A Novel Approach to Mobile Money Fraud Detection and Prediction Using Dynamic Locality Sensitive Hashing(DLSH)

A presentation by :

TEKOH PALMA ACHU

Registration Number: 19W2615

Under the supervision of :

Dr. DJAM Xaveria Youh KIMBI

Senior Lecturer , UYI

University of Yaounde I

Department of Computer science

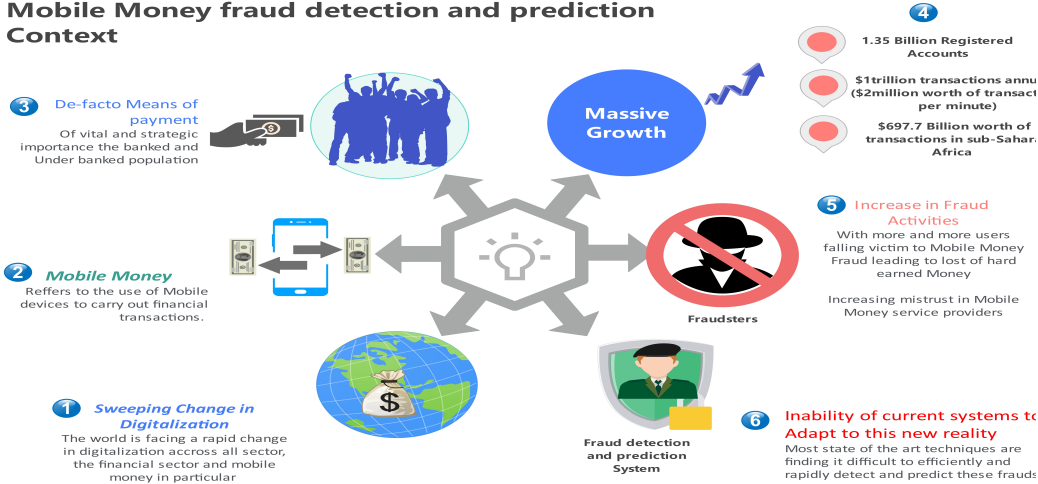
July 12, 2023

Presentation Roadmap

Presentation Plan

- Context
- Problem Statement
- Aim and Objectives
- Research Question
- Review of state of the art
- Research Methodology
- Results and Discussion
- Conclusion

Mobile Money fraud detection and prediction Context



Problem Statement

Problems faced by Mobile Money Users

- More and More fraudsters use Smishing mobile Money Messages to trick users to perform fraudulent transactions or reveal Pin code
- Users loose their phones which ends up in the hands of fraudsters who end up effectuating fraudulent transactions

Problems Statement

Problems related to state of the art Mobile Money fraud Detection and Prediction

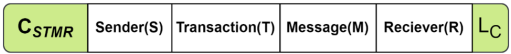
- use arbitrary threshold assignment mechanism to combat mobile money fraud which is highly inefficient.
- Are not Flexible and very costly to maintain and update[6]
- Are black boxes that are difficult to trace the rationale used for detection and prediction [7]
- Are Very expensive to build and don't scale well on massive amounts of dataset[8]
- Do a trade-off between efficiency and speed and don't perform high on both measures.

Illustrative Example: Mobile Money fraud detection and prediction Problem

Let:

- U represent a set of m Mobile Money users for a given service provider $U = \{u_1, u_2, \cdots, u_m\}$

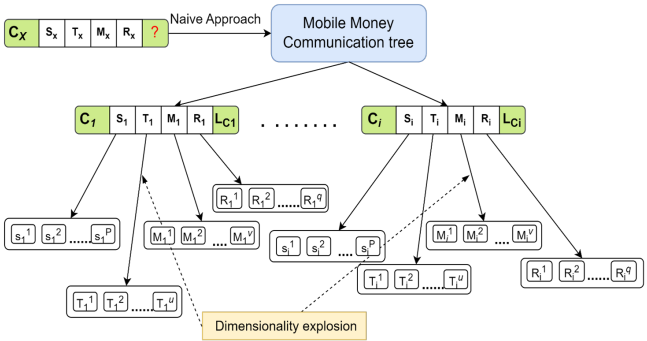
-The communication C_{STMR} between $u_x, u_y \in U$ be modeled as follows:



Where:

