Master Thesis

MSc Economics and Business

Data Science & Marketing Analytics

Detection of Money Laundering Transaction Network Structures and Typologies using Machine Learning Techniques

Author: Floris Visser

Student Number: 406508

Supervisor: Michel van de Velden

Second Assessor: Arash Yazdiha

KPMG Supervisor: Jiri Brummer

Submission Date Final Draft: 03-05-2020

Abstract

This thesis researches the detection of money laundering transactions network structures and typologies using machine learning techniques. These techniques have potential benefits over time consuming human investigations to detect money laundering transactions. Four machine learning models – Decision Tree, Conditional Inference Tree, Random Forests, Neural Network – are researched on synthetic data having various different combinations of money laundering typologies. This thesis contributes to existing research by comparing the performance of these four machine learning models, by comparing the performance on various money laundering typologies, and by an analysis of the performance of the models on data using an oversampling technique. Results indicate that – although some minor problems have to be solved - machine learning models can be used for the detection of money laundering transactions. Further research should be done on the fine tuning of each model to achieve better performance. Furthermore, the models should be trained and tested on real data to compare the performance on a real-life scenario.

Keywords: Machine learning, money laundering, fraud detection, F1-score, synthetic data.

Acknowledgements

I would like to thank all the people that assisted me through this thesis project. First of all, I would like to thank KPMG for giving me the opportunity to write my thesis at the Forensic Technology Department. In particular I want to thank Jiri Brummer, my supervisor at KPMG, for supporting me throughout the process with useful and interesting feedback. Furthermore, I want to thank Patrick Özer for planning and inviting me to a very interesting meeting with Rabobank. Secondly, I want to sincerely thank professor Michel van de Velden for being my supervisor at the Erasmus University of Rotterdam and helping me with the problems faced during the process. At last, I would like to thank ING for giving me the opportunity to attend a very intriguing meeting concerning the detection of money laundering.

Table of Contents

1. Introduction

The rise of technology forces us to constantly adapt in many different industries. Technology has been important for financial and banking systems, changing the structure of such institutions. Innovations improving the ease of internet payments and transactions have majorly impacted the services provided by financial institutions. New technologies are used to create a faster, safer, and more convenient online environment for their services (Van Vlasselaer, et al., 2015). Therefore, saving time and money, promoting financial markets, and increasing the easiness of turnaround capital in the labor market. Simultaneously, with the growth of online payments and mobile payments in particular, an attractive environment for fraudsters is created. Accordingly, in combination with the pressure from De Nederlandsche Bank (2017) and politics, the role of financial institutions is converging towards an "e-bank" – an electronical bank functioning as a gateway for transactions - with multiple controlling responsibilities to detect fraudsters. E-banks face – among other challenges such as regulatory, operational efficiency and strategic (Flaunet, 2019) - new challenges, which are detection of fraud (Barraclough, Hosssain, Tahir, Sexton, & Aslam, 2013) and especially money laundering (Choo, 2015).

E-banks have to fulfill a crucial role, since money laundering is seen as one of the main problems to continue with a reliable international financial system. It damages the legal world economy since the money is earned illegally, prevention of money laundering brings high investigation costs due to the detection monitoring systems, and it promotes weak detection policies considering weaker policies are more attractive and accessible to launder money. Eventually, financial markets could become unstable, and ultimately, it could affect economic growth (Omar, Amirah Johari, & Arshad, 2014). Additionally, money laundering is often used as a gateway to finance terroristic activities, hurting our society financially and politically. According to the International Monetary Fund (IMF), the size of money laundering worldwide is approximately five percent of the global gross domestic product (GDP), similar to 3,000 billion US dollars.

As aforementioned, the detection of suspicious financial transactions has become an essential task for financial institutions. Due to the privacy-sensitive nature of financial transactions, very few institutions have the right to fulfill the monitoring role. Therefore, financial institutions that have a banking license are obliged to function as a gatekeeper to detect suspicious financial transactions. However, transaction monitoring to detect financial transactions involves tasks

that are resource-intensive, tedious, and time-consuming that require many employees to screen and analyze all suspicious transactions. For instance, Rabobank has 2600 employees on its payroll to detect suspicious financial transactions and possible criminal clients (Wiessing, 2019). As a tool to support the monitoring process, software programs exist to pre-define specific rules to classify transactions as suspicious or not, called the rule-based monitoring method.

A problem concerning this rule-based method is the inability to define a single set of rules to detect all different money laundering typologies – explained in more detail below - and relations because of the complexity and wide variety of these patterns (Watkins, et al., 2003). A consequence of this inability is the risk of undetected actual money laundering transactions being classified as normal transactions, defined as false-negatives. In addition, rule-based methods are incapable of detecting novel suspicious transaction structures within the financial transaction network. Solely, known patterns and structures of money laundering can be defined using the rule-based method. To minimize the missed instances of money laundering transactions, strict rules are applied, resulting in many transactions that are falsely classified as suspicious, defined as false-positives. The false-positive rate (FPR), is the number of normal transactions falsely detected as money laundering transactions of the total transactions detected. The false-positive rate is used as a measurement of detection methods. A high FPR is inefficient due to the thorough investigation required of each generated alert, causing large amounts of data to be analyzed.

More efficient methods of detecting financial transactions could lower the false-positive rate - while decreasing or maintaining the number of false negatives - and potentially more accurately detect suspicious transaction typologies within the financial transaction network. A typology is a term used in the field of anti-money laundering. According to the International Monetary Fund (IMF) typologies refer to a particular combination of patterns, structures and techniques used to launder money or finance terrorism (IMF, 2020). These typologies are heavily influenced by anti-money laundering regimes, financial markets, the economy, geographical location and time. Typologies include patterns and structures used of money laundering between financial bank accounts. The term typology is used in this thesis since it is used in the anti-money laundering context. Methods that can be a valuable contribution to the current complications of money laundering detection, are machine learning methods. Introducing new methods can bring new insights, reveal patterns that beforehand were unknown, and may predict suspicious transactions more accurately. On the other hand, machine learning methods

have limitations that may affect the detection of financial transactions negatively. Discovering money laundering transactions remains a complicated task due to the complex structures and patterns of these transactions. Since machine learning methods have potential benefits and limitations to the money laundering detection process, this research aims to examine and identify these benefits and limitations, and examines if and how machine learning can complement to the rule-based method analyzing the financial network typologies. This leads to the following research question:

> *"What are the benefits and limitations of machine learning techniques applied in the Anti-Money Laundering financial transaction network typology detection process?"*

In order to substantiate the benefits and limitations of machine learning techniques, four sub-questions are formulated to answer the research question. The first sub-question relates to the typology of transactions within a financial transaction network. According to experts from financial institutions, most detection monitoring systems use rule-based methods in combination with certain indicators of money laundering instead of using more advanced techniques that incorporate multiple data combinations (ING, 2020). Since money laundering can occur through a wide variety of different typologies, it is interesting to examine the performance, advantages and disadvantages of machine learning techniques on the detection of various money laundering typologies. This leads to the first sub-question:

(1) *"How do machine learning techniques perform on various typologies of money laundering?"*

Machine learning techniques can be useful once the detection process - or part of the detection process - is optimized, enhanced or replaced. Once the machine learning models perform equally well as humans, machine learning techniques could partially replace the screening and analysis work done by employees. However, the complexity and wide variety of money laundering typologies create a challenging environment for machine learning models to perform well. In a real-world scenario, many different types of transaction behavior occur. This implies that it is difficult to detect specific patterns of money laundering, but how significant is the difference in the performance of machine learning models on a scenario with many money laundering typologies compared to a scenario that includes one typology of money laundering. This leads to the second sub-question:

(2) *"How do machine learning techniques perform on data where only one typology of money laundering exists compared to data with multiple money laundering typologies?*

Within money laundering typologies, each typology can differ regarding the period within the money laundering typology is performed and the number of interactions with other accounts when performing the money laundering typology. Normal transaction typologies have more extended periods and fewer account interactions than money laundering typologies (Drezewski, Filipkowski, & Sepielak, 2012). This characteristic implies that an adjustment to one of these factors would affect the performance of the machine learning models. This leads to the third sub-question:

(3) *"How do machine learning techniques perform on differences in the number of account interactions and periods wherein the typology is performed?"*

The detection of money laundering transactions has been monitored and investigated primarily by each financial institution individually due to legislative restrictions. However, since September 2019, the five major Dutch financial banking institutions decided to collaborate to combat money laundering (Wiessing, 2019). This implies exchanging client information, transaction data, and knowledge about anti-money laundering. Therefore, monitoring transactions between banks - something that is never done before - could significantly improve the detection of suspicious transactions. Exchanging information of transactions provides the opportunity to monitor network structures of transactions more accurately, potentially improving the detection monitoring systems. This collaboration aims to decrease the false-positive rate and more accurately detect suspicious transactions. Implying that the collaboration improves the detection of money laundering transactions. This leads to the fourth sub-question:

(4) *"What is the difference in the performance of machine learning models when detecting suspicious transactions within one bank compared to transaction monitoring between multiple banks?"*

This research aims to give new insights, concerning the detection of money laundering transactions, and predict suspicious financial transactions more accurately. This research is a contribution to existing work and research done in the field of money laundering for a couple of reasons. First, the research is focused on the detection of money laundering transactions examining the financial transaction network structures and provides an analysis of the effects of changes in certain parameters of money laundering typologies on machine learning techniques. Second, this thesis investigates the performance of machine learning techniques on various datasets with different money laundering typologies and comparing the performance of these machine learning models. Third, this thesis compares the differences in performance

of the models on imbalanced- and balanced money laundering transaction data.

The rest of this paper is organized as follows: The second section of this thesis presents the existing literature related to money laundering and the detection thereof. The third and fourth section explain the data and methodology used. Finally, the results, discussion conclusion are presented in the fifth, sixth and seventh section.

2. Theoretical Framework

Money laundering has been around for decades. It is argued that the first cases of money laundering originated from the 1930s, during the times of Al Capone (Unger, 2013). During the 1980s, the years of Pablo Escobar and the war on drugs, money laundering became a global problem causing economic and social problems. The global problems, scandals, and drug ramifications have led to the founding of the Financial Action Task Force (FATF), an organization created by the G-7[1]. The FATF is a Group of Seven Industrial Democracies, and tries to combat money laundering and its consequences. In 1990, the FATF published 40 recommendations of anti-money laundering standards. Additionally, they recommended implementing the Suspicious Activity Report (SAR), which is still used today. Although measures are taken to combat money laundering, methods to launder money became increasingly sophisticated and complex, as new measures arose and technology advanced (Gao & Xu, 2006). The constant battle between new complex, ever-changing money laundering typology and the detection of those typology seem like a never-ending story.

2.1. Money Laundering

It is essential to have a clear understanding of the process and definitions of money laundering to interpret the outcome of this research. The first challenge found in the literature is the wide interpretation of definitions used in the field of money laundering. It is known to be a difficult task to define a boundary between fraud and money laundering (Banirostam & Safari, 2018). Therefore, multiple variations of definitions are found in the literature to define both terms. According to Banirostam & Safari (2018), there is no exact and precise boundary between the two concepts and often are applied simultaneously. However, fraud is commonly used as a more general term to deliberate misrepresentation, and this could include money laundering (Nimmo, 2007). Whereas money laundering refers to *'The funneling of cash or other funds generated from illegal activities through legitimate financial institutions and businesses to conceal the source of the funds'* (Anti-Money Laundering, 2004). These activities could indicate anything related to account openings, large cash sums, deposits, withdrawals, payments, and wire transfers. Since this research aims to detect money laundering transactions using information from wire transfer network structures, the following definition from the simulator – explained in further detail in section 3.1 - is used: Suspicious Activity Report

[1] Including Canada, France, Germany, Italy, Japan, United Kingdom and the United States.

(SAR). A SAR is a group of transactions and accounts - accounts being bank accounts performing financial transactions – involved in a money laundering transaction typology. Therefore, a transaction in the dataset that is flagged as SAR, is a transaction used to launder money and should be detected. Furthermore, a definition is used for a potential money laundering transaction but requires further investigation, such a transaction is defined as an alert.

The second essential part is to understand the different phases of money laundering. In general, money laundering can be divided into three phases, broadly known as the placement, layering, and integration phase (Madinger & Kinnison, 2011). However, not all money laundering transactions are involved in all three phases. Still, it is useful to classify different phases in what can be a complicated process. The first phase of the process is placement. During the placement phase, according to the Board of Governors of the Federal Reserve System (Reuter, 2005), the money or other funds derived from illegal activities is physically moved to a place or into a less suspicious form to law enforcement authorities and extra advantageous to the criminal. The income then is introduced inside the (non)-traditional financial institutions or into the retail economy (Reuter, 2005). This insertion can be done by smuggling the money to another country, use security brokers or bank individuals to facilitate the process, purchase assets with cash, and through many other methods to get rid of cash. The second phase, layering, involves complex financial transactions to create sophisticated paths to obscure and hide the money from their illegal source, making it difficult for law enforcement agencies to reproduce where the money came from (Kharote, 2014). Layering methods used are: circulating money between multiple financial accounts, shifting the money among many different entities, or by reselling assets - bought in the placement phase - locally or abroad. The third and last phase is integration. The illegally obtained money is converted to apparently legitimate business revenues through normal financial operations. Being creative with the paperwork of the company's earnings or expenses ensures the money appears legal (Strandberg, 1997). This operation is also done by dealing with properties and creating false loans.

To better understand the phases of money laundering, the following example is provided. Consider a drug dealer receiving cash from his illegal business. He is not able to spend his money since it has an evident connection with illicit businesses. Therefore, specific measures have to be taken by the criminal to pretend the money is legally earned. First, the money is brought to a place where cash is exchanged to digital money or converted to assets.

For example, a bank with fewer restrictions or a casino, and afterward, the money is placed at the financial institution. Once placed, the money is layered by paying false invoices, loans, wire transfers, and other ways to create a complex path using different financial institutions in similar or different countries. To finalize the process, the money is transferred to legal business, and false paperwork emphasizes that it is legal. An example is the taxi business, where a taxi company with only a couple of cars makes millions of dollars. Spending the profits of the taxi company appears to be legal, and it is almost untraceable where the money originally came from.

As mentioned in the previous paragraph, not every money laundering transaction follows the same path. In comparison to other crimes, money laundering is known for its many different forms, players, and environments (Financial Intelligence Unit Belize, sd). It is argued that a characteristic common to most money laundering transaction patterns is the use of numerous transactions done by either natural or legal bank accounts among different entities in a relatively short period (Drezewski, Filipkowski, & Sepielak, 2012). Types of suspicious transactions to launder money occur in many different forms, such as the use of cash-, bank-, and investment transactions. Apart from that, many other forms such as offshore activities, the involvement of employees and agents working for financial institutions, secured and unsecured lending, sales and dealing staff, settlements, and company's management and formation involvement are used to launder money (Financial Intelligence Unit Belize, sd).

Since this research emphasizes on money laundering transactions using retail bank accounts and analyzing the typologies used by these accounts to launder money, many variations exist. For instance, a large number of individuals transferring money to one single account, having accounts at several institutions and pay cash to each of them, and circulate the flow of money between different financial institutions and countries. Nevertheless, the wide variety of typologies and the chance of undetected occurrences make it hard to draw conclusions on what typologies are often used for money laundering.

This research focuses on the second phase of money laundering, the layering phase. More specifically, the method of layering using multiple retail bank accounts in order to create difficult paths to stay undetected by law enforcement and banking investigations.

2.2. Detection of money laundering

De Nederlandsche Bank proposed specific guidelines to improve the detection and monitoring process of money laundering. Two guidelines to develop this process are approaches defined

as pre-transaction monitoring and post-event transaction monitoring, which conveys the monitoring is done before and after transactions are executed. Pre-transaction monitoring takes place before the transaction is executed; this is related to the face-to-face situations between the client and the bank employee (De Nederlandsche Bank, 2017). Two situations can occur when an alert is generated. The first, when monitoring is done beforehand, for instance, in situations when a client wants to deposit high amounts of cash or change high amounts of currencies. The second situation to generate an alert beforehand is done with the creation of a comprehensive client profile, including the transaction profile of clients containing the expected transaction behavior of the client. Once the client wants to transfer money outside of this client profile, the transaction can be detected in advance. A transaction profile should be actual, complete, specific, organized, substantiated, and stored (De Nederlandsche Bank, 2017).

Furthermore, a client profile consists of the type of client, type of service to the client, and client segment. The transaction process is customized for each client segment individually. Additionally, risk identification is created based on an analysis of the money laundering risks involved in different clients, products, distribution channels, and transactions. These have to be saved in the SIRA (systematic integrity risk analysis). The SIRA is a file containing information about the identification, analysis, and evaluation of integrity risks. It is possible to have multiple SIRA's. For example, for different business lines or a subsidiary company. Eventually, clients are organized in low, medium, and high-risk clients based on their expected transactions, risk identification, and client profile (De Nederlandsche Bank, 2017).

The second approach, post-event transaction monitoring, is focused on the detection of digital money transactions between bank accounts. Monitoring is done using specific detection rules for different scenarios to detect patterns and transactions of money laundering. Detection rules are based on country, client profile, and thresholds on the volume of transactions and transaction amounts. The detection rules combined are called a set of business rules. A different set of business rules should be generated for different client segments, countries, products, kinds of transactions, and distribution channels. However, more advanced methods, like artificial intelligence, can be switched to if the concerning institution demands it. According to these business rules and the outcome of more advanced methods, alert transactions are generated that need further investigation. The outcome of the thorough investigation then decides whether the transaction has to be reported to the Financial Intelligence Unit (FIU) of the Netherlands. Lastly, the banking institution has to evaluate the consequences of the reported

transaction to the FIU and decide if new measures have to be taken. All information should be saved to be able to reconstruct the situation for at least five years after the alert (De Nederlandsche Bank, 2017).

