University of Tartu

Faculty of Science and Technology

Institute of Mathematics and Statistics

Ismayil Aghahasanli

Detecting money laundering in transaction monitoring using hidden Markov model

Actuarial and Financial Engineering

Master's Thesis (30 ECTS)

Supervisor: Kaur Lumiste, PhD

Tartu 2021

Detecting money laundering in transaction monitoring using hidden Markov model

Master's thesis

Ismayil Aghahasanli

Abstract. The purpose of the thesis is to introduce, build and test HMM as a method of detecting suspicious financial transactions that might be correlated with money laundering. HMM is a statistical Markov model in which the system being modelled is assumed to be Markov process with unobserved (i.e., hidden) states. These hidden states however generate observable outcomes. HMM fits the context of transaction monitoring in the fight against money laundering as the intent of a transaction (part of money laundering scheme or not) is and only some parameters of the transaction can be observed. The model was built and tested on artificial datasets provided by Salv Technologies and commonly used k-means clustering model was chosen for comparison. Analysis and testing showed that overall, HMM outperforms k-means clustering. Based on analysis, it can be concluded that in essence, HMM can be used in transaction monitoring but getting high precision needs expert knowledge and practical testing. A brief overview of money laundering, anomaly detection methods and HMM are given. Empirical part includes application of HMM on 3 different study cases using R software.

CERCS research specialization: P160 Statistics, operations research, programming, actuarial mathematics.

Keywords: hidden Markov model, HMM, money laundering, anomaly detection.

Rahapesu tuvastamine finantstehingute seires varjatud Markovi mudeli abil

Magistritöö

Ismayil Aghahasanli

Lühikokkuvõte. Lõputöö eesmärk on tutvustada, ehitada ja testida varjatud Markovi mudelit (*hidden Markov* model - HMM) kui meetodit kahtlaste tehingute tuvastamiseks, mis võivad olla seotud rahapesuga. HMM modelleerib süsteemi, kus esmalt eeldame Markovi protsessi, mis on vaatlejale varjatud. Selle varjatud protsessi seisund genereerib aga vaadeldavaid väärtuseid. HMM sobitub finantstehingute seire olukorda rahapesu tuvastamiseks, nimelt tehingu eesmärk (panna toime rahapesu või mitte) on varjatud ja finantsasutus näeb ainult loetud tehingu parameetreid. Mudel ehitati ja testiti Salv Technologies'i kunstlike andmete põhjal ning

võrdlemiseks valiti tavaliselt kasutatav k-keskmiste (*k-means)* klasterdamine. Testimised ja analüüs näitasid, et HMM edestab k-keskmiste klasterdamis meetodit. Järeldusena võib öelda, et olemuslikult sobib HMM finantstehingute seiresse, aga täpsuse saavutamiseks on vaja valdkonna teadmisi ja praktilist testimist. Lõputöös antakse ülevaade rahapesust, käsitletakse anomaaliate avastamise meetodeid, HMM metoodikat. Praktilises osas käsitletakse HMM-i rakendamist andmekogumites kolmel erineval juhtumil. kasutades tarkvara R.

CERCS teaduseriala: P160 Statistika, operatsioonianalüüs, programmeerimine, finants-ja kindlustusmatemaatika.

Märksõnad: varjatud Markovi mudel, HMM, rahapesu, anomaaliate tuvastamine.

Contents

1. Introduction

Money Laundering is the illegal process of concealing the origins of the money obtained illegally by passing it through a complex sequence of banking transfers or commercial transactions (UNODC, 2021). Almost all criminals and criminal organizations that deal with human trafficking, drug trafficking, illegal arms trafficking, fraud, scams etc. on a larger scale need ways to "legitimize" their earnings. Illegally avoiding taxation of legally obtained finances can also be referred to as money laundering.

No-one can be sure when money laundering first began. However, we can be certain that it has been done for several thousand years. In "Lords of the Rim" Sterling Seagrave explains how, in 2000 B.C. China, merchants would hide their wealth from rulers who would simply take it off them and banish them. In addition to hiding it, they would move it and invest it in businesses in remote provinces or even outside of China (Morris-Cotterill, 2001).

Fight against money laundering aims to hinder, or at least make it very difficult for criminals to legitimize their earnings and thus make the illegal venture less appealing, save potential future crime victims, and make everyday lives of people safer. Due to the importance of detecting money laundering, nearly all international organizations urge the state or private companies to take actions and help them to prevent money laundering.

The Bank of International Settlements (BIS), OECD, the G8, G20, EU members' finance and justice ministers, several departments in UN, World Bank, International Monetary Fund and The Financial Stability Forum (FSF) are the main participants in regulatory efforts designed to reduce and assess money laundering. (Unger, 2007)

One of the most influential organizations tasked with preventing money laundering in a large scale is the Financial Action Task Force (FATF). In response to mounting concern over money laundering, FATF was established by the G-7 Summit in Paris in 1989 to develop a coordinated international provision. One of the first tasks of FATF was to develop a list of recommendations, which establish measures for national governments what they should implement to fight effectively against money laundering (FATF, 2021).

To understand the importance of the fight against money laundering, it is good to get an idea of the scale of criminal finances being laundered through banking systems. According to a study conducted by United Nations Office on Drug and Crime (UNODC) in 2009 it is estimated that the overall amount of criminal proceeds generated in 2009, excluding those derived from tax evasion, may have been approximately $2.1 trillion, or 3.6 per cent of global GDP in that year (UNODC, 2021). Of that total, the proceeds of transnational organized crime - such as drug trafficking, counterfeiting, human trafficking, and small arms smuggling - may have amounted to 1.5 per cent of global GDP, and 70 per cent of those proceeds were likely to have been laundered through the financial system.

According to Sullivan (2015, pp 15-16), FATF (2021), there are basically three methods to make the money clean:

- Using the legitimate financial system (for example, moving money from bank to bank, or to money service businesses (MSB-s));
- Physically moving the money (for example, transporting bulk cash via shipments across the border);
- Physically moving goods through the trade system.

The thesis focuses on the 1st option-and more specifically on detection of suspicious activities (that might be related to money laundering) through the financial institutions. Emphasis goes on *suspicious* since financial institutions can only spot signals of money laundering, formal investigations can only be done by the state law enforcement and fixations of money laundering can only be done by the court system. Financial institutions can however manage their risks and deny transferring the finances even based on suspicions.

In the past few decades, the scale of money laundering has increased because of digitalization and automation of international money transfers. This makes it a lot easier for criminals to transfer money to all sides of the world through different accounts within a short amount of time. But on the positive side, it has become a lot easier to check, monitor and to detect illicit international money transfers and unusual activities. (Muller *et al*, 2007). Detection of unusual activities and illicit money transfers can be done with different methods.

Transaction monitoring - i.e., setting out to find patterns and signs of suspicious or risky behavior – is a practical way of tracking down suspicious activities that might be associated with money laundering. Many financial institutions have their own internal way to define, categorize ongoing transactions as suspicious or normal. Vast majority of financial institutions use rule-based approach such as setting a limit on daily, weekly, monthly incoming, or outgoing transactions amount. If the set limit is passed, then the activity is investigated in more detail. One of the main limitations of this approach is that it can cause unnecessary false alerts. Machine learning methods, such as anomaly detection approaches in statistics investigated by Hawkins (1980) and anomaly techniques to detect credit card fraud used by Aleskerov *et al* (1997) can be applied to trail the features of a transaction.

The thesis considers Hidden Markov Model (HMM) to detect suspicious transactions which might be associated with money laundering. HMM itself includes hidden states which perfectly suits to apply on transaction monitoring process. The intent of the transaction – either conduct an act of money laundering or a normal transaction - is unknown. These hidden states are assumed to depend only on the previous transaction (Markov property). According to hidden state, observable variable values are generated, like transaction amount, currency, time of the transaction, counterparty etc. Several auxiliary variables were considered such as sum of total transaction amount within one day and number of transactions within one day for each bank customer at the time of the transaction. For the model, based on the observable variables, a single *observable* variable was constructed to classify a transaction as either *low risk*, *medium risk*, or *high risk*. Then HMM is used to predict the hidden state with the help of the observable variable. As mentioned earlier, then a financial institution can only detect suspicious actions of their customers, not actual intent of money laundering or illicit behavior. Therefore, the hidden states are relaxed to *suspicious* and *normal*.

The goal of the thesis is to introduce HMM, build a model to detect suspicious transactions, and test it on 3 separate study cases based on artificial data. To compare the results of HMM, another method, the k-means clustering was chosen.

The thesis is separated into 6 sections. Section 2 defines general overview of money laundering, its phases, anomaly detection methods and different applications of HMM. Section 3 covers the methodology part. This section focuses on the theoretical aspects of HMM and necessary information is given that is needed in the empirical part. A brief introduction to k-means clustering is also provided in this section. Section 4 covers the empirical study. Section 5 discusses the results, and finally, conclusions are given.

The analysis was carried out with R software (version 4.0.2) (R Core Team, 2020). Packages such as *data.table* (Dowle *et al* (2021)), *dplyr* (Wickham *et al*, 2021, package version 1.0.5), *HMM* (Himmelmann, 2010, package version 1.0) were used throughout the analysis. Some visualizations were finalized in Tableau (version 2021.1).

2. Literature review

Literature review part gives an overview about money laundering and its phases, then anomaly detection methods and their importance are highlighted, finally, HMM is described as one of the anomaly detection methods.

2.1 Money laundering and its phases

According to Cox (2014, pp 6), money laundering can be defined as the process when a person who has received some form of ill-gotten gains, will seek to ensure that they can use these funds without people realizing that these are obtained by the result of inappropriate behavior. To do this they will need to disguise the proceeds such that the original source of the proceeds is hidden and therefore the funds themselves appear to be legitimate.

There are 3 cycles of money laundering and Cox (2014, pp 15) defines them as follows:

Placement – initial proceeds enter the banking system at a perceived point of weakness.

Layering – the funds are moved around such that the initial source of the funds is disguised.

Integration – the funds are eventually reintegrated into the mainstream banking system as clean funds.

Madinger (2012, pp 8) summarizes those cycles as in Figure 1.

