Anti-Money Laundry and Financial Crime

Name: Arun Narayanaswamy
ID: 001122220
Course Leader: Dr. George Samakovitis
Date of submission: 26/04/2021

School of Computing and Mathematical Sciences

Contents

Introduction: 3
Overview of the model: 4
Description of AML services: 6
Data Requirements and Data Security: 8
Architecture plan and Tools involved 9
Lab Portfolio 11
Lab 1: 11
Lab 2: 12
Lab3 14
References 15

Figure table

Figure 1 Suspicious activity report in UK 3
Figure 2 Flow Chart by Financial institutes 4
Figure 3 Transaction graph 5
Figure 4 Parallel Topology 6
Figure 5 Star Topology 6
Figure 6 Complete reticular formation 6
Figure 7 Efficient subgraph isomorphism algorithm 7
Figure 8 System Architecture 8
Figure 9 Closed triangle 9
Figure 10 Identifying fraud transaction 11
Figure 11 Identifying origin of transaction 12
Figure 12 Fraud Transactions 12
Figure 13 Balancing the datasets 13
Figure 14 One-hot encoding 13
Figure 15 Confusion matrix 14
Figure 16 Important feature of a model 14

Introduction:

"Money laundering is generally understood as the concealment of an illegitimate source of assets, providing an apparent legal origin." (Lecture slides)
Money laundering is the illegal process of making huge money generated from criminal activity, such drugs, terrorist funding, which looks like it comes from legitimate source. The money from the criminal activity is considered as dirty, hence the name "laundry".

Money launders would try to move the dirty money into financial system and later receive it in from of legitimate money and spent it on expensive art items, luxury cars and real estate.

All the Money launders follow 3 steps before making the dirty money into legitimate money. (Lecture Slides)

1. Placement: The process of placing the dirty money into financial ecosystem is called placement. (Lecture slides)
2. Layering: Concealing the money through series of transaction. (Lecture slides)
3. Integration: Launderer would receive money from the legitimate account which can be used for any activities. (Lecture slides)

The different forms of money laundry are smurfing, currency exchange, bitcoins, tax evasion, capital flight etc. The most common form of Money laundry is smurfing. Smurfing is a process where large chunk of cash would be divided into multiple small deposits and spread into different accounts, to avoid detection. Today far more sophisticated criminal enterprises conceal money movements through online instruments, digital payments, and dizzying numbers of global transactions. (Murphy,A., Meyer,A)

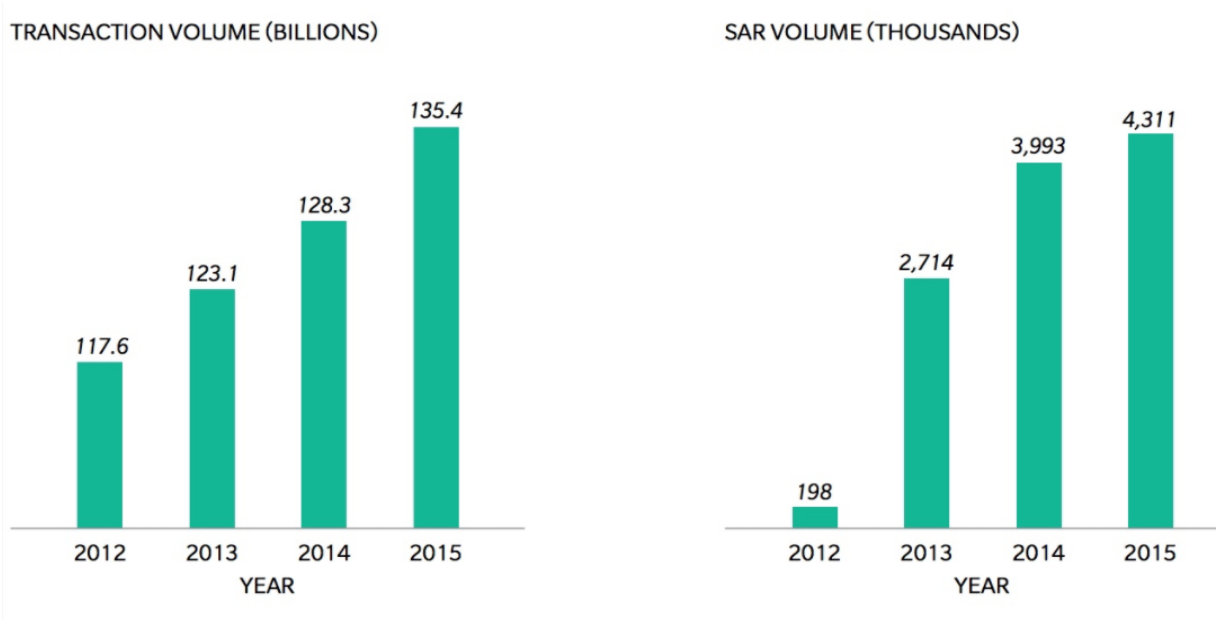


Figure 1 Suspicious activity report in UK

The above figure indicates the increase of 2,000 percent in just five years, in part because banks have gotten better at ferreting out illicit transactions. Every year it is evident that smurfing is increasing and traditional methods are unable to stop it. Hence, adapting advanced analytical tools which are familiar in many areas of risk management, but financial institutions have only recently started using them to detect money laundering. These

statistical techniques can reduce both false negatives and false positives—in some cases, false positives have fallen by 50 percent or more—and they help institutions respond rapidly to emerging threats. (Murphy,A., Meyer,A)

