Exploring the possibilities of applying an unsupervised machine learning model to detect VAT/carousel fraud

Francine Verbeek

A thesis presented for the degree of
Business Analytics

Exploring the possibilities of applying an unsupervised machine learning model to detect VAT/carousel fraud

Francine Verbeek

VU supervisor: Mark Hoogendorn
VU second reader: Dennis Dobler
Deloitte supervisor: Evert Haasdijk

Vrije Universiteit Amsterdam
Faculty of Science
Business Analytics
De Boelelaan 1111
1081 HV Amsterdam

Deloitte
Gustav Mahlerlaan 2970
1081 LA Amsterdam

April 2022

Preface

For the final phase of the Master Business Analytics, students must execute a graduation project. This graduation project is in the form of a six month internship at a company. During this internship the student will write a report that describes the research question, motivation for the research, literature, used methodology, results, conclusions and recommendations. This report, next to an oral presentation, will prove that the student is capable of independent research at an academic level on a specific topic in the field of Business Analytics. The student will be supervised by one of the staff members of the Faculty of Science during the internship.

My internship was conducted at the 'Forensic and Financial Crime' department of Deloitte, in collaboration with a bank.

My research was conducted in the financial area of VAT/carousel fraud and during my internship I developed an outlier detection model for this type of fraud. Here I investigated whether using a multivariate analysis is an improvement over a univariate analyses as input for the Variational Autoencoder (VAE), where the latent space will be used for outlier detection. Next to this I investigated whether a method exists which could improve the interpretability and explainability of the latent space of a VAE.

First of all, I want to thank all my supervisors. The first one being my main supervisor from the VU, Mark Hoogendorn, for guiding me through this whole process. The biweekly meetings to discuss obstacles I was encountering and my progress were of great help. I also want to thank you for the quick feedback and always making sure I had a clear vision of which direction I was going. I could not have done it without your help. Furthermore, I want to thank my second reader from the VU Denis Dobler for taking the time to read my report. Second, I want to thank my supervisor from Deloitte Evert Haasdijk for the guidance throughout my internship. Whenever I had a problem, I could always schedule a meeting to discuss the problem and quickly go on with my research. I also want to thank you for always asking critical question about my work and by doing so ensuring that I was making correct scientific decisions. Additionally, I want to thank Evert for introducing me at Deloitte and making me feel at home at the department. Furthermore, I want to thank my supervisor from the bank. She was always there when I needed help with understanding the data and with the technical requirements for the bank. She really took her time to explain everything to me. Additionally, I want to thank her for the weekly meetings to discuss my findings and the scientific problems I was encountering. Lastly, special thanks to my Deloitte buddy Frederic Chamot for helping me through the first few months and always being there to ask question regarding Deloitte or to discuss my research.

I also want to thank Deloitte 'Forensic and Financial Crime' department for giving me the opportunity to conduct my internship at their department. During my internship I learned a lot of things from the people at the department. Not just on the technical side but also on the consulting and soft skills side. I want to thank to whole department for that. In addition, I want to thank all the people from the bank for welcoming me into the team and helping me to better understand the financial part of my research. I could not have done it without everyone's help and support.

Lastly, I want to thank my family, boyfriend and friends for supporting me throughout the past six months. Sometimes it was a bit stressful and you were always there to support and help me. I could not have done it without your support and belief in me.

For everyone reading this report, thank you for taking the time to do so and I hope you enjoy it.

Abstract

Millions of euros are lost yearly due to Value-Added Tax (VAT)/carousel fraud in Europe. This type of fraud can be classified as a form of tax evasion. Nowadays, financial institutions use rule-based systems to detect VAT/carousel fraud. A large disadvantage of rule-based systems is its inability to discover new types of VAT/carousel fraud since it relies heavily on past experiences. Currently, a shift has taken place towards artificial intelligence and machine learning systems to detect fraud. There is a growing need for unsupervised machine learning methods to detect fraud, since these require no labeled data. The goal of this project was to explore the possibilities of applying an unsupervised machine learning method to detect VAT/carousel fraud. The project tried to answer the following two research questions: *(1) Is using multivariate analyses for an outlier detection in the latent space of a VAE an improvement compared to univariate analyses?; (2) How can implementing the β-VAE improve the interpretability and explainability of the underlying representation of the VAE?*

For the first research question a total of five features were developed and implemented. These were balance over time, Rapid Movement of Funds (RMoF), cross-border, counterparties, and cash. RMoF occurs when large sums of money are flowing in and out of a bank account in a very short period of time, mostly within a few hours. This phenomenon is an indicator for all sorts of fraud. The implemented RMoF obtained an accuracy of above 80%, which is extremely high. The features were put together into different combinations, such that a total of six VAE models were investigated. The three feature VAE model, consisting of balance over time, RMoF and cross-border, obtained the highest mean recall score. So, it was concluded that using a multivariate analyses for outlier detection in the latent space of a VAE is indeed an improvement compared to univariate analyses.

The second research question tries to improve the interpretability and explainability by implementing the β-VAE. This model demonstrated a higher mean recall score compared to the three feature VAE model. Also, this investigation indicated that the model more clearly disentangles the different features. However, further research into the β-VAE is needed to explore all the possibilities.

This project contributed to filling the research gap on applying unsupervised machine learning methods for the detection of VAT/carousel fraud. The results from this project are promising, but further research into the VAE, β-VAE, the features, and other unsupervised machine learning methods is needed.

Contents

1 Introduction

The Dutch government loses 9 million euro over a span of three years [38, 114]. The Spanish government loses 26.5 million euro in the past year and a half [32, 78]. These huge amounts of money are lost due to an act of financial crime. Many people do not realize that an act of financial crime can have such a big negative impact on governments. A possible reason for this is that financial crime is not a widely known concept. Additionally, newspapers and other news channels do not publish this crime very prominently. However, the number above indicate that governments lose enormous sums of money due to financial crime. In the examples above only one particular type of financial crime caused the losses; namely Value-Added Tax (VAT)/carousel fraud. Currently, Europol estimates that around 60 billion euro is lost yearly due to this type of fraud [34]. VAT fraud can be classified as a form of tax evasion which avoids the payment of VAT money to the government, resulting in a profit for the fraudster [13, 27]. Carousel fraud is a more complicated form of VAT fraud, where the money can cycle continuously.

It is very important to combat this type of fraud, since these loses are such a substantial amount of governments' budget. These governments could use the VAT obtained money to reinvest in society. As a consequence of the increasing fraud, the quality of public services could decrease, which is undesirable. Another reason why this type of fraud should be combatted is that criminal activities and even terrorist attacks can be funded using money obtained from VAT/carousel fraud [13, 27].

To combat VAT/carousel fraud, detection systems are implemented at financial institutions, since the fraud occurs there. Nowadays, financial institutions use rule-based systems to detect VAT/carousel fraud. Rule-based systems often implement a set of *if-then* rules to detect VAT/-carousel fraud. This set of rules is based on historical fraud cases and personal experiences [26]. A large disadvantage of rule-based systems is its inability to discover new types of VAT/carousel fraud since it relies heavily on past experiences. Furthermore, building, maintaining and updating these rule-based systems is expensive and extremely time consuming.

Currently, a shift has taken place towards artificial intelligence and machine learning systems to detect fraud [22]. Machine learning techniques can be divided into two categories: supervised and unsupervised [51]. Supervised techniques build detection systems based on labeled data, while unsupervised techniques are more flexible and do not require labeled data to build detection systems [58]. A problem with supervised techniques is that labeled data is required and often the data needs to be manually labeled by domain experts. Therefore, it is not feasible to label data when the data set is too large. Next to this, errors can be made and as a result the model could be training on incorrect data. Often financial institutions do not have labeled data available. This is why there is a growing need for unsupervised methods to develop a VAT/carousel fraud detection system.

Not many research papers can be found in the area of unsupervised machine learning for VAT/carousel fraud detection. To the best of our knowledge there is only one paper that uses an unsupervised method for VAT/carousel fraud. In this paper the nearest neighbour algorithm is implemented [120]. When looking at the area of financial crime detection, much more research has been conducted. Multiple research papers have stated that unsupervised machine learning techniques are very useful in this area [4, 18, 26, 109]. Two papers about the different machine learning techniques to detect financial fraud state that techniques such as logistic models, neural networks, Bayesian belief network, and decision trees have been applied the most to detect and classify financial fraudulent behavior in data [4, 109].

The most prominent techniques in the area of unsupervised machine learning are generative adversarial networks, convolutional neural networks and autoencoders [53]. These techniques can learn deeper and more complex representations of the data. They can also detect fraud cases previously unknown to detection systems [53]. Among these unsupervised techniques, the variational autoencoder (VAE) seems preferred [77, 128]. The data in this project is transactions

over time, so it can be seen as time series data. The variational autoencoder can handle time series data but it can work more efficiently with time series data if the dimensions would be reduced [74, 128]. A possible solution would be to incorporate long-short term memory layers into the variational autoencoder [75, 87, 97, 128]. Different studies have shown that this combination outperforms other commonly applied detection methods [75, 87, 97, 124, 128].

A remaining problem for deep learning models is the lack of interpretability and explainability [53, 109]. As a result these models are rarely utilized by businesses. A possible solution for this problem would be to implement a β-variational autoencoder (β-VAE). The extra hyperparameter encourages a more disambiguated representation of the results [17, 52].

This project is executed in cooperation with a financial client of Deloitte, namely a bank. The project will use real financial data regarding transactions to develop a detection system for VAT/carousel fraud. The goal is to use an unsupervised machine learning method to find abnormal behavior in accounts which in turn could point to fraudulent behavior. This project can be defined into the following two research questions:

1. Is using multivariate analyses for an outlier detection in the latent space of a VAE an improvement compared to univariate analyses?;
2. How can implementing the β-VAE improve the interpretability and explainability of the underlying representation of the VAE?

Multivariate analysis means that more than one feature will be used as input for the model. A total of six feature combinations will be explored. The explainability of the β-VAE will be measured by investigating if the latent space shows a more disentangled representation of the outlier results. This project will contribute to filling the research gap on unsupervised machine learning methods for the detection of VAT/carousel fraud.

The structure of this report is as follows. This report will first dive deeper into financial crime and its three most common types. Secondly, the different machine learning terms and techniques are explained. Thirdly, an extensive list of literature is presented. Then, the data analysis to gain important insights into the data. Next, the developed features and the methods of the VAE and β-VAE are explained. From this follows the experimental setup and the training of the models. Then the results are presented. Lastly, the discussion and conclusion will be presented where the results are displayed, a reflection and some recommendations for further research are made.

2 Financial Crime

In this second chapter an explanation about financial crime will be given, because most people do not yet know what exactly is meant with this term, or which crimes are classified as financial crime. For this project it is important that the readers have a basic understanding of financial crime. Next to the explanation about financial crime itself, the three most common types of financial crime are explained, since VAT/carousel fraud is part of one of these types.

The term financial crime is nowadays more frequently used, but there is no internationally accepted definition [23, 101]. In the literature financial crime is also referred to as white-collar crime. Below are some examples which show different financial crime definitions used in practice:

- The Federal Bureau of Justice Statistics (Dictionary of Criminal Justice Data Terminology) defines financial crime as: "Nonviolent crime for financial gain committed by means of deception by persons whose occupational status is entrepreneurial, professional or semi-professional and utilizing their special occupational skills and opportunities; also nonviolent crime for financial gain utilizing deception and committed by anyone having special technical and professional knowledge of business and government, irrespective of the person's occupation" [94].
- The International Monetary Fund (IMF) interprets financial crime more in a broad sense, as any non-violent crime resulting in a financial loss [101].
- Within the UK, the Financial Services and Markets Act 2000 (FSMA 2000) states that financial crime includes any offence involving fraud or dishonesty; misconduct in, or misuse of information relating to, a financial market; or handling the proceeds of crime [40].
- Another definition defines financial crime as an crime that is specifically committed against property or money, where an individual or criminal organization takes something belonging to someone else for their own personal benefit [70].

All the different financial crime definitions are in the core similar but focus on slightly different aspects. The following definition of financial crime will be used in this project:

Financial crime is defined as any nonviolent crime that involves fraudulent or dishonest behavior for the purpose of financial gain, targeting individuals, companies and/or the public sector.

The three most common types of financial crimes are: money laundering, terrorist financing and fraud. These three types will be explained more in depth later in this section. But these are not the only types of financial crime. A great variety of criminal activities can be classified as financial crime. Examples are: bribery and corruption, embezzlement, forgery, counterfeiting, identity theft, information security, market abuse and insider dealing, electronic crime, human trafficking and drug trafficking [23, 24, 60, 70, 101].

As was stated in the introduction the financial crime rates have increased significantly over the last few years. That is why combating financial crime has been high on the agenda for governments worldwide. The reason for governments to combat financial crime is that this crime has the power to corrupt and destabilize communities or whole national economies [35, 60]. Next to this, there is also the threat to national security since terrorists use the money obtained from financial crime to carry out their illegal activities [101].

Financial crime occurs at financial institutions, because at this place the money moves. Detecting this crime is a significant ongoing challenge for financial institutions, due to the often complex nature of financial services [60]. To better understand this, one can think of it as a cat-and-mouse game between the financial institutions and the criminals. The institutions are constantly trying to detect and prevent financial crime using the newest strategies. At the same time, the criminals

are developing innovative tactics and more sophisticated methods to prevent detection while still committing the crime. [24, 70]

As stated previously, the three most common types of financial crime are: money laundering, terrorist financing and fraud. In the next sections an outline of these types is presented, starting with the biggest one: money laundering.

2.1 Money Laundering

Money laundering is the process of making money generated by criminal activity appear to have come from a legitimate source [19, 37]. The process of money laundering involves three phases: placement, layering and integration. All three phases are explained more in depth below and are also presented in Figure 1. [16, 19, 37, 62]

- Placement
  The first phase is bringing the 'dirty money' (often cash) into the financial system. The criminals want to deposit the large sums of money derived from the criminal activities into the legitimate financial system. Money laundering is most vulnerable at this phase, because the crime is most apparent and thus at high risk of detection. The money launderer has to find a solution to move around the large amounts of money into a more manageable form to introduce it into the financial system. There are different techniques for this. Think of: breaking large amounts of money into less suspicious sums that are then deposited into a bank account, or purchasing a series of monetary instruments (cheques, etc.) which are collected and deposited into accounts at another location. After this has happened the next phase of the cycle starts, namely layering.

- Layering
  So now the 'dirty money' has entered the financial system. The money launderer tries to conceal or disguise the source or ownership of the funds by creating a web of financial transactions that in terms of the frequency, complexity and volume often resemble legitimate financial activity. The purpose of this is to distance themselves as much as possible from the source or ownership. Due to the web of parallel and serial transactions it is becoming increasingly difficult to reconstruct the trail of a money launderer. Examples of this process are: the launderer wires the funds through a series of accounts at various banks across the globe, or the launderer disguises the transfers as payments for goods or services. In an attempt to hide its true origin, the funds are often wire transferred into financial or banking systems through offshore accounts.

- Integration
  The final phase of the money laundering process involves reintegrating the 'cleaned money' within formal sectors of the economy. This 'cleaned money' is now in a legitimate account and can be used for whatever purposes the criminals have in mind. For example the criminal might choose to invest the funds into real estate or luxury assets. At this point is it almost impossible to distinguish between legal and illegal money at bank accounts.

The concept of money laundering is essential for criminal organizations that want to use illegally obtained money effectively. Money launders tend to seek out countries or sectors within countries where there is a low risk of detection or where there are not effective anti-money laundering programs [37].

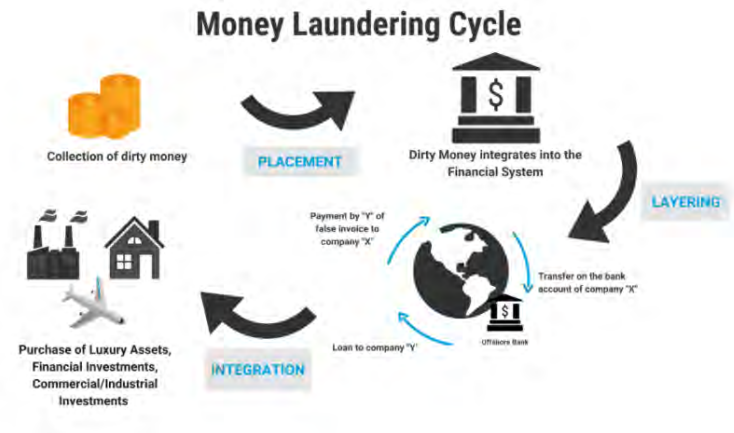

Figure 1: Money Laundering Cycle [118]

To determine the exact scale of how much money is lost due to money laundering is difficult. The reason for this is that it depends on multiple variables of which many are unknown to us. The United Nations Office on Drugs and Crime (UNODC) estimates that between 2% and 5% of the global GDP is laundered each year. That is between 175 billion and 1.87 trillion euro each year. [33] Although this is an estimated number, it indicates that large sums of money are laundered each year. Also have a look at what the estimated number of money laundering activities are in the Netherlands. In 2010 there were 335 registered money laundering cases and in 2021 this number went up to 2460 registered money laundering cases [112]. A reason for this growth could be that money laundering activities have increased in recent years, which most likely happened. But another reason for this increase of cases could be that an attention shift has taken place so there is a lot more focus on detecting and combating money laundering activities in contrast with a few years ago.

2.2 Terrorist Financing

The second big type of financial crime is terrorist financing. Terrorist organizations need financial support to achieve their goals. Therefore, a successful terrorist organization is one that can build and maintain an effective financial infrastructure [60]. The concept of terrorist financing is rather straightforward. It means the act of providing or collecting funds with the intention that these funds should be used or have the knowledge that these funds will be used to carry out terrorist acts [23, 24, 71]. This includes raising, moving, storing and using financial resources to support terrorism [23]. Terrorist financing can be split up into three stages: [107]

- Raising funds
  The first stage is about how funds are raised to support terrorist financing. These funds can be raised via different types of activities. For example, via legitimate sources, such as profit of legitimate businesses, donations, government funding and religious or cultural organizations. Another option would be that the funds originate from illegal sources, such as drug trafficking, the smuggling of weapons or other goods, kidnapping, extortion and government corruption. [45, 66]

- Transferring funds
  In the second stage it is about how the funds are transferred to a terrorist organization. The terrorists use the same techniques as in money laundering to move the funds around and try to not catch the attention of the authorities and protect the identity of their sponsors of the funds. It resembles money laundering in the sense that terrorists want to conceal the transfer of funds within the legitimate financial system. As is the same with money

laundering, if the terrorist funds enter the legal financial system it becomes harder to detect and track these funds. [24, 45, 60, 66]

- Using funds
  During the final stage, the funds can be used for direct and/or indirect support of terrorist activities carried out by the terrorist organization. The money is among other things used to buy weapons, travel, train the terrorists and for future development of the organization.

Money laundering and terrorist financing are often linked. So when authorities are able to detect and prevent money laundering activities, it will most likely also prevent the funds from being used to finance acts of terrorism [66]. Disrupting and preventing the terrorism-related financial flows and transaction is one of the most effective ways to fight terrorism [36], and this detection and prevention can be done using anti-money laundering systems.

2.3 Fraud

Fraud is next to money laundering and terrorist financing also a significant part of financial crime activities. Fraud occurs when a person intentionally acts dishonest to illegally deprive another person or entity of money, property, or legal rights, thus the person gains something unlawfully or unfairly [21, 72, 117]. The following three points must be met to classify a crime as fraud: [72, 82]

1. Using dishonest methods to gain something personally or financially (so take something valuable from another person)
2. A person who pretends to be something or someone he/she is not
3. Intentional deception to persuade another person to part with something of value. This can be done for example by a copy of something to trick people

The criminal that is committing fraud is often aware of information that the intended victim is not, allowing the criminal to deceive the victim. At heart, the individual or company committing fraud is taking advantage of information asymmetry [21]. Fraud often results in some kind of loss for the victim. This could be monetary, non-monetary assets or damage to the reputation (fraud can have devastating impact on a business [21]). Technically, anyone can fall prey to fraudulent crimes, so the impact of fraud ranges from individual level to large organizations [119]. Some well-known examples of fraud are: tax fraud, credit card fraud, identity theft, health care fraud and health insurance fraud, securities fraud, telemarketing fraud, and mail fraud [21, 25, 72, 117, 119].

This project will focus on a specific type of fraud, namely Value-Added Tax (VAT)/carousel fraud. In the next section a more in depth explanation will be given on VAT/carousel fraud.

2.3.1 VAT/carousel fraud

To get a understanding of Value-Added Tax (VAT)/carousel fraud, one first needs to understand VAT and what the rules and regulations are for VAT. The VAT is a consumption tax and is based on the value added to goods and services, which happens at every stage of the production and distribution [31, 61]. This type of tax is used worldwide, but for this project the decision was made to focus on countries within the European Union (EU). Each country in the EU is responsible for setting its own VAT rates, but must comply with the EU law which states that the standard VAT rate must be at least 15% and the reduced rate must be at least 5% [31, 125]. For the Netherlands the standard VAT rate is 21% and the reduced rate is 9% [7].

VAT fraud can be classified as a form of tax evasion which avoids the payment of VAT money to the government, resulting in a profit for the fraudster [13, 27]. It is a very common type of fraud, since companies simply charge the VAT when selling a good or service to consumers, but

keep the VAT amount to themselves instead of paying this amount to the tax authority. This type of fraud occurs at financial institutions and then mostly banks. VAT fraud can occur through different channels. Examples of this are: [46, 68]

- Under-reported sales (also called 'off the book sales')
  A business only reports a proportion of the sales, is falsifying records, or makes some sales completely off the books.
- No firm registration to tax authorities (ghost firms)
  When a business fails to register, they are both saving the VAT for which they are liable and also VAT compliance costs. This happens mostly at relatively small businesses.
- Misclassification of commodities
  This happens when businesses sell several goods taxed at different VAT rates. They may reduce their liability by falsifying the proportion of sales in lower-taxed categories.
- Tax collected but not remitted
  Businesses charge their consumers VAT and collect this, but disappear before paying the VAT money to the authorities (called missing trader).
- Imported goods which are not brought into tax
  When no tax needs to be paid at the border, there is a profit from purchasing imported goods (or services) having no tax and then reselling them in the home market. This type arises under the current EU payment system and is a major weakness in the system.

Carousel fraud is a more complicated form of VAT fraud. It involves a chain of events, starting with purchasing a good from a company in another EU Member State. When importing a good from another EU Member State it is VAT free. This is an agreement the EU made for its member states. The good is then sold through different companies, each company charging VAT, but this VAT is not paid to the tax authorities. Eventually the good is exported across border, often back to the original seller. When exporting across border to another EU Member State, again no VAT needs to be charged. This is a continuously cycle, hence the name carousel. [98, 111] The profit for the fraudsters is that they steal VAT at each turn from the tax authority [13]. It must be noted that it is possible that not every company in the cycle is committing fraud, or at least not intentionally.

Carousel fraud can involve any type of good or service, but the goods often have in common that they are readily available in large quantities, have a high value and low weight, and become rapidly technically out-of-date [13]. In an extreme case it is possible that an empty box is moved around the different companies. It can be stated that VAT/carousel fraud is a complicated and highly sophisticated process which is very difficult to detect [111]. Below a simplified example is presented to get a better feeling of how carousel fraud works. This example is also visually presented in Figure 2.