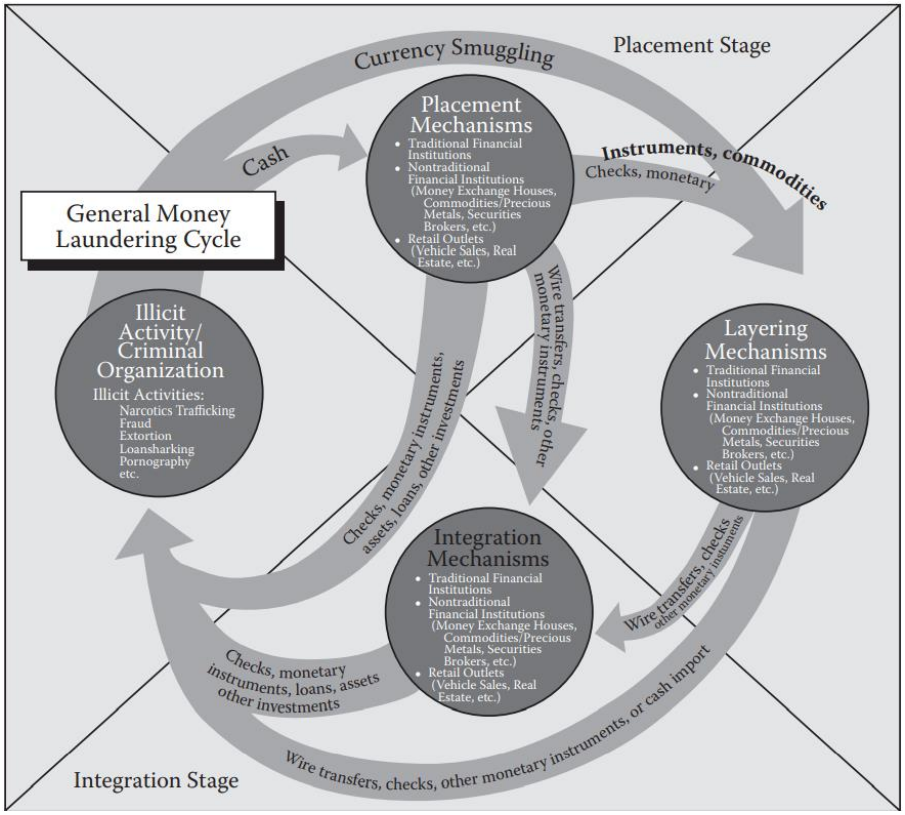

Figure 1. The money laundering cycle, Madinger (2012, pp 8).

In this three-stage process, it is not an easy task to detect "dirty" money. One common reason of difficulty to detect money laundering is that it is related to other crimes. Whenever multiple counts of money laundering and interrelated criminal activity become spliced, a complex network of illicit activity is created and that is extremely hard to fully track and break down. Many financial institutions, businesses, governments set controls to prevent money laundering. These controls are summarized under the name of Anti-Money Laundering (AML). AML is a set of policies, procedures and technologies that try to prevent money laundering. It is implemented within government systems and large financial institutions to monitor potentially fraudulent activity.

According to Sullivan (2015) generally there are 3 goals of quality AML programs:

- To prevent money laundering and terrorist financing.
- To report suspicious activities.

- To train all personnel on legal and internal procedures.

Technological innovations such as applying anomaly detection methods have made it substantially easier to detect when the financial system is being abused, as well as to gather information about the individuals who are abusing it. Manually searching for data and monitoring accounts is time-consuming, inefficient, and often ineffective. Fortunately, screening systems have replaced this old-fashioned process, and have made it easier than ever before to monitor transactions of clients and determine if someone is trying to launder money.

2.2 Anomaly detection methods

To use anomalous behavior detection methods, it should first be underlined how we define what is called “normal” behavior. As the anomaly itself cannot be described straightforwardly, it needs a model or a system that should clearly define what it will refer as a “normal” pattern. Then the model will be compared with expected (normal) values.

According to Dunning and Friedman (2014, pp 14) the key steps in anomaly detection are:

- What is normal?
- What will you measure to identify things that are “far” from normal?
- How far is “far” if something is to be considered anomalous?

We can classify many transactions as anomalous. In the context of a banking system, Cox (2014, pp 226-227) brings out some examples of suspicious transactions or actions:

- A customer opens a greater number of different accounts than would be expected for the type of business they are purportedly conducting and/or frequently transfers funds among those accounts.

- A customer's corporate account(s) has deposits or withdrawals primarily in cash rather than online transfers.
- Generally, if a customer frequently makes large dollar transactions (such as deposits, withdrawals, or purchases of monetary instruments) without an explanation how they will be used in the business, or the purchases allegedly are for a business that generally does not deal in large amounts of cash, then investigation will be required.
- If a business that does not normally generate overseas currency suddenly starts to make numerous currency transactions (i.e., a sanitation company that makes numerous deposits of cash), then this should be identified and reviewed.
- If a business owner, such as an owner who has only one store, makes several deposits the same day using different bank branches, then this will be highly unusual.

According to Mehrotra *et al* (2017), there are three desired goals when applying an anomaly detection algorithm:

1. Correct detection - Detected abnormalities in data correspond exactly to abnormalities in the process.
2. False positives - The process continues to be normal, but unexpected data values are observed, e.g., due to intrinsic system noise.
3. False negatives - The process becomes abnormal, but the consequences are not registered in the abnormal data, e.g., due to the signal of the abnormality being insufficiently strong compared to the noise in the system.

In practice it is nearly impossible to reach the maximum of all goals and detect every abnormality. General approach here is to minimize the false positives and false negatives.

According to Alla *et al* (2019) an anomaly can be split into 3 general categories:

- Data point-based anomalies
- Context-based anomalies

- Pattern-based anomalies

Main idea of data point-based anomalies that they are not expected to have in data set. These types of anomalies can be found wherever a data set of values exists. An example of this is a data set of thyroid diagnostic values, where most of the data points are indicative of normal thyroid functionality (The thyroid gland is a small butterfly-shaped gland in the neck. One of its main functions is to produce hormones that help regulate the body's metabolism). In this case, anomalous values represent sick thyroids. While they are not necessarily outliers, they have a low probability of existing when considering all the normal data.

Context-based anomalies consist of data points that might seem normal firstly, but if the context is considered then can be underlined as anomalies. For example, a person who makes a high volume of purchases towards Black Friday (Black Friday refers to the day after the U.S. Thanksgiving holiday, it is typically a day full of special shopping deals and heavy discounts and is considered the beginning of the holiday shopping season) is not flagged because it is typical for people to do so around that time. However, if the purchases are made in a month where it is out of place given previous purchase history, it would be flagged as an anomaly.

Lastly, third group of anomalies are the pattern-based anomalies which as the name suggests deviate from its long-term patterns or trends. For example, in the context of financial transactions, if a person usually has been withdrawing her money from a bank on a specific day each month for a long time, suddenly starts to withdraw on unusual days, then this action can be considered as anomaly as it breaks the long-term pattern.

According to Alla *et al* (2019), there are three kinds of styles of anomaly detection:

- Supervised anomaly detection
- Semi-supervised anomaly detection
- Unsupervised anomaly detection.

Supervised anomaly detection is a technique which can be applied to the training data where both anomalies and normal data point are identified beforehand. Basically, model knows which data point is normal and which one is not. An example of this can be a temporal convolutional network (Alla *et al*, 2019)

If the training data is partially identified, then semi-supervised anomaly detection techniques can be applied. For example, initial conditions can be set that how a normal data point looks like in the dataset. Examples of models that can use semi-supervised learning for anomaly detection include autoencoders.

If training data is not labelled, and data points are classified as “anomaly” or “normal” after the training process, then it is referred to as unsupervised anomaly detection. Isolation forest is an example of technique that can be applied on unsupervised dataset.

Different anomaly techniques are used to find out frauds in credit card and insurance areas which are the closest areas to money laundering. Aleskerov *et al* (1997), Ghosh and Reilly (1994), Dorronsoro *et al* (1997) investigated neural networks techniques to unveil credit card frauds. Brause *et al* (1999) and Bolton (2001) used rule-based systems clustering methods for detection of credit card frauds. Neural network-based techniques have been applied to identify insurance claim fraud (Li *et al* ,2008, Brockett *et al*, 1994), but generally this kind of fraud is handled as a generic activity monitoring problem (Fawcett *et al*, 1997).

HMM can also be particularly useful for detecting anomalous behavior, for example, Ourston *et al* (2003) have proposed the application of Hidden Markov Models in detecting multistage network attacks.

2.3 Hidden Markov model

Among anomaly detection methods, HMM is less used, but it offers many advantages, especially, in the context of financial transaction monitoring.

HMM has a finite set of states, each of which is associated with a (generally multidimensional) probability distribution. Transitions between the states are defined by a set of probabilities called transition probabilities. In a particular state an outcome or observation is generated, according to the associated probability distribution. It is only the outcome, not the state, that is visible to an external observer and therefore states are “hidden” to the outside, hence the name is hidden Markov model.

HMM has been used to successfully to model many real-world processes. The two hierarchy-level structure is the main idea and advantage of HMM, it can be used to model much more complicated stochastic processes than traditional Markov model.

Mhamane and Lobo (2012) introduced HMM to detect internet banking fraud in their article. They use Baum-Welch algorithm to estimate HMM parameters like state and transition probabilities but in the paper no real simulation is given as it is mainly focused on theoretically explaining feasibility of HMM. Jadhav and Bandari (2012) implemented HMM on credit card transaction. They found that HMM helped to reduce the number of false negatives.

Kasianova (2020) applied HMM on defining a type of transaction of each client as either being “suspicious” or “normal”. HMM was applied for each user separately and observable variable was set to get value either high risk or low risk. It was concluded that HMM is a reliable model for detecting the riskiness of transaction.

3. Methodology

In this section theoretical aspects of HMM are discussed, k-means - the comparison method is introduced, and lastly, to compare the results of both models, quality assessment metrics are provided.

3.1. Hidden Markov models

Part of the HMM is a sequence of states that assume the Markov chain property to hold, so before proceeding with HMM, the concept of Markov chain is introduced. Then HMM, its properties and the algorithm used to estimate transition and emission probabilities - Baum-Welch and Viterbi algorithm - are provided.

3.1.1 Markov chain and its properties

Markov chain is named after Prof. Andrei A. Markov (1856-1922) who first published his results in 1906. Theoretically, he showed that the weak law of large numbers and other important results of the calculus of probability were valid not only for independent events, as assumed by classical stochastics, but also for samples that were connected in simple or multiple chains. It is widely applied on different problems in game theory, genetics, social science, finance, economics, computer science etc. Being the simplest Markov model, Markov chain concerns about a sequence of random variables, which are related to the states of stochastic process, in such a way that the state at one time depends only on the one in the previous time (Ching *et al*, 2013). The state space, or set of all possible states, can be anything: letters, numbers, weather conditions, baseball scores, or stock performances.

A basic example is the two-state process. For example, let us assume that $S = \{S_1, S_2\}$ is the 2-state process (takes values of $(0,1)$) based on stock market trend. $S_1 = 0$ if stock market exhibits bear market (bear market- downward market trend, decrease in stock prices) or $S_2 = 1$ stock market exhibits bull market (bull market- upward market trend, increase in stock prices) in time instants - $t =$

$1,2,..n$. As we have only 2 states and future market trend depends only on current state, then it is a Markov chain process. We can easily set up transition probabilities:

$$\pi_{11} = \alpha \text{ then } \pi_{12} = 1 - \alpha \text{ and accordingly } \pi_{21} = \beta \text{ and } \pi_{22} = 1 - \beta$$

where α and β represent the initial probabilities in each state (α if stock market is bull market and β if stock market is bear market). So, we can develop one step transition matrix P based on this information:

$$A = \{\pi_{ij}\} = \begin{pmatrix} \alpha & 1-\alpha \\ \beta & 1-\beta \end{pmatrix}$$

Consider a more general system which may be described at any time as being in any of a set of *N* states, S_1 , S_2 … S_N. If time instants associated with state changes are denoted as $t = 1,2,..n$, then actual state as time t can be marked as q_t. According to the Markov chain property, being in actual state q_t depends only on previous state q_{t-1}. Then the relationship can be expressed as below:

$$a_{ij} = P\,[\,q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, ..] = P[q_t = S_j | q_{t-1} = S_i]$$

We can generalize relationships in transition probability matrix:

$$A = \{a_{ij}\} = \begin{pmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{pmatrix}$$

where probabilities a_{ij} have properties $a_{ij} \geq 0$ and $\sum_{j=1}^{N} a_{ij} = 1$. We assume that transition probabilities do not depend on time t, i.e., we have a homogeneous Markov chain.