To stop money laundering, Financial systems need to adopt more strategic, end-to-end processes that take advantage of recent innovations in data analysis like conducting back ground check on the consumers, verifying the given documents, KYC, analysing the transaction patterns and the relationships between the transactions. Most important, "they must replace the familiar check-the-box compliance mind-set in favour of full-on engagement." (He *et al., 2021)*

Overview of the model:

Flow Chart:

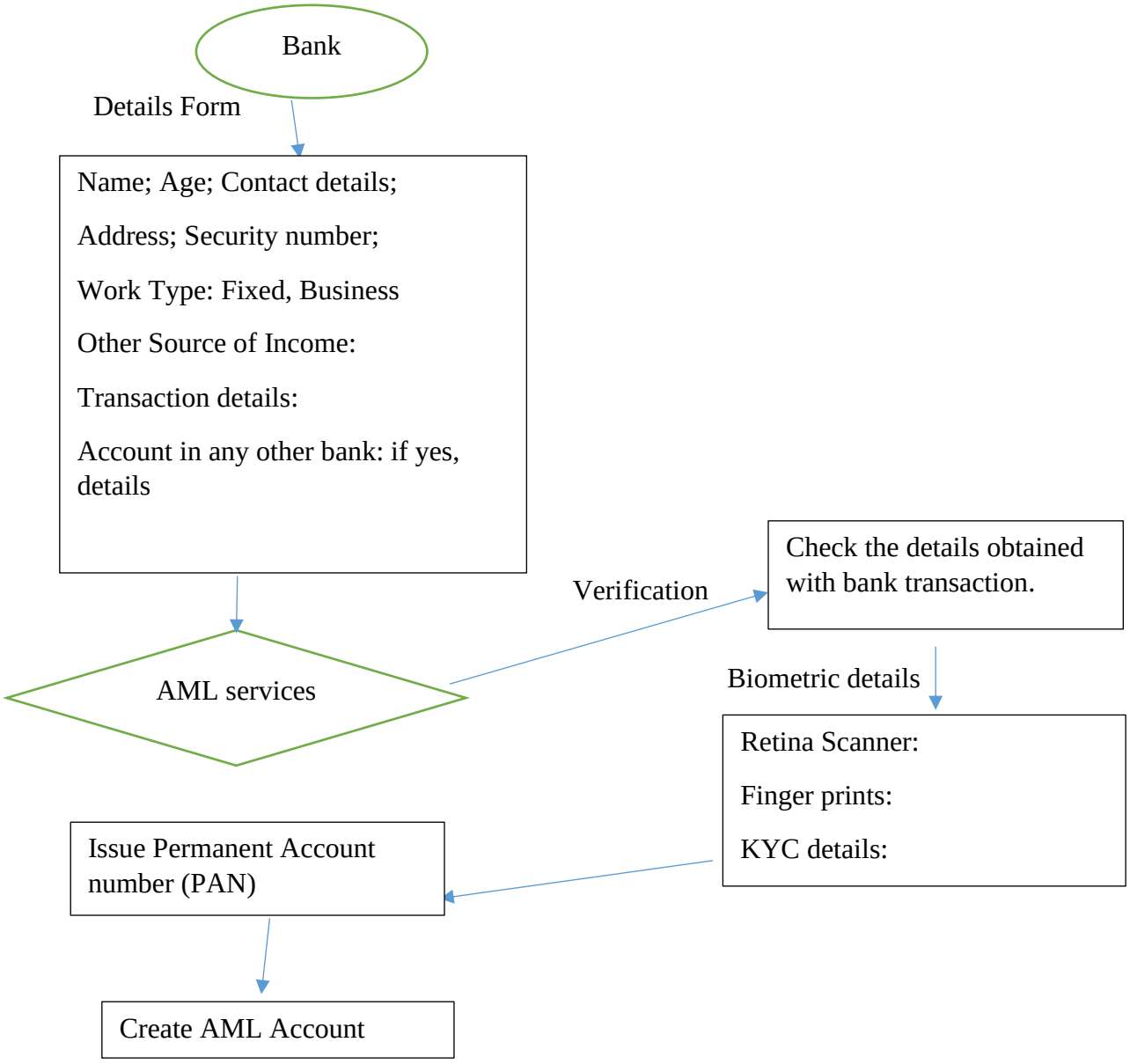


Figure 2 Flow Chart by Financial institutes

Placement of the money by a launderer is the place where they are most vulnerable we can deploy our AML service to analyse the pattern, check the details and source of money. An Anti-Money laundry software can be designed efficiently by obtaining the background details of a particular customer from the registered bank. All the banks need to provide requested details mentioned in Details form if they would need services in identifying money laundry. (Lecture slides)

The above flow chart shows the AML service would require few details from banks. Bank would need to provide details of the customer like Account number, nature of job, age, annual income and any other business details (all these details would be just to apply regressions in understanding the flow and expenses of money by an individual). The provided details need to be verified before giving the biometric access. Biometric access would be a key for transactions and can be used for security purposes.

Permanent account number (PAN) these number generated will be unique and issued to the users who are verified from AML services. PAN would be linked to all the accounts if the user has multiple accounts and can be used to understand the spending patterns using cluster algorithms.