Example carousel fraud

Business A (resides outside the Netherlands but within the EU, say for example France) delivers a good to business B in the Netherlands. Since the good is sold across EU borders the zero rate policy applies and no VAT needs to be paid. Business B sells this imported good to business C and adds 21% VAT to the good, which is allowed. The obtained money from the 21% VAT should be handed over to the government in the Netherlands by B, only he does not do this and commits fraud. Business C pays the 21% VAT to B and thus can ask this VAT amount back from the government, C then sells the good to business D with 21% VAT. Then business D sells the good to business A again, since this is outside the Netherlands there is again no VAT. This circle is repeated many times to make as much profit as possible. The profit for the fraudsters is in the concept that the businesses charge VAT but do not pay it to the government. This implies that the transactions between business is about large sums of money, often in the thousands or millions of euros, because on small transactions there is almost no profit. [8, 38]

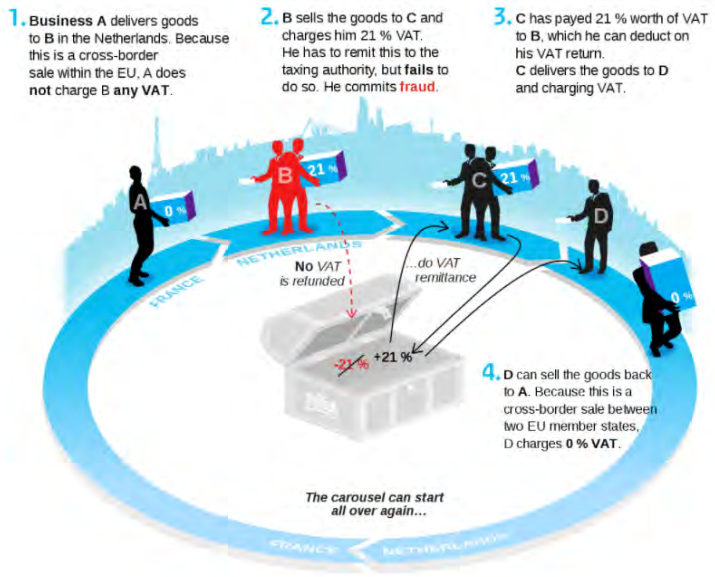

Figure 2: Example of carousel fraud [114]

The last few years there has been an increasing concern regarding the money lost due to VAT/-carousel fraud within the EU [13, 46, 68]. The most likely reason for this increase in VAT/carousel fraud in the EU is the use of the Single Market. In 1993 the EU wanted to simulate the trade between EU Member States and therefore introduced the concept of the Single Market. The Single Market ensures that people, goods, services and money can move around freely within the EU [29]. With the introduction of the Single Market also came the zero rates for VAT. A consequence of these zero rates is that goods and services within the EU can cross borders VAT free, as a result the trade between EU Member States is very vulnerable to VAT/carousel fraud. As can be imagined with a growing number of EU Member States, so will the numbers of VAT/carousel fraud grow, since goods and services can be imported to more countries VAT free.

The reason why VAT/carousel fraud is a big problem for EU Member States is that the money obtained trough VAT is one of the most important sources of government revenue [46]. As already stated in the introduction, the government revenue is reinvested into society and used for example for health care, public transport and education. When VAT/carousel fraud increases, less money is available to be reinvested into society. Another concern from EU Member States is that the money obtained from VAT/carousel fraud is used to fund criminal activities and terrorist attacks [27, 13]. It can be stated that VAT/carousel fraud is a considerable crime, and can be seen as large-scale theft since it empties the governments public purse.

It is hard to determine the exact scale of how much money is lost due to VAT/carousel fraud. However, Europol estimates that around 60 billion euros in Europa is missed yearly in government revenue due to VAT/carousel fraud [34]. For example, a few years ago it was discovered that multiple companies in Malta were involved in a Europa-wide VAT/carousel fraud ordeal. These companies were robbing around 50 billion euro yearly from European governments. [84] The EU commission on taxation forecasts that due to the coronavirus the numbers of VAT/carousel fraud cases has increased, but this is a forecast and no numbers have been released yet that deny or confirm this prediction [30]. These number indicate that large sums of money are lost due to VAT/carousel fraud.

For the development of such a VAT/carousel fraud detection system, some domain specific challenges must be kept in mind: [120]

- Most of the available data is unlabeled. The reason for this is that obtaining labeled data is extremely time consuming, since tax experts need to manually assess every transaction. A possible consequence of this could be that a human error is made, so the data is wrongly labeled and the model is trained on incorrect data.
- Continuing on the aforementioned point, if there is labeled data available it is mostly only a small proportion of the whole dataset. Thus making the labeled data not very representative for the whole population.
- Fraud is assumed to be a rare phenomenon, so only a small fraction of the data will consist of fraud, making it harder to train models.
- Fraudulent behavior changes over time and this requires that detection systems are regularly updated to keep up with the changing patterns.
- The detection systems need to deal with large datasets, which require fast algorithms for the training part of the systems.

3 Machine learning background

In this chapter the different machine learning components will be explained. As was stated in the introduction, there are two different kinds of machine learning: supervised and unsupervised. The main difference between supervised and unsupervised is the availability of labeled data [58]. Supervised techniques build machine learning models based on labeled data, while unsupervised techniques use unlabeled data. Supervised techniques are nowadays mostly implemented at financial institutions to detect fraud. These methods are not ideal, because not every fraud cases will be found and because labeled data is often not available. Currently, a shift has taken place towards unsupervised techniques for fraud detection [22]. Previous research state that deep learning architectures are the most prominent techniques for outlier detection and the Variational Autoencoder (VAE) is preferred for the detection of VAT/carousel fraud. Since the model needs to work with time series data, Long Short-Term Memory (LSTM) layers are incorporated into the VAE. From the literature, it was obtained that Isolation Forest is the most promising outlier detection technique. Therefore, this technique was applied in this project. A more extensive literature review on the different supervised and unsupervised techniques on fraud detection can be found in the next chapter.

The first section in this chapter explains Neural Networks because this project will implement a deep learning neural network. Then, we will dive deeper into the Autoencoder and its expansion to the Variational Autoencoder. The basis of the Variational Autoencoder is used to explain the β-Variational Autoencoder. The models use Long Short-Term Memory layers, so these will be explained next. Lastly, a description will be given on Isolation Forest, which is the applied outlier detection technique.

3.1 Neural Networks

The creation of neural networks was inspired by the human brain [20, 56, 115] and it uses components that behave like biological neurons [95]. A neural network consists of interconnected elements, which are called neurons. These networks try to recognize patterns and solve problems using these patterns [56]. Each neural network has an input and output layer and some networks also have hidden layers. The artificial neurons in the network can be seen as a mathematical function, where each neuron takes inputs, weighs them separately, sums them up and then passes this sum through a nonlinear activation function to produce an output [20, 110]. These neural networks can be divided into two groups: single-layer and multi-layer networks.

Start with the simplest and oldest neural network, the single-layer neural network, also called perceptron [56]. The single-layer networks consists of only one neuron. The input $\mathbf{x} = (x_1, \ldots, x_m)$ is a vector consisting of m inputs. These inputs are multiplied with the corresponding learned weight coefficients $W = (w_1, \ldots, w_m)$ where W is the weight matrix. Then, the bias term (b) is added to the weighted input.

$$\sum_{i=1}^{m} w_i x_i + b$$

This obtained value is then used as input for the activation function. In the single-layer neural network the step function is used as activation function. The step function gets triggered above a certain neuron output value, otherwise it outputs zero. This can be notated as

$$f(x) = \begin{cases} 1 & \text{if } \sum_{i=1}^{m} w_i x_i + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

A visual representation of the structure of the single-layer neural network can be seen in Figure 3. [56, 85, 110]

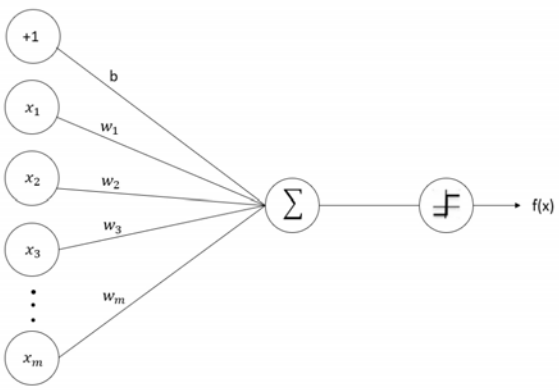

Figure 3: Structure of single-layer neural network

Next, there is the multi-layer neural network which is an expansion of the single-layer neural network, also known as multi-layer perceptron. The first layer in this network is the input layer, the last layer is the output layer, and the layers between are the hidden layers. It is possible that there are many hidden layers in a neural network. Each artificial neuron in a layer is connected to all artificial neurons in the next layer. The connection is again characterized by the corresponding learned weight coefficients. At each layer the weighted sum of inputs is taken, including a bias term. This value is passed on to an activation function. Examples of activation functions are presented in the next part of this section. The outcome of this activation function becomes the input for the next layer. This process is repeated until a final output is created.

For example, when looking at a neural network that has an input layer, one hidden layer, and an output layer, the neural network consists of three layers. The following equations apply:

$$
\begin{aligned}
z^{(2)} &= W^{(1)}\mathbf{x} + b^{(1)} \\
a^{(2)} &= f(z^{(2)}) \\
z^{(3)} &= W^{(2)}a^{(2)} + b^{(2)} \\
a^{(3)} &= f(z^{(3)})
\end{aligned}
$$

The quantities $z^{(i)}$ are known as activations and each of them is transformed using an activation function $f(\cdot)$. The $z^{(i)}$ is the total weighted sum of the inputs of a layer, including the bias term. The $W^{(i)}$ is the weight parameter from the weight matrix associated with each layer and $b^{(i)}$ is the bias term associated with each layer. The $a^{(i)}$ terms denotes the output unit activations, where an activation function has been applied to $z^{(i)}$. This outcome is the input for the next layer. In the aforementioned example, the last equation indicates the output of the neural network. Since there are three layers, the final output is calculated at $a^{(3)}$. In multi-layer neural networks activation functions are often the sigmoid function or the tanh function. These look as follows:

$$
\begin{aligned}
f(z) &= \frac{1}{1 + exp(-z)} \\
f(z) &= tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}
\end{aligned}
$$

The equations of the example can be generalized. Using the fact that $a^{(1)} = x$, which denotes the values from the input layer, it can be concluded that $a^{(l)}$ can compute output unit activation $a^{(l+1)}$ as follows:

$$
\begin{aligned}
z^{(l+1)} &= W^{(l)}a^{(l)} + b^{(l)} \\
a^{(l+1)} &= f(z^{(l+1)})
\end{aligned}
$$

In the above-mentioned example only one hidden layer was used, but it is possible that multiple hidden layers are implemented in the neural network. A visual representation of the structure of the multi-layer neural network, with only one hidden layer, can be seen in Figure 4.

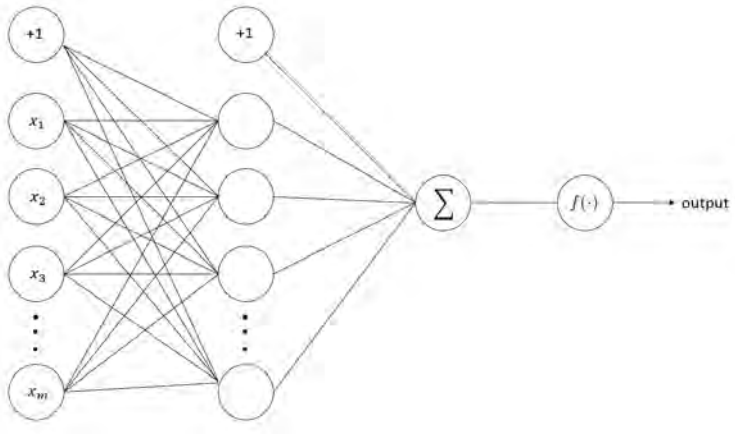

Figure 4: Structure of multi-layer neural network, with one hidden layer

The key difference between the single-layer and multi-layer neural network is that hidden layers are applied in the multi-layer neural network and not in the single-layer neural network. Another possible difference could be that the multi-layer network uses continuous activation function, such as sigmoid and tanh functions, while the single-layer network uses the step function. [12, 20, 56, 85, 115]

A neural network that consists of two or more hidden layers can be considered deep learning [20, 56]. With a higher number of hidden layers the network becomes more complex. Deep learning makes it possible to solve much more complex problems. Therefore deep learning shows to be very useful in the area of unsupervised data [105].

3.2 Autoencoder

An Autoencoder (AE) is a type of deep learning neural network. An AE has three components: encoder, latent space and decoder. The encoder compresses the input data to a lower dimensional space, called the latent space $\mathbf{z}$. The decoder then reconstructs the original data using the compressed version in the latent space. The structure of an AE can be seen in Figure 5. For simplification this figure has only one hidden layer before and after the latent space, but it could be that there are multiple hidden layers.

The input data is passed through a small number of hidden layers, reducing the dimensionality of the input data which forces a compressed representation of the data. This representation is then used to reconstruct the data in the output layer. [14, 47, 48, 59, 64, 80, 121] There are fewer nodes in the hidden layer(s) than in the input and output layers, because we want to ensure that the network is not memorizing the input data [64]. The main idea for the AE is that the network only takes the core information from the input data. This process occurs in the latent space. The latent space is then used to reconstruct the input data.

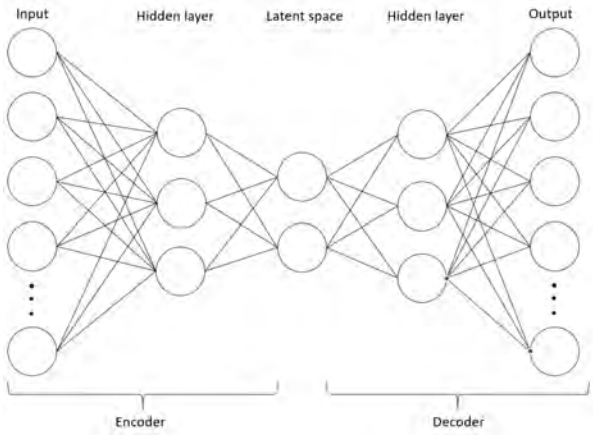

Figure 5: Structure of an autoencoder

The network has an encoder function $g(\cdot)$ that is parameterized by ϕ, also known as $g_\phi(\cdot)$, and a decoder function $f(\cdot)$ that is parameterized by θ, also known as $f_\theta(\cdot)$. The input is $\mathbf{x}$, the compressed version in the latent space of the input is $\mathbf{z} = g_\phi(\mathbf{x})$ and the reconstructed input is then $\mathbf{x}' = f_\theta(g_\phi(\mathbf{x}))$. [80, 121] The difference between the input and the reconstructed input can be measured using the reconstruction loss, $\mathcal{L}(\mathbf{x}, \mathbf{x}')$ [64, 80, 85]. The goal of the AE is to obtain identical values for $\mathbf{x}$ and $\mathbf{x}'$ [47], thus having a reconstruction loss of zero. However, obtaining identical values is often not possible. Therefore, we try to minimize the reconstruction loss. [127] The goal is to try to find the optimal parameters for the encoder and decoder function (ϕ and θ) to achieve this. The reconstruction loss can be calculated using the cross entropy when the activation function is sigmoid or using the Mean Squared Error (MSE): [80, 121]

$$\mathcal{L}(\theta, \phi) = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}^{(i)} - f_\theta(g_\phi(\mathbf{x}^{(i)})))^2$$

3.3 Variational Autoencoder

The previously explained Autoencoders have one fundamental problem. The latent space where the inputs are converted and the encoded vectors are, is often not continuous [108]. So, it is not possible to generate variations of the input. As a consequence, the model generates unrealistic outputs for input variations it has never seen before. A possible solution for this problem would be an encoder which describes a probability distribution for each latent attribute, instead of outputting a single value. This is done in a Variational Autoencoder (VAE). [44, 65] As a result, the latent space is continuous, allowing interpolation and random sampling [65, 108]. This ability differentiates them from the standard AEs. An advantage of the VAE compared to the AE, is that the VAE can better handle cases it has never encountered before. Since fraudulent behavior changes frequently, it is important that the model can still recognize this behavior, even if it has never encountered it before.

Like a standard AE, the VAE consists of an encoder, latent space and a decoder and the goal is to minimize the reconstruction loss [49, 100]. In the encoder the input sample $\mathbf{x}$ is encoded as a distribution over the latent space, instead of a single point [100]. Because of this, an extra step in the network is needed. The encoder outputs the vector of means, $\boldsymbol{\mu}$, and the vector of standard deviations, $\boldsymbol{\sigma}$. These two parameters are used to obtain the sampled encoding, which is passed on to the decoder. Due to sampling, every pass to the decoder will be slightly different, even though the same mean and standard deviation are used. [65, 108] The encodings are generated using the two vectors. The decoder learns that all nearby points in the latent space refer to a sample of that class, instead of a single point. As a result, the decoder learns variations of the input. [108]

The VAE network consists of the following steps: first, the input is encoded as a distribution over the latent space with parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. Second, a sample $\mathbf{z}$ is taken from the latent space using the distribution. Then, the sample $\mathbf{z}$ is decoded and the reconstruction error is computed. Lastly, the reconstruction error is backpropagated through the whole network to determine the optimal values for the parameters of the network. [100] The structure of the VAE is also visible in Figure 6

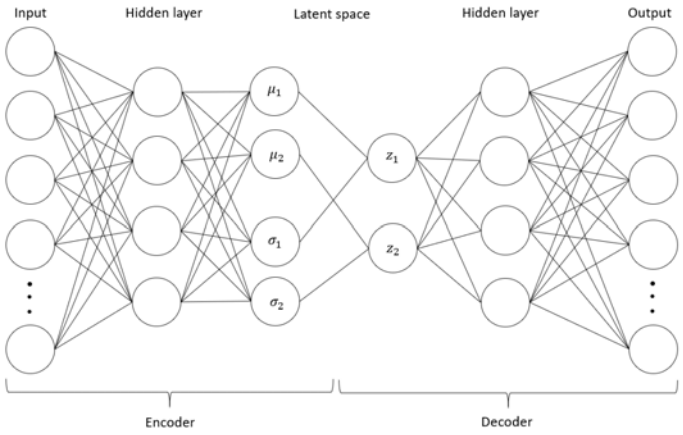

Figure 6: Structure of an Variational Autoencoder

Below, a mathematical formulation of the VAE is given. First, the input needs to be mapped to a distribution, this distribution can be labeled as p_θ, so it is parameterized by θ. The relationship between the input $\mathbf{x}$ and the latent space vector $\mathbf{z}$ can be defined as

$$\begin{aligned} \text{Prior } & p_\theta(\mathbf{z}) \\ \text{Likelihood } & p_\theta(\mathbf{x}|\mathbf{z}) \\ \text{Posterior } & p_\theta(\mathbf{z}|\mathbf{x}) \end{aligned}$$

The prior denotes the probability of observing $\mathbf{z}$. The likelihood is a conditional probability, which calculates the probability of $\mathbf{x}$ given that $\mathbf{z}$ has been observed. The posterior is also a conditional probability, which calculates the probability of $\mathbf{z}$ given that $\mathbf{x}$ has been observed. $\mathbf{z}$ is a hidden variable and with the use of the posterior we can infer the characteristics of $\mathbf{z}$ whenever $\mathbf{x}$ is observable. [44, 65, 88] For the calculation of the posterior, the prior and likelihood are needed. The formula to determine the posterior is as follows

$$p_\theta(\mathbf{z}|\mathbf{x}) = \frac{p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})}{p_\theta(\mathbf{x})}$$
$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

There is a problem with these formulas, which is that the computation of $p_\theta(\mathbf{x})$ is quite difficult and is usually an intractable distribution. These intractabilities can appear in complicated likelihood functions, such as neural networks with nonlinear hidden layers [69]. A solution to this problem would be to approximate the posterior by $q_\phi(\mathbf{z}|\mathbf{x})$, such that it becomes a tractable distribution. Desirably, the approximated posterior $q_\phi(\mathbf{z}|\mathbf{x})$ is close to the real posterior $p_\theta(\mathbf{z}|\mathbf{x})$. Then the parameters of $q_\phi(\mathbf{z}|\mathbf{x})$ are somewhat similar to $p_\theta(\mathbf{z}|\mathbf{x})$, and the approximated posterior can be used for inference of the intractable distribution. To measure the similarity between two probability distributions, the Kullback-Leibler (KL) divergence is used. This can mathematical be formulated as follows, where $p(x)$ and $q(x)$ are two probability distributions

$$D_{KL}(p(x)||q(x))$$

Since we want to ensure that $q_\phi(\mathbf{z}|\mathbf{x})$ is similar to $p_\theta(\mathbf{z}|\mathbf{x})$, the KL divergence needs to be minimized. As a result, the probability distribution parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are optimized. Minimizing the KL divergence of the two posteriors is notated as follows

$$\min D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{x}|\mathbf{z}))$$

The marginal likelihood of the data can be written as

$$logp_\theta(\mathbf{x}) = D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{x}|\mathbf{z})) + \mathcal{L}(\theta, \phi; \mathbf{x})$$

The KL divergence is non-negative. Therefore, the $\mathcal{L}(\theta, \phi; \mathbf{x})$ can be called the (variational) lower bound on the marginal likelihood data and can be rewritten as

$$\begin{aligned} logp_\theta(\mathbf{x}) \geq \mathcal{L}(\theta, \phi; \mathbf{x}) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}(-logq_\phi(\mathbf{z}|\mathbf{x}) + logp_\theta(\mathbf{x}|\mathbf{z})) \\ &= -D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} logp_\theta(\mathbf{x}|\mathbf{z}) \end{aligned}$$

The goal is to maximize $\mathcal{L}(\theta, \phi; \mathbf{x})$ such that the parameters θ and ϕ are optimized given the input $\mathbf{x}$. The first term of the equation is the KL divergence between the approximated posterior and the prior of the latent variable $\mathbf{z}$. This forces the posterior distribution to be similar to the prior distribution. The second term of the equation is the reconstruction likelihood of $\mathbf{x}$ through posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$ and likelihood $p_\theta(\mathbf{x}|\mathbf{z})$. By minimizing the loss, we are maximizing the lower bound of the probability of generating data samples. [5, 44, 49, 65, 69, 108, 121]

The gradients in the loss functions are backpropagated through the network to update the parameters and train the VAE. The expectation term in the loss function requires random sampling of sample $\mathbf{z}$. Since random sampling is a stochastic process, backpropagation for the gradient is not possible and thus training the VAE is not possible. A solution to this problem would be to apply the reparameterization trick in the sampling step of the VAE. The reparameterization trick ensures that the random variable $\mathbf{z}$ can be expressed as a deterministic variable $\mathbf{z}$, instead of a random variable. This is done by taking a random sample $\boldsymbol{\epsilon}$ from a unit Gaussian, $N(0, 1)$. Then the randomly sampled $\boldsymbol{\epsilon}$ is scaled by the latent distribution variance $\boldsymbol{\sigma}$ and shifted by the latent distribution mean $\boldsymbol{\mu}$. Mathematically this is formulated as

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$$

So, the random variable shifts from $\mathbf{z}$ to $\boldsymbol{\epsilon}$ by the reparameterization trick. This is also visible in Figure 7. The $\boldsymbol{\epsilon}$ reparameterizes the VAE network. This allows the $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ to still remain learnable parameters of the network and ensures that these parameters can be optimized when training the VAE. Next to this, the ability to randomly sample is retained for the entire network. [5, 65, 69, 90, 121]

An overview of the reparameterization trick can be seen in Figure 7.

