solutions were supplied to both PSO and GA as particles and chromosomes respectively with appropriate fitness functions adopted. From experimental results, the hybrid algorithm of PSO and BW algorithms performed much better in terms of convergence and recognition rates compared to the hybrid of GA and BW but requires more training time.



Hewahi (2015) proposed that, given a set of definite Hidden states and observation symbols, an approach based on PSO concepts to obtain an optimal HMM structure design with better performance is feasible. Generating new states and updating likelihood values were the new components he included in the optimization process. In their proposed approach, they began by producing random parameters for each HMM and then converting the generated HMM to PSO required format. These PSO formats are then used to determine the most appropriate HMM parameters for optimal performance. The importance of the proposed strategy is that, different strategies deal solely with probability updating. Their proposed method is regarded to be the first that evolved HMM in terms of probability values and states. For optimal performance however, their models required large amount of data for training.

Another hybrid algorithm comprising the PSO and TS for optimizing the parameters of HMMs was proposed by Fang & Zhang (2016). Their proposed model possesses a fast convergence speed and a guaranteed global solution by incorporating the advantages of both the TS and PSO. The novelty in their proposed approach is that, a network random key is introduced which completely guarantees the generation of only feasible solutions during the training process allowing the new hybrid algorithm to perform to its fullest potential. Simulation results confirmed that, when compared to other available techniques in literature, it has a better recognition rates and computationally efficient. However their proposed models performed poorly when applied to solve problems in different related domains.

Shuangqing et al. (2018) proposed and implemented a hybrid algorithm involving the PSO and fireworks with a focus on finding global solutions to problems with relatively faster convergence rate as compared to the other hybrid algorithms proposed in literature at the time. In their proposed algorithm, some operators of Fireworks Algorithm are adjusted and incorporated into the optimization technique of the PSO algorithm the an enhanced explosion and mutation operators are meant to facilitate the global convergence of the proposed hybrid algorithms to avoid pre-convergence. Experimental results showed that, the proposed hybrid PSO-FW algorithm is comparatively more robust, efficient and fast converging for finding global solutions to optimization problems.



Chang et al. (2018) also optimized the parameters of HMMs with PSO algorithm with a main focus on finding global solutions to optimization problems which otherwise would be too complex to handle using the BW, GA and TS Algorithms. Their proposed approach incorporated a re-normalization and re-mapping mechanisms so as to effectively overcome the statistical constraints in HMMs. Based on experimental results, they achieved a better optimization ability than BW, GA and TS algorithms, with a faster convergence speed but performs poorly when the number of observation symbols and hidden exceeded 3.

2.3 Detecting real-time electronic banking fraud on highly imbalanced datasets using HMMs

Fraud Detection in Electronic Banking is understudied in literature perhaps due to security and data privacy concerns. Related works in electronic banking in general are reviewed and then those related specifically to the use of credit cards are also given considerable attention. Kovach & Ruggiero (2011) proposed and implemented an online banking Fraud detection system by employing differential analysis to discover local evidence of fraud where a potential fraud is indicated by a significant deviation from normal trained profile. To obtain a suspicion score, they employed the dempster's rule to combine these evidences. A novel idea in their approach was the ability to detect fraud based on an efficient identification of devices used in assessing a particular account. However, their system performs poorly in terms of precision and recall for higher number of hidden states and also when users' transaction patterns changes frequently.

VanHoose (2011) proposed and implemented an intelligent real-time HMM-based Fraud detection system capable of identifying abnormal user behavior in online banking transactions. Due to the uncertain nature of customer's transaction behavior, fuzzy theory was employed to detect customers behaviors to categorize potential fraudulent transactions with varying intensity levels. Based on simulation results their proposed system obtained an accuracy of 94 percent but with low precision and Recall making it not ideal for real-time fraud detection.



Mhamane & Lobo (2012) considered transaction amounts and purchases types as the emission symbols and hidden states respectively in their proposed HMM for online banking Fraud detection system. The model is trained with the normal behavior of an account holder using BW algorithm and a One-time-Password is sent to the Customers contact for authorization if an incoming transaction violates the Behavior sequence. Although, the accuracy of their system was close to 72 percent over a wide variation in the input data, False Positives were still high especially when the transaction data is highly skewed.

With the main aim of effectively detecting electronic banking fraud on highly imbalanced data, Wei et al. (2013) combined several data mining techniques by constructing a contrast vector for each transaction by employing customer's normal historical transaction profile. Their proposed approach is able to effectively identify patterns within a dataset to distinguish fraudulent from genuine behavior after which a pattern selection and risk scoring mechanism is used to combine predictions from the various different models. By incorporating domain knowledge and traditional fraud detection methods, simulation results using online banking data demonstrated that, the proposed system is able to accurately identify fraud and with comparatively lower number of false positives. there was however a performance degradation especially when the transaction data is highly imbalanced.

Sangita & Madhuri (2015) modeled the sequence of operations in online banking transaction processing using HMMs and outlined how it could be used for the detection of frauds. The observation sequence length is fixed to two, whilst varying the sequence length for training i.e., changing dataset length from 10 to 80 with difference of ten. Simulation results revealed that, although the complexity of the system also increases for increased observation sequence length, the accuracy of the proposed system is close to 60 percent with reduced false Positive rate. However, there is an increased complexity leading to Poor performance when number of states exceeds 2.

Xiaoguo et al. (2018) employed HMMs and k-means algorithm for detecting fraud in online banking transactions. In their proposed model, a variable is used to keep the number of transactions within a period of time before and after each transaction as well as the quantified amounts as the observation symbols. If an incoming transaction is



not accepted by the trained HMM with sufficiently high probability, it is considered fraudulent. The feasibility of their proposed model is demonstrated through simulation experiments using real-world bank transaction data. In the case of enough historical transactions, their model performs well for low, medium frequency and amount of user groups. An efficient Prior determination of the number of clusters using the K-means clustering technique is considered a major challenge in their proposed approach.

Specifically on fraud detection relating to the use of Credit Cards, Srivastava et al. (2008) considered purchase types and transaction amounts as hidden states and observation symbols respectively in their proposed HMM. In order to estimate the model parameters, the K-means clustering algorithm is employed to determine the spending profile of cardholders. An incoming transaction is considered fraud if it is not accepted by the HMM with a significantly high probability. Experimental results revealed that, their proposed model recorded an accuracy close to eighty (80) percent over a wide variation of the data. An efficient prior determination of the number of clusters and significant number of false positives were considered major drawbacks of their proposed approach.

A four-level security system including an Image Click Point Authentication for Credit Card Fraud Detection using HMMs was proposed by Helode et al. (2017). The proposed system produced better detection results by incorporating secret authentication and a strong transaction checking rather than relying solely on only user behavior-based security. Based on the details of a credit card and the value of purchase and goods, the Model classifies a transaction as fraud or otherwise with reference to the spending profile of the card holder. If a transaction is classified as fraud, an alarm is raised and proceeds to the Image Click Point Authentication (ICPA) validation where some questionnaire are provided for the user to enhance security. Based on simulation results, the system provides close to 95 percent detection accuracy but with low precision and recall with a significant number of false positives especially when the data is highly skewed data.

Talekar (2015) modeled the operational phases in credit card transaction where the detail of a transaction location is traceable from the IP address. A HMM is used to detect the authenticity of a transaction that considers the transaction amount as a parameter, and then the DBSCAN algorithm rather takes the location of the transaction as



its parameter. The card holder is alerted via an email if a transaction fails to go through any of these phases successfully. Although their proposed system was seen as an initial framework which has been tested on a dataset of limited records, the results generated from it was promising. Their proposed approach is however very insensitive to frequent changes in transaction pattern of users and also performs poorly when the dataset is highly imbalanced.

A comparative analysis of the performance of various Fraud detection techniques such as Decision Trees, Random Forest, Support Vector Machines (SVM) and logistic regression on highly imbalanced Transaction data was performed by Khare et al. (2018). They randomly partitioned the dataset into equal number of s -subgroups where each group was held back for validation purposes whilst the remaining $s-1$ subgroups were used for training data. Simulation results using a dataset provided by the Universit Libre de Bruxelles (ULB) machine learning revealed that, Logistic regression and Random forest performed best in terms of precision and accuracy but requires very large dataset for training and also suffer from performance degradation when the dataset is highly imbalanced and even after appropriate data pre-processing techniques are applied.

Wedge et al. (2019) proposed a model based on automated feature engineering to automatically derive behavioral features based on the historical data of a credit card associated with a transaction. A number of features for each transaction is generated, and random forest classifier is then employed to train the model. One important feature of their proposed approach was the fact that, it utilizes the distance between two locations transactions on an account has occurred and whether it happened in person or remotely. The proposed model was tested on data from a large multinational bank and compared to existing solutions and revealed that, on an unseen data of almost 2 million transactions, false positives were reduced by about 54 percent as compare to other fraud detection techniques at the time. However, since their models Performed classification based on a single transaction there was a performance degradation when transaction pattern of users changes frequently.



2.4 Reducing False Positives in Real-time Fraud Detection using Multi-Layer HMMs

A multi-layer HMM-based approach for anomaly detection where system calls are processed to build a normal program behaviors with enhanced detection process as compared to single-layer HMMs making them suitable for real-time anomaly detection was proposed by Hoang et al. (2003). Simulation results established that, although the proposed model is specifically a host-based intrusion detection scheme, it also proved very efficient when applied in environments that is networked. Although more training time is required, their proposed model proved better in detecting anomalous behavior of programs in terms of accuracy and response time as compared to available existing techniques at the time.

Penagarikano & Bordel (2004) outlined and demonstrated the theoretical and practical concepts of Multi-layered HMMs and applied it to automatic speech recognition with the main purpose of drastically reducing false positives. They presented each level of knowledge by various set of Models organized in a form of a layered structure. Their proposed approach revealed the possibility a developing and implementing a new architectural method to recognition problems using a multi-layer approach. They still however recommended the adoption of adoption more improved multi-layer HMMs to improve on the training and detection times to further reduce false positives and negative rates.

Abouabdalla et al. (2009) in an attempt to reduce false positive alerts in intrusion detection systems, proposed and implemented an improved HMM-based FDS by incorporating correlation methods which has a potential of handling extremely large imbalanced amount of both real and false alerts. The techniques they employed possessed the tendency of assisting the system administrator to efficiently analyze these alerts to distinguish between those generated by real intrusions. Unfortunately, the number of false positives was still significant especially when the data is highly skewed and when transaction patterns changes frequently.

An improved Intrusion/Fraud detection System build with the capability of filtering out outliers after data pre-processing to significantly reduce false positives specifically in networked environments was proposed by Spathoulas & Katsikas (2010). The filtering



process utilizes various components that are based on statistical properties of the input alert set. They discovered special features of notifications that has some relationships with identified and confirmed intrusions which can be displayed in a sequence of batches containing a resemblance in destination and source IPs otherwise, anomalies are identified in the distribution of notifications of the same signature. In their proposed approach, it is possible to identify FPs by the rate at which their signature triggers these false positives. Simulation results obtained by employing the DARPA 1999 dataset indicate that the proposed approach significantly reduces false positives up to about 75 percent when compared to other intrusion detection techniques at the time, although there was a performance degradation when transaction patterns of users change frequently.

Xu et al. (2013) however focused on making the patterns of intrusions that are weak much more visible for easy identification and prevention. To this end, they proposed and implemented a multi-module intrusion detection system where they employed the Viterbi algorithm in the prediction of the most likely next intrusion more efficiently. Experimental results confirmed the effectiveness of their proposed scheme in significantly detecting intrusions in terms of the reduced false alarm rate. However, they still recommended a detail investigation into more enhanced intrusion detection methods to further reduce false alarm rates.

Bandgar (2013) proposed and implemented a novel Multi-layer HMM with the main focus of detecting Internet attacks. They constructed a signature of attacks where these signatures embodies specific traffic or activities that is based on known intrusive activity. After performing simulation analysis using single-layer and multiple-layer HMMs model for source separation both on IP and port information of source and destination, it was obvious that, their proposed approach succeeded in reducing the false positive rate making this type of source separation the basic step for building their proposed HMM. The challenge however with their approach was significant high number of false positives and negatives with increased training times.

A more efficient system to detect and prevent intrusion that produces very negligible number of false-positive alarms by utilizing risk analysis and assessment mechanisms was proposed and implemented by Qassim et al. (2014). As part of future enhancement



plans, they proposed to extend this idea by implementing it with the use of information security ontology curved out of the classic components of risk analysis (assets, threats, vulnerability and countermeasure). They also recommended the implementation of risk assessment using fuzzy logic and autonomic computing which allow for features such as self-configuring, self-optimization, self-detection, self-protection, self-prevention and self-healing. They are of the conviction that, autonomic computing has the potential of dramatically improving the detection rates whilst enabling the development of the knowledge-base of new detected attacks and thereby significantly reducing false alarm rates.

Prakash & Chandrasekar (2015) proposed and implemented an optimized Multiple-layer Semi-HMMs and employed the Cuckoo search algorithm for fraud detection in credit card fraud detection. The appropriate number of states and the optimal model parameters are determined by the Cuckoo Search algorithm. Simulation results demonstrated that, although the number of false positives were still significant, their proposed approach could effectively detect fraud better when compared to most existing techniques at the time.

Li & Lau (2016) employed Multi-layered HMM to fit the spatial-temporal trajectories by decomposing a problem into distinct layers that HMMs and each operated at different spatial granularity independently. They went ahead to utilize the BW algorithm and Viterbi algorithm to optimize the parameters of the proposed models. Simulation results reveals that, their proposed approach is efficient and more stable in handling intrusions with large number of hidden states when compared to the standard single-layer HMMs in handling optimization problems especially those related to intrusion detection.

Goeschel (2016) demonstrated the possibility of maintaining high accuracy whilst drastically reducing the rate of false positives by developing and implementing a IDS comprising Decision trees, Support Vector Machines and Naive Bayes. They began by training the SVM algorithm using an improved binary classification approach and then passed any suspicious intrusion through the decision tree for classification. The Naïve Bayes and decision tree are then employed to finally determine the faith of any potential intrusion. They however recommended further work on testing this model on other sample Datasets for more in-depth analysis to improve detecting rates and adjust appropriately to frequently



changing patterns of attacks.

An Intrusion detection system based on HMMs using a multi-layer approach was proposed and implemented by Zegeye et al. (2018) where the system is used to verify and attempt to resolve some common challenges such as the curse of dimensionality which typically arises when HMMs are applied to intrusion detection problems. They also demonstrated the effectiveness of their proposed approach when expanded beyond 2 layers where multiple attacks that spans over a longer periods can be detected effectively. The concepts of this novel approach have been developed but the full potential has not been demonstrated especially on its computational complexity.

Burgio (2019) in his PhD research also proved the possibility of significantly minimizing false positives whilst maintaining the detection accuracy of an IDS by incorporating extreme Learning Machine concepts into HMMs. Simulation results based on the University of New South Wales-Network Based 2015 (UNSW-NB15) data set which is more representative of contemporary cyber-attack and normal network traffic revealed that, their proposed approach provides better results than either employing only HMMs or extreme Learning Machines alone and also achieved significant lesser number of False Positives than other comparable approaches that also used the UNSW-NB15 data set, but with requires more time for training.



Zegeye et al. (2019) also proposed Multi-layered HMM for Intrusion Detection that addresses multi-stage attacks due to the fact that intrusions are increasingly being launched through multiple phases instead of single stage intrusion. This layered model divides the problem space into smaller manageable pieces reducing the curse of dimensionality associated with HMMs. Although the system performed better when compared to a single-layer approach, significant number of false positives and negatives.

2.5 Chapter Summary

This chapter reviewed various algorithms and techniques that has been proposed and implemented by various researchers for optimizing the parameters of HMMs with a focus effort on outlining the outcomes and limitations associated with each. It is established that, the BW algorithm is very sensitive to initial estimates and can easily be trapped in local optimal solution. GA performs better than BW algorithm procedure in terms of recognition rate but has a slower convergence rate. A hybrid of GA and BW algorithms overcomes the slow convergence rate of a simple GA-HMM approach but requires more training time.

In TS, when initial parameters and factors are not carefully chosen, it does not show a great improvement over the GA and BW algorithms. A hybrid of BW and PSO algorithms performs better as compared to the BW algorithm and the hybrid of GA and PSO algorithms but Performs poorly with larger number of Hidden states. A hybrid of GA, TS and BW algorithms improved on the slow convergence speed of GA and TS and ensured that the BW algorithm converges to a global optimal solution. The challenge however is the significant increase in training time and a performance degradation when number of hidden states and observation symbols are large.

The chapter also performed a critical analysis of the various techniques in the application of HMMs in intrusion detection especially when the dataset used is highly imbalanced. With approaches that incorporated the various devices used to access a particular account, fraud suspicion increases with the number of accounts accessed from the same source but performs poorly with large number of Hidden states and also when transaction pattern of a user changes frequently. On the other hand, when only Transaction amounts



and purchases types are considered, significant number of False positives are obtained especially when data is highly imbalanced. Utilizing transaction amounts and frequency of transactions with K-Means to determine the transaction pattern of users, the models performs well only for some particular transaction profile groups. When transaction pattern of users changes frequently ,an effective prior determination of the number of clusters for optimal performance using K-means remains a challenge.

Although Multiple-Layer HMMs architectures as proposed and implemented in literature are comparatively better than single-layer HMMs in terms of recognition rates and precision , higher training or detection times remains a challenge. Incorporating risk analysis and assessment resulted in a further reduction of the false positive rate but a reduced performance when the number of states and observation symbol are large.

Based on the drawbacks identified, the next chapter focuses on developing an enhanced algorithm for optimizing the parameters of HMMs and applying that algorithm to optimize single-layer and Multiple Layer HMMs with enhanced training and detection processes to detect fraud in Electronic banking on highly imbalanced data.



CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter presents the methodology adopted to achieve each of our stated research objectives. It begins by presenting the methodology adopted in the design and implementation of a Hybrid algorithm for optimizing the parameters of HMMs inspired by the BW, GA and PSO algorithms.

The next section outlines the approach adopted to effectively detect and prevent real-time fraud in Electronic banking on highly imbalanced datasets using HMMs. We conclude the chapter by outlining the methodology adopted in developing an improved Multiplayer HMM to drastically reduce false positives in anomaly/fraud detection with improved training and detection times.

3.2 An improved Hybrid Algorithm for optimizing the parameters of HMMs

It has been established from literature that, the effective training of HMMs for enhanced performance has remained computationally challenging and there is no generally agreed method that can guarantee best performance within reasonable time. Although the BW algorithm is always guaranteed to converge to a local maximum of the likelihood function, it likely suffers from slow convergence and very sensitive to preliminary estimates.

GA although searches from a population of possible solutions more effectively by employing an objective function, when the number of possible solutions (chromosomes) which are uncovered to the genetic operators and mutation is large, there is most likely to be an exponential increase in the search space which leads to a terrible performance of



the algorithm. PSO algorithm has emerged as a new training algorithm for HMMs based on its simplicity and robust optimization capacity but has a potential of falling into local optimum in high-dimensional space with low convergence rate in the iterative process.

In this section therefore, a hybrid algorithm inspired by the BW, GA and PSO algorithms is developed and implemented for optimizing the parameters of HMMs.

3.2.1 The Baum-Welch Algorithm

Consider an Observation sequence, $\sigma = (\sigma_0, \sigma_1, \sigma_2, \dots, \sigma_{T-1})$ and a possible state sequence $X = (X_0, X_1, X_2, \dots, X_{T-1})$ where the interest is to compute the probability of the Observation sequence with respect to a given HMM, λ . The probability of an observation sequence using a direct computation is computed as in Eqn.3.1).

$$P(\sigma|\lambda) = \sum_X \pi_{x_0} b_{x_0}(O_0) a_{x_0, x_1} b_{x_1}(O_1) \dots a_{x_{T-2}, x_{T-1}} b_{x_{T-1}}(O_{T-1}) \quad (3.1)$$

To estimate $P(O|\lambda)$, a forward algorithm known as $\alpha - pass$, is adopted. For $t = 0, 1, 2, \dots, T - 1$ and $j = 0, 1, 2, \dots, N - 1$, the probability of the partial observation sequences when the system is in state X_i at time t denoted as $\alpha_t(j)$, is defined in Eqn.(3.2).

$$\alpha_t(j) = P(\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_t, q_t = X_j | \lambda) \quad (3.2)$$



$\alpha_t(j)$ is obtained recursively as from Eqn.(3.3) to Eqn.(3.4):

1. For $i = 0, 1, 2, \dots, N-1$

$$\alpha_0(j) = \pi_i b_j(O_0), \quad (3.3)$$

2. For $i = 0, 1, 2, \dots, N-1$ and $t = 0, 1, 2, \dots, T-1$, we compute

$$\alpha_t(j) = \sum_{i=0}^{N-1} [\alpha_{t-1}(i) a_{ij}] b_j(O_t) \quad (3.4)$$

3. From Eqn.(3.4) it is clear that

$$P(\sigma|\lambda) = \sum_{j=0}^{N-1} \alpha_{T-1}(j) \quad (3.5)$$

To determine the optimal state sequence, the most probable symbol at each position is chosen denoted as $\beta - pass$

For $t = 0, 1, 2, \dots, T - 1$ and $j = 0, 1, 2, \dots, N - 1$, $\beta_t(j)$ is defined as in Eqn.(3.6);

$$\beta_t(j) = P(\sigma_{t+1}, \sigma_{t+2}, \sigma_{t+3}, \dots, \sigma_{T-1} | q_t = X_j, \lambda) \quad (3.6)$$

Then $\beta_t(j)$ can be computed recursively as follows as in Eqn.(3.7) and Eqn.(3.8):

1. for $j = 0, 1, 2, \dots, N - 1$, let

$$\beta_{T-1}(j) = 1 \quad (3.7)$$

2. For $j = 0, 1, 2, \dots, N - 1$ and $t = T - 2, T - 3, T - 4, \dots, 0$, $\beta_t(j)$ is defined as in Eqn.(3.8);

$$\beta_t(j) = \sum_{i=0}^{N-1} [\beta_{t+1}(i) a_{ji}] b_j(\sigma_{t+1}) \quad (3.8)$$



For $t = 0, 1, 2, \dots, T - 1$ and $j = 0, 1, 2, \dots, N - 1$, $\gamma_t(j)$ is defined as in Eqn.(3.9);

$$\gamma_t(j) = P(q_t = x_j | \sigma, \lambda) \quad (3.9)$$

From Eqn.(3.2) and Eqn.(3.6), Eqn.(3.10) is obtained:

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{P(\sigma|\lambda)} \quad (3.10)$$

For $j, i \in (0, 1, 2, \dots, N - 1)$ and $t = 0, 1, 2, \dots, T - 1$;

$\gamma_t(j, i)$ which represents the probability of the system transitioning from state q_j to state q_i is obtained in Eqn.(3.11):

$$\gamma_t(j, i) = P(q_t = x_j, q_{t+1} = x_i | \sigma, \lambda) \quad (3.11)$$

In terms of α, β, A and B , the di-gammas, $\gamma_t(j, i)$ from (16) is defined as in Eqn.3.12);

$$\gamma_t(j, i) = \frac{\alpha_t(j)a_{ij}b_i(O_{t+1})\beta_{t+1}(i)}{P(\sigma|\lambda)} \quad (3.12)$$

The relationship between $\gamma_t(j, i)$ and $\gamma_t(j)$ is represented as in Eqn.(3.13);

$$\gamma_t(j) = \sum_{i=0}^{N-1} \gamma_t(j, i) \quad (3.13)$$

The Model parameters can then re-estimated as from Eqn.(3.14) to (3.16):

1. For $j = 0, 1, \dots, N - 1$, let

$$\pi_j = \gamma_0(j) \quad (3.14)$$

2. For $0 \leq j, i \leq N - 1$, a_{ji} is computed as in Eqn.(3.15);

$$a_{ji} = \frac{\sum_{t=0}^{T-2} \gamma_t(j, i)}{\sum_{t=0}^{T-2} \gamma_t(j)} \quad (3.15)$$



3. For $i = 0, 1, \dots, N - 1$, and $d = 0, 1, \dots, H - 1$, $b_i(d)$ is computed as in Eqn.(3.16):

$$b_i(d) = \sum_{t \in (0,1,2,\dots,T-1), \sigma_t=k} \gamma_t(i) / \sum_{t=0}^{T-1} \gamma_t(i) \quad (3.16)$$

It is desirable to terminate the process if the value of $P(O|\lambda)$ remain unchanged for some consecutive number of iterations or by at least some predetermined threshold.