In AML detection following the traditional methods like sorting, merging, and manually analysing the accounts is not possible hence we will be adapting and extracting the data which is only required for transaction analysis and setting up a threshold for a particular amount. Based on the variability of suspicious transaction behaviours, an efficient and systematic method is needed to simplify transaction behaviours among related suspicious entities. (He *et al., 2021)*

The data would be simplified and categorised into blocks before modelling the data, the proposed graph-based method can fully reveal the financial relationship between linked accounts (PAN), thereby facilitating the capital chain analysis of the accounts. This method attempts to transform the relationship between the accounts into path relationships corresponding to topological diagrams from a topological perspective (He *et al., 2021)*

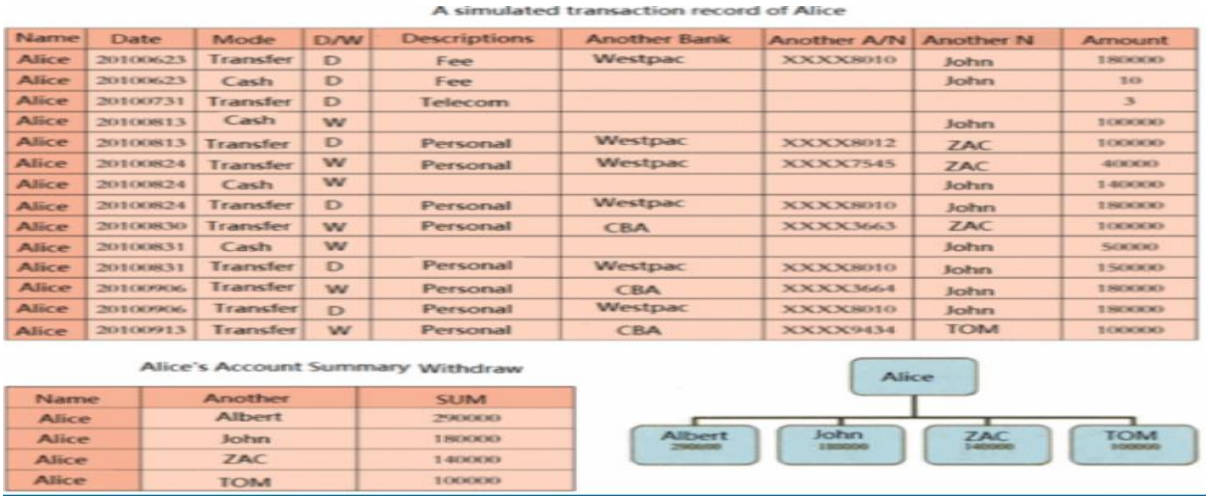


Figure 3 Transaction graph (He et al., 2021).

The AML model will flag the transactions which clearly show smurfing, Alice makes the total transfer of 1800000 to John by splitting the money into multiple transactions (Transfer, Cash) and the same applies for the transaction of Albert and Zac. (He *et al., 2021)*

Description of AML services:

The AML service remains in the centre where the details are received from the bank, based on which the smurfing is detected.

"Smurfing structure is another primary transaction type that always happens in ML. Original sender with a considerable amount of money tries to evade scrutiny from reporting suspicious behaviour by splitting it into smaller transfers." (He *et al., 2021)*

There are different types of smurfing namely Parallel Structure, Star Topology, Incomplete Reticular Formation, Cycle Structure, Complete Reticular Formation and Tree Topology. Each smurfing has different characters of parallel structure are its completion, irreversibility, and no loop. The direction of the flow is visible from start to end point. It would be easy to detect the parallel structure money laundry as the result can be predicted and analysed based on the previous happened steps. (He *et al., 2021)*

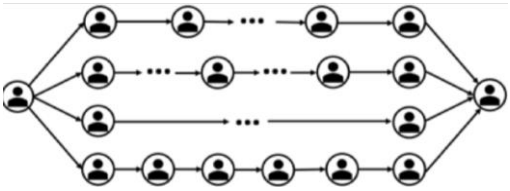


Figure 4 Parallel Topology

In Start topology the important character of transaction is the edge members. The Edge members are the core account to complete the transaction. Hence, the core customer is vital to track criminal behaviours and estimate the degree of crime. (He *et al., 2021)*

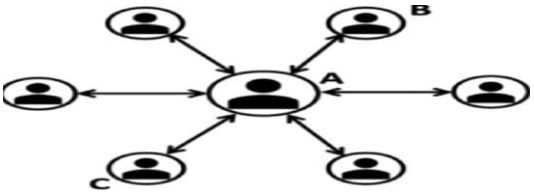


Figure 5 Star Topology

In complete reticular formation the customer can have many accounts. As shown in the figure the dirty money enters the financial system from A1, then the accounts money can be deposited into accounts (B1,B2,…) randomly to other commercial activity accounts (C1,C2,…),…, (E1,E2,…). (He *et al., 2021)*

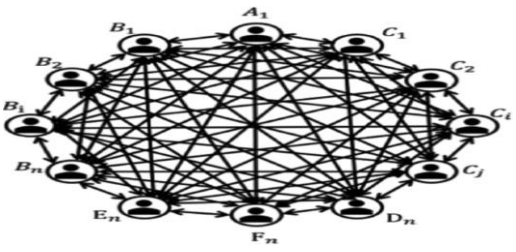


Figure 6 Complete reticular formation

From all the above topologies we can infer that smurfing is hard to detect if more accounts are involved. The AML service will adapt in detect smurfing containing complex topologies.