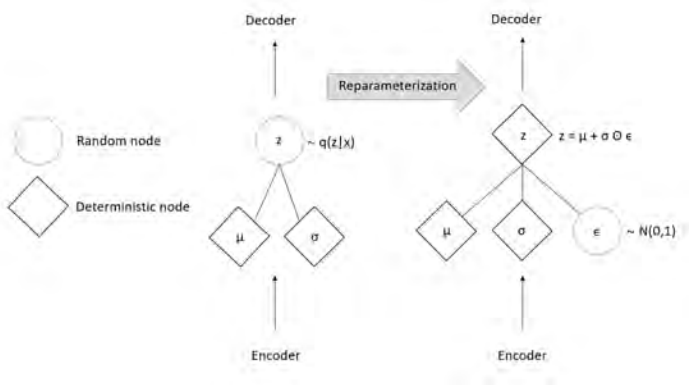

Figure 7: Reparameterization trick

3.4 β-Variational Autoencoder

Learning an interpretable and explainable representation of black box models is something that is strived for. The reason for this is that improvement of interpretability and explainability of these models and results will lead to more human interaction and can help support decision-making [53, 109]. The β-Variational Autoencoder (VAE) is a type of Variational Autoencoder that can disentangle the data in an unsupervised manner. A disentangled representation can be defined as one where single latent units are sensitive to changes of single generative factors, but are relatively invariant to changes of other factors [17, 52]. An advantage of this disentangled representation is that it has good interpretability [17, 121].

For example, a machine learning model is trained on photos of human faces. This model might then capture the skin color, hair color, emotion, whether or not person is wearing glasses, and other relatively independent factors in separate dimensions. These factors are relatively independent and therefore will be visible in the latent space in different dimensions. [121]

The β-VAE applies a modification to the objective in the VAE framework, which was described in the previous section. It adds an extra hyperparameter β to the objective, which encourages the latent representation to be more factorized. The hyperparameter β modulates the learning constraints. These constraints impose a limit on the capacity of the latent information channel and control the emphasis on learning independent latent factors [52]. As was the same in the VAE framework, we want to develop a model that can learn the joint distribution of the data $\mathbf{x}$ and a set of latent factors $\mathbf{z}$, such that $\mathbf{z}$ can generate the observed data $\mathbf{x}$. Again, we want to maximize the marginal likelihood of the observed data $\mathbf{x}$

$$\max_\theta \mathbb{E}_{p_\theta(\mathbf{z})}(p_\theta(\mathbf{x}|\mathbf{z}))$$

The same as with the VAE, when given input sample $\mathbf{x}$, the real posterior $p_\theta(\mathbf{z}|\mathbf{x})$ cannot be calculated since it is an intractable distribution. That is why the posterior is approximated by $q_\phi(\mathbf{z}|\mathbf{x})$. The aim is that the latent factors in $q_\phi(\mathbf{z}|\mathbf{x})$ capture the factors in a disentangled manner. The factors in the latent space would statistically be independent and it would be observable which combination of factors contribute to the result. A constraint is introduced to encourage this disentangling. The constraint tries to match the approximated posterior $q_\phi(\mathbf{z}|\mathbf{x})$ to prior $p(\mathbf{z})$. This controls the capacity of the latent space and ensures that the factors are independent. This can be established if the prior is set to be an isotropic unit Gaussian, $p(\mathbf{z}) \sim N(0, 1)$. The following constrained optimization problem is obtained, where ϵ specifies the strength of the constraint

$$\begin{aligned} &\max_{\phi,\theta} \mathbb{E}_x(\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}(logp_\theta(\mathbf{x}|\mathbf{z}))) \\ &\quad s.t.\ D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) < \epsilon \end{aligned}$$

The equation above can be rewritten as a Lagrangian under the Karush-Kuhn-Tucker (KKT) condition, from which we obtain the following

$$\mathcal{F}(\theta, \phi, \beta; \mathbf{x}, \mathbf{z}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}(logp_\theta(\mathbf{x}|\mathbf{z})) - \beta(D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) - \epsilon)$$

The β is the regularization coefficient which constrains the capacity of the latent space $\mathbf{z}$. It also puts independent pressure on the approximated posterior due to the isotropic nature of the Gaussian prior. Since $\epsilon > 0$, it is possible to rewrite the equation

$$\begin{aligned} \mathcal{F}(\theta, \phi, \beta; \mathbf{x}, \mathbf{z}) &\geq \mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}, \beta) \\ &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}(logp_\theta(\mathbf{x}|\mathbf{z})) - \beta(D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \end{aligned}$$

This results in the same objective as in the VAE framework, only a β is added in front of the KL divergence term. Changing the value of β changes the degree of applied learning pressure during training, which encourages different learning representations. When setting $\beta = 1$, β-VAE

objective becomes equivalent to the VAE objective. With $\beta > 1$ the model is forced to learn a more efficient latent space representation of the data, since a higher β limits the capacity of the latent space $\mathbf{z}$. The above presented β-VAE objective minimizes the β-weighted KL divergence term and maximizes the data log likelihood. This is to encourage that the approximated posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$ efficiently transmits information about data input samples $\mathbf{x}$. This will eventually result in a more disentangled representation. [17, 52, 121]

3.5 LSTM layers

A Recurrent Neural Network (RNN) is a different neural networks which deals well with temporal/time series data [106]. The VAE and β-VAE however, do not have memory and therefore are worse at dealing with temporal data. An RNN uses time series data or sequential data as input [57]. The RNN takes prior information as inputs to influence the current input and output. In contrast, the VAE and β-VAE have inputs and outputs which are independent of each other. Another difference is that RNN share weight parameters within each layer of the network. [57, 41] As a result, RNNs can handle flexible input lengths.

In Figure 8 the structure of an RNN can be seen. The 'rolled' part in the figure represents the whole neural network of the RNN. The 'unrolled' part in the figure represents every individual layer of the neural network. [57] For simplicity, in the example the 'unrolled' part of the RNN only has three layers. In practice this are many layers.

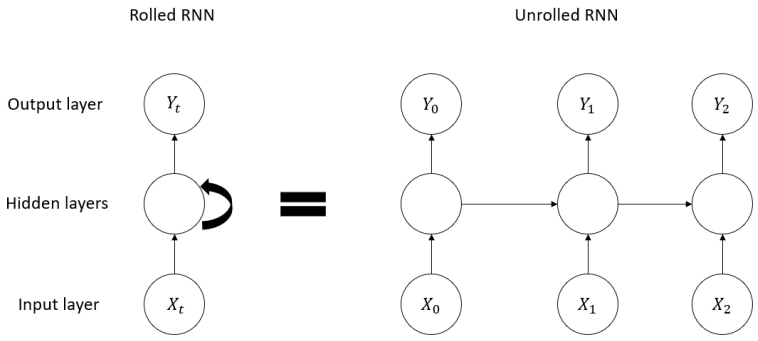

Figure 8: Structure of an RNN

An RNN takes a sequence of data vectors $\mathbf{x} = (x_1, \ldots, x_T)$ as input. This sequence has a length of T. The hidden states in the RNN are updated on each time point t using input x_t. This can be mathematical defined as [11, 106]

$$h_t = f(h_{t-1}, x_t)$$

where f is the activation function. As was earlier discussed in this chapter, there are different options for the activation function, namely sigmoid, tanh or relu [57]. The workings of such an RNN cell are displayed in Figure 9, where a tanh activation function is used.

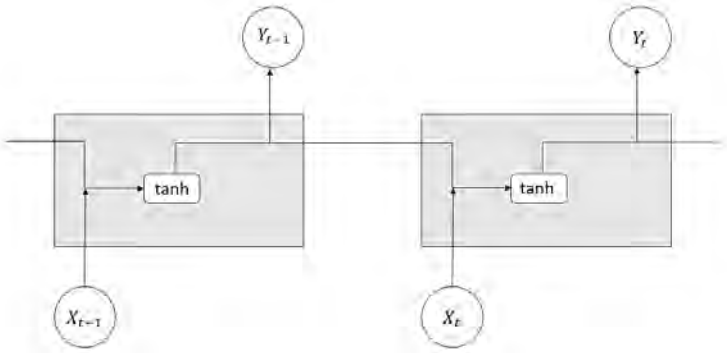

Figure 9: RNN cells with tanh activation function

The RNNs tend to run into two problems, these being vanishing gradients and exploding gradients. Vanishing gradients occur when the gradients are too small and continue to become smaller, resulting in a gradient of zero. When this occurs, the algorithm is no longer learning. The opposite is true with exploding gradients. Then, gradients become too large, creating an unstable model. [57, 41, 10]. A variant exists of the RNN architecture which overcomes the vanishing gradient, namely Long Short-Term Memory (LSTM) [57].

LSTM layers employ multiplicative gates that apply constant error flow through the internal states of special units, called 'memory cells' [54, 67, 81]. There are three gates in the LSTM that control the flow of information to the next memory cell. These gates ensure that the irrelevant information is discarded from the memory cells, and allows for long term memory storage [54, 67, 81]. A visual representation of the structure of an LSTM cell can be seen in Figure 10. It must be noted that in the figure h_t is notated as $a^{<t>}$.

The first gate is the 'forget gate', which determines to what extent the information from the previous memory cell should be forgotten. This gate is visible on the left side in Figure 10, and employs a sigmoid function for the multiplication of the previous output with the current input. Second there is the 'input gate'. This determines how much of the information from the previous layer is written on the internal cell state. In other words, this gate decides which information will be added to the new memory cell state. This gate can be seen in the middle part of Figure 10, using sigmoid and tanh functions. Lastly, there is the 'output gate' with determines to what extent the information is outputted towards the next cell (the next hidden state). This is shown in the right part of Figure 10, using sigmoid and tanh functions. [43, 67, 102]

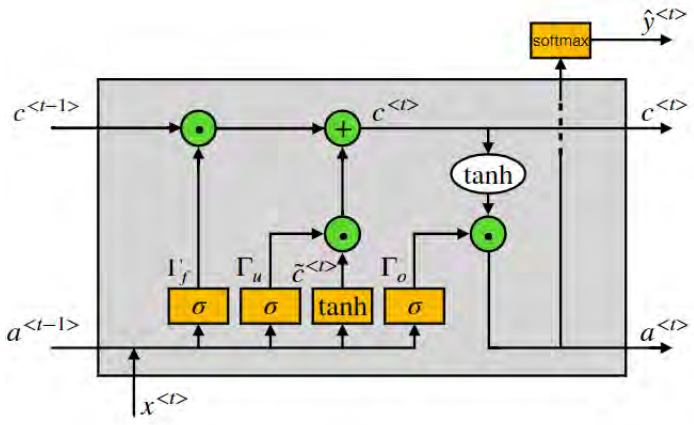

Figure 10: Structure of one LSTM memory cell [11]

In the next part of this section, a more indepth mathematical explanation is presented on the inner workings of the LSTM.

The memory cell in an LSTM unit works as follows. Have $c^{<t>}$ indicate the long memory and $h^{<t>}$ ($= a^{<t>}$ in figure) indicate the short memory in the cell. Then, take the current input $x^{<t>}$ and the previous long and short memory, $c^{<t-1>}$ and $h^{<t-1>}$. Use these to calculate the values for the forget gate and the update gate for candidate memory.

$$\Gamma_f = \sigma(W_f[h^{<t-1>}, x^{<t>}] + b_f)$$
$$\Gamma_u = \sigma(W_u[h^{<t-1>}, x^{<t>}] + b_u)$$

where W are the weight matrices and b is the corresponding bias. In both equations, the parameterized vector is calculated for the current input and the previous short memory by element-wise multiplication with the corresponding weights. Then apply the sigmoid activation function to both equations. After the forget and update gate have been calculated, the next step is to determine the candidate memory cell from which the new long memory can be calculated

$$\tilde{c}^{<t>} = \tanh(W_c[h^{<t-1>}, x^{<t>}] + b_c)$$
$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + \Gamma_f \cdot c^{<t-1>}$$

For the candidate memory cell $\tilde{c}^{<t>}$ the parameterized vector is calculated the same way as for the forget and update gate. The only difference is the presence of the *tanh* activation function. The current long memory value is calculated using the forget and update gate in combination with the previous long memory and candidate memory cell. Next, the output gate is calculated. This calculation follows the same steps as the forget and update gate, only the weights and bias will have different values.

$$\Gamma_o = \sigma(W_o[h^{<t-1>}, x^{<t>}] + b_o)$$

The values for the output gate and the long memory are then used to obtain new values for the short memory and produce an output $\hat{y}^{<t>}$.

$$h^{<t>} = \Gamma_o \cdot \tanh(c^{<t>})$$
$$\hat{y}^{<t>} = g(W_y a; = h^{<t>} + b_y)$$

For the new short memory, a *tanh* activation function is applied on the long memory and then element-wise multiplied with the output gate. The activation function in the output of the memory cell $\hat{y}^{<t>}$ is often the $softmax$ function, but it depends on the problem which activation function is best to select. The obtained long and short memory values are than passed on to the next memory cell, and the whole process is repeated. [11]

3.6 Isolation Forest

There are different outlier detection techniques available. According to the literature, which will be presented in the next chapter, Isolation Forest is the most appropriate outlier detection technique to apply on the latent space of the VAE and β-VAE. The reason for selecting Isolation Forest is that it shows to be an excellent method to efficiently identify outliers [28]. Isolation Forest is a tree-based algorithm, built around the concept of random forest and decision trees [3]. In this technique a tree structure can be build to isolate every single data point. Because anomalies are sensitive to segregation, the anomalies are isolated closer to the root of the tree. In contrast, normal points are isolated at the deeper end of the tree. Thus, anomalies are the instances which have a short average path length on the trees. This is because, (i) fewer instances of anomalies result in a smaller number of partitions - shorter paths in a tree structure, and (ii) instances with distinguishable attribute-values are more likely to be separated in early partitioning. So for the detection of anomalies, a forest collection of random trees that produce shorter path lengths for some points is desirable.

The first step is to define what an isolation tree is. Let T be a node of an isolation tree. T is either an external node with no child, or an internal node with one test and exactly two daughter nodes (T_l, T_r) in the left or right branch of a tree. The test node consists of an attribute q and a split value p such that the test $q < p$ divides the data points into T_l and T_r. There is a sample of data $X = \{x_1, \ldots, x_n\}$ of n instances. When building an isolation tree we recursively divide X by randomly selecting q and p, until either: (i) the tree reaches a height limit, (ii) there is only one data point left ($|X| = 1$), or (iii) all data points in X are identical. Assuming that all instances are distinct, so each instance is isolated to an external node when the tree is fully grown, then n is the number of external nodes and $n - 1$ is the number of internal nodes.

A ranking is needed that reflects the degree of irregularity. A way to do this is to determine an anomaly score for each data point and then sort these scores from high to low. A path length is needed to calculate these anomaly scores. Assume that the path length $h(x)$ of a point x is measured by the number of edges x traverses in a tree from the root node until the external node. It is difficult to derive an anomaly score from $h(x)$. This is because the maximum possible height of a tree grows in the order of n, while the average height grows in the order of $log\ n$. When normalizing $h(x)$ with n or $log\ n$ it is either bounded or cannot be directly compared. A solution would be to estimate the average $h(x)$ using the unsuccessful search in Binary Search Tree (BST). This is possible since the trees have an equivalent structure to BST. The average path length of unsuccessful search in BST, given that there are n instances, can be calculated as follows

$$c(n) = 2H(n-1) - (2(n-1)/n)$$

where $H(i)$ is the harmonic number, which can be estimated by $ln(i) + 0.5772156649$ (Euler's constant). As $c(n)$ is the average of $h(x)$ given a set n, it is possible to normalize $h(x)$ using $c(n)$. So, the path length $h(x)$ and the average path length $c(n)$ are used to obtain the anomaly score s of an instance x

$$s(x, n) = 2^{-\frac{E((h(x))}{c(n)}}$$

where $E(h(x))$ is the average path length from a collection of isolation trees. From the equation above the following observation can be made: when $E(h(x)) \to c(n)$ then $s \to 0.5$; when $E(h(x)) \to 0$ then $s \to 1$; and when $E(h(x)) \to n - 1$ then $s \to 0$. Using these observations the following assessment can be made based on the anomaly score s:

- if a data point returns an anomaly score (s) close to 1, there is a very high chance that this data point is an anomaly
- if a data point returns an anomaly score (s) much smaller than 0.5, the data point can be regarded as a normal instance

[73, 76, 123]

The paper by Liu et al. [76] demonstrated a visual example of the anomaly scores, which can be seen in Figure 11. In the example there are sixty-four points from a Gaussian distribution. The anomaly scores are plotted using contour lines. In the example the contour lines are $s = 0.5, 0.6, 0.7$. When looking closely in the figure, it can be observed that there are three points that have an anomaly score $s > 0.6$. Since these three points are above 0.6, they are identified as potential anomalies and need to be investigated further to determine if they are indeed anomalies.

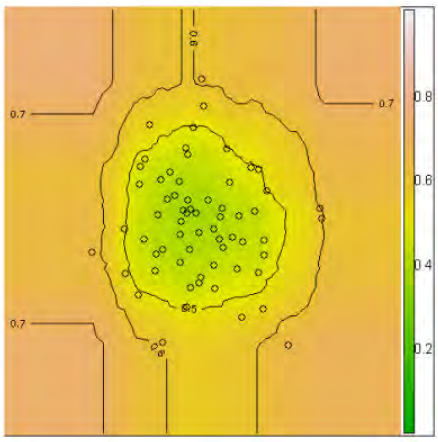

Figure 11: Example of anomaly scores for Isolation Forest [76]

4 Literature

In this chapter an extensive literature review is presented. The detection methods for VAT/-carousel fraud are mostly based on supervised machine learning techniques [26]. As was stated in the introduction, supervised machine learning techniques use labeled data. Unfortunately, obtaining a labeled data set of tax declarations is a slow and costly process [26]. Therefore, there is a growing need for an unsupervised detection system for VAT/carousel fraud.

Currently applied techniques by financial institutions

At present, financial institutions often use rule-based systems to detect VAT/carousel fraud [18, 26]. Rule-based systems implement a set of *if-then* rules to detect VAT/carousel fraud. This set of rules is developed with the knowledge of VAT/carousel fraud experts and historical cases [26]. These rule-based systems have two disadvantages: the first one being that these systems cannot discover new types of VAT/carousel fraud since they heavily rely on past experiences and secondly, building, maintaining and updating these systems is expensive and extremely time consuming [26]. Nowadays, fraudsters are applying more sophisticated techniques to commit VAT/carousel fraud. Therefore, it is not effective to identify fraud by following the well-known patterns which are obtained in rule-based systems. [1, 2] It can even be stated that it is unrealistic to find fraudsters using the current methods. The solution for this would be to utilize more sophisticated techniques, such as machine learning, to develop models that can discover new patterns which can not be found manually [1].

Machine learning techniques for VAT/carousel fraud detection

Fraud is a rare phenomenon and a useful technique to detect fraud is anomaly detection. Anomaly detection is the identification of data points that do not conform with the expected pattern of the data, these anomalies occur infrequently and can be used to detect fraud [2, 53, 116]. Different surveys and studies have shown that anomaly detection techniques only account for 2% of the total research for VAT/carousel fraud detection systems [120]. In fact, very few research papers can be found on the area of VAT/carousel fraud detection. To the best of our knowledge, there is only one paper that uses an unsupervised anomaly detection technique for VAT/carousel fraud. The authors implement the nearest neighbour algorithm as a detection system [120]. The results in this paper were evaluated using hit curves and lift rates. These evaluation metrics demonstrated an success in most sectors.

Because there are not many studies that discuss unsupervised methods for VAT/carousel fraud detection, we will zoom out and dive into detection methods for financial fraud, because more research has been conducted in that area. Since VAT/carousel fraud is a subcategory of financial fraud, the applied detection techniques can most likely also be used in this domain.

Machine learning techniques for financial fraud detection

In the area of financial fraud detection much more research has been conducted. Examples are detection systems for credit card fraud or health insurance fraud [4]. The current approach for financial fraud detection is to implement machine learning techniques [4, 18, 26, 109]. There are several reason why machine learning techniques work well as detection systems: it automates the whole process, helps to reduce the manual work of screening and checking statements, and it is able to identify valuable information by finding hidden trends, relationships, and patterns in a large database which were previously unknown [4, 26, 109]. Machine learning can be defined as "a process that uses statistical, mathematical, and artificial intelligence techniques to extract and identify useful information and subsequently gain knowledge from a large database" [4, 109]. Therefore, machine learning techniques can be used to discover new fraud patterns.

The following methods are often employed for financial fraud detection: logistic regression, decision trees, neural networks, support vector machines, Bayesian belief networks, and naïve

Bayes [4, 109]. A review on the different machine learning techniques used for detecting financial fraud by Albashrawi, M., stated that logistic regression is the most frequently used technique. Neural networks, decision trees, naïve Bayes and support vector machines have also been employed frequently for the detection of financial fraud [4, 109]. Based on the review by Albashrawi, M. and Hilal et al. it can be stated that supervised learning has been applied more frequently than unsupervised learning.

Examples of supervised learning methods are: support vector machines, decision trees, random forest, hidden markov models, multilayered perceptron networks, logistic regression, k-means clustering, k-means nearest neighbour, and naïve Bayes [2, 53]. The graph-based anomaly detection has also been on the rise, since it can analyze connectivity and relation patterns in a large network to identify unusual patterns [53, 96]. As was mentioned earlier, obtaining labeled data for VAT/carousel fraud detection is extremely time consuming. Thus, there is a need for unsupervised learning methods for the detection of VAT/carousel fraud.

Unsupervised learning for financial fraud detection

Recently, more research has been conducted on unsupervised learning methods, since no labeled data is needed. According to a review of Adamov et al., financial fraud detection methods perform differently based on the specific type of financial fraud [1]. In recent years, most financial fraud detection methods were applied to credit card fraud [53]. One paper demonstrated supervised and unsupervised methods for credit card fraud. Here, the authors investigated the following unsupervised methods: one class support vector machines, restricted boltzmann machines, autoencoders and generative adversarial networks. The methods were applied to a public data set containing credit card transactions made by European cardholders in September 2013. The methods were measured with the use of the area under the receiving operating curve (AUROC) and displayed quite promising results for all methods with a AUROC score of above 0.90. [86] Another paper on unsupervised methods for credit card fraud applied one class support vector machines and autoencoders. The data set was based on real-life data of credit card transactions. In this paper, the error rate was used as a measuring scale. Both performed rather well but a slight preference to the autoencoder. [99] A different paper described unsupervised methods for other types of financial fraud, namely: temporal evolution of large dynamic graphs, unsupervised neural networks, recurrent neural networks, cluster analysis, outlier detection, spike detection, principal component analysis, and hidden markov models [93]. The authors stated that link analysis and graph mining are very promising techniques for anti-terrorism and law enforcement, so perhaps these methods perform likewise in the area of financial fraud [93]. The paper by Hilal et al. describes the following unsupervised methods for financial fraud detection: isolation forest, self-organizing maps, autoencoders, sparse autoencoders, denoising autoencoders, contractive autoencoders, variational autoencoders, hidden markov models and generative adversarial networks [53].