3.2.2 Particle Swarm Optimization Algorithm

Consider an objective function, F_n formulated on a search space P , for a particular swarm with m particles, A D-dimensional vector $X_j(j = 1, 2, \dots, n)$ is represented by the j^{th} particle which indirectly implies that $X_j(t) = (x_j(1), x_j(2), \dots, x_j(d))$ where $j = 1, 2, \dots, n$ are the possible locations to in the search space to find the j^{th} particle.

It is also worth noting that, potential results are represented by the position of each particle in the search space. A D-dimensional vector, denoted as $V_j(t) = (v_j(1), v_j(2), \dots, v_j(d))$ where $j = 1, 2, \dots, m$ also denotes the velocity of the j^{th} particle. From the beginning of the search, each particle identifies a particular solution which is considered the best represented as $X_j^+(t)$ whereas $\hat{x}_j(t)$ denotes the best position otherwise known as the social knowledge discovered by the entire swarm.

According to Daniel & Macas (2014), the following are formulated based on each particle j , at time t , to obtain a global solution.

- $X_j(t)$: It's current position in the search space S ,
- $V_j(t)$: It's current velocity,
- $X_j^+(t)$: It's best solution discovered since the beginning of the search.
- $\nu_j(t)$: The closest particles to it considered as its neighborhood at time t .
- $\hat{x}_j(t)$: The best position of the entire swarm which represents it's social knowledge.



As represented in Eqn.(3.17) and Eqn.(3.18) respectively, the position and velocity is computed for a given particle;

$$X_j(t + 1) = X_j(t) + V_j(t) \tag{3.17}$$

$$V_j(t + 1) = wV_j(t) + r_1v([0, 1])(X_j^+(t) - x_j(t)) + r_2v([0, 1])(\hat{x}_j(t) - x_j(t)) \tag{3.18}$$









The term $v([0, 1])$ is a uniform random value chosen from 0 to 1 and the parameters w , r_1 and r_2 controls the entire particle system. However the velocity of a particle is controlled by the variable w considered as the an inertia weight.

The movement of some particles in a search space is represented in Fig 3.1 where $X^1(t)$ and $X^2(t)$ as indicated by the red dotted and blue circles respectively represents two particles. It is also observed that, $P^1(t)$ and $P^2(t)$ denotes previous best positions obtained by the two particles respectively with a single global best solution (G). The following directed velocities as outlined are shown in the diagram;

- v^1 and v^2 denotes the initial direction of particles
- v_p^1 and v_p^2 represents the direction of the previous best positions
- the v_g^1 and v_g^2 denotes the direction of the best positions

$(v^1(t + 1)$ and $v^2(t + 1))$ denotes the velocities of the particles and computed as in Eqn.(3.18). From Fig. 3.1, it is clear that, in the next iteration $t + 1$, the positions of the two particles inches closer to the desired global solution.



 Particle 1 (Current Position $x_{(t)}^1$)	 Original Velocity ($v_{(t)}^i$)
 Particle 1 (Next Position $x_{(t+1)}^1$)	 Velocity to P (v_p^i)
 Particle 2 (Current Position $x_{(t)}^2$)	 Velocity to G (v_G^i)
 Particle 2 (Next Position $x_{(t+1)}^2$)	 Resultant Velocity ($v_{(t+1)}^i$)

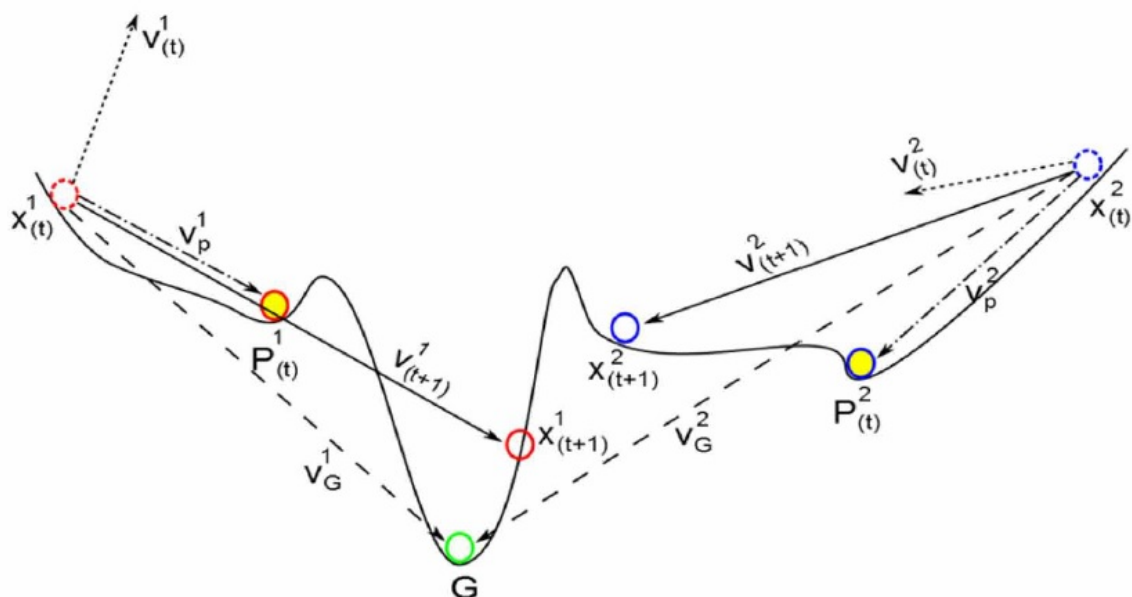


Figure 3.1: A diagrammatic representation of the movement of particles using PSO algorithm

Note: Adopted from Daniel & Macas (2014), <https://www.researchgate.net/publication/25364472>

Particle-swarm-optimization-of-hidden-markov-models-A-comparative-study



3.2.3 Genetic Algorithm

Based on the likelihood/fitness function, $P(\sigma|\lambda)$, the quality of chromosomes are determined using the following steps outline in Fig. 3.2.

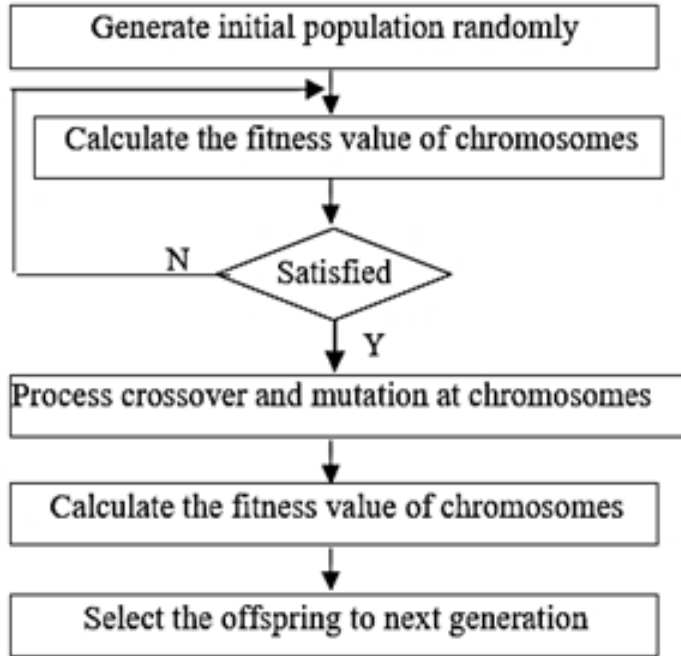


Figure 3.2: Structure of GA

Note: Adopted from Xiao et al. (2007), <https://www.atlantis-pess.com/proceedings/iske2007/121>

Specifically, the GA incorporated into the proposed hybrid algorithm is as follows;

1. At $t = 1$, The fitness Values of an initial population, $N(t)$ of 100 individuals (chromosomes) generated from using the BW algorithm where each individual corresponds to a HMM are evaluated.
2. $t = t + 1$ and While($t \leq 100$)
3. The best 50 individuals is created in the next population of $N(t)$ by:
 - Selecting randomly two chromosomes, λ_a and λ_b from the population
 - Recombining them with a multiple point crossover operator to obtain λ_c
 - λ_c is mutated into λ'_c
 - $N(t + 1) = N(t) \cup \lambda'_c$
4. Let $N'(t)$ denote the best 100 individuals obtained using the proposed GA



5. The PSO algorithm is then applied on the obtained $N'(t)$ individuals

A linear recombination crossover as adopted by the study is as outlined below;

- The off-springs are obtained as a weighted sum of two parent models λ_a and λ_b ,
- The sum of each row in the Initial probability vector, Transition probability and observation Emission Matrices always equals 1.
- Given two rows of the matrix, $R_1 = (X_1, X_2, \dots, X_N)$ and $R_2 = (Y_1, Y_2, \dots, Y_N)$, a new row, R_3 , is computed using the linear recombination crossover as in Eqn.(3.19)

$$R_3 = \alpha R_1 + (1 - \alpha)R_2 = (\alpha X_1 + (1 - \alpha)Y_1, \quad (3.19)$$
$$\alpha X_2 + (1 - \alpha)Y_2, \dots, \alpha X_N + (1 - \alpha)Y_N)$$

The variable α denoted by αX is set to 0.5 for all crossovers so that the sum of the coefficients in each new row equals 1 as required.

3.2.4 Existing Hybrid Algorithms that combines GA, PSO and BW Algorithms for optimizing the Parameters of HMMs

The GA and PSO algorithms are much similar in their inherent parallel characteristics, whereas experiments show that they have their specific advantages when solving different problems. A hybrid optimization algorithm involving the GA and PSO algorithms as proposed by (Shi et al., 2005) is depicted as in Fig.(3.3). The steps outlined in the proposed hybrid algorithm is as follows;

1. Initialize GA and PSO sub-systems, respectively.
2. Execute GA and PSO simultaneously.
3. Memorize the best solution as the final solution and stop if the best individual in one of the two sub-systems satisfies the termination criterion.
4. Perform hybrid process if generations could be divided exactly by the designated iterative times N . Select P individuals from both sub-systems randomly according to their fitness and exchange. Go to step 2.



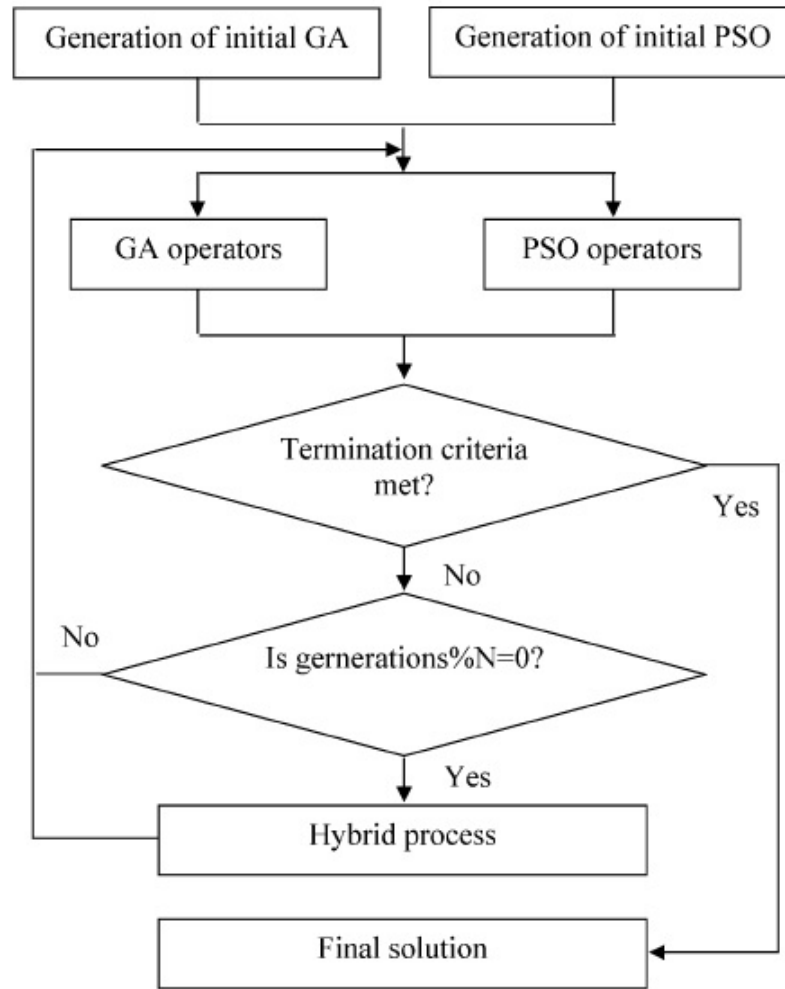


Figure 3.3: Flow chart of a GA-PSO hybrid optimization algorithm

Note: Proposed by Shi et al. (2005), <https://www.researchgate.net/publication/222694693-An-improved-GA-and-a-novel-PSO-GA-based-hybrid-algorithm>.



A hybrid GA and BW algorithm is also proposed for optimizing the parameters of HMMs due to the slow convergence and high computational power needed by the conventional GA, especially when the generated chromosomes cannot satisfy the row stochastic property of the Initial(π), State transition (A) and Emission probability (B) matrices as in Eqns. (1.1) to (1.5) respectively (Kwong et al., 2001).

For this reason, these chromosomes are not included among the offsprings and they are replaced by new chromosomes. For every ten (10) generations, three iterations of B-W algorithm are applied to all chromosomes to allow the algorithm to converge faster. To overcome the slow convergence problem, the initial generation $P(O)$ is not generated randomly as in normal GA, but by applying an iteration of B-W algorithm as outlined below;

1. Apply B-W algorithm to generate the initial population $P(O)$, where $P(O) = C_1, C_2, \dots, C_N$, and C_i is a single chromosome.
2. Calculate the fitness function $F(C_i)$ of every chromosome C_i within the actual population $P(t)$.
3. Select a few chromosomes for the intermediate population $P'(t)$.
4. Apply crossover to some chromosomes in $P'(t)$.
5. Apply mutation to few chromosomes in $P'(t)$.
6. Apply three iterations of B-W algorithm to the population $P(t)$ for each ten(10) generations.
7. $t = t + 1$; if not convergence, then go to step (2).

As depicted in Eqn(3.20), the log likelihood is employed in the fitness function which represents the probability that the training observation sequences are been generated by the current model parameters.

$$p(n) = \left(\sum_{K=1}^M \log(P(O_k/\lambda_n)) \right) / m \quad (3.20)$$



For Mutation and Crossovers, the 'mutationadaptfeasible' and the default GA Toolbox selection and crossover functions respectively in Python are employed to satisfy the row stochastic property of the Initial, State Transition and Emission probability Matrices.

As mentioned earlier, the PSO algorithm possesses a global search ability with no gradient information required. When combined with the BW algorithm exhibits good properties of global search and efficient local search. In a proposed algorithm, the BW algorithm is applied to the new positions of particles in order to locally improve their positions as outlined below (Fengqin & Changhai, 2008);

1. Set the initial parameters including the maximum iterative count $IterCount$, $PCount$ and $BWCount$, the population size $Psize$, w , c_1 and c_2 .
2. Initialize a population of particles with random positions and velocities in the search space.
3. Set iterative count $Gen1 = 0$.
4. Set iterative count $Gen2 = 0$.
5. For each particle i , update the position and velocity as in Eqn.(3.17) and Eqn.(3.18) respectively.
6. $Gen2 = Gen2 + 1$.
7. If $Gen2 < PCount$, go to 5.
8. Set iterative count $Gen3 = 0$.
9. For each particle i do
10. Apply the BW algorithm to the position of particle i .
11. $Gen3 = Gen3 + 1$.
12. If $Gen3 < BWCount$, go to 9.
13. $Gen1 = Gen1 + 1$.
14. If $Gen1 < IterCount$, go to 4, otherwise terminate.



3.2.5 The Proposed Hybrid Optimization Algorithm

Fig. 3.4 depicts the logical flow of the proposed hybrid optimization algorithm comprising the GA, BW and PSO algorithms. The probability obtained when a HMM is used to generate a retrieved observation sequence from the validation data denoted by $P(\sigma|\lambda)$ determines the appropriateness of each solution.

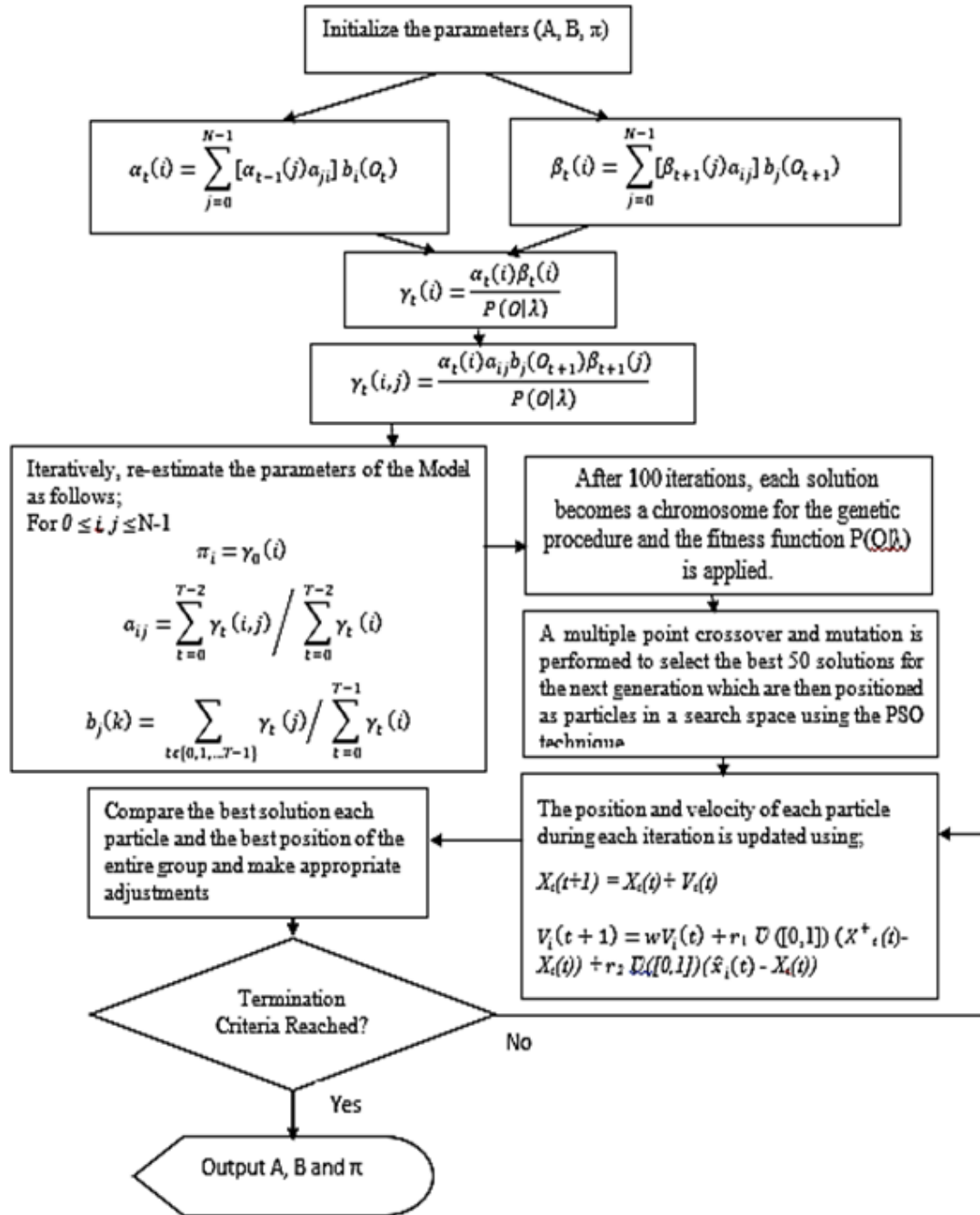


Figure 3.4: A flow of the Proposed Hybrid Optimization Algorithm



As depicted in Fig. 3.4, to ensure that the BW algorithm is not trapped in a local optimal solution due to its excessive number of iterations, possible solutions obtained after few iterations to obtain the values of A, B and π are considered chromosomes for the GA where $P(\sigma|\lambda)$ is used as the fitness function.

However, to avoid an exponential increase in the search space which usually leads to performance degradation when the number of possible solutions exposed to the genetic operators is large when GA is employed, the best solutions are positioned as particles in a search space using the PSO.

The proposed hybrid algorithm ensures that the PSO algorithm which requires minimal number of parameters and correspondingly lower number of iterations converges faster.

The various transaction types are considered the internal states of the proposed model. Transaction amounts are categorized as Low (l) = (0; 500], Medium (m) = (501; 1000], and High (h) = (1001; Transaction Limit] values. Also, the frequency at which these transaction occur are also categorized into a Low (Less than 5 times a month), Medium (Between 5 and 10 times a month), and High (at least 10 times a month) are also considered by the proposed model. For instance, if an account-holder performs about seven (7) transactions with the month with an average value of say 700, then the corresponding observation symbol is medium-frequency medium-amount (mm). The transaction amounts are combined with the frequency at which they occur in order to group customers according to their transaction profiles using the K-Means clustering algorithm. This results in the formulation of nine (9) observation symbols for the proposed model. The profiles considered are detailed in Table 3.1.

In order to demonstrate how the proposed hybrid algorithm will perform with varying mixes of transactions in terms of amounts and frequency, the combination (x, y, z) as in amount parameter represents an account-holder who performed x , y and z transactions in percentage terms in the high, Medium and low amounts categories respectively. Likewise, (x, y, z) as used in Frequency refers to a profile where x , y and z represents a high, Medium and low transaction frequencies respectively.