- C_i is the i^{th} communication
- S = Sender or initiator of C_{STMR} with attributes $\{s_i^1, s_i^2, \cdots, s_i^p\}$
- T = Mobile Money Transaction with attributes $\{T_i^1, T_i^2, \cdots, T_i^u\}$
- M = Mobile Money Message with attributes $\{M_i^1, M_i^2, \cdots, M_i^v\}$
- R = Receiver or Receptor of C_{STMR} with attributes $\{R_i^1, R_i^2, \cdots, R_i^q\}$
- L_c = label for C_{STMR} ie Fraudulent or Non-Fraudulent
- With $S, R \in U$

Illustrative Example: Mobile Money Communication tree



Dimensionality Explosion in Mobile Money Communication Tree

Given a communication tree of 100,000 communications(search space representation)

Questions

- How efficiently can we Identify similar transactions in our search space?
- How fast can we do the identification process?

Answers

- Need to perform exhaustive search throughout the entire search space to guarantee efficiency
- Perform fast pair-wise comparison of communication instance with inference communications

At what Cost?

Number of pair-wise Comparison $= \frac{100{,}000 \times 99{,}999}{2} \simeq 5,000,000,000$ operations

Standard PC:

- Performs 10,000 operations per second
- $\simeq 5.787 days$ to perform computation
- For a dataset of 5-dimensions like C_{STMR} it takes 28.9 days

Aim and Objective

Aim

The aim of this research work is to use dynamic locality sensitive hashing to efficiently and rapidly detect and predict mobile money fraud related to SMS and Mobile Money transactions.

Objectives

- Use dynamic locality sensitive hashing techniques to efficiently and rapidly detect and predict mobile money SMS fraud.
- Use dynamic Locality sensitive hashing technique to efficiently and rapidly detect and predict mobile money transaction fraud

Research Question

Main Research Question

RQ: How can a novel dynamic locality sensitive hashing approach be used to efficiently and rapidly detect and predict mobile money fraud related to SMS and transactions?

Sub Research Questions

- **RQ1:** How can we use a novel dynamic locality sensitive hashing techniques to efficiently and rapidly detect and predict mobile money SMS fraud?
- **RQ2:** How can a novel dynamic Locality sensitive hashing technique to efficiently and rapidly detect and predict mobile money fraud related to transactions?

Review of State of the art Mobile Money Prediction and detection Systems

Approach and Author	Problematic	Methodology	Limitations
Mobile Money SMS Fraud Detection: Josh et al [2021]	Detection of Fraud in Mobile Money SMS	Application of SVM, logistic regression, Multinomial naive bayes classifier, Random Forest and CNN while focusing on different feature extraction techniques (TF-IDF, count vectorizer, Tokenizer, Transformer) and their impact on the performance of machine learning model	- Fail to address Mobile Money transaction Fraud - Did not use state-of-the-art Semantic encoding scheme such as Google USE. - Computationally costly to train models
Efficiency of SVM, GBDT and Naive Bayes algorithms in Mobile Money Transaction fraud detection: Botchey et al [2020]	Determining the efficiency of SVM, GBDT and Naive Bayes algorithms in Mobile Money Transaction fraud detection	Apply two (2) under sampling (RandomUnderSampler, NearMiss) and two (2) oversampling (RandomOverSampler, SMOTE) technique and finally a hybrid of both sampling techniques on the imbalance dataset Evaluate the performance of SVM, GBDT and Na"ive Bayes algorithms in MMT fraud detection while evaluating the impact of Sampling techniques on the imbalance data-set.	- Does Not address SMS fraud - Model serves as a black box - Data intensive and very costly to train
Improved Case based reasoning(CBR): Adedoyin et al [2017]	Identification of Mobile Transfer fraud in Mobile Money Environment.	Improved CBR approach by combining features to single dimension vector then using Genetic algorithm to compute optimal feature weights then using KNN for classification of MMT into normal or fraudulentt	- computationally very expensive and does not scale well on massive amounts of data. - Dataset used to evaluate this approach was very small(2000)

Review of State of the Art Technique

Mobile Money SMS Fraud Detection: Josh et al [2021]