In the last few years, deep learning architectures such as generative adversarial networks, convolutional neural networks, long short-term memory networks and autoencoders have become the most promising techniques [53]. The reason for the increase in popularity is that these architectures can learn deeper and more complex representations of features in the latent space. These techniques have shown to outperform other techniques in addressing real-world problems. Also, these methods can detect fraud cases previously unknown to domain experts or currently applied detection systems. [53]

Variational autoencoder

In this project the unsupervised method needs to deal with time series data. From the literature, it was obtained that the variational autoencoders seem to be successful as an unsupervised anomaly detection technique for time series data [77, 128]. A variational autoencoder is a type of autoencoder that incorporates a reconstruction probability instead of a reconstruction error [5, 53]. This probability covers the concept of variability better than the error [5, 53]. Another reason why the variational autoencoders are preferred, is that they have a solid theoretical background

and stable performance [5, 77]. In the paper by An et al. it was shown that the variational autoencoder method outperforms standard autoencoders and principal component analysis methods. In that paper the MNIST data set was used and the performance was measured by calculating the AUROC and f1 scores. Based on these metrics the variational autoencoders outperform regular autoencoders and principal component analysis. [5] Next to this, the variational autoencoder significantly outperforms the supervised anomaly detection methods [74].

Time series data is often highly dimensional. The variational autoencoder already works well with time series data, but if the dimensions could be reduced, the variational autoencoder will perform even better [74, 128]. A possible solution to reduce the dimensions would be to incorporate long short-term memory layers as the encoder and decoder part of the variational autoencoder framework [75, 87, 97, 128]. The variational autoencoder module can handle short windows of features well, while the long short-term memory module can handle the long term correlations in time series data on top of the features [75]. By combining the variational autoencoder and the long short-term memory techniques, a detection algorithm is developed that can identify anomalies which span over multiple time windows [75]. Different research papers have shown that this combination is proficient and outperform other commonly applied anomaly detection methods [75, 87, 97, 124, 128].

The paper by Pereira et al. implements the variational autoencoder in combination with long short-term memory layers and demonstrates promising results. The authors apply anomaly detection in the latent space of the variational autoencoder. The papers uses the ECG500 data set to detect anomalies in heart rates. The authors obtained an accuracy score of 0.9596, an area under the curve score of 0.9819 and an f1 score of 0.9522 for the predictions of five labels in the ECG500 data using the latent space of the variational autoencoder. [92] This project will use this paper's approach as a base to detect anomalies in financial time series data to investigate whether anomalies can be spotted using the same techniques but in a different data set.

Outlier detection techniques

In the paper we previously cited, the authors applied k-means clustering as an outlier detection technique in the latent space of the variational autoencoder [92]. Based on literature, we will investigate the different possibilities for the outlier detection technique and choose the most promising one. Notably, there is no generic or single universally applicable outlier detection approach [55]. In a survey by Domingues et al. and Hodge et al. the following outlier detection techniques were explored: gaussian mixture models, dirichlet process mixture models, kernel density estimators, principal component analysis, least-squares anomaly detection, local outlier factor, angle-based outlier detection, kullback-leibler divergence, grow when required networks, one-class support vector machines, self organising maps, k-nearest neighbours, decision trees, similarity-based matching and isolation forest [28, 55]. The survey by Domingues et al. applied the techniques on unseen testing data and used performance measures such as AUROC and precision-recall curves [28]. The authors demonstrated that isolation forest is outperforming all other detection algorithms on different data sets and that it has a good average performance, making it a reliable choice [28]. The paper by Hilal et al. also stated that isolation forest is capable of handling massive data sets and high-dimensional problems [53]. The application of isolation forest has only emerged in the last few years and has mostly be applied to credit card fraud detection. The paper by Ounacer et al. applied an isolation forest technique to detect fraudulent behavior in credit cards. This method was compared to other models such as local outlier factor, k-means clustering and one-class support vector machine. The results of this paper demonstrated that isolation forest has the best performance with the highest accuracy of 0.9512 and the highest area under the curve score of 0.9168. In second place, there is the k-means clustering algorithm. [89] Other techniques, such as principal component analysis and discounting learning algorithms, were also applied to credit card fraud detection, but showed less promising results compared to isolation forest [79, 91]. All in all, the literature demonstrated that isolation forest is an efficient method to identify outliers and this method already displays promising results for the detection of credit card fraud.

Interpretability and explainability of black-box models

A problem for all deep learning models, which are also called black-box models, is that they lack interpretability and explainability [53, 109]. If it would be possible to improve the interpretability of these models, it could allow for results that are explainable and thus making human interaction possible. Then these results could be better used to support decision-making. [53, 109]. The variational autoencoder and long short-term memory are considered to be deep learning models. Recently, a shift has taken place towards increasing the interpretability and explainability of these black-box models. A way to progress would be to apply the β-variational autoencoder. The extra hyperparameter β encourages the representation of the latent space to be more factorized and thus more disentangled. [17, 52] This β-variational autoencoder ensures that dimensions represent more distinct features and thus can be used to interpret the result more clearly.

Main conclusions

- Currently, financial institutions implement rule-based system to detect VAT/carousel fraud. However, these systems are not effective in identifying fraud.
- Very few research papers can be found on the area of applying unsupervised machine learning methods to detect VAT/carousel fraud.
- Many supervised machine learning methods in the area of financial crime have been researched. Some examples are: support vector machines, decision trees, and random forest.
- Variational autoencoder in combination with long short-term memory layers seems the most promising unsupervised machine learning technique for VAT/carousel fraud detection.
- Isolation Forest demonstrated high results in other areas of fraud as outlier detection technique.
- The β-variational autoencoder can help disentangle the latent space of the variational autoencoder. As a result it can perhaps improve the interpretability and explainability.

5 Data analysis

This project is executed in cooperation with a financial client of Deloitte, a bank which requested to remain anonymous. The goal of this project is to develop an unsupervised machine learning detection system for VAT/carousel fraud. The bank provided a data set with transactions of clients, for the development of this model. The data set contains transactions where VAT/carousel fraud and Rapid Movement of Funds (RMoF) occurred, where only RMoF occurred and where neither occurred. The exact number of available business numbers, contract numbers and transactions for the data set can be seen in Table 1. The data set contains 25 columns with information. Some examples of these columns are: business number, contract number, mutation amount, debit or credit mutation, balance, date and time of transaction, the currency, from which bank account the transactions were made and to which bank account the money is transferred. Each client receives a unique business number, but it is possible that within a business number there are multiple contract numbers. The contract numbers are unique and each contract number displays the movement of the balance over time, combined with other information. Since the contract numbers are unique, the detection of VAT/carousel fraud will happen at this level. Fraud is a rare phenomenon and thus resulting in limited available cases of actual fraud, making the data set often highly imbalanced. This can also be observed in this data set. The initial data set this project used was later expanded with more data points. The division of this can be found in Table 2. The initial data set was used for the data analysis and feature selection. The initial data set will not be part of the test set, to prevent leakage of information.

	# business numbers	# contract numbers	# transactions
RMoF & VAT/carousel fraud	20	31	22.756
RMoF	4	4	1.172
no RMoF & VAT/carousel fraud	280	451	692.448
Total	304	486	716.376

Table 1: Complete data set division

	# business numbers	# contract numbers	# transactions
RMoF & VAT/carousel fraud	10	18	17.324
RMoF	4	4	1.172
no RMoF & VAT/carousel fraud	96	145	252.8083
Total	110	167	270.579

Table 2: Initial data set division

A strong indicator of VAT fraud is the appearance of the phenomenon Rapid Movement of Funds (RMoF). No official definition for RMoF can be found in the literature, so the following definition was used for this project: *Rapid Movement of Funds (RMoF) occurs when large sums of money are flowing in and out of a bank account in a very short period of time, mostly within a few hours.* RMoF is not only an indicator for VAT fraud, but also an indicator for other types of fraud, making the detection of RMoF extremely valuable for the bank. A more indepth explanation and investigation about RMoF will be presented in the next chapter, where the features for the models are discussed.

In this chapter, the cleaning method is discussed first. Then, a data analysis is conducted to obtain some useful insights. The part of the data set which contains RMoF and VAT/carousel fraud will be referred to as the fraud set, and the non-RMoF and non-VAT/carousel fraud part will be referred to as the non-fraud set. For the cleaning and analysis not all 25 columns are used, but a selection of 7 columns was made. The following columns were selected: business number, contract number, date, time, debit or credit mutation, mutation, and balance. These columns were selected to provide the information needed to observe balance over time.

5.1 Cleaning

The first step for cleaning the data is checking if there are any missing values or NaN values in the columns of the data set. Only the columns needed for the data analysis and the feature selection were checked for missing values or NaN values. For the columns 'balance' and 'mutation' it would be possible to calculate missing values, using the previous information. In the initial data set, there were three rows with missing values. Upon further investigation all the 25 columns had missing values, so it was not possible to retrieve which information should have been there and hence these rows were deleted. It can be stated that the data set was of good quality, since in the complete data set only three rows of missing values were found.

Next, two specific steps are required to properly process the data in Python. The first step is to format the time and date correctly. In the original data set, it was not possible to order transactions in chronological order, since the time and date were in two separate columns and written as an integer. This was corrected, such that it was possible to order transactions on chronological order within a contract number. The second step is to convert commas to dots in the columns 'balance' and 'mutation' for Python to work.

The data is now cleaned and ready for the data analysis

5.2 Analysis

The data analysis is performed to obtain better insight into the data. The contract numbers have transactions at different times and dates, making it hard to compare them. A solution for this problem would be to aggregate the transactions of the different contract numbers, such that it is possible to compare them with one another. To execute this, an appropriate aggregation rate and moment need to be determined. This rate and moment will be the same for all contract numbers.

The aggregation rate determines how many times the balance needs to be aggregated on one day. For example, a rate of 4 would indicate that on 4 moments in the day the balance is aggregated. So for every day there are 4 data points. The aggregation moment determines on which hours of the day the data is aggregated. Continuing on the previous example, four hours in the day need to be selected on which the balance is aggregated. For example, for every day at the hours 00:00, 06:00, 12:00 and 18:00 the balance is aggregated. The aggregation rate and moment are both needed to aggregate over all the data.

Aggregation rate and moment

Start with some descriptive numbers about the transactions for the fraud and non-fraud set. These numbers can be found in Table 3 and Table 4. The numbers obtained from the table show the maximum and minimum number of transactions on one day within one contract number. Both sets have the same minimum of 1, but the maximum differs between the two sets. In these tables the median for the fraud set can also be found, which equals 3. This implies that 50% of all transactions would be preserved when 3 aggregation moments are taken on one day. For the non-fraud set the median equals 2, which implies that 2 aggregation moments need to be taken on one day to preserve 50% of all transactions. The two tables show a slight difference between the two sets.

Total number of days with transactions	2859
Maximum number of transactions on one day	80
Minimum number of transactions on one day	1
Median of number of transactions on one day	3

Table 3: Descriptives of number of transaction for fraud set

Total number of days with transactions	61962
Maximum number of transactions on one day	133
Minimum number of transactions on one day	1
Median of number of transactions on one day	2

Table 4: Descriptives of number of transaction for non-fraud set

To determine the aggregation rate, look at the distribution on how many transactions there are on one day. The days on which no transactions were made are omitted from the figures. This can be seen in Figure 12 on the left for the fraud set and on the right for the non-fraud set. With the use of this figure, an appropriate aggregation rate can be determined. In the figure it can be observed that for around 30% of cases, only one transaction occurs on a day for both sets. This means that when taking an aggregation rate of one (so one sample a day), this will result in the preservation of 30% of all transactions. This is a bit low, so an aggregation rate of more than one needs to be chosen. In order to determine an appropriate aggregation rate, it is also important to look at which hours of the day the transactions occur. A distribution was developed which shows how many transactions are made on which hour of the day, displayed as percentages of the total amount of transactions. This is visible in Figure 13 for fraud set on the left and the non-fraud set on the right. For both sets, it is visible that most transactions occur between 08:00 and 17:00, thus during working hours. However, two additional peaks are visible at 00:00 and 04:00 for both sets. The most likely reason for these peaks are automated collections.

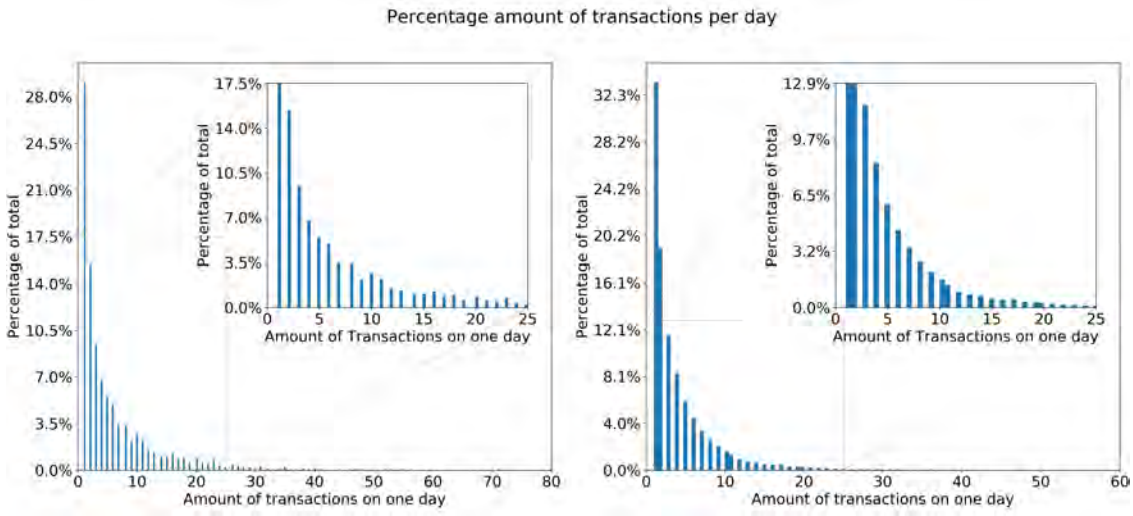

Figure 12: Percentage of total amount of transactions on one day, left fraud set and right non-fraud set

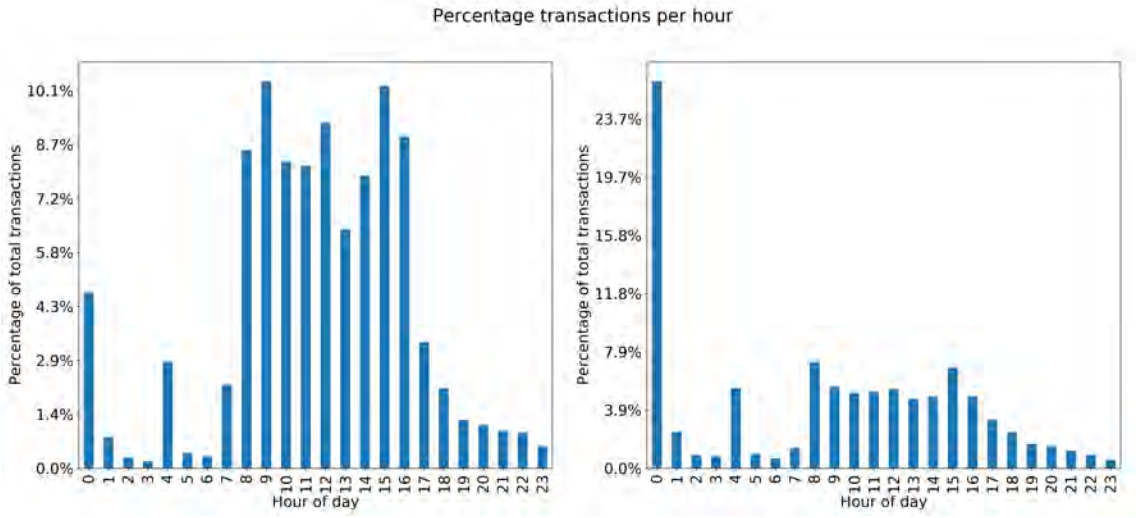

Figure 13: Percentage transactions per hour, left fraud set and right non-fraud set

The aggregation moment determines on what moments of the day samples should be taken. Preferably, these occur on a constant interval. The observed peaks at 00:00 and 04:00 in Figure 13 cannot be missed, otherwise lots of information regarding transactions is lost. Taking this into account, the following aggregation moments are possible: every 4 hours, every 2 hours, or every hour. This results in the following aggregation rate for both sets: for an interval of 4 hours an aggregation rate of 6 is required, resulting in a preservation of around 70% of all transactions. For an interval of 2 hours an aggregation rate of 12 is required, resulting in a preservation of around 85% of all transactions. For an interval of 1 hour an aggregation rate of 24 is required, resulting in a preservation of around 95% of all transactions. For the last aggregation rate not 100% preservation is obtained, since multiple transactions can occur within one hour.

RMoF behavior can be spotted by the fact that large sums of money are flowing in and out of a bank accounts in short periods of time. Choosing too low of a sampling rate will have an ill effect on the results, because some transactions will be missed, making it harder to detect VAT/carousel fraud. The three aforementioned aggregation rates were investigated. The first thing that can be observed is that with a higher aggregation rate, a higher percentage of transactions is preserved. For the first scenario, an aggregation rate of 6, it was concluded that too many transactions were missed during working hours. The second scenario, an aggregation rate of 12, showed more promising results. The last scenario, an aggregation rate of 24, missed almost no transactions. It was discussed with the bank what was more desirable. Missing almost no transactions and thus having more data, or missing a few transactions and achieving faster computation time due to having slightly less data. It was determined that missing almost no transactions was more important. Therefore, an aggregation rate of 24 was chosen.

Thus with an aggregation rate of 24, every hour an aggregation moment occurs for every contract number. This is done as follows: the current balance is taken and set as value until a new value for the balance appears. Then this becomes the new value for the next hours until a new value occurs, etc. This continues till every contract number has a data point for every hour on each day.

Normalization

The aggregation has been applied to the data. Now it is possible to compare the different contract numbers. But there is still a problem which is the difference in magnitude of balance between contract numbers, making it hard to compare. The solution is to normalize the data. Another

reason why normalizing the data is advantageous is that the Variational Autoencoder (VAE) works better with values between zero and one. This way, the VAE sees the magnitude of the transactions not as a possible feature. A disadvantage of applying normalization is that information regarding the real amount of balance over time will be lost.

Normalization can be applied in many different ways. For this project the decision was made to apply normalization within each contract number and scale all balance values between zero and one. The following formula is applied [6]

$$z_i = (x_i - min(x))/(max(x) - min(x))$$

where x_i indicates the current balance value of the contract number, $min(x)$ is the minimum balance value in a contract number, $max(x)$ is the maximum balance value in a contract number, and z_i is the normalized balance value. By applying this formula per contract number, the lowest balance value within a contract number will get the value zero and the highest balance value will get the value one. The rest of the balance values within that contract number are scaled between zero and one. Using this normalization technique, the normalized samples in one account represent other values relative to other accounts. As a result, the contract numbers can be compared, even though they deal with different orders of magnitude.

Normalized balance over time

The mean and standard deviation over the normalized balance are calculated for the fraud and the non-fraud set. This was done to investigate whether there are differences. The mean in the fraud set is lower compared to the non-fraud set. This indicates that the average balance for the fraud set is lower compared to the non-fraud set. The standard deviation is almost the same for both sets. What can be observed is that both standard deviations are low, which indicates that most balance values do not deviate much from the mean value. The exact numbers for the mean and standard deviation can be found in Table 5 for both sets.

Mean fraud set	0.1896
Standard deviation fraud set	0.1173
Mean non-fraud set	0.3357
Standard deviation non-fraud set	0.1539

Table 5: Mean and standard deviation over the normalized balance for both sets

The balance over time is plotted for fraud and non-fraud set contract numbers separately to investigate if these display different behaviors. Based on the previous table, the expectation is that a different behavior will be visible, since the mean values differ between the two sets.

The balance over time for contract numbers in the fraud set display very short high peaks and after the peak the balance returns to around zero. This behavior would support a low mean, which was obtained for the fraud set. A visual example is presented in Figure 14. In the figure, the balance over time for two random contract numbers from the fraud set are plotted. The other 16 contract numbers are also plotted and can be seen in the Appendix. In the fraud set there are two contract numbers that only have one transaction. These will not be shown in the Appendix since no figure could be generated. A few contract numbers in the fraud set have very few transactions. As a consequence, the behavior is not really visible. That said, the majority of the contract numbers of the fraud set display the short high peaks behavior.

The balance over time for contract numbers in the non-fraud set display a very different behavior. Here, the behavior is more a slow incline or decline that occurs over time, so no short peaks are visible. This behavior would support a higher mean compared to the fraud set, since there are no sudden peaks and the balance is not around zero often. A visual example is presented in Figure 15. In the figure, the balance over time for two random contract numbers from the non-fraud set are plotted. Visually it can be observed that the pattern of balance over time is quite different compared to the fraud set. This different behavior is also supported by the difference in the mean.

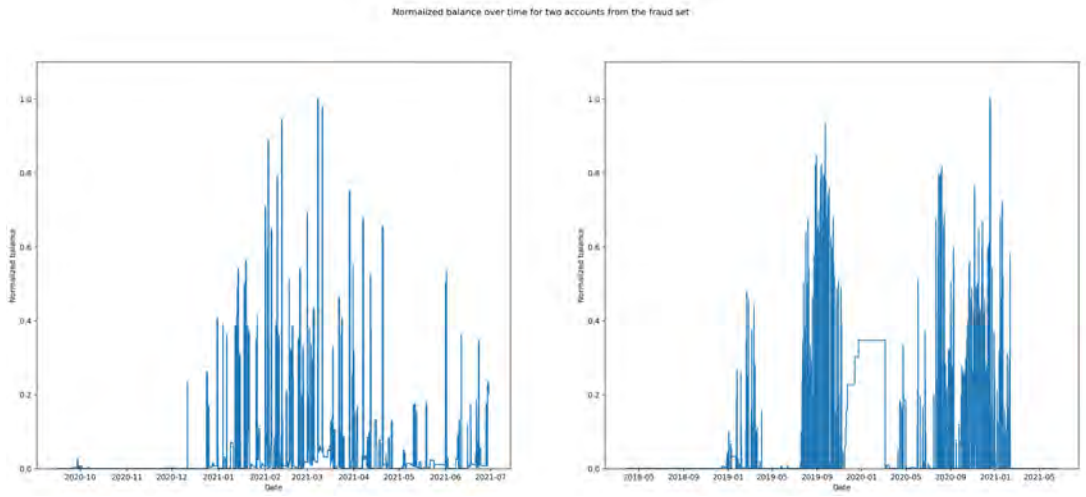

Figure 14: Balance over time for two random contract numbers from fraud set

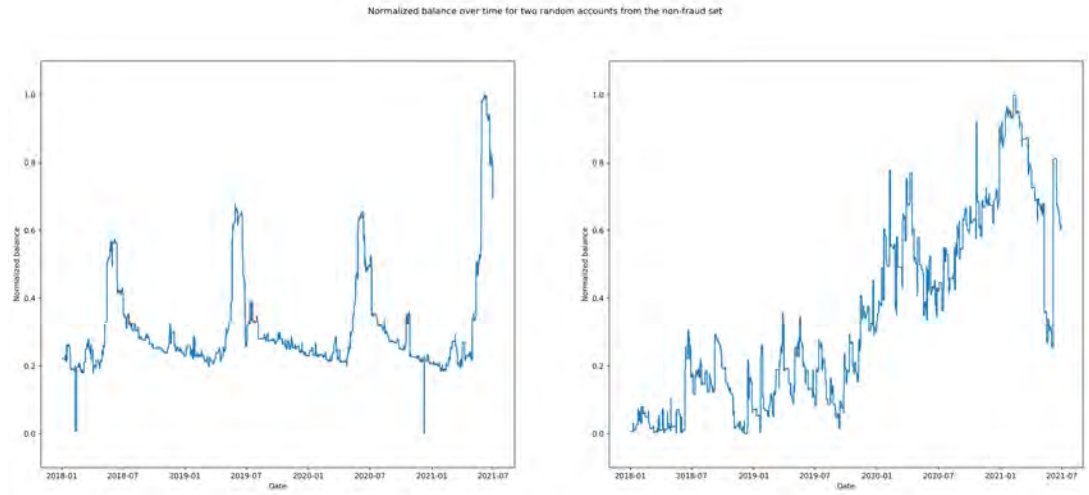

Figure 15: Balance over time for two random contract numbers from non-fraud set

The figures presented above already give a good visual of the balance over time. Since the balance is normalized, it is possible to compare the different contract numbers. However, most contract numbers have a different duration, because for every contract number the amount of available data is different. Some contract numbers have six months of data, while others have three years of data. Consequently, it can be difficult to compare the balance over time for contract numbers based on the duration.