Table 3.1: Transaction mix representing the various transaction profiles of customers

Transaction Profile	Parameter	Transaction mix
High Amount, High Frequency	Amount	(60,30,10)
	Frequency	(70,25,5)
High Amount, Medium Frequency	Amount	(80,10,10)
	Frequency	(25,75,5)
High Amount, Low Frequency	Amount	(90,7,3)
	Frequency	(25,5,75)
Medium Amt, High Frequency	Amount	(30,60,10)
	Frequency	(80,15,5)
Medium Amount, Medium Frequency	Amount	(15,75,10)
	Frequency	(15,80,5)
Medium Amount, Low Frequency	Amount	(10,70,20)
	Frequency	(15,5,80)
Low Amount, Low Frequency	Amount	(3,2,95)
	Frequency	(30,20,50)
Low Amount, Medium Frequency	Amount	(5,10,85)
	Frequency	(30,50,20)
Low Amount, High Frequency	Amount	(10,10,80)
	Frequency	(50,30,20)



3.3 Detecting Real-Time Electronic Banking Fraud on highly imbalanced datasets using HMMs

3.3.1 Dataset

Publicly available datasets on electronic banking for research purposes are very limited mainly due to personal and security concerns. In this research therefore, a dataset which represents a simulated electronic financial transactions obtained from Kaggle is adopted. As detailed in Table 3.2, the dataset is highly imbalanced since only 8,312 transactions out of the almost six (6) million transactions are labeled fraud.

The imbalanced nature of the dataset is a typical characteristic of electronic financial transaction datasets as there are usually more non-fraud transactions than fraud transaction

Table 3.2: Details of the Paysim Dataset adopted for the Research

Transaction Type	Genuine Transactions	Fraudulent Transactions	Total
Transfer	528812	4097	532909
Cash-out	2233384	4116	2237500
Cash-in	1399284	0	1399284
Debit	41432	0	41432
Payments	2151494	0	2151494
Total	6354407	8213	6362620

Note: Synthetic Financial Datasets For Fraud Detection, by E. Lopez-Rojas, 2017 (<https://www.kaggle.com/ealaxi/paysim1>). Copyright 2021 by kaggle.

'CASH IN' and 'CASH OUT' represents an increase in account balance of a customer as a result of cash inflow and a decrease in account balance as a result of cash outflow



respectively. 'TRANSFER' refers to movement of money among users and 'PAYMENT' represents the settlements made for goods and services to merchants. 'DEBIT' denotes the movement of E-cash from an personal electronic wallet to a specified bank account.

3.3.2 Data Pre-Processing

To establish the effectiveness of the proposed Model on the highly class imbalanced dataset, the Synthetic Minority Oversampling Technique (SMOTE) which operates by selecting one or more of the k-nearest neighbors randomly for all fraudulent transaction as outlined in Algorithm 1 is adopted (Seeja & Zareapoor, 2014). The data is normally reconstructed afterwards before applying the proposed HMMs. Specifically, the sampling rate is set to 73000 percent based on the ratio between the genuine and fraudulent cases..

Algorithm 1 : SMOTE(f , R , k)

1. Set the minority class (Fraudulent Transactions) as set f
2. The Euclidean distance that exists between each point, $y \in f$, and any other point are computed and represented as the k-nearest neighbors of the point y .
3. Based on the class inbalanced proportion of the Dataset, R , which denotes the sampling rate is determined.
4. R samples are chosen randomly for each point $y \in f$ from its k-nearest neighbors to produce another group of points denoted as f_1
5. A new set of population is generated for each point $y_t \in f_1, (t = 1, 2, 3, \dots R)$, as in Eqn.(3.21):

$$y' = y + rand(0, 1) * |y - y_t| \quad (3.21)$$



3.3.3 Identifying Transaction Profile of Customers using a Modified DBSCAN Clustering Algorithm

Instead of randomly selecting the values of the Initial vector, Transition and Emission probability Matrices, we modified the Density Based Spatial Clustering of Applications with Noise (DBSCAN) as presented in Algorithm 2 to determine the transaction pattern of customers by considering the amount and frequency of transactions for optimal training of the the proposed HMMs. The DBSCAN clustering technique eliminates outliers unlike the K-means algorithm which is widely used in literature. In the modification of the existing DBSCAN algorithm, a step which computes the centroid of each cluster to enable us dynamically convert an incoming transaction into an observation symbol in the fraud detection process is incorporated.

Algorithm 2: DBSCAN (*dataset*, *d*, *minpts*)

Input: A set of points, *dataset*, distance threshold *d*, and *minpts* which refers to the minimum number of points suitable to constitute a cluster.

Output: Customers Transaction Profiles represented by a set of clusters.

1. $C = 1$, initialise the cluster index to 1
2. Each point pt , which is not visited in dataset is marked *visited* and its neighboring points are generated denoted as N of pt
3. If $|N| \geq minpts$ then $N = N \cup N'$
4. if pt' is not a member of any cluster, Mark it as noise and Compute the centroid \bar{h} of each cluster using Eqn.(3.22) where n_i represents the number of points in cluster c_i

$$\bar{h} = \frac{1}{n_i} \sum_{x_j \in c_i} x_j \quad (3.22)$$



After successfully executing the clustering procedure on the Dataset using the modified DBSCAN algorithm, the spending profiles of account-holders are appropriately determined.

Let ψ_i denote the total number of transactions of an account-holder, ϑ in percentage terms, it therefore implies that, the transaction profile ρ of that account holder is computed as in Eqn.(3.23)

$$\rho(\vartheta) = \arg \text{Max}_i(\psi_i) \quad (3.23)$$

At the end of the clustering step, the transaction profile of an account holder corresponds to the cluster which constitutes majority of the transactions. An observation symbol corresponding to an incoming transaction ϕ (denoted by ϕ_m) is generated using the computed centroids of these clusters as defined in Eqn.(3.24).

$$\phi_m = V_{\arg \min}_i |m - n_i| \quad (3.24)$$

For this research, the clusters, are denoted by $C = \{\text{low-freq low-amt, low-freq medium-amt, low-freq high-amt, medium-freq low-amt, medium-freq medium-amt, medium-freq high-amt, high-freq low- amt, high-freq medium-amt, high-freq high-amt}\}$ and the attributes, denoted by $R = \{\text{transaction-amount, frequency-of-transaction}\}$ are used to generate the clusters.

For an observed sequence, $\sigma = (\sigma_0, \sigma_1, \sigma_2, \dots, \sigma_{T-1})$, it's probability of occurrence with respect to the HMM λ , where $v = (v_0, v_1, v_2, \dots, v_{T-1})$ represents the various hidden states, the definition of the Emission transition Matrix is defined as in Eqn.(3.25).;

$$P(\sigma|v, \lambda) = b_{v_0}(\sigma_0)b_{v_1}(\sigma_1), \dots, b_{v_{T-1}}(\sigma_{T-1}) \quad (3.25)$$



The Initial transition vector, π and State Transition Matrix, A are also defined as in Eqn.(3.26) and Eqn.(3.27) below;

$$P(v|\lambda) = \pi_v a_{v,v_1}, \dots, a_{v_{T-2}, v_{T-1}} \quad (3.26)$$

$$P(\sigma, v|\lambda) = P(\sigma|v, \lambda)P(v|\lambda) \quad (3.27)$$

Eqn.(3.28) computes the sum of all state sequences that has some possibility of occurrence;

$$P(\sigma, v) = \sum_v P(\sigma, v|\lambda) \quad (3.28)$$

$$P(\sigma, v) = \sum_v P(\sigma|v, \lambda)P(v|\lambda) \quad (3.29)$$

$$P(\sigma|\lambda) = \sum_v \pi_{v_0} b_{v_0}(\sigma_0) a_{v_0, v_1} b_{v_1}(\sigma_1) \dots a_{v_{T-2}, v_{T-1}} b_{v_{T-1}}(\sigma_{T-1}) \quad (3.30)$$

The probability of the observation sequences denoted as $e_t(i)$, where the system is in state qi at time t is defined in Eqn.(3.31).

$$e_t(j) = P(\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_t, v_t = q_j|\lambda) \quad (3.31)$$

$e_t(j)$ is obtained as from Eqn.(3.32) to Eqn.(3.34):

1. For $i = 0, 1, 2, \dots, N-1$

$$e_0(j) = \pi_j b_j(\sigma_0) \quad (3.32)$$

2. For $i = 0, 1, 2, \dots, N-1$ and $t = 0, 1, 2, \dots, T-1$, we compute

$$e_t(j) = \sum_{i=0}^{G-1} [e_{t-1}(i) a_{ij}] b_j(\sigma_t) \quad (3.33)$$

3. Eqn.(3.34) is obtained from Eqn.(3.33)

$$P(\sigma|\lambda) = \sum_{j=0}^{G-1} e_{T-1}(j) \quad (3.34)$$



3.3.4 Scaling the proposed HMMs

During the optimization process, a number of calculations consist of multiplications of probabilities where our forward variable, $e_t(i)$ gradually converges to zero exponentially as time increases. It is necessary therefore to scale the values rather than implement the formulas directly as given above which has a high tendency to result in an underflow. At the same time, ensuring the validity of the formulae that performs the re-estimation is crucial. The basic recurrence in the calculation of $e_t(i)$ is illustrated as in Eqn.(3.35).

$$e_t(j) = \sum_{i=0}^{G-1} [e_{t-1}(j)a_{ij}]b_j(\sigma_t) \quad (3.35)$$

The term $e_t(j)$ can be normalized by dividing by the summation of $e_t(i)$ whilst ensuring that the re-estimation formulae remains valid. We assume that, $\tilde{e}_0(j) = e_0(j)$ for $j = 0, 1, 2, \dots, G - 1$ at time, $t = 0$. Also for $j = 1, 2, 3, \dots, G - 1$, we take $r_0 = \sum_{i=0}^{G-1} \tilde{e}_0(i)$ and $\hat{e}_0(j) = r_0 \tilde{e}_0(j)$. It therefore implies that, the following steps can be executed to scale the proposed models for $t = 1, 2, 3, \dots, T - 1$.

1. For $j = 1, 2, 3, \dots, G - 1$, $\tilde{e}_t(j)$ is computed as in Eqn. (3.36)

$$\tilde{e}_t(j) = \sum_{j=0}^{G-1} \hat{e}_{t-1}(i)a_{ij}b_j(\sigma_t) \quad (3.36)$$

2. let

$$R_t = \frac{1}{\sum_{i=0}^{G-1} \hat{e}_t(i)} \quad (3.37)$$

3. For $j = 1, 2, 3, \dots, G - 1$ compute

$$\hat{e}_t(j) = R_t \tilde{e}_t(j) \quad (3.38)$$



Apparently $\hat{e}_0(j) = R_0 e_0(j)$ Suppose that

$$\hat{e}_t(j) = R_0 C_1, \dots, R_t e_t(j) \quad (3.39)$$

Then

$$\hat{e}_{t+1}(j) = R_{t+1} \tilde{e}_{t+1}(j) \quad (3.40)$$

$$= R_{t+1} \sum_{i=0}^{G-1} \hat{e}_t(i) a_{ij} b_j(\sigma_{t+1}) \quad (3.41)$$

$$= R_0 R_1, \dots, R_t R_{t+1} \sum_{i=0}^{G-1} e_t(i) a_{ij} b_j(\sigma_{t+1}) \quad (3.42)$$

$$= R_0 R_1, \dots, R_t R_{t+1} e_{t+1}(j) \quad (3.43)$$

By induction, its clear that for all values of t, Eqn (3.43) holds. From the definitions of \tilde{e} and \hat{e} and also from Eqn. (3.43), it implies that;

$$\hat{e} = \frac{e_t(j)}{\sum_{i=0}^{G-1} e_t(i)} \quad (3.44)$$

It is observed that, for all possible values of time, t, the values obtained by $\hat{e}_t(j)$ are the preferred scaled values of $e_t(j)$. As a consequence of Eqn.(3.44),

$$\sum_{i=0}^{G-1} \hat{e}_{T-1}(i) = 1 \quad (3.45)$$

Also, from Eqn.(3.43) we have

$$\sum_{i=0}^{G-1} \hat{e}_{T-1}(i) = R_0 R_1, \dots, R_{T-1} \sum_{i=0}^{G-1} e_{T-1}(i) \quad (3.46)$$

$$= R_0 R_1, \dots, R_{T-1} P(\sigma|\lambda) \quad (3.47)$$

Combining these results gives

$$P(\sigma|\lambda) = \frac{1}{\prod_{j=0}^{T-1} R_j} \quad (3.48)$$



To avoid underflow, we instead compute

$$\text{Log}[P(\sigma|\lambda)] = - \sum_{i=0}^{T-1} \log R_i \quad (3.49)$$

In a similar manner, as with $\beta_t(j)$ the same scale factor denoted as R_t is employed for $e_t(j)$ and therefore yielding $\hat{\beta}_t(j) = R_t\beta_t(j)$. The variables $\gamma_t(j, i)$ and $\gamma_t(j)$ are obtained by replacing $e_t(j)$ and $\beta_t(j)$ with $\hat{e}_t(j)$ and $\hat{\beta}_t(j)$ respectively. The re-estimation of the initial probability vector, π , State transition Matrix, A and Emission Transition Matrix denoted as B .

3.3.5 Training the proposed HMMs

The various transaction types are considered the hidden states whilst the transaction amounts combined with the frequency at which they occur denoted as low-freq low-amt(ll), low-freq medium-amt(lm), low-freq high-amt(lh), medium-freq low-amt(ml), medium-freq medium-amt(mm), medium-freq high-amt(mh), high-freq low- amt(hl), high-freq medium-amt(hm), high-freq high-amt(hh) represents the observation symbols of the proposed HMM.

After formulating the hidden states and observation symbols, the hybrid optimization algorithm as presented in Algorithm 3 comprising the BW, PSO and GA is used to effectively train the proposed models.



Algorithm 3: A hybrid algorithm for training the proposed HMMs

1. For each HMM, λ corresponding to each account holder, initialize the parameters A, B, π using the spending profile of that customer.
 2. The forward variable, $\alpha_t(j)$ is computed as in Eqn (3.4)
 3. The backward variable, $\beta_t(j)$ is computed as in Eqn (3.8)
 4. Compute the gamma variable as in Eqn(3.10)
 5. Compute the Di-gammas as in Eqn(3.12)
 6. For $0 \leq i, j \leq G - 1$, The Initial Probabilities are obtained as in Eqn(3.14)
 7. For $0 \leq i, j \leq G - 1$, The transition probability matrix is computed as in Eqn(3.15);
 8. For $i = 0, 1, \dots, G - 1$, and $d = 0, 1, \dots, H - 1$, Emmission prob Matrix is computed as in Eqn(3.16).
 9. After 100 iterations, each solution becomes a chromosome for the genetic procedure and the fitness function $P(O|\lambda)$ is applied and then positioned as particles in a search space using the PSO technique.
 10. Each particle's updated position and velocity is obtained using Eqn(3.17) and Eqn(3.18) respectively in each iteration;
 11. The entire swarms best position is compared with that obtained by each particle appropriate adjustments made.
 12. If the result remains same for several iterations, go to 13, otherwise go to 10
 13. Output A, B and π
-



3.3.6 Fraud Detection

To effectively classify an incoming transaction as fraudulent or otherwise, sequence of observation symbols, say $\sigma = \sigma_1, \sigma_2, \dots, \sigma_r$ are extracted from the validation data of an account holder and its probability of acceptance, δ_1 is computed by the model as in Eqn(3.34) by employing Eqn(3.50)

$$\delta_1 = p(\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_t | \lambda) \quad (3.50)$$

An incoming transaction occurring at time, t is converted to an observation symbol denoted as σ_{t+1} using (33) is used to replace the first observation symbol, δ_1 and its probability of acceptance by the model denoted as δ_2 is also computed as in Eqn(3.51).

$$\delta_2 = p(\sigma_2, \sigma_3, \sigma_4, \dots, \sigma_{t+1} | \lambda) \quad (3.51)$$

The newly generated transaction is classified as fraud and if the difference between δ_1 and δ_2 denoted as $\Delta\delta$ is above a predefined threshold φ as in Eqn(3.52).

$$\frac{\Delta\delta}{\delta_1} \geq \varphi \quad (3.52)$$

A genuine transaction is added to the sequence permanently to contribute to determining the validity or otherwise of the next transaction since transaction behavior of an account-holder could be dynamic. Otherwise, the transaction is declined, and the symbol is discarded. The steps involved in detecting fraud using the proposed system are outlined in Figure 3.5.



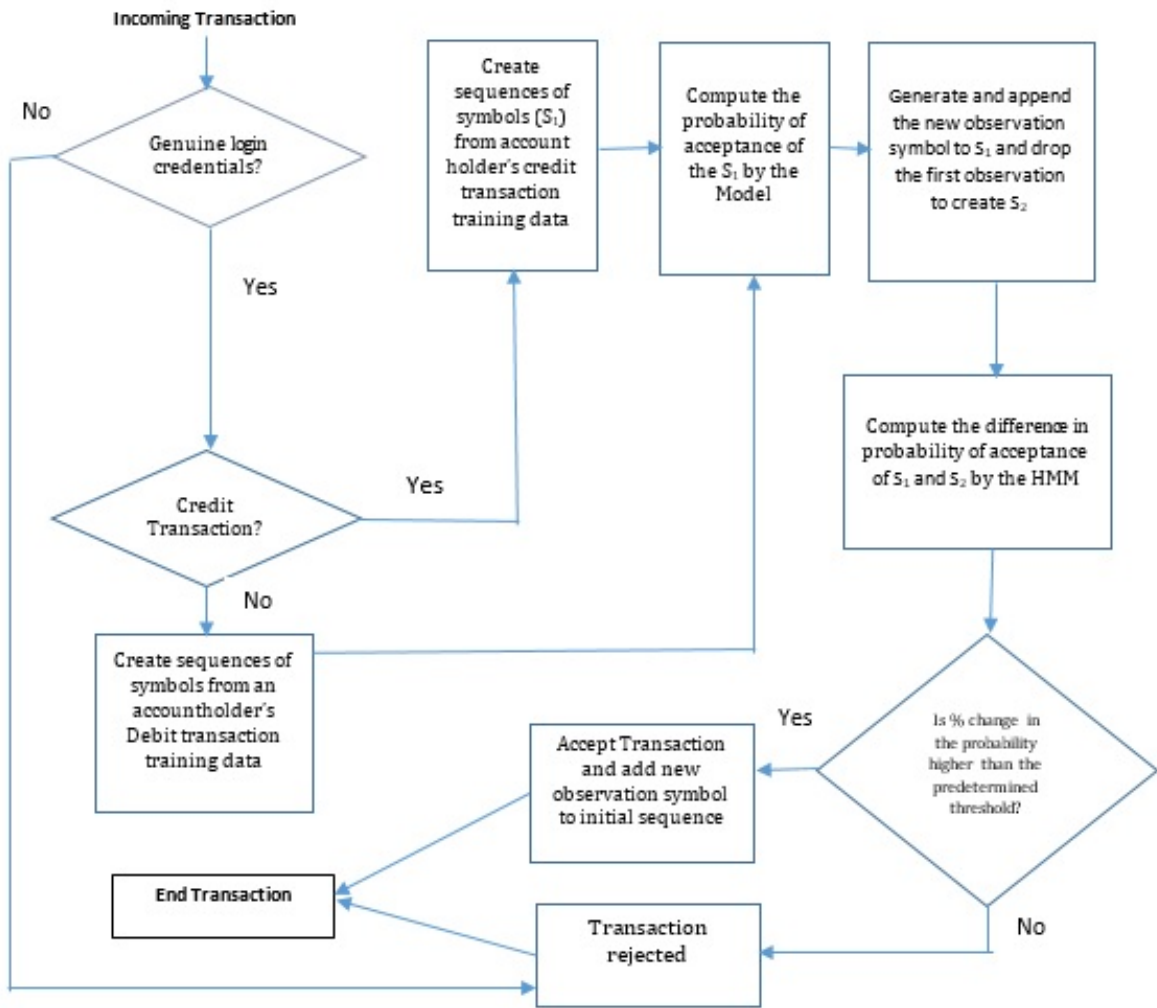


Figure 3.5: Flowchart of the detection phase of the proposed system



3.3.7 Evaluation Metrics

Precision(P), recall(R) and F1-scores(F_1) as presented from Eqn(3.53) to Eqn(3.55) respectively are employed as evaluation metrics due to the highly class imbalanced nature of the dataset used instead of relying solely on only accuracy (Wedge et al., 2019).

$$p = \frac{T_p}{T_p + F_p} \quad (3.53)$$

$$R = \frac{T_p}{T_p + F_n} \quad (3.54)$$

$$F_1 = 2 * \frac{P * R}{P + R} \quad (3.55)$$

The number of True and False positives as well as the number of False negatives are denoted by T_p, F_p and F_n respectively. The total number of accurate predictions relating to the positive cases out of the total number of positive predictions made is known as precision whilst out of the total number of positive cases, Recall refers to the number of accurate positive predictions. However, the F_1 -Measure combines the effects of both recall and precision (Sulaiman & AbdelKarim, 2019).

The research also seeks to establish the convergence rate of the model proposed by the research by plotting how it learns with respect to time, t often known as the learning curve. According to Bhattacharya (2014), for algorithms that learns gradually from training datasets, employing learning curves are ideal where the main target is to obtain a good fit that exists between an underfit and overfit model.

3.4 Reducing False Positives in Real-Time Fraud Detection Using Improved Multi-Layer HMMs

3.4.1 Data Pre-processing

The PAYSIM dataset as presented in Table 3.2 was adopted. The Synthetic Minority Oversampling Technique (SMOTE) and DBSCAN clustering algorithm as presented in Algorithm 1 and Algorithm 2 respectively were also employed to handle the highly class imbalance nature of the dataset and to determine the transaction profiles of customers respectively.

3.4.2 Structure of the proposed Improved Multi-layer HMM

The proposed Improved Multi-layer HMM is presented as in Figure 3.6 where there is a decomposition of the problem into multiple separate layers of HMMs. The various transaction types are considered the internal states whilst the Transaction Time Gaps, Daily spending amounts as well as transaction amounts and the frequency at which they occur represents the observation symbols of each of the HMM in each layer.