We are applying Efficient Subgraph Isomorphism Solution and Distant Measurement to solve smurfing. (He *et al., 2021)*

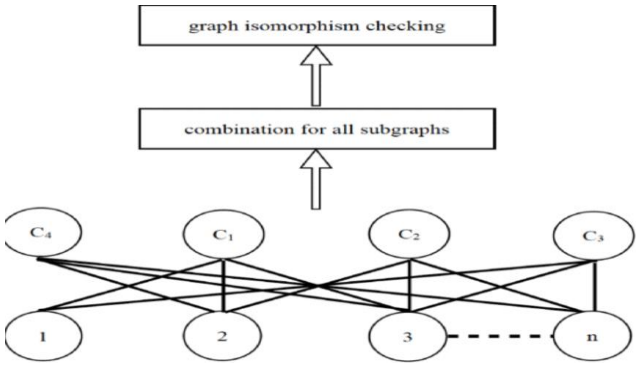


Figure 7 Efficient subgraph isomorphism algorithm

To calculate the combination of the graphs all the nodes needs to be connected. The combination of three vertices according to the graph will be tested. If there exist more graphs, then they are called isomorphic. If the number are same to the closed triangle, then there is a relationship between the coefficient and the closed triangle. If the subgraph has the same number of closed triangle. The second order of the combination needs to be checked. If the existence is proved, then use the O(n6) theorem based on permutation to test whether the generated subgraph is isomorphic with the original graph. *(He et al., 2021)*

The adjacent matrix of the vertices and edges will be calculated (If the number of vertices is n4 and the space complexity is n in this step). The sum of each row of the matrices, which could produce the degree sequence. (The spatial complexity is n2 and n4, respectively, in this step). Determine whether the degree sequence of the vertex and edge adjacency matrix is a permutation relationship. If not, the two graphs are not isomorphic. (The spatial complexity is n6 in this step). The distance will be computed based on permutation theorem and equinumerosity theorem. *(He et al., 2021)*

"Applying Subgraph isomorphism is more efficient and effective to solve ML problems with a full understanding of the trading network graph's characteristics. Moreover, references to the subgraph isomorphism, and a combination of traditional search rules can significantly reduce the labour force. The limitation of monitoring system can be overcome by using advanced subgraph isomorphism techniques and models by correlating various Events as a comprehensive network detection." *(He et al., 2021)*

After receiving the output mentioned we can check the original transactions made by the users in the given bank, comparing both the result it would be easy to find out the accounts used for smurfing and can be blocked. The genuine transaction can be considered and issue a PAN number. Where these numbers would be the next reference where for each transaction the number would be carried.

All the historical data would be saved in Hbase using the PAN it would be easily monitored. Using this number, we can apply linear regression to predict the next transactions which would help in minimizing smurfing.

Data Requirements and Data Security:

Data mining is a process to extract knowledge from existing data. It is used as a tool in banking and finance, in general, to discover useful information from the operational and historical data to enable better decision-making. Banks use data mining in various application areas like marketing, fraud detection, risk management, money laundering detection and investment banking. Detecting activities related to money laundering is necessary and inevitable for the economy, industries, banks and financial institutions. The main aim is to review the field of fraud detection with an emphasis on detecting money laundering and examine deficiencies based on data mining techniques. For this we will be adapting ETL process. (Ahmad, Fathian)

ETL indicates Extract, Transform and load. First step is to extract the data from the banks storages like ERP, CRM, FLAT. For extracting the data and proving security our AML services will be using Multi keyword Ranked Keyword Search over Encrypted cloud data(MRSE). In this architecture the data privacy is ensured to be encrypted before outsourcing so as to provide end-to-end data confidentiality assurance in the cloud and beyond. (Narasimha, Latha)

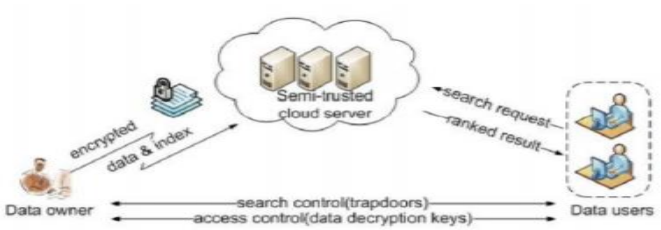


Figure 8 System Architecture

The cloud encryption follows 3 steps namely Cloud setup, Cryptography cloud storage, Vector model. (Narasimha, Latha)

The data from the bank needs to be push the data into the cloud sever. When the data is outsourced into cloud, the cloud service providers are able to control and monitor the data and the communication between users. Hence the bank data needs to be secured by Crypto graphing in the cloud Storage. The data contains sensitive information's like Account number, Transaction code etc., the cloud servers cannot be fully entrusted in protecting data. For this reason, outsourced files must be encrypted. Any kind of information leakage that would affect data privacy are regarded as unacceptable. Later the AML services would request for keys to the encrypted file. This can be communicated via emails or phone to provide access for the file. (Narasimha, Latha)

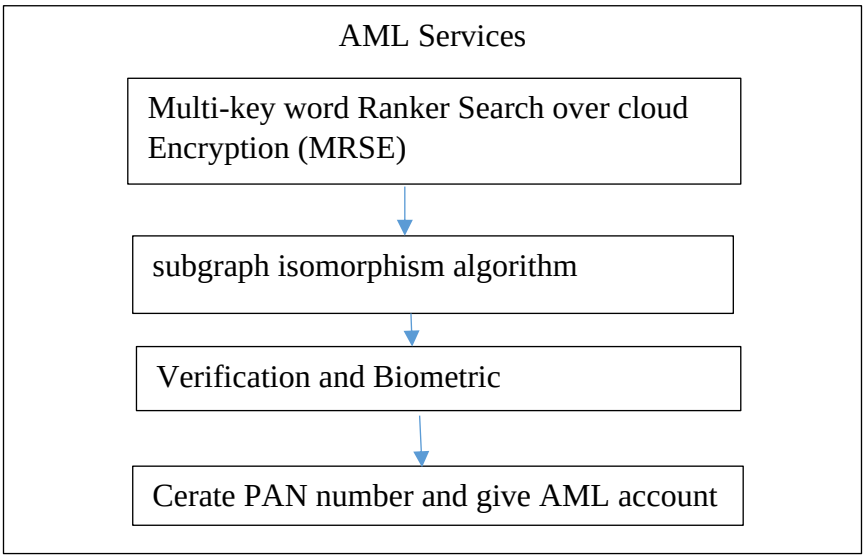
The Vector Model is used as a series of searchable symmetric encryption schemes have been enable search on cipher text. In the former, files are ranked only by the number of retrieved keywords, which impairs search accuracy. (Narasimha, Latha)

This is an important step where the data needs to be handled safely and taken care of because the data contains Account number, SWIFT code and transaction details like (date, currency, country code). These data would be considered and would be transformed into required format by checking the details and values. The transaction code pays an important role in analysing the patterns. The transaction codes will be broken down in understanding the

patterns. In AML services the country codes, bank codes, branch code, location code will be predefined in our database. These codes will be matched and a transformed report will be obtained.