2.2.1. Rule-based method

Criminals laundering money do everything in their power to avoid being stapled as a high-risk client profile or perform transaction outside the thresholds set by financial institutions (Levi & Reuter, 2014). A before-mentioned approach to detect money laundering transactions to be discussed in more detail is the rule-based protocol approach. This approach analyzes transactions based on a predetermined set of rules. For example, all transactions coming from country B higher than €7.500,- are flagged as an alert and require further investigation. The transactions qualified for further investigation include the 'near misses'. These are proceedings containing the same value but spread over different transactions in a short period; for example, three transactions are done within 24 hours, each having a value of €2.500,-. Another example of a 'near miss' is a transaction having a value close to the predetermined rule, like €7.300,-.

Financial institutions have analysts of different levels to analyze these qualified transactions. These analysts work in a pyramid hierarchy structure. Once a transaction is qualified to be investigated in more detail, a first-level analyst has to decide whether or not the transaction has to be reported to a higher-level analyst or is qualified as false-alert. Eventually, the highest-level analyst decides if the transaction is indeed a money laundering transaction and therefore has to be reported to the FIU. The rule-based method detection system, in combination with human investigations, is not as effective as it should be. The disadvantages of the method are the high number of false-positives, ineffective thresholds, inefficient data processing, and the inability to recognize money laundering typologies automatically (Gao & Xu, 2006).

In order to improve the rule-based transaction monitoring method, specific tools have been developed. For instance, an artificial intelligence tool entitled Financial Crimes Enforcement Network AI Systems (FAIS). This tool uses 336 rules, based on historical reports, to classify transactions and determine alerts. Each rule individually contributes positively or negatively to the outcome to be categorized as an alert (Watkins, et al., 2003). Although tools and software programs improve the speed of classifying transactions, there still is a need for human resources to monitor and analyze millions of transactions. Since anti-money laundering specialists are scarce, the banking institutions are competing among themselves to acquire the right employees

(Rooijers & Leupen, 2020).

As a consequence of the thorough investigation of each alert transaction once detected, the process is resource-intensive and inefficient. Additionally, rule-based monitoring only matches money laundering typologies known by the system or typologies that are based on rules the system receives as input. With the continually evolving technology and new money laundering schemes, it looks like law enforcement is continuously trailing behind in this persistent losing game.

2.2.2. Process scheme typologies

A method proven to be very valuable to detect historical money laundering transactions and patterns is the five-step process scheme of typologies used by the money transaction offices, as presented in Figure 1 (De Nederlandsche Bank, 2017). This method analyzes historical transactions based on existing money laundering typologies to detect if other accounts perform similar illegal money laundering activities. The newly detected accounts could reveal novel money laundering typologies, that eventually, can be used to analyze new historical transactions.

> *Step 1: Based on public information or existing typologies a possible money laundering transaction is identified*
>
> *Step 2: The banking institution investigates if the possible money laundering transaction exists in their databases*
>
> *Step 3: If the possible money laundering transaction exists, analysis is done based on the specific features of the performed transactions, involved countries, and related persons or entities.*
>
> *Step 4: If established that new transaction patterns can be interpreted as money laundering transactions, these are translated to money laundering typologies.*
>
> *Step 5: Typologies are translated into new scenarios to monitor transactions based on the features of the typology to identify new possible money laundering transactions. Eventually, the process starts over again.*

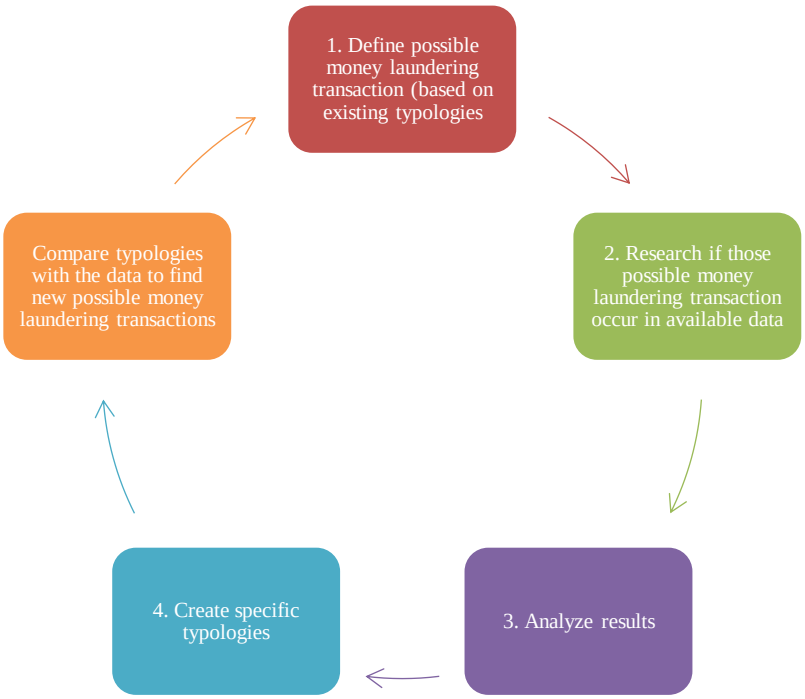


Figure 1: Visualization of the process scheme detection of typologies (De Nederlandsche Bank, 2017).

These methods used by banking institutions and law enforcement agencies are capable of monitoring millions of transactions but generate a high rate of false positives. Since machine learning techniques require high computational power, they should be thoroughly trained and tested before implementing them in their monitoring system. In the next section, previous research done on machine learning techniques to detect money laundering activities is discussed.

2.3. Previous research done on the detection of suspicious transactions

The methods and guidelines proposed by De Nederlandsche Bank (2017) generate large numbers of false-positives, high investigation costs due to the demanding human factor, and are easily avoidable by criminals. For instance, the rule-based method uses defined thresholds to detect money laundering transactions, once the criminals know these thresholds and operate below the maximum, no alert is generated. Therefore, newer and more advanced methods have to be incorporated into the monitoring system to achieve better results. Advanced research has potential to more precisely detect suspicious transactions, decreasing the false-positive rate – while maintaining or decreasing the number of false negatives - and reducing the workload on employees investigating financial transactions. Metrics used to measure the performance of methods are accuracy, precision and recall. Accuracy is defined as the correctly predicted

money laundering- and normal transactions of the total transactions of the dataset. Precision is the ratio of correctly predicted money laundering transactions of the total predicted money laundering transactions, and the recall rate is the ratio of correctly predicted money laundering transactions out of all actual money laundering transactions. In the next section, the previous research done and more advanced money laundering detection methods are discussed.

2.3.1. Rule-Based Bayesian Network

A notable method, related to the rule-based method, proposed by Khan, Larik, Rajput, & Haider (2013), is the use of a Bayesian network based on rules from the State Bank of Pakistan and regulations of Pakistan to measure the suspected behavior of customers by assigning them a score. Then, the deviation is computed between the assigned score and the historical behavior of the customer. Once the deviation is significant from defined rules and regulations of normal behavior, the transaction is marked as suspicious and requires further investigation to explain the difference in behavior. The Bayesian network does not allow us to model complicated structures, such as money laundering typologies, but it provides an easy to understand visualization of money laundering occurrences.

Further research on rule-based methods is provided by Rajput, Sadaf Khan, & Larik (2014). Their research aims to detect suspicious transactions by monitoring each transaction independently using more sophisticated rules. Khanuja & Adane (2014) also provide a method similar to rule-based monitoring, using preset instructions to monitor the transfers of financial transactions continually.

2.3.2. Clustering

A machine learning technique with historical research in the field of money laundering is clustering. Since clustering is an unsupervised machine learning technique, it does not need labeled data, and therefore is favored for anomaly- and money laundering detection. Considering the unlabeled datasets of financial institutions, this is a significant advantage compared to supervised techniques.

Clustering methods classify data to different groups based on similarities of members within one group. Research from Le Khac, Markos, & Kechadi (2010) proposes a combination of clustering and classification techniques to detect investment transactions. The paper researches customer behavior in investment activities, seemingly to be fairly complicated to the many factors involved as fund prices, market climate, currency exchange rates, and the political environment. The paper examines two investment values: subscription value, being

the value wherefore existing shareholders can participate in company rights offerings. The second value, the redemption value, is the price wherefore an issuing company could buyback securities before the maturity date - the end date of a security. In order to reflect the relationship, the paper uses the following parameters: the proportion of redemption value of the subscription value and the proportion of redemption value of the total value of investors' shares. Using these parameters, the paper tries to cluster groups to detect suspicious transactions. The paper also emphasizes the importance of automation in detection monitoring, since the volume of financial transactions that have to be investigated by financial institutions is increasing. Le Khac, Markos, & Kechadi (2010) provide promising results of the detection of suspicious transactions using clustering but on the other hand the model requires validation on larger datasets to draw better conclusions.

Zhu (2006) aims to detect money laundering transactions by comparing individual customers with their past transaction behavior. This behavior is clustered to create a customer profile and eventually detect alerts from these profiles. In addition, Zengan (2009) applies multiple clustering techniques to detect outliers and classify those as alerts, comparing different cluster techniques. Another clustering method is the method proposed by Wang & Guang (2009). Based on the criteria of dissimilarity of transactions, a minimum spanning tree is built. This tree algorithm assigns weights of similarity to connections within a graph. Subsequently, it connects all points – each point represents an account - from the graph using the lowest weights assigned without the use of circle connections. Eventually, this tree is clustered into k clusters, detecting outliers as suspicious transactions.

Moreover, Cao & Do (2012) experiment with clustering using the CLOPE algorithm, an algorithm used for categorical data based on histograms. Clustering research has been practical and functional in the real world on unlabeled data. However, cluster techniques are based on similarities between transactions, implying that outliers are alert transactions. Hence, generating a high rate of false-positives.

2.3.3. Social Network Analysis

A method focused on the relationship between financial accounts is the social network analysis performed in the paper of Colladon & Remondi (2017). The paper builds on the assumption that for illegal activities to happen, the interaction between multiple social actors is required. Therefore, the relationships between actors are analyzed by assigning individual weights to actors based on geographical, transactional, and economic factors. Since the study is based on

transactions from an Italian factoring business, geographical weights are assigned a risk score depending on the region in Italy. The transactional weight is based on the importance, frequency, and amount of transactions between two actors. The economic factor is referring to historical activities. Similar weight scores, as the scores of each actor, are assigned to the transactions, based on the in-, out-, and all degree of transactions. It concludes that social network analysis is indeed valuable when defining risk profiles and when detection suspicious patterns and transactions. The research also mentioned the increase in informative power when analyzing multiple networks.

2.3.4. Decision Tree

An interpretable method of machine learning is the Decision Tree. The Decision Tree machine learning technique creates a tree-like structure having one root node and multiple leaf nodes representing categories, including distribution of those categories. It is a prediction and classification method dealing with production rules decided by the Decision Tree itself. Wang & Yang (2007) use Decision Trees to identify rules of money laundering based on customer profiles of a commercial bank in China. It concludes that Decision Trees are useful to generate anti-money laundering rules from customer profiles. A predictive method of Decision Trees is provided by Liu, Qian, Mao, & Zhu (2011) to discover money laundering patterns and rules. It states to identify suspicious transactions more effectively than rule-based methods.

2.3.5. Random Forests

An extension of the Decision Tree method is the Random Forests method, developed by Breiman (2001). A Random Forests builds multiple Decision Trees, each created with a unique bootstrap sample – random sampling with replacement - from the training dataset. Alvarez-Jareno, Badal-Valero, & Pavia (2017) reveal that Random Forests obtain accurate results of correctly classifying money laundering transactions. The study is based on a real-money laundering case in Spain. The study uses the SMOTE methodology to deal with the imbalanced data, Random Forests results are improved. However, with the SMOTE algorithm the true positives increase but the accuracy is similar, implying an increase in false-positives when using the SMOTE algorithm.

2.3.6. Support Vector Machines

An alternative supervised classification method is the Support Vector Machines (SVM). SVM can process complex and nonlinear relations, and therefore is useful for the detection of money

laundering transactions. Tang & Yin (2005) propose a SVM method to detect suspicious transactions and recognize money laundering patterns. It is argued that the proposed SVM model could replace the rule-based method of transaction monitoring, surmounting problems such as processing large amounts of data. The paper Keyan & Tingting (2011) states that the selection of parameters for the SVM model affects the detection of suspicious financial transactions. To find the optimal parameters, a cross-validation method is performed based on the highest accuracy rate. This avoids the problem of overfitting and underfitting, improving the classifier and its performance.

2.3.7. Neural Network

Another classification method is the Neural Network, similar to Support Vector Machines, Decision Trees and Random Forests, Neural Networks can process complex and nonlinear relations. A Neural Network uses a set of connected nodes and tries to represent the functions of a human brain. The Neural Network has three layers with nodes: input nodes, hidden nodes, and output nodes. The Neural Network has single input- and output layers, being the input variables and the predicted result, but can have multiple hidden layers. Lv, Ji, & Zhang (2008) Provide a Neural Network to detect money laundering composed of different layers. It concludes that Neural Networks could improve the reduction of false-positive rate, enhances the recall score, and adapts to changing risks and means of money laundering. However, being a method, where its outcome is hard to interpret, it is complicated to explain why transactions are detected. This complication is a disadvantage for real-world practices, according to ING, since investigators also value the interpretability of money laundering patterns (ING, 2020). A consideration has to be made between the accuracy of correctly predicting money laundering transactions compared to the interpretability of these detected transactions.

2.3.8. Summary of methods

Automation of the detection process is essential since financial institutions have to monitor high volumes of transactions on daily basis. Machine learning techniques are very advantageous, detecting money laundering transactions and patterns. Due to the constant invention of new methods to launder money, supervised and unsupervised methods are powerful techniques to detect new money laundering typology. New development and testing of machine learning models are useful to improve the accuracy, recall and precision of detection methods (Salehi, Ghazanfari, & Fathian, 2017). The research mentioned in the previous sections indicate that the detection of money laundering transactions can be improved

because of machine learning methods but difficult to implement due to the high volumes of transactions. As mentioned at ING, interpretability is important to reveal and understand the typology of money laundering (ING, 2020). Liu, Qian, Mao, & Zhu (2011) implement an interpretable core Decision Tree method to identify and discover money laundering typologies and strategies. The paper argues abnormal transactions are more effectively identified using Decision Tree algorithms. The paper from Wang & Yang (2007) also states that Decision Trees are an effective method used for the creation of anti-money laundering rules, based on the company's customer profiles. Sahin & Duman (2011) confirm these statements, and even conclude that Decision Tree approaches outperform SVM methods by the number of frauds caught and by the investigation problem of interpretability.

To compare the performance of machine learning models, the metric accuracy is often used in the literature discussed. Also Keyan & Tingting (2011) select their optimal parameters based on accuracy, however, dealing with highly skewed data, it is preferable to analyze the precision and recall metrics (Jeni, Cohn, & De La Torre, 2013). The paper Jeni, Cohn, & De La Torre (2013) also concluded that skewed data in either direction affects the Kappa measure, and therefore, primarily focusing on precision and recall metrics is preferred.

Methods that proved to increase the performance of the detection of money laundering transactions are the Random Forests and Neural Network. The paper from Alvarez-Jareno, Badal-Valero, & Pavia (2017) states that the Random Forests method in combination with the SMOTE algorithm increases the correct detected money laundering transactions. Similar results are revealed by Lv, Ji, & Zhang (2008), validating a Neural Network model in comparison with SVM and outlier detection methods, arguing the Neural Network achieves the lowest false-positives rate and the highest recall rate.

However, machine learning techniques require high computational power, dealing with large amounts of data, monitoring millions of transactions every day can be a time-consuming process. Nonetheless, based on previous research, machine learning techniques provide noteworthy results when detecting money laundering transactions.

2.4. Synthetic data

Some problems arise in order to succeed to improve the detection of money laundering transactions. The main and foremost problem found in the literature is the difficulty of obtaining real-world transaction data to validate and test machine learning models on to improve the detection of money laundering transactions. Therefore, for research purposes, synthetic data is an alternative to real-world data, which has its advantages and disadvantages.

The main disadvantage of simulated synthetic data is the incapability to incorporate all different transaction patterns occurring in real-world data into one simulation model (Alonso Lopez-Rojas, 2016). Besides the limited money laundering typologies, simulated data can be restricted due to the limited features created by the simulation to more accurately predict a money laundering transaction (Alonso Lopez-Rojas, 2016). On the other hand, synthetic data has its advantages. Sophisticated transaction monitoring systems require thorough testing, tuning, and a large amount of data to improve and automate the transaction monitoring to outweigh the increase in human workload and costs associated with the investigations (Barse, Kvarnstrom, & Jonsson, 2003). Synthetic data is suitable for the fine-tuning of models because of the endless simulation possibilities of various simulation models (Alonso Lopez-Rojas & Axelsson, 2012). An additional advantage is the possibility to experiment using different properties and conditions, changing specific environments of testing to improve results, test multiple money laundering patterns to evaluate performance parameters, and use it as a benchmark to real data (Barse, Kvarnstrom, & Jonsson, 2003). Furthermore, labeled synthetic datasets with flagged money laundering transactions are suitable to test supervised learning methods. The knowledge of knowing all transactions that are used to launder money is something that seems impossible to achieve in a real-world dataset. However, according to Lopez-Rojas & Axelsson (2012) it can be challenging to build a valid model on the synthetic data available because of the complexity of variables and parameters occurring in the real world.

3. Data

In order to achieve a proper performance with machine learning models, profound training and testing on suitable data are needed to obtain quality results. This is a problem since it is difficult to obtain applicable data for research purposes because of the legal complications, competitive reasons, and the inaccessibility of this data in general (Watkins, et al., 2003). Some organizations have such data, for instance financial institutions that gather their data internally. An alternative to data containing real financial transactions is the use of a synthetic dataset created by a simulator. The simulation of synthetic data is a sophisticated alternative when using representative real-world behavior data to examine its output (Alonso Lopez-Rojas, 2016). Also, simulating data brings advantages such as creating different scenarios of money laundering under different conditions, changing the specific environment of testing to improve results, and having the option to validate models on labeled data. In order to do significant research on the detection of money laundering transactions, a representative dataset should be used (Barse, Kvarnstrom, & Jonsson, 2003). In order to achieve similar real-world behavior of normal and money laundering transaction typologies, this research uses the simulation created by IBM & MIT, called the AMLSim. The simulation uses the python language software package NetworkX and the Paysim, developed by Alonso Lopez-Rojas & Axelsson, 2016. Paysim is a simulator that represents financial transaction behavior between financial accounts during a certain time period. The simulator is created using the program languages java and python.