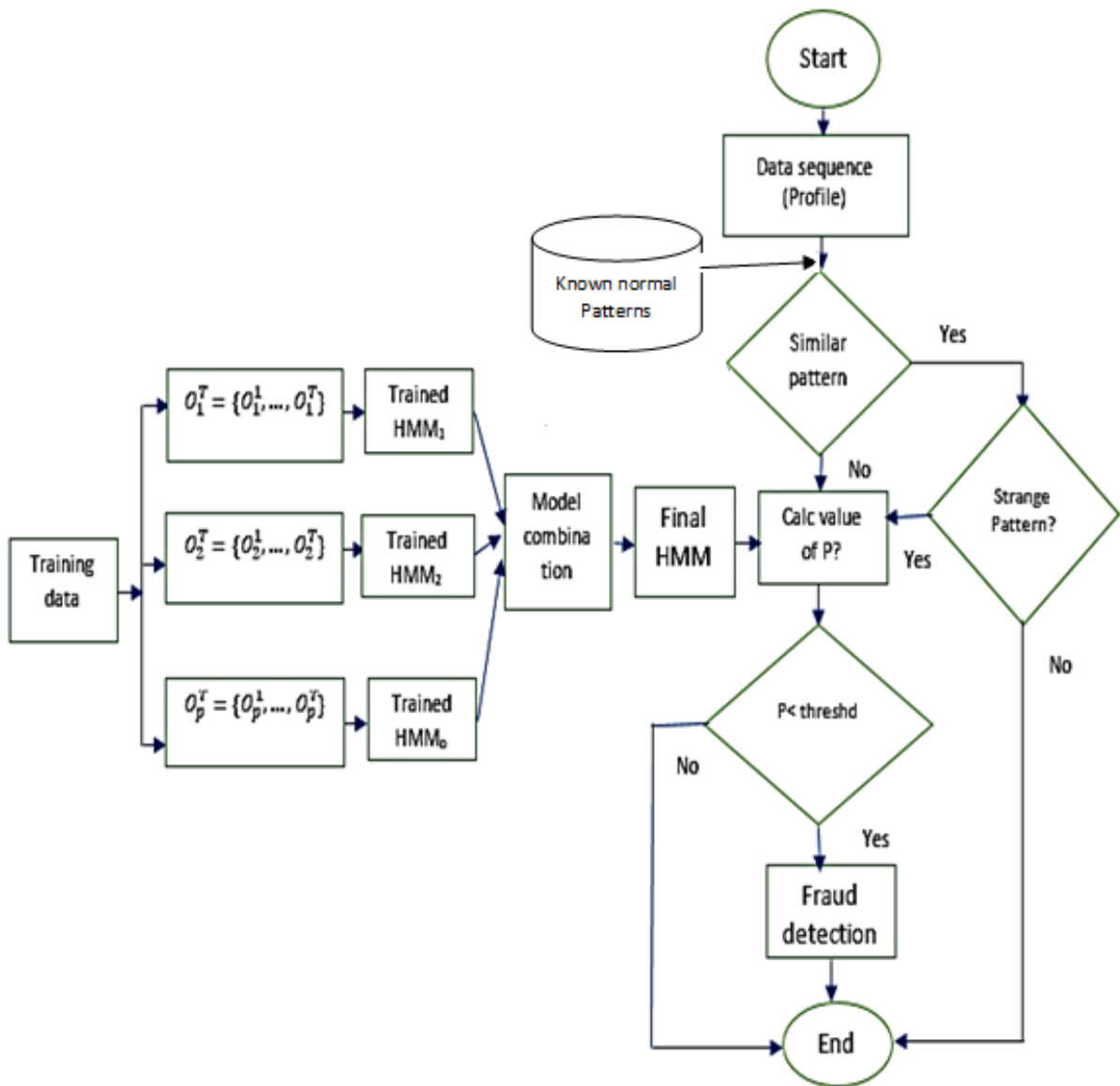


Figure 3.6: A flow of the proposed Improved Multi-layer HMM



The structure of the proposed improved multi-layer HMM as outlined in Figure 3.6 is detailed below;

1. The various transaction types are considered the internal states of each HMM
Transaction Time Gaps, Daily spending amounts as well as transaction amounts and the frequency of transactions represents the observation symbols of each of the HMM in each layer.
2. Multiple HMMs, $\lambda_1, \lambda_2, \dots, \lambda_p$ with their corresponding number of hidden states q_1, q_2, \dots, q_p and emission symbols are constructed.
3. At time $t = T$, Each of the HMMs is trained using a subsequence of the training data with; The observation sequence and the probable sequence of states is given as in Eqn(3.56) and Eqn(3.57) respectively:

$$O_1^T = O_1^1, O_1^2, \dots, O_1^T, O_2^T = O_2^1, O_2^2, \dots, O_2^T, \dots, O_p^T = O_p^1, O_p^2, \dots, O_p^T \quad (3.56)$$

$$Q_1^T = q_1^1, q_1^2, \dots, q_1^T, q_2^1, q_2^2, \dots, q_2^T, \dots, q_p^1, q_p^2, \dots, q_p^T \quad (3.57)$$

4. Each sub-sequence is used to train a specific HMM which are merged to create the final model using a linear recombination procedure.
5. A comparison between available normal sequences in a database is first and foremost compared to any retrieved observation sequence, $\sigma = \sigma_1, \sigma_2, \dots, \sigma_r$ confirm whether there exist any possible matches.
6. If no significant matches are detected, the proposed HMM is then employed to compute the probability of occurrence of that particular observation sequence denoted as δ as in Eqn(3.58)). A final decision regarding the validity or otherwise of the new sequence is facilitated by comparing the value of δ with a threshold.

$$\delta = p(\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_r | \lambda) \quad (3.58)$$



3.4.3 Training the Proposed Improved Multilayer-HMM

Since the parameters of each HMM are optimized using various sub-sections of the data which are finally combined to create the final model using a linear recombination procedure, the hybrid algorithm comprising the BW, GA and PSO algorithms as outlined in Algorithm 3 is modified as in Algorithm 4.

Algorithm 4: A hybrid algorithm for training the proposed HMMs

1. For each HMM, initialize the parameters (A, B, π) using the spending profile of the customer

2. Compute the forward variable as in Eqn(3.59)

$$\alpha_t(j) = \sum_{i=0}^{G-1} [\alpha_{t-1}(i)a_{ij}]b_j(O_t) \quad (3.59)$$

3. Compute the backward variable as in Eqn(3.60)

$$\beta_t(j) = P(\sigma_{t+1}, \sigma_{t+2}, \sigma_{t+3}, \dots, \sigma_{T-1} | q_t = X_j, \lambda) \quad (3.60)$$

4. Compute the gamma variable as in Eqn(3.61)

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{P(O|\lambda)} \quad (3.61)$$

5. Compute the Di-gammas as in Eqn(3.62)

$$\gamma_t(j, i) = P(q_t = x_j, q_{t+1} = x_i | \sigma, \lambda) \quad (3.62)$$

6. For $0 \leq i, j \leq G - 1$, Compute the Initial Probabilities as in Eqn(3.63)

$$\pi_j = \gamma_0(j) \quad (3.63)$$

7. For $0 \leq i, j \leq G - 1$, The transition probability matrix is computed as in Eqn(3.64);

$$a_{ji} = \sum_{t=0}^{T-2} \gamma_t(j, i) / \sum_{t=0}^{T-2} \gamma_t(j) \quad (3.64)$$

8. For $i = 0, 1, \dots, G - 1$, and $d = 0, 1, \dots, H - 1$, Emmission prob Matrix is computed as in Eqn(3.65):

$$b_i(k) = \sum_{t \in (0,1,2,\dots,T-1), \sigma_t=d} \gamma_t(i) / \sum_{t=0}^{T-1} \gamma_t(i) \quad (3.65)$$

9. After 100 iterations, each solution becomes a chromosome for the genetic procedure and the fitness function $P(O|\lambda)$ is applied and then positioned as particles in a



search space using the PSO technique.

10. The position and velocity of each particle during each iteration is updated using Eqn(3.66) and Eqn.(3.67);

$$X_j(t + 1) = X_j(t) + V_j(t) \quad (3.66)$$

$$V_j(t + 1) = wV_j(t) + r_1v([0, 1])(X_j^+(t) - x_j(t)) \\ + r_2v([0, 1])(\hat{x}_j(t) - x_j(t)) \quad (3.67)$$

11. The necessary adjustments are effected after comparing the best position obtained by the entire swarm as against the best position arived at by each particle.
12. Termination Criteria Reached? If yes go to 15, otherwise go to 12
13. Next HMM
14. For each given pair of HMMs, for each row $R1 = (S_1, S_2, \dots, S_N)$ for λ_1 and $R_2 = (T_1, T_2, \dots, T_N)$ for λ_2 in A, B and π , The resulting row, $R3$ is computed as in Eqn(3.68)

$$R_3 = \alpha R1 + (1 - \alpha)R_2 = (\alpha S_1 + (1 - \alpha)T_1, \\ \alpha S_2 + (1 - \alpha)T_2, \dots, \alpha S_N + (1 - \alpha)T_N) \quad (3.68)$$

15. Output A, B and π

As indicated in Eqn(3.68), The two models λ_a and λ_b are combined in such a way that the values in each row in the initial probability vector, π , as well as the Transition and Emission probability Matrices, A and B respectively must sum to one. The variable α denoted by α_x is set to 0.5 for all recombination.



Precision(p), Recall(R)/True Positive Rate (TPR), F1-Score (F1), as contained in Eqns. (3.69) to (3.72) below respectively as well as a Confusion Matrix and Receiver Operating Characteristic (ROC) graphs were used as our evaluation metrics owing to the highly imbalanced nature of the Dataset employed (Wedge et al., 2019). The number of True Positives, False Positives and False Negatives are denoted by T_p , F_p and F_n respectively.

$$p = \frac{T_p}{T_p + F_p} \quad (3.69)$$

$$R = \frac{T_p}{T_p + F_n} \quad (3.70)$$

$$F1 = 2 * \frac{P * R}{P + R} \quad (3.71)$$

$$FPR = F_p / (F_p + T_n) \quad (3.72)$$

3.5 Chapter Summary

The chapter presents the methodology adopted in developing an enhanced Hybrid algorithm for optimizing the parameters of HMMs inspired by the BW, GA and PSO algorithms. It also outlines the approach adopted to effectively detect and prevent real-time fraud in electronic banking on highly imbalanced datasets using HMMs. A modified DBSCAN algorithm and the SMOTE technique is used to effectively determine the transaction profiles of users and to handle the class imbalanced nature of our dataset respectively for optimal performance. The chapter concludes by detailing how the improved Multi-layer HMM proposed by this study is constructed to drastically reduce false positives in real-time fraud detection with improved training and detection times.

In the next chapter, detailed simulation results and discussion of the proposed algorithms and Models are presented.



CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This Chapter presents a simulation results and detailed analysis of the proposed algorithms and models as outlined in the methodology in order to achieve each of our stated research objectives. The first section discusses the simulation results on the implementation of the Hybrid algorithm for optimizing the parameters of HMMs inspired by the BW, GA and PSO algorithms.

The next section focuses on the effective detection and prevention of electronic banking fraud on a highly imbalanced dataset using HMMs. The chapter concludes by discussing the results obtained in implementing the improved Multi-layer HMMs as proposed by the research aimed at drastically reducing false positives in real-time fraud detection with improved training and detection times.

4.2 Implementing the improved Hybrid Algorithm for optimizing the parameters of HMMs

To establish the efficiency of the proposed hybrid optimization algorithm, the Dataset obtained through simulation by using the Transaction profiles as in Table 3.1 is employed. Eighty percent of the dataset was used for training and the rest held back for validation.

For different number of hidden states, simulations were performed in Python programming language by implementing the BW, GA, PSO, A hybrid of BW and GA (BWGA), A hybrid of BW and PSO (BWPSO), a hybrid of GA and PSO (GAPSO) and then the hybrid Model proposed by this study (BWGAPSO) and their performances compared. Series of observations, O from the validation data are constructed and the algorithms



Table 4.1: Average values of $P(O/\lambda)$ for a low-amount, Low-frequency transaction profile for all values of N

Algorithm	Number of States				Average
	2	3	4	5	
BW	0.689	0.731	0.723	0.521	0.666
GA	0.685	0.712	0.785	0.742	0.731
PSO	0.754	0.751	0.684	0.631	0.705
BWPSO	0.775	0.785	0.802	0.769	0.783
BWGA	0.762	0.795	0.811	0.768	0.784
GAPSO	0.785	0.892	0.821	0.803	0.825
Proposed	0.824	0.932	0.852	0.832	0.860

used to evaluate the probability, $P(O/\lambda)$ of generating O with the model.

For each number of hidden state, N , A hundred simulation runs were performed with each of the algorithms and the average value of $P(O/\lambda)$ computed using the same set of hidden states and emission symbols as outlined earlier in our methodology. For a low-amount, low-frequency transaction profile, the values of $P(O/\lambda)$ and their averages for the different values of N are shown in Fig. 4.1 and Table 4.1 respectively.

Figure 4.1 reveals that, for a low-amount, low-frequency transaction profile customer, the BW algorithm produced the worst results whilst the hybrid algorithm proposed by this study gave the best values and for all the different number of hidden states, N . The hybrid algorithms performed relatively better as compared to the non-hybrid ones. Apart from the algorithm proposed by this study, the hybrid of GA and PSO algorithms performed relatively better as compared to the others.

In terms of the average values of $P(O/\lambda)$ as contained in Table 4.1 , all the algorithms obtained values less than 80 percent, except the hybrid of GA and PSO and also by the algorithm proposed by this study. Largely, the algorithms performed better when the number of Hidden States is set to 3 and 4. The $P(O/\lambda)$ values obtained for all the nine (9) observation sequences corresponding to the different transaction profiles are displayed in Figure 4.2.



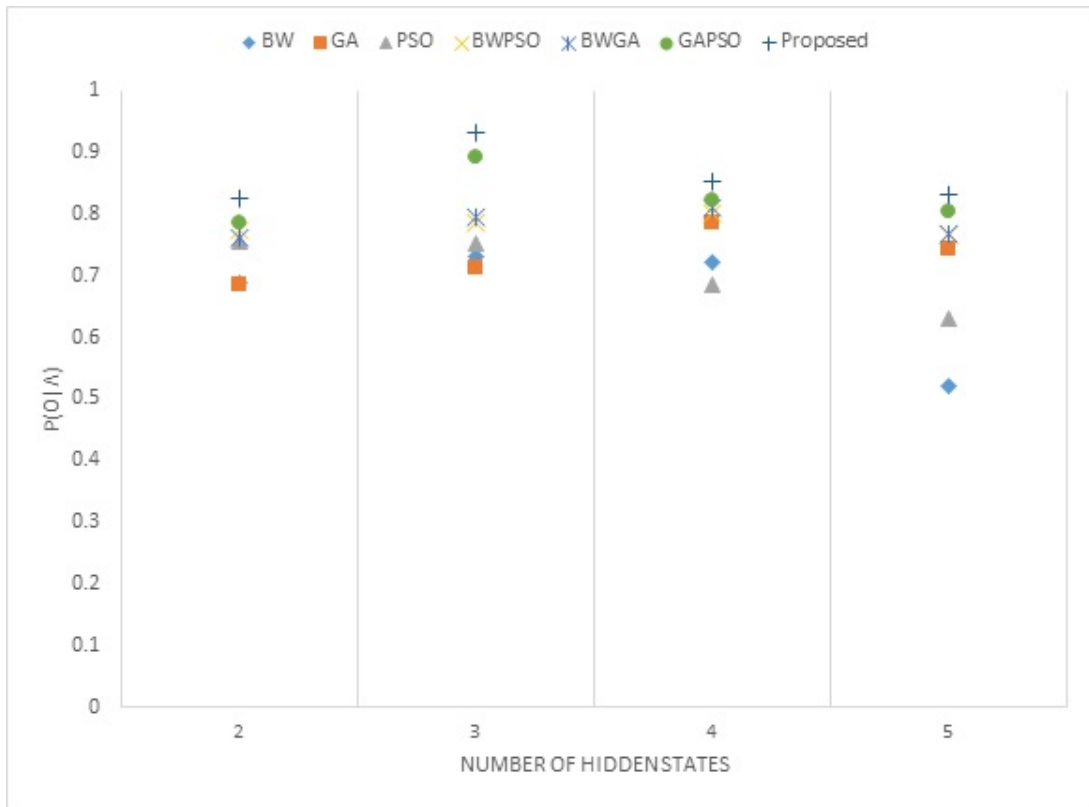


Figure 4.1: Values of $P(O|\lambda)$ for a Low-Amount, Low-Frequency transaction profile for all values of N

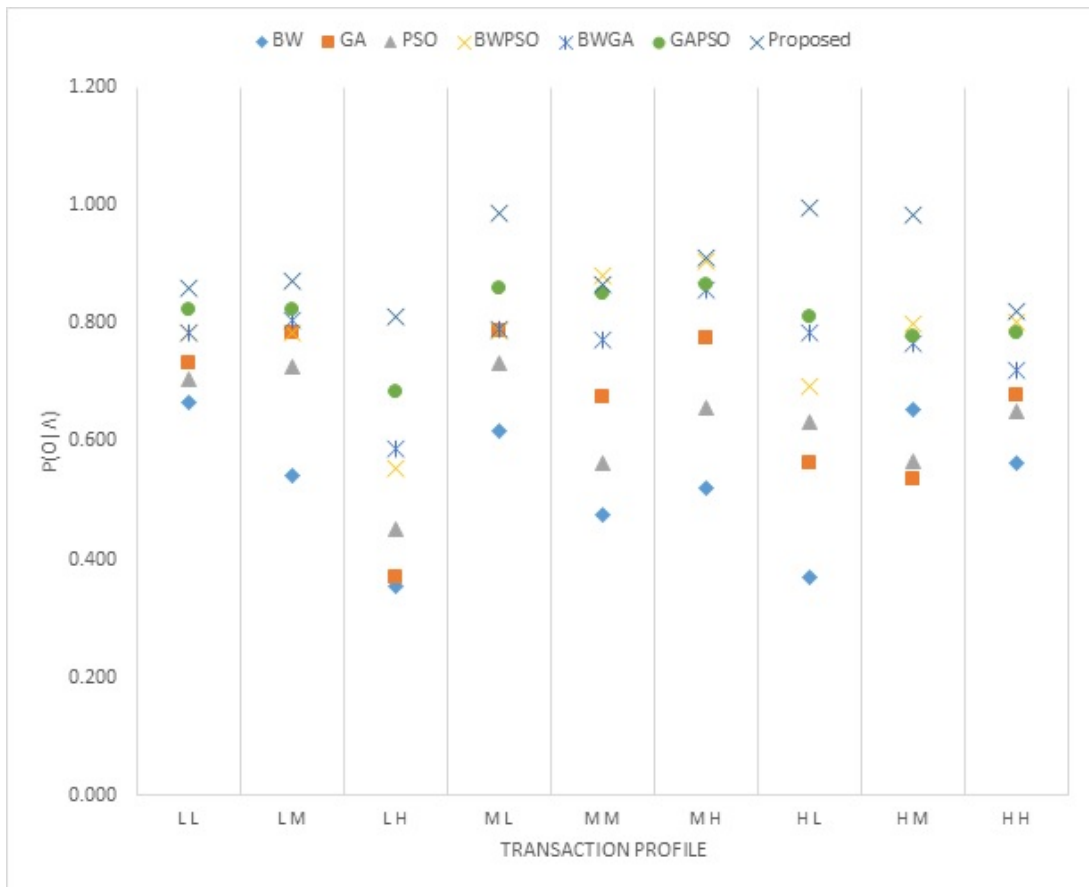


Figure 4.2: Values of $P(O|\lambda)$ for the various transaction profiles for various hidden number of states

It is evident from Figure 4.2 that, for all transaction profiles, the algorithm proposed

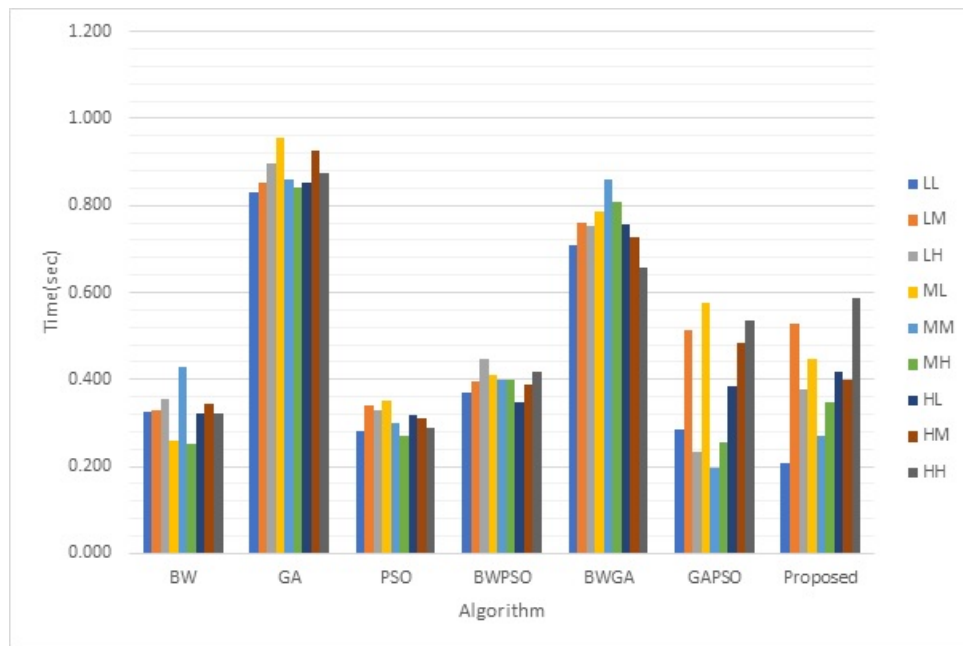


Figure 4.3: Time taken to optimize the HMMs by the various algorithm for all Transaction profiles

by this study consistently produced higher values of $P(O/\lambda)$ for all number of hidden states, N . Hybrid algorithms involving the PSO perform relatively better than those that utilized GA. The efficiency of the algorithm proposed by this study in terms of the time taken to optimize the HMMs for all the transaction profiles is shown in Figure 4.3. On the average, the BW and PSO algorithms obtained relatively better training times although they recorded relatively lower probability values as compared to the approach as proposed by this model as evident in Fig. 4.3. However, among all the hybrid approaches, the model proposed by this study performed better in terms of the execution time for all the possible number of hidden states.

4.3 Detecting Real-Time Electronic Banking Fraud on highly imbalanced dataset using HMMs

For various number of hidden states, four (4) sets of simulations were performed in two (2) stages in Python programming environment and their performance compared. For all the four sets of experiments, the proposed HMMs was executed in the second stage.

In the first stages of the first and second set of experiments, the K-means and the modified DBSCAN clustering algorithms were executed respectively to determine the transaction pattern of users. However, in the first stage of the third set of experiment,



both SMOTE and K-means techniques were employed to handle the class imbalanced problem of the Datasets and to determine the transaction pattern of users respectively whereas SMOTE and the modified DBSCAN clustering techniques are executed during the last set of experiments to handle the class imbalanced problem of the Datasets and to determine the transaction pattern of users respectively. Eighty (80) percent of dataset was used for training and the rest held back for evaluation and validation.

4.3.1 Precision Comparison

The precision of the four (4) approaches for various number of hidden states are presented in Fig. 4.4. The approach proposed by this study obtained the highest precision for the various hidden states. Approaches that employed the modified DBSCAN clustering technique performed better than those that utilized K-Means to determine the transaction patterns of users. It is also worth noting that, relatively higher values of precision were obtained when the SMOTE technique is adopted to handle the class imbalanced problem of the dataset.

4.3.2 Recall Rates Comparison

A comparison of the Recall rates of the four (4) different approaches for various number of hidden states is presented in Fig. 4.5. It is evident that, comparatively, utilizing the SMOTE technique resulted in higher recall rates. Similarly approaches that employed the modified DBSCAN clustering technique to determine the transaction patterns of users performed better as compared to the K-means. It can also be observed that, for larger number of Hidden States, all approaches obtained higher recall rates except for where K-means is employed without handling the data imbalance classification problem.