As we have discussed in Markov chain, each observable state indicates certain process. But there is a special form of Markov model in which it is not possible

directly observe the true state. In this type of model only some indications can be measured about the true state and based on that true state or hidden state can be determined. This special type of model is called Hidden Markov model as the true states are hidden.

3.1.2 Overview of Hidden Markov model

Let us present an example of where HMM can be fitted. Jonas and Elvis are pen pals and constantly write to each other. Assume that based on weather Jonas decides to wear clothes with certain color - either black or white. His friend Elvis due to pandemic is stuck in another country and does not know about the weather condition where Jonas lives. Elvis only knows about his color preferences. So how must Elvis figure out the weather condition, based on information of Jonas's decision on color?

Let us give some initial, transition probabilities and figure out how Hidden Markov model can be built.

Firstly, there are hidden states as being weather conditions:

- S_1 - sunny
- S_2 - rainy

Secondly, based on Markov property, current state depends on only previous one, it is possible to develop transition probabilities between states. For example, consider that Elvis knows that if today is sunny then tomorrow will be sunny with probability 0.8 and rainy 0.2; if weather is rainy today then tomorrow will be rainy with probability 0.6 and sunny 0.4. So, transition probability matrix A can be described as:

$$A = \begin{pmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{pmatrix}$$

As was stated above, Elvis knows Jonas's decision on color: if it is sunny then Jonas wears white clothes with probability 0.7 and black 0.3. If the weather is rainy then Jonas decides to wear black cloth with probability 0.6 and white with probability 0.4. These probabilities are called emission probabilities which indicate probabilities of observations are emitted from hidden states. We can express those ones in a matrix:

$$B = \begin{pmatrix} 0.7 & 0.3 \\ 0.6 & 0.4 \end{pmatrix}$$

And lastly, Elvis knows about the weather condition on that day with probabilities called initial probabilities: 1st day probability of being rainy is 0.4 and sunny is 0.6.

$$\pi = \{\pi_1 = 0.6;\ \pi_2 = 0.4\}$$

Considering all these probabilities, Jonas's decision on what color clothes he wore each day, then Elvis can calculate what's today's weather.

All the process can be described as in the Figure 2.

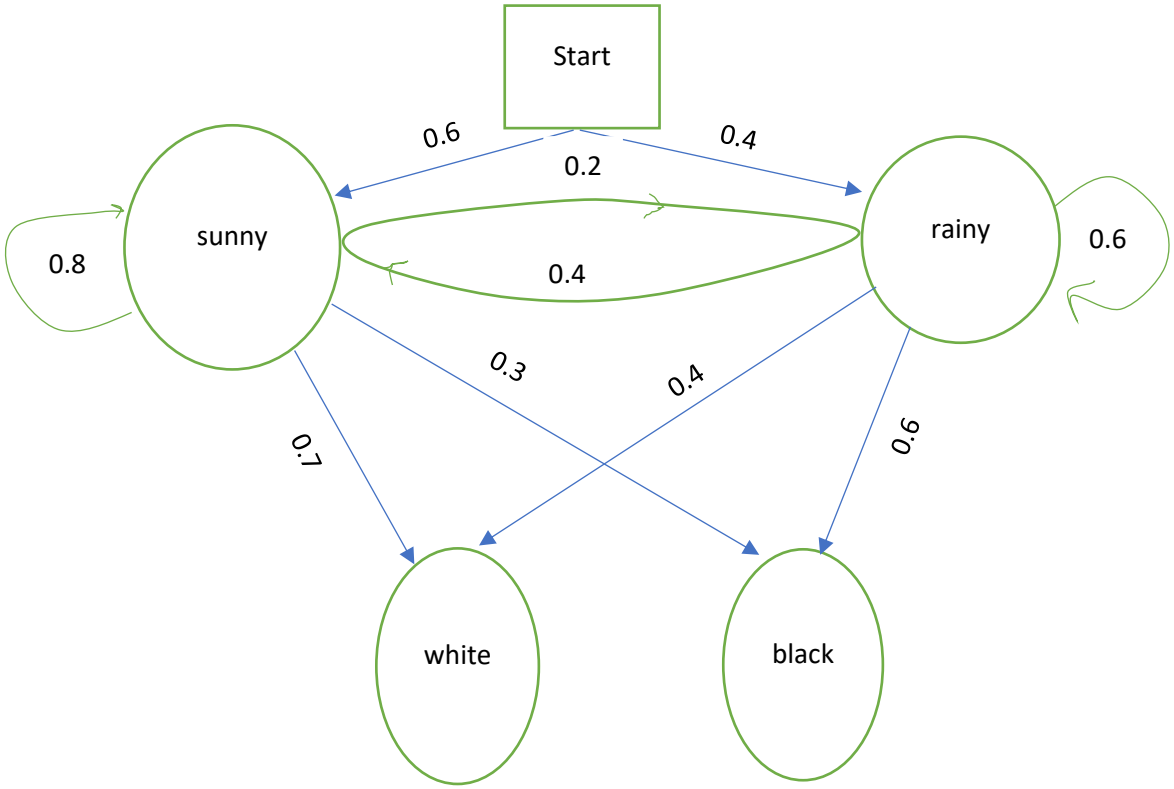

Figure 2. Visualization of HMM (source: Author)

As mentioned above HMM describes the process where it is not possible directly to figure out the needed outcome. But in each state like in Markov model certain processes could be detected and main role and usefulness of HMM is the deriving of outcome from this complex puzzle. In certain fields HMM suits very well to the case, for example, in financial transaction monitoring in the context of money laundering detection. It is not possible to directly say that a transaction is suspicious or not. But based on features or parameters of the transaction it is possible to estimate if it was done with ill-intentions in mind. This translates to the observer - is this transaction suspicious or not.

3.1.3 Elements of a Hidden Markov Model

Above example showed the concept of a hidden Markov Model. We now can formally indicate and define the elements of HMM.

According to Rabiner (1989) HMM is specified by the following:

1. N, the number of hidden states in the model. As we saw from above example that although the states are hidden, they are associated with certain kinds of indicators. In our example those indicators were weather states – sunny or rainy. It is much alike the ergodic model in essence that states are interrelated (next day's weather condition depends on previous day's weather and if today is sunny most like tomorrow will be sunny as well or vice versa). Individual states are denoted as $S = \{S_1, S_2 \dots S_n\}$ and state at the time t as q_t.
2. M represents the number of distinct observation symbols. In our example, there were only 2 of them as color preference either black or white. Distinct observations can be denoted as $V = \{v_1, v_2 \dots, v_M\}$.
3. The state transition probability distribution $A = \{a_{ij}\}$, where

$$a_{ij} = P\big(q_{t+1} = S_j \mid q_t = S_i\big), \qquad 1 \le i \le N$$

For the special case in which any state can reach to any other one with single step, then $a_{ij} > 0 \; \forall \; i, j$. If state S_j cannot be reached from state S_i in a single step, then $a_{ij} = 0$.

4. The emission probability distribution (or observation symbol probability distribution) in state $S_j, B = \{ b_j(k)\}$, where

$$b_j(k) = P(v_k \; at \; t | q_t = S_j), \qquad 1 \le j \le N, \quad 1 \le k \le M.$$

5. The initial state probabilities at the start of the process are $\pi = \{\pi_i\}$, where

$$\pi_i = P(q_1 = S_i), \qquad 1 \le i \le N$$

Considering all these elements of N, M, A, B, and π, then the defined HMM generates an observation sequence- $O = o_1 o_2 \dots o_T$, as below, where o_i represent one of the symbols from V, and T is the number of observations:

- Choose an initial state.
- Set $t = 1$.
- Choose observation according to the emission probability distribution in state S_i.
- Transit to a new state $q_{t+1} = S_j$ according to transition probability distribution for S_i.
- Set $t = t + 1$; keep going until t reaches T.

For simplicity based on the elements of HMM, we will use from now on a compact notation for defining the complete parameters of HMM as below:

$$\lambda = (A, B, \pi)$$

According to Jurafsky and Martin (2008) there are 3 assumptions of HMM:

- Markov Assumption - Simply predicting future the past does not matter, only present is encountered.
- Independent Assumption - Probability of an out observation o_i depends only on the state q_i that produced the observation, not any other state or other observations.
- The Stationary Assumption - State transition probabilities are independent of the actual time which the transition takes place. So, transition probabilities are fixed.

3.1.4 Problems of HMM and computation algorithms

As the structure of HMM is described, it is logical to step into the algorithms which can be used to calculate the probability of hidden state in any given time t. Before proceeding with algorithms, it should be defined that HMM is characterized by three fundamental problems (Rabiner, 1989):

1. Likelihood - Given an HMM $\lambda = (A, B, \pi)$ and observation sequence O, determine the likelihood $P\ (O|\ \lambda)$.
2. Decoding - Given an observation sequence O and HMM $\lambda = (A, B, \pi)$, figure out the best hidden state sequence Q, where $Q = q_1 q_2 \ldots q_T$
3. Learning - Given an observation sequence O and set of states in the HMM, learn HMM parameter A and B.

Our main goal in this article is to focus on the second problem as we will determine that based on data either transaction is suspicious or normal. Learning phase is also done to estimate A and B.

There are many decoding algorithms defined, the most well-known are Viterbi, and PSA (Prefix Sum Arrays) decoding algorithms.

Viterbi algorithm - is the most often used in practice for defining the most likely hidden states in HMM by considering maximum likelihood probabilities for each state (Forney, 2005).

Considering complete parameters $\lambda = (A, B, \pi)$ of HMM, Viterbi algorithm defines the most likely path (also called Viterbi path) in the sequence by calculating each likelihood probabilities of hidden states based on the observation sequence O.

Baum-Welch algorithm - is used to solve mentioned 3rd problem of HMM and estimate transition and emission probabilities in the empirical part. The algorithm is a special case of the expectation-maximization (EM) algorithm (Jurafsky and Martin, 2008). EM is an iterative algorithm which estimates initial probabilities then uses those ones to get a better result, iteratively improving the probabilities. Baum-Welch uses this feature of EM algorithm to find the maximum likelihood estimate of the parameters A, B for HMM.

3.2. K-means clustering algorithm.

K-means clustering algorithm is one of the basic and most used algorithms to group data. This algorithm gets its name based on the logic that observations $(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_m)$, where each observation is a d - dimensional real vector, are divided into K clusters, where each observation is related to the cluster with the nearest mean. In empirical part, k-means algorithm is used to conclude the reliability of HMM by comparing results of two models. Basically, k-means algorithm was used in this thesis to split the transactions in datasets into 2 clusters - *suspicious* or *normal*, based on a set of transaction characteristics.

According to Wu (2012), k-means clustering is a prototype-based, simple partitional clustering algorithm that aims to find K non-overlapping clusters. Centroids represent each cluster (a cluster centroid is typically the mean of the points in that cluster).

Steps in k-means algorithm can be described as follows (Hartigan and Wong, 1979):

- Number of clusters - K is defined.
- Select random points from data as centroids.
- Assign every point in the data to a cluster with the closest centroid.
- Recompute the centroids of newly formed clusters until there is no change to the centroids i.e., assignment of data points to clusters is not changing.
- Compute the sum of the squared distance between data points and their assigned cluster centroids.
- Assign each data point to the closest cluster (centroid).
- Compute the centroids for the clusters by taking the average of all data points that belong to each cluster.