The next stage is to divide the data into Remitter account and Beneficiary account and the mode of transaction made, and the duration. This table would provide the relationship between the accounts in understanding the credit and debit of the accounts. Both the tables will be combined and will be passed as input to the subgraph isomorphic algorithm. After getting the output from AML further analysis like Statistics, Data integration, Information science, Machine learning, and Visualization can be carried out.

Architecture plan and Tools involved



The AML services follows 4steps in identifying smurfing.

Step1: The services will be using MRSE for extracting the data from Banks to AML storage. This is applied to ensure data security. When the data is extracted the pre-processing of data will be carried out and the data which is required to fit into the algorithm will be considered like date, mode, transaction details, amount credited and debited.

Step2: Subgraph isomorphic algorithm an extra node is added to the original graph, so that there are no orphan nodes present. This would ensure that all the graph stays connected and there will not exist any isolate nodes. It is necessary to prove that if the number of closed triangles is the same as well as the number of vertex and degree is the same.

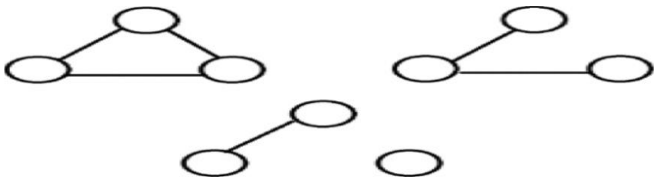


Figure 9 Closed triangle

From the above figure the closed triangle refers to where all the three vertexes are connected is called a closed three-tuple group.

Step3: Verification is a process where the obtained output from the algorithm will be compared with the original transactions. This would be possible by making use of Neo4j a graphical data representation, Neo4j makes it possible to understand the transactions made by making use of nodes and vertices. The output from isomorphic algorithm makes use of nodes and vertices and closed triangles which is similar to the representation from Neo4j.

The flagged users shall be reported to banks and the account can be suspended example Alice from figure3.

Step4: If the transaction seems genuine then the biometric details will be collected and PAN number should be linked to the account by the banks so that all the transactions will be monitored through PAN by our AML services.

Uses of linking PAN to account number.

When a transaction is made from an account through the bank the same will be recorded in by the PAN number. Using this unique number our AML service can compare the previous transactions with the new transactions. The similarities can be recorded batten the transactions. Later a classification algorithm will be implemented in exploring the hidden patterns.

Based on the historical data collected a simple linear regression shall be applied in analysing the spending patterns based on which a values will be predicted and saved in separate column. If the spending rate increases the predicted values, then it would be easy to track the source from where the value would be exceeded.

The obtained data can be visualised using python. Python is a powerful tool which help in proving the summary of the users. Boxplot is one such example where the differentiation can be made from lower hinge, upper hinge and median. If there any outlayers are present even after running isomorphic algorithm and verifying with Noe4j then it would be easy to find out the transactions corresponding to the details.

Lab Portfolio

Lab 1: Credit card fraud scenario

In this lab we will make use graph databases Neo4j to discover the criminal who stole credit card details.

There are 3 steps in identifying fraud transactions.

1. Identify fraudulent transactions.
2. Identify the point of origin of the scam.
3. Zero in on the criminal.

The below code help us get all the fraudulent transactions made by different customers, at which store, money spent and the date of transac

```
[ ] print('Step 3: Collect all the illicit transactions', end = '\r')
    illicitTxs = '''
          MATCH (victim:Customer)-[r:HAS_BOUGHT_AT]->(merchant)
          WHERE r.status = "Disputed"
          RETURN victim.name AS `Customer Name`, merchant.name AS `Store Name`, r.amount AS Amount, r.time AS `Transaction Time`
          ORDER BY `Transaction Time` DESC
    '''
    output_illicitTxs = neo4jGraph.run(illicitTxs).to_data_frame()
    print('Step 3: Collect all the fraudulent transactions. Successfull!', '\r')
    print('\n', output_illicitTxs, '\n')
    input('Press ENTER to continue.')
    print()
```

```
Step 3: Collect all the fraudulent transactions. Successfull!

    Customer Name         Store Name Amount Transaction Time
0          Olivia  Urban Outfitters   1152        8/10/2014
1          Olivia        RadioShack   1884         8/1/2014
2            Paul       Apple Store   1021        7/18/2014
3            Marc       Apple Store   1914        7/18/2014
4          Olivia       Apple Store   1149        7/18/2014
5        Madison        Apple Store   1925        7/18/2014
6        Madison  Urban Outfitters   1374        7/10/2014
7        Madison        RadioShack   1368         7/1/2014
8            Paul  Urban Outfitters   1732        5/10/2014
9            Marc  Urban Outfitters   1424        5/10/2014
10           Paul        RadioShack   1415         4/1/2014
11           Marc        RadioShack   1721         4/1/2014
12           Paul             Macys   1849       12/20/2014
13           Marc             Macys   1003       12/20/2014
14         Olivia             Macys   1790       12/20/2014
15       Madison             Macys   1816       12/20/2014
```