4.3.3 F1-Measure Comparison

In Fig. 4.6, the F1-score for the four approaches are presented for the various Hidden states. It is observed that, higher F1-scores are obtained when the modified DBSCAN



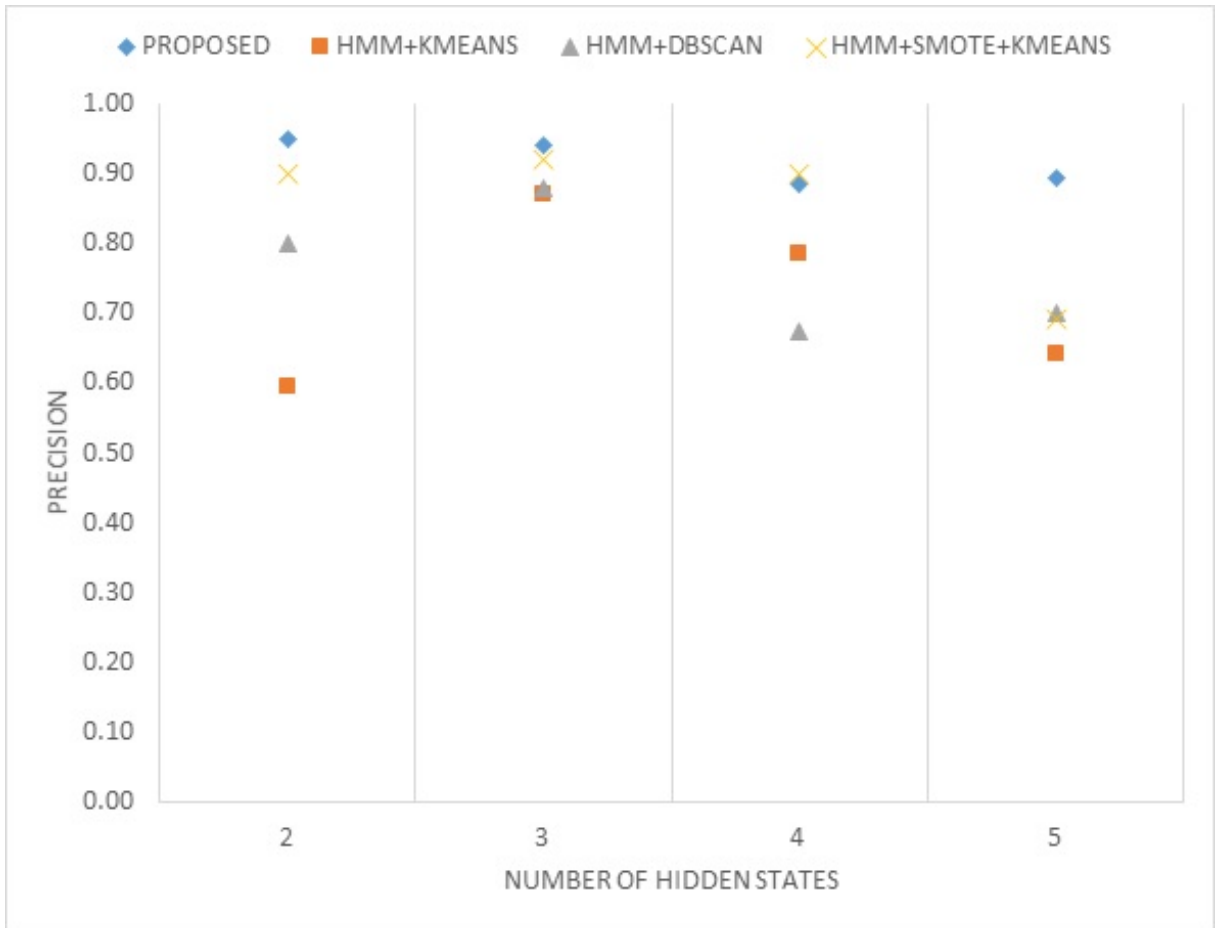


Figure 4.4: Comparison of the precision of the four (4) different approaches for various number of hidden states

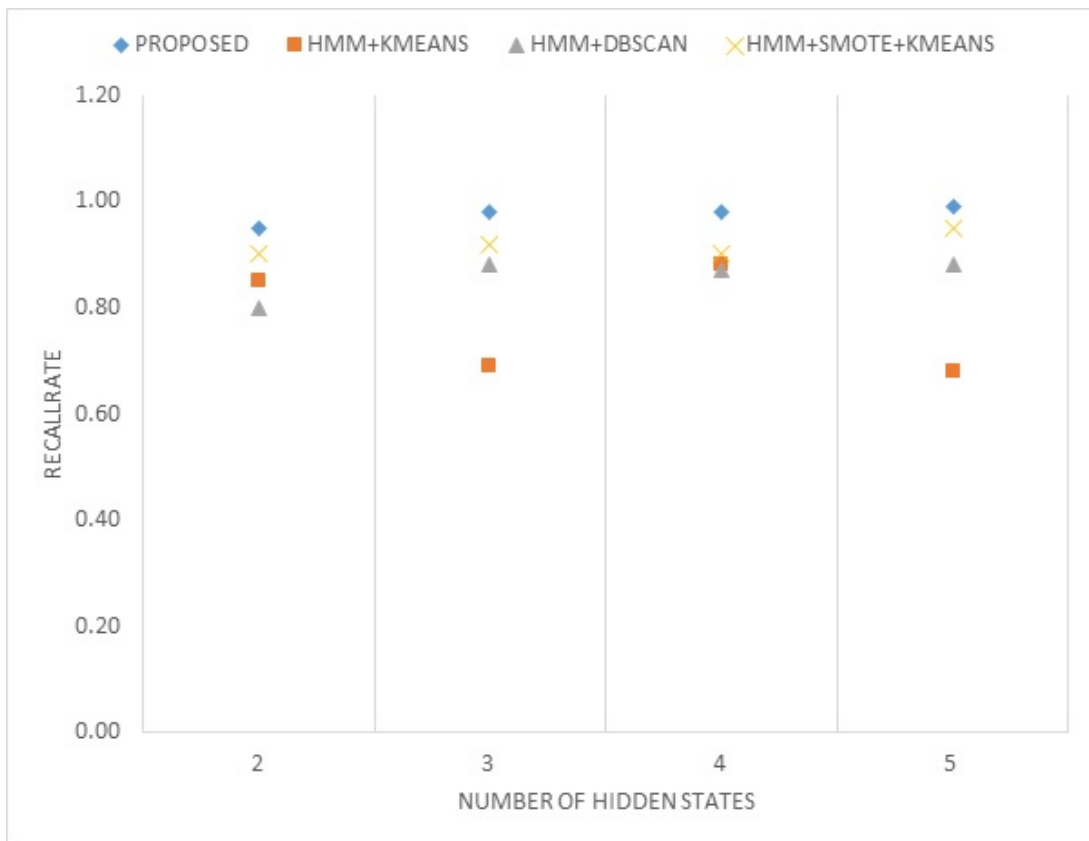


Figure 4.5: Recall rates of the four (4) approaches for different number of Hidden states

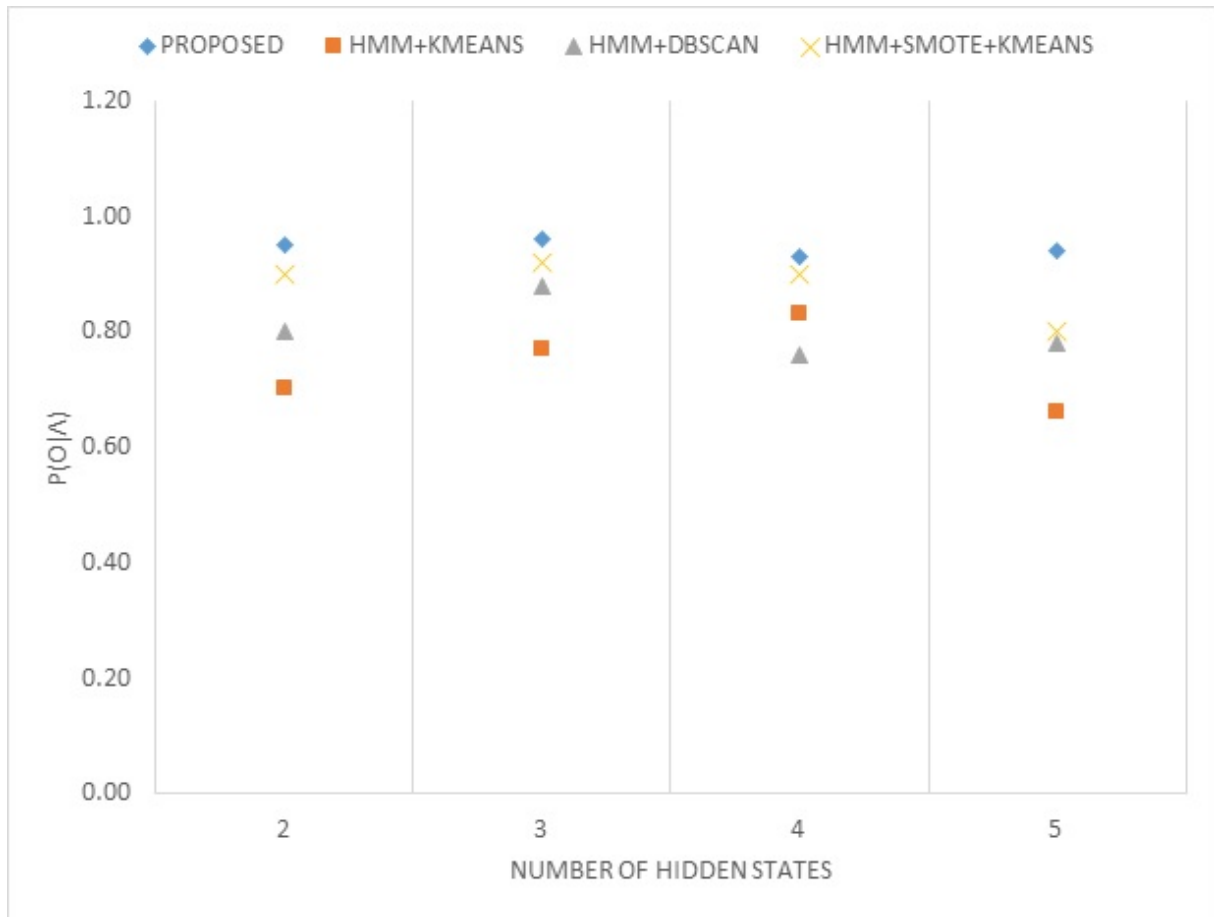


Figure 4.6: F1 Sores of the four (4) approaches for different number of Hidden states

clustering technique is employed as compared to using the K-means to determine the transaction patterns of users. Again, adopting the SMOTE technique performed' resulted in higher F1-Scores. Employing both SMOTE and the modified DBSCAN clustering algorithms to handle the class imbalanced problem and determine the transaction patterns of users respectively has proven to yield better F1-Scores. However all approaches performed relatively well when the number of hidden states is 3.

4.3.4 Learning and AUC-ROC Curves

The learning curve which is a plot of the learning performance of the various models over time is presented in Figure 4.7. Whilst an over-fitted model will usually display a wider variation between the training and cross validation scores, a significant low score signifies an under-fitting. It is evident that, the best score with regards the cross-validation and training datasets is obtained by model proposed by this study. The Receiver Operating Characteristic (ROC) graph which depicts the performance of a Model with respect to

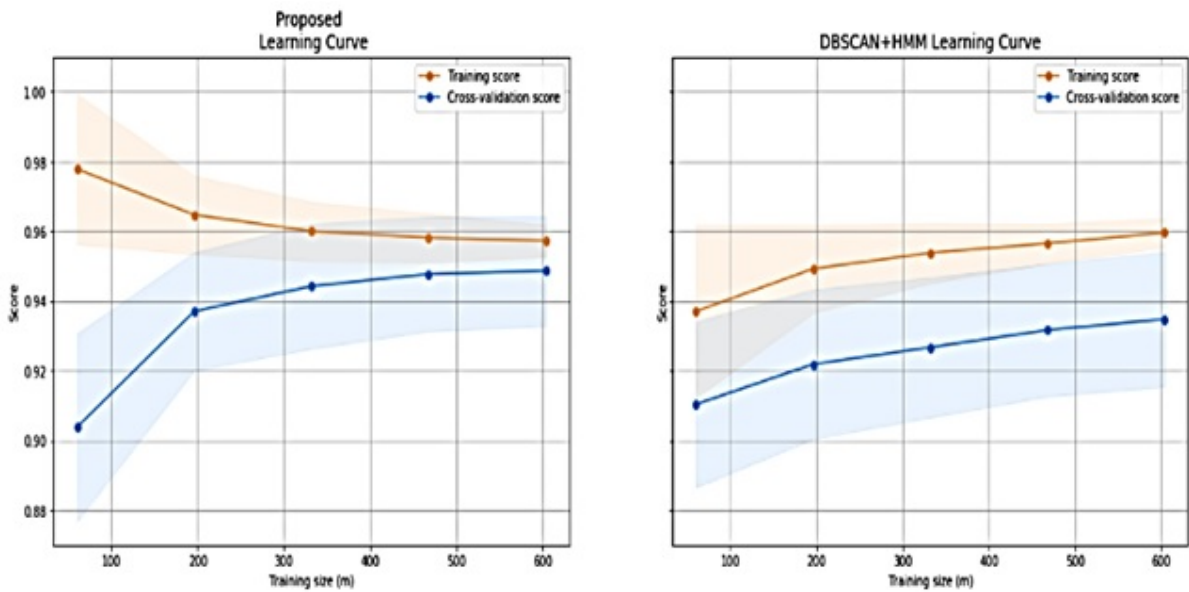


Figure 4.7: Learning curve for approaches where the Modified DBSCAN is adopted

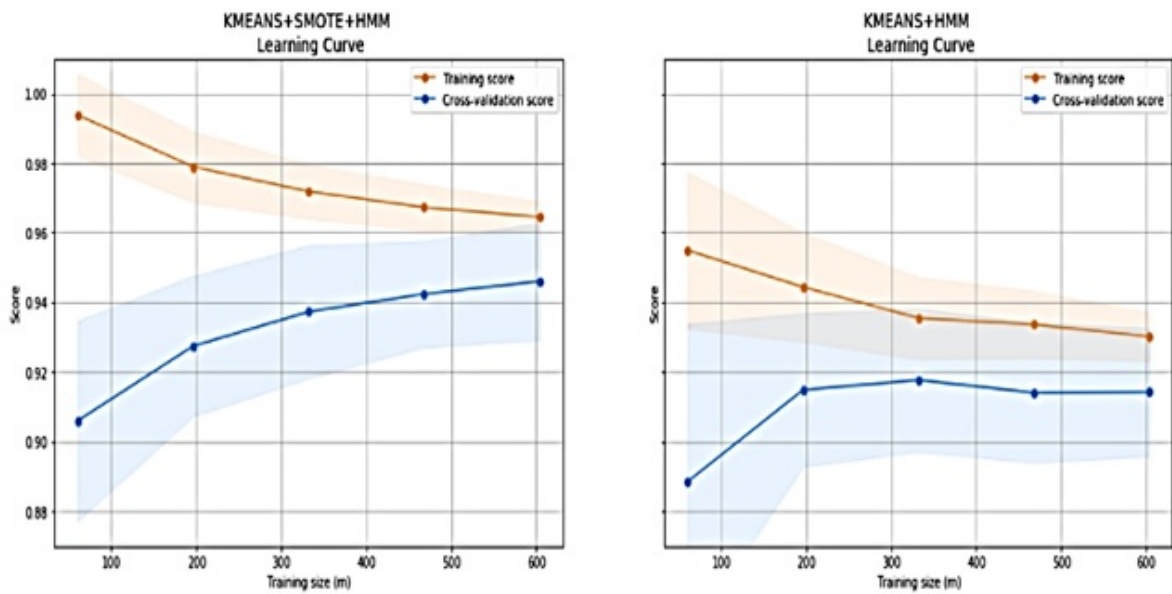


Figure 4.8: Learning curve for approaches where K-Means is adopted

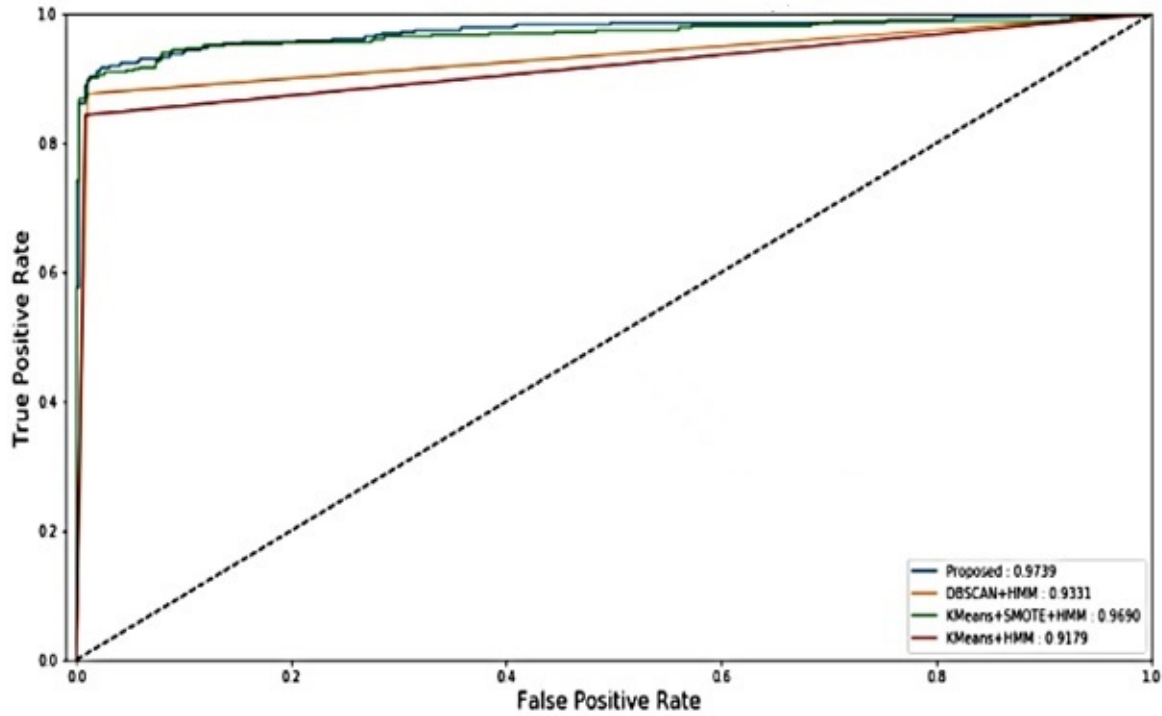


Figure 4.9: AUC-ROC Curve obtained for the various approaches

the FPR and TPR is presented in Fig. 4.9 for the various approaches. It is evident that, the ROC curve obtained by each of the approaches are above the random classifier and are therefore considered better than random guessing. Comparatively however, the ROC-Curve produced by the approach proposed by this study is the highest. We can therefore say that, the approach proposed by this study did a better job of classifying the positive class in the dataset.

4.4 Reducing False Positives in Real-Time Fraud Detection Using an Improved Multi-Layer HMM

For various number of hidden states, simulations were performed using a single-layer HMM, a Multi-layer HMMs proposed and implemented by Zegeye et al. (2019) and the improved Multi-layer HMM proposed by this study in Python programming environment and their performances compared in terms of Precision, Recall, F1-score, True and False Positive rates. Training and detection times were also compared to establish the computational efficiency of the various Models. Eighty percent of the dataset was used for training and the rest held for validation.



Table 4.2: Confusion Matrix for the various number of hidden states

Model	Actual #	Value of N	Positive Pred	Negative Pred
Single-layer HMM	+ve Class	2	115	12
	-ve Class		10	1253
	+ve Class	3	110	17
	-ve Class		6	1257
	+ve Class	4	119	8
	-ve Class		5	1258
	+ve Class	5	118	9
	-ve Class		4	1259
Proposed	+ve Class	2	125	2
	-ve Class		2	1261
	+ve Class	3	124	3
	-ve Class		3	1260
	+ve Class	4	121	6
	-ve Class		2	1261
	+ve Class	5	120	7
	-ve Class		1	1262
Zegeye et al. (2019)	+ve Class	2	121	6
	-ve Class		3	1260
	+ve Class	3	118	9
	-ve Class		4	1259
	+ve Class	4	120	7
	-ve Class		4	1259
	+ve Class	5	118	9
	-ve Class		3	1260



4.4.1 Confusion Matrix

Out of one-hundred and twenty seven (127) fraudulent and one-Thousand two hundred and sixty three (1,263) genuine transactions, the confusion matrix which reveals the performance of machine learning classification models with a focus on which classes are being predicted correctly or otherwise is presented in Table 4.2. To reveal a better picture regarding the performance of the proposed models, Recall and Precision evaluation metrics obtained from the confusion matrix are also employed. The relative low number of false positives and negatives obtained by the Model proposed by this study suggests that, our it is able to classify both positives and negative cases correctly more appropriately as compared to the Single-Layer and the Multi-layer HMMs proposed by Zegeye et al. (2019).

4.4.2 Precision, Recall and F1-score and ROC Curve Plot

In Table 4.3, the Precision, Recall and F1-Scores obtained by the various models which are computed from the confusion matrix as in Table 4.2 for the various number of hidden states are presented. It is evident that the values obtained by the model proposed by this study are relatively better than those obtained by using the single-layer HMM and the Multi-layer HMMs proposed by Zegeye et al. (2019). for all the various number of hidden states. The results also establishes that, the proposed model is able to predict fraudulent and genuine transaction most effectively.

Table 4.3: Precision, Recall and F1-scores obtained by the various Models

Metric	# Hidden States	Proposed	SLHMM	Zegeye et al. (2019).
Precision	2	0.984	0.920	0.976
	3	0.976	0.948	0.967
	4	0.984	0.960	0.968
	5	0.992	0.967	0.975
	2	0.984	0.906	0.953
Recall	3	0.976	0.866	0.929
	4	0.953	0.937	0.945
	5	0.945	0.929	0.929
	2	0.984	0.913	0.964
	3	0.976	0.905	0.948
F1 Score	4	0.968	0.948	0.956
	5	0.968	0.948	0.952

For the various number of hidden states, the Receiver Operating Characteristic (ROC) graph which is used for selecting an appropriate classification models using the True and False positive rates is presented in Figure 4.10. Considering the various number of hidden states, all the three (3) models have their ROC curves above random classifier. However,



it is evident that, the model proposed by this study obtained the highest AUC values for all values comparatively, signifying the very high values of TPs and extremely low values of FPs obtained by using the models proposed by this study.