Unfortunately, this is exactly what this project needs because we want to compare the different contract numbers. A solution would be to look at the balance over time per month. Then the same length is used for every contract number, making it possible to compare them. The phenomenon RMoF is also more observable when looking at monthly data. The pattern of small high peaks and then returning back to zero is clearly visible per month. An example of this can be seen in Figure 16, where the balance over time for two months for a random contract number in the fraud set is plotted.

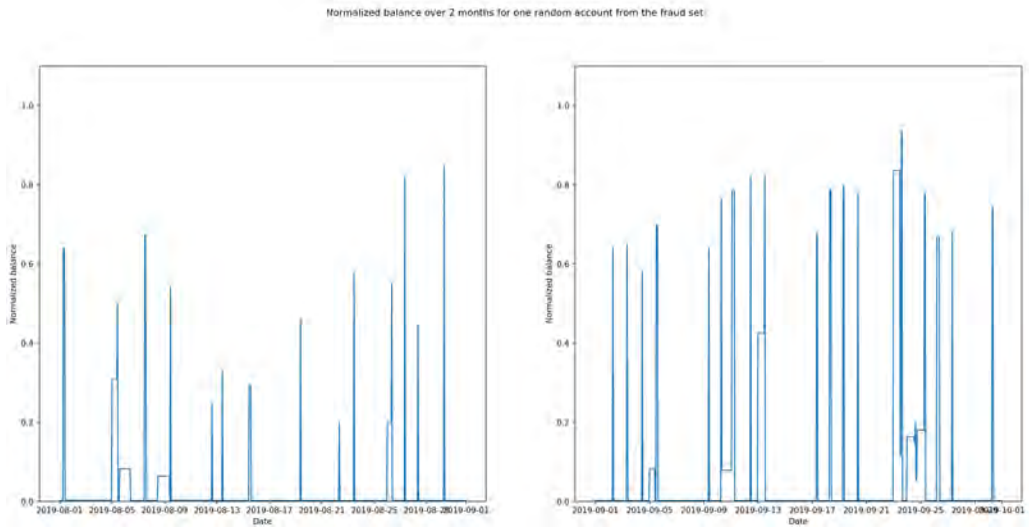

Figure 16: Balance over time per month for one random contract number from the fraud set

6 Method

In this chapter the methods are described. As was stated in the first part of the research question, we want to investigate whether a multivariate analyses improves detection of outliers compared to a univariate analyses for a Variational Autoencoder (VAE). The first method that will be implemented is the VAE. The second part of the research question explores whether the β-Variational Autoencoder (β-VAE) improves the interpretability and explainability. With explainability is meant that the features are more disentangled in the latent space, and as a result it can be determined which combination of features result in outliers. So, the second method that will be implemented is the β-VAE. Besides the implementation, both methods need input. The input for both methods is created using features. A total of five features were developed and are described in the first section of this chapter. The last two sections in this chapter describe the implementation of the VAE and β-VAE. An overview of the steps in the process is presented in Figure 17.

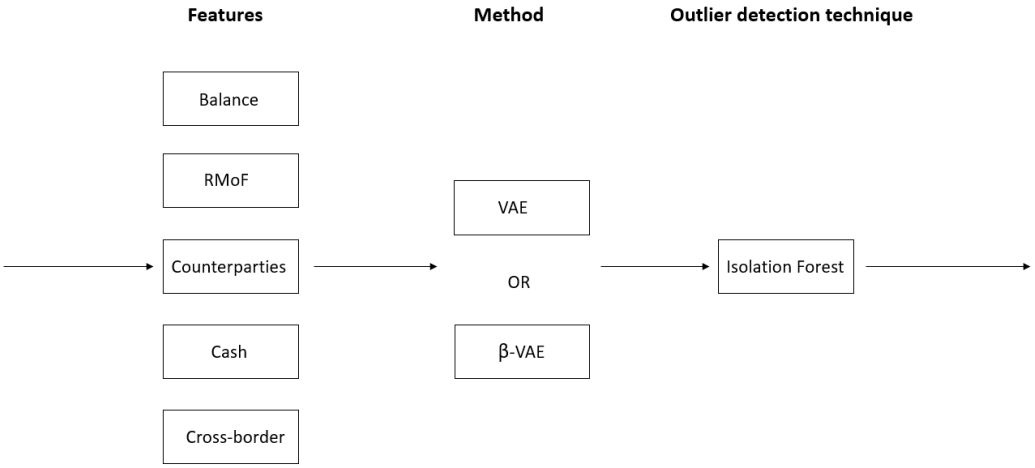

Figure 17: Overview of the process

6.1 Features

A total of five features were developed and implemented, namely: balance over time, Rapid Movement of Funds, unique counterparties, cash, and cross-border transactions. All features are developed as a rhythm over time. It was decided to use a sliding window concept, to increase the number of samples. Every window consists of 28 days and has data points for every hour, thus resulting in 672 data points per window. The window is shifted after 7 days and again a window of 28 days is taken. This process is repeated for the lifespan of a contract number. The five features are developed with the initial data set and the division of this can be found in Table 2. In the next sections each feature will be explained more indepth.

6.1.1 Balance

The first feature that is developed is a rather straightforward one, namely balance over time. This feature displays the rhythm of balance over the time window. The pseudo code for this feature can be seen in the box below. The steps are as follows: first data points are created for each hour, then the balance is normalized per contract number between zero and one. Third the sliding window concept is applied, so create windows of 28 days and shift it after 7 days.

Pseudo code balance

1. *Create a data point for every hour, by aggregating the balance*
2. *Normalize the balance per contract number:* $z_i = (x_i - min(x))/(max(x) - min(x))$
3. *Apply the sliding window concept, with length = 28 days and shift every 7 days*

A visual representation of the balance over time for three windows can be seen in Figure 18. The balance in this figure is rather low, but it is more to demonstrate how the sliding window for balance over time looks like. In the figure no sudden high peaks are observed, so in these three windows most likely no fraudulent behavior is occurring.

The obtained data points for each window are saved in a file, to be used later as input for the outlier detection model.

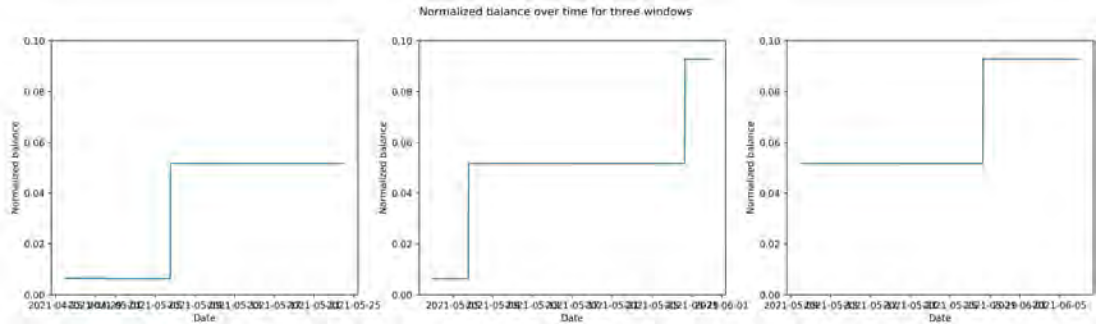

Figure 18: Balance over time for three windows, where each window consists of 28 days

6.1.2 Rapid Movement of Funds

Rapid Movement of Funds (RMoF) occurs when large sums of money are flowing in and out of a bank account in a very short period of time, mostly within a few hours. It must be noted that the detection of RMoF does not automatically imply that VAT/carousel fraud is occurring. A closer investigation of the account is needed to decide if VAT/carousel fraud is occurring or not.

From the previous chapter it was observed that different behavior was visible between RMoF and non-RMoF contract numbers. The balance in RMoF contract numbers have small high peaks and after these peaks return back to zero. This behavior is visually best observed when looking per month at the contract numbers, such as in Figure 16. So, the accounts that have small high peaks and return back to zero in a month need to be identified for the detection of RMoF.

Smoothing filter

A smoothing filter will be applied to the data to detect RMoF in the contract numbers. The approach of applying a smoothing filter to detect RMoF has not been used in the literature, or at least not to our knowledge. The expectation is that this smoothing filter will flatten out the small high peaks that occur in the data. After the smoothing filter has been applied, the area between the normalized balance and the smoothing filter is calculated. A visual example of this can be seen in Figure 19. In the figure the blue line indicates the normalized balance, and the orange line indicates the smoothing filter. The area between the two lines is the grey area. A high value for the area would indicate that many small high peaks have occurred during that month for a specific contract number, which is exactly what we want to detect. While a low value for the area indicates that the balance slowly increased or decreased over time, but no sudden peaks have happened in that month for that specific contract number.

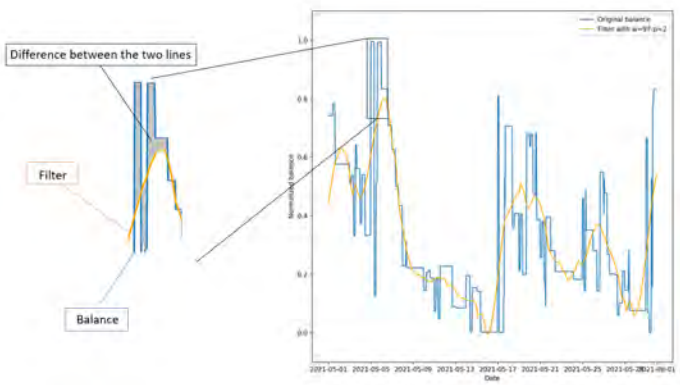

Figure 19: Calculation of the area between two lines

Since there are many smoothing filters available, we need to determine what an appropriate smoothing filter would be for this data. The smoothing filter needs to work in the time-domain, and from the literature it was obtained that the Savitzky-Golay filter would be suitable [104]. Some other frequently used filters were investigated as well, namely: moving average filter, median filter, Wiener filter, and forward/backward signal. The median and Wiener filter almost exactly followed the normalized balance, so these filters did not flatten out the data and were therefore not suitable. For the moving average and forward/backward signal, frequencies are needed as parameter for the filters. Since choosing an appropriate frequency is depended on the signal, it is hard to choose one frequency that could be applied to all data. This was determined based on some example cases from the data. Therefore, it was decided that these two filters were unsuitable too. The filter that remains is the Savitzky-Golay filter. With this information and further investigation it was decided that this filter is appropriate and the parameters needed for the filter can be chosen equally for all data.

The Savitzky-Golay filter is a data smoothing method based on local least-squares polynomial approximation and uses the following formula [39, 103, 104]

$$Y_i = \sum_{i=-(m-1)/2}^{i=(m-1)/2} C_i y_{j+1}$$

where m indicates the windowsize and y_j the observed value. The data is treated with a set of m convolution coefficients and C_i indicates the polynomial degree. The filter works as follows: for each sample in the filtered sequence the filter takes the direct neighborhood of m and fits a polynomial to it. Then the polynomial is evaluated at its center, which is the smoothed value for this point. [122, 83] The Savitzky-Golay filter has two parameters that need to be determined: *windowsize* and *degree of the polynomial*.

Windowsize parameter

The first parameter which needs to be determined is the *windowsize*. In the literature no information was found on what an appropriate windowsize could be to spot RMoF. A possible reason for this lack in literature could be that RMoF is a too specific domain for fraud, such that little research has been done in this area. Because no information was found in the literature it was discussed with Financial Crime investigators from the bank what they viewed as a reasonable windowsize to detect RMoF, based on their recent finished project about RMoF and personal experience. The investigators decided that a windowsize of 4 days would be reasonable. The data containing the fraud cases was used to check if indeed a windowsize of 4 days would be acceptable to use. For this investigation the data was used before it is normalized. The investigation was done as follows, when a balance exceeds a defined boundary it is calculated how many hours it takes before the balance is again below the boundary. This way the duration of peaks are obtained. In

consultation with investigators of the bank the boundary was set at €10.000. It is to be expected that for RMoF the peaks occur in a short period of time, most likely within a few hours.

The results of this investigation for the parameter windowsize are visualized in a histogram, which can be seen in Figure 20. From this histogram it can be observed that most transactions occur within a few hours. The number of peaks occurring in 10 hours is shown in Table 6, this gives an even better insight of how quickly the high peak transactions occur. By combing the information from the histogram and the table it can be concluded that most of the time the balance goes above and below the boundary within just 2 hours. This matches with the expectation that money is transferred in a very short period of time for RMoF cases. In the fraud data a total of 950 transactions were found that went above and below the boundary within a certain time length, the shortest time for these peaks was 2 hours and the longest time was 12743 hours. The result show that the more hours have passed the less peaks occur, this supports the statement that RMoF occurs in short period of time.

To determine an appropriate windowsize from these results it was calculated how many days were needed for a certain coverage percentage. These results can be found in Table 7. When taking a windowsize of 4 days as stated by the experts a 90% coverage is obtained, which is quite high. The statement from the experts is thus supported by the data, and therefore it is reasonable to use 4 days as the windowsize for the Savitzky-Golay filter. For the implementation, the windowsize needs to be multiplied with 24, the reason for this is that we took data points for every hour and not every day. Thus, the filter will have a value of $4 \times 24 = 96$ for the windowsize. To implement this filter in Python the following package is used, $scipy.signal.savgol\_filter$. A requirement of this package is that the windowsize must be odd, so a value of 96 is not allowed. To solve this problem the windowsize is increased with one, having a windowsize of 97.

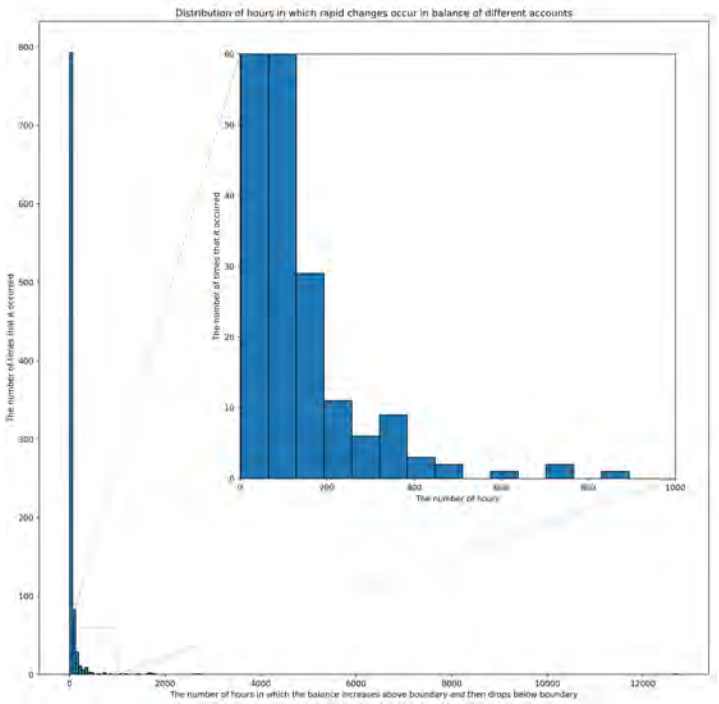

Figure 20: Distribution of hours in which rapid changes occur in balance over different accounts from the fraud data set

Hours passed	Number of times it occurred
2	183
3	88
4	43
5	35
6	16
7	18
8	9
9	14
10	25

Table 6: Descriptives of number of transactions in fraud data set

Percent coverage	Hours	Days
80%	48	2
85%	70	2.92
90%	99	4.12

Table 7: Required days for a certain coverage percentage

Degree of the polynomial parameter

Now the second parameter value needs to be determined, which is the *degree of the polynomial*. Three different options were implemented: 1, 2, and 3. In Chapter 8.1 the best option for the degree of the polynomial will be decided. In this project only three options for the polynomial were investigated due to computation time, in general it would be possible to use any number for the degree of the polynomial.

The Savitzky-Golay filter has a value of 97 for windowsize and the options 1, 2, and 3 for the degree of the polynomial. All three options were implemented, resulting in three filters. The Savitzky-Golay filter is applied to the whole duration of a contract number and after the filter has been applied the contract number is split into months. The reason for this is that the filter has a starting up phase, so the begin of the filter is not yet very accurate. If the filter was applied on each month again it would not have accurate values at the beginning of each month. To prevent this the filter is applied to the whole duration of the contract number and then split into months.

The data points obtained from the filter with polynomial 2 and 3 are almost the same, the difference can be found in the decimals. Since a lower polynomial takes less computing time and power it was decided to go for the lower degree of the polynomial, thus value 2, and ruling out the use of degree 3. So, the decision must be made between polynomial values 1 and 2, which will be done in Chapter 8.1.

Implementation

Using the investigation just conducted, the Savitzky-Golay filter is applied to implement the Rapid Movement of Funds (RMoF) feature. In Chapter 8.1 it was determined to use the following parameter values: *windowsize* of 97 and a *degree of the polynomial* of 2. Apply the pseudo code presented in Chapter 8 to calculate the area between the data points. The difference with the above investigation is that the area per data point is calculated, in contrast to calculating the area per month. This way a rhythm over time can be determined, instead of calculating a number for every 28 days. Over this area a sliding window is applied, again take windows of 28 days and shift it with 7 days. A visual representation of the area over time for three windows can be seen in Figure 21. The peaks in the figure demonstrate that the normalized balance deviates from the smoothing filter, high peaks indicate that it is likely that RMoF is occurring in that window, while low peaks indicate that it is likely that no RMoF is occurring in that window. In this figure the peaks are rather low, so the expectation is that no RMoF is occurring in the three windows.

The obtained data points for each window are saved in a file, to be later used as input for the outlier detection model.

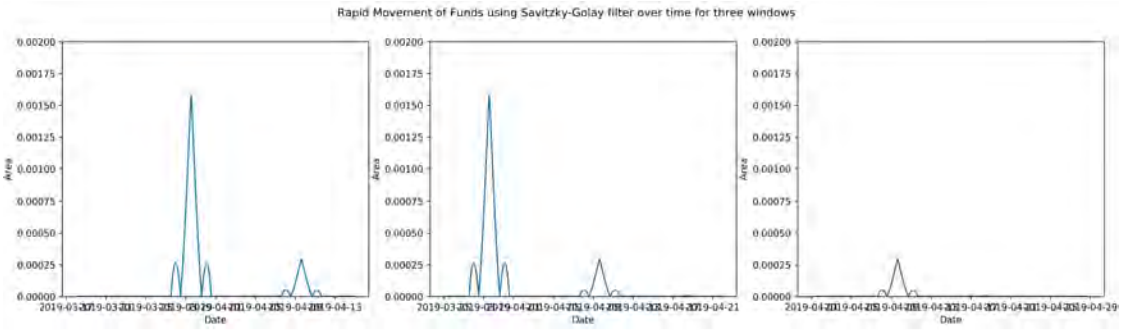

Figure 21: Area between normalized balance and Savitzky-Golay filter for windows, where each window consists of 28 days

6.1.3 Counterparties

The third developed feature is that of unique counterparties in the windows. For this feature we again wanted to display a signal over time. The expectation is that contract numbers which are committing VAT/carousel fraud have fewer counterparties compared to 'normal' contract numbers. Also the pattern of transactions coming from counterparties will be more or less the same for each month for 'normal' counterparties, while fraud contract number are more likely to display a very irregular pattern over the different months.

The development of the counterparty feature was done by determining how many unique counterparties there are for each 28 day window, taking the debit and credit side separately. The debit and credit side are taken separately because it is important to make a distinction between incoming and outgoing money. A debit transaction indicates that money from the contract number is transferred to another account, while a credit transaction indicates that money from a different account is transferred to the contract number.

We do not want to take all the unique counterparties that occurred in the 28 day window. Especially, when a lot of unique counterparties are present in that window. From domain-knowledge it was obtained that in account where VAT/carousel fraud occurred not many unique counterparties were present. Also, these counterparties transfer constantly large sums of money, occur a few months and then disappeared. While at 'normal' contract numbers many counterparties are available and the frequency of the transactions with these counterparties is mostly the same for many months or even years. Therefore, the decision was made to only take the top five counterparties per 28 day window, per debit and credit side. It is hoped that with taking only the top five, counterparties dealing with relatively small amounts of money are discarded. A downside of only taking five counterparties for debit and credit side is that some valuable information might be lost.

The steps in the pseudo code for the counterparty feature are as follows: first data points are created for each hour, then the balance is normalized per contract number between zero and one. Third the sliding window concept is applied, so create windows of 28 days and shift it after 7 days. Then we loop over all the windows. Fifth step is to select per window the five counterparties with highest mutation amount in that window. This is done separately for debit and credit. Next the order of the counterparties is determined, based on the mutation amount in that window. The next steps are done per unique counterparty in the window. A column is added with indicates the counterparty transactions. Then fill in this table per data point and set the normalized balance to zero if no transaction occurred of that counterparty at that data point. So per window five debit

and five credit rhythms are determined.

Pseudo code counterparties

1. Create a data point for every hour, by aggregating the balance
2. Normalize the balance per contract number: $z_i = (x_i - min(x))/(max(x) - min(x))$
3. Apply the sliding window concept, with length = 28 days and shift every 7 days
4. Loop over the windows
5. If number of counterparties > 5, select 5 counterparties with highest mutation amount
6. Determine order of 5 counterparties based on mutation amount in the window
Following steps are done per unique counterparty in the window
7. Create extra column which indicates counterparty transaction with 1 and no counterparty transaction with 0
8. If data point = counterparty transaction, label it with 1
9. If data point ≠ counterparty transaction, label it with 0
10. Where labels equal 0, the normalized balance is set to zero

With the use of the pseudo code the unique counterparties transactions are displayed as peaks over time, which is exactly what we wanted. The model can combine the balance over time and these counterparties peaks to determine the volume of the mutation and can learn at which moments in time a unique counterparty transaction occurred. Smaller peaks would likely indicate that not a lot of money is transferred, while higher peaks are more likely to display transactions where a high sum of money is transferred. The expectation is that in fraudulent transactions the peaks are consistently high and occur frequently in the window of 28 days, while in non-fraudulent transaction the peaks are a bit lower and if peaks are high they occur not as often in the time window as for fraudulent transactions.