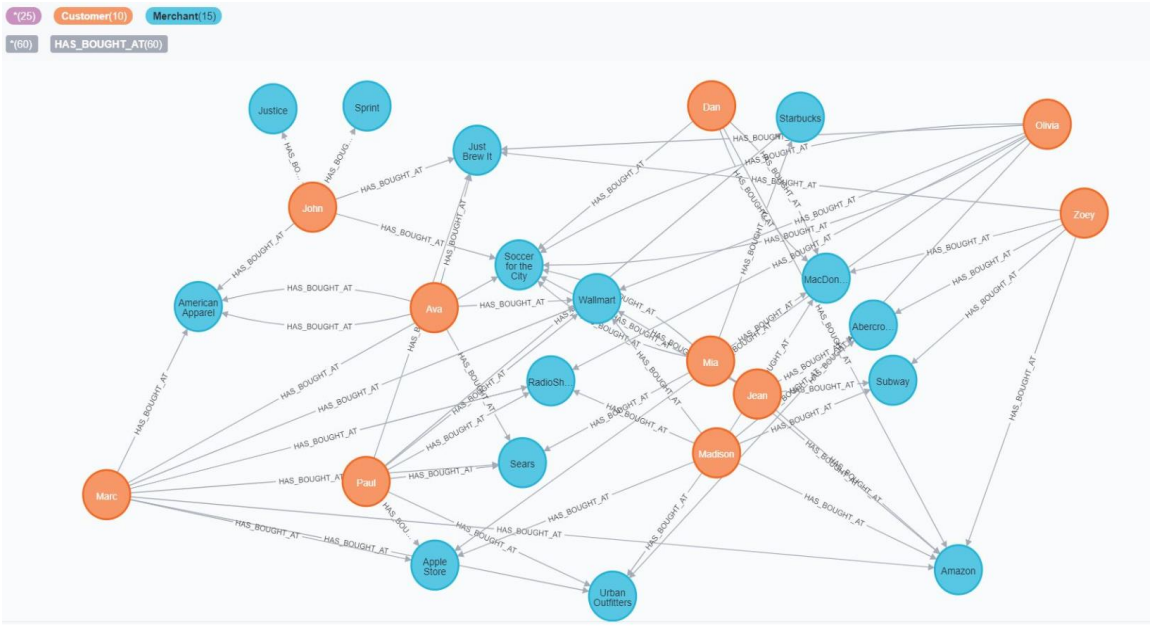


Figure 10 Identifying fraud transaction

The below code helps in getting the point of origin of the fraud transactions. Hence we will be collecting all the transactions before the fraud transactions.

```
print('Step 4: Identify the Point of Origin of the Fraud', end = '\r')
pointOfOrigin = '''
        MATCH (victim:Customer)-[r:HAS_BOUGHT_AT]->(merchant)
        WHERE r.status = "Disputed"
        MATCH (victim)-[t:HAS_BOUGHT_AT]->(othermerchants)
        WHERE t.status = "Undisputed" AND t.time < r.time
        WITH victim, othermerchants, t ORDER BY t.time DESC
        RETURN victim.name AS `Customer Name`, othermerchants.name AS `Store Name`, t.amount AS Amount, t.time AS `Transaction Time`
        ORDER BY `Transaction Time` DESC
...

output_pointOfOrigin = neo4jGraph.run(pointOfOrigin).to_data_frame()
print('Step 4: Identify the Point of Origin of the Fraud. Successfull!')
print('\n', output_pointOfOrigin, '\n')
input('Press ENTER to continue.')
print()
```

```
Step 4: Identify the Point of Origin of the Fraud. Successfull!

    Customer Name            Store Name Amount Transaction Time
0          Olivia               Wallmart    231       7/12/2014
1          Olivia               Wallmart    231       7/12/2014
2          Olivia               Wallmart    231       7/12/2014
3         Madison               Wallmart     91       6/29/2014
4         Madison               Wallmart     91       6/29/2014
5         Madison               Wallmart     91       6/29/2014
6            Paul              Starbucks    239       5/15/2014
7            Marc       American Apparel    336        4/3/2014
8            Marc       American Apparel    336        4/3/2014
9            Paul           Just Brew It    986       4/17/2014
10           Paul           Just Brew It    986       4/17/2014
11           Marc                 Amazon    134       4/14/2014
12           Marc                 Amazon    134       4/14/2014
13           Paul                  Sears    475       3/28/2014
14           Paul                  Sears    475       3/28/2014
15           Paul                  Sears    475       3/28/2014
16           Marc               Wallmart    964       3/22/2014
17           Marc               Wallmart    964       3/22/2014
18           Marc               Wallmart    964       3/22/2014
19           Paul               Wallmart    654       3/20/2014
20           Paul               Wallmart    654       3/20/2014
21           Paul               Wallmart    654       3/20/2014
22        Madison                 Subway    352      12/16/2014
23        Madison                 Subway    352      12/16/2014
24        Madison                 Subway    352      12/16/2014
25        Madison                 Subway    352      12/16/2014
26        Madison             MacDonalds    630       10/6/2014
27        Madison             MacDonalds    630       10/6/2014
28        Madison             MacDonalds    630       10/6/2014
29        Madison             MacDonalds    630       10/6/2014
30         Olivia  Soccer for the City    924       10/4/2014
31         Olivia  Soccer for the City    924       10/4/2014
32         Olivia  Soccer for the City    924       10/4/2014
33         Olivia  Soccer for the City    924       10/4/2014
```