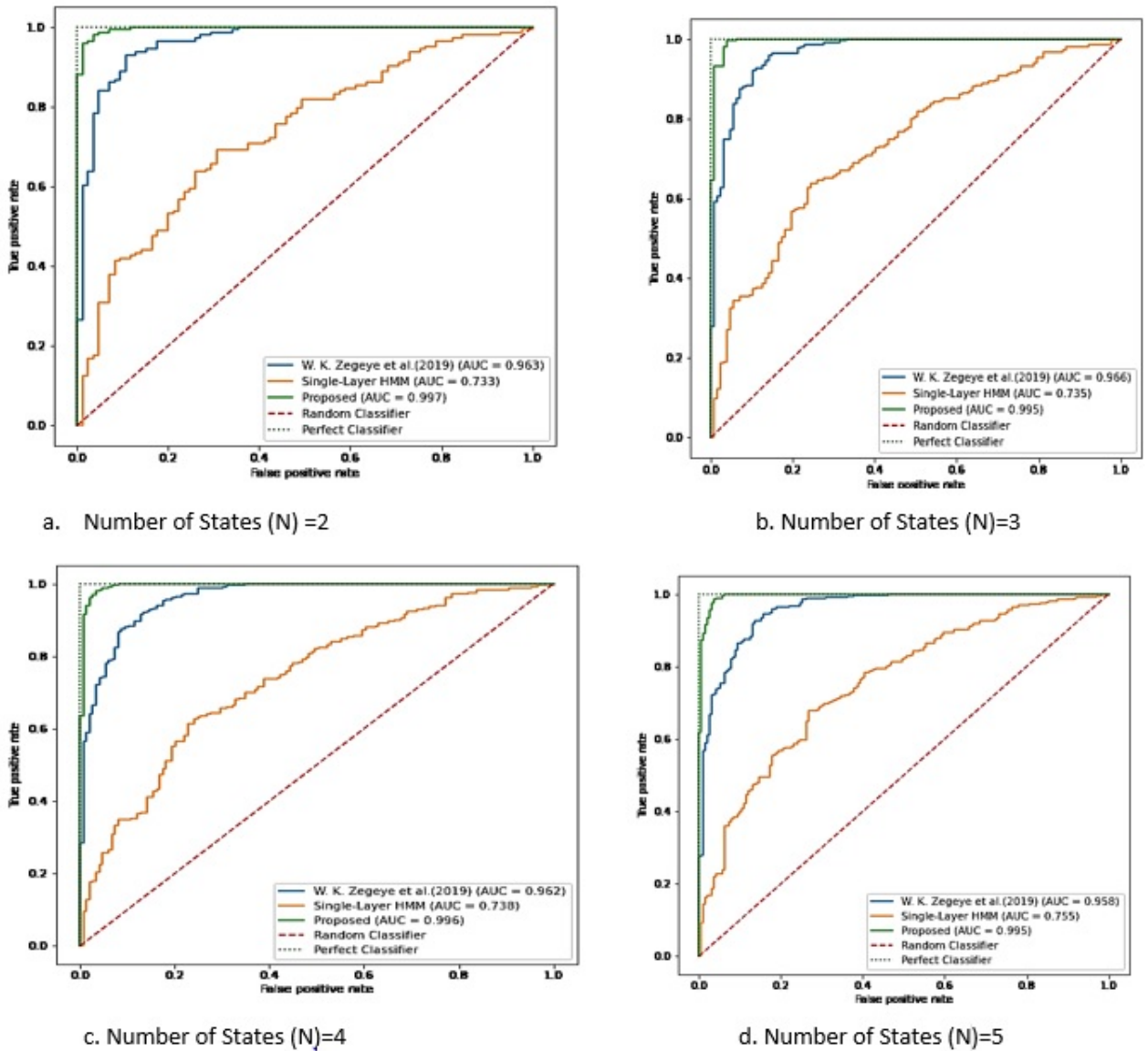


Figure 4.10: ROC Curve Plot of the models for various number of Hidden States

4.4.3 Computational efficiency of the Models

The efficiency of the various models in terms of training and detection times (in secs) for all the possible number of hidden states are established and summarized in Table 4.4 and their averages presented in Figure 4.11. Comparatively, the training and detection times obtained by the model proposed by this study are lowest. However, whilst the single-layer HMM performs better than the Multi-layer HMMs proposed by Zegeye et al. (2019) in terms of training times, it is the reverse for the detection times. The relative extremely



Table 4.4: Training and Detection times (secs) obtained by the various models for all the possible values of Hidden states

	# of Hidden States	Training time(s)	Detection time(s)
SLHMM	2	0.025	0.012
	3	0.098	0.015
	4	0.099	0.023
	5	0.102	0.025
	2	0.034	0.024
Zegeye et al. (2019)	3	0.052	0.025
	4	0.063	0.032
	5	0.078	0.034
	2	0.021	0.002
Proposed	3	0.032	0.002
	4	0.045	0.005
	5	0.036	0.013

lower values of the detection times makes the model proposed by this study ideal for real-time fraud detection.

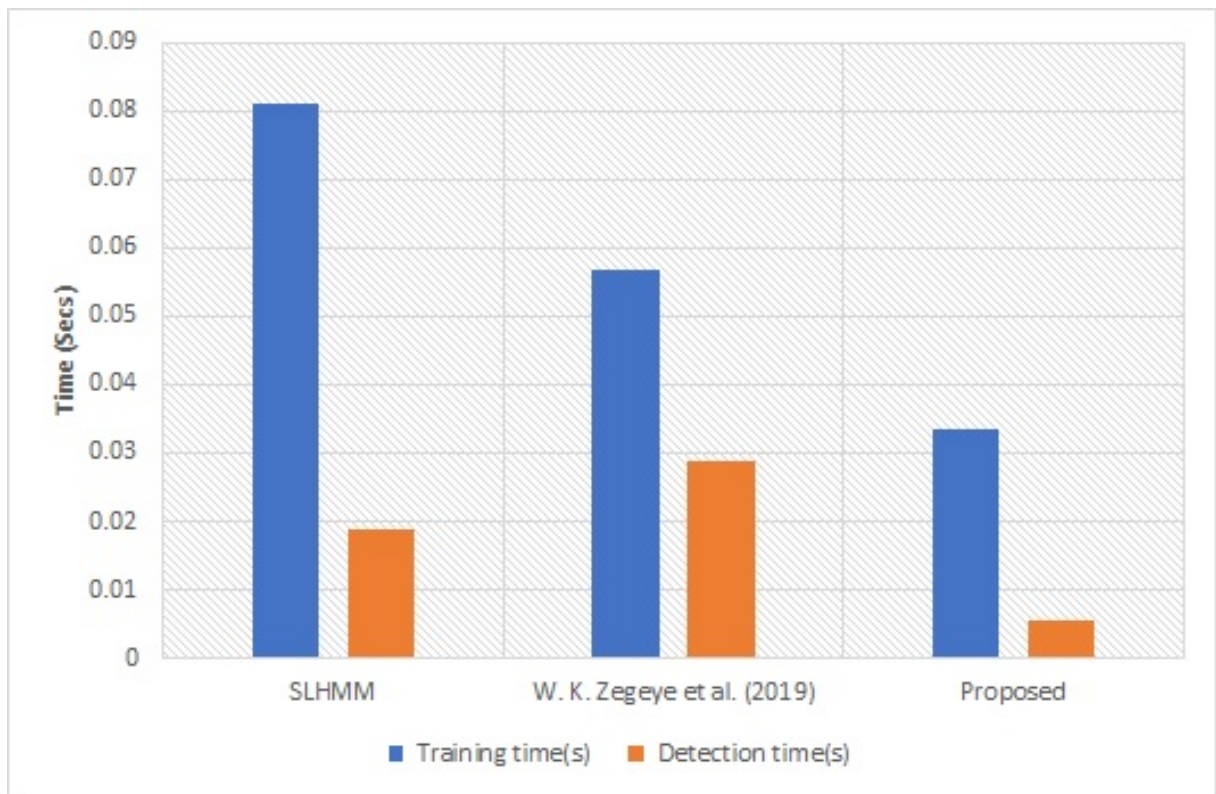


Figure 4.11: Average training and detection times (sec) obtained by the various models



4.5 Chapter Summary

The proposed hybrid optimization algorithm as well as the single and multiple layer HMMs proposed by Zegeye et al. (2019). have been simulated and their performances established with comparisons made against some existing models/techniques in literature. In the next chapter, conclusions and some recommendations will be made based on the major research findings.



CHAPTER 5

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.1 Introduction

A summary of research findings based on the simulation results obtained in the previous chapter, Conclusions based on the summary of findings and some recommendations outlined for future work are presented in this chapter.

5.2 Summary of Major Research Findings

5.2.1 An improved Hybrid Algorithm for optimizing the parameters of HMMs

Comparatively, in computing the probability of an observed sequence from the validation dataset, $P(O|\lambda)$ by the models which are optimized using various algorithms for all the number of hidden states, the BW algorithm performed the worst (Highest value =0.731 when N=4, and lowest value =0.531 when N=5) whilst the hybrid algorithm proposed by this study produced the best results (Highest value=0.932 when N=3, and lowest value =0.824 when N=2). Averagely, PSO and GA produced probabilities below 80 percent. Comparatively, apart from our proposed algorithm, the hybrid algorithms (Highest value =0.892 when N=3 for GAPSO, and lowest value =0.762 when N=2 for BWGA) performed better than the single ones (Highest value =0.785 when N=4 for GA, and lowest value =0.0.521 when N=5 for BW) Largely, the algorithms performed better when the number of Hidden States is set to 3 and 4.



In terms of the time taken to optimize the parameters of a HMM, the PSO algorithm performed best (best time=0.253secs for a Medium-Amount, High-Frequency transaction profile, and worst time=0.354secs for a Low-Amount, High-Frequency transaction profile). The hybrid algorithm proposed by this study obtained slightly lower training times (best time=0.269secs for a Medium-Amount, Medium-Frequency transaction profile, and worst time=0.589 for a High-Amount, High-Frequency transaction profile) but performs best in computing the probability of an observed sequence from the validation dataset for all the possible number of hidden states. Generally, apart from the algorithm proposed by this study, the time taken to optimize hybrid algorithms which incorporates GA are slightly higher (best time=0.708secs for a Low-Amount, Low-Frequency transaction profile, and worst time=0.954secs for a Medium-Amount, Low-Frequency transaction profile)

5.2.2 Detecting Real-Time Electronic Banking Fraud on highly imbalanced datasets using HMMs

In terms of Precision, employing K-Means without SMOTE to determine the transaction profile of customers produced the worse result (Best value=0.88 for N=4, and Worst value =0.69 for N=3) as compared to using the modified DBSCAN clustering algorithm as proposed by this study without SMOTE (Best value=0.88 for N=3 or 5, and Worst value =0.80 for N=2). Approaches that employed the SMOTE technique to handle the class imbalanced nature of the dataset produced better results (Best value=0.99 when N=5 for the approach proposed by this study, and Worst value =0.90 for N=2 or 3 for employing SMOTE with K-Means) as compared to techniques that did not employ SMOTE (Best value=0.88 for N=4 when K-means is employed without SMOTE and Worst value =0.69 for N=3, when the modified DBSCAN is used without SMOTE).

Likewise, in terms of Recall Rates, employing K-Means without SMOTE to determine the transaction profile of customers produced the worse result (Best value=0.83 for N=4, and Worst value =0.70 for N=2) as compared to using modified DBSCAN clustering algorithm as proposed by the study without SMOTE (Best value=0.88 for N=3, and Worst value =0.76 for N=4). Generally, employing the SMOTE technique to handle the



class imbalanced nature of the dataset produced better recall rates (Best value=0.96 when N=3 for the approach proposed by the study, and Worst value =0.90 when N=2 when SMOTE is employed with K-Means) as compared to techniques that failed to handle the class imbalanced problem (Best value=0.88 for N=2 when the modified DBSCAN is employed without SMOTE and Worst value =0.70 for N=3, when K-Means is used without SMOTE).

In terms of F1-Scores, using K-Means to determine the transaction profile of customers without handling the class imbalanced problem produced the worse result (Best value=0.87 for N=3, and Worst value =0.60 for N=2) as compared to using the modified DBSCAN clustering algorithm as proposed by this study without SMOTE (Best value=0.88 for N=3, and Worst value =0.67 for N=4). Comparatively, employing the SMOTE technique to handle the class imbalanced nature of the dataset produced better F1-Scores (Best value=0.95 for N=2 and Worst value =0.88 for N=2 both in approach proposed by this study) as compared to techniques that did not employ the SMOTE technique (Best value=0.88 for N=3 when the modified DBSCAN is employed without SMOTE and Worst value =0.60 for N=3, when K-Means is used without SMOTE)

5.2.3 Reducing False Positives in Real-Time Fraud Detection Using an Improved Multi-Layer HMMs

Out of 127 and 1,263 fraudulent and genuine transactions respectively, the Confusion Matrix revealed that, the proposed model recorded the least false positives (lowest=1 for N=5 and Highest=3 for N=3) and lowest False Negatives (lowest=2 for N=2 and Highest=7 for N=5). However, the Multi-layer HMM proposed by Zegeye et al. (2019) performed better than the single-layer HMM in terms of the number of false positives (lowest=3 for N=5,4 and Highest=4 for N=3,4) and false negatives (lowest=6 for N=2 and Highest=9 for N=3,5). The highest number of false positives and false negatives are obtained by the single-layer HMM (highest FP=10 for N=2, Highest FN=17 for N=3).

In terms of Precision, Recall rates and F1-Scores, the improved multilayer HMM proposed by this study obtained the highest scores (Precision=0.992 for N=5, Recall



Rate= 0.984 for N=2 and F1-Score=0.984 for N=2) whilst the single-Layer HMM recorded the lowest (Precision=0.0.948 for N=3, Recall Rate= 0.866 for N=3 and F1-Score=0.905 for N=3). From the ROC graph, although all the models have their ROC curves above the random classifier for all the various number of hidden states and can therefore be considered as better than random guessing, the model proposed by the study obtained the highest AUC for all the various number of hidden states (AUC=0.997,0.995,0.996 and 0.995 for N=2,3,4 and 5 respectively) whilst the single-layer HMMs obtained the least (AUC=0.733,0.735,0.738 and 0.755 for N=2,3,4 and 5 respectively) .

5.3 Conclusion

5.3.1 Implementing the improved Hybrid Algorithm for optimizing the parameters of HMMs

The proposed hybrid optimization algorithm leverages on the advantages the BW, GA and PSO optimization algorithms which has been widely used in literature for optimizing the parameters of HMMs for improved performance. In the proposed algorithm, the BW algorithm estimates reasonably rather than random guess the particles of the GA. However, when the number of elements which are exposed to mutation is large in GA, there is often an exponential increase in search space size which leads to a poor performance. The PSO algorithm which has guaranteed convergence and averagely performs efficiently when applied to comparatively complex problems is then introduced to produce a global solution of the model parameters.

In essence, it overcomes the shortcomings of the slow convergence speed of the PSO, and also assist the BW escape from local optimal solution whilst improving the performance of the GA by reducing its search space. The execution time of the proposed algorithm is slightly higher than that obtained by the BW and the PSO algorithms.



5.3.2 Detecting Real-Time Electronic Banking Fraud on highly imbalanced datasets using the proposed HMMs

An improved electronic banking fraud detection framework based on HMM and modified Density Based Spatial Clustering of Applications with Noise (DBSCAN) is proposed and implemented. The Synthetic Minority Oversampling Technique (SMOTE) is also implemented to handle the highly class imbalance nature of the dataset adopted.

As outline in section 4.3, for various number of hidden states, the approach proposed by this study performed relatively better when compared to some common approaches employed in literature in terms of precision, recall and F1-Scores. Employing the modified DBSCAN clustering technique to determine the spending profile of customers produced better recognition rates as compared to using K-Means algorithm since it filters out most of the easily recognizable fraudulent transactions before the proposed HMMs are applied. It is also evident from the simulation analysis that, the SMOTE technique effectively handles the class imbalance classification necessary to achieve improved performance.

The AUC-ROC plot and learning curve also reveals that, the approach proposed by the study did a better job in classifying the positive class in the dataset and shows the best score in both training and cross-validating sets.

5.3.3 Reducing False Positives in Real-Time Fraud Detection Using the Improved Multi-Layer HMMS proposed by the study

An enhanced multi-layer HMM which incorporates an effective training and detection processes is developed and implemented for real-time detection with a focus on reducing false positives. The training dataset is divided into a number of sub-groups where each group includes a specific user behavior pattern is used to train specific HMMs which are merged to create the final model using a linear recombination procedure. A comparison between available normal sequences in a database is first and foremost compared to any retrieved observation sequence.



From simulation results, the significant fewer number of false positives and negatives obtained by the models proposed by the study as compared to the single-layer and the Multi-layer HMM proposed by Zegeye et al. (2019) suggest that, it is able to best classify both positives and negative cases most accurately. It is also evident that from the ROC-AUC plot that, the model proposed by the study recorded the highest AUC values for all the various number of hidden states which signifies that, comparatively, it is able to predict fraudulent and genuine transactions more accurately with better training and detection times.

5.4 Recommendations for future research

The results obtained by the research are very promising and future work will focus on applying the developed models in other areas of applications such as speech and image recognition, motion/action analysis in videos, bio-informatics among others. In addition, the following are recommended for future research;

1. With the current rise in interest and development in the area of big data and machine learning, it is essential to develop and implement Fraud detection models using heterogeneous data sources in distributed environments.
2. Proposing techniques for fast incremental learning using HMMs for both misuse and anomaly detection.
3. The combination of HMMs and other machine-learning techniques to precisely estimate the number of hidden states and the values of the transition matrix since they have a significant effect on the complexity of the Models.
4. Proposing robust architectures to drastically reduce false-positive by combining HMM and other data mining techniques.



References

- Abdallah, A., Maarof, M. A., & Zainal, A. (2016). Fraud detection system: A survey. *Journal of Network and Computer Applications*, 68(1), 90–113. doi: 10.1016/j.jnca.2016.04.007
- Abomhara, M., & Køien, G. M. (2015). Cyber security and the internet of things: Vulnerabilities, threats, intruders and attacks. *Journal of Cyber Security and Mobility*, 4(1), 65–88. doi: 10.13052/jcsm2245-1439.414
- Abouabdalla, O., El-Taj, H., Manasrah, A., & Ramadass, S. (2009). False positive reduction in intrusion detection system: A survey. *Proceedings of 2009 2nd IEEE International Conference on Broadband Network and Multimedia Technology, IEEE IC-BNMT2009*, 3(6), 463–466. doi: 10.1109/ICBNMT.2009.5348536
- Achituve, I., Kraus, S., & Goldberger, J. (2019). Interpretable Online Banking Fraud Detection Based on Hierarchical Attention Mechanism. *IEEE International Workshop on Machine Learning for Signal Processing, MLSP, 2019-October(1)*, 1–6. doi: 10.1109/MLSP.2019.8918896
- Adedoyin, A. (2018). *Predicting fraud in mobile money transfer* (Unpublished doctoral dissertation). University of Brighton.
- Aggelis, V. (2006). Offline internet banking fraud detection. *Proceedings - First International Conference on Availability, Reliability and Security, ARES 2006*, 2006(4), 904–905. doi: 10.1109/ARES.2006.89
- Ahmadian Ramaki, A., Rasoolzadegan, A., & Javan Jafari, A. (2018). A systematic review on intrusion detection based on the Hidden Markov Model. *Statistical Analysis and Data Mining*, 11(3), 111–134. doi: 10.1002/sam.11377



- Alghamdi, R. (2016). Hidden Markov Models (HMMs) and Security Applications. *International Journal of Advanced Computer Science and Applications*, 7(2), 39–47. doi: 10.14569/ijacsa.2016.070205
- Ali, M. A., Hussin, N., & Abed, I. A. (2019). E-banking fraud detection: A short review. *International Journal of Innovation, Creativity and Change*, 6(8), 67–87. doi: 10.1147/IJICC.2009.5588852
- Alimolaei, S. (2016). An intelligent system for user behavior detection in Internet Banking. *4th Iranian Joint Congress on Fuzzy and Intelligent Systems, CFIS 2015*, 9(5), 46–46. doi: 10.1109/CFIS.2015.7391642
- Amalina, F., Targio Hashem, I. A., Azizul, Z. H., Fong, A. T., Firdaus, A., Imran, M., & Anuar, N. B. (2020). Blending Big Data Analytics: Review on Challenges and a Recent Study. *IEEE Access*, 8(7), 3629–3645. doi: 10.1109/ACCESS.2019.2923270
- Andrade, E. L., Blunsden, S., & Fisher, R. B. (2006). Hidden Markov Models for optical flow analysis in crowds. *Proceedings - International Conference on Pattern Recognition*, 1(4), 460–463. doi: 10.1109/ICPR.2006.621
- Asare, M., & Sakoe, J. (2015). The Effects of Electronic Banking on Financial Services in Ghana. *Research Journal of Finance and Accounting*, 6(16), 147–155.
- Atwi, A., Savla, K., & Dahleh, M. A. (2011). A case study in robust quickest detection for hidden Markov models. *Proceedings of the American Control Conference*, 4(9), 780–785. doi: 10.1109/acc.2011.5991027
- Aupetit, S., Monmarché, N., & Slimane, M. (2007). Hidden Markov models training by a particle swarm optimization algorithm. *Journal of Mathematical Modelling and Algorithms*, 6(2), 175–193. doi: 10.1007/s10852-005-9037-7
- Bandgar, M. (2013). Intrusion Detection System using Hidden Markov Model (HMM). *IOSR Journal of Computer Engineering*, 10(3), 66–70. doi: 10.9790/0661-01036670
- Bank of Ghana. (2020). BANK OF GHANA Banking Sector Report. *Corporate governance directive*, 2(December), 1–1161.