Problematic: Detection of Fraud in Mobile Money SMS
Methodology: Application of SVM, logistic regression, Multinomial naive bayes classifier, Random Forest and CNN while focusing on different feature extraction techniques (TF-IDF, count vectorizer, Tokenizer, Transformer) and their impact on the performance of machine learning model

Limitations

- Fail to address Mobile Money transaction Fraud
- Did not use state-of-the-art Semantic encoding scheme such as Google USE.
- Computationally costly to train models

Efficiency of SVM, GBDT and Naïve Bayes algorithms in Mobile Money Transaction fraud detection. Botchey et al [2020]

Problematic: Determining the efficiency of SVM, GBDT and Naïve Bayes algorithms in Mobile Money Transaction fraud detection

Methodology: - Apply two (2) under sampling (RandomUnderSampler, NearMiss) and two (2) oversampling (RandomOverSampler, SMOTE) technique and finally a hybrid of both sampling techniques on the imbalance dataset Evaluate the performance of SVM, GBDT and Naïve Bayes algorithms in MMT fraud detection while evaluating the impact of Sampling techniques on the imbalance data-set.

Limitations

- Does Not address SMS fraud
- Model serves as a black box
- Data intensive and very costly to train

Review of State of the Art Technique

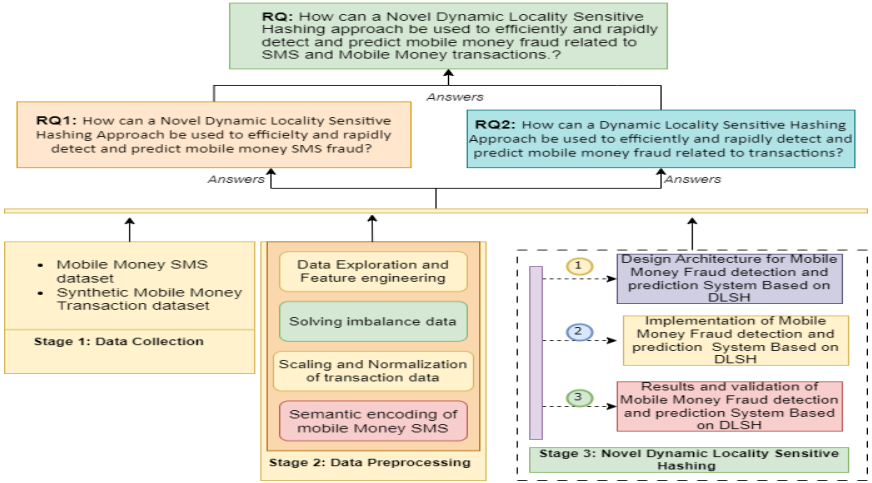
Improved Case based reasoning(CBR): Adedoyin et al [2017]

Problematic: Identification of Mobile Transfer fraud in Mobile Money Environment.
Methodology: Improved CBR approach by combining features to single dimension vector then using Genetic algorithm to compute optimal feature weights then using KNN for classification of MMT into normal or fraudulent

Limitations

- computationally very expensive and does not scale well on massive amounts of data
- Dataset used to evaluate this approach was very small(2000)

Overview of Our Methodology(DLSH)



Stage-1: Data Collection

Collection of Mobile Money Transaction datatset

1. Synthetic data from Paysim simulator as proposed by Lopez et al [3].
2. Data was generated using a real-world data obtained from a mobile money operator operating in over 14 African countries.