The objective function J is defined as:

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^2$$

where $w_{ik} = 1$ for data point x_i if it belongs to cluster k; otherwise, $w_{ik} = 0$, $\| \ldots \|$ is a distance, $\boldsymbol{\mu}_k$ is the centroid of $\boldsymbol{x}_i$'s cluster, K is the number of clusters and m is the number of data points.

Above function is two-part minimization problem. Firstly, the function J is minimized with respect to w_{ik} and the centroids $\boldsymbol{\mu}_k$ are assumed to be fixed. Secondly, function J is minimized with respect to centroids $\boldsymbol{\mu}_k$ and w_{ik} is assumed to be fixed. In other words, function J is differentiated with respect to w_{ik} first and cluster assignments updated. Then function J is differentiated with respect to centroids $\boldsymbol{\mu}_k$ and the centroids are recomputed after the cluster assignments from previous step. So, the first step is solving the following equation:

$$\text{argmin}_{w_{ik}} \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^2 \rightarrow w_{ik} = \begin{cases} 1, & \text{if } k = \text{argmin}_j \left\|\boldsymbol{x}_i - \boldsymbol{\mu}_j\right\|^2 \\ 0, & \text{otherwise} \end{cases}$$

It basically means assign the data point $\boldsymbol{x_i}$ to the closest cluster judged by its sum of squared distance from cluster's centroid.

The second step can be mathematically expressed as:

$$\frac{\partial J}{\partial \boldsymbol{\mu}_k} = 2 \sum_{i=1}^{m} w_{ik}(\boldsymbol{x}_i - \boldsymbol{\mu}_k) = 0 \rightarrow \boldsymbol{\mu}_k = \frac{\sum_{i=1}^{m} w_{ik} \boldsymbol{x}_i}{\sum_{i=1}^{m} w_{ik}}$$

Which translates to recomputing the centroid of each cluster to reflect the new assignments.

In the empirical part the *Stats* (Hartigan and Wong (1979) package was used to apply this algorithm in R.

3.3. Quality assessment metrics

To compare the results of selected models, we use Precision, Recall (Sensitivity), and F- score for comparison.

Confusion matrix is used to give a better overview about precision and recall. Confusion matrix is a 2x2 table that cross-checks predictions with actual values. The confusion matrix provides profound information not only about the performance of predictive model, but also which classes are being predicted correctly, which incorrectly and what type of error is being made. General form of confusion matrix can be described as on Figure 3.

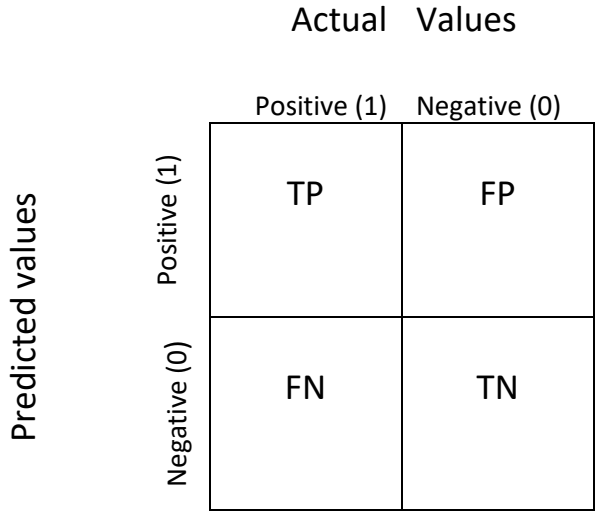

Figure 3. Confusion Matrix

A 2x2 confusion matrix on Figure 3 has 2 states for actual values – e.g., positive (True) and negative (False) – and 2 states for predicted values. The result is a table with 4 different combinations of predicted and actual values:

- true positives (TP): These are cases in which we predicted positive, and it is true.
- true negatives (TN): We predicted negative, and it is true.
- false positives (FP): We predicted positive, and it is false (also known as "Type I error").
- false negatives (FN): We predicted negative, and it is false (also known as "Type II error").

Precision (Positive Predictive Value) turns out all positive classes how much it has predicted correctly, how many are actually positive:

$$Precision = \frac{TP}{TP+FP}$$

Recall (Sensitivity) turns out all positive classes, how much it is predicted correctly:

$$Recall = \frac{TP}{TP+FN}$$

Generally, precision is appropriate for minimizing false positives and recall is appropriate for minimizing false negatives.

But neither precision nor recall alone gives the basis for a reliable conclusion. It is highly possible to get excellent precision with terrible recall or vice versa. F-score provides a way to handle both concerns with a single score. F-score is the harmonic mean of the two fractions. It is described mathematically as below:

$$\text{F-Score} = \frac{2*Recall*Precision}{Recall+Precision}$$

4. Empirical Study

This section gives a general overview of data and applied HMM and k-means clustering on 3 different study cases.

4.1. Data

Artificial data provided by Salv Technologies was used to implement HMM for detection of suspicious financial transactions. The data consists of a made-up population (persons and entities) who then perform financial transactions with each other. The generated financial transactions include details such as bank details, counterparty details, date of transaction, amount, transaction type (incoming or outgoing) and currency. Some transactions are generated by scripts that mimic money laundering transactions or scenarios. The fact if a transaction was generated as a “normal” or “suspicious” transaction is recorded with the data. In other words, the data is labelled, we know which transaction is suspicious and based on that we can measure the implemented HMM and k-means clustering models’ reliabilities and efficacies.

Study case 1 - a model was set up on “training” data which contained small amount of the data. Overall, 5000 transactions of 10 users were analyzed and 5 transactions (0.1%) were “suspicious”.

Study case 2 - a model was built only on suspicious transactions. Overall, 439 unique users and their 6854 transactions were analyzed.

Study case 3 - a model was built on a large dataset. Overall, 55275 transactions of 32 users were analyzed and 239 transactions (0.4%) were “suspicious”.

ï..id	user_id	status	type	date_completed	sender_account	from_cur	to_cur	amount_in_eur	meta_sar_id
23830541	74093	Completed	I	01-04-20 4:44	EE255924620987810888	EUR	EUR	47896.66	4
23830542	74093	Completed	I	02-04-20 10:19	EE65496356039486690088	EUR	EUR	77572.71	4
23830543	88177	Completed	I	02-04-20 20:04	EE45806865121799287877	EUR	EUR	77613.89	9
23816777	80993	Completed	I	02-04-20 5:15	EE344264109541425663	EUR	EUR	30000.00	11
23816778	28227	Completed	I	03-04-20 19:48	EE159122404493878963 05	EUR	EUR	24786.70	11
23830545	86874	Completed	I	03-04-20 22:58	EE089286735910438438 29	EUR	EUR	64649.18	9
23830544	88177	Completed	I	03-04-20 7:27	EE324255767097361694	EUR	EUR	17532.30	9
23830547	88177	Completed	I	05-04-20 12:41	EE759070211291663603 11	EUR	EUR	96112.69	9
23830546	88177	Completed	I	05-04-20 7:18	EE243164452063762815 79	EUR	EUR	90213.27	9
23830548	86279	Completed	I	07-04-20 2:21	EE653503680249607281 47	EUR	EUR	79578.65	9

Figure 4. Example of raw data

Below are the variables that were used for analyzing data:

- user_id - is a variable that uniquely defines the person who is one counteparty of the transaction within the artificial bank;
- type - shows the direction of transaction (either outgoing or incoming);
- Date_completed - shows the date and time when the transaction is completed;
- from_cur and to_cur - variables that show currencies of the transaction;
- amount_in_eur - amount of transaction in Euro equivalent;
- meta_sar_id - identifies if the transaction was deliberately generated to be suspicious and the ID number gives the exact scenario. In raw dataset “suspicious” transactions had values greater than 0, values was fixed for simplicity for all three study cases. If generated as normal behavior, then value is equal to 0. The variable is used to create a binary variable – 1 if suspicious, 0 if normal;
- counterparty_country - shows the country of origin of the transaction counterparty.

4.2 Study Setup

The main idea is that transactions are considered as either "normal" or "suspicious" and this fact is taken as the hidden states in the HMM. So, the state space will be $S = \{S_1 = normal, S_2 = suspicious\}$.

The next step to build HMM is to define the "observable" variable. In all three studies, an intermediary variable was created from a linear combination of transaction features that are considered risk factors in the fight against money laundering. This composition was taken from AML practice. The new composite variable was named the *"score"* variable.

Table 1. Example of an auxiliary variable – "*score*"

user_id	status	type	date_completed	amount	from_cur	to_cur	amount_in_eur	meta_sar id	score
11028	Completed	I	02-12-19 17:08	130	EUR	EUR	130	0	10
11028	Completed	I	03-12-19 19:32	840.14	EUR	EUR	840.14	0	10
11032	Completed	I	26-03-20 9:55	21.79	EUR	EUR	21.79	0	5
11032	Completed	I	27-04-20 4:08	23023	EUR	EUR	23023	1	50
11032	Completed	o	27-04-20 19:37	20797.17	EUR	EUR	20797.17	1	55

"*Score*" is an auxiliary numeric variable which was created based on features of other variables such as date of transaction, counterparty country, type of currencies etc. For example, if transaction is done late or early time of the day, then the "*score"* variable value is increased. Also, type of currency and transaction counterparty country affect the increment of *"score"* variable, for example if a transaction originates from a high-risk country, then *"score"* is increased. The list of high-risk countries (Appendix C) includes high-risk and other monitored jurisdictions from FATF (FATF, 2020),- list of offshore countries managed by the International Monetary Fund (International Monetary Fund, 2019) and the European Commission (European Commission, 2021). The components of *"score"*

are taken by rules, which are usually used in the rule-based method to detect money laundering and were built using domain knowledge.

Finally, if *"score"* crosses a certain threshold, defined for each study case separately, then the "observable" variable defines transaction as "high risk", "medium risk" or "low risk", i.e., $V = \{v_1 = low\_risk, v_2 = high\_risk, v_3 = medium\_risk\}$. The higher the value of variable *"score"*, the riskier the transaction.

4.2.1. Study case 1

To make analysis more reliable, additionally as mentioned in table 2, other 3 variables created:

- sum_1in - sum of incoming transaction amounts within 1 day for same user;
- sum_1out - sum of outgoing transaction amounts within 1 day for same user;
- count_1 - number of transactions within 1 day for the same user.

After analyzing the data, the *"score"* variable for study case 1 and 2 was manually created based on rules indicated in Table 2.

Table 2. Formation of ""score"" variable for study cases 1 and 2

Rules	Increment of *"score"*
Time Range of transaction: 21PM-7AM	15
amount_in_eur>=500 amount_in_eur<1000	5
amount_in_eur>=1000 amount_in_eur<5000	10

amount_in_eur>=5000 amount_in_eur<20000	15
amount_in_eur>=20000	20
Counterparty_risk country	10
Currency other than USD or EUR	10
sum_1in>=1000 and sum_1in<5000	5
sum_1in>=5000 and sum_1in<10000	10
sum_1in>=10000	15
sum_1out>=500 and sum_1out<1000	5
sum_1out>=1000 and sum_1out<10000	10
sum_1out>=10000	15
count_1>=2 and count_1<3	5
count_1>=3 and count_1<5	10
count_1>=5	15

After estimation of "*score*" variable for every transaction, maximum of *"score"* variable was calculated for every user. Based on the maximum "*score*" variable, the observable variable for each transaction was calculated as follows:

- If $score < \max(score) * \frac{1}{2}$, then observable variable for this transaction is *low_risk;*
- If $score \geq \max(score) * \frac{1}{2}$ and $score \leq \max(score) * \frac{9}{10}$ then observable variable for this transaction is *medium_risk;*
- If $score \geq \max(score) * \frac{9}{10}$, then observable variable for this transaction is *high_risk;*
- If $\max(score) = 0$, then all transactions for this user are considered as *low_risk*.