3.1. AMLSim

AMLSim is specially built for the simulation of financial transactions, money laundering typologies, and the behavior of retail bank accounts. The simulation is based on the multi-agent-based simulation principle, which pursues to replicate real-world behavior of agents – each agent represents a financial bank account - wherein a couple of agents perform illegal money laundering activities (Weber, et al., 2018). The simulation consists of two steps. During the first step, it creates a graph network of accounts – using NetworkX - that are connected through transactions. During the second step, the simulation creates a time series of transactions using PaySim (Alonso Lopez-Rojas & Axelsson, 2016). The time series consists of a number of timestamps, each timestamp new accounts and transactions are generated. First, these two steps are explained in further detail. Next the input files required for the simulation are clarified. At last, the normal and money laundering typologies occurring in the simulation are described.

3.1.1. AMLSim phases

In order to create a network graph of accounts and transactions, the simulation requires four input files. The following four excel input files are required: account information, transaction information, degree distribution, and money laundering patterns. Figure 2, provides a visualization of the two main steps - the transaction graph generator and the transaction simulator - executed by the simulator. To create a network of accounts and transactions the transaction graph generator first creates all the different accounts, followed by the creation of transactions between these accounts. The last step of the transaction graph generator, incorporates the suspicious transactions in the network. During the construction process of the network, accounts and transactions are assigned specific properties such as amount, timestamp, account to, and bank id. These properties are explained in more detail in section 3.2.

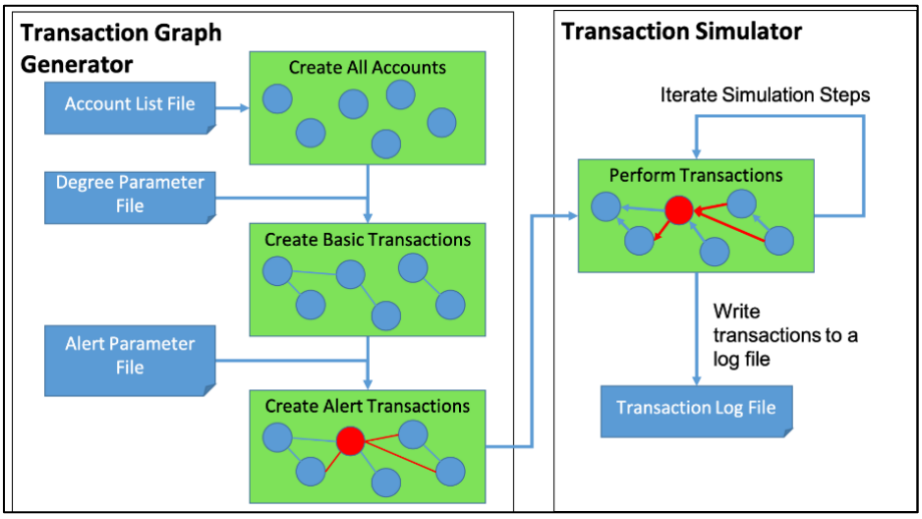


Figure 2: The AMLSim transaction graph generation and transaction simulator by IBM & MIT.

Secondly, a time series of transactions is created using the PaySim from Lopez (Alonso Lopez-Rojas & Axelsson, 2016). The simulation performs transactions based on the multi-agent-based simulation method. This method incorporates behavior similar to the real-world behavior of financial accounts, carrying out transactions. It accomplishes this by placing the agent – a financial account - in an environment wherein, based on the information received it has to make certain decisions. These decisions are based on statistical distributions of possibilities to execute a transaction. In addition, each agent has a probability factor to perform transactions in future steps. This information, distributions, and probability factors are based on a real original dataset of mobile transactions from Ericsson (Alonso Lopez-Rojas, 2016).

3.1.2. AMLSim input files

As mentioned in section 3.1.1, the simulation requires four input files. The transaction file input specifies the type of transactions occurring in the simulation. These types can vary between wire transfers, deposits, and credit. In addition, the account information file contains data about the financial account. This file can be specified by the initial balance, country, which bank it belongs to, and what behavior the account will perform. The behavior of the account can vary between multiple money laundering transaction typologies, explained in more detail in section 3.1.3. The third input file contains the degree distribution. This file assigns the in- and out degree of each account, being transactions incoming and outgoing from accounts. These degrees create connections – transactions - between accounts within the graph network of accounts.

The last file includes the money laundering patterns. These input parameters generate the kind of money laundering typologies and patterns that occur in the network. The patterns can be specified by amount, the period in which the transaction is transferred, and the number of accounts interacted with. The table below is an example of the money laundering pattern excel file.

Table 1: The money laundering pattern input file for the AMLSim.

C.	**Trans_ typology**	**Sched _id**	**Min_ acc**	**Max_ acc**	**Min_ amount**	**Max_ amount**	**Min_ period**	**Max_ period**	**Bank_ id**	**Is_sar**
15	Fan_ in	2	2	5	100	200	5	10	Bank a	True
15	Cycle	2	5	7	100	200	5	10	Bank c	False

In order to clarify the money laundering pattern file in more detail, both rows presented in Table 1 are described in more detail. The first row of the money laundering pattern input file provides the following information to the simulator. It describes 15 *fan in* - multiple accounts transfer money to one single account - transactions using *Schedule id* 2, this denotes that accounts send their money randomly within the margins given in the other columns of this input file. The minimum and maximum accounts to interact with are set to 2 and 5 respectively. Since it is a *fan in* transaction typology, this indicates there is one beneficiary account receiving the money and two to four accounts sending the money. The transaction value is set to *€100,-* and *€200,-* respectively, within a period of 5 to 10 days in which the typology *fan in* is transferred. The accounts performing the money laundering transactions all belong to bank a.

At last, these transactions are flagged as SAR, this indicates whether these simulated transactions are indeed money laundering transactions.

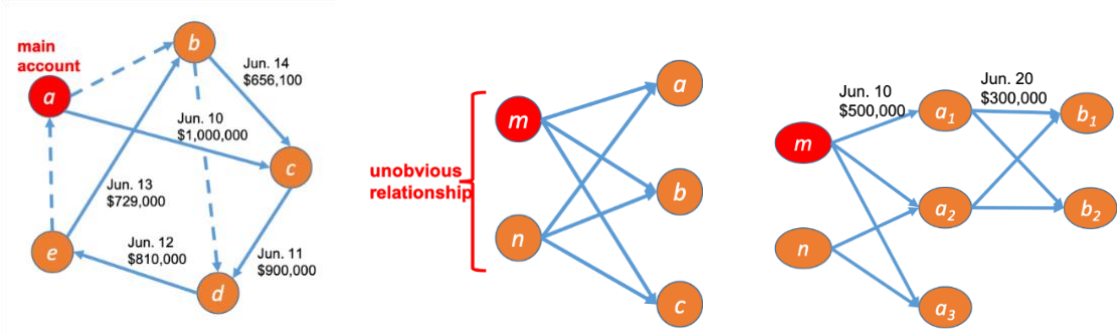
The second row of input parameters is slightly different from the first row. It describes 15 *cycle* - a transaction transferred to multiple accounts and eventually coming back to the first initiator account - transactions with schedule id 2. The alteration from the first row is the number of accounts interacted with each cycle. This number is set to 5 to 7 accounts that are involved in a *cycle* transaction. Another modification is the *is sar* column. This column is set to false, indicating the transactions are detected as alerts, but they are in fact normal transactions and therefore not flagged as SAR but false alerts. The simulator provides this option for researchers to create false alerts on purpose, this option is neglected in this research.

3.1.3. AMLSim normal and money laundering typologies

The AMLSim incorporates a wide variety of normal and money laundering typologies. The normal typologies consist of *'single', 'fan out', 'fan in', 'mutual', 'forward',* and *'periodical'*. A *single* transaction implies a transaction from account A to account B. The *fan out* and *fan in* typologies are situations when one account receives money from or sends money to multiple other accounts. The typology *mutual* indicates transactions from account A transferred to account B, within a short period account A receives the money-back again from account B. *Forward* typology denotes money transferred from account A to account B, and account B forwards the money to account C. The last typology *periodical* occurs when account A transfers money to account B on a monthly basis for example.

The money laundering typologies used are more complex compared to normal typologies. Money laundering includes *'fan-out', 'fan-in', 'cycle', 'bipartite', 'stacked bipartite', 'random', 'scatter gather'*, and *'gather scatter'*. *Fan in* and *fan out* are similar to the normal equivalent typologies. The *scatter gather* and *gather scatter* typologies combine *fan in* and *fan out* typology after each other. For example, using *fan out* and *fan in* after each other such that an account transfers money to multiple accounts, these multiple accounts then again send the money to one single account. This can be executed the other way around as well. The *cycle* typology indicates account A sends money to account B, and account B then forwards the money to account C, whereas account C transfers the money back again to account A. This is an example of the shortest cycle possible, with three account interactions. The typologies *random*, *bipartite,* and *stacked bipartite* are visualized in Figures 3, 4 and 5. The *random* typology, similar to the *cycle* typology, but the transactions are performed randomly to other

accounts instead of in a cycle pattern. The *bipartite* typology indicates two accounts having an unobvious relationship with each other, both of these accounts send money to the same three other accounts. The *stacked bipartite* is an extension of the *bipartite*, as shown in Figures 3, 4 and 5.



Figures 3, 4 and 5: A visualization of the random (left), bipartite (middle) and stacked bipartite (right) typologies.

Normal transaction typologies appear more simplistic compared to money laundering typologies. However, complications arise when the money laundering typologies resemble similar transaction patterns to normal transaction typologies. Because when the amount, period, and the number of accounts interacted are close to normal transaction patterns, money laundering transactions can be hard to detect.

3.2. Data preparations

The simulation generates two output files: the first containing information of the accounts, the second contains information of the financial transactions. Table 2 and 3 in the appendix provide an overview of each dataset with descriptions of each unique variable. Table 2 represents the financial transaction data and Table 3 the account data.

3.2.1 Fixed parameters

In order to answer the research questions, the first and foremost challenge is to decide which parameters to use as input for the simulation. There is no clear consensus found in the literature on what parameters are best or most representative to use. Therefore, the default settings proposed by (Weber, et al., 2018) in their latest simulated dataset *'bank-mixed'* are used concerning the degree-, accounts-, and transaction typology input files. Throughout the research, these parameters stay fixed for every dataset generated. Using these default parameters, the simulation generates a network of twenty thousand accounts, performing all

variations of normal transaction typologies distributed proportionately over all accounts. Each individual account of these 20 thousand accounts belongs to one of the five simulated banking institutions (A untill E). Banks A and D both have 2000 accounts registered, 4000 accounts belong to banks B and E, and Bank C has 8000 registered accounts. Furthermore, each account is generated by having an initial balance between 25- and 100 thousand. At last, all transactions performed are transactions of the type transfer, implying money transfers between two retail bank accounts.

3.2.2. Added variables

It is also important to extract valuable information from the network to the dataset used as input for the machine learning models. This information is not directly converted to the dataset but is still hidden in the network generated by the AMLSim. In order to translate the information from the network to the generated data, and thereby provide a better, more valuable input for the machine learning models, new created variables are added to the transaction dataset (Xingrong, 2014). These newly created variables are based on value, account balance, number of transactions transferred, number of interactions with other accounts, and the interval period between two transactions.

First, the *value* variable is converted to a representative distribution to real-world transactions. Since the simulator generates normal transactions with a value in between 90,- and 110,-, and of these transactions 99 percent has the exact value of 100,-, the value of transactions should be replaced with a representative distributed variable. The new *value* distribution is based on transaction values of real-world bank accounts with an age of in between 18 and 30 years old. The transactions executed have a value in between 0.01 and 5000,-. However, since transactions above 1100,- and below 3,- are rarely occurring in the data, and mostly are wage payments and small interest transactions, these are not included in the dataset. To perform significant research on the structures and patterns of money laundering typologies, money laundering transaction simulated have similar values.

In extend of the *value* variable, individual columns are created for the average, minimum, maximum, total, and sum of all transactions incoming and outgoing for the origin and destination account of each transaction. A similar process is done for the creation of the interval variables, creating a separate column for the minimum- and maximum interval between two transactions incoming and outgoing for the origin- and destination account. Additionally, variables for the number of interactions incoming and outgoing with unique accounts for the

origin- and destination account are created. The same is done with regard to the number of transactions outgoing, incoming, and total for the origin- and destination account. Also, since the simulator did not incorporate the initial balance of the account file, the initial balance is added from the account dataset to create additional columns containing information of the old and new balance, of the origin- and destination accounts, prior and after each transaction. Lastly, the column including the *bank_id* is converted to dummy variables for origin- and destination account. A more comprehensive explanation of the generated variables is given in Table 4 of the appendix.

3.2.3. Removed variables

The third important part is to omit variables from the dataset that do not contribute to the prediction of the machine learning models and therefore are considered not valuable. The variables *start date, end date, country, account type,* and *bank ID* are not valuable due to the fact that they all contain singular values. Furthermore, the *customer ID* is similar to the *account ID* variable and therefore provides no added value. At last, *tx behavior ID* - ranging from 1 to 5 - representing one of the normal transaction typologies, implying it is not a money laundering transaction. Since this variable indicates a distinct split to decide if a transaction is a money laundering transaction, it is removed from the dataset.

The financial transaction dataset contains information from the transactions executed between accounts over a time period. However, some variables are not of added value to this research. The variables *old balance orig, new balance orig, old balance dest,* and *new balance dest* are removed because the simulation did not incorporate the initial balance from the account dataset into the transaction dataset, resulting in the miscalculation of balances of accounts. Furthermore, the *transaction type* variable is removed due to the singular value of this variable, since all transaction types are transfer transactions. Also, the *origin* and *destination accounts* are removed from the dataset because they have no predicted value. Furthermore, the *bank_id_org* and *bank_id_dest* are omitted after the dummy variables are created for *bank_id*. Lastly, the *alert ID* variable is excluded from the dataset due to the high correlation with the *SAR* variable. The final dataset is provided in Table 4 of the appendix.

3.3. Limitations of the data

Data of financial transactions either synthetic or real-world data have their limitations. Real-world data mainly incorporates unsupervised learning methods since the data consist primarily of unlabeled transactions. Also, if labeled data is obtainable, there is a chance of undetected

money laundering transactions being mistaken for normal transactions. However, since this can happen for any given dataset, it depends on the purpose of the model. If the goal is to replace the human work tasks to flag money laundering transactions, the model can be of additional value to increase the speed of the process.

Furthermore, the data used to train and validate the machine learning models has some limitations. First, since the data is simulated, it is specified on a particular part - the transaction network - of money laundering and therefore does not include every scenario possible. Therefore, the main disadvantage of simulated synthetic data is the incapability to incorporate all different transaction patterns occurring in real-world data (ING, 2020). The data from the simulator, is simulated with a limited number of transaction patterns and input parameters to modify. Second, apart from the limited typologies, the simulated data is also restricted because the simulator simulates only one transaction type and does not incorporate low and high-risk countries. Since money laundering activities also involve cash deposits and withdrawals, it is valuable to include these transactions. Unfortunately, the simulator requires further development to incorporate cash transactions. Also, according to De Nederlandsche Bank (2017), an effective detection system should incorporate risk weights to differentiate between low and high-risk countries. Since the simulator does not integrate different 'risk' countries, this research focuses on the network of transactions within one country. At last, the simulator cannot simulate a balanced dataset with equal number of money laundering and normal transactions. Therefore, techniques to cope with imbalanced data are still needed. Due to the limitations of the data available, only structures within one country and with one transaction typology are researched. However, these structures are possible between multiple banks and additional variations.

4. Methodology

4.1. Creation of datasets

In order to answer the research- and sub questions, alterations are made in the money laundering pattern file before generating each dataset. The money laundering pattern file of the first dataset used for the first sub-question is presented in Table 5.

Table 5: The money laundering pattern input file for the first dataset created by the AMLSim.

C.	Trans _typology	Sched_ id	Min_ acc	Max_ acc	Min_ amount	Max_ amount	Min_ period	Max_ period	Bank _id	Is_sar
400	Fan_in	2	4	10	30	1000	10	30	“”	True
400	Fan_out	2	4	10	30	1000	10	30	“”	True
400	Cycle	2	4	10	30	1000	10	30	“”	True

To answer the first sub-question, ‘*How do machine learning techniques perform on various typologies of money laundering?’,* all input parameters maintain unchanged except for the transaction typologies. The number of account interactions is set to 4 and 10, and the period in which the money laundering transactions are done is kept at 10 to 30, these are the default settings of the simulator. As Drezewski, Filipkowski, & Sepielak (2012) states that a common characteristic of money laundering patterns is the execution of numerous transactions in a relatively short period, the default parameters are reliable since the normal transaction interval varies around 30 days. Furthermore, the parameter of column *schedule id* is kept at 2, indicating the transactions are performed randomly within the margins given of the parameters: accounts involved and the period in which the typologies are executed. The “” sign in the *bank id* column indicates that each account is randomly assigned to one of the five - A till E - banking institutions. Also, all transactions are set to True with regard to the *is sar* column. This specifies that all transactions generated with these parameters are indeed money laundering transactions. Throughout this research no false positives are created using the money laundering input file. This because in a real-world situation, law enforcement agencies would want to detect a false alert - normal transaction - that has the exact same typology, pattern, and structure as a money laundering transaction. At last, for the first dataset 1200 unique money laundering typologies are created, shown in the first column of Table 5. Each exact money laundering pattern input file for each dataset can be found at section 8.2. in the appendix.