To get a better idea of how this looks, examples can be seen in Figure 22 and 23. The first figure displays the rhythm of debit mutations of two different counterparties for the same contract number in the same window. It can be observed that the different counterparties have different moments of transactions, and also the height differs. The second figure displays the rhythm of credit mutations of two different counterparties for the same contract number in the same window. From this it can clearly be observed that when the transactions from one counterparty end, the transactions of the next occur. This figure also displays frequently high peaks, so the expectation is that this contract number in this particular time window is committing some type fraudulent behavior.

The obtained data points per unique counterparty per window are saved in a file, to be later used as input for the outlier detection model.

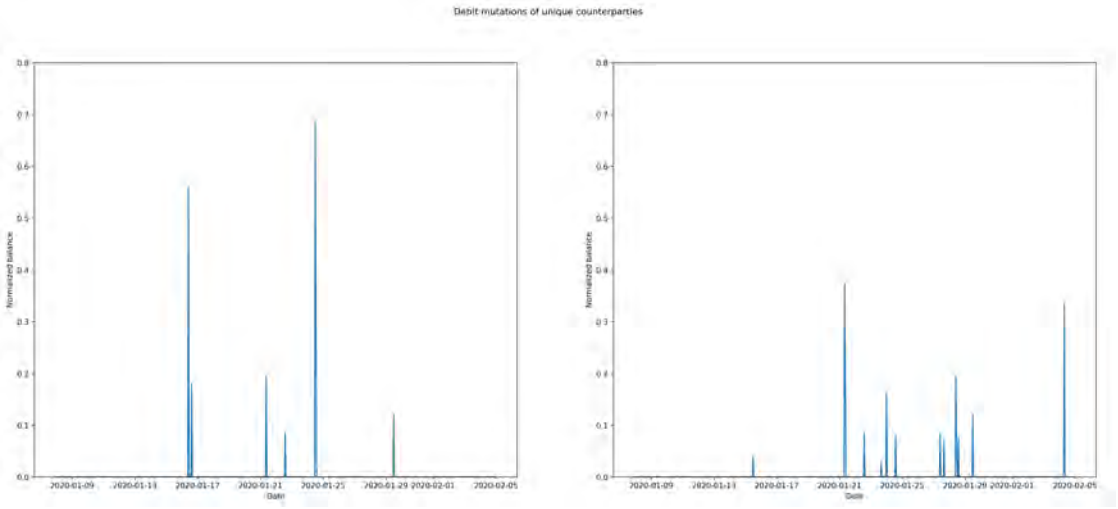

Figure 22: The debit transaction rhythm of two different counterparties for the same contract number in the same window

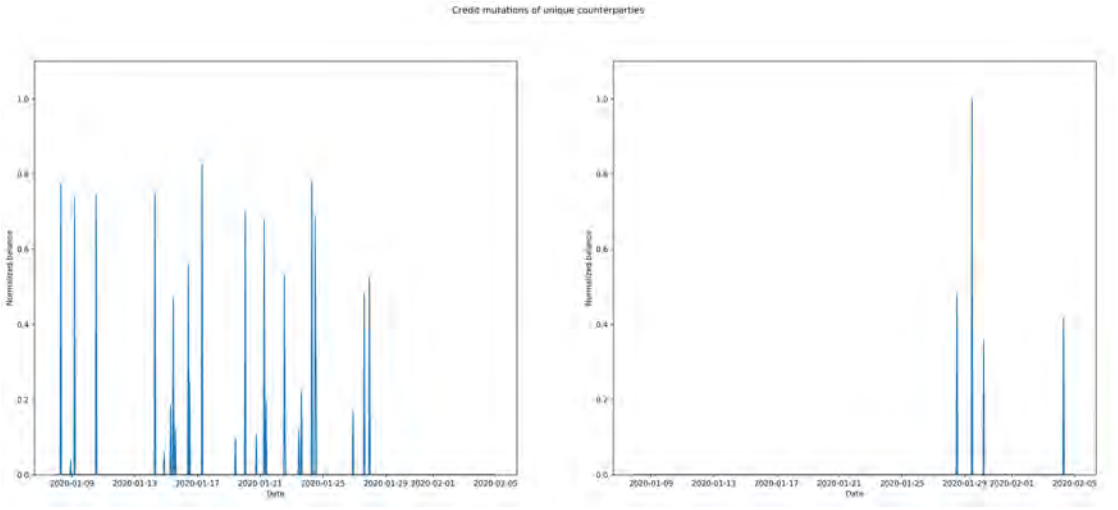

Figure 23: The credit transaction rhythm of two different counterparties for the same contract number in the same window

6.1.4 Cash

The fourth developed feature is that of the rhythm of cash transaction in a 28 day window. The expectation is that for VAT/carousel fraud not necessarily cash transaction are used a lot. But it is still valuable for the model since a lot of cash transactions is not 'normal' behaviour. So, many cash transaction in 28 days can point to fraudulent behaviour, however most likely not VAT/carousel fraud. But it is still very valuable information and therefore this feature is interesting for the model.

The concept of the previously developed feature will also used for the development and implementation of the cash feature. The steps in the pseudo code for the cash feature are as follows: first data points are created for each hour, then the balance is normalized per contract number between zero and one. Third a column is added with indicates cash transactions. Then fill in this

table per data point and set the normalized balance to zero if no cash transaction occurred at that data point. Lastly, apply the sliding window concept, so create windows of 28 days and shift it after 7 days.

Pseudo code cash

1. *Create a data point for every hour, by aggregating the balance*
2. *Normalize the balance per contract number:* $z_i = (x_i - min(x))/(max(x) - min(x))$
3. *Create extra column which indicates cash transaction with 1 and no cash transaction with 0*
4. *If data point = cash transaction, label it with 1*
5. *If data point* $\neq$ *cash transaction, label it with 0*
6. *Where labels equal 0, the normalized balance is set to zero*
7. *Apply the sliding window concept, with length = 28 days and shift every 7 days*

With the use of the pseudo code the cash transaction are displayed as peaks over time, which is exactly what we wanted. The model can combine the balance over time and these cash peaks to determine the volume of the mutation and can learn at which moments in time a cash transaction occurred. The expectation is that a small peak will more likely indicate a transaction with no a large amount of money, while higher peaks will more likely display a transaction where a high sum of cash money is transferred. It could be the case that no cash transactions occur, then the input will be just a flat line, so a value of zero for each data point.

An example of the peaks of cash transactions over time for two different contract numbers in different windows can be seen in Figure 24. From the left figure it can be observed that only two cash transactions occurred in 28 days. As the peaks in this figure are rather low, below 0.4, it is expected that no large sums of cash money are transferred or withdrawn. So based on this figure no obvious fraudulent behavior can be spotted. In contrast, from the right figure it can be observed that a lot of cash transaction occur in 28 days. Also the peaks in this figure are quite high, so it is expected that large sums of cash money are transferred or withdrawn from this contract number in 28 days. Thus, solemnly based on this figure this contract number shows signs of fraudulent behavior in the 28 day window.

The obtained data points for each window are saved in a file, to be later used as input for the outlier detection model.

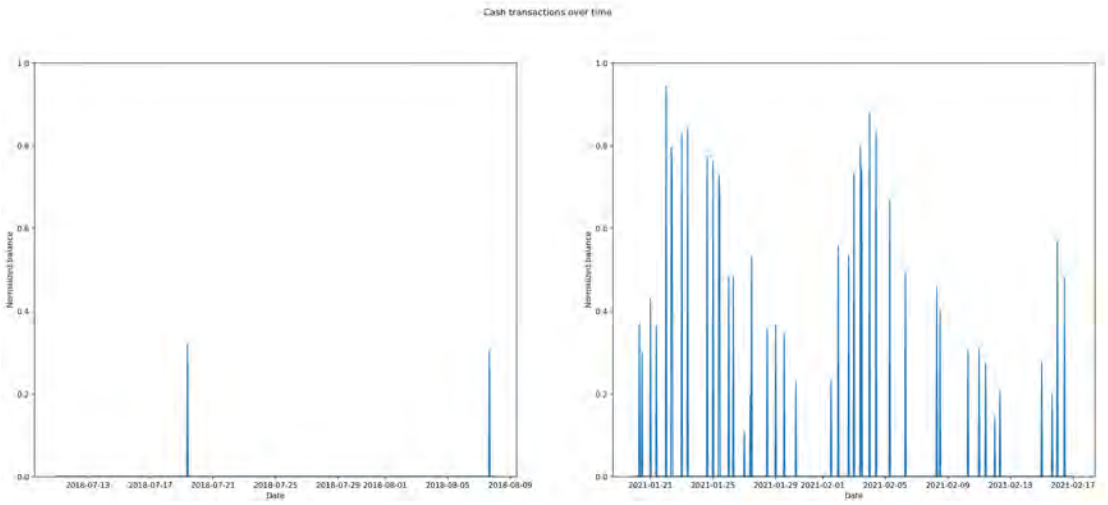

Figure 24: Cash transaction rhythm over time for two different contract numbers in two different windows, each window consists of 28 days

6.1.5 Cross-border transactions

The last developed feature for the outlier detection model is that of the rhythm of cross-border transactions in the 28 day window. The reason for developing a feature regarding cross-border transactions is that it is known that in carousel fraud a cross-border transaction needs to take place. So it is valuable for the detection of VAT/carousel fraud to known the frequency and time points of cross-border transactions. This is especially interesting if these transactions cover large sums of money and occur many times in just 28 days.

The concepts of the previously developed feature will also used for the development and implementation of the cross-border feature. The steps in the pseudo code for the cross-border feature are as follows: first data points are created for each hour, then the balance is normalized per contract number between zero and one. Third a column is added with indicates cross-border transactions. Then fill in this table per data point and set the normalized balance to zero if no cross-border transaction occurred at that data point. Lastly, apply the sliding window concept, so create windows of 28 days and shift it after 7 days.

Pseudo code cross-border

1. Create a data point for every hour, by aggregating the balance
2. Normalize the balance per contract number: $z_i = (x_i - min(x))/(max(x) - min(x))$
3. Create extra column which indicates cross-border transaction with 1 and no cross-border transaction with 0
4. If data point = cross-border transaction, label it with 1
5. If data point $\neq$ *cross-border transaction, label it with 0*
6. Where labels equal 0, the normalized balance is set to zero
7. Apply the sliding window concept, with length = 28 days and shift every 7 days

With the use of the pseudo code the cross-border transaction are displayed as peaks over time, which is exactly what we wanted. The model can combine the balance over time and these cross-border peaks to determine the volume of the mutation and can learn at which moments in time a cross-border transaction occurred. The expectation is that small peaks will display a transaction where not a large sum of money is transferred cross-border, while high peaks will

display a transaction where a large sum of money is transferred. It could be the case that no cross-border transactions occur, then the input will be just a flat line, so a value of zero for each data point.

An example of the rhythm of cross-border transactions over time for two different contract numbers in different windows can be seen in Figure 25. From the left figure it can be observed that only a few cross-border transactions occur in 28 days. The peaks in this figure are rather low, at most 0.2, so it is expected that no large sum of money are transferred or withdrawn from this contract number. Thus based on this figure no obvious fraudulent behavior can be spotted. In contrast, from the left figure it can be observed that a lot of cross-border transactions occur in 28 days. Also the peaks in this figure are quite high, each one above 0.7, so it is expected that large sums of money are transferred or withdrawn from this contract number. Thus, solemnly based on this figure this contract number shows sign of fraudulent behavior in this particular time window.

The obtained data points for each window are saved in a file, to be later used as input for the outlier detection model.

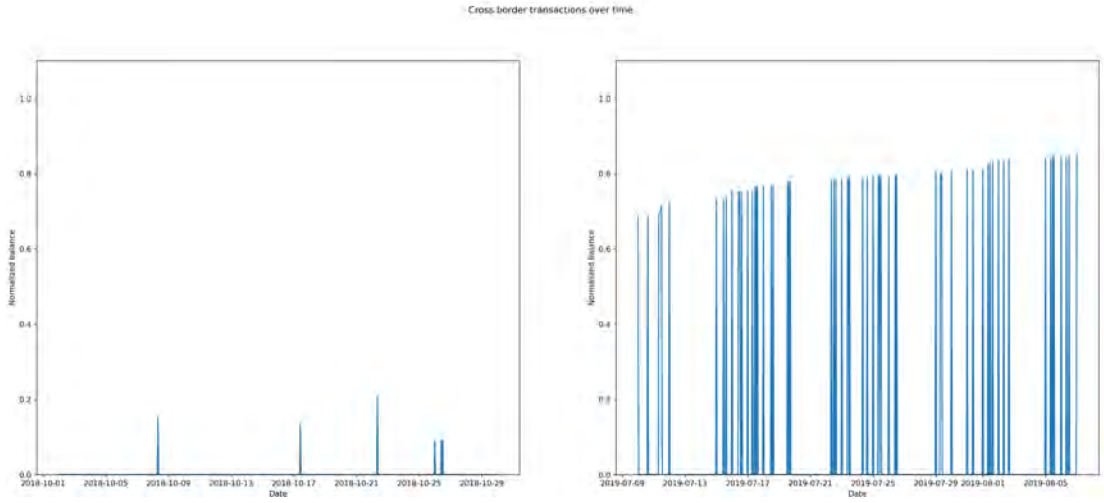

Figure 25: Cross-border transaction rhythm over time for two different contract numbers in two different windows, each window consists of 28 days

6.2 Variational Autoencoder

The outlier detection model consists of a Variational Autoencoder (VAE) with Long-Short Term Memory (LSTM) layers, this combination was shown to be very promising from the literature. Since the VAE can handle short windows of features well, while the LSTM can handle the long term correlation in time series data. Thus combining these two will result in a model that can identify anomalies which span over multiple time windows.

The mathematical details of the VAE and LSTM can be found in Chapter 3. The code developed for this project was inspired by the article from Pereira et al. [92], and an example of this papers implementation can be found here: `https://github.com/RL1993/Thesis_ECG_Example/blob/790c163aaf39a2edc418f61602f88c7dade217d5/ECG_VAE.ipynb`.

The implementation of the VAE is done as follows: first create the encoder part of the model where the input consists of the sample size and number of dimensions of the input. Then the hidden layers for the encoder are created, which are bidirectional LSTM layers. Next the μ and σ are computed using the number of latent dimensions and the hidden layers. Using these the z can be determined. Combine the encoder input, μ, σ and z to create the encoder module. Next the

decoder part is created, the input for this is the number of latent dimensions and the number of dimensions. Again the hidden layers are bidirectional LSTM layers and the output of the decoder is generated. Combining the decoder input and decoder output results in the decoder module. The output of the loss function is determined with the use of the decoder module. Finally, the VAE is created by combining the encoder input and the output of the loss function. Then lastly, the VAE is compiled and fitted on the train data. A visual representation of the VAE architecture can be seen in Figure 26.

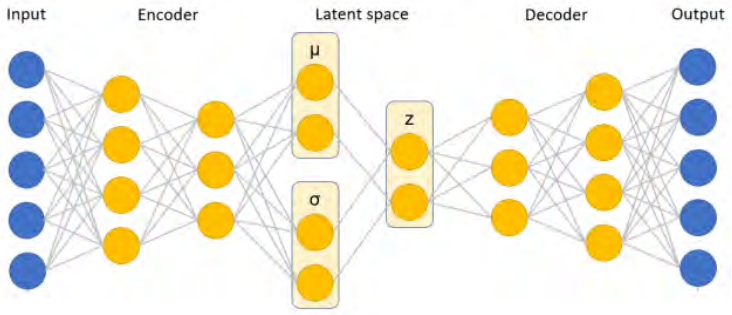

Figure 26: VAE model architecture [63]

In total there are five parameters which need to be fine tuned in the VAE. The fine tuning of these parameters will happen in the next chapter, there a more indepth explanation will be given on these parameters.

6.3 β-Variational Autoencoder

The second outlier detection models consists of a β-Variational Autoencoder (β-VAE) with Long-Short Term Memory (LSTM) layers. From literature it was obtained that the β-VAE could possibly improve the interpretability and explainability of deep learning models, such as the VAE.

The mathematical details of the β-VAE and LSTM can be found in Chapter 3. The implementation of the β-VAE is done in the same manner as for the VAE and has the same architecture. The only difference is that the loss function gets an additional hyperparameter β. The addition of the β results in a total of six parameters which need to be determined for the model. Again the fine tuning of these parameters is described in the next chapter of this report.

7 Experimental setup

In the previous chapter the developed features and architecture of the methods were explained. In this chapter the experimental setup is explained. First, the train and test split is made. Then, the parameters are tuned and the evaluation metric for the methods is presented.

7.1 Train/test split

Before the parameters are tuned and the evaluation metric is described, the data set needs to be split into a train and a test set. The division of the data set was presented in Chapter 5 Table 1. In the data set the contract numbers are unique and therefore the decision was made to make the split based on these. In total, there are 486 contract numbers available. An 80/20 split is used in this project. That translates to 389 contract numbers in the train set and 97 contract numbers in the test set. As was previously stated, the data set is highly imbalanced. To ensure that enough fraud cases are present in the test set, the decision was made to have 10% of the test set be a known fraud case. Thus, 10 contract numbers that contain fraud are put into the test set. These 10 contract numbers were not used for the exploration and development of the features. A disadvantage of making sure that 10% of the test set is fraud, is that the train set has very few fraud cases to train on. However, this would comply with real life because fraud is such a rare phenomenon. The remaining fraud cases are put into the train set. All the non-fraud contract numbers are randomly divided between the train and test set.

The next step is to tune the parameters of the methods.

7.2 Parameter tuning

The tuning of parameters is very important in deep learning methods, since these strongly influence the performance. The VAE and β-VAE have long computation times, especially since LSTM layers are incorporated in these methods. The following parameters are to be tuned for the VAE: number of latent space dimensions, epochs, batch size, number of LSTM layers, and learning rate. For the β-VAE model an additional parameter needs to be tuned, namely the β. Unfortunately, this project did not have enough time and resources to tune all the different parameters in the models. The focus in this project will be on tuning the parameters *number of latent dimensions* and β. The other parameter values were set based on literature, where similar situations occurred.

The implementation of the methods was inspired by the paper by Pereira et al. [92]. This paper used electrocardiogram (ECG) time series data, namely the ECG500. In this data set there are five heart rate classes and the data is highly imbalanced. As is the same in this project, the data is highly imbalanced. The authors used the following values for the parameters. A learning rate of 0.001 to a variant of Adam. The batch size, which indicates after how many samples the model weights are updated, is set to 500 and 1500 epochs. The latent dimensions was fixed at 5. Lastly, 128 bidirectional LSTM layers were used, resulting in a total of 256 units.

This project will implement the same values for the learning rate and number of LSTM layers. A value of 0.001 for the learning rate and 128 bidirectional LSTM layers. For the batch size a different value was chosen. According to literature, a batch size of 32 is a good default value for deep learning models [9]. Therefore, the choice was made to use the value of 32 as batch size, instead of 500 as in the paper by Pereira et al.. The number of epochs will be set to a lower number compared to the paper. The reason for this is that the methods will have a lower computation time when exploring the different *number of latent dimensions*. It was decided to start with a value of 100 epochs. This number will be increased to 300 when the number of latent space dimensions has been determined and the final training of the methods takes place. The reason for only increasing it to 300 and not higher such as in the paper, is that preliminary results have shown that this relatively low number of epochs is sufficient for the loss to stability.

Variational Autoencoder

A total of five VAE methods need to be tuned. Each VAE model has a different combination of features. Investigations will be done to observe which combination performs best in detecting outliers. Every VAE model is extended with another feature. The order of adding these features is based on domain knowledge. Start with just the feature balance over time. From domain knowledge it was obtained that RMoF is a big indicator for fraud, therefore this feature is added next. Then, the feature cross-border is added, since we know that cross-border transactions needs to occur in VAT/carousel fraud. Another possible indicator for VAT/carousel fraud is the relatively low number of counterparties. Thus this feature is added in the fourth model. Lastly, the feature cash is added. The different combinations of features for the models can be seen in Table 8.

Model 1	Balance
Model 2	Balance + RMoF
Model 3	Balance + RMoF + Cross-border
Model 4	Balance + RMoF + Cross-border + Counterparties
Model 5	Balance + RMoF + Cross-border + Counterparties + Cash

Table 8: Combination of features for the VAE models

For the five VAE models only the parameter *number of latent dimensions* needs to be tuned. Due to time limitations, the decision was made to try five different values for this parameter and investigate with the use of an evaluation metric which value performs best. The paper by Pereira et al. used 5 latent dimensions [92]. Therefore, it was decided to start with 5 latent dimensions and we increased this by five until we reached 25 latent dimensions. The reason for not taking a higher number of latent space dimensions is that we want to ensure that the model learns a compressed version of the data. To ensure this, the number of latent space dimensions cannot be too large. This is especially true with a low number of dimensions as input, which is the case when taking model 1, 2 and 3. So, the investigated latent space dimensions values are: 5, 10, 15, 20 and 25. After training, Isolation Forest will be applied to the latent space of the VAE to determine the outlier scores. Because random sampling is involved in the latent space of the VAE, the outlier detection technique Isolation Forest produces slightly different results for each run. Taking this into account, Isolation Forest will be run ten times per parameter value.

The outlier scores are used as input for the evaluation metric. This metric will produce a mean recall score per parameter value and based on this score the value for the number of latent dimensions will be chosen.

β-Variational Autoencoder

The β-VAE needs to tune an additional parameter β. The decision was made to apply the β-VAE to the VAE feature combination with the highest mean recall score. To investigate the effect of the parameter β, all the other parameter values will be kept the same. According to the literature, an optimal value for β is yet to be found. Therefore, it was decided to try the following values for the β: 2, 5 and 10. Again, the evaluation metric is applied to obtain recall scores and determine which β is most appropriate.

In the next section, a description of the evaluation metrics that was applied to determine the values of the *number of latent space dimensions* and the β is given. This metric will also be used when evaluating the methods on the test data.

7.3 Evaluation metric

The evaluation metric will be applied in two stages in this project. The first one being to determine the best value for the parameter *number of latent dimensions* and the β, which will be done based on the train data. After the best value has been chosen, the model will be trained on the best parameter values with a higher number of epochs. Then, the evaluation metric is applied again, only this time on the test data. This is to investigate how well the methods can detect outliers on new data.

For this project the decision was made to use the recall as the evaluation metric. The reason for this is that it is often the goal to improve the recall for imbalanced data sets and this project deals with imbalanced data [50]. It is possible to calculate the recall on the data since we had access to some labeled data. It must be taken into account that there could be more fraud cases in the data, only these are unknown to us. So, the recall gives an indication on the performance of the methods. The recall score will first be applied to determine the best parameter values, using the train data. Then the recall score will be used to determine the performance of the methods on the test data.

For every parameter setting the models have ten runs on Isolation Forest. Each run produces outlier scores and these are used to obtain the recall score, resulting in ten recall scores per parameter setting. Also, per parameter setting the mean recall score and standard deviation of the ten runs is calculated. A high recall would indicate that more known fraudulent contract numbers were found. Thus, the goal is to obtain the highest possible recall score.