Table 3 gives an overview of data with calculated *"score"*, maximum *"score"*, and observable variables before applying HMM.

Table 3. Example of data for Study 1 before application of HMM

user_id	type	amount_in_eur	meta_sar_id	score	counterparty_country	sum_1in	sum_1out	count_1	max_score	observation
11028	O	51.5	0	15	EE	0	51.5	1	40	low_risk
11032	I	23023	1	50	EE	23193	0	2	55	high_risk
11032	O	20797.17	1	55	EE	23023	20887.17	4	55	high_risk
11097	I	31258	1	55	GT	31279.5	6.24	4	55	high_risk
15177	I	15.52	0	25	EE	97.49	31.76	5	40	medium_risk

Last step was application of HMM to define the hidden state based on the observation sequence. The transition probability from "suspicious" state to "normal" was set a little bit higher as a person can manipulate and commit "normal" transactions as well. So, the model parameters were set as below:

- Hidden states: $S = \{S_1 = normal, S_2 = suspicious\}$;
- Observable values: $V = \{v_1 = \text{“}low\_risk\text{”}, v_2 = \text{"}medium\_risk\text{"}, v_3 = \text{“}high\_risk\text{”}\}$;
- Initial probabilities: $\pi = \{\pi_1 = 0.5,\ \pi_2 = 0.5\}$;
- Transition probabilities: $A = \{\ a_{11} = 0.9, a_{12} = 0.1,\ a_{21} = 0.4, a_{22} = 0.6\}$;
- Emission probabilities:

$$B = \{\ b_1(low\_risk) = 0.7, b_1(medium\_risk) = 0.25, b_1(high\_risk = 0.05,$$
$$b_2(low\_risk) = 0.01, b_2(medium\_risk) = 0.09, b_2(high\_risk) = 0.9\}$$

After all parameters were defined, HMM was applied on 5000 transactions of 10 users.

4.2.2. Study Case 2

In study case 2, only suspicious transactions were considered. 6854 transactions of 239 users were analyzed.

Conditions for defining observable variable for every transaction was changed after analyzing the data. It was identified that many transactions had a small "*score*" value in the defined dataset. So, new conditions were as follows:

- If $score < \max(score) * \frac{1}{3}$, then observable variable for this transaction is *low_risk;*
- If $score \geq \max(score) * \frac{1}{3}$ and $score \leq \max(score) * \frac{2}{3}$ then observable variable for this transaction is *medium_risk;*
- If $score \geq \max(score) * \frac{2}{3}$, then observable variable for this transaction is *high_risk;*
- If $\max(score) = 0$, then all transactions for this user are considered as *low_risk*.

After all was set up, the data before applying HMM looked as below in table 4.

Table 4. Example of data for study 2 before application of HMM

User_id	status	type	from_cur	to_cur	amount_in_eur	meta_sar_id	score	Country	sum_1in	sum_1out	count_1	max_score_s	observation
11229	Completed	I	EUR	EUR	4295.25	1	25	EE	5611.24	0	2	55	medium_risk
11229	Completed	I	EUR	EUR	3777	1	15	EE	3777	0	1	55	low_risk
11229	Completed	I	EUR	EUR	2223.46	1	15	EE	2223.46	0	1	55	low_risk
11229	Completed	I	EUR	EUR	8017.55	1	25	EE	8017.55	0	1	55	medium_risk

4.2.3. Study case 3

Total of 55275 transactions of 32 users were analyzed in Study case 3 where 235 (0.4%) transactions were suspicious. Table 5 shows the changes were done on estimation of *"score"* variables as value range of transaction amount was much narrower than the datasets in study 1 and study 2 cases.

Table 5. Updates on *"score"* variable for study case 3

Conditions	Increment of *"score"* variable
amount_in_eur>=1000 and amount_in_eur<3000	10
amount_in_eur>=3000 and amount_in_eur<10000	15
amount_in_eur>=10000,	20
sender_account=="" and counterparty_country== "" (which means it a cash deposit transaction)	10

Conditions for defining observable variable for every transaction and model parameters stayed the same as in study 1.

Better transition and emission probabilities were calculated by applying Baum-Welch algorithm, then Viterbi algorithm was used to estimate the "hidden" state for each transaction. The same algorithms were applied in all three study cases. It was defined that initial transition probabilities did not have much impact on defining the "hidden" states unlike initial emission probabilities.

K-means clustering method was applied for each study case. Then HMM was compared with k-means clustering for each study cases separately. Quality assessment metrics- Precision, Recall, and F-score were used for comparison the results of both models.

5 Results

For study case 1, among 5000 transactions only 5 (0.1%) were suspicious and HMM correctly figured out 4 of them. Overall, 9 transactions were defined as suspicious by HMM and an important point is that HMM turned only those transactions as suspicious which got high risk in "*observation"* variable including identified false positives. Those transactions should be investigated by AML specialist further. K-means, on the other hand, turned out huge amount of false positives - 4980 out of total 4985 predicted suspicious transactions, which may take lots of investigation time for sorting out true positive ones. Due to having huge number of false positives, precision was only around 0.1% for k-means. Considering all of these it can be said, HMM outraced k-means in study 1. Outcome of HMM in study 1 generalized in Figure 5. As it is seen from Figure 5 that detecting "hidden" states is highly dependent on type of observed variable. None of the "low risk" transactions turned out to be suspicious and 0.08% was in "high risk" category out of 0.1% of actual suspicious transactions. A huge part of "medium risk" transaction was predicted as normal. We can imply that HMM tended to turn out mainly "high risk" transaction as suspicious.

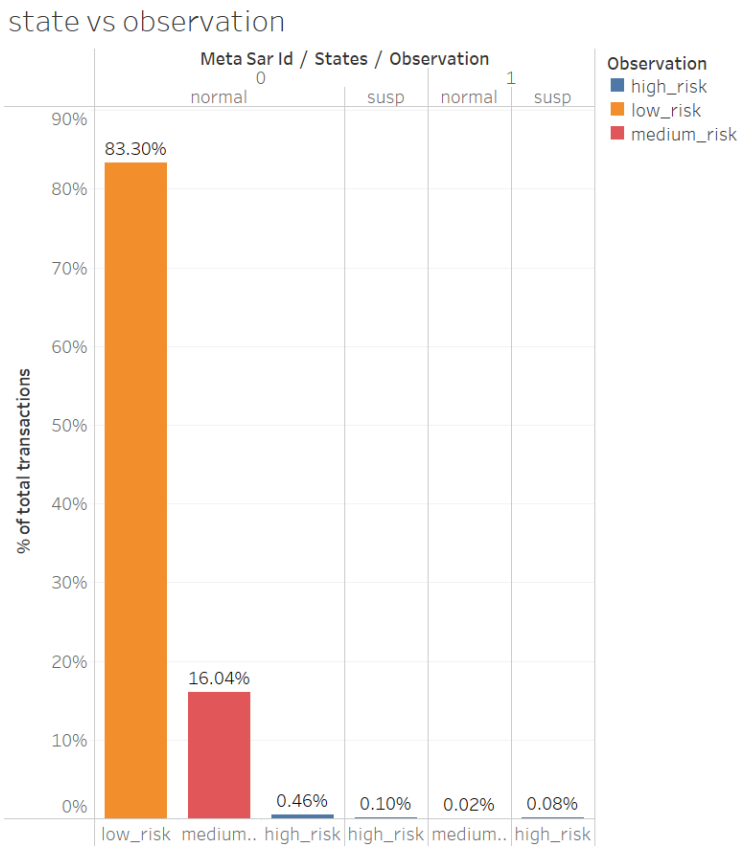

Figure 5. Study 1 results for HMM

HMM showed its real reliability in Study 2 where all analyzed 6854 transactions were considered as suspicious. HMM correctly predicted 5771 (84%) of them as suspicious. As all transactions were suspicious in study 2, precision for both models was in its maximum. HMM falsely predicted only 16% of transactions as normal. On the other hand, k-means clustering predicted 5613 (82%) transactions as suspicious, and number of false negatives was dramatically high as being 1241 (18%) transactions. As a result, HMM outperformed k-means clustering in all 3 quality assessment metrics. Outcome of HMM in study 2 generalized in Figure 6. As it is seen, HMM mostly turned out “high risk” transaction as suspicious. Over 59% of total 60.8 % “high risk” transactions were defined as suspicious.

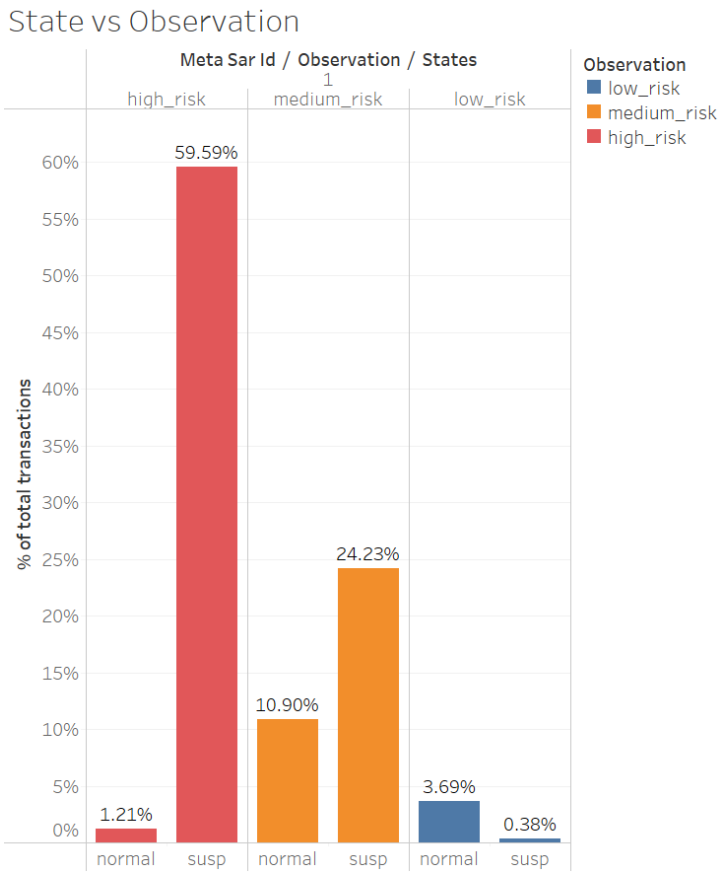

Figure 6. Study 2 results for HMM

And lastly in study 3, overall, 55275 transactions of 32 users were analyzed where 235 transactions of 11 users were suspicious. HMM truly predicted 95 (40%) transactions of 9 users as suspicious. HMM incorrectly identified 83 transactions of 5 users as suspicious which were not. But important point is that from 80 transactions of 83 incorrectly defined transactions were categorized as “high risk” transactions. Those transactions should be investigated further. In contrast, k-means clustering predicted 54537 transactions falsely as suspicious which significantly lowered precision and F-score for the model, despite of having higher recall than HMM. To sum up, it can also be concluded that HMM outperformed k-means in terms of reliability in study 3. Overview of HMM in study 3 in terms of riskiness, feature and state of transactions was given in table 6. As it is seen from table 6, HMM is more sensitive again to turn out “high risk” transaction as suspicious.