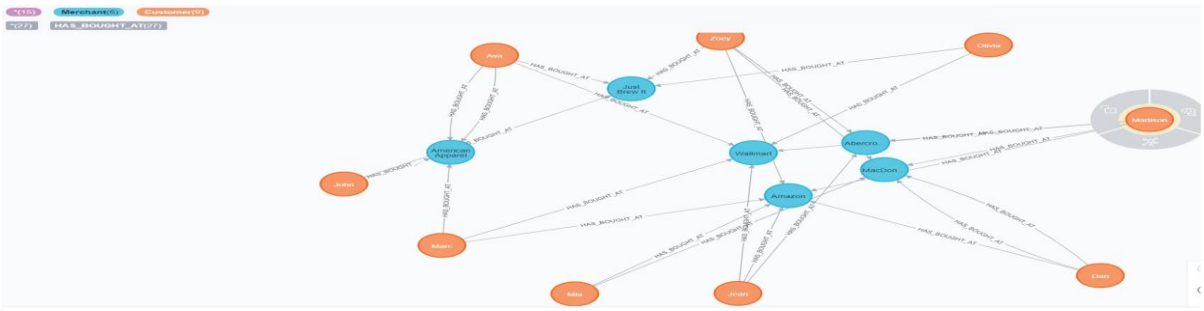


Figure 11 Identifying origin of transaction

The below code narrows down the fraud transactions. Walmart is the common denominator in both the transactions. Hence we can infer the credit card information was stolen in Walmart.

```
[ ] print('Step 5: Zero in the criminal.', end = '\r')

    zeroInTheCriminal = '''
            MATCH (victim:Customer)-[r:HAS_BOUGHT_AT]->(merchant)
            WHERE r.status = "Disputed"
            MATCH (victim)-[t:HAS_BOUGHT_AT]->(othermerchants)
            WHERE t.status = "Undisputed" AND t.time < r.time
            WITH victim, othermerchants, t ORDER BY t.time DESC
            RETURN DISTINCT othermerchants.name AS `Suspicious Store`, count(DISTINCT t) AS Count, collect(DISTINCT victim.name) AS Victims
            ORDER BY Count DESC
            '''
    output_zeroInTheCriminal = neo4jGraph.run(zeroInTheCriminal).to_data_frame()
    print('Step 5: Zero in the criminal. Successfull!')
    print('\n', output_zeroInTheCriminal, '\n')
    input('Press ENTER to continue.')
    print()
```

```
Step 5: Zero in the criminal. Successfull!

      Suspicious Store  Count                    Victims
0            Wallmart      4  [Olivia, Madison, Marc, Paul]
1           Starbucks      1                      [Paul]
2    American Apparel      1                      [Marc]
3       Just Brew It       1                      [Paul]
4              Amazon      1                      [Marc]
5               Sears      1                      [Paul]
6             Subway       1                   [Madison]
7          MacDonalds      1                   [Madison]
8  Soccer for the City      1                    [Olivia]
```

Figure 12 Fraud Transactions

Lab 2:

Classifying the fraudulent transaction using Decision tree classifier.

First the data needs to be imported into google Colab and needs be checked for blank Colum fields and replace it with null value or mean, median or mode. The received data has no blank columns hence continues to the next stage pre-processing.

Pre-processing is a stage where the data will be symmetrically organised into fraudulent and non-fraudulent data sets. If pre-processing is not executed, then the model training would be difficult and the test data would overfit.

In the given data there is a huge difference in between fraudulent and non-fraudulent data because of this data sampling is essential. After applying Random Under-sampling, the data sets of both the columns contains equal elements.

```
[ ]  balanced_df= pd.concat([sampled_non_Fraud,Fraud]).reset_index(drop=True)
     print(balanced_df.head(10))
     balanced_df["isFraud"].value_counts().plot(kind="bar")
```

```
   step      type      amount  ... oldbalanceDest  newbalanceDest  isFraud
0    11   PAYMENT    25712.41  ...            0.00            0.00        0
1    11  TRANSFER   680682.25  ...      1648820.18      2329502.42        0
2    13  CASH_OUT   156356.05  ...            0.00       156356.05        0
3    12   CASH_IN   457955.68  ...            0.00            0.00        0
4    10  CASH_OUT    28855.83  ...       209484.38       238340.21        0
5    10   CASH_IN   232019.55  ...      4774170.68      4768870.41        0
6     8  TRANSFER    58450.79  ...            0.00       484819.58        0
7    13   PAYMENT     2861.92  ...            0.00            0.00        0
8    10  CASH_OUT   294773.85  ...       408573.27      1548908.93        0
9    11   CASH_IN    44882.25  ...       468358.61       423476.36        0

[10 rows x 10 columns]
<matplotlib.axes._subplots.AxesSubplot at 0x7fee92ef3810>
```