Collection of Mobile Money SMS datatset

1. Spam collection dataset Tiago et al [29]
2. 100 Mobile Money Spam SMS gotten from colleagues, friends and family.

Stage-2: Data Preprocessing

Preprocessing of Mobile Money Transaction datatset

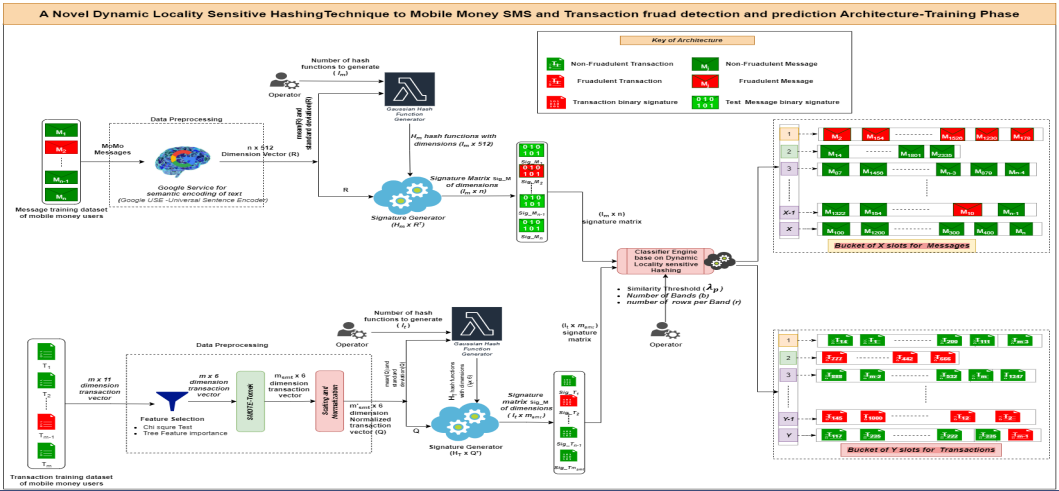
1. **Data exploration:** To gain better understanding of data representation and elimination of data anomalies and null values.
2. **Feature Engineering:** Sklearn's ExtraTreesClassifier feature importance measure and Chi Square score to eliminate features that don't contribute to prediction
3. **Balancing of Data(SMOTE-Tomek):** Applied SMOTE to increase minority class and Tomek to eliminate noise introduced by SMOTE
4. **Min-Max Normalization:** to reduce the range of data points and hence speed up training and testing phase

Stage-2: Data Preprocessing

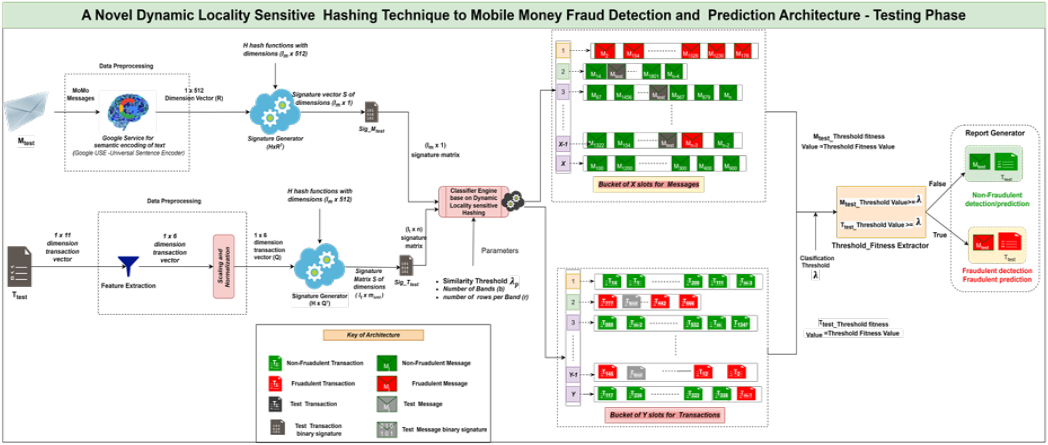
Preprocessing of Mobile Money SMS datatset

1. **Data exploration:** To gain better understanding of data representation and elimination of data anomalies and null values.
2. **Semantic Encoding:** To capture and represent the Semantic meaning of Mobile Money messages in form that a computer can understand we used Google USE(Universal Sentence Encoder).

Stage 3: Dyanamic Locality Sensitive Hashing (DLSH) -Training Architecture



Stage 3: Dyanamic Locality Sensitive Hashing (DLSH) -Testing Architecture



Dynamic Locality Sensitive Hashing (DLSH) Algorithms

Algorithm *Generate_Signature*

Input: *An array* ***D*** *of transactions or messages, An array* ***H*** *of hash functions.*
Output: ***Sig_D*** *a 2D binary signature representation for* ***D***

```
1:  Sig_D ← empty array list [[]]
2:  i←0
3:  for i=0 to length(D) do
4:      for j=0 to length(H) do
5:          temp← D[i]*H[j]
6:          if temp ≥ 0 do
7:              temp ← 1
8:          Else
9:              temp ← 0
10:         end if
11:         Sig_D[i] ← temp
12:         j ← j+1
13:     end for
14:     i ← i+1
15:   end for
16  return Sig_D
```