In total, 14 unique datasets are generated to examine various money laundering typologies and analyze the performance of machine learning techniques on those typologies. In Table 6, the typologies and unique alterations of each dataset are presented. Table 5 presents the default input parameters used for each simulation, solely the parameters mentioned in Table 6 are changed, other parameters not mentioned in Table 6 remain unchanged.

The specific datasets created to answer each sub-question are explained in more detail below Table 6. Moreover, datasets from different questions will be compared and analyzed with each other in order to strengthen and broaden the reasoning behind the answers.

Table 6: Overview of different datasets generated.

	Typologies	**Other alterations**
Research sub question 1 (Clusters of money laundering typologies)		
Dataset 1	Fan in, Fan out, Cycle	-
Dataset 2	Scatter Gather, Gather Scatter	-
Dataset 3	Bipartite, Stacked Bipartite, Random	-
Research sub question 2 (One typology versus multiple typologies)		
Dataset 4	Cycle	-
Dataset 5	Scatter Gather	-
Dataset 6	Random	-
Dataset 7	Fan in, Fan out, Cycle, Scatter Gather, Gather Scatter, Bipartite, Stacked Bipartite and Random	-
Research sub question 3 (Change in account interactions and periods)		
Dataset 8	Cycle	Min- & Max account interactions is set to 2 and 5. Min- & Max period is set to 20 and 40.
Dataset 9	Random	Min- & Max account interactions is set to 2 and 5. Min- & Max period is set to 20 and 40.
Dataset 10	Cycle	Min- & Max account interactions is set to 10 and 15. Min- & Max period is set to 5 and 20.
Dataset 11	Random	Min- & Max account interactions is set to 10 and 15. Min- & Max period is set to 5 and 20.
Research sub question 4 (One bank versus multiple banks)		
Dataset 12	Fan in, Fan out, Cycle	Filtered for Bank C accounts
Dataset 13	Scatter Gather, Gather Scatter	Filtered for Bank C accounts
Dataset 14	Bipartite, Stacked Bipartite, Random	Filtered for Bank C accounts

As shown in Table 6, three datasets are generated to answer the first sub-question: '*How do machine learning techniques perform on various typologies of money laundering?'*. For this research question, only the typologies differ between each dataset. The typologies are divided into three clusters based on similarity with normal transaction typologies, complex structures, and unobvious – more complex - account relationships. The first cluster contains the *fan in, fan out* and *cycle* typologies. The second cluster includes the *scatter gather* and *gathers scatter* typologies and the third cluster consists of the *bipartite, stacked bipartite* and *random* typologies.

The second sub-question: *'How do machine learning techniques perform on data where only one typology of money laundering exists compared to data with multiple money laundering typologies?'* is evaluated by generating four additional datasets. The first three new simulated datasets, contain one typology of each cluster created in sub-question one. The typologies analyzed in more detail are: *cycle, scatter gather* and *random*. These three typologies possess very different characteristics and therefore are interesting to conduct further research on. These three datasets are compared to the fourth created dataset, which includes all eight money laundering typologies possible to simulate using the AMLSim.

In order to answer the third sub-question: *'How do machine learning techniques perform on differences in the number of account interactions and periods wherein the typology is performed?'* four new datasets are generated. The datasets contain the typologies *cycle* and *random*. These are similar to the second sub-question, but for each dataset, changes are made to the parameters: number of account interactions and period within the typologies are performed. Two datasets are generated with the same typology, but one with a short period and many account interactions and the second dataset with extended periods and a few account interactions.

The last sub-question '*What is the difference in the performance of machine learning models when detecting suspicious transactions within one bank compared to transaction monitoring between multiple banks?'* is answered using three datasets having the same cluster typologies as the datasets used for sub-question one. However, each dataset is filtered to include only transactions from and to accounts belonging to bank C.

4.2. Modelling

Before modelling and validating the machine learning techniques on each dataset, the data is split into a training and test dataset, of which 80 percent is training data and 20 percent is test data. All sub-questions are answered using the following three machine learning techniques explained in the next sections.

4.2.1. Decision Tree

The first machine learning technique investigated is the classification Decision Tree. This tree uses a classification algorithm to understand, interpret, and partition the data but it is also applied for prediction purposes. The main idea is to classify each observation into a class using variables to split the data. This is accomplished by growing a tree upside down to partition the data at every node. At the beginning of the model's construction, all observations belong to one single group. The observations are divided into two groups at each split, based on the best possible split. This is done for each group until every observation is contained in its own group or a stopping condition is met. The method to define the best split is the Classification And Regression Trees (CART). The CART algorithm partitions the data based on a cost function, aiming to minimize these costs. The cost is minimized for a classification problem by using the Gini impurity function. This impurity indicates the probability of incorrectly classifying a randomly chosen element in the data. The Gini impurity function is shown in the following formula:

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i)) \tag{1}$$

Where C is the total classes and $p(i)$ is the probability of selecting a data point with class i, preferably wanting the lowest possible Gini impurity index. This is done by maximizing the Gini Gain, being the value of impurity that is removed at a particular binary split. This process is repeated at each node to decide on the best possible split. The variable most valuable to split the data is presented at the root of the tree. Each next split the variable to divide the data becomes less valuable.

Decision Trees can suffer from over- and underfitting. The tree suffers from underfitting when the underlying process is not represented and only a couple leaves are grown. Overfitting occurs when the tree is overly focused on the data, this can happen when no stopping criteria are used. Stopping criteria to prevent the tree from overfitting are: providing a maximum depth

for the tree, provide a cost complexity parameter, and define a minimum number of observations at a node in order to make the split.

The Decision Tree model is trained and validated using two different packages. The first package is *rpart*, using the CART algorithm. This tree is trained with a complexity parameter (cp) – a value to control the size of the Decision Tree - of 0.01, the default. Furthermore, the tree is trained with a maximum depth of 8 in order to prevent the tree from overfitting. The second three package used is *ctree*, this package constructs a Conditional Inference Tree, trained with a max depth of 8, similar to the *rpart* tree. This package is used as a comparable method, because the *rpart* package tends to select variables that have many different values or possible splits. The *ctree* package deals with this problem by using a significance test procedure to select the best variables to split instead of, for example the Gini impurity function. After the model is trained and tested, a confusion matrix is used to analyze the performance metrics, explained in section 4.3. Classification trees are known for their interpretability, however, small modifications in the data can cause unreliable and different results. A method to improve this significantly is the machine learning technique Random Forests.

4.2.2. Random Forests

Random Forests, being an ensemble method, combines multiple algorithms to improve prediction accuracy. Constantly combining multiple Decision Trees together, aims to improve the robustness and generalizability of a single Decision Tree. Each tree is different because each individual tree within a Random Forests is a subset of variables of the initial data. Using only a couple of the available variables at each split, it tries to decorrelate the trees. When predicting a class, it uses the outcome of all individual Decision Trees to predict accordingly, reducing the variance when compared to a single Decision Tree. In order to train and validate the Random Forests, the package *ranger* is used. The package builds 100 trees with a maximum depth of 8 and uses 8 variables to base its best split on. The package *ranger* is chosen because the dataset contains many observations and requires high computational power. The *ranger* package is designed to cope with datasets that have many observations and performs the analysis in a reasonable time. Therefore, this research requires the use of additional packages to be able to run Random Forests.

In conclusion, the Random Forests technique provides an intriguing model to improve the prediction of suspicious money laundering transactions compared to the classification Decision Trees. However, a problem with Random Forests is the algorithm becomes slow quickly once

the number of trees is increased. Especially concerning the predictions of Random Forests after they are trained since this research aims to detect suspicious financial transactions, the model should be suitable for larger datasets. A Random Forests package known to improve the speed of Random Forests is used to solve this problem. However, to give another insight into the machine learning possibilities to detect suspicious transactions, this research considers the Neural Network.

4.2.3. Neural Network

Due to the complexity of relationships between accounts and their financial transaction behavior, more sophisticated methods can be used to predict suspicious financial transactions. Neural Networks are applied to solve large-scale problems, and improve prediction accuracy when having more observations to train on. The Neural Network uses three different node types. Having three layers with nodes: input nodes, hidden nodes, and output nodes. The Neural Network has single input- and output layers, being the input variables and the predicted result, but can have multiple hidden layers. Between the layers, a signal is sent to a node from a connected node in the previous layer. This signal, being the output value of the previous node multiplied by an assigned weight. Combining all the signals with the constant of each node value is presented and put through an activation function onto the next layer.

This is repeated for every node until it produces a result in the output layer. By running the Neural Network multiple times, called epochs, the prediction accuracy increases. At first, the prediction can be fairly random. However, after each iteration the cost function is analyzed. Based on the information of the model's performance, it improves by using the optimizing function, calculating new weight and bias values. Using the new improved values, the model runs again, this process continues until no improvement of the cost function is possible or a stopping criterion is met.

It is essential to use the proper number of nodes in hidden layers to achieve relevant results for a given issue. Using too few nodes results in the underfitting of the model, implying it is not complex enough. On the other hand, including many nodes results in overfitting, meaning the model is too specific for the data used and cannot be generalized or used on other data. In order to define the appropriate number of nodes, different methods are used such as 'trial and error', 'heuristic approach', and 'pruning' (start with many weights and removing small weights each run). Removing and adding hidden layers have similar effects on the complexity and generalization of the model. In addition, a suitable activation function has to be chosen. An

activation function transforms the input of a node before sending it to the following layer. This transformation can have a significant influence on the model's outcome. Activations functions used are Sigmoid, ReLu (rectified linear), and Softmax. Sigmoid is used for binary classification and Softmax for multi classification, however, since the results are similar for binary classifications the Softmax activation function is used for all Neural Networks performed in this research.

The Neural Network is trained and validated using the Neural Network from Keras. The Keras model is a high-level Neural Network API written in Python and runs on top of Tensorflow. Developed to enable fast experimentation, since the complex and large dataset used, the Keras Neural Network is chosen to perform the analysis. To run the Keras Neural Network model the data is converted to numeric variables, normalized and transformed to a matrix. The Keras Neural Network is constructed with 1 hidden layer that has 27 units, the optimal size of the hidden layer according to Heaton (2015), is the mean between the input, and size of the output layers. In order to prevent the tree from overfitting a dropout layer is introduced, 1 layer with dropout rate of 0.2. Based on the documentation of the Keras model, the first three layers use the ReLu activation function and the output layer uses the Softmax activation function in order to predict a single money laundering transaction and produce an output in between 0 and 1 that can be easily converted to class values (Dunne & Campbell, 1997). The model uses the Categorical Cross-Entropy loss function since a binary classification problem is researched. This is the default and preferred loss function, it summarizes the average difference between the predicted and actual probability distributions to predict 1. This score is minimized, once the score has a value of 0 it is a perfect Cross-Entropy. Also, the combination of the Softmax activations function and the Cross-Entropy loss function leads to more accurate results (Dunne & Campbell, 1997). Furthermore, the model runs 5 epochs having a batch size of 32, a validation split of 0.2, and uses the *adam* optimizer.

4.3. Evaluation

Since the data is highly skewed with approximately 0.004 percent transactions being flagged as money laundering transactions, the vast majority of data consists of normal transactions. Therefore, it is important to choose the most suitable metrics to measure performance of the models. The performance metrics chosen to evaluate the machine learning models are the F1-score, balanced accuracy, precision and recall. The F1-score represents the mean of the metrics recall and precision. The numbers used to calculate the precision, recall, balanced accuracy and

F1-score will be retrieved from a confusion matrix. Table 7 presents an example of a confusion matrix to better understand the calculations made.

Table 7: Example of a confusion matrix to analyze the results.

Predicted / Actual	**0**	**1**
0	53 (True negative)	8 (False negative)
1	3 (False positive)	22 (True positive)

Also, precision and recall are used for interpretation of the results instead of accuracy due to the highly skewed data (Jeni, Cohn, & De La Torre, 2013). Precision and recall are metrics, not much affected by the skewness of the data. Precision, is the ratio of correctly predicted money laundering transactions divided by the total number of predicted money laundering transactions. For instance, if 22 money laundering transactions are predicted correctly from a total of 25 predicted money laundering transactions, the precision ratio is $\frac{22}{25}$. The recall ratio, is the number of correctly predicted money laundering transactions divided by the actual total number of money laundering transactions in the data. For example, if 22 money laundering transactions are correctly predicted of the total of 30 money laundering transactions in the data, the recall ratio is $\frac{22}{30}$. The formula for recall is given below:

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

Where TP stands for true positives – the correctly predicted money laundering transactions – and FN stands for false negatives, the actual money laundering transactions predicted as normal transactions. The precision formula:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Where FP stands for false positives, the normal transactions predicted as money laundering transactions. The formula for the F1-score is:

$$F1 = \frac{2 * precision * recall}{precision + recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{4}$$

Also, the balanced accuracy is analyzed, calculated as presented in formula 5. Where P stands for positives, TN for true negatives, and N stands for negatives.

$$Balanced\ accuracy\ = \frac{\left(\frac{TP}{P} + \frac{TN}{N}\right)}{2} \tag{5}$$

According to Timotius & Miaou (2010) the kappa value of balanced- and imbalanced data differ significantly and therefore not recommended to use as a metric of validation. To cope with imbalanced data one additional method, the Synthetic Minority Over-Sampling Technique (SMOTE), is introduced. All machine learning methods are applied to the original- and SMOTE data to analyze if this technique can improve the performance of the models. The SMOTE technique balances the data by creating a synthetic data point. The technique takes the vector between one of the k nearest neighbors – in feature space – and multiply this vector by a number between 0 and 1, randomly assigned. The new synthetic data point is created by adding the result of the multiplication to the current data point. Since the simulator cannot generate an evenly distributed dataset with equal number of money laundering and normal transactions, these techniques are needed for evaluation.

The results for each machine learning model and imbalance technique on each dataset are discussed in section 5.

5. Results

This section presents the results of this study. First, the F1-score and balanced accuracy results are connected to each individual sub-question and substantiated with meaningful results from other sub-questions. In this part, the results are also compared between the different machine learning models used in this thesis and a comparison is made between the performance metrics - F1-score and balanced accuracy - used to evaluate the models. Second, in order to better clarify the differences in F1-score and balanced accuracy the precision- and recall scores are analyzed in more detail. Lastly, the results are presented to evaluate the SMOTE technique to cope with imbalanced data. In this section, only the most meaningful and interesting results are presented, the full results of performance metrics from each model can be found in Tables 8, 9, 10 and 11 in the appendix.

5.1. F1-score and balanced accuracy

5.1.1 Sub-question one

The F1-score and balanced accuracy of the first three datasets belonging to the first sub-question: '*How do machine learning techniques perform on various typologies of money laundering?'* are presented in Table 12. The first dataset contains the *fan in, fan out* and *cycle* typologies. The second and third dataset contain respectively the *scatter gather, gather scatter,* and the *random, bipartite* and *stacked bipartite* money laundering typologies.

Table 12: Comparison of F1-scores and balanced accuracy between four different machine learning models on the original data of datasets 1 till 3.

Metric	Dataset	Dtree	Ctree	Rforest	Neural Net	Average
F1-score						
	1	**0.464**	0.294	**0.662**	**0.512**	0.362
	2	0.433	**0.359**	0.651	0.306	0.337
	3	0.281	0.225	0.413	0.115	0.163
	Average	0.393	0.293	0.575	0.311	
Balanced accuracy						
	1	**0.653**	0.586	**0.748**	**0.817**	0.780
	2	0.641	**0.610**	0.744	0.798	0.780
	3	0.582	0.564	0.631	0.532	0.663
	Average	0.625	0.587	0.708	0.716	

Three of the four machine learning models perform best on the first dataset containing the money laundering typologies *fan in, fan out* and *cycle*. This can be explained by the fact that

the cluster of money laundering typologies of the first dataset are not as complex as the clusters of typologies occurring in the other two datasets. This assumption is strengthened since the third dataset - containing the most complex money laundering typologies – seems to be the most difficult to predict money laundering transactions. The average F1-scores of all models on each dataset show that machine learning models in general have more difficulty predicting money laundering typologies with unobvious account relationships, explained in section 3.1.3, occurring in dataset three, compared to the typologies that are fairly similar to the normal transaction typologies.

Random Forests outperforms the other models on every dataset, implying it copes best with various clusters of money laundering typologies. The Random Forests performing better than the Decision Tree and Conditional Inference Tree, can be argued by the fact that a model's prediction, based on the prediction of multiple trees is more precise than one good tree. However, it is surprising that the Neural Network on the other hand, is highly affected by the differences in money laundering typology clusters. The F1-score is decreasing by 0.206 from dataset one to two and 0.191 from two to three. One would expect that the Neural Network can deal with complex data relationships such as the *random, bipartite* and *stacked bipartite* money laundering typologies.

Furthermore, the Decision Tree performs better than the Conditional Inference Tree, this can be clarified because the splitting criteria from the Decision Tree is biased towards certain variables with many possible splits compared to the unbiased splitting criteria of the Conditional Inference Tree. This can be a positive bias for this dataset and therefore outperforming the Conditional Inference Tree.

The balanced accuracy performance measure shows fairly similar results in comparison with the F1-scores. Therefore, substantiating the assumptions regarding the differences in datasets. However, the average score of balanced accuracy of the Neural Network is higher in comparison with the other models. This could indicate that the precision and recall scores of the Neural Network do not differ as much as the difference in scores of the other models. A more detailed analysis of the precision and recall scores is given in the section 5.2.

5.1.2 Sub-question two

The results from the second sub-question: *'How do machine learning techniques perform on data where only one typology of money laundering exists compared to data with multiple money laundering typologies?'* are given in Table 13. Dataset four, five and six contain

respectively the *cycle, scatter gather* and *random* money laundering typologies. The seventh dataset incorporates all eight possible typologies.

Table 13: Comparison of F1-scores and balanced accuracy between four different machine learning models on the original data of datasets 4 till 7.