Table 6. Study 3 results for HMM

Meta Sar Id	Observation	normal	suspicious
0	high_risk	308	80
	low_risk	47,943	0
	medium_risk	6,706	3
1	high_risk	20	67
	low_risk	29	0
	medium_risk	91	28

The all results of quality assessment metrics for each study were summed up in table 7.

Table 7. Results of both models for each study case.

Results	Study 1		Study 2		Study 3	
	HMM	K-means	HMM	K-means	Hmm	K-means
Precision	44%	0.1%	100%	100%	53%	0.003%
Recall	80%	100%	84%	82%	40%	78%
F-Score	0.57	0.002	0.91	0.90	0.46	0.007

6 Conclusions

The purpose of the thesis was to introduce, build and test HMM as a method of detecting suspicious transaction which might be correlated with money laundering. The model was built and tested on artificial datasets and commonly used k-means clustering model was chosen for comparison.

The thesis gives an overview about money laundering, anomaly detection methods used for detecting suspicious activity in various fields and profound summary of hidden Markov model. HMM was analyzed in 3 three different study cases. In total, more than 70000 transactions were used to test the HMM. As a benchmark k-means clustering was also applied and both models were compared with each other. Quality of models was concluded based on assessment metrics such as precision, recall, F-score. In all three studies HMM showed better results in terms of precision and F-score. As suspicious transactions were randomly generated in artificial datasets, some of them did not fit any logic as being suspicious. But despite of this discrepancy, HMM performed quite well.

The most important part for HMM is the composition of *"score"* variable as it defines the "observable" variable. And based on "observable" variable states are defined. It was realized that HMM mostly tended to identify "high risk" transactions as suspicious. Another important point is the setting up initial emission probabilities. Studies showed that initial emission probabilities have a huge impact on defining "hidden" states. So, formation of the *"score"* variable and setting up initial emission probabilities are very sensitive, and they should be carefully created according to the features of transactions. But for better and much reliable HMM, the formation of *"score"* variable could be constantly reviewed, and several observable levels could be added.

Based on analysis, it can be concluded that in essence, HMM can be accepted as a good model in transaction monitoring but getting high precision needs expert knowledge and practical testing.

References

Aleskerov, E., Freisleben, B., and Rao, B. “CARDWATCH: a neural network based database mining system for credit card fraud detection,” *Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering (CIFEr)*, 1997, pp. 220-226, doi:10.1109/CIFER.1997.618940.

Alla, S., and Adari, K. S. *Beginning Anomaly Detection Using Python-Based Deep Learning: With Keras and PyTorch.* Apress, Berkeley, CA, 2019.

Bolton, R., and Hand, D. *Unsupervised Profiling Methods for Fraud Detection.* Credit Scoring and Credit Control VII, 2001.

Brause, R., Langsdorf, T., and Hepp, M. "Neural data mining for credit card fraud detection," *Proceedings 11th International Conference on Tools with Artificial Intelligence*, 1999, pp. 103-106, doi:10.1109/TAI.1999.809773.

Brockett, Patrick. L., Cooper, W. W., Golden, L. L., and Pitaktong, U. "A Neural Network Method for Obtaining an Early Warning of Insurer Insolvency." *The Journal of Risk and Insurance* 61, no. 3 (1994): 402-24. doi:10.2307/253568.

Ching, K., Huang, X., Ng, K., and Siu, K. *Markov Chains: Models, Algorithms and Applications*. Springer, Heidelberg, 2013.

Cox, D. *Handbook of anti-money laundering*. John Wiley and Sons, Ltd, 2014.

Dorronsoro, J. R., Ginel, F., Sgnchez, C., and Cruz, C. S. "Neural fraud detection in credit card operations," in *IEEE Transactions on Neural Networks*, vol. 8, no. 4, pp. 827-834, July 1997, doi:10.1109/72.595879.

Dowle, M., and Srinivasan, A. data.table: Extension of “data.frame”, 2021. https://github.com/Rdatatable/data.table

Dunning, T., and Friedman, E. *Practical Machine Learning: A new Look at anomaly Detection.* O’Reilly Media, Inc, 2014, pp. 14.

European Commission. “Anti-money laundering and counter terrorist financing.” Accessed May 10, 2021. https://ec.europa.eu/info/business-economy-euro/banking-and-finance/financial-supervision-and-risk-management/anti-money-laundering-and-counter-terrorist-financing/eu-policy-high-risk-third-countries_en

Fawcett, T., and Provost, F. “Adaptive fraud detection.” *Data Mining and Knowledge Discovery*, vol. 13, pp. 291-316, 1997, doi:10.1023/A:1009700419189.

Financial Action Task Force. “Money Laundering.” Accessed March 2021. https://www.fatf-gafi.org/faq/moneylaundering/

Forney, D. *The Viterbi Algorithm: A Personal History.* University of Southern California. Libraries, 2005.

Ghosh and Reilly. "Credit card fraud detection with a neural-network," *1994 Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences*, 1994, pp. 621-630, doi:10.1109/HICSS.1994.323314.

Hartigan, J. A., and Wong, M. A. "Algorithm AS 136: A K-Means Clustering Algorithm." *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28, no. 1 (1979): 100-08, doi:10.2307/2346830.

Hawkins, D. (1980). *Identification of Outliers.* Springer Netherlands,1980.

Himmelmann, L. HMM: HMM - Hidden Markov Models. R package version 1.0, 2010. https://CRAN.R-project.org/package=HMM

International Monetary Fund. “Past IMF Staff Assessments on Offshore Financial Centers.” Accessed March, 2021. https://www.imf.org/external/NP/ofca/OFCA.aspx

Jadhav, S., and Bhandari, K. “Anomaly Detection Using Hidden Markov Model.” *International Journal of Computational Engineering Research*, 2013, vol. 3, pp. 28-35.

Jurafsky, D., and Martin, J. *Speech and Language Processing.* Pearson Prentice Hall, 2008.

Kasianova, K. "*Detecting Money laundering Using Hidden Markov Models."* Master's thesis., University of Tartu, 2020.

Li, J., Huang, KY., Jin, J., and Shi, J. "A survey on statistical methods for health care fraud detection." *Health Care Manage Sci* 11**,** pp. 275–287, 2008.

Madinger, J. *Money laundering: A guide for Criminal Investigators.* CRC Press, 2012, pp. 8.

Mehrotra**,** K. G., Mohan**,** C., and Huaming, H. *Anomaly Detection Principles and Algorithms.* Springer,2017, doi:10.1007/978-3-319-67526-8.

Mhamane, S., Lobo, L. "Use of Hidden Markov Model as Internet Banking Fraud Detection." *International Journal of Computer Applications*, 2012, vol 45.

Morris-Cotterill, Nigel. "Money Laundering." *Foreign Policy*, 2001, no. 124, pp 16-22. doi:10.2307/3183186.

Muller, H. W., Kalin, H. C., Goldsworth, G. J. *Anti-Money Laundering : International Law and practice.* John Wiley & Sons, Ltd, 2007.

Ourston, D., Matzner, S., Stump, W., and Hopkins, B (2003). "Applications of hidden Markov models to detecting multi-stage network attacks," *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*, 2003, pp. 10 pp.-, doi:10.1109/HICSS.2003.1174909.

Rabiner, L. R. "A tutorial on hidden Markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-286, Feb. 1989, doi:10.1109/5.18626.

Sullivan, K. *Anti–Money Laundering in a Nutshell: Awareness and Compliance for Financial Personnel and Business Managers.* Apress, 2015.

Tableau Software. Version 2021.1.1. https://www.tableau.com/

The United Nations Office on Drugs and Crime (2021). "Money laundering" Accessed April 2021. https://www.unodc.org/unodc/en/money-laundering/overview.html

Unger, B. *Scale and impacts of Money Laundering*. Edward Elgar Publishing Limited, 2007.

Wickham, H., François, R., Henr, L., and Müller, K. dplyr: A Grammar of Data Manipulation. R package version 1.0.5. https://CRAN.R-project.org/package=dplyr

Wu, J. *Advances in K-means Clustering.* Springer-Verlag Berlin Heidelberg, 2012.

Appendices

Appendix A. List of high-risk countries

Country code	Country Name	Country code	Country Name	Country code	Country Name
AF	Afghanistan	GY	Guyana	WG	Grenada
AI	Anguilla	HK	Hong Kong	SC	Seychelles
AG	Antigua and Barbuda	IR	Iran	SX	Sint Maarten
AW	Aruba	IQ	Iraq	SO	Somalia
PT-20	Azores	IM	Isle of Man	SS	South Sudan
BS	Bahamas	JA	Jamaica	LK	Sri Lanka
BH	Bahrain	JE	Jersey	SD	Sudan
BB	Barbados	KE	Kenya	SZ	Swaziland (Eswatini)
BY	Belarus	LA	Lao People's Democratic Republic	SY	Syria
BZ	Belize	LB	Lebanon	PF	Tahiti (French Polynesia)
BM	Bermuda	LR	Liberia	TL	Timor-Leste
BA	Bosnia and Herzegovina	LY	Libya	TO	Tonga
BN	Brunei Darussalam	MO	Macao	TT	Trinidad and Tobago
BF	Burkina Faso	PT-30	Madeira	TN	Tunisia
KH	Cambodia	MV	Maldives	TC	Turks and Caicos Islands
KY	Cayman Islands	MH	Marshall Islands	UG	Uganda
CF	Central African Republic	MU	Mauritius	UY	Uruguay
CG	Congo	MS	Montserrat	VU	Vanuatu
CK	Cook Islands	MZ	Mozambique	VE	Venezuela
CI	Cote d'Ivoire	MM	Myanmar (Burma)	VG	Virgin Islands, British
CU	Cuba	NA	Namibia	VI	Virgin Islands, U.S.
CW	Curacao	NR	Nauru	YE	Yemen
KP	Democratic People's Republic of Korea (DPRK)	NC	New Caledonia	DW	Guinea Bissau
DJ	Djibouti Republic	NU	Niue	RS	Serbia

DM	Dominica	PK	Pakistan	GT	Guatemala
DO	Dominican Republic	PW	Palau	GG	Guernsey
EC	Ecuador	PS	Palestine State of	VC	Saint Vincent and the Grenadines
ER	Eritrea	PA	Panama	WS	Samoa
ET	Ethiopia	RU	Russian Federation		
GH	Ghana	SH	Saint Helena, Ascension and Tristan da Cunha		
GI	Gibraltar	KN	Saint Kitts and Nevis		