Complexity of ***Generate_Signature: O(|D|*|H|)***

Algorithm: *DLSH_Bucketing*

Input: ***Sig_D*** a 2D binary signature representation for ***D,*** *number of buckets* ***b*** *and number of rows per bucket* ***r***
Output: ***Returns a bucket***

```
1:  Sig_ID ← 1
2:  bucket←empty harsh_table{}
3:  Comment: assert that the signature can be divisible into b bands
4:  Assert |Sig_D| % b==0
5:  for i=0 to 2^r-1 do
6:  Comment: Creating empty list and assigning to various bucket keys
7:      bucket[i] ←new_empty_list[ ]
8:      i ← i+1
9:  endfor
10: for i=1 to length(Sig_D) do
11:         Comment: |Sig_D| is the number of signatures in the dataset
12:         for j=1 to length (Sig_D[i]) do
13:             hash ←BinaryToInteger (Sig_D [ j : j + r])
14:             bucket[hash] ← Sig_ID
15:             j ← j + r
16:         end for
17:         Sig_ID ← Sig_ID +1
18:     i ← i+1
19: end for
20: return bucket
```

Complexity of *DLSH_Bucketing:* ***O(2^r) + O(|Sig_D| * (r*b)) = O(|Sig_D| * (r*b))***

Dynamic Locality Sensitive Hashing (DLSH) Algorithms

Algorithm:	***DLSH_Query***
Input Parameters:	***q*** *data to query* ***HT*** *hash table of messages/transactions* ***H*** *hash functions used in signature generation algorithm* ***b*** *number of bands* ***r*** *number of rows per band* λ_p *similarity threshold* λ classification threshold
Output:	Returns ***1*** *if* ***q*** *is fraudulent and* ***0*** *otherwise*

```
1.  q_keys ← empty hash Table{ }, candidates ← empty array list[ ]
2.  candidateLabels← empty hash Table{ }
3.  Comment: generating signature for q
4.  Sig_q ←Generate_signature (q, H))
5.  q_keys .append(Sig_q))
6.  Assert |Sig_q| % b==0
7.  for i=1 to |Sig_q| do
8.        Comment: Convert a band of the signature into integer and store it in q_keys
9.        q_keys.append(BinaryToInteger(Sig_q[i : i + r]))
10.       i ← i+(r+1)
11. end for
12.  for key in q_keys do
13.       Comment:  using the keys extract candidates from hash table HT
14.       candidates.append (HT[key])
15.  end for
16.  for candidate in candidates do
17.       candidateLabels.append(extractLabelFromCandidate(candidate))
18.  end for
19. Threshold Fitness value ←Average(candidateLabels)
20. if Threshold Fitness value ≥ λ
21.       return 1
22. Else
23.       return 0
24. end if
```

complexity: ***O(n)*** *where* ***n*** *is* the total number of transactions or messages stored in the hash bucket.

Mathematical Model for Query

1) Let x be a sample message or transaction we want to know if its fraudulent or not
2) Its a 2-Stage Process: Threshold Fitness Value and Threshold Fitness Extractor calculations

Stage 1:Threshold Fitness Value

- $C(x) = Q(x, HT)$
- Threshold Fitness value $= \sum_1^n \frac{L_{cx}}{n_{cx}}$

Stage 2:Threshold Fitness Extractor

We define a threshold value λ such that:

$$ThresholdFitnessExtractor = \begin{cases} 1 & \text{if Threshold Fitness value} \geq \lambda \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where:

- C(x) are the candidates of transaction/message x
- HT is a hash table of messages or transactions
- Q is the Query function
- n_{cx} is the number of candidates of x.
- L_{cx} are the labels for C(x) ,$L_{cx} \in [0, 1]$

Experiment Setup and Parameter configuration

Dynamic Locality Sensitive Hashing Experiment Parameter Configuration							
Mobile Money SMS				Mobile Money Transaction			
Total		5,750		Total		600,000	
Test Dataset	232	Training Dataset	5,175	Test Dataset	120,000	Training Dataset	480k
Similarity Threshold(λ_p)	Band (b)	Rows (r)		Similarity Threshold(λ_p)	Bands (b)	Rows (r)	
1.0	1	512		1.0	1	1113	
0.997	2	256		0.997	3	371	
0.989	4	128		0.987	7	159	
0.968	8	64		0.944	21	53	
0.917	16	32		0.827	53	21	
0.805	32	16		0.485	159	7	
0.5946	64	8		0.1391	371	3	
0.2973	128	4		0.00089	1113	1	
0.0625	256	2					
0.00195	512	1					
Number of hash functions for SMS (l_m)		1					
Number of hash functions for Transaction (l_t)		1		Classification Threshold (λ)			