- Barrios, R. (2013). A Multi-Levelled Approach to Intrusion Detection and the Insider Threat. *Journal of Information Security*, 04(01), 54–65. doi: 10.4236/jis.2013.41007
- Bhattacharya, A. (2014). Curse of Dimensionality. *Fundamentals of Database Indexing and Searching*, 141–148. doi: 10.1201/b17767-13
- Bhusari, & Patil, S. (2011). Application of Hidden Markov Model in Credit Card Fraud Detection. *International Journal of Distributed and Parallel systems*, 2(6), 203–211. doi: 10.5121/ijdps.2011.2618
- Bignell, K. B. (2006). Authentication in an internet banking environment; Towards developing a strategy for fraud detection. *International Conference on Internet Surveillance and Protection, ICISP'06*, 11(3), 23–30. doi: 10.1109/ICISP.2006.3
- Burgio, D. (2019). Reduction of False Positives in Intrusion Detection Based on Extreme Learning Machine with Situation Awareness. *CCE Theses and Dissertations*, 45(1093). Retrieved from <https://nsuworks.nova.edu/gscis-etd/1093/>
- Calvo-Zaragoza, J., Toselli, A. H., & Vidal, E. (2019). Hybrid hidden Markov models and artificial neural networks for handwritten music recognition in mensural notation. *Pattern Analysis and Applications*, 22(4), 1573–1584. doi: 10.1007/s10044-019-00807-1
- Cao, W., Zhu, W., & Demazeau, Y. (2019). Multi-Layer Coupled Hidden Markov Model for Cross-Market Behavior Analysis and Trend Forecasting. *IEEE Access*, 7(10), 158563–158574. doi: 10.1109/ACCESS.2019.2950437
- Carminati, M., Caron, R., Maggi, F., Epifani, I., & Zanero, S. (2015). BankSealer DETAILS. *Computers and Security*, 53(2), 175–186. doi: 10.6786/545345345-2
- Chang, L., Ouzrout, Y., Nongailard, A., & Bouras, A. (2018). Optimized Hidden Markov Model based on Constrained Particle Swarm Optimization. , 02(2), 1–5. Retrieved from <http://arxiv.org/abs/1811.03450> doi: 1811.03450



- Chau, C. W., Kwong, S., Diu, C. K., & Fahrner, W. R. (1997). Optimization of HMM by a genetic algorithm. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 3(5), 1727–1730. doi: 10.1109/icassp.1997.598857
- Chen, T.-Y., Yi, T., Mei, X. D., Pan, J. S., & Sun, S. H. (2004). Optimization of HMM by the tabu search algorithm. *Journal of Information Science and Engineering*, 20(5), 949–957. doi: 10.6688/JISE.2004.20.5.4
- Correa Bahnsen, A., Aouada, D., Stojanovic, A., & Ottersten, B. (2016). Feature engineering strategies for credit card fraud detection. *Expert Systems with Applications*, 51(1), 134–142. Retrieved from <http://dx.doi.org/10.1016/j.eswa.2015.12.030> doi: 10.1016/j.eswa.2015.12.030
- Dandash, O., Wang, Y., Le, P. D., & Srinivasan, B. (2008). Fraudulent internet banking payments prevention using dynamic key. *Journal of Networks*, 3(1), 25–34. doi: 10.4304/jnw.3.1.25-34
- Daniel, N., & Macas, M. (2014). Particle Swarm Optimization of Hidden Markov Models : a comparative study. *journal of Development*, 9(3). doi: 10.1109/JOD.2016.7569180
- Devi, A. (2018). Mobile Banking: The Revolution in Digitalization of Financial Services with Special Reference to State Bank of India. *International Research Journal of Management Science and Technology*, 9(4), 49-58.
- Diadiushkin, A., Sandkuhl, K., & Maiatin, A. (2019). Fraud Detection in Payments Transactions: Overview of Existing Approaches and Usage for Instant Payments. *Complex Systems Informatics and Modeling Quarterly*, 6(20), 72–88. doi: 10.7250/csimq.2019-20.04
- Fang, S., & Zhang, X. (2016). A Hybrid Algorithm of Particle Swarm Optimization and Tabu Search for Distribution Network Reconfiguration. *Mathematical Problems in Engineering*, 20(1), 16–20. doi: 10.1155/2016/7410293
- Fengqin, Y., & Changhai, Z. (2008). An effective hybrid optimization algorithm for HMM. *Proceedings - 4th International Conference on Natural Computation, ICNC 2008*, 4(3), 80–84. doi: 10.1109/ICNC.2008.367



- Ghahramani, Z. (2001). An introduction to hidden Markov models and Bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1), 9–42. doi: 10.1142/S0218001401000836
- Ghamri, N. S. (2017). Positive and Negative Effects of Using Electronic Banking on Customers and Small Entrepreneurs: An Exploratory Study in the Western Region of Saudi Arabia. *Business and Economic Research*, 7(2), 311. doi: 10.5296/ber.v7i2.11999
- Goeschel, K. (2016). Reducing false positives in intrusion detection systems using data-mining techniques utilizing support vector machines, decision trees, and naive Bayes for off-line analysis. *Conference Proceedings - IEEE SOUTHEASTCON*, 9(78). doi: 10.1109/SECON.2016.7506774
- Görnitz, N., Braun, M., & Kloft, M. (2015). Hidden Markov anomaly detection. *32nd International Conference on Machine Learning, ICML 2015*, 3(1), 1833–1842. doi: 10.1455/2015/12121
- Grewal, J. K., Krzywinski, M., & Altman, N. (2020). Markov models — training and evaluation of hidden Markov models. *Nature Methods*, 17(2), 121–122. doi: 10.1038/s41592-019-0702-6
- Guan, F., Wu, J., & Cui, W. (2017). A method of fault alarm recognition based on hidden Markov model. *Proceedings of 2016 Prognostics and System Health Management Conference, PHM-Chengdu 2016*, 8(7), 1–4. doi: 10.1109/PHM.2016.7819760
- Gupta, A., Kumar, D., & Barve, A. (2017). Hidden Markov Model based Credit Card Fraud Detection System with Time Stamp and IP Address. *International Journal of Computer Applications*, 166(5), 33–37. doi: 10.5120/ijca2017914060
- Haikuan, L., Dachao, Y., Lei, Z., Zhiyuan, L., & Dawei, J. (2019). A New Improved Simplified Particle Swarm Optimization Algorithm. *Journal of Physics: Conference Series*, 1187(4). doi: 10.1088/1742-6596/1187/4/042077
- Hassan, R., Cohanin, B., De Weck, O., & Venter, G. (2005). A comparison of particle swarm optimization and the genetic algorithm. *Collection of Technical Papers -*



AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 2(April), 1138–1150. doi: 10.2514/6.2005-1897

Helode, P. S., Dr. K. H. Walse, & Karande M.U. (2017). An Online Secure Social Networking with Friend Discovery System. *International Journal of Innovative Research in Computer and Communication Engineering*, 5(4), 8198–8205. Retrieved from www.ijircce.com doi: 10.15680/IJIRCCE.2017

Hewahi, N. M. (2015). Particle Swarm Optimization For Hidden Markov Model. *International Journal of Knowledge and Systems Science*, 6(2), 1–15. doi: 10.4018/ijkss.2015040101

Hoang, X. D., Hu, J., & Bertok, P. (2003). A multi-layer model for anomaly intrusion detection using program sequences of system calls. *IEEE International Conference on Networks, ICON*, 1(1), 531–536. doi: 10.1109/icon.2003.1266245

Islam, S., & Mahfuz, A. S. (2014). Adoption of e-banking in Bangladesh: Evolution, status and prospects. *16th Int'l Conf. Computer and Information Technology, ICCIT 2013*, 1(3), 255–260. doi: 10.1109/ICCITechn.2014.6997303

Kang, H. (2019). Fraud Detection in Mobile Money Transactions Using Machine Learning. *Information Systems and Business Analytics*, 5(32), 320–332. Retrieved from <https://lib.dr.iastate.edu/creativecomponents> doi: 20.500.12876/16959

Kar, D., Agarwal, K., Sahoo, A. K., & Panigrahi, S. (2016). Detection of SQL injection attacks using hidden markov model. *Proceedings of 2nd IEEE International Conference on Engineering and Technology, ICETECH 2016*, 7(6), 1–6. doi: 10.1109/ICETECH.2016.7569180

Khare, N., Sait, S. Y., Campus, K., & Campus, K. (2018). Credit Card Fraud Detection Using Machine Learning Models and Collating Machine Learning Models. *International Journal of Pure and Applied Mathematics*, 118(20), 825–838. doi: 10.1109/IJ-PAP.2018.522



- Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, 2(1). doi: 10.1186/s42400-019-0038-7
- Ko, K. E., Park, S. M., Park, J., & Sim, K. B. (2012). Training HMM structure and parameters with genetic algorithm and harmony search algorithm. *Journal of Electrical Engineering and Technology*, 7(1), 109–114. doi: 10.5370/JEET.2012.7.1.109
- Kovach, S., & Ruggiero, W. (2011). Online banking fraud detection based on local and global behavior. *ICDS 2011, The Fifth International Conference on Digital Society*, 5(2), 166–171. Retrieved from <http://www.thinkmind.org/index.php?view=articleandarticleid=icds-2011-6-40-90006> doi: 10.46532/ICDS-2020210101
- Kumari, N., Kannan, S., & Muthukumaravel, A. (2014). Credit card fraud detection using Hidden Markov Model-A survey. *Middle - East Journal of Scientific Research*, 20(6), 697–699. doi: 10.5829/idosi.mejsr.2014.20.06.11387
- Kwong, S., Chau, C. W., Man, K. F., & Tang, K. S. (2001). Optimisation of HMM topology and its model parameters by genetic algorithms. *Pattern Recognition*, 32(12), 17–28. doi: 10.1016/S0031-3203(99)00226-5
- Li, Q., & Lau, H. C. (2016). A layered hidden Markov model for predicting human trajectories in a multi-floor building. *Proceedings - 2015 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT 2015*, 2(1), 344–351. doi: 10.1109/WI-IAT.2015.239
- Lookingbill, T. R., & Urban, D. L. (2005). Gradient analysis, the next generation: Towards more plant-relevant explanatory variables. *Canadian Journal of Forest Research*, 35(7), 1744–1753. doi: 10.1139/x05-109
- Lopez-Rojas. (2014). *On the simulation of financial transactions for fraud detection research* (Unpublished doctoral dissertation). Blekinge Institute of Technology.



- Maruatona, O. O., Vamplew, P., & Dazeley, R. (2012). Prudent fraud detection in internet banking. *Proceedings - 2012 3rd Cybercrime and Trustworthy Computing Workshop, CTC 2012*, 5(2), 60–65. doi: 10.1109/CTC.2012.13
- Mehra, P. (2015). Controlling Attacks and Intrusions on Internet Banking using Intrusion Detection System in Banks. , 4(11), 346–348. doi: 10.17148/IJARCCE.2015.41176
- Mhamane, S. S., & Lobo, L. M. (2012). Internet banking fraud detection using HMM. *2012 3rd International Conference on Computing, Communication and Networking Technologies, ICCCNT 2012*, 37(7), 200–204. doi: 10.1109/ICCCNT.2012.6395910
- Moerland, P. (1996). *MicroNeuro'96, February 12–14, 1996, Lausanne, Switzerland* (Vol. 12) (No. 4). doi: 10.1016/0925-2312(96)00016-1
- Moin, N. H., Chung Sin, O., & Omar, M. (2015). Hybrid genetic algorithm with multiparents crossover for job shop scheduling problems. *Mathematical Problems in Engineering*, 2015(8), 67–75. doi: 10.1155/2015/210680
- Mor, B., Garhwal, S., & Kumar, A. (2021). A Systematic Review of Hidden Markov Models and Their Applications. *Archives of Computational Methods in Engineering*, 28(3), 1429–1448. Retrieved from <https://doi.org/10.1007/s11831-020-09422-4> doi: 10.1007/s11831-020-09422-4
- NGGL. (2013). Kwame Nkrumah University of Science and Technology , Kumasi Institute of Distance Learning. *Information Development*, 20(1), 207–220. Retrieved from <http://www.knust.edu.gh/>
- Nikravesh, A. Y., Ajila, S. A., & Lung, C. H. (2018). Using genetic algorithms to find optimal solution in a search space for a cloud predictive cost-driven decision maker. *Journal of Cloud Computing*, 7(1). doi: 10.1186/s13677-018-0122-7
- Panigrahi, S., Kundu, A., Sural, S., & Majumdar, A. K. (2009). Credit card fraud detection: A fusion approach using Dempster-Shafer theory and Bayesian learning. *Information Fusion*, 10(4), 354–363. Retrieved from <http://dx.doi.org/10.1016/j.inffus.2008.04.001> doi: 10.1016/j.inffus.2008.04.001



- Penagarikano, M., & Bordel, G. (2004). Layered Markov models: A New architectural approach to automatic speech recognition. *Machine Learning for Signal Processing XIV - Proceedings of the 2004 IEEE Signal Processing Society Workshop(Lmm)*, 305–314. doi: 10.1109/mlsp.2004.1422988
- Pérez, Ó., Piccardi, M., García, J., Patricio, M. A., & Molina, J. M. (2007). Comparison between genetic algorithms and the Baum-Welch algorithm in learning HMMs for human activity classification. *Lecture Notes in Computer Science (including sub-series Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4448 LNCS(10), 399–406. doi: 10.1007/978-3-540-71805-5-44
- Philip, D. J., Sudarsanam, N., & Ravindran, B. (2018). A partial parameter HMM based clustering on loan repayment data: Insights into financial behavior and intent to repay. *Proceedings of the Annual Hawaii International Conference on System Sciences, 2018-Janua*(8), 1376–1385. doi: 10.24251/hicss.2018.170
- Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A Comprehensive Survey of Data Mining-based Fraud Detection Research. , 28(3). Retrieved from doi: 10.1016/j.chb.2012.01.002
- Prakash, A., & Chandrasekar, C. (2012). A Novel Hidden Markov Model for Credit Card Fraud Detection. *International Journal of Computer Applications*, 59(3), 35–41. doi: 10.5120/9532-3960
- Prakash, A., & Chandrasekar, C. (2015). An optimized multiple semi-hidden markov model for credit card fraud detection. *Indian Journal of Science and Technology*, 8(2), 176–182. doi: 10.17485/ijst/2015/v8i2/58081
- Prasad, N. R., Almanza-Garcia, S., & Lu, T. T. (2009). Anomaly detection. *Computers, Materials and Continua*, 14(1), 1–22. doi: 10.1145/1541880.1541882
- Qassim, Q., Patel, A., & Zin, A. M. (2014). Strategy to reduce false alarms in intrusion detection and prevention systems. *International Arab Journal of Information Technology*, 11(5), 500–506. doi: 10.5888/IAJIF.54354355



- Qin, F., Auerbach, A., & Sachs, F. (2000). A direct optimization approach to hidden markov modeling for single channel kinetics. *Biophysical Journal*, 79(4), 1915–1927. Retrieved from [http://dx.doi.org/10.1016/S0006-3495\(00\)76441-1](http://dx.doi.org/10.1016/S0006-3495(00)76441-1) doi: 10.1016/S0006-3495(00)76441-1
- Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 72(2), 257–286. doi: 10.1109/5.18626
- Rambola, R., Varshney, P., & Vishwakarma, P. (2018). Data mining techniques for fraud detection in banking sector. *2018 4th International Conference on Computing Communication and Automation, ICCCA 2018*, 56(23), 1–5. doi: 10.1109/CCAA.2018.8777535
- Rieke, R., Zhdanova, M., Repp, J., Giot, R., & Gaber, C. (2013). Fraud detection in mobile payments utilizing process behavior analysis. *Proceedings - 2013 International Conference on Availability, Reliability and Security, ARES 2013*, 662–669. doi: 10.1109/ARES.2013.87
- Sangita, D., & Madhuri, S. (2015). Securing Online Banking Transaction using Predictive Approach of Hidden Markov Model. *International Journal of Computer Applications*, 128(7), 14–17. doi: 10.5120/ijca2015906603
- Seeja, K. R., & Zareapoor, M. (2014). FraudMiner: A novel credit card fraud detection model based on frequent itemset mining. *Scientific World Journal*, 2014(15), 300–315. doi: 10.1155/2014/252797
- Shi, X. H., Liang, Y. C., Lee, H. P., Lu, C., & Wang, L. M. (2005). An improved GA and a novel PSO-GA-based hybrid algorithm. *Information Processing Letters*, 93(5), 255–261. doi: 10.1016/j.ipl.2004.11.003
- Shuangqing, C., Liu, Y., Wei, L., & Guan, B. (2018). PS-FW: A Hybrid Algorithm Based on Particle Swarm and Fireworks for Global Optimization. *Computational Intelligence and Neuroscience*, 2018(3), 200–214. doi: 10.1155/2018/6094685
- Slimane, M., Venturini, G., Debeauville, J. P., Brouard, T., & Brandeau, A. (1996). Optimizing hidden Markov models with a genetic algorithm. *Lecture Notes in Computer*



Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1063(19), 384–396. doi: 10.1007/3-540-61108-8-52

- Spathoulas, G. P., & Katsikas, S. K. (2010). Reducing false positives in intrusion detection systems. *Computers and Security*, 29(1), 35–44. Retrieved from <http://dx.doi.org/10.1016/j.cose.2009.07.008> doi: 10.1016/j.cose.2009.07.008
- Srivastava, A., Kundu, A., Sural, S., & Majumdar, A. K. (2008). Credit card fraud detection using Hidden Markov Model. *IEEE Transactions on Dependable and Secure Computing*, 5(1), 37–48. doi: 10.1109/TDSC.2007.70228
- Sulaiman, M. S., & AbdelKarim, N. (2019). Electronic Banking Strategies and Their Impact on Customers' Satisfaction: Empirical Evidence from Palestine. *Asian Social Science*, 15(10), 20. doi: 10.5539/ass.v15n10p20
- Sultana, A., Hamou-Lhadj, A., & Couture, M. (2012). An improved Hidden Markov Model for anomaly detection using frequent common patterns. *IEEE International Conference on Communications*, 2(4), 1113–1117. doi: 10.1109/ICC.2012.6364527
- Talekar, D. L. (2015). Credit Card Fraud Detection Using Hmm And Image Click Point Authentication. , 4(3), 1–7. doi: 10.21275/ART20192099
- Thatphithakkul, N., & Kanokphara, S. (2004). HMM parameter optimization using Tabu search. *IEEE International Symposium on Communications and Information Technologies: ISCIT 2004*, 2(30), 904–908. doi: 10.1109/iscit.2004.1413850
- Thiruvadi, S., & Patel, S. C. (2011). *Survey of data-mining techniques used in fraud detection and prevention* (Vol. 10) (No. 4). doi: 10.3923/itj.2011.710.716
- Vaidya, A. H., & Mohod, P. S. W. (2013). A Review on Internet Banking Fraud Detection using HMM and BLAST-SSAHA Hybridization. *International Journal of Engineering Research and Technology*, 2(10), 2287–2291. doi: 10.1016/IJERTV2IS100807.2013
- VanHoose, D. D. (2011). Internet Banking. *SSRN Electronic Journal*, 3(1). doi: 10.2139/ssrn.1525474



- Vipparla, V. K., Chilukuri, P. K., & Kande, G. B. (2021). Attention Based Multi-Patched 3D-CNNs with Hybrid Fusion Architecture for Reducing False Positives during Lung Nodule Detection. *Journal of Computer and Communications*, 09(04), 1–26. doi: 10.4236/jcc.2021.94001
- Wang, Y., Wang, L., & Yang, J. (2020). Egonet based anomaly detection in E-bank transaction networks. *IOP Conference Series: Materials Science and Engineering*, 715(1). doi: 10.1088/1757-899X/715/1/012038
- Wedge, R., Kanter, J. M., Veeramachaneni, K., Rubio, S. M., & Perez, S. I. (2019). Solving the false positives problem in fraud prediction using automated feature engineering. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (Vol. 11053 LNAI, pp. 372–388). doi: 10.1007/978-3-030-10997-4-23
- Wei, W., Li, J., Cao, L., Ou, Y., & Chen, J. (2013). Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web*, 16(4), 449–475. doi: 10.1007/s11280-012-0178-0
- Xiang, C., Wang, Z. X., & Pan, X. (2019). HIV-1 tropism prediction by the XGboost and HMM methods. *Scientific Reports*, 9(1), 1–8. doi: 10.1038/s41598-019-46420-4
- Xiao, J., Zou, L., & Li, C. (2007). Optimization of Hidden Markov Model by a Genetic Algorithm for Web Information Extraction. (6). doi: 10.2991/iske.2007.48
- Xiaoguo, W., Wu, H., & Yi, Z. (2018). Research on Bank Anti-Fraud Model Based on K-Means and Hidden Markov Model. *2018 3rd IEEE International Conference on Image, Vision and Computing, ICIVC 2018*, 10(4), 780–784. doi: 10.1109/ICIVC.2018.8492795
- Xu, Z., Yu, X., Feng, Y., Hu, J., Tari, Z., & Han, F. (2013). A multi-module anomaly detection scheme based on system call prediction. *Proceedings of the 2013 IEEE 8th Conference on Industrial Electronics and Applications, ICIEA 2013*, 1376–1381. doi: 10.1109/ICIEA.2013.6566581



- Xue, L., Yin, J., Ji, Z., & Jiang, L. (2007). A particle swarm optimization for hidden Markov model training. *International Conference on Signal Processing Proceedings, ICSP*, 1(12), 2–5. doi: 10.1109/ICOSP.2006.345542
- Yang, F., Zhang, C., & Sun, T. (2008). Comparison of Particle Swarm Optimization and Genetic Algorithm for HMM training. *Proceedings - International Conference on Pattern Recognition*, 1(12), 8–11. doi: 10.1109/icpr.2008.4761282
- Zamini, M., & Hasheminejad, S. M. H. (2019). A comprehensive survey of anomaly detection in banking, wireless sensor networks, social networks, and healthcare. *Intelligent Decision Technologies*, 13(2), 229–270. doi: 10.3233/IDT-170155
- Zegeye, W., Farzad, M., & Richard, D. (2019). Multi-stage attack detection using layered Hidden Markov model intrusion detection system. *Proceedings of the International Telemetering Conference*, 55(4), 273–282. doi: 10.1145/1541880.5488551
- Zegeye, W., Richard, D., & Farzad, M. (2018). Multi-Layer Hidden Markov Model Based Intrusion Detection System. *Machine Learning and Knowledge Extraction*, 1(1), 265–286. doi: 10.3390/make1010017
- Zhang, Q. (2009). Study on fraud risk prevention of online banks. *Proceedings - International Conference on Networks Security, Wireless Communications and Trusted Computing, NSWCTC 2009*, 2(2), 181–184. doi: 10.1109/NSWCTC.2009.312





APPENDICES

APPENDIX A

SAMPLE PYTHON SOURCE CODES

A.1 Clustering Algorithms

Kmeans clustering algorithm

```
class KMeansClustering:
    def __init__(self, n_clusters=4):
        self.n_clusters = n_clusters
        self._model = KMeans(n_clusters=n_clusters, random_state=0)
    def run(self, data):
        print('Clustering ...')
        data = np.array([[x] for x in data])
        print('Clustering is finished.')
        return self._model.fit(X=data).labels_
```

```
    def predict(self, sample):
        return self._model.predict(X=[[sample]])
```

Modified DBSCAN clustering algorithm

```
class DBSCANClustering:
    def __init__(self, n_neighbors=5, eps=5):
        self._model =
        DBSCAN(min_samples=n_neighbors, eps=eps)
    def run(self, data):
        print('Clustering ...')
        data = np.array([[x] for x in data])
```



```
print('Clustering is finished.')
```

```
return self._model.fit(X=data).labels_ + 1
```

```
def predict(self, sample):
```

```
return self._model.fit_predict(X=[[sample]]) + 1
```

```
def run(self, data):
```

```
print('Clustering ...')
```

```
data = np.array([[x] for x in data])
```

```
print('Clustering is finished.')
```

```
return self._model.fit(X=data).labels_
```



```
def predict(self, sample):
```

```
x = [[sample]]
```

```
for i in range(self.n_clusters):
```

```
x.append([sample + i])
```

```
return self._model.fit_predict(X=x)[0]
```

```
def run(self, data):
```

```
print('Clustering ...')
```

```
data = np.array([[x] for x in data])
```

```
print('Clustering is finished.')
```

```
return self._model.fit(X=data).labels_
```

```
def predict(self, sample):
```

```
x = [[sample]]
```

```
for i in range(self.n_clusters):
```

```
x.append([sample + i])
```

```
return self._model.fit_predict(X=x)[0]
```



A.2 Fraud Detection

```
import numpy as np

from Source.clustering import KMeansClustering
from Source.driver import Driver
from Source.hidden_markov_model import HMM
from config import *

def get_input():
    while True:
        new_transaction = input('Please add your new transaction : ')
        if int(new_transaction) == TERMINATE:
            break

        new_transaction = k.predict(int(new_transaction))
        new_observation = np.append(observations[1:], [new_transaction])
        if h.detect_fraud(observations, new_observation,
            THRESHOLD):
            print('Fraud')
        else:
            print('Normal')

if __name__ ==
    '_main_':
    d = Driver('./Data/train_data.txt')
    h = HMM(n_states=STATES,
        n_possible_observations=CLUSTERS)
    k = KMeansClustering()
    observations = k.run(d.get_data()[0:192])
    h.train_model(observations=list(observations), steps=STEPS)
    get_input()
```



A.3 Defining the structure of HMM

```
import numpy as np
from hidden_markov import hmm
def divide(num, denom):
    if num = 0:
        return 0
    return num / denom
HMM class HMM:
def _init_(self, n_states, n_possible_observations):
    Number of states self.n_states = n_states
    Number of possible observations
self.n_possible_observations = n_possible_observations
    Create states and possible observations
self.states, self.possible_observations = self._init_names()
    Create transition matrix, emission matrix and start probability matrix self.pi_prob,
self.transition_prob, self.emission_prob = self._init_probabilities()
```