Appendix B. R code

```
library(HMM)
library(data.table)
library(dplyr)
library(stats)
library(factoextra)

##
View(all_dd)##dataset for study 1
all_dd=data.table(all_dd)
nrow(all_dd)#5000
setkey(all_dd,user_id)
NROW(unique(all_dd$user_id))#10 users and 5 transaction is suspicious.0.1% of all 5000
transaction is suspicious


#defining conditions and adding new variable score

all_dd[,score:=0] #creating empty score column
all_dd[as.ITime(date_completed)<=as.ITime('07:00:00'),score:=score+15 ]
all_dd[as.ITime(date_completed)>=as.ITime('21:00:00'),score:=score+15]
all_dd[amount_in_eur>=500 & amount_in_eur<1000, score:=score+5]
all_dd[amount_in_eur>=1000 & amount_in_eur<5000, score:=score+10]
all_dd[amount_in_eur>=5000 & amount_in_eur<20000, score:=score+15]
all_dd[amount_in_eur>=20000, score:=score+20]
counterparty_country=substr(all_dd$sender_account,start=1, stop=2)
all_dd[,counterparty_country :=counterparty_country]
all_dd[counterparty_country %in% c('AF','AI','AG','AW','PT-
20','BS','BH','BB','BY','BZ','BM','BA','BN','BF','KH','KY','CF','CG','
CK','CI','CU','CW','KP','DJ','DM',

'DO','EC','ER','ET','GH','GI','WG','GT','GG','DW','GY','HK','IR',

'IQ','IM','JA','JE','KE','LA','LB','LR','LY','MO','PT-30','MV','MH',

'MU','MS','MZ','MM','NA','NR','NC','NU','PK','PW','PS','PA','SC','RU',

'WS','RS','SX','SO','SS','LK','SH','KN','PM','VC','SD','SZ','SY','PF',

'TL','TO','TT','TN','TC','UG','UY','VU','VE','VG','VI','YE'),
       score:=score+5]


all_dd[to_cur!='EUR'& to_cur!='USD', score:=score+10]
all_dd[from_cur!='EUR'& from_cur!='USD', score:=score+10]

#creating additional variables sum_1in,sum_1out and count_1

all_dd[,sum_1in:=0] # sum of amount in eur in last 1 days for incoming transactions

all_dd[,sum_1out:=0] # sum of amount in eur in last 1 days for outgoing transactions

all_dd[,count_1:=0] # count of transaction in last 1 days
```

```
for (i in 1:nrow(all_dd)) {

    all_dd$sum_1in[i]=sum(all_dd[difftime(all_dd$date_completed[i], date_completed,
                                  units='days')<=1
&(date_completed<=date_completed[i])&(user_id[i]==user_id) & type=='I',
                          amount_in_eur])}
  for (i in 1:nrow(all_dd)){
    all_dd$sum_1out[i]=sum(all_dd[difftime(all_dd$date_completed[i], date_completed,
                                   units='days')<=1
&(date_completed<=date_completed[i])&(all_dd$user_id[i]==user_id) & type=='O',
                           amount_in_eur])}
  for (i in 1:nrow(all_dd)){
    all_dd$count_1[i]=length(all_dd[difftime(all_dd$date_completed[i], date_completed,

units='days')<=1&(date_completed<=date_completed[i])&(all_dd$user_id[i]==user_id) ,
amount_in_eur
    ])}


#analyzing sum_1in,sum_1out and count_1
summary(all_dd$sum_1in)
summary(all_dd$sum_1out)
summary(all_dd$count_1)

#
all_dd[sum_1in>=1000 & sum_1in<5000, score:=score+5]
all_dd[sum_1in>=5000 & sum_1in<10000, score:=score+10]
all_dd[sum_1in>=10000 , score:=score+15]
all_dd[sum_1out>=500 & sum_1out<1000, score:=score+5]
all_dd[sum_1out>=1000 & sum_1out<10000, score:=score+10]
all_dd[sum_1out>=10000 , score:=score+15]
all_dd[count_1>=3 & count_1<5, score:=score+5]
all_dd[count_1>=5 & count_1<10, score:=score+10]
all_dd[count_1>=10 , score:=score+15]


 #creating max_score which is max score for each user_id
max_score=all_dd[,max_score:=0]
  for (i in 1:nrow(all_dd)){all_dd$max_score[i]=
max(all_dd[all_dd$user_id[i]==user_id,score])} # maximum score for person


#creating observable variable
 all_dd[score<max_score*1/2 | max_score==0, observation:='low_risk']
 all_dd[score>=max_score*1/2 & score< max_score*9/10 , observation:='medium_risk']
 all_dd[score>=max_score*9/10, observation:='high_risk']

 View(all_dd)
```

```
#defining parameters for HHM
  states <- c("normal", "susp") # define the names of the states
  normprobs <- c(0.9, 0.1) # set the probabilities of switching states, where the
previous state was "normal"#does not matter

  suspprobs <- c(0.4, 0.6) # set the probabilities of switching states, where the
previous state was "susp"

  thetransitionmatrix <- matrix(c(normprobs, suspprobs), 2, 2, byrow =
                                  TRUE) # create a 2 x 2 matrix
  rownames(thetransitionmatrix) <- states
  colnames(thetransitionmatrix) <- states
  observations <- c("low_risk","medium_risk","high_risk") # define the alphabet of
observations

  normstateprobs <- c(0.7,0.25,0.05) # set the values of the emission probabilities, for
the normal state

  suspstateprobs <- c(0.01,0.09,0.9) # set the values of the emission probabilities,for
the susp state

  theemissionmatrix <- matrix(c(normstateprobs, suspstateprobs),2, 3,
                              byrow = TRUE) # Create a 2 x 3 matrix
  rownames(theemissionmatrix) <- states
  colnames(theemissionmatrix) <- observations
  myseq<- all_dd$observation # create a vector of observable variable
  #initialization HMM
  hmm = initHMM(c("normal","susp"),observations,
                transProbs=thetransitionmatrix,
                emissionProbs=theemissionmatrix)
  # Baum-Welch algorithm for updating transition and emission probabilities

  bw = baumWelch(hmm,myseq,5)
  thetransitionmatrix<-bw$hmm$transProbs
  theemissionmatrix<-bw$hmm$emissionProbs
# using Viterbi algorithm to predict state for every transaction

res=viterbi(hmm,myseq)#getting the hidden states based on most probable path by using
built in viterbi algorithm of HMM package


#adding states column to the dataset

all_dd[,states:=res]


#Results
all_dd[states=="susp"]#9 suspicious transactions HMM identified
```

```
#Applying Kmeans on dataset

Xkmeans=all_dd%>%select(amount_in_eur,score,max_score,sum_1in,sum_1out)#taking into
account 4 variables


Y=kmeans(Xkmeans,2)#using stats package



table(Y$cluster,all_dd$meta_sar_id)#


###results HMM
TPHMM=nrow(all_dd[meta_sar_id>0&states=="susp"])#4    #True positives
FPHMM=nrow(all_dd[meta_sar_id==0&states=="susp"])#5  #False positives
FNHMM=nrow(all_dd[meta_sar_id>0&states!="susp"])#1   #False Negatives


#precision
Precision_HMM=TPHMM/(TPHMM+FPHMM)#0.4444444
Precision_HMM
#sensitivity
Sensitivity_HMM=TPHMM/(TPHMM+FNHMM)#80%
Sensitivity_HMM
#F score
F_scoreHMM=(2*Sensitivity_HMM*Precision_HMM)/(Sensitivity_HMM+Precision_HMM)##0.5714286
F_scoreHMM

# Results of kmeans
TPkmeans=5
FPkmeans=4980
FNkmeans=0

Precision_kmeans=TPkmeans/(TPkmeans+FPkmeans)
Precision_kmeans## 0.001003009
Sensitivity_kmeans=TPkmeans/(TPkmeans+FNkmeans)#1
Sensitivity_kmeans#1
F_score_kmeans=(2*Precision_kmeans*Sensitivity_kmeans)/(Sensitivity_kmeans+Precision_kmea
ns)
F_score_kmeans#0.002004008
```

```
###Study 2
all_dd_s=data.table(all_dd_s)###

nrow(all_dd_s)#6854 suspicious transactions of 439 users
NROW(unique(all_dd_s$user_id))#439 unique users all suspicious

#setting up the rules for score
all_dd_s[,score:=0] #creating empty score column
all_dd_s[as.ITime(date_completed)<=as.ITime('07:00:00'),score:=score+15 ]
all_dd_s[as.ITime(date_completed)>=as.ITime('21:00:00'),score:=score+15]
all_dd_s[amount_in_eur>=500 & amount_in_eur<1000, score:=score+5]
all_dd_s[amount_in_eur>=1000 & amount_in_eur<5000, score:=score+10]
all_dd_s[amount_in_eur>=5000 & amount_in_eur<20000, score:=score+15]
all_dd_s[amount_in_eur>=20000, score:=score+20]
counterparty_country=substr(all_dd_s$sender_account,start=1, stop=2)
all_dd_s[,counterparty_country :=counterparty_country]
all_dd_s[counterparty_country %in% c('AF','AI','AG','AW','PT-
20','BS','BH','BB','BY','BZ','BM','BA','BN','BF','KH','KY','CF','CG','
CK','CI','CU','CW','KP','DJ','DM',

'DO','EC','ER','ET','GH','GI','WG','GT','GG','DW','GY','HK','IR',
                                    'IQ','IM','JA','JE','KE','LA','LB','LR','LY','MO','PT-
30','MV','MH',

'MU','MS','MZ','MM','NA','NR','NC','NU','PK','PW','PS','PA','SC','RU',

'WS','RS','SX','SO','SS','LK','SH','KN','PM','VC','SD','SZ','SY','PF',

'TL','TO','TT','TN','TC','UG','UY','VU','VE','VG','VI','YE'),
       score:=score+5]

all_dd_s[to_cur!='EUR'& to_cur!='USD', score:=score+10]
all_dd_s[from_cur!='EUR'& from_cur!='USD', score:=score+10]

#creating additional variables sum_1in,sum_1out and count_1

all_dd_s[,sum_1in:=0] # sum of amount in eur in last 1 days for incoming transactions

all_dd_s[,sum_1out:=0] # sum of amount in eur in last 1 days for outgoing transactions

all_dd_s[,count_1:=0] # count of transaction in last 1 days
for (i in 1:nrow(all_dd_s)) {

  all_dd_s$sum_1in[i]=sum(all_dd_s[difftime(all_dd_s$date_completed[i], date_completed,
                                           units='days')<=1
&(date_completed<=date_completed[i])&(user_id[i]==user_id) & type=='I',
                                 amount_in_eur])}
for (i in 1:nrow(all_dd_s)){
  all_dd_s$sum_1out[i]=sum(all_dd_s[difftime(all_dd_s$date_completed[i], date_completed,
                                            units='days')<=1
&(date_completed<=date_completed[i])&(all_dd_s$user_id[i]==user_id) & type=='O',
                                  amount_in_eur])}
for (i in 1:nrow(all_dd_s)){
  all_dd_s$count_1[i]=length(all_dd_s[difftime(all_dd_s$date_completed[i],
date_completed,
units='days')<=1&(date_completed<=date_completed[i])&(all_dd_s$user_id[i]==user_id) ,
amount_in_eur
  ])}
```

```
#analyzing sum_1in,sum_1out and count_1
summary(all_dd_s$sum_1in)
summary(all_dd_s$sum_1out)
summary(all_dd_s$count_1)

# conditions for score
all_dd_s[sum_1in>=1000 & sum_1in<5000, score:=score+5]
all_dd_s[sum_1in>=5000 & sum_1in<10000, score:=score+10]