Metric	Dataset	Dtree	Ctree	Rforest	Neural Net	Average
F1-score						
	4	0.423	0.205	0.524	0.414	0.301
	5	0.374	**0.364**	**0.693**	**0.479**	0.369
	6	0.352	0.187	0.512	0.354	0.245
	7	**0.439**	0.337	0.619	0.459	0.321
	Average	0.397	0.273	0.587	0.427	
Balanced accuracy						
	4	0.637	0.557	0.678	0.693	0.738
	5	0.616	**0.613**	**0.767**	**0.742**	0.782
	6	0.608	0.552	0.673	0.664	0.732
	7	**0.643**	0.602	0.725	0.693	0.757

It is interesting that the seventh dataset performs best or second best since all eight money laundering typologies occurs in that dataset. At first, it was expected that data that includes many variations of typologies is more difficult to predict than data containing only one typology. However, the results indicate that data containing only one complex money laundering typology – dataset six – is harder to predict than data including all typologies. Even dataset four – including the *cycle* typology – performs less on average of all models compared to dataset seven. Including the results from dataset five, it can be said that there is no large difference in performance of models on the differences of typologies in data. This is strengthened by the scores of the balanced accuracy.

However, the Random Forests – similar as the results in sub-question one – outperforms the other models. Also, the Neural Network is not as affected by the changes in money laundering typology occurring in the dataset as was seen in the results of sub-question one. This could indicate that the Neural network performs better if data contains one typology of money laundering. On the other hand, the model still performs decent on the data including all eight money laundering typologies. This assumes that the Neural Network is suitable for data containing one money laundering typology or many money laundering typologies.

When including the F1-score results from sub-question one, shown in Table 14, it is arguable that there is a slight trend visible when comparing data containing a single typology compared to data containing a cluster of typologies (datasets 1, 2, 3 compared to datasets 4, 5,

6). When money laundering typologies become more complex, machine learning models perform better on data that has one complex typology. Data that consist of relatively simple typologies are better predictable if these typologies occur in the same dataset.

Furthermore, the averages of F1-scores from each model do not differ significantly between the different datasets of sub-question one and two except for the Neural Network. The Neural Network perform better on the datasets form the second sub-question. These results are surprising since the money laundering typologies in the first three datasets are fairly similar as in the fourth, fifth and sixth dataset. This could be explained that the Neural Network has more difficulty predicting the *bipartite* and *stacked bipartite* since those money laundering typologies are included in dataset three and are excluded in dataset six.

Table 14: Comparison of F1-scores between four different machine learning models on the original data of datasets 1 till 7.

Metric	Dataset	Dtree	Ctree	Rforest	Neural Net	Average
F1-score Q1						
	1	**0.464**	**0.294**	**0.662**	**0.512**	0.362
	2	**0.433**	0.359	0.651	0.306	0.337
	3	0.281	**0.225**	0.413	0.115	0.163
	Average	0.393	0.293	0.575	0.311	
F1-score Q2						
	4	0.423	0.205	0.524	0.414	0.301
	5	0.374	**0.364**	**0.693**	**0.479**	0.369
	6	**0.352**	0.187	**0.512**	**0.354**	0.245
	7	0.439	0.337	0.619	0.459	0.321
	Average	0.397	0.273	0.587	0.427	

5.1.3. Sub-question three

The results from the third sub-question: *'How do machine learning techniques perform on differences in the number of account interactions and periods wherein the typology is performed?'* are demonstrated in Table 15. Dataset eight and ten contain the *cycle* typology. In dataset eight the typology is performed with few account interactions and within a longer period. This dataset is compared to dataset ten, where the typology is performed with many account interactions in a shorter period of time. This is similar with regards to dataset nine and eleven, however, these datasets contain the *random* typology.

Table 15: Comparison of F1-scores and balanced accuracy between four different machine learning models on the original data of datasets 8 till 11.

Metric	Dataset	Dtree	Ctree	Rforest	Neural Net	Average
F1-score						
	8	0.463	**0.288**	**0.607**	0.376	0.295
	10	**0.471**	0.223	0.534	**0.412**	**0.313**
	9	0.341	0.146	0.457	**0.253**	0.201
	11	**0.361**	**0.162**	**0.528**	0.219	**0.237**
	Average	0.409	0.205	0.532	0.315	
Balanced accuracy						
	8	0.651	**0.584**	**0.718**	0.636	0.755
	10	**0.655**	0.563	0.682	**0.699**	0.737
	9	0.603	0.539	0.648	0.629	0.716
	11	**0.613**	**0.544**	**0.681**	**0.720**	0.732

To relate the results to the sub-question, I first compare the data that contains similar typologies. Thereafter, results from other datasets and sub-questions are incorporated into the analysis. The differences in F1-scores are fairly small, this makes it harder to interpret and make certain assumptions. As for the F1-score and balanced accuracy of the data containing the *cycle* typology, the Conditional Inference Tree and Random Forests perform better on data when the typology is performed with few accounts and over a longer period, and the Decision Tree and Neural Network perform better on data when the typology is performed with many accounts and within a shorter time period.

When the typology is more complex, the prediction is more accurate when the typology is performed with many accounts and in a short period. In general, the models perform slightly better on data when the typologies are executed in line with the common characteristics of money laundering transactions. These are transactions transferred using many accounts and executed within a short period of time since this allows the criminals to launder higher amounts of money. This can be explained by the fact that these typologies are more divergent because of the larger number of account interactions and the short period of time compared to normal transaction typologies. However, since the differences are small, no strong assumptions can be made. Also, when including the F1-scores from the *cycle* and *random* typologies performed with the default configuration – datasets four and six - with medium number of account interactions and medium length of time shown in Table 16, the results deviate too much per model to draw certain assumptions. Although assumptions related to sub-question three are hard to draw, the results enhance the assumption of sub-question one that more complex money

laundering typologies are more difficult to predict compared to the simpler typologies since dataset six, nine and eleven score lower compared to the other three datasets.

Table 16: Comparison of F1-scores between four different machine learning models on the original data of datasets 4, 6, and 8 till 11.

Metric	Dataset	Dtree	Ctree	Rforest	Neural Net	Average
F1-score						
	4	0.423	0.205	0.524	**0.414**	0.301
	8	0.463	**0.288**	**0.607**	0.376	0.295
	10	**0.471**	0.223	0.534	0.412	**0.313**
	6	0.352	**0.187**	0.512	**0.354**	**0.245**
	9	0.341	0.146	0.457	0.253	0.201
	11	**0.361**	0.162	**0.528**	0.219	0.237

5.1.4. Sub-question four

The results of the fourth sub-question: '*What is the difference in the performance of machine learning models when detecting suspicious transactions within one bank compared to transaction monitoring between multiple banks?'* are displayed in Table 17. Dataset twelve, thirteen and fourteen include transaction data involving bank C only. Since the performance metric, balanced accuracy, showed a similar pattern as the F1-score, only the F1-scores are presented in Table 17.

Table 17: Comparison of F1-scores between four different machine learning models on the original data of datasets 1, 2, 3 and 12, 13, 14.

Metric	Dataset	Dtree	Ctree	Rforest	Neural Net	Average
F1-score						
	12	**0.638**	0.005	**0.764**	0.489	0.361
	13	**0.566**	0.290	**0.716**	0.264	0.326
	14	0.191	**0.323**	**0.661**	0.104	0.215
	Average	0.465	0.206	0.714	0.286	
F1-score						
	1	0.464	**0.294**	0.662	**0.512**	0.362
	2	0.433	**0.359**	0.651	**0.306**	0.337
	3	**0.281**	0.225	0.413	**0.115**	0.163
	Average	0.393	0.293	0.575	0.311	

Two findings stand out when analyzing the results of Table 17. First, the performance of the Random Forests. Beforehand it was not expected that the model performs better on transaction data that consists of one bank. The expectation that when banking institutions share transaction data, structures and typologies are easier detectable does not apply to the Random Forests

model performed on the data in this thesis. On the contrary, the Neural Network does strengthen this expectation by performing better on the data including transactions of all banks. This can be explained by the fact that Neural Networks can cope with multiple and complex data relationships and therefore can better distinguish and implement all structures of transactions between multiple banks. However, the variances in F1-scores are fairly minimal in order to make strong assumptions.

Second, as well on data with transactions from one bank as on transactions data from all banks, in general, the models have more difficulty predicting more complex money laundering typologies.

5.2. Precision and recall scores

5.2.1 Sub-question one

The precision and recall scores of the datasets belonging to the first sub-question: '*How do machine learning techniques perform on various typologies of money laundering?'* are presented in Table 18.

Table 18: Comparison of precision and recall scores between four different machine learning models on the original data of dataset 1,2 and 3.

Metric	Dataset	Dtree	Ctree	Rforest	Neural Net
Precision					
	1	0.951	1.000	0.997	0.427
	2	0.936	0.983	0.982	0.205
	3	0.951	0.884	0.984	0.625
Recall					
	1	0.307	0.172	0.496	0.638
	2	0.282	0.220	0.487	0.606
	3	0.165	0.129	0.262	0.064

The Random Forests outperformed the other models when analyzing the F1-scores. The reason of the higher F1-score can be explained because the model achieves a higher recall score than the other Tree models. A possible explanation of the higher recall score is that by combining the prediction of multiple trees into one prediction, the Random Forests misses less money laundering transactions while maintaining a high precision score.

The precision score of the Decision Tree, Conditional Inference Tree and Random Forests are high and stay relatively similar when comparing the three datasets. The explanation of these similar results can be that these models are all Tree models, and use fairly similar ways of splitting the data. This assumption is strengthened since the Neural Network shows larger

differences between the three datasets, making the Neural Network less precise. However, on average it outperforms the other models based on recall score. This indicates that the Neural Network misses less actual money laundering transactions in its prediction, but is less precise, resulting in a higher number of false positives. The Neural Network is most precise when predicting money laundering transaction typologies that have an unobvious relationship – *random, bipartite and stacked bipartite in dataset 3* - between accounts. The small differences between precision and recall scores of the Neural Network imply that it copes best with imbalanced data. With regards for the other models, large differences are visible between high precision scores and low recall scores, making the Tree models more sensitive for imbalanced data.

The difficulty to predict more complex typologies occurring in the third datasets is caused by a decrease in recall score, the precision score is fairly similar throughout all datasets except with regards to the Neural Network. This implies that on data with more complex typologies, the model only predicts money laundering transactions if the model is relatively certain that the transactions is indeed a money laundering transaction. Furthermore, since the precision and recall scores of the Neural Network do not differ as much in comparison with the other models, it can be said that the Neural Network best copes with the imbalanced data.

5.2.2. Sub-question two

The precision and recall scores of the second sub-question: *'How do machine learning techniques perform on data where only one typology of money laundering exists compared to data with multiple money laundering typologies?'* are given in Table 19.

Table 19: Comparison of precision and recall scores between four different machine learning models on the original data of datasets 4, 5, 6 and 7.

Metric	Dataset	Dtree	Ctree	Rforest	Neural Net
Precision					
	4	0.919	0.987	0.994	0.445
	5	0.962	0.954	0.998	0.472
	6	0.947	1.000	0.978	0.382
	7	0.944	0.983	0.996	0.565
Recall					
	4	0.275	0.114	0.356	0.388
	5	0.232	0.225	0.534	0.487
	6	0.216	0.103	0.347	0.330
	7	0.286	0.204	0.449	0.386

A similar trend as seen in sub-question one with regards to the precision and recall scores is visible at the results of sub-question two. The Random Forests achieves a high precision score – similar to the other Tree models – and maintains the higher recall score. This results in a higher overall F1-score as discussed in section 5.1.2.

An interesting finding is that the precision and recall scores of the Neural Network are fairly similar. This strengthens the assumption made in section 5.2.1, that the Neural Network – without any data balancing techniques – copes better with the imbalanced data. It can be argued that because of the multiple layers incorporated in the Neural Network, it more evenly distributes its prediction in terms of precision and recall. The Tree models on the other hand, are biased towards a higher precision score what negatively affects their recall score. This can be explained by the method of how the data is divided, that is used in the Tree methods.

Furthermore, the assumption that a decrease in recall caused the lower F1-score for data including more complex money laundering typologies is strengthened by the results presented in Table 19, with dataset six scoring lowest on recall for all four models.

5.2.3. Sub-question three

The precision and recall scores of the third sub-question: *'How do machine learning techniques perform on differences in the number of account interactions and periods wherein the typology is performed?'* are demonstrated in Table 20.

Table 20: Comparison of precision and recall scores between four different machine learning models on the original data of datasets 8, 9, 10 and 11.

Metric	Dataset	Dtree	Ctree	Rforest	Neural Net
Precision					
	8	0.995	0.991	0.997	0.611
	9	1.000	0.982	0.995	0.247
	10	0.997	0.991	0.994	0.425
	11	0.888	0.959	0.982	0.145
Recall					
	8	0.302	0.169	0.436	0.272
	9	0.205	0.079	0.297	0.260
	10	0.310	0.125	0.365	0.400
	11	0.227	0.088	0.361	0.446

With regard for the Tree models, the recall and precision are fairly similar when comparing data containing typologies with many account interactions within a short period of time – datasets ten and eleven – to data containing typologies with few account interactions and performed over a longer period (datasets eight and nine). The Neural Network achieves

relatively high precision scores on data containing simpler typologies – dataset eight and ten – compared to the more complex data. Furthermore, the increase in recall score of the Neural Network when typologies are performed with many accounts within a shorter period of time is interesting. This indicates that the Neural Network misses less money laundering transactions in its prediction, but on the other hand the results show the model is less precise when typologies have these characteristics. It can be said that the Neural Network is biased towards more correctly predicted money laundering transactions out of the total actual money laundering predictions, even if it is at a cost of an increase in false positives. The Neural Network recall score is not affected by the change in typology, but is affected by the changes in number of account and period. The precision of the Neural Network is on the other hand, affected by the differences in typologies and differences in number of accounts and period.

5.2.4 Sub-question four

The precision and recall results of the fourth sub-question: '*What is the difference in the performance of machine learning models when detecting suspicious transactions within one bank compared to transaction monitoring between multiple banks?'* are displayed in Table 21.

Table 21: Comparison of precision and recall scores between four different machine learning models on the original data of datasets 12, 13, 14 and 1, 2, 3.

Metric	Dataset	Dtree	Ctree	Rforest	Neural Net
Precision					
	12	0.969	**0.667**	0.998	**0.795**
	13	1.000	1.000	0.991	0.303
	14	1.000	1.000	1.000	**0.120**
Precision					
	1	0.951	1.000	0.997	**0.427**
	2	0.936	0.983	0.982	0.205
	3	0.951	0.884	0.984	**0.625**
Recall					
	12	0.475	**0.003**	0.618	0.353
	13	0.395	0.170	0.560	0.234
	14	0.106	0.192	0.493	0.092
Recall					
	1	0.307	0.172	0.496	0.638
	2	0.282	0.220	0.487	0.606
	3	0.165	0.129	0.262	0.064

The increase of performance of the Random Forests model on transactions data from one bank is explained by an increase in recall score. This is surprising since the increase in recall does not affect the precision score. Therefore, the assumption can be made that the Random Forests

performs better with transaction data of one bank than of multiple banks when comparing the models. The Neural Network, on the other hand, achieves a significantly higher precision score on transaction data from all banks with complex typologies. Once the typologies become more simplistic, the precision increases on transaction data from one bank only. Furthermore, on average the reduction of transaction data to one bank causes the recall score to decrease. Resulting in a Neural Network model that misses more money laundering transactions, but because of the increase of precision generates less false positives.

5.3. Original data versus SMOTE data

In order to compare the performance of the models on original- and SMOTE data, the F1-scores of all datasets are given in Table 22. In this section, only the differences between original data and SMOTE data are discussed.

Table 22: Comparison of the F1-scores on original- and SMOTE data of four different machine learning models on all fourteen datasets.

Dataset	Dtree	Dtree Smote	Ctree	Ctree Smote	Rforest	Rforest Smote	Neural Net	Neural Net Smote
1	0.464	0.046	0.294	0.071	0.662	0.658	0.512	0.187
2	0.433	0.044	0.359	0.092	0.651	0.612	0.306	0.199
3	0.281	0.001	0.225	0.013	0.413	0.214	0.115	0.043
4	0.423	0.040	0.205	0.035	0.524	0.526	0.414	0.243
5	0.374	0.042	0.364	0.083	0.693	0.687	0.479	0.227
6	0.352	0.024	0.187	0.032	0.512	0.398	0.354	0.103
7	0.439	0.035	0.337	0.048	0.619	0.543	0.459	0.090
8	0.463	0.021	0.288	0.032	0.607	0.467	0.376	0.103
9	0.341	0.015	0.146	0.024	0.457	0.281	0.253	0.094
10	0.471	0.045	0.223	0.037	0.534	0.528	0.412	0.250
11	0.361	0.025	0.162	0.039	0.528	0.416	0.219	0.143
12	0.638	0.032	0.005	0.054	0.764	0.739	0.489	0.165
13	0.566	0.026	0.290	0.047	0.716	0.596	0.264	0.103
14	0.191	0.008	0.323	0.015	0.661	0.382	0.104	0.038

The results of the Decision Tree and Conditional Inference Tree show a steep decline in F1-score once performed on SMOTE data. This is explained by an increase in recall score, and an even larger decrease of precision score as seen in Table 23.

Table 23: Confusion matrix dataset 1 of Decision Tree on Original data (left) and SMOTE data (right).