The recall can be defined as a metric that quantifies the number of correct positive predictions made out of all positive predictions that could have been made [15]. So, the recall displays the coverage of the positive class. The recall is calculated in this project as follows: the number of unique fraudulent contract numbers in the top x is divided by the total number of fraudulent contract numbers in the set, where x is manually chosen. So, first the outlier scores per run for Isolation Forest are sorted from highest to lowest. Then, the top unique contract numbers are taken and it is determined how many of these are labeled as fraudulent. This obtained amount is then divided by the total number of fraudulent contract numbers. By taking a fixed number of contract numbers, the outcomes for the different parameter values can be compared accurately. When taking a fixed window size it is possible that some contract number have many duplicates in the top x and thus few unique contract numbers are found. This will have a negative influence on the recall scores.

For the train and test set a different number of highest ranked outlier contract numbers are taken. The train set has a total of 389 contract numbers. Then the top 30, 60 and 80 unique contract numbers are taken per run. This set contains 25 contract numbers which were labeled as fraudulent. The test set has a total of 97 contract numbers, and here the top 5, 10 and 20 unique contract number per run are taken. The number of known fraudulent contract numbers in the test set equals 10. The calculation of the recall for the train and test set can be formalized as follows:

$$\text{recall}_{train} = \frac{\text{number of unique fraudulent contract numbers in top 30, 60 or 80}}{\text{total number of fraudulent contract numbers in train set } (= 25)}$$

$$\text{recall}_{test} = \frac{\text{number of unique fraudulent contract numbers in top 5, 10 or 20}}{\text{total number of fraudulent contract numbers in test set } (= 10)}$$

Per parameter value for each model a total of ten recall scores are obtained. With the use of these recall scores also the mean recall per parameter value and the standard deviation can be determined. All this information will be combined to determine per model the best value for the *number of latent dimensions* and β, and to investigate the performance on the test set. This all

will be done on the top 80 for the train set and the top 20 for the test set. The reason for this is that these tops contain roughly 20% of the contract numbers in the data sets.

The recall scores on the train and test data can be found in Chapter 8.

7.4 Statistical test

To investigate if there is a significant difference between the models on the test data, a statistical test will be executed. The statistical test will be performed on the ten recall scores per model obtained from the top 20 unique contract number, to investigate if there is a significant difference between the models. Since the data is not normally distributed the *t-test* could not be used. Therefore, the decision was made to perform the *Kolmogorov-Smirnov (KS) test* and then the two sampled version. This test can efficiently determine if two samples are significantly different from one another [42]. The null hypothesis in this test is defined as: both samples come from a population with the same distribution [126]. This null hypothesis is rejected if the p-value is below $\alpha = 0.05$. Then there is sufficient evidence that the two sample data sets do not come from the same distribution. [113] All models will be compared with one another.

The results of the statistical test can be found in Chapter 8.

7.5 Training

In the previous section the evaluation metric was explained, which was used to investigate the different options for the parameters. The results of these are presented in Chapter 8.2. Based on these outcomes, the best values for the parameter *number of latent dimension* and β were chosen. The next step in the process would be to fit the models for a longer duration on the train data. So, the epochs are increased to 300. After training, the test set was taken and encoded on to the latent space. On these windows, Isolation Forest was executed ten times and per run the outlier scores were obtained. Then, the evaluation metric is applied to determine how well the methods perform on the test data. These results are presented in the next chapter.

The exact parameter settings per VAE model and β-VAE can be found in the Appendix.

8 Results

This chapter presents the results obtain in this project. First, the results of the Rapid Movement of Funds analysis are shown and discussed. From this analysis the best value for the parameters of the Savitzky-Golay filter will be determined. Second, the results of tuning the parameters is presented. Here it is determined which *number of latent dimensions* and β values are the best for the different feature combinations. Third, the outcome of Isolation Forest on the different feature combinations of the VAE models using the test data is presented. The evaluation metric described in the previous chapter will be used to determine the performance of the methods on the test data. In this section also a paragraph will be dedicated to a fraud investigator perspective, to review if the results make sense and if perhaps new cases or patterns of fraudulent behavior are discovered. Lastly, the β-VAE results are displayed and discussed to investigate if this model improves the explainability and interpretability of the VAE.

8.1 Rapid Movement of Funds

To decide which polynomial value to use for the Savitzky-Golay filter a Receiver Operator Characteristic (ROC) curve was made. But first the pseudo code is presented on how the area between the normalized balance and smoothed line is calculated. The steps are as follows: first data points are created for each hour, then the balance is normalized per contract number between zero and one. Third the Savitzky-Golay filter is applied and finally the area between the normalized balance line and the smoothing filter is calculated per month. These areas are needed to classify RMoF. The expectation is that a high area will indicate a month for a contract number were RMoF took place, while a low area indicates that no RMoF took place.

Pseudo code RMoF

1. *Create a data point for every hour, by aggregating the balance*
2. *Normalize the balance per contract number:* $z_i = (x_i - min(x))/(max(x) - min(x))$
3. *Apply the Savitzky-Golay filter:* $Y_i = \sum_{i=-(m-1)/2}^{i=(m-1)/2} C_i y_{j+1}$, *with* $m = 97$ *and* $C_i = 1$ *or* 2
4. *Calculate the area per month:* $\sum \mid$ *Savitzky-Golay - normalized balance* $\mid$

The obtained areas are then sorted from highest to lowest for both polynomials. To asses how many months with a high area are indeed labelled as RMoF, the top 100 months are checked. For the top 5, 10, 50 and 100 months the number of found RMoF cases are determined. The results of this can be seen in Table 9 for polynomial 1 and in Table 10 for polynomial 2. For example, in Table 9 in the top 5 a total of 2 RMoF cases and 3 not RMoF cases were found. These numbers give a slight preference for polynomial 2, but no definite conclusion can be drawn yet.

Top	**Number of RMoF cases**
5	2
10	4
50	19
100	34

Table 9: Number of RMoF cases found in the top 100 months for polynomial 1

Top	Number of RMoF cases
5	4
10	6
50	21
100	38

Table 10: Number of RMoF cases found in the top 100 months for polynomial 2

The next step is to plot the ROC curve for both polynomials. The ROC curve shows the diagnostic ability of a classifier. In this plot the true positive rate is plotted against the false positive rate. A curve that comes closer to the top-left corner indicates a better performance. To calculate the true positives and false positives rate the gold labels and probability prediction percentage are needed for both polynomials. The gold labels are available, since it is known which contract numbers have RMoF. The probability prediction percentages are obtained as follows, the area for both polynomials is split into buckets. Each bucket has the length of 5% of the highest area, resulting in 21 buckets for both polynomials. Then for each bucket the percentage of RMoF is calculated, indicating how many months in the bucket are indeed classified as RMoF. The expectation is that with a higher bucket number, which indicates a higher area, the percentage of RMoF cases increase. The probability prediction percentages are then obtained using the percentage of RMoF cases in each bucket. That is, each area falls within a certain bucket and the calculated percentage is the RMoF probability for a month with such an area value.

A disadvantage of the applied probability prediction model is that it is a rather simplified model. As a consequence, it can negatively impact the choice and optimal value for the parameter *degree of the polynomial*, especially in more complex models. This project will use the RMoF feature as input for a Variational Autoencoder (VAE), which can be defined as a complex model. Therefore, it good to keep this disadvantage in mind.

The ROC curves for both polynomial values can be seen in Figure 27. In this figure the blue line indicates the ROC curve for polynomial 1 and the orange line indicates the ROC curve for polynomial 2. The orange line comes closer to the top-left corner, compared to the blue line, thus indicating a better performance. Based on this figure it is concluded that the best degree of the polynomial would be the value of 2. So the parameter values for the Savitzky-Golay filter are a *windowsize* of 97 and a *degree of the polynomial* of 2.

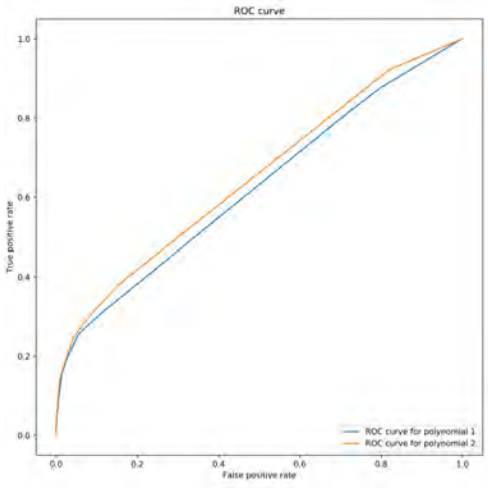

Figure 27: The ROC curves for both polynomials

The four highest areas per month for windowsize 97 and polynomial 2 are presented in Figure 28 and 29. The blue line indicates the normalized balance over time for that month, and the orange line indicates the Savitzky-Golay smoothing filter line. From these figure a clear RMoF pattern can be observed. The left figure in Figure 28 is the contract number with the highest area for a month, this contract number was not labeled as RMoF by the bank. After a closer inspection from an investigator it was determined that indeed RMoF occurred during this month. The RMoF occurred in the form of cash, so cash money was rapidly transferred and withdrawn. For the top 20 contract numbers an accuracy score of above 80% was obtained. So, it can be concluded that the Savitzky-Golay filter helps in detecting RMoF.

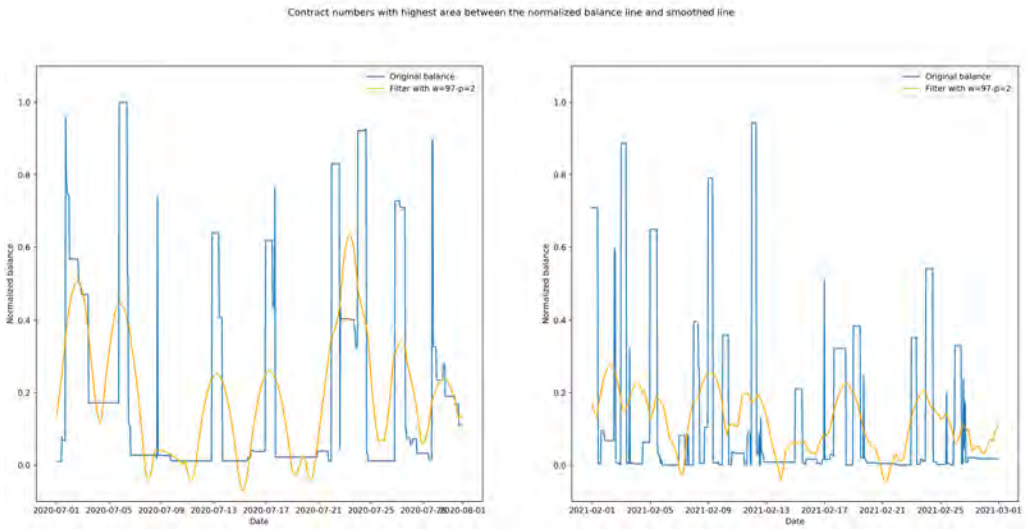

Figure 28: Monthly normalized balance over time for contract numbers with highest area between the smoothing filter (orange line) and the normalized balance line (blue)

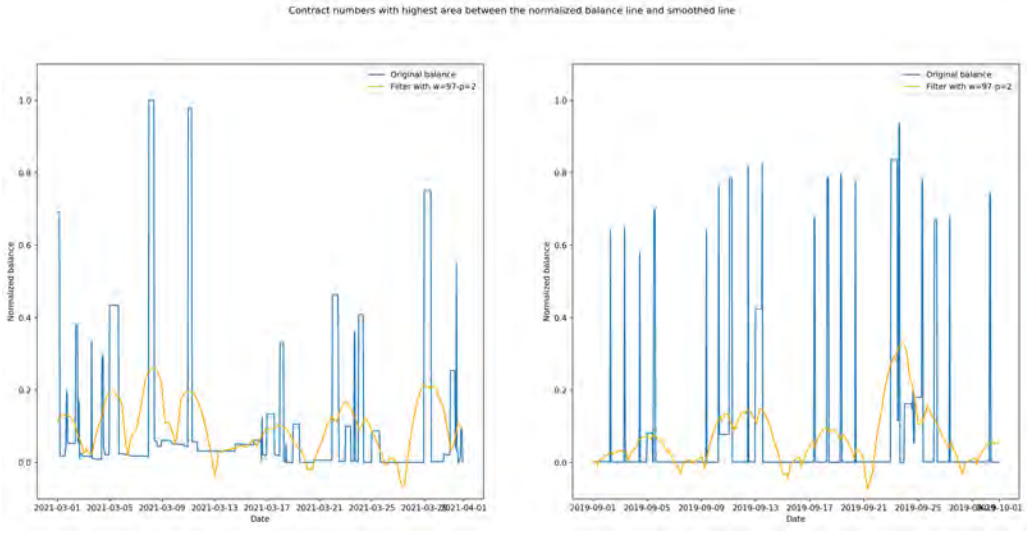

Figure 29: Monthly normalized balance over time for contract numbers with highest area between the smoothing filter (orange line) and the normalized balance line (blue)

8.2 Parameter tuning

In this section the results of tuning the parameters *number of latent dimensions* and β are presented. For the fine tuning of the parameters the train set was used. As was explained in Chapter 7, per run a recall score is calculated. The mean and standard deviation are then determined based on the ten recall scores per parameter setting, per feature combination. The information will be visualized in a figure. The blue dot in the figures indicates the mean recall score and the blue bar the standard deviation. The grey crosses mark the ten recall scores per latent dimension value.

The obtained recall scores in the top 80 unique contract numbers will be used to determine the best parameter value per feature combination. In the top 80 the ten recall scores, mean recall score and standard deviation will be compared between the different latent dimension values to choose the best option.

One feature

The results of the recall can be seen in Figure 30. From this figure it can be observed that the recall scores increase when increasing the top unique contract numbers, as expected. In the right figure it can be observed that the mean recall for the latent dimensions 5, 10, 15 and 25 has almost the same value. However, the difference between these latent dimensions can be found in the standard deviation. Latent dimension 25 has the largest standard deviation. This implies that the recall scores for latent dimension 25 can vary a lot more compared to recall scores for latent dimension 5, 10, and 15. The value 20 obtains the highest mean recall score of 0.29. Also, the spread for this latent dimension is small which means that the recall scores do no deviate much from the mean. From this figure it is concluded that in case of one feature the value of 20 for the number of latent dimension would be optimal.

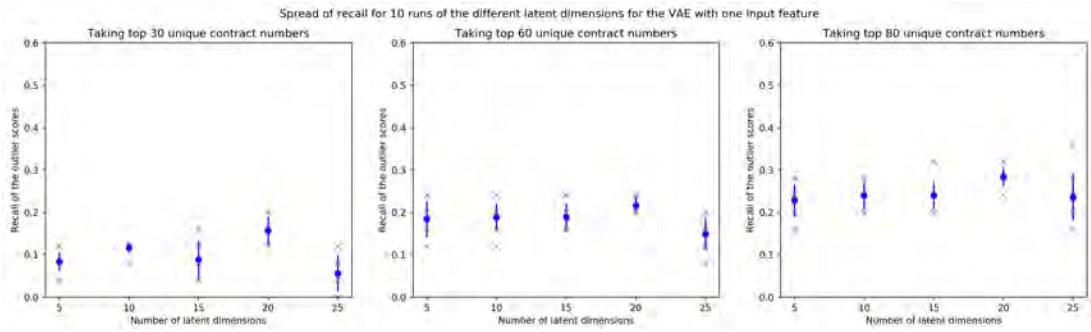

Figure 30: Recall on outlier scores for different latent space dimension values for VAE with one feature, taking top 30, 60, 80 unique contract numbers

	Top 30	Top 60	Top 80
5 latent dimensions	0.08	0.18	0.23
10 latent dimensions	0.12	0.19	0.24
15 latent dimensions	0.09	0.19	0.24
20 latent dimensions	0.16	0.22	0.29
25 latent dimensions	0.06	0.15	0.24

Table 11: Mean recall scores over ten runs, for VAE with one feature

Two features

The outcome of the recall scores for two features is shown in Figure 31. As expected, the recall increases when taking more unique contract numbers. As was the same with one feature, the recall scores are low when taking the top 30 unique contract numbers. In the right figure, it can be seen that a value of 5 outputs the lowest mean score of 0.21. The latent dimensions 15 and 20 follow and have scores around 0.30. The values 10 and 25 have the highest mean recall scores observed so far, of respectively 0.33 and 0.38. From the figure can be observed that latent dimension 10 has higher standard deviation compared to latent dimension 25. Combining the information that the value of 25 produces the highest mean recall score and has a relatively low standard deviation, the decision was made to used this value for the number of latent dimensions in the two feature VAE model.

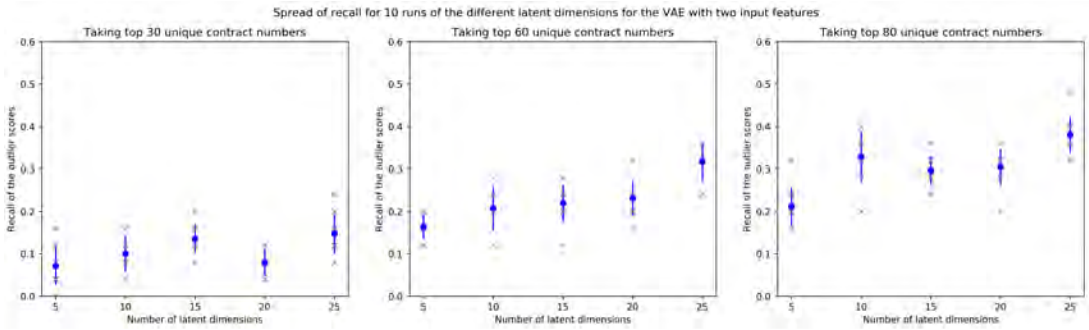

Figure 31: Recall on outlier scores for different latent space dimension values for VAE with two features, taking top 30, 60, 80 unique contract numbers

	Top 30	Top 60	Top 80
5 latent dimensions	0.07	0.16	0.21
10 latent dimensions	0.10	0.21	0.33
15 latent dimensions	0.14	0.22	0.30
20 latent dimensions	0.08	0.23	0.30
25 latent dimensions	0.15	0.32	0.38

Table 12: Mean recall scores over ten runs, for VAE with two features

Three features

The recall scores for three features is presented in Figure 32. The right figure displays the highest observed mean recall score so far in the top 80. This occurs with 15 latent dimension and has a score of 0.43. The values 10 and 25 also perform rather well with scores of 0.33 and 0.36. Again, latent dimension 5 scores the lowest. However, the smallest standard deviation occurs at latent dimension 5. While value 10 and 25 have the highest standard deviations. Based on the figure, the decision was made to use a value of 15 for the number of latent space dimensions for the VAE model with three features, even though the standard deviation is relatively high.

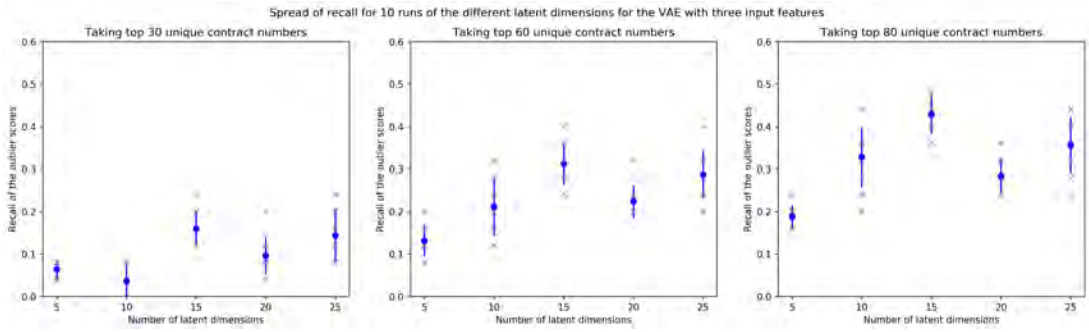

Figure 32: Recall on outlier scores for different latent space dimension values for VAE with three features, taking top 30, 60, 80 unique contract numbers

	Top 30	Top 60	Top 80
5 latent dimensions	0.06	0.13	0.19
10 latent dimensions	0.04	0.21	0.33
15 latent dimensions	0.16	0.31	0.43
20 latent dimensions	0.10	0.22	0.28
25 latent dimensions	0.14	0.29	0.36

Table 13: Mean recall scores over ten runs, for VAE with three features

Four features

The outcome of the recall scores for four features is shown in Figure 33. The scores obtained for the top 80 are much lower compared to the previous feature combinations. What can be seen is that for the top 30 the lowest recall scores so far are obtained. For the top 30, only latent dimension 10 has not a score of zero. The scores slightly increase when taking more unique contract numbers, but are still low. The latent dimensions 15, 20 and 25 all have scores between 0.07 and 0.08. The value of 25 has a slightly bigger standard deviation compared to 15 and 20. The highest mean recall scores is achieved at latent dimension 10, with a score of 0.21. Also, the standard deviation is smallest at this latent dimension value. Combining the information that the value of 10 produces the highest mean recall score and low standard deviation, the decision was made to use this value for the number of latent dimensions in the four feature VAE model.

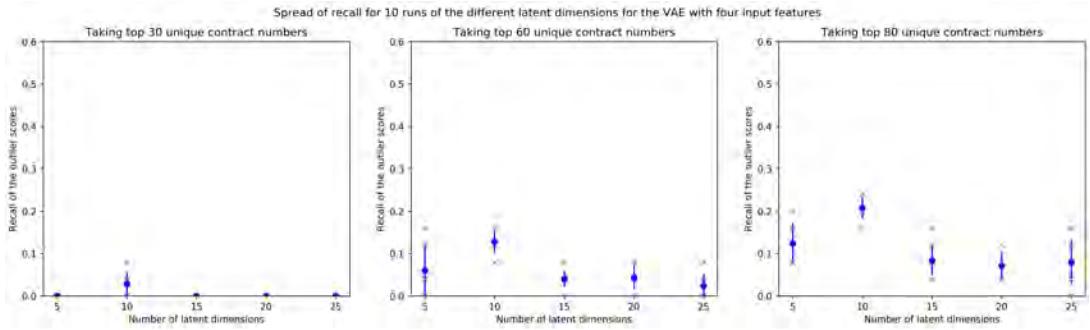

Figure 33: Recall on outlier scores for different latent space dimension values for VAE with four features, taking top 30, 60, 80 unique contract numbers

	Top 30	Top 60	Top 80
5 latent dimensions	0.00	0.06	0.12
10 latent dimensions	0.03	0.13	0.21
15 latent dimensions	0.00	0.04	0.08
20 latent dimensions	0.00	0.04	0.07
25 latent dimensions	0.00	0.02	0.08

Table 14: Mean recall scores over ten runs, for VAE with four features (13 dimensions)

As was observed in the above figure, the recall scores drop significantly compared to the three feature model. A possible explanation for this would be that the dimensions increase from 3 to 13. This is because the top 5 debit and top 5 credit counterparties are taken as input. It is also investigated what happens if only the top 1 debit and top 1 credit counterparty are given as features to the model, together with the features balance, RMoF and cross-border. This would result in 5 dimensions.