all_dd_s[sum_1in>=10000 , score:=score+15]
all_dd_s[sum_1out>=500 & sum_1out<1000, score:=score+5]
all_dd_s[sum_1out>=1000 & sum_1out<10000, score:=score+10]
all_dd_s[sum_1out>=10000 , score:=score+15]
all_dd_s[count_1>=3 & count_1<5, score:=score+5]
all_dd_s[count_1>=5 & count_1<10, score:=score+10]
all_dd_s[count_1>=10 , score:=score+15]


#creating max_score_s which is max score for each user_id
max_score_s=all_dd_s[,max_score_s:=0]
for (i in 1:nrow(all_dd_s)){all_dd_s$max_score_s[i]=
max(all_dd_s[all_dd_s$user_id[i]==user_id,score])} # maximum score for person



##
#creating observable variable
all_dd_s[score<max_score_s*1/3 | max_score_s==0, observation:='low_risk']
all_dd_s[score>=max_score_s*1/3 & score< max_score_s*2/3 , observation:='medium_risk']
all_dd_s[score>=max_score_s*2/3, observation:='high_risk']

##Setting up HMM

states <- c("normal", "susp") # define the names of the states
normprobs <- c(0.9, 0.1) # set the probabilities of switching states, where the previous
state was "normal"#does not matter

suspprobs <- c(0.4, 0.6) # set the probabilities of switching states, where the previous
state was "susp"

thetransitionmatrix <- matrix(c(normprobs, suspprobs), 2, 2, byrow =
                                TRUE) # create a 2 x 2 matrix
rownames(thetransitionmatrix) <- states
colnames(thetransitionmatrix) <- states
observations <- c("low_risk","medium_risk","high_risk") # define the alphabet of
observations

normstateprobs <- c(0.7,0.25,0.05) # set the values of the emission probabilities, for
the normal state

suspstateprobs <- c(0.01,0.09,0.9) # set the values of the emission probabilities,for the
susp state
theemissionmatrix <- matrix(c(normstateprobs, suspstateprobs),2, 3,
                            byrow = TRUE) # Create a 2 x 3 matrix
rownames(theemissionmatrix) <- states
colnames(theemissionmatrix) <- observations
myseq<- all_dd_s$observation # create a vector of observable variable
```

```
#initialization HMM
hmm = initHMM(c("normal","susp"),observations,
              transProbs=thetransitionmatrix,
              emissionProbs=theemissionmatrix)
# Baum-Welch algorithm for updating transition and emission probabilities

bw = baumWelch(hmm,myseq,5)
thetransitionmatrix<-bw$hmm$transProbs
theemissionmatrix<-bw$hmm$emissionProbs
# using Viterbi algorithm to predict state for every transaction

res=viterbi(hmm,myseq)#getting the hidden states based on most probable path by using
built in viterbi algorithm of HMM package

##
#adding states to the dataset
all_dd_s[,states:=res]


#Results  of HMM
all_dd_s[states=="susp"] #5771 suspicious transactions HMM identified


###results HMM
TPHMM_s=nrow(all_dd_s[meta_sar_id>0&states=="susp"])#5771    #True positives
TPHMM_s

FPHMM_s=nrow(all_dd_s[meta_sar_id==0&states=="susp"])#0  #False positives
FPHMM_s

FNHMM_s=nrow(all_dd_s[meta_sar_id>0&states!="susp"])#1083   #False Negatives
FNHMM_s

#precision
Precision_HMM_s=TPHMM_s/(TPHMM_s+FPHMM_s)#1
Precision_HMM_s
#sensitivity
Sensitivity_HMM_s=TPHMM_s/(TPHMM_s+FNHMM_s)# 0.8419901
Sensitivity_HMM_s
#F score
F_scoreHMM_s=(2*Sensitivity_HMM_s*Precision_HMM_s)/(Sensitivity_HMM_s+Precision_HMM_s)##0
.5714286
F_scoreHMM_s  # 0.9142178

Application and results of K means

 ##Kmeans, for study 2 as all transactions were suspicious we assume that kmeans assumes
transactions as suspicious in highest number of cluster.
 Xkmeans=all_dd_s%>%select(amount_in_eur,score,max_score_s,sum_1in,sum_1out)#taking into
account 4 variables



 Y=kmeans(Xkmeans,2)#using stats package


 table(Y$cluster,all_dd_s$meta_sar_id)#
```

```
#kmeans q.a.m
 TPkmeans_s=5613 #out of 6854
 FPkmeans_s=0
 FNkmeans_s=1241

 Precision_kmeans_s=TPkmeans_s/(TPkmeans_s+FPkmeans_s)
 Precision_kmeans_s##1
 Sensitivity_kmeans_s=TPkmeans_s/(TPkmeans_s+FNkmeans_s)# 0.6865128
 Sensitivity_kmeans_s#0.8189378

F_score_kmeans_s=(2*Precision_kmeans_s*Sensitivity_kmeans_s)/(Sensitivity_kmeans_s+Precis
ion_kmeans_s)
 F_score_kmeans_s#0.9004572

View(transaction)###study 3

transaction=data.table(transaction)

transaction[,c("date_created","receiver_account","cur_rate","reference_text","eur_rate","
sender_name","receiver_name","counterparty_name","counterparty_swift",
       "counterparty_institution_name","transaction_type"):=NULL]

setkey(transaction,user_id)
nrow(transaction[meta_sar_id>0])#235 suspicious transactions overall
nrow(transaction)#55275 overall, 0.425147 % suspicious transaction in dataset

transaction[,score:=0] #creating empty score column
transaction[as.ITime(date_completed)<=as.ITime('09:00:00'),score:=score+15 ]
transaction[as.ITime(date_completed)>=as.ITime('21:00:00'),score:=score+15]
transaction[amount_in_eur>=500 & amount_in_eur<1000, score:=score+5]
transaction[amount_in_eur>=1000 & amount_in_eur<3000, score:=score+10]
transaction[amount_in_eur>=3000 & amount_in_eur<10000, score:=score+15]
transaction[amount_in_eur>=10000, score:=score+20]
counterparty_country=substr(transaction$sender_account,start=1, stop=2)
transaction[,counterparty_country :=counterparty_country]
transaction[counterparty_country %in% c('AF','AI','AG','AW','PT-
20','BS','BH','BB','BY','BZ','BM','BA','BN','BF','KH','KY','CF','CG','
CK','CI','CU','CW','KP','DJ','DM',

'DO','EC','ER','ET','GH','GI','WG','GT','GG','DW','GY','HK','IR',
                                'IQ','IM','JA','JE','KE','LA','LB','LR','LY','MO','PT-
30','MV','MH',

'MU','MS','MZ','MM','NA','NR','NC','NU','PK','PW','PS','PA','SC','RU',

'WS','RS','SX','SO','SS','LK','SH','KN','PM','VC','SD','SZ','SY','PF',

'TL','TO','TT','TN','TC','UG','UY','VU','VE','VG','VI','YE'),
   score:=score+5]


transaction[to_cur!='EUR'& to_cur!='USD', score:=score+10]
transaction[from_cur!='EUR'& from_cur!='USD', score:=score+10]
transaction[sender_account==""&counterparty_country=="", score:=score+10]
```

```
##creating max score for each user
max_score=transaction[,max_score:=0] #new column

for (i in 1:nrow(transaction)){transaction$max_score[i]=
max(transaction[transaction$user_id[i]==user_id,score])} # maximum score for person

summary (transaction$max_score)

#creating observable variable
transaction[score<max_score*1/2 | max_score==0, observation:='low_risk']
transaction[score>=max_score*1/2 & score< max_score*9/10 , observation:='medium_risk']
transaction[score>=max_score*9/10, observation:='high_risk']

###Applying HMM
states <- c("normal", "susp") # define the names of the states
normprobs <- c(0.9, 0.1) # set the probabilities of switching states, where the previous
state was "normal"

suspprobs <- c(0.4, 0.6) # set the probabilities of switching states, where the previous
state was "susp"

thetransitionmatrix <- matrix(c(normprobs, suspprobs), 2, 2, byrow =
                                TRUE) # create a 2 x 2 matrix
rownames(thetransitionmatrix) <- states
colnames(thetransitionmatrix) <- states
observations <- c("low_risk","medium_risk" ,"high_risk") # define the alphabet of
observations

normstateprobs <- c(0.7,0.25,0.05) # set the values of the emission probabilities, for
the normal state

suspstateprobs <- c(0.01,0.09,0.9) # set the values of the emission probabilities,for the
susp state

theemissionmatrix <- matrix(c(normstateprobs, suspstateprobs), 2, 3,
                            byrow = TRUE) # Create a 2 x 3 matrix
rownames(theemissionmatrix) <- states
colnames(theemissionmatrix) <- observations
myseq<- transaction$observation # create a vector of observable variable


#initialization HMM
hmm = initHMM(c("normal","susp"),observations,
              transProbs=thetransitionmatrix,
              emissionProbs=theemissionmatrix)


# Baum-Welch algorithm for updating transition and emission probabilities

bw = baumWelch(hmm,myseq,5)
thetransitionmatrix<-bw$hmm$transProbs
theemissionmatrix<-bw$hmm$emissionProbs

# using Viterbi algorithm to predict state for every transaction
res=viterbi(hmm,myseq)

#adding state column
transaction[,state:=res]
```

```
#coding all susp. transaction as 1 in meta_sar_id
transaction$meta_sar_id=(transaction$meta_sar_id != 0)*1

TPHMM_all=nrow(transaction[meta_sar_id>0&state=="susp"])#95 out of 235
TPHMM_all
FPHMM_all=nrow(transaction[meta_sar_id==0&state=="susp"])#83
FPHMM_all
FNHMM_all=nrow(transaction[meta_sar_id>0&state=="normal"])#140
FNHMM_all



Precision_HMM_all=TPHMM_all/(TPHMM_all+FPHMM_all)#0.5337079
Precision_HMM_all

Sensitivity_HMM_all=TPHMM_all/(TPHMM_all+FNHMM_all)#0.4042553
Sensitivity_HMM_all

F_scoreHMM_all=(2*Sensitivity_HMM_all*Precision_HMM_all)/(Sensitivity_HMM_all+Precision_H
MM_all)## 0.4600484
F_scoreHMM_all



##Kmeans
Xkmeans=transaction%>%select(amount_in_eur,score,max_score)#taking into account 4
variables


Y=kmeans(Xkmeans,2)#using stats package

table(Y$cluster,transaction$meta_sar_id)#

#kmeans
TPkmeans_t=184 #out of 55275
FPkmeans_t=54537
FNkmeans_t=51

Precision_kmeans_t=TPkmeans_t/(TPkmeans_t+FPkmeans_t)
Precision_kmeans_t#0.003362512
Sensitivity_kmeans_t=TPkmeans_t/(TPkmeans_t+FNkmeans_t)# 0.6865128
Sensitivity_kmeans_t#0.7829787
F_score_kmeans_t=(2*Precision_kmeans_t*Sensitivity_kmeans_t)/(Sensitivity_kmeans_t+Precis
ion_kmeans_t)
F_score_kmeans_t#0.006696266
```

Non-exclusive licence to reproduce thesis and make thesis public.

I, Ismayil Aghahasanli,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the terof copyright, **Detecting money laundering in transaction monitoring using hidden Markov model**, supervised by Kaur Lumiste.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Ismayil Aghahasanli

14/06/2021