Predicted / Actual	0	1
0	411241	**1045**
1	*24*	***463***

Predicted / Actual	0	1
0	363544	**340**
1	*47721*	***1168***

The bold- and italic numbers indicate the recall- and precision score respectively. The confusion matrices of the other Tree models on other datasets present similar findings as the confusion matrix presented in Table 23. An explanation for the major decrease in precision – resulting in high numbers of false positives – can be that the model has more difficulty to distinguish differences between normal and money laundering transactions. On the original data the model focusses on the differences of the small percentage of money laundering transactions compared to the 99% of normal transactions, to split the data on. Once the SMOTE technique is applied, the data contains 50% money laundering transactions, it confuses the model more since the generated money laundering transactions – some of them – are fairly similar to the normal transactions, what results in a higher number of false positives, thus a lower precision score. On the other hand, having more data of money laundering transactions, it is easier for the model to miss less money laundering transactions and thus the recall score increases. However, the decrease in precision score still outweighs the increase of recall score, resulting in an overall worse model.

Table 24: Confusion matrix dataset 1 of Random Forests on Original data (left) and SMOTE data (right).

Predicted / Actual	**0**	**1**
0	411263	**765**
1	*2*	***752***

Predicted / Actual	**0**	**1**
0	411001	**639**
1	*264*	***869***

In Table 24, it is interesting to witness that the Random Forests model is not affected as much from the SMOTE data in comparison with the other Tree models. The Random Forests shows a small decline of F1-score when trained on SMOTE data, this because the precision does not decrease as much as on the other Tree models. This indicates that when taking the average of multiple trees in order to predict partially solves the problem of major decline in precision.

Table 25: Confusion matrix dataset 1 of Neural Network on Original data (left) and SMOTE data (right).

Predicted / Actual	**0**	**1**
0	409761	**545**
1	*1285*	***960***

Predicted / Actual	**0**	**1**
0	398613	**48**
1	*12652*	***1460***

Similar to the Random Forests model, the Neural Network does not show a major decline when comparing the original- and SMOTE data. This because – as seen in Table 25 – the precision

score of Neural Network decreases, but the recall score increases with fairly similar numbers. This can be explained by the fact that the Neural Network did already cope better with the imbalanced data, and therefore is not as much affected from the SMOTE technique, to balance the data.

6. Discussion

The F1-score consists of the recall and precision score, therefore a change in these scores affects the F1-score. An increase in recall score implies that the model misses less money laundering transactions out of all actual money laundering transactions. The precision score infers a deviation concerning the number of correctly predicted money laundering transactions out of all predicted money laundering transactions. A model predicting many money laundering transactions, possibly generates high numbers of false positives since only few money laundering transactions exist in the data. A decent detection model should have fairly similar scores of recall and precision, when finetuning a model, a constant consideration between the recall and precision score should be made to suit the purpose of each model. A model that has fairly similar precision- and recall scores suggests that it copes better with imbalanced data. This is necessary in real-life detection of money laundering transaction since the percentage of money laundering transactions only is a very small portion of the total transactions.

A model that consists of a low recall and high precision score implies that the model is strict when deciding on what transactions are money laundering transactions. Such a model only classifies a money laundering transaction once it is almost certain that prediction is correct. This model only predicts just a few money laundering transactions but generates a low number of false positives. Since in real-life detection, on does not want many false positives due to the fact that these transactions require time consuming human investigations to decide whether or not the transactions are indeed a money laundering transaction. On the other hand, a model that consists of a high recall and low precision score infers a model that is overpredicting the money laundering transactions. Such a model predicts high numbers of money laundering transactions, and therefore correctly predicting more money laundering transactions. However, this is at a cost of generating higher number of false positives.

It is important to fine tune a model in such a way that it suits the purpose of the model. One could argue that for the detection of money laundering transactions it is better to create a model that has less false-positives – implying fewer human investigations – and detects less money laundering transactions than a model that performs the other way around. On the other hand, in real-life detection, one wants to detect as many money laundering transactions as possible. Therefore, it is difficult to create a model that suits different purposes and is applicable on various combinations of data.

With regards to the results of sub-question one: '*How do machine learning techniques perform on various typologies of money laundering?'* it was expected that data including the more complex money laundering typologies would be harder to predict compared to the other typologies. This assumption is strengthened by the results shown in section 5.1.1. On the contrary, it is also arguable that it would be more challenging to distinguish money laundering typologies from normal typologies if both typologies have fairly similar characteristics, such as the typologies occurring in the first two datasets. However, taken the results of sub-question two into account, all results indicate that complicated typology are more challenging to predict. The problem to predict these typologies is caused by a difficulty to achieve a high recall score, the detection of these complex typologies is highly challenging and therefore the models mis more money laundering transactions. This assumption is strengthened by the results of all four sub-questions.

The Random Forests outperform all other models, on all datasets. At first, this indicates that the Random Forests is the best suitable model if one aims to detect money laundering typologies. However, an arguable disadvantage of the Random Forests is that is has – similar to the other Tree models – a high difference in recall- and precision score. This suggests that the Random Forests is not an appropriate model to use when dealing with imbalanced data. The Neural Network shows the smallest differences between recall- and precision scores. Therefore, this model could be better suitable for data with large imbalances, such as the transaction data, to detect money laundering transactions. However, a disadvantage of the Neural Network is that it does not achieve decent F1-scores, especially on the data including complex typologies it performs not as expected. Since the Neural Networks usually outperform the Tree models on data with complex relationships, it was expected that the Neural Network would perform better than the results presented in this thesis. Further research should be conducted on the Neural Network to improve the F1-scores on data to detect money laundering transactions.

The decline in F1-score of the Neural Network on the first three datasets in particular is unexpected. It can be that this decline is explained by a certain combination of typologies that the Neural Network cannot cope with. Another possible reason is the fact that the characteristic or settings from the Neural Network used in this thesis does not suit these cluster of money laundering typologies. Furthermore, one could argue that the Neural Network model is underfitted on these datasets.

Concerning the second sub-question: *'How do machine learning techniques perform on data where only one typology of money laundering exists compared to data with multiple money laundering typologies?'* the results show that there is no hard proof of a significant difference in the performance of the models on data that consist of all eight typologies compared to data including only one typology. However, the results confirm the assumption that complex typologies are most difficult to predict. One complex typology in the data makes it difficult to predict the money laundering transactions, still a cluster of complex typologies is the most difficult to predict. The differences in performance of the models on the clustered typologies in comparison with single typologies, can also be explained by the fact that the *random* and *scatter gather* typologies are better detectable than the other typologies in their cluster. Therefore, the results improved once these typologies are separated from their cluster. With regards to the *cycle* typology, it can be said that this typology is more difficult to predict than the other typologies occurring in the same cluster since the results decline once the *cycle* typology occurs solely in data.

Furthermore, it is surprising that the simpler typologies are better to predict once they are clustered in the data. At first, one would assume that data that contains only one relatively simple typology would be better to detect compared to data containing multiple relatively simple typologies.

From the results of sub-questions one and two can be concluded that the changes in F1-scores - for the Tree models - are caused by changes in recall score since the precision maintains its high score. The precision is not as affected because the Tree models' (Random Forests and Decision Tree) method of data splitting are biased towards variables that have higher possible splits. However, this could indicate that if these models are improved in order to generate a higher recall score while maintaining a similar precision score, the overall F1-score increases. Therefore, are more interesting to use in the field of money laundering detection.

With regards to the third sub-question: *'How do machine learning techniques perform on differences in the number of account interactions and periods wherein the typology is performed?'* one would argue that a typology performed with many account interactions and within a short period of time is better detectable since these are unusual transactions characteristics of normal transactions. However, since the differences in results are fairly small, it is hard to make certain assumptions. There is no clear assumption made on what characteristics of money laundering typologies are better detectable when comparing the data with typologies performed with many account interactions in a short period of time to data with

typologies performed with few accounts and over a longer period. Some explanations can be given why there are no larger differences visible between the changes in parameters. First, the changes in parameters are too small in order to differentiate the typologies from each other. Second, these patterns and parameters are not valuable to split large amount of data and therefore, are not suitable to use as thresholds for detection purposes. Third, the variables in the datasets are not capable of identifying these differences in parameters.

The results of the fourth sub-question: '*What is the difference in the performance of machine learning models when detecting suspicious transactions within one bank compared to transaction monitoring between multiple banks?'* show marginal results in order to make strong assumptions. With two models (Decision Tree and Random Forests) performing better on data containing transactions from one bank, and the Conditional Inference Tree and Neural Network performing better on transaction data from all banks, the models are fairly evenly distributed. An explanation of the improved results of the two Tree models can be that since the data from one bank has fewer complex structures – due to the fact that typology structures are restricted to one bank – these models performance excels. Where on the other hand, the Neural Network improves on data with complete typology structures since it performs better when having more data relationships to train on. However, because this evenly distribution between the models on the different datasets the assumptions made are fairly weak. A reason for the small differences between one bank data and multiple bank data can be that because bank C is the largest bank (containing 8000 registered accounts) and therefore most money laundering typologies include an account from bank c. In order to conclude this, further research is needed to compare different datasets from various bank transactions.

The comparison of performance of machine learning models on original data and SMOTE data show large differences. The performance measured in F1-score is lower for every model trained on SMOTE data compared to the original, imbalanced data. This is surprising in a way, since it is arguable that a model trained on balanced data has more money laundering transactions to define certain characteristics of these transactions in order to distinguish the transactions better and improve the performance of the model. However, on the other hand, one can argue that the number of normal transactions decreases on balanced data, and therefore the model has less input to distinguish normal transactions from a small portion of money laundering transactions. Furthermore, since the SMOTE technique does not exactly copy the money laundering transactions and duplicates them, instead small variations of money laundering transactions (as the existing ones) are created, implies that the model has more 'different' money laundering

transactions to detect compared to the original data. Therefore, it is more challenging to detect money laundering transactions correctly.

However, this does not explain the difference in machine learning models, the decrease in performance is less for the Random Forests and Neural Network model. The study from Alvarez-Jareno, Badal-Valero, & Pavia (2017) revealed that the Random Forests results were improved on SMOTE data. However, it implied that due to the increase of true positives and similar accuracy, the false positives increased. First, the SMOTE technique does not improve the results in this thesis. Second, there is a small increase in false-positives as mentioned by Alvarez-Jareno, Badal-Valero, & Pavia (2017). With regards for the Random Forests and Neural Network models, the performance is not decreasing as much since the precision score is not affected as heavily as the Decision Tree and Conditional Inference Tree models. These models generate high numbers of false positives on balanced data.

The Neural Network and Random Forests better cope with balanced data, since the precision score slightly decreases and the recall score slightly increases. This can be explained since these models' better deal with more complex data having multiple variations of relationships. Since the Neural Network, does not show large differences for the recall- and precision score, and is not as affected in differences of data imbalance, the Neural Network is most suitable for the detections of money laundering transactions. Lv, Ji, & Zhang (2008) state that the Neural Network reduces the false-positive rate and increases the recall score. The results from this thesis support these statements since the Neural Network achieves a relatively low false-positive rate and high recall score in comparison with the other models. However, the Neural network needs improvement in order to achieve a higher F1-score.

The Random Forests on the other hand, achieves a high F1-score on all datasets researched, and copes fairly decent with data imbalance. Once the Random Forests recall score can be increased, without significant decreasing the precision score it can be used as a decent model in the field of anti-money laundering detection. The other Tree models underperform compared to the Neural Network and Random Forests based on prediction metrics, and therefore could best be used to give insights on the impact of certain variables on the detection decisions made by the model. According to ING (2020) these insights are as imporant as high prediction metrics since investigators highly value interpretabilty. However, for accurate prediction these models need to be improved.

7. Conclusion

To conclude and answer the question: *'What are the benefits and limitations of machine learning techniques applied in the Anti-Money Laundering financial transaction network typology detection process?'* first a short summary of each sub-question is given. Since the study from this thesis provides a broad comparison of various models, datasets and techniques it is difficult to clarify specific reasons of certain changes in the performance metrics.

In relation to the first sub-question, machine learning models have difficulty predicting money laundering typologies that have more complex typologies. On the other hand, all four models perform fairly equal on data that contains more simpler money laundering typologies. Furthermore, the Random Forests model outperforms all other techniques because of a higher recall score, this is the case for all results of each sub-question.

The results of the second sub-question, does not imply that data containing only one money laundering typology is easier to predict using machine learning models compared to data including multiple money laundering typologies. However, from the precision- and recall scores of sub-question two, it can be concluded that the Neural Network best copes with de imbalanced data.

With regards to the third sub-question, there are no significant differences in F1-scores of the performance of machine learning models on money laundering typologies with shorter periods and more account interactions compared to longer periods and fewer account interactions. However, because of the increase in recall score of the Neural Network it can be said that the money laundering typologies with shorter periods and more account interactions are easier distinguishable. Also, it confirms that machine learning models find it more difficult to predict – more complex - money laundering typologies with unobvious account relationships than – more simplistic - money laundering typologies having obvious account relationships.

The fourth and last sub-question, does not provide results to conclude that having data from multiple bank transactions improve the performance of the machine learning models since Decision Tree and Random Forests perform better on data with transactions from only one bank. On the other hand, the Conditional Inference Tree and Neural Network perform better on data with transactions from multiple banks.

To conclude, machine learning models, Random Forests and in particular, could create beneficial contributions to the money laundering detection process by accurately predicting money laundering transactions from normal transactions. Limitations of the machine learning techniques arise when there are unobvious relationships between accounts. In addition, Neural

Networks deal best with the highly imbalanced data, and therefore could assist in real-world scenarios of anti-money laundering detections purposes. The Decision Tree and Conditional Inference Tree could be used to provide clarifying insights of money laundering typologies because of their interpretability. However, in order to provide substantiated conclusions and more precise answers to the sub questions, further research is needed.

7.1. Limitations

Besides the limitations of the data discussed in section 3.3 - that the data does not include low- and high-risk countries, does not incorporate cash deposits and withdrawals, and only simulates a couple of money laundering typologies - additional limitations arose during the methodology section of this research. The first and foremost limitation of this research is that the data and analysis is done on synthetic data. Therefore, it is difficult to use the results and interpretation of the outcome for real-world scenarios. The second limitation is that only a specific number of money laundering typologies are included in this thesis, with the endless possibilities to launder money, only a very specific part of money laundering is discussed in this thesis. This also applies for the different stages of money laundering, this research focusses on the second phase of money laundering, the layering phase. Therefore, a limitation is that the other phases are not included in the research of this thesis. The third limitation of the research is that each machine learning model could be more precisely finetuned for various money laundering typologies. The trained models use the default – average - settings, this because the focus of this research is on the differences between machine learning models and different generated datasets, rather than differences in performance when finetuning specific machine learning models.

7.2. Further research

In order to provide substantiated conclusions, further research should be conducted since the detection of money laundering transactions is a complex task. Further research can bring clarification of limitations and unanswered questions from this thesis. First, further research should be done using a more advance simulation model, that includes cash deposits and withdrawals, and incorporates low- and high-risk countries. Also, researching multiple and other money laundering typologies can bring different insights. Second, the machine learning models could be researched individually to increase the performance of each model and analyze how and why the changes affect certain prediction metrics. This would be useful to create a model where neither the recall or precision score are negatively affected by changes in the data.

Once a model is improved on certain data, it is necessary to compare the performance of the improved model to the – currently used - rule-based method. A decision can be made whether the model is suitable to use as replacement or addition to the rule-based method. At last, the most important to research in the future is that the results and interpretation of the outcomes should be validated on real-life data, to substantiate the performance of machine learning models on the detection of money laundering transactions.

8. Appendix

8.1. Tables

Table 1: The money laundering pattern input file for the AMLSim.

C.	Trans_ typology	Sched_ id	Min_ acc	Max_ acc	Min_ amount	Max_ amount	Min_ period	Max_ period	Bank_ id	Is_sar
15	Fan_ in	2	2	5	100	200	5	10	Bank a	True
15	Cycle	2	5	7	100	200	5	10	Bank c	False

Table 2: Generated dataset of financial transactions.

Number	Variable	Explanation
1	Step	The step in the simulation of the executed transaction
2	Type	The type of transaction performed
3	Amount	The amount involved in a transaction
4	Name origin	The name of the sending account
5	Old balance origin	The old balance of the sending account
6	New balance origin	The new balance of the sending account
7	Name dest	The name of the receiving account
8	Old balance dest	The old balance of the receiving account
9	New balance dest	The new balance of the receiving account
10	Is SAR	Whether the transaction is flagged as SAR (1) or non SAR (0)
11	Alert ID	The alert group of transaction and accounts involved in a similar pattern. Each group has its own ID

Table 3: Generated dataset of financial accounts.

Number	Variable	Explanation
1	Account ID	The ID number of each account
2	Customer ID	The customer ID number, similar to account ID
3	Initial balance	The initial balance of an account when opened
4	Start date	The date when account is opened
5	End date	The date when account is closed
6	Country	The country where the account is located
7	Account type	The type of account
8	Is SAR	Whether the account is flagged as SAR (1) or non SAR (0)
9	Tx behavior ID	The account behavior, what typologies the account performs
10	Bank ID	The bank ID of an account, to which bank the account belongs to

Table 4: Final dataset including generated variables.