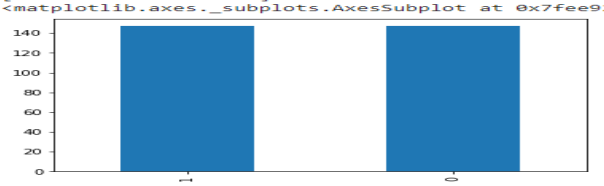


Figure 13 Balancing the datasets

To check the correlation between the elements we use Heat maps and subplots

In training the models there should not be any categorical column so converting the type column into numeric column by using one-hot encoding.

```
[ ]  balanced_df["type"].unique()
     balanced_df['type']=balanced_df['type'].replace(['TRANSFER', 'CASH_OUT', 'PAYMENT', 'CASH_IN', 'DEBIT'],['0','1','2','3','4'])

     dummy1=pd.get_dummies(balanced_df["type"])
     master=pd.concat([balanced_df,dummy1],axis=1)
     print(master)
```

```
     step type      amount     nameOrig  oldbalanceOrg  ...  isFraud  0  1  2  3
0      11    2    25712.41  C1660456119       21127.00  ...        0  0  0  1  0
1      11    0   680682.25  C2073838008       91141.62  ...        0  1  0  0  0
2      13    1   156356.05  C1581879975       52539.00  ...        0  0  1  0  0
3      12    3   457955.68    C43935542      884784.38  ...        0  0  0  0  1
4      10    1    28855.83   C218879330      465975.85  ...        0  0  1  0  0
..    ...  ...         ...          ...            ...  ...      ... .. .. .. ..
289    13    1      408.00  C1894004688         408.00  ...        1  0  1  0  0
290    13    0    48375.02   C920803432       48375.02  ...        1  1  0  0  0
291    13    1    48375.02  C1894578299       48375.02  ...        1  0  1  0  0
292    13    0  4022667.54   C735463888     4022667.54  ...        1  1  0  0  0
293    13    1  4022667.54    C79951219     4022667.54  ...        1  0  1  0  0

[294 rows x 14 columns]
```

Figure 14 One-hot encoding

For training and testing the decision tree the columns which are required needs to be considered, and the rest of the column needs to be dropped. From importing sklearn the test train will be carried out.

The performance and summary of the model can be checked by using Confusion matrix.

```
[ ]  from sklearn import metrics
     print(metrics.classification_report(y_test, y_pred))
     metrics.plot_confusion_matrix(dt_clf, X_test, y_test)
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        41
           1       1.00      1.00      1.00        48

    accuracy                           1.00        89
   macro avg       1.00      1.00      1.00        89
weighted avg       1.00      1.00      1.00        89

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7fee8d23ef50>
```

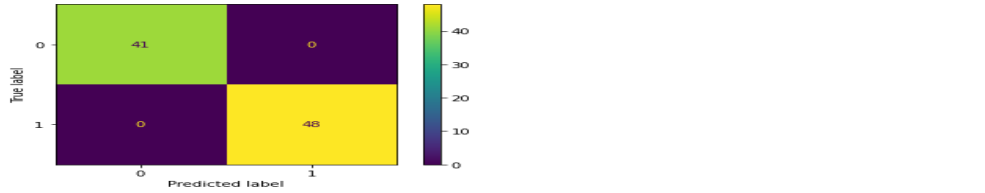


Figure 15 Confusion matrix

From the above confusion Metrix, we can generate precision, recall and accuracy is at 100% the model has been trained well and is able to predict the datasets accurately.

The column “isFraud” is considered to be important column as the training of decision tree was made on it. So it is considered to be important column.

5	isFraud	1.0

Figure 16 Important feature of a model

Lab3

Classifying the fraudulent transaction using SVM.

The data extraction and data pre-processing will remain the same as previous Lab2.

In training and testing the data instead of decision tree we will be making use of SVM.

A.2: Train an SVM classifier on the provided dataset

```
[ ]  from sklearn.metrics import accuracy_score
     from sklearn.tree import plot_tree
     from sklearn.svm import SVC
     from sklearn.preprocessing import StandardScaler
     from sklearn.svm import SVC
     from sklearn import svm
```

```
[ ]  trans=trans.drop(columns=['step','type','nameOrig','nameDest'],axis=0)
     X=trans
```

```
[ ]  y=trans['isFraud']
     trans= trans.drop(columns=['isFraud'])
```

```
[ ]  from sklearn.model_selection import train_test_split
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)
     print(X_train.shape)
     print(y_train.shape)
     print(X_test.shape)
     print(y_test.shape)
```

```
[ ]  sc=StandardScaler()
     sc.fit(X_train)
     X_train_std=sc.transform(X_train)
     X_test_std=sc.transform(X_test)
```

```
[ ]  svm=SVC(kernel='linear',random_state=1)
     svm.fit(X_train_std,y_train)
```

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
    max_iter=-1, probability=False, random_state=1, shrinking=True, tol=0.001,
    verbose=False)
```

```
[ ]  y_pred=svm.predict(X_test_std)
     print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     59958
           1       1.00      1.00      1.00        42

    accuracy                           1.00     60000
   macro avg       1.00      1.00      1.00     60000
weighted avg       1.00      1.00      1.00     60000
```

we can generate precision, recall and accuracy is at 100% the model has been trained well and is able to predict the datasets accurately. Both the decision tree and SVM gives a good result in accuracy.

References

Ahmad,S., Fathian3,M(2017) Title*:" Data Mining Techniques for Anti Money Laundering*", Available at : https://www.ripublication.com/ijaer17/ijaerv12n20_120.pdf Accessed on:24/04/2021

He *et al.J (2021) Title: "An Efficient Solution to Detect Common Topologies in Money Launderings Based on Coupling and Connection*" Availed at : https://ieeexplore.ieee.org/abstract/document/9352484 Accessed on: 22/04/2021

Murphy,A., Meyer,A (2017) Title: "*Cleaning Up Money Laundering*" Available at : https://www.brinknews.com/cleaning-up-money-laundering/. Accessed on: 20/04/2021

This is an online service for AML

Narasimha,K., Swarna,T Title:" *Secured Multi-Keyword Ranked Search over Encrypted Cloud Data*", Available at: https://www.arcjournals.org/pdfs/ijrscse/v2-i7/2.pdf Accessed on: 23/04/2021

Many reference as from lecture slides and videos, mentioned as (lecture slides)

Thank you note:

I sincerely thank Dr. George Samakovitis for teach me this course and for proving your feedback to improve my work on Anti Money Laundry and Financial Crime. I feel confident in understanding the difference between fraud and legitimate transactions. The subject has provided me good insights and will help in applying to real world transactions.