Table: Experiment setup and parameter configuration

Review of State of the art Mobile Money Prediction and detection Systems

Mobile Money SMS fraud detection and prediction

ID	Message (M)	Encoded M	signature M	λ_p	Bucket ID(s)	Candidate ID(s)	Threshold fitness value	Fitness Extraction	Label

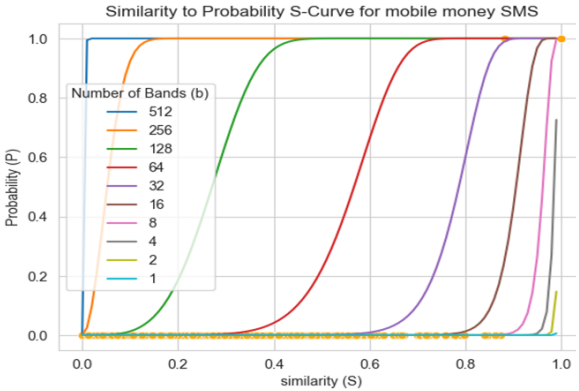
Table: Review of State of the Art Technique

Mobile Money Transaction fraud detection and prediction

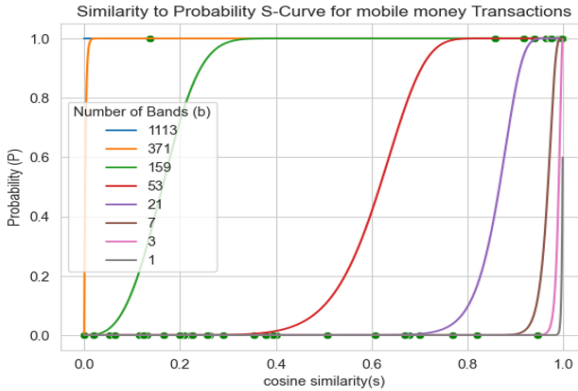
ID	Message (M)	Encoded M	signature M	λ_p	Bucket ID(s)	Candidate ID(s)	Threshold fitness value	Fitness Extraction	Label

Table: Review of State of the Art Technique

RQ1: How can we use a novel dynamic locality sensitive hashing techniques to efficiently detect and predict mobile money SMS fraud?

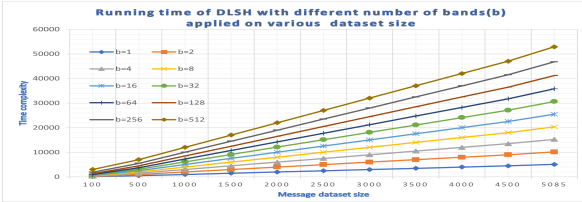


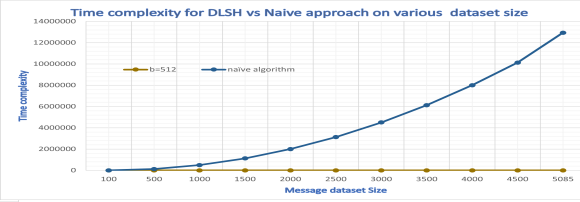
575 Mobile Money Messages

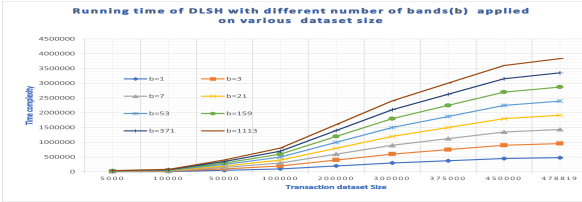


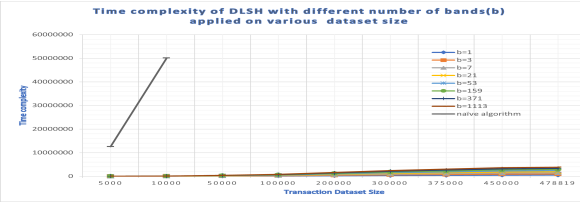
1200 Mobile Money transactions

RQ2: How can we use a novel dynamic locality sensitive hashing techniques to Rapidly detect and predict mobile money SMS fraud?