Number	Variable	Explanation
1	Sar	Whether the transaction is flagged as SAR (1) or non SAR (0)
2	Timestamp	The step in the simulation of the executed transaction
3	Value	The amount involved in a transaction
4	Initial balance	The initial balance of the origin account when opened
5	Old balance orig	The old balance of the sending account
6	New balance orig	The new balance of the sending account
7	Old balance dest	The new balance of the receiving account
8	New balance dest	The new balance of the receiving account
9	Count in orig	The number of transactions the sending account has received from another account during the time period of the simulation
10	Count out orig	The number of transactions the sending account has made to another account during the time period of the simulation
11	Total count orig	The total number of transactions the sending account has made and received during the time period of the simulation
12	Count interactions in orig	The number of interactions the sending account has with other unique accounts when receiving transactions
13	Count interactions out orig	The number of interactions the sending account has with other unique accounts when sending transactions
14	Min interval in orig	The minimum interval between two incoming transactions from the sending account
15	Max interval in orig	The maximum interval between two incoming transactions from the sending account
16	Min interval out oirg	The minimum interval between two outgoing transactions from the sending account
17	Max interval out orig	The maximum interval between two outgoing transactions from the sending account
18	Average amount in orig	The average amount incoming of the sending account
19	Average amount out orig	The average amount outgoing of the sending account
20	Sum amount in oirg	The total amount incoming of the sending account
21	Sum amount out orig	The total amount outgoing of the sending account
22	Max amount in orig	The maximum amount incoming of the sending account
23	Max amount out orig	The maximum amount outgoing of the sending account
24	Min amount in orig	The minimum amount incoming of the sending account
25	Min amount out orig	The minimum amount outgoing of the sending account
26	Count in dest	The number of transactions the receiving account has received from another account during the time period of the simulation
27	Count out dest	The number of transactions the receiving account has made to another account during the time period of the simulation
28	Total count dest	The total number of transactions the receiving account has made and received during the time period of the simulation
29	Count interactions in dest	The number of interactions the receiving account has with other unique accounts when sending transactions
30	Count interactions out dest	The number of interactions the receiving account has with other unique accounts when receiving transactions
31	Min interval in dest	The minimum interval between two incoming transactions from the receiving account
32	Max interval in dest	The maximum interval between two incoming transactions from the receiving account
33	Min interval out oirg	The minimum interval between two outgoing transactions from the receiving account
34	Max interval out dest	The maximum interval between two outgoing transactions from the receiving account
35	Average amount in dest	The average amount incoming of the receiving account
36	Average amount out dest	The average amount outgoing of the receiving account
37	Sum amount in deat	The total amount incoming of the receiving account
38	Sum amount out dest	The total amount outgoing of the receiving account
39	Max amount in dest	The maximum amount incoming of the receiving account

40	Max amount out dest	The maximum amount outgoing of the receiving account
41	Min amount in dest	The minimum amount incoming of the receiving account
42	Min amount out dest	The minimum amount outgoing of the receiving account
43	Bank a orig	Dummy variable if sending account belongs to bank a
44	Bank b orig	Dummy variable if sending account belongs to bank b
45	Bank c orig	Dummy variable if sending account belongs to bank c
46	Bank d orig	Dummy variable if sending account belongs to bank d
47	Bank e orig	Dummy variable if sending account belongs to bank e
48	Bank a dest	Dummy variable if receiving account belongs to bank a
49	Bank b dest	Dummy variable if receiving account belongs to bank b
50	Bank c dest	Dummy variable if receiving account belongs to bank c
51	Bank d dest	Dummy variable if receiving account belongs to bank d
52	Bank e dest	Dummy variable if receiving account belongs to bank e

Table 5: The money laundering pattern input file for the first dataset created by the AMLSim.

C.	**Trans _typology**	**Sched_ id**	**Min_ acc**	**Max_ acc**	**Min_ amount**	**Max_ amount**	**Min_ period**	**Max_ period**	**Bank _id**	**Is_sar**
400	Fan_in	2	4	10	30	1000	10	30	“”	True
400	Fan_out	2	4	10	30	1000	10	30	“”	True
400	Cycle	2	4	10	30	1000	10	30	“”	True

Table 6: Overview of different datasets generated.

	Typologies	Other alterations
Research sub question 1		
Dataset 1	Fan in, Fan out, Cycle	-
Dataset 2	Scatter Gather, Gather Scatter	-
Dataset 3	Bipartite, Stacked Bipartite, Random	-
Research sub question 2		
Dataset 4	Cycle	-
Dataset 5	Scatter Gather	-
Dataset 6	Random	-
Dataset 7	Fan in, Fan out, Cycle, Scatter Gather, Gather Scatter, Bipartite, Stacked Bipartite and Random	-
Research sub question 3		
Dataset 8	Cycle	Min- & Max account interactions is set to 2 and 5. Min- & Max period is set to 20 and 40.
Dataset 9	Random	Min- & Max account interactions is set to 2 and 5. Min- & Max period is set to 20 and 40.
Dataset 10	Cycle	Min- & Max account interactions is set to 10 and 15. Min- & Max period is set to 5 and 20.
Dataset 11	Random	Min- & Max account interactions is set to 10 and 15. Min- & Max period is set to 5 and 20.
Research sub question 4		
Dataset 12	Fan in, Fan out, Cycle	Filtered for Bank C accounts
Dataset 13	Scatter Gather, Gather Scatter	Filtered for Bank C accounts
Dataset 14	Bipartite, Stacked Bipartite, Random	Filtered for Bank C accounts

Table 7: Example of a confusion matrix to analyze the results.

Predicted / Actual	**0**	**1**
0	5300 (True negative)	250 (False negative)
1	1 (False positive)	280 (True positive)

Table 8: F1-scores of each machine learning model on each dataset.

Dataset	Dtree	Dtree Smote	Ctree	Ctree Smote	Rforest	Rforest Smote	Neural Net	Neural Net Smote
1	*0.464*	*0.046*	*0.294*	*0.071*	*0.662*	*0.658*	*0.512*	*0.187*
2	*0.433*	*0.044*	*0.359*	*0.092*	*0.651*	*0.612*	*0.306*	*0.199*
3	*0.281*	*0.001*	*0.225*	*0.013*	*0.413*	*0.214*	*0.115*	*0.043*
4	*0.423*	*0.040*	*0.205*	*0.035*	*0.524*	*0.526*	*0.414*	*0.243*
5	*0.374*	*0.042*	*0.364*	*0.083*	*0.693*	*0.687*	*0.479*	*0.227*
6	*0.352*	*0.024*	*0.187*	*0.032*	*0.512*	*0.398*	*0.354*	*0.103*
7	*0.439*	*0.035*	*0.337*	*0.048*	*0.619*	*0.543*	*0.459*	*0.090*
8	*0.463*	*0.021*	*0.288*	*0.032*	*0.607*	*0.467*	*0.376*	*0.103*
9	*0.341*	*0.015*	*0.146*	*0.024*	*0.457*	*0.281*	*0.253*	*0.094*
10	*0.471*	*0.045*	*0.223*	*0.037*	*0.534*	*0.528*	*0.412*	*0.250*
11	*0.361*	*0.025*	*0.162*	*0.039*	*0.528*	*0.416*	*0.219*	*0.143*
12	*0.638*	*0.032*	*0.005*	*0.054*	*0.764*	*0.739*	*0.489*	*0.165*
13	*0.566*	*0.026*	*0.290*	*0.047*	*0.716*	*0.596*	*0.264*	*0.103*
14	*0.191*	*0.008*	*0.323*	*0.015*	*0.661*	*0.382*	*0.104*	*0.038*

Table 9: Precision scores of each machine learning model on each dataset.

Dataset	Dtree	Dtree Smote	Ctree	Ctree Smote	Rforest	Rforest Smote	Neural Net	Neural Net Smote
1	*0.951*	*0.024*	*1.000*	*0.037*	*0.997*	*0.767*	*0.427*	*0.103*
2	*0.936*	*0.023*	*0.983*	*0.049*	*0.982*	*0.644*	*0.205*	*0.111*
3	*0.951*	*0.000*	*0.884*	*0.006*	*0.984*	*0.161*	*0.625*	*0.022*
4	*0.919*	*0.021*	*0.987*	*0.018*	*0.994*	*0.635*	*0.445*	*0.140*
5	*0.962*	*0.021*	*0.954*	*0.044*	*0.998*	*0.724*	*0.472*	*0.129*
6	*0.947*	*0.012*	*1.000*	*0.016*	*0.978*	*0.385*	*0.382*	*0.054*
7	*0.944*	*0.018*	*0.983*	*0.025*	*0.996*	*0.606*	*0.565*	*0.047*
8	*0.995*	*0.011*	*0.991*	*0.016*	*0.997*	*0.432*	*0.611*	*0.055*
9	*1.000*	*0.008*	*0.982*	*0.012*	*0.995*	*0.231*	*0.247*	*0.049*
10	*0.997*	*0.023*	*0.991*	*0.019*	*0.994*	*0.768*	*0.425*	*0.143*
11	*0.888*	*0.013*	*0.959*	*0.020*	*0.982*	*0.442*	*0.145*	*0.077*
12	*0.969*	*0.016*	*0.667*	*0.028*	*0.998*	*0.744*	*0.795*	*0.090*
13	*1.000*	*0.013*	*1.000*	*0.024*	*0.991*	*0.538*	*0.303*	*0.054*
14	*1.000*	*0.004*	*1.000*	*0.008*	*1.000*	*0.286*	*0.120*	*0.019*

Table 10: Recall of each machine learning model on each dataset.

Dataset	Dtree	Dtree Smote	Ctree	Ctree Smote	Rforest	Rforest Smote	Neural Net	Neural Net Smote
1	*0.307*	*0.775*	*0.172*	*0.777*	*0.496*	*0.576*	*0.638*	*0.968*
2	*0.282*	*0.832*	*0.220*	*0.788*	*0.487*	*0.582*	*0.606*	*0.961*
3	*0.165*	*0.787*	*0.129*	*0.812*	*0.262*	*0.319*	*0.064*	*0.979*
4	*0.275*	*0.759*	*0.114*	*0.810*	*0.356*	*0.449*	*0.388*	*0.936*
5	*0.232*	*0.868*	*0.225*	*0.849*	*0.534*	*0.654*	*0.487*	*0.968*
6	*0.216*	*0.850*	*0.103*	*0.846*	*0.347*	*0.412*	*0.330*	*0.995*
7	*0.286*	*0.766*	*0.204*	*0.788*	*0.449*	*0.492*	*0.386*	*0.987*
8	*0.302*	*0.817*	*0.169*	*0.865*	*0.436*	*0.509*	*0.272*	*0.957*
9	*0.205*	*0.831*	*0.079*	*0.836*	*0.297*	*0.358*	*0.260*	*0.929*
10	*0.310*	*0.752*	*0.125*	*0.816*	*0.365*	*0.402*	*0.400*	*0.974*
11	*0.227*	*0.737*	*0.088*	*0.767*	*0.361*	*0.393*	*0.446*	*0.979*
12	*0.475*	*0.822*	*0.003*	*0.848*	*0.618*	*0.734*	*0.353*	*0.942*
13	*0.395*	*0.790*	*0.170*	*0.884*	*0.560*	*0.668*	*0.234*	*0.954*
14	*0.106*	*0.770*	*0.192*	*0.772*	*0.493*	*0.572*	*0.092*	*0.932*

Table 11: Balanced accuracy of each machine learning model on each dataset.

Dataset	Dtree	Dtree Smote	Ctree	Ctree Smote	Rforest	Rforest Smote	Neural Net	Neural Net Smote
1	*0.653*	*0.829*	*0.586*	*0.852*	*0.748*	*0.788*	*0.817*	*0.969*
2	*0.641*	*0.835*	*0.610*	*0.859*	*0.744*	*0.790*	*0.798*	*0.963*
3	*0.582*	*0.541*	*0.564*	*0.834*	*0.631*	*0.658*	*0.532*	*0.964*
4	*0.637*	*0.822*	*0.557*	*0.834*	*0.678*	*0.724*	*0.693*	*0.959*
5	*0.616*	*0.840*	*0.613*	*0.881*	*0.767*	*0.827*	*0.742*	*0.969*
6	*0.608*	*0.828*	*0.552*	*0.852*	*0.673*	*0.705*	*0.664*	*0.973*
7	*0.643*	*0.826*	*0.602*	*0.852*	*0.725*	*0.746*	*0.693*	*0.966*
8	*0.651*	*0.846*	*0.584*	*0.889*	*0.718*	*0.754*	*0.636*	*0.965*
9	*0.603*	*0.824*	*0.539*	*0.861*	*0.648*	*0.678*	*0.629*	*0.949*
10	*0.655*	*0.808*	*0.563*	*0.817*	*0.682*	*0.701*	*0.699*	*0.974*
11	*0.613*	*0.795*	*0.544*	*0.835*	*0.681*	*0.696*	*0.720*	*0.975*
12	*0.738*	*0.865*	*0.501*	*0.896*	*0.809*	*0.867*	*0.676*	*0.962*
13	*0.679*	*0.835*	*0.585*	*0.906*	*0.780*	*0.834*	*0.616*	*0.960*
14	*0.553*	*0.798*	*0.596*	*0.837*	*0.747*	*0.785*	*0.546*	*0.943*

Table 12: Comparison of F1-scores and balanced accuracy between four different machine learning models on the original data of datasets 1 till 3.

Metric	Dataset	Dtree	Ctree	Rforest	Neural Net	Average
F1-score						
	1	**0.464**	0.294	**0.662**	**0.512**	0.362
	2	0.433	**0.359**	0.651	0.306	0.337
	3	0.281	0.225	0.413	0.115	0.163
	Average	0.393	0.293	0.575	0.311	
Balanced accuracy						
	1	**0.653**	0.586	**0.748**	**0.817**	0.780
	2	0.641	**0.610**	0.744	0.798	0.780
	3	0.582	0.564	0.631	0.532	0.663
	Average	0.625	0.587	0.708	0.716	

Table 13: Comparison of F1-scores and balanced accuracy between four different machine learning models on the original data of datasets 4 till 7.

Metric	Dataset	Dtree	Ctree	Rforest	Neural Net	Average
F1-score						
	4	0.423	0.205	0.524	0.414	0.301
	5	0.374	**0.364**	**0.693**	**0.479**	0.369
	6	0.352	0.187	0.512	0.354	0.245
	7	**0.439**	0.337	0.619	0.459	0.321
	Average	0.397	0.273	0.587	0.427	
Balanced accuracy						
	4	0.637	0.557	0.678	0.693	0.738
	5	0.616	**0.613**	**0.767**	**0.742**	0.782
	6	0.608	0.552	0.673	0.664	0.732
	7	**0.643**	0.602	0.725	0.693	0.757

Table 14: Comparison of F1-scores between four different machine learning models on the original data of datasets 1 till 7.

Metric	Dataset	Dtree	Ctree	Rforest	Neural Net	Average
F1-score Q1						
	1	**0.464**	**0.294**	**0.662**	**0.512**	0.362
	2	**0.433**	0.359	0.651	0.306	0.337
	3	0.281	**0.225**	0.413	0.115	0.163
	Average	0.393	0.293	0.575	0.311	
F1-score Q2						
	4	0.423	0.205	0.524	0.414	0.301
	5	0.374	**0.364**	**0.693**	**0.479**	0.369
	6	**0.352**	0.187	**0.512**	**0.354**	0.245
	7	0.439	0.337	0.619	0.459	0.321
	Average	0.397	0.273	0.587	0.427	

Table 15: Comparison of F1-scores and balanced accuracy between four different machine learning models on the original data of datasets 8 till 11.

Metric	Dataset	Dtree	Ctree	Rforest	Neural Net	Average
F1-score						
	8	0.463	**0.288**	**0.607**	0.376	0.295
	10	**0.471**	0.223	0.534	**0.412**	**0.313**
	9	0.341	0.146	0.457	**0.253**	0.201
	11	**0.361**	**0.162**	**0.528**	0.219	**0.237**
	Average	0.409	0.205	0.532	0.315	
Balanced accuracy						
	8	0.651	**0.584**	**0.718**	0.636	0.755
	10	**0.655**	0.563	0.682	**0.699**	0.737
	9	0.603	0.539	0.648	0.629	0.716
	11	**0.613**	**0.544**	**0.681**	**0.720**	0.732

Table 16: Comparison of F1-scores between four different machine learning models on the original data of datasets 4, 6, and 8 till 11.

Metric	Dataset	Dtree	Ctree	Rforest	Neural Net	Average
F1-score						
	4	0.423	0.205	0.524	**0.414**	0.301
	8	0.463	**0.288**	**0.607**	0.376	0.295
	10	**0.471**	0.223	0.534	0.412	**0.313**
	6	0.352	**0.187**	0.512	**0.354**	**0.245**
	9	0.341	0.146	0.457	0.253	0.201
	11	**0.361**	0.162	**0.528**	0.219	0.237

Table 17: Comparison of F1-scores between four different machine learning models on the original data of datasets 1, 2, 3 and 12, 13, 14.

Metric	Dataset	Dtree	Ctree	Rforest	Neural Net	Average
F1-score						
	12	**0.638**	0.005	**0.764**	0.489	0.361
	13	**0.566**	0.290	**0.716**	0.264	0.326
	14	0.191	**0.323**	**0.661**	0.104	0.215
	Average	0.465	0.206	0.714	0.286	
F1-score						
	1	0.464	**0.294**	0.662	**0.512**	0.362
	2	0.433	**0.359**	0.651	**0.306**	0.337
	3	**0.281**	0.225	0.413	**0.115**	0.163
	Average	0.393	0.293	0.575	0.311	

Table 18: Comparison of precision and recall scores between four different machine learning models on the original data of dataset 1,2 and 3.

Metric	Dataset	Dtree	Ctree	Rforest	Neural Net
Precision					
	1	0.951	1.000	0.997	0.427
	2	0.936	0.983	0.982	0.205
	3	0.951	0.884	0.984	0.625
Recall					
	1	0.307	0.172	0.496	0.638
	2	0.282	0.220	0.487	0.606
	3	0.165	0.129	0.262	0.064

Table 19: Comparison of precision and recall scores between four different machine learning models on the original data of datasets 4, 5, 6 and 7.

Metric	Dataset	Dtree	Ctree	Rforest	Neural Net
Precision					
	4	0.919	0.987	0.994	0.445
	5	0.962	0.954	0.998	0.472
	6	0.947	1.000	0.978	0.382
	7	0.944	0.983	0.996	0.565
Recall					
	4	0.275	0.114	0.356	0.388
	5	0.232	0.225	0.534	0.487
	6	0.216	0.103	0.347	0.330
	7	0.286	0.204	0.449	0.386

Table 20: Comparison of precision and recall scores between four different machine learning models on the original data of datasets 8, 9, 10 and 11.