Creating the HMM

```
self._model = hmm(states=list(self.states), observations=list(self.possible_observations),
start_prob=np.matrix(self.pi_prob), trans_prob=np.matrix(self.transition_prob), em_prob=np.matrix(s
Initialize states and possible observations
def _init_names(self):
    states = np.array(range(self.n_states))
    possible_observations = np.array(range(self.n_possible_observations))
    return states, possible_observations
```



Initialization of the probability of transition matrix and emission matrix

```
def _init_probabilities(self):
    pi_prob = np.zeros(self.n_states)
    transition_prob = np.zeros((self.n_states,
                                self.n_states))
    emission_prob = np.zeros((self.n_states,
                               self.n_possible_observations))
    for i in range(self.n_states):
        pi_prob[i] = 1 / self.n_states
    for i in range(self.n_states):
        for j in range(self.n_states):
            transition_prob[i][j] = 1 / self.n_states
    for i in range(self.n_states):
        for j in range(self.n_possible_observations):
            emission_prob[i][j] = 1 / self.n_possible_observations
    return pi_prob, transition_prob,
    emission_prob
```



A.4 The Hybrid Optimization Algorithm

Baum-Welch algorithm *def train_model(self, observations, steps):*

```
print('HMM is training ...')
pi_prob = np.zeros(self.n_states)
transition_prob = np.zeros((self.n_states,
                             self.n_states))
emission_prob = np.zeros((self.n_states,
                           self.n_possible_observations))
```

Calculation of Forward-Backward variables from the current observations

```
for _ in range(steps):
    fwd = self.forward_process(observations)
```

```
bwd = self.backward_process(observations)
```

Re-estimating of initial state probabilities

```
for i in range(self.n_states):
```

```
    pi_prob[i] = self.calculate_gamma(i, 0, fwd, bwd)
```

Re-estimating of transition probabilities

```
for i in range(self.n_states):
```

```
    for j in range(self.n_states):
```

```
        num, denom = 0, 0
```

```
        for t in range(len(observations)):
```

```
            num += self.calculate_path_probability(t, i, j, observations, fwd, bwd)
```

```
            denom += self.calculate_gamma(i, t, fwd, bwd)
```

```
        transition_prob[i][j] = divide(num, denom)
```

Re-estimating of emission probabilities

```
for i in range(self.n_states):
```

```
    for k in range(self.n_possible_observations):
```

```
        num, denom = 0, 0
```

```
        for t in range(len(observations)):
```

```
            g = self.calculate_gamma(i, t, fwd, bwd)
```

```
            if k == observations[t]:
```

```
                num += g
                denom += g
            emission_prob[i][k] = divide(num, denom)
```

```
self.pi_prob = pi_prob
```

```
self.transition_prob = transition_prob
```

```
self.emission_prob = emission_prob
```

Forward algorithm

Calculate Forward-Variables $fwd[i][t]$ for state i at time t for current observations

```
def forward_process(self, observations):
```

```
    fwd = np.zeros((self.n_states, len(observations)))
```

Initialization at time = 0

```
    for i in range(self.n_states):
```

```
        fwd[i][0] = self.pi_prob[i] *
```



```
self.emission_prob[i][observations[0]]
for t in range(len(observations) - 1):
    for j in range(self.n_states):
        fwd[j][t + 1] = 0
    for i in range(self.n_states):
        fwd[j][t + 1] += (fwd[i][t] * self.transition_prob[i][j])
    fwd[j][t + 1] = (fwd[j][t + 1] * self.emission_prob[j][observations[t + 1]])
return fwd
```

Backward algorithm

Calculate Backward-Variables $bwd[i][t]$ for state i at time t for current observations

```
def backward_process(self, observations):
```

```
    bwd = np.zeros((self.n_states, len(observations)))
```

Initialization at time = 0

```
    for i in range(self.n_states):
```

```
        bwd[i][len(observations) - 1] = 1
```

Calculate the probability of $P(x_t = s_i, x_{t+1} = s_j \mid \text{observations})$.

```
def calculate_path_probability(self, t, i, j, observations, fwd, bwd):
```

```
    num, denom = 0, 0
```

```
    if t == len(observations) - 1:
```

```
        num = fwd[i][t] * self.transition_prob[i][j]
```

```
    else:
```

```
        num = fwd[i][t] * self.transition_prob[i][j] * self.emission_prob[j][observations[t + 1]] *
```

```
        bwd[j][t + 1]
```

```
    for k in range(self.n_states):
```

```
        denom += (fwd[k][t] * bwd[k][t])
```

```
    return divide(num, denom)
```



Incorporating the Genetic Algorithm

calculate fitness of gen

```
def calc_fit(genn, targetn, target):  
    fit_n = 0  
    for j in range(target):  
        if gen[j:j+1] == targ[j:j+1]:  
            fit_n += 1  
    fit_n = fit_n / target * 100  
    return fit
```

create population

```
def create_pop(target_p,  
maximum_pop, target):  
    my_popul =  
    for i in range(maximum_pop):  
        gen = create_gen(target)  
        genfit_n = calc_fit(gen, target_p, target) my_popul[gen] = gen_fit  
    return my_popul
```

selection process

```
def select_n(my_popul):  
    p = dict(my_popul)  
    parent_sel =  
    for j in range(2):  
        g = max(pop, key=p.get)  
        gen_fit = p[g]  
        parent_sel[gen] = genfit_n  
    if j == 0:  
        del p[g]  
    return parent_sel
```



Crossover

```
def cross_O(parent_sel, target_p, target):  
    child_c =  
    ju = round(len(list(parent_sel)[0])/2)  
    for j in r(2):  
        gen_fit = list(parent_sel)[j][: ju] + list(parent_sel)[1-j][: ju:]  
        genfit_n = calculate_fitness(gen, target, target)  
        child_c[gen] = genfit_n  
    return child_c
```

mutation

```
def mut(child_c, target, mut_R, target):  
    mutant =  
    for j in range(len(child_c)):  
        data = list(list(child_c)[j])  
        for j in range(len(data)):  
            if up.random.rand(1) <= mut_R:  
                ch = chr(np.random.randint(32, 126))  
                data[j] = ch  
        g = ".join(data)  
        genfit_n = calc_fit (gen_fit, target_p, target)  
        mutant_f[gen_fit] = genfit_n  
    return mutant_f
```

Incorporating the PSO Algorithm

```
class PSO_Particle:  
    def _init_(self,x0):  
        s.p_j=[]  
        s.v=[]
```



```
s.p_best_j=[]  
s.error_best_j=-1  
s.error_j=-1  
for j in range(0,n_dim):  
s.v_i.append(rand.uniform(-1,1))  
s.p_i.append(x0[i])
```

evaluate current fitness

```
def calc(self,costFunction):  
s.error_j=costFunction(s.p_j)  
check to see if the current position is an individual best  
if s.error_j< s.error_best_j or  
s.error_best_j==-1:  
s.p_best_j=s.p_j  
s.error_best_j=s.error_j
```

Updating the velocity of a new particle

```
def upd_v(s,pos_b_g):  
w=0.5  
a1=1  
a2=2  
for i in r(0,n_dimen):  
b1=random.random()  
b2=random.random()  
v_cog=a1*b1*(s.p_best_j[j]-s.p_j[j])  
v_social=a2*b2*(pos_best_g[j]-s.p_j[j])  
s.velocity_j[j]=w*s.v_j[j]+vel_cog+v_s
```

Update the particle position based off new velocity updates

```
def upd_pos(s,bnds):  
for j in range(0,num_dim):
```



```
s.p_j[j]=s.p_j[j]+s.v_j[j]
```

adjust maximum position if necessary

```
if s.p_j[j]>bnds[j][1]:
```

```
s.p_j[j]=bnds[j][1]
```

Adjust minimum position if necessary

```
if s.p_j[j] < bnds[j][0]:
```

```
s.p_j[j]=bnds[j][0]
```

```
class PSO_Algo():
```

```
def _init_(s, cost_Function, x0, bnds, number_part, max_iter):
```

```
global num_dim
```

```
num_dim =len(x0)
```

```
error_best_g=-1
```

```
position_best_g=[]
```

establishing the swarm

```
swm=[]
```

```
for j in r(0, number_part):
```

```
swm.append(Particle(x0))
```

determine if current particle is the best (globally)

```
if swm[k].error_j< error_b_g or
```

```
error_b_g == -1:
```

```
pos_b_g=list(swm[k].p_j)
```

```
error_b_g=float(swm[k].error_j)
```

```
for k in r(0,number_part):
```

```
swm[k].update_v(p_best_g)
```

```
swm[k].update_pos(bnds)
```

```
j+=1
```

```
print('HMM has successfully trained.')
```



Calculate the probability of the occurrence of specific observation

```
def calculate_occurrence_probability(self, observations):  
    fwd = self.forward_process(observations)  
    bwd = self.backward_process(observations)  
    result = np.zeros(len(observations))  
    for i in range(len(observations)):  
        for j in range(self.n_states):  
            result[i] += fwd[j][i] * bwd[j][i]  
    return result
```

Fraud Detection

```
def detect_fraud(self, observations, new_observation, threshold):  
    print('Fraud evaluation ...')  
    alpha_1 =  
self.calculate_occurrence_probability(observations) alpha_2 =  
self.calculate_occurrence_probability(new_observation)  
    delta = alpha_1[0] - alpha_2[0]  
    delta = delta / alpha_1[0]  
    if delta >= threshold:  
        return True  
    else:  
        return False
```



A.5 Generating a Learning Curve for our models

```
Xt_Trn, Xt_Tst, yt_Trn, yt_Tst = train_test_split(X, y, test_size=0.2, stratify=yt, random_state=1)  
pipln = make_pipeline(StandardScaler(), hmm(penalty='l2', solver='lbfgs', random_state=1,  
max_iter=5000))  
Trn_sizes, Trn_scores, Tst_scores = learning_curve(estimator=pipeline, Xt=Xt_train, yt=yt_train,  
cv=10, Trn_sizes=np.linspace(0.15, 1.0, 12), n_jobs=1)  
Trn_mean = np.mean(Trn_scores, axis=1)
```

```
Trn_std = np.std(Trn_scores, axis=1)
Tst_mean = np.mean(Tst_scores, axis=1)
Tst_std = np.std(Tst_scores, axis=1)
plt.plot(Trn_sizes, Trn_mean, color='blue', marker='o', markersize=5, label='Accuracy
using Training Data')
plt.fill_between(Trn_sizes, Trn_mean + Trn_std, Trn_mean - Trn_std, alpha=0.15, color='blue')
plt.plot(Trn_sizes, Tst_mean, color='green', marker='+', markersize=5, linestyle='-', la-
bel='Accuracy using Validation Data')
plt.fill_between(Trn_sizes, Trn_mean + Tst_std, Tst_mean - Tst_std, alpha=0.10, color='red')
plt.title('Learning Curve of classifier')

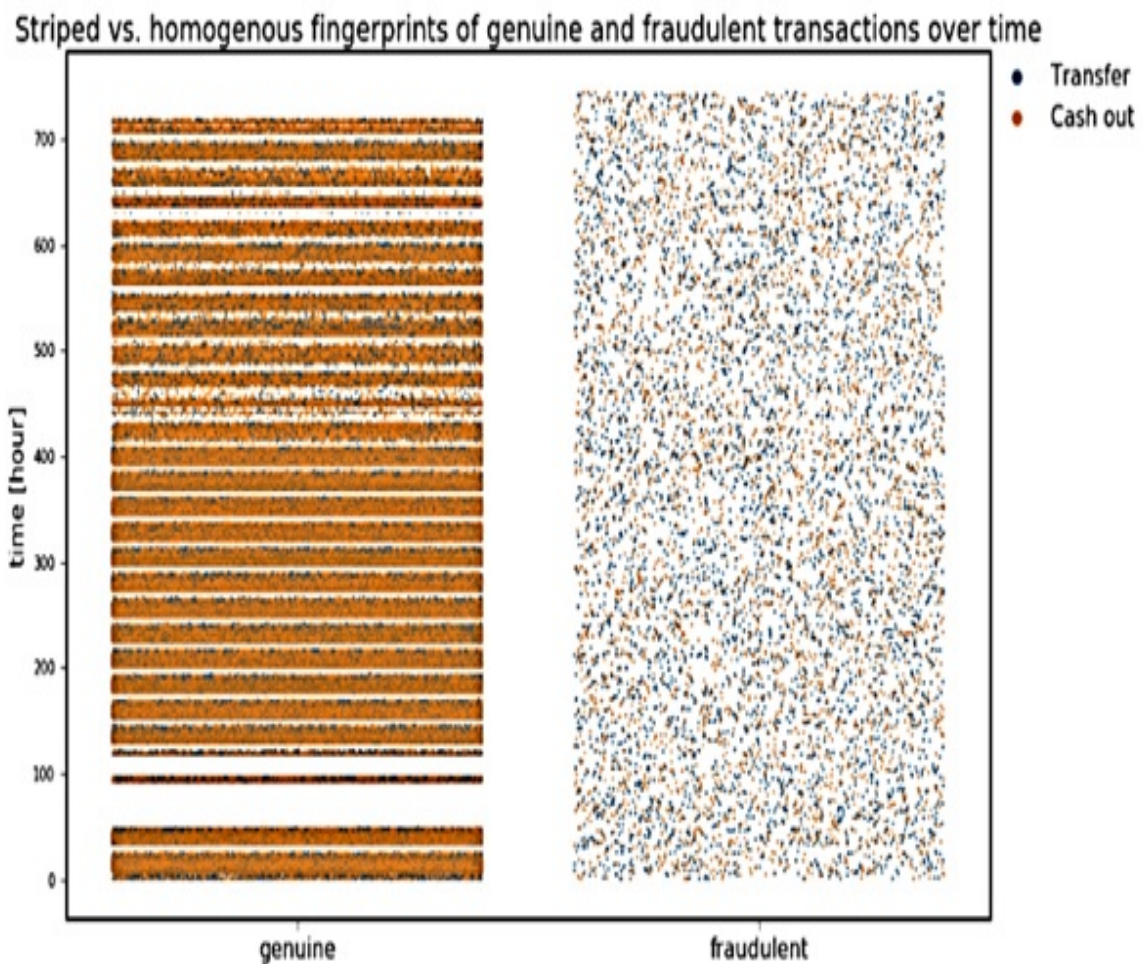
plt.xlabel('Size of Training Data')
plt.ylabel('Model Accuracy')
plt.grid()
plt.show()
```



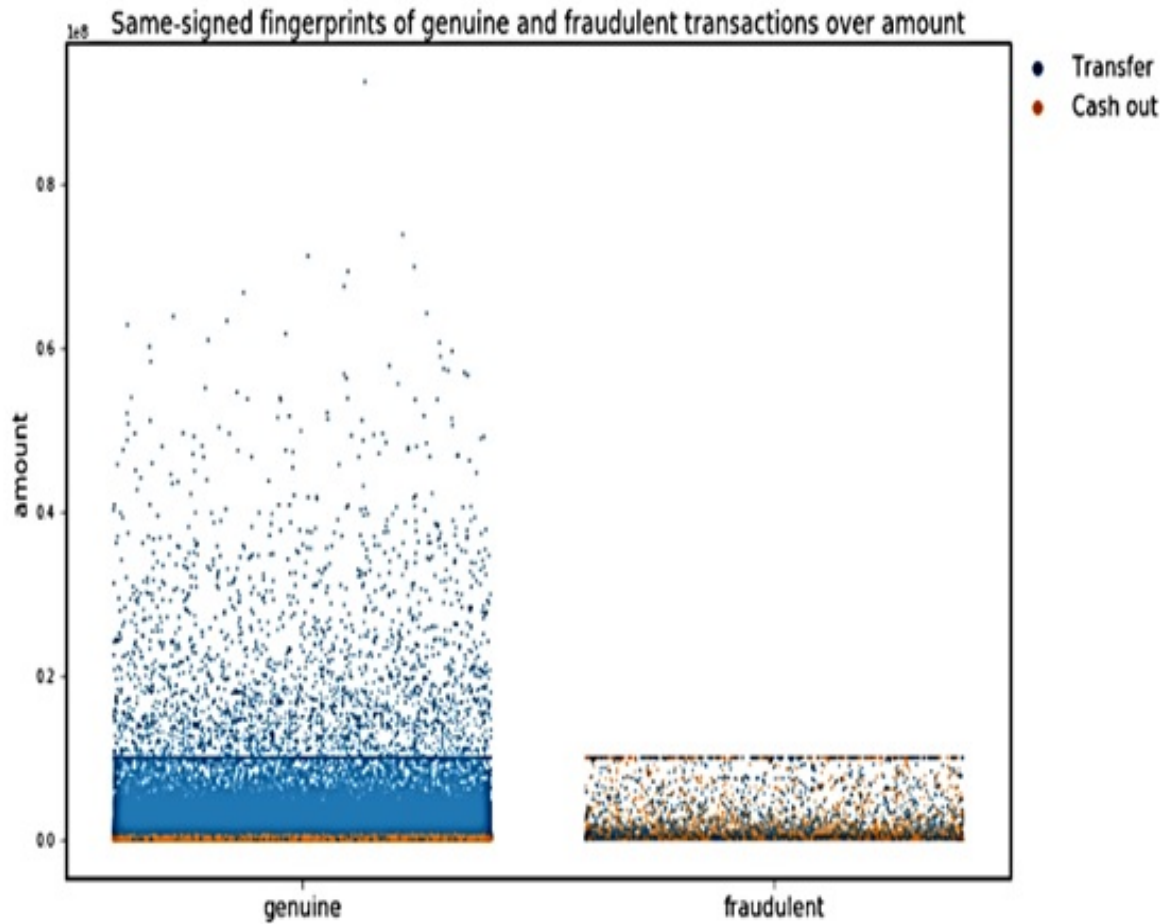
APPENDIX B

SAMPLE SIMULATION RESULTS

B.1 Fraudulent and genuine transactions Dispersion over time in our sample Dataset



B.2 Fraudulent and genuine transactions dispersed over amount in our sample Dataset



B.3 Values of $P(O|\lambda)$ for the various transaction profiles for various hidden number of states for BW, PSO, GA and BWPSO

	BW		GA		PSO		BWPSO	
	P(O λ)	Time(sec)	P(O λ)	Time(sec)	P(O λ)	Time(sec)	P(O λ)	Time(sec)
LL	0.667	0.325	0.731	0.830	0.705	0.280	0.783	0.369
LM	0.543	0.328	0.785	0.852	0.726	0.340	0.785	0.397
LH	0.354	0.354	0.370	0.895	0.453	0.330	0.555	0.449
ML	0.617	0.258	0.786	0.954	0.732	0.350	0.787	0.409
MM	0.476	0.427	0.675	0.859	0.564	0.300	0.882	0.398
MH	0.521	0.253	0.776	0.842	0.658	0.270	0.905	0.401
HL	0.368	0.321	0.563	0.852	0.633	0.320	0.695	0.349
HM	0.654	0.345	0.537	0.925	0.568	0.311	0.800	0.389
HH	0.563	0.321	0.678	0.875	0.650	0.290	0.803	0.419
Average	0.529	0.326	0.656	0.876	0.632	0.310	0.777	0.398



B.4 Values of $P(O|\lambda)$ for the various transaction profiles and various hidden number of states for BWGA, GAPSO, and the Proposed Algorithm

	BWGA		GAPSO		Proposed	
	$P(O \lambda)$	Time(sec)	$P(O \lambda)$	Time(sec)	$P(O \lambda)$	Time(sec)
LL	0.784	0.708	0.825	0.285	0.860	0.209
LM	0.806	0.762	0.823	0.515	0.873	0.529
LH	0.589	0.754	0.685	0.235	0.811	0.379
ML	0.789	0.788	0.860	0.575	0.988	0.449
MM	0.772	0.858	0.852	0.195	0.865	0.269
MH	0.856	0.808	0.865	0.255	0.913	0.349
HL	0.784	0.757	0.812	0.384	0.997	0.419
HM	0.765	0.728	0.777	0.485	0.984	0.400
HH	0.721	0.658	0.785	0.535	0.821	0.589
Average	0.763	0.758	0.809	0.385	0.901	0.399



B.5 Time taken to optimize the HMMs by various algorithm for all Transaction profiles

Trans Profile	BW	GA	PSO	BWPSO	BWGA	GAPSO	Proposed
LL	0.325	0.830	0.280	0.369	0.708	0.285	0.209
LM	0.328	0.852	0.340	0.397	0.762	0.515	0.529
LH	0.354	0.895	0.330	0.449	0.754	0.235	0.379
ML	0.258	0.954	0.350	0.409	0.788	0.575	0.449
MM	0.427	0.859	0.300	0.398	0.858	0.195	0.269
MH	0.253	0.842	0.270	0.401	0.808	0.255	0.349
HL	0.321	0.852	0.320	0.349	0.757	0.384	0.419
HM	0.345	0.925	0.311	0.389	0.728	0.485	0.400
HH	0.321	0.875	0.290	0.419	0.658	0.535	0.589
Average Time	0.326	0.876	0.310	0.398	0.758	0.385	0.399



B.6 Precision, Recall and F1-Scores of the Various Algorithms/Approaches for various Number of Hidden States

Precision				
	PROPOSED	HMM+KMEANS	HMM+DBSCAN	HMM+SMOTE+KMEANS
2	0.95	0.85	0.80	0.90
3	0.98	0.69	0.88	0.92
4	0.98	0.88	0.87	0.90
5	0.99	0.68	0.88	0.95
F1-Score				
	PROPOSED	HMM+KMEANS	HMM+DBSCAN	HMM+SMOTE+KMEANS
2	0.95	0.60	0.80	0.90
3	0.94	0.87	0.88	0.92
4	0.88	0.79	0.67	0.90
5	0.89	0.64	0.70	0.69
Recall				
	PROPOSED	HMM+KMEANS	HMM+DBSCAN	HMM+SMOTE+KMEANS
2	0.95	0.70	0.80	0.90
3	0.96	0.77	0.88	0.92
4	0.93	0.83	0.76	0.90
5	0.94	0.66	0.78	0.80



Publications and Conferences attended

List of publications

1. Abdul, A., Danaa, A., Daabo, M. I. (2021). An Improved Hybrid Algorithm for Optimizing the Parameters of Hidden Markov Models. 10(1), 63–73. Asian Journal of Research in Computer Science. doi: 10.9734/AJRCOS/2021/v10i130235
2. Abdul, A., Danaa, A., Daabo, M. I., Abdul-barik, A. (2021). Detecting Electronic Banking Fraud on Highly Imbalanced Data using Hidden Markov Models. 7(2), 315–332. doi: 10.34198/ejms.7221.315332 Earthline Journal of Mathematical Sciences.

Under Review

1. Reducing false positives in real-time Fraud detection using an improved Multi-Layer Hidden Markov Models (HMMS)

Conference presentation

1. Abdul, A., Danaa, A., Daabo, M. I., Abdul-barik, A. (2021). Detecting Electronic Banking Fraud on Highly Imbalanced Data using Hidden Markov Models. In: International Conference on Informatics and Mathematical Sciences (ICIMS-21) 28th July, 2021 in ACCRA, Ghana.