DLSH Comparison with State of the Art

Approach	Efficiency	Speed	Scalability	Detection and prediction ability	Computational cost
Dynamic Locality Sensitive hashing (our approach)	- SMS: 98.6 % -Transaction: 99%	Very Fast	YES	- SMS -Transaction	Low
Case based reasoning [8]	98% prediction efficiency	Fast	NO	Transaction	Very High
Cross-case analysis Approach [7]	- Support Vector Machine: 99.91% - Naïve Bayes algorithm: 99.65% - Gradient boosted Decision tree: 89.9%	Fast	YES	Transaction	High
Mobile money SMS fraud detection [23]	99.82%	Fast	YES	SMS	High

Table: DLSH comparison with State of the Art

Tools and Technologies used

Integrated Development Environment

- Data spell
- Visual studio code

Platforms

- Kaggle
- Github

Libraries

(1) Numpy
(2) Pandas
(3) Sci-Kit Learn
(4) Google Universal Sentence Encoder
(5) Seaborn
(6) Matplotlip
(7) Xgboost

Contribution

i) We Introduced a Novel Approach to Mobile money fraud detection and prediction using Dynamic Locality sensitive Hashing

ii) Approach could tackle both SMS and Transaction Fraud

iii) Approach was efficient on SMS (98.6%) and Transaction(99%) datasets

Difficulties

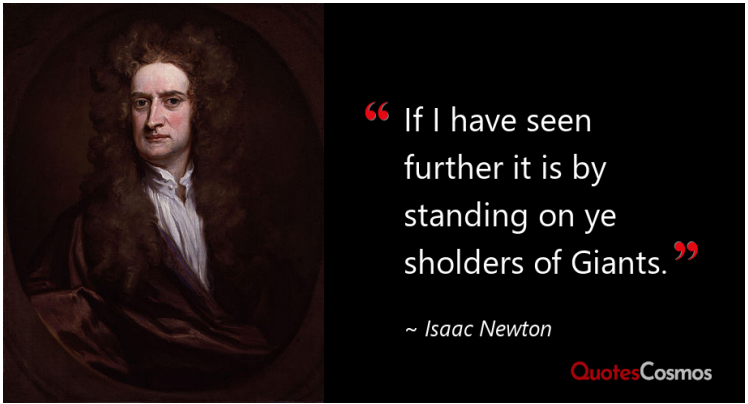
1. Difficulty in obtaining real world company data for the research work.
2. Difficulty in gaining access to the servers of a real mobile money service provider in order to test the approach

Recommendations for future studies

- Explore means of apply DLSH to taggling Mobile fraud related to fraudulent calls
- Extend DLSH to tackle process related fraud such as money laundry within the mobile money sector
- Implement the proposed architecture on a distributed computing model such as Map-Reduce which will drastically increase the speed
- Extend DLSH to related financial domains such as bank transactions.
- Adapt DLSH to handle Mobile Money Fraud involving SIM Swap

References

- [6] R. Rieke, M. Zhdanova, J. Repp, R. Giot, and C. Gaber, "Fraud Detection in Mobile Payments Utilizing Process Behavior Analysis," 2013. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00841002
- [7] F. E. Botchey, Z. Qin, and K. Hughes-Lartey, "Mobile money fraud prediction-A cross-case analysis on the efficiency of support vector machines, gradient boosted decision trees, and Naïve Bayes algorithms," Information (Switzerland), vol. 11, no. 8, Aug. 2020, doi: 10.3390/INFO11080383.
- [8] A. Adedoyin, S. Kapetanakis, G. Samakovitis, and M. Petridis, "Predicting fraud in mobile money transfer using case-based reasoning," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10630 LNAI, pp. 325–337, 2017, doi: 10.1007/978-3-319-71078-5_28
- [23] J. Nkoy and A. Mohammed, "Mobile Money SMS Fraud Detection," Stanford CS 229 Final Project Report. transactions. (2021)



Thanks for Your Kind Attention