Metric	Dataset	Dtree	Ctree	Rforest	Neural Net
Precision					
	8	0.995	0.991	0.997	0.611
	9	1.000	0.982	0.995	0.247
	10	0.997	0.991	0.994	0.425
	11	0.888	0.959	0.982	0.145
Recall					
	8	0.302	0.169	0.436	0.272
	9	0.205	0.079	0.297	0.260
	10	0.310	0.125	0.365	0.400
	11	0.227	0.088	0.361	0.446

Table 21: Comparison of precision and recall scores between four different machine learning models on the original data of datasets 12, 13, 14 and 1, 2, 3.

Metric	Dataset	Dtree	Ctree	Rforest	Neural Net
Precision					
	12	0.969	**0.667**	0.998	**0.795**
	13	1.000	1.000	0.991	0.303
	14	1.000	1.000	1.000	**0.120**
Precision					
	1	0.951	1.000	0.997	**0.427**
	2	0.936	0.983	0.982	0.205
	3	0.951	0.884	0.984	**0.625**
Recall					
	12	0.475	**0.003**	0.618	0.353
	13	0.395	0.170	0.560	0.234
	14	0.106	0.192	0.493	0.092
Recall					
	1	0.307	0.172	0.496	0.638
	2	0.282	0.220	0.487	0.606
	3	0.165	0.129	0.262	0.064

Table 22: Comparison of the F1-scores on original- and SMOTE data of four different machine learning models on all fourteen datasets.

Dataset	Dtree	Dtree Smote	Ctree	Ctree Smote	Rforest	Rforest Smote	Neural Net	Neural Net Smote
1	0.464	0.046	0.294	0.071	0.662	0.658	0.512	0.187
2	0.433	0.044	0.359	0.092	0.651	0.612	0.306	0.199
3	0.281	0.001	0.225	0.013	0.413	0.214	0.115	0.043
4	0.423	0.040	0.205	0.035	0.524	0.526	0.414	0.243
5	0.374	0.042	0.364	0.083	0.693	0.687	0.479	0.227
6	0.352	0.024	0.187	0.032	0.512	0.398	0.354	0.103
7	0.439	0.035	0.337	0.048	0.619	0.543	0.459	0.090
8	0.463	0.021	0.288	0.032	0.607	0.467	0.376	0.103
9	0.341	0.015	0.146	0.024	0.457	0.281	0.253	0.094
10	0.471	0.045	0.223	0.037	0.534	0.528	0.412	0.250
11	0.361	0.025	0.162	0.039	0.528	0.416	0.219	0.143
12	0.638	0.032	0.005	0.054	0.764	0.739	0.489	0.165
13	0.566	0.026	0.290	0.047	0.716	0.596	0.264	0.103
14	0.191	0.008	0.323	0.015	0.661	0.382	0.104	0.038

Table 23: Confusion matrix dataset 1 of Decision Tree on Original data (left) and SMOTE data (right).

Predicted / Actual	**0**	**1**
0	411241	**1045**
1	*24*	***463***

Predicted / Actual	**0**	**1**
0	363544	**340**
1	*47721*	***1168***

Table 24: Confusion matrix dataset 1 of Random Forests on Original data (left) and SMOTE data (right).

Predicted / Actual	**0**	**1**
0	411263	**765**
1	*2*	***752***

Predicted / Actual	**0**	**1**
0	411001	**639**
1	*264*	***869***

Table 25: Confusion matrix dataset 1 of Neural Network on Original data (left) and SMOTE data (right).

Predicted / Actual	**0**	**1**
0	409761	**545**
1	*1285*	***960***

Predicted / Actual	**0**	**1**
0	398613	**48**
1	*12652*	***1460***

8.2. Datasets

Dataset 1: Alert pattern input parameters for dataset containing *fan in, fan out, cycle typologies.*

Count	Typology	Sched	Min acc	Max acc	Min am	Max am	Min per	Max per	Bank id	Is sar
400	**Fan in**	2	4	10	30	1000	10	30	“”	True
400	**Fan out**	2	4	10	30	1000	10	30	“”	True
400	**Cycle**	2	4	10	30	1000	10	30	“”	True

Dataset 2: Alert pattern input parameters for dataset containing containing *scatter gather* and *gather scatter* typologies.

Count	Typology	Sched	Min acc	Max acc	Min am	Max am	Min per	Max per	Bank id	Is sar
600	**Scatter gather**	2	4	10	30	1000	10	30	“”	True
600	**Gather scatter**	2	4	10	30	1000	10	30	“”	True

Dataset 3: Alert pattern input parameters for dataset containing *bipartite, stacked bipartite,* and *random* typologies.

Count	Typology	Sched	Min acc	Max acc	Min am	Max am	Min per	Max per	Bank id	Is sar
400	**Bipartite**	2	4	10	30	1000	10	30	“”	True
400	**Stacked bipartite**	2	4	10	30	1000	10	30	“”	True
400	**random**	2	4	10	30	1000	10	30	“”	True

Dataset 4: Alert pattern input parameters for dataset containing *cycle* typology.

Count	Typology	Sched	Min acc	Max acc	Min am	Max am	Min per	Max per	Bank id	Is sar
1000	**Cycle**	2	4	10	30	1000	10	30	“”	True

Dataset 5: Alert pattern input parameters for dataset containing *scatter gather* typology.

Count	Typology	Sched	Min acc	Max acc	Min am	Max am	Min per	Max per	Bank id	Is sar
1000	**Scatter gather**	2	4	10	30	1000	10	30	“”	True

Dataset 6: Alert pattern input parameters for dataset containing *random* typology.

Count	Typology	Sched	Min acc	Max acc	Min am	Max am	Min per	Max per	Bank id	Is sar
1000	**Random**	2	4	10	30	1000	10	30	“”	True

Dataset 7: Alert pattern input parameters for dataset containing *fan in, fan out, cycle, scatter gather, gather scatter, bipartite, stacked bipartite* and *random* typologies.

Count	Typology	Sched	Min acc	Max acc	Min am	Max am	Min per	Max per	Bank id	Is sar
135	**Fan in**	2	4	10	30	1000	10	30	“”	True
135	**Fan out**	2	4	10	30	1000	10	30	“”	True
135	**Cycle**	2	4	10	30	1000	10	30	“”	True
135	**Scatter gather**	2	4	10	30	1000	10	30	“”	True
135	**Gather scatter**	2	4	10	30	1000	10	30	“”	True
135	**Bipartite**	2	4	10	30	1000	10	30	“”	True
135	**Stacked bipartite**	2	4	10	30	1000	10	30	“”	True
135	**Random**	2	4	10	30	1000	10	30	“”	True

Dataset 8: Alert pattern input parameters for dataset containing *cycle* typology.

Count	Typology	Sched	Min acc	Max acc	Min am	Max am	Min per	Max per	Bank id	Is sar
1000	Cycle	2	**2**	**5**	30	1000	**20**	**40**	“”	True

Dataset 9: Alert pattern input parameters for dataset containing *random* typology.

Count	Typology	Sched	Min acc	Max acc	Min am	Max am	Min per	Max per	Bank id	Is sar
1000	Random	2	**2**	**5**	30	1000	**20**	**40**	“”	True

Dataset 10: Alert pattern input parameters for dataset containing *cycle* typology.

Count	Typology	Sched	Min acc	Max acc	Min am	Max am	Min per	Max per	Bank id	Is sar
1000	Cycle	2	**10**	**15**	30	1000	**5**	**20**	“”	True

Dataset 11: Alert pattern input parameters for dataset containing *random* typology.

Count	Typology	Sched	Min acc	Max acc	Min am	Max am	Min per	Max per	Bank id	Is sar
1000	Random	2	**10**	**15**	30	1000	**5**	**20**	“”	True

Dataset 12: Alert pattern input parameters for dataset containing *fan in*, *fan out* and *cycle* typologies. ***(Filtered accounts of bank C)***

Count	Typology	Sched	Min acc	Max acc	Min am	Max am	Min per	Max per	Bank id	Is sar
400	**Fan in**	2	4	10	30	1000	10	30	“”	True
400	**Fan out**	2	4	10	30	1000	10	30	“”	True
400	**Cycle**	2	4	10	30	1000	10	30	“”	True

Dataset 13: Alert pattern input parameters for dataset containing *scatter gather* and *gather scatter* typologies. ***(Filtered accounts of bank C)***

Count	Typology	Sched	Min acc	Max acc	Min am	Max am	Min per	Max per	Bank id	Is sar
600	**Scatter gather**	2	4	10	30	1000	10	30	“”	True
600	**Gather scatter**	2	4	10	30	1000	10	30	“”	True

Dataset 14: Alert pattern input parameters for dataset containing *bipartite, stacked bipartite*, and *random* typologies. ***(Filtered accounts of bank C)***

Count	Typology	Sched	Min acc	Max acc	Min am	Max am	Min per	Max per	Bank id	Is sar
400	**Bipartite**	2	4	10	30	1000	10	30	“”	True
400	**Stacked bipartite**	2	4	10	30	1000	10	30	“”	True
400	**Random**	2	4	10	30	1000	10	30	“”	True

9. Bibliography

Alonso Lopez-Rojas, E. (2016). Applying simulation to the problem of detecting financial fraud. *Blekinge Institute of Technology, department of computer science and engineering*, 151-173.

Alonso Lopez-Rojas, E., & Axelsson, S. (2012). Money Laundering Detection using Synthetic Data. *linkoping university electronic press*, 33-40.

Alonso Lopez-Rojas, E., & Axelsson, S. (2016). Paysim: A financial mobile money simulator for fraud detection. *Dime university of Genoa, Euroean modleing and simulation sypmosium* , 249-255.

Alvarez-Jareno, J. A., Badal-Valero, E., & Pavia, J. M. (2017). Using machine learning for financial fraud detection in the accounts of companies investigated for money laundering. *Economics department Universitat Jaume*, 3-21.

Anti-Money Laundering. (2004). *International Federation of Accountants*.

Banirostam, T., & Safari, N. (2018). Detection of the Suspicious Transactions by Integrating the Neural Network and Bat Algorithm. *Specialty Journal of Electronic and Computer Sciences*, 9-19.

Barraclough, P., Hosssain, M., Tahir, M., Sexton, G., & Aslam, N. (2013). Intelligent phishing detection and protection scheme for online transactions. *Expert Systems with Applications*, 4697-4706.

Barse, E. L., Kvarnstrom, H., & Jonsson, E. (2003). Synthesizing Test Data for Fraud Detection Systems. *Annual Computer Security Applications Conference*, 384-394.

Breiman, L. (2001). Random forests. *Machine learning*, 5-32.

Cao, D. K., & Do, P. (2012). Applying data mining in money laundering detection for the Vietnamese banking industry. *Intelligent Information and Database Systems*, 207-2016.

Choo, K.-K. R. (2015). Cryptocurrency and Virtual Currency: Corruption and Money Laundering/Terrorism Financing Risks? *Handbook of Digital Currency*, 283-307.

Colladon, A. F., & Remondi, E. (2017). Using social network analysis to prevent money laundering. *Expert systems with Applications*, 49-58.

De Nederlandsche Bank. (2017). Post_event transactie-monitoringproces bij banken. *Eurosysteem*, 1-48.

Drezewski, R., Filipkowski, W., & Sepielak, J. (2012). System supporting money laundering detection. *Digital investigation*, 1-35.

Dunne, R. A., & Campbell, N. A. (1997). On the Pairing of the Softmax Activation and Cross-Entropy Penalty Functions and the Derivation of the Softmax Activation Function. *Conference on Neural Networks Melbourne*, 181-185.

Financial Intelligence Unit Belize. (n.d.). *Types of Suspicious Activities or Transactions*. Retrieved from fiubelize: http://fiubelize.org/types-of-suspicious-activities-or-transactions/

Flaunet, M. (2019). *Banking industry challenges*. Retrieved from Deloitte: https://www2.deloitte.com/lu/en/pages/banking-and-securities/articles/banking-industry-challenges.html

Gao, C. S., & Xu, D. (2006). Conceptual modelling and development of an intelligent agent-assisted decision support system for anti-money laundering. *11th annual conference of Asia pacific decision science institute Hong Kong*, 442-450.

Heaton, J. (2015). *Volume 3: Deep Learning and Neural Networks.* Heaton Research inc.

IMF. (2020, 04 01). *Anti-Money Laundering/Combating the Financing of Terrorism*. Retrieved from International Monetary Fund: https://www.imf.org/external/np/leg/amlcft/eng/aml1.htm

ING. (2020, Februari 17). Detection of money laundering transactions. (F. Visser, Interviewer)

Jeni, L., Cohn, J., & De La Torre, F. (2013). Facing imbalanced data recommendations for the use of perfomance metrics. *Humaine association conference on affective computing and intelligent interaction*, 245-251.

Keyan, L., & Tingting, Y. (2011). An Improved Support-Vector Network Model for Anti-Money Laundering. *Fifth International Conference on Management of E-Commerce and E-Government*, 193-196.

Khan, N. S., Larik, A. S., Rajput, Q., & Haider, S. (2013). A Bayesian approach for suspicious financial activity reporting. *International Journal of Computers and Applications*, 181-187.

Khanuja, H. K., & Adane, D. S. (2014). Forensic Analysis for Monitoring Database Transactions. *Security in Computing and Communications*, 201-210.

Kharote, M. (2014). Data mining model for money laundering detection in financial domain. *International journal of computer applications*, 61-64.

Le Khac, N. A., Markos, S., & Kechadi, M.-T. (2010). A data mining-based solution for detecting suspicious money laundering cases in an investment bank. *Second International COnference on Advances in Databases, Knowledge, and Data Applications*, 235-240.

Levi, M., & Reuter, P. (2014). Money Laundering. *Chicago Journals*, 289-375.

Liu, R., Qian, X.-l., Mao, S., & Zhu, S.-z. (2011). Research on anti-money laundering based on core decision tree algorithm. *Chinese Control and Decision Conference*, 4322-4325.

Lopez-Rojas, E. A., & Axelsson, S. (2012). Money Laundering Detection using Synthetic Data. *27th annual workshop of the Swedish Artificial Intelligence Society*, 33-40.

Lv, L.-T., Ji, N., & Zhang, J.-L. (2008). A RBF Neural Network Model for Anti-Money Laundering. *Wavelet Analysis and Pattern Recognition*, 209-215.

Madinger, J., & Kinnison, N. (2011). *Money Laundering A guide for Criminal Investigators.* Boca Raton: CRC Press.

Nimmo, M. (2007). Fraud and money laundering. *Topic Gateway Series*.

Omar, N., Amirah Johari, Z., & Arshad, R. (2014). Money laundering - FATF special recommendation VIII: a review of evaluation reports. *Social and Behavioral Sciences*, 211-225.

Rajput, Q., Sadaf Khan, N., & Larik, A. H. (2014). Ontology Based Expert-System for Suspicious Transactions Detection. *Computer and Information Science*, 103-114.

Reuter, P. (2005). *Chasing Dirty Money: The Fight Against Money Laundering.* Peterson Institute.

Rooijers, E., & Leupen, J. (2020, januari 3). *financieel dagblad*. Retrieved from fd.nl: https://fd.nl/ondernemen/1327786/banken-betalen-vorstelijk-om-honderden-sherlocks-te-vinden-in-een-leeggeviste-markt

Sahin, Y., & Duman, E. (2011). Detecting Credit Card Fraud by Decision Trees and Support Vector Machines. *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 978-988.

Salehi, A., Ghazanfari, M., & Fathian, M. (2017). Data Mining Techniques for Anti Money Laundering. *International Journal of Applied Engineering Research ISSN* , 10084-10094.

Strandberg, K. W. (1997). Money Laundering. *Law Enforcement Technology*, 28-33.

Tang, J., & Yin, J. (2005). Developing an intelligent data discriminating system of anti-money laundering based on SVM. *Machine Learning and Cybernetics*, 3453-3457.

Timotius, I. K., & Miaou, S. G. (2010). Arithmetic means of accuracies: A classifier performance measurement for imbalanced data set. *International conference on Audio, Language and Image processing*, 1244-1251.

Unger, B. (2013). Money laundering regulation: from Al Capone to Al Qaeda. *Research handbook on Money Laundering*, 19-32.

Van Vlasselaer, V., Bravo, C., Caelen, O., Eliassi-Rad, T., Akoglu, L., Snoeck, M., & Baesens, B. (2015). APATE: A Novel Approach for Automated Credit Card Transaction Fraud Detection using Network-Based Extensions. *Decision Support Systems*, 1-31.

Wang, S.-N., & Yang, J.-G. (2007). A Money Laundering Risk Evaluation Method Based on Decision Tree. *International Conference on Machine Learning and Cybernetics*, 283-286.

Wang, X., & Guang, D. (2009). Research on money laundering detection based on improved minimum spanning tree clustering and its application. *2009 Second International Sypmosium on Knowledge Acquisition and Modeling*, 62-64.

Watkins, R., Reynolds, K., DeMara, R., Georgiopoulos, M., Gonzalez, A., & Eaglin, R. (2003). Tracking Dirty Proceeds: Exploring Data Mining Technologies As Tools To

Investigate Money Laundering. *Journal of Policing Practise and Research: An international Journal*, 163-178.

Weber, M., Kanezashi, H., Chen, J., Suzumura, T., Pareja, A., Ma, T., . . . B. Schardl, T. (2018). Scalable Graph Learning for Anti-Money Laundering: A First Look. *arXiv preprint arXiv:1812.00076*.

Wiessing, E. (2019, November 27). *Banken willen doorpakken met gezamenlijke aanpak witwassen.* Retrieved from NOS: https://nos.nl/artikel/2312329-banken-willen-doorpakken-met-gezamenlijke-aanpak-witwassen.html

Xingrong, L. (2014). Suspicious Transaction Detection for Anti-Money Laundering. *International Journal of Security and its Applications*, 157-166.

Zengan, G. (2009). Application of cluster-based local outlier factor algorithm in anti-money laundering. *2009 International Conference on Management and Service Science*, 1-4.

Zhu, T. (2006). An Outlier Detectino Model Based on Cross Datasets Comparison for Financial Surveillance. *2006 IEEE Asia-Pacific Conference on Services Computing (APSCC'06)*, 601-604.