The recall scores for this can be found in Figure 34. From this figure can be seen that the obtained recall scores are higher compared to the four feature model with 13 dimensions. In the top 30 unique contract numbers no recall scores of zero are obtained. For the top 80, the latent dimensions 20 and 25 score the lowest recall of 0.16 and 0.17, and have almost an equal standard deviation. The other three values, 5, 10 and 15, have scores very close to another. Namely, of 0.25, 0.27 and 0.27. It can be observed that the value 5 and 15 have a smaller spread compared to value 10. Given the information that latent dimension 15 has the highest recall and also a relatively small spread, the decision was made to this value for the number of latent dimensions in the four feature model with 5 dimensions.

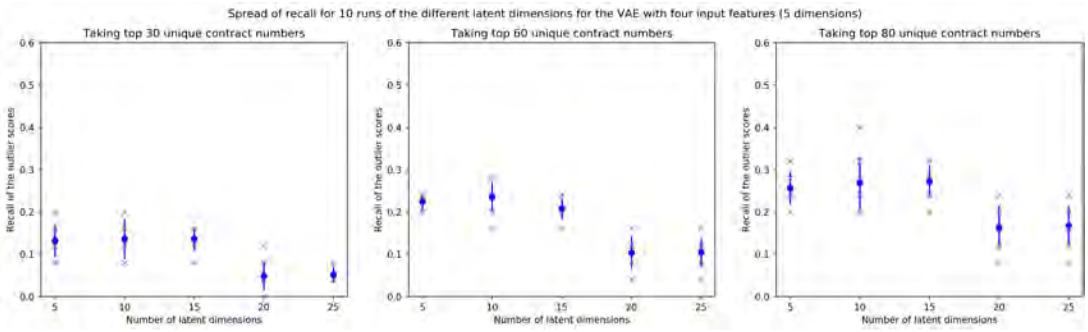

Figure 34: Recall on outlier scores for different latent space dimension values for VAE with four features (5 dimensions), taking top 30, 60, 80 unique contract numbers

	Top 30	Top 60	Top 80
5 latent dimensions	0.13	0.22	0.26
10 latent dimensions	0.14	0.24	0.27
15 latent dimensions	0.14	0.21	0.27
20 latent dimensions	0.05	0.10	0.16
25 latent dimensions	0.05	0.10	0.17

Table 15: Mean recall scores over ten runs, for VAE with four features (5 dimensions)

Five features

The results of the recall can be seen in Figure 35. From this figure it can be observed that the recall scores for the top 30 are extremely low, for most latent dimensions a score of zero is obtained. When taking more unique contract numbers these scores increase, but are still rather low. As was the same with four features (and 13 dimensions). The highest score is obtained for a value of 5, namely a score of 0.26. This is remarkable since the value 5 performed rather bad in all the previous feature combinations. Also the standard deviation is very small for latent dimension 5. This entails that the recall scores for all ten runs are very close to the mean recall score. The values 10, 20 and 25 are between 0.11 and 0.16. The value 15 performs poorly in case of five features, obtaining a recall score of 0.04. All four values have a relatively large standard deviation. A possible explanation for these low recall scores is that with five features the model has as input 14 dimensions, then 25 latent dimensions is still too low. It could be the case that when taking a higher value for this parameter, the recall increases significantly. From this figure it can be concluded that a value of 5 for the number of latent dimensions is the best value when working with five features.

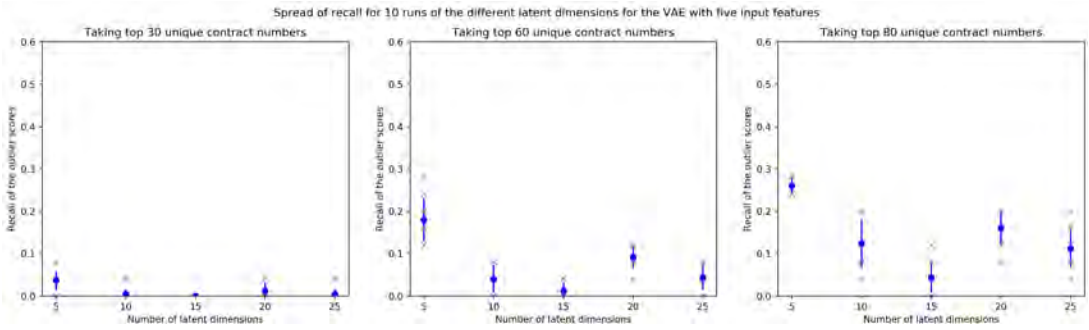

Figure 35: Recall on outlier scores for different latent space dimension values for VAE with five features, taking top 30, 60, 80 unique contract numbers

	Top 30	Top 60	Top 80
5 latent dimensions	0.04	0.18	0.26
10 latent dimensions	0.00	0.04	0.12
15 latent dimensions	0.00	0.01	0.04
20 latent dimensions	0.01	0.09	0.16
25 latent dimensions	0.00	0.04	0.11

Table 16: Mean recall scores over ten runs, for VAE with five features

β-VAE

As stated in Chapter 7, the decision was made to apply the β-VAE to the VAE feature combination with the highest mean recall score. From the previous section it was obtained that the model with the highest mean recall score is the VAE model with three features. So the feature combination consisting of balance, RMoF and cross-border. To investigate the effect of the parameter β, all the other parameter values will be kept the same. It was decided to try the following values for the β: 2, 5 and 10.

The results of the recall scores for the β-VAE can be seen in Figure 36. It can be observed that the mean recall scores for the values 2 and 5 in the top 80 unique contract numbers does not deviated much from the recall scores obtained in the three feature VAE model. The value of 10 for the β displays much lower performance compared to the other two values. Therefore, this value

was not considered as the best option for the parameter β. The standard deviation for β equal to 2 is small. Which means that most recall scores are around the mean. For this β value the mean is slightly lower than the mean recall for the VAE three feature model, but the standard deviation is much smaller compared to the standard deviation of the VAE three feature model. This can be seen as an advantage, since the results do not deviate much and still produce relatively high recall scores. The standard deviation for the value of 5 for β is larger, which implies that the recall scores can vary more from the mean recall score. To summarize, the value of 5 has a higher mean recall score but larger standard deviation, while the value of 2 has a lower mean recall score but smaller standard deviation. Since the recall scores for the values 2 and 5 are so close, the decision was made to explore both values with a higher epoch setting and on the test data.

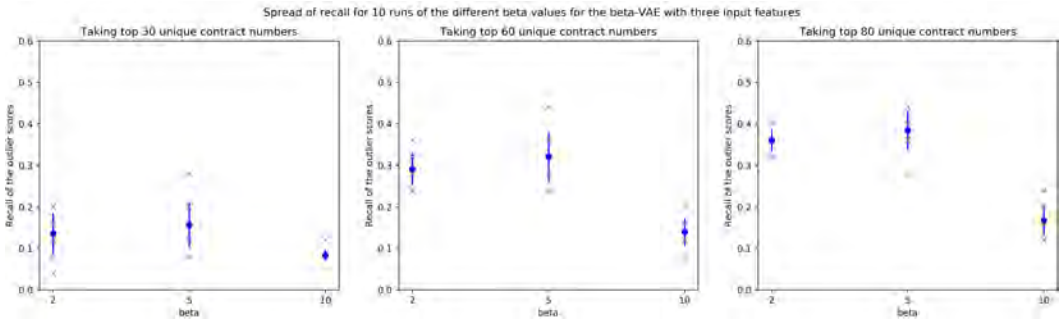

Figure 36: Recall on outlier scores for different β's values for β-VAE with three features, taking top 30, 60, 80 unique contract numbers

	Top 30	Top 60	Top 80
$\beta = 2$	0.14	0.29	0.36
$\beta = 5$	0.16	0.32	0.38
$\beta = 10$	0.08	0.14	0.17

Table 17: Mean recall scores over ten runs, for β-VAE with three features

8.3 Outlier detection

This section presents the outlier detection results for the different feature combination in the VAE model. A total of six different combinations were investigated. The recall formula for the test set, which was presented in Chapter 7.3, will be used to investigate how well the models perform on new data. Per feature combination ten recall scores are obtained, and from these ten scores the mean and standard deviation will be calculated.

Next to the recall scores, the three windows with the highest outlier scores, the lowest outlier scores and the median outlier scores are presented per feature combination. This is to investigate which windows are labeled as likely outliers and which are not. The median outlier scores are the windows which lie in the middle of the highest and lowest scores. The reason for only showing the top 3 is that otherwise the reader will be overwhelmed with figures. Based on these figures, in combination with the calculated recall scores, it can be stated whether the outliers make sense from a data science perspective.

In the last subsection the contract numbers with the highest outlier scores for all feature combinations are discussed with fraud investigators. This is to determine if the results make sense from there perspective and if new fraud cases or patterns can be found using an unsupervised machine learning model.

8.3.1 Analysed with data science perspective

One feature

The results of the recall scores on the test data can be seen in Figure 37. From the figure it can be observed that the recall score increases when taking more unique contract numbers, as expected. The figure shows that a VAE with only one feature, which is balance over time, obtains a mean recall score of 0.34 when taking the top 20 unique contract numbers. This recall score is an improvement compared to the recall score on the train data. There a mean recall score of 0.29 was obtained for the top 80 unique contract numbers. Although a different number for the top was taken, both are around 20% of the data sets. So, a slight increase in terms of the mean recall score has taken place. The standard deviation is not that large and deviates between 0.4 and 0.3. When combining this information a fairly good result is obtained when only taking one feature as input for the VAE model. Next the three windows with the highest, median and lowest outlier scores are presented.

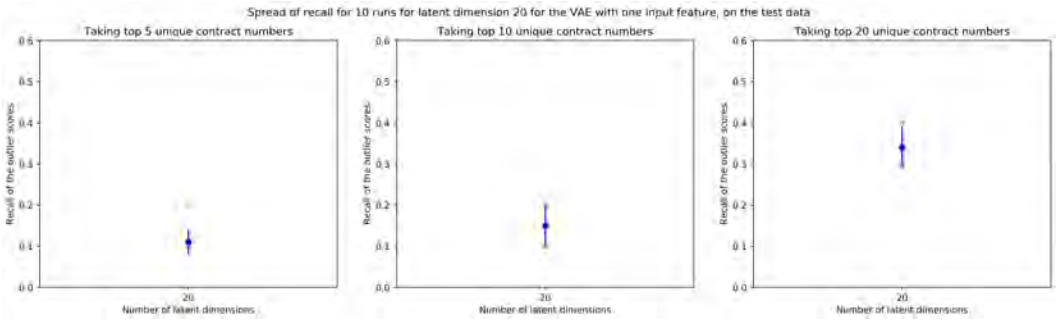

Figure 37: Recall on outlier scores for latent dimension 20 for VAE with one feature on the test data, taking top 5, 10 and 20 unique contract numbers

	Top 5	Top 10	Top 20
20 latent dimensions	0.11	0.15	0.34

Table 18: Mean recall scores over ten runs on test data, for VAE with one feature

The three windows with the highest outlier score are shown in Figure 38. The order of the windows is from highest score to lowest score in the top three. In this figure only the balance over time is displayed, since that is the only feature of the model. On the x-axis no values are displayed due to confidentially, but the length of the x-axis indicates 28 days. From the figure it can observed that the top two windows go from one to zero or the other way around. Only the third window is an exception, there the balance is around 0.5 and then drops. The pattern what can be seen in the top 3 is that the balance was constant and then drops or increases with a significant amount. It can be argued from a data science perspective that these are indeed outliers, because suddenly a large increase or decrease occurs which is not 'normal' behavior. However, it is not the pattern which indicates VAT/carousel fraud. For this type of fraud large peaks occur regularly in a time frame of 28 days, and in these cases such peaks only occur one or twice.

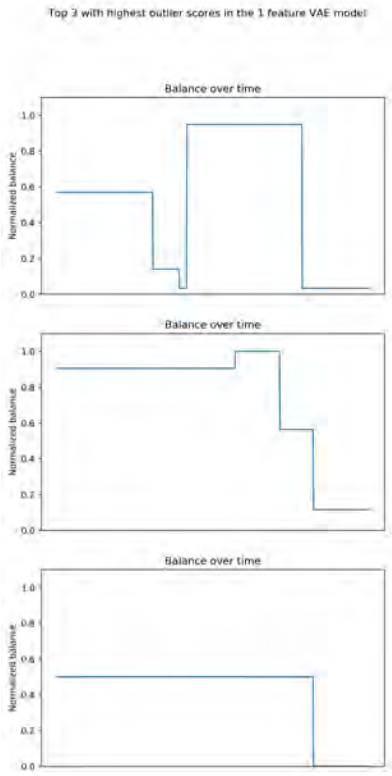

Figure 38: Three windows with the highest outlier score, for one feature combination VAE model

The top three windows with a median outlier score were also determined. The expectation was that no sudden peaks or drops in the balance would be observed in these windows. This was indeed true, also not a lot of activity is occurring in these three windows. The balance almost remains constant or very slowly decreases. Because this figure does not add a lot of value it was decided to display this figure in the Appendix.

Next, the three windows with the lowest outlier scores are determined. This is presented in Figure 39. What can be noticed from the figure is that there is no change or activity in the balance over time. It is just a flat line. This corresponds with the expectation, because the lowest mean outlier scores would indicate no fraudulent behavior. Windows where no activity is happening definitely indicate no fraudulent behavior.

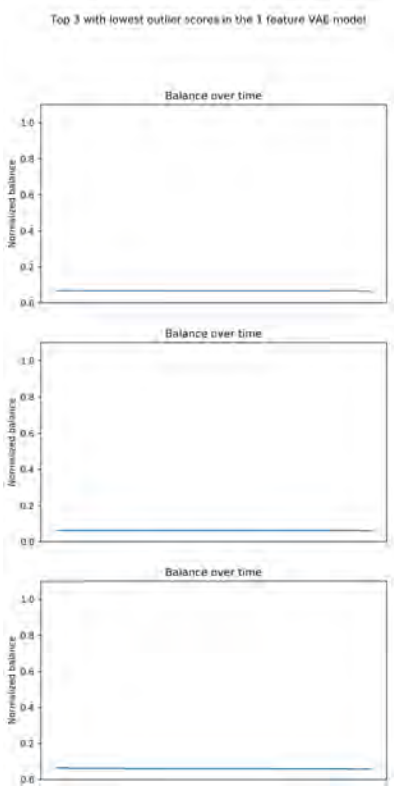

Figure 39: Three windows with the lowest outlier score, for one feature combination VAE model

Two features

The second VAE model which was investigated was the two feature combination, containing balance over time and Rapid Movement of Funds (RMoF). The outcome of the recall scores for the two features combination is shown in Figure 40. In this figure the recall scores are based on the test data. From the figure it can be observed that the recall scores increase when taking more unique contract numbers, as expected. Furthermore, the mean recall score equals 0.23 and has a large standard deviation. The values of the ten runs deviate between 0.50 and 0.10. The mean recall score has decreased compared to the mean recall score on the train data. There a score of 0.38 was obtained. Also, the mean recall score has decreased compared to the one feature VAE model. So, comparing the recall scores of the one and two feature VAE models would indicate that the expansion of the RMoF feature does not result in a higher fraud detection number.

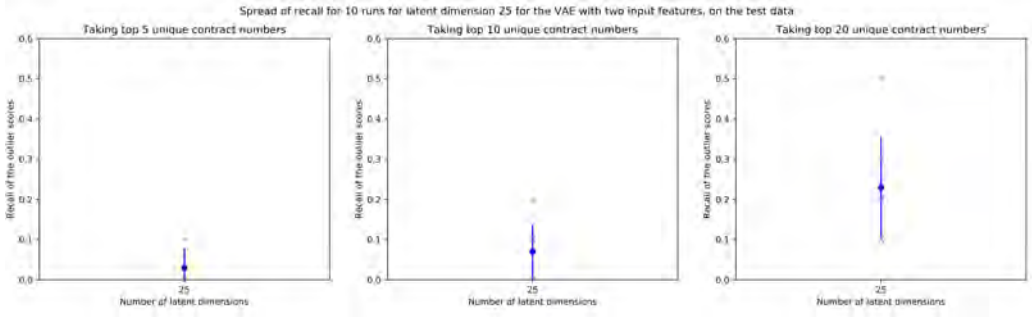

Figure 40: Recall on outlier scores for latent dimension 25 for VAE with two features on the test data, taking top 5, 10 and 20 unique contract numbers

	Top 5	Top 10	Top 20
25 latent dimensions	0.03	0.07	0.23

Table 19: Mean recall scores over ten runs on test data, for VAE with two features

Next to the recall score, the three windows with the highest outlier score are determined and presented in Figure 41. The VAE has as input two features, therefore the figure displays the rhythm of both these features for the windows. The time frame on the x-axis is still 28 days. From the figure can be observed that the top three windows all display a balance rhythm that goes from zero to one or vice versa, and during this drop also RMoF shows a peak. However, the RMoF peak is not necessarily as high as one would expect in a fraud case. Again from a data science perspective, this is definitely an outlier because a huge drop takes place. But it is not quite the rhythms we want to find. For VAT/carousel fraud regularly high peaks are observed in 28 days. Which is not the case in these three windows.

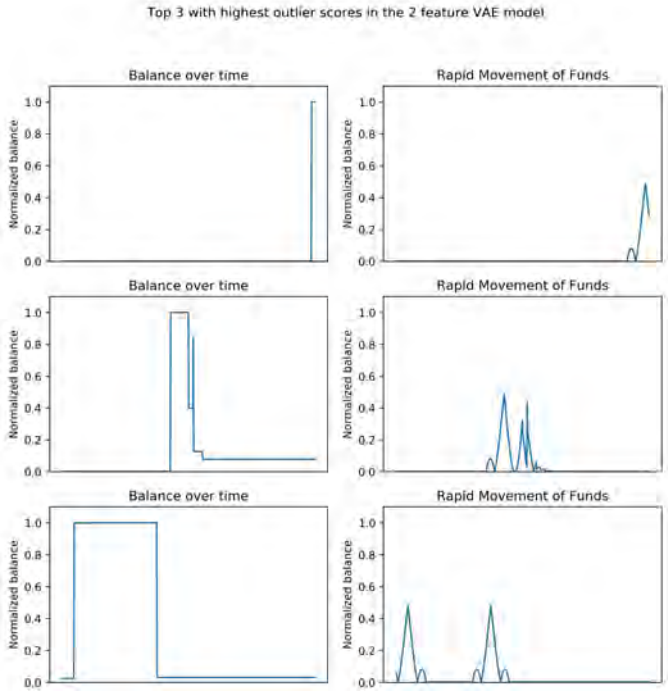

Figure 41: Three windows with the highest outlier score, for two feature combination VAE model

The three windows with median outlier scores were also determined. Compared to the one feature model, a bit more activity is happening. In the figure it is visible that the balance changes very slightly over time and the RMoF displays little peaks. The outcome is conform the expectation and can be seen in the Appendix.

Lastly, the three windows with the lowest outlier scores are shown in Figure 45. The expectation is that the balance is constant, so no transactions occur and as a consequence RMoF will give a flat line. The figure supports this expectation, since the balance remains constant around a value of 0.20. So there is no activity in the contract number for the 28 days window.

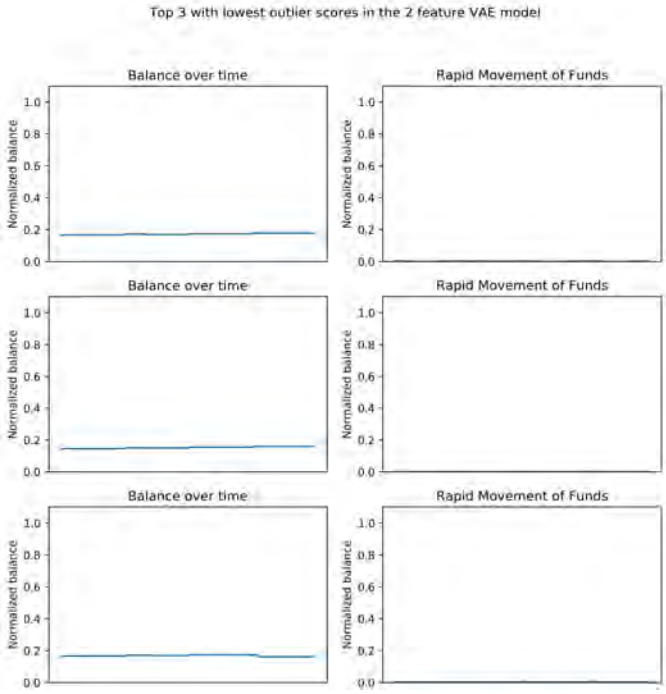

Figure 42: Three windows with the lowest outlier score, for two feature combination VAE model

Three features

The third VAE model which was investigated was the three feature combination, containing balance over time, Rapid Movement of Funds (RMoF) and cross-border. The results of the recall score on the test data are presented in Figure 43. These scores are higher compared to the one and two feature model. The mean recall score for the top 20 is the highest score observed up to now on the test set, namely a score of 0.40. Compared to the mean recall score on the train set, a slight decrease has taken place. The difference between the score is 0.03 so that is rather small. The standard deviation is higher compared to the one feature model. The values deviate between 0.3 and 0.5. These recall scores demonstrate that the three feature model is the highest performance model thus far.

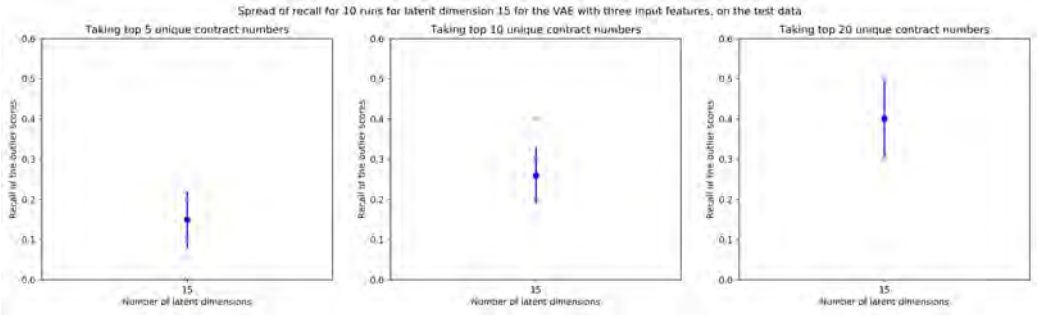

Figure 43: Recall on outlier scores for latent dimension 15 for VAE with three features on the test data, taking top 5, 10 and 20 unique contract numbers

	Top 5	Top 10	Top 20
15 latent dimensions	0.15	0.26	0.40

Table 20: Mean recall scores over ten runs on test data, for VAE with three features

In Figure 44 the three windows with the highest outlier score are presented. Since this is a three feature model, a total of three rhythms are displayed per window. Which are the balance over time, RMoF and cross-border over a time span of 28 days. The first thing that can be noticed from the figure is the regularly cross-border transactions. For VAT/carousel fraud this would be a good indication, since cross-border transactions need to occur. But when investigating the balance over time, not a lot of high peaks are occurring. The balance behaves more as one would expect in a 'normal' contract number. Also the RMoF does not display high peaks. A possible explanation why these contract numbers are labeled with high outlier scores is that indeed a lot of cross-border transaction occur, but these transactions cover relatively small amounts of money. Since the height of the peaks equals the height of the normalized balance, the information regarding the mutation amount is lost. Perhaps when setting the height of the peak equal to the normalized mutation amount, more accurate fraudulent cases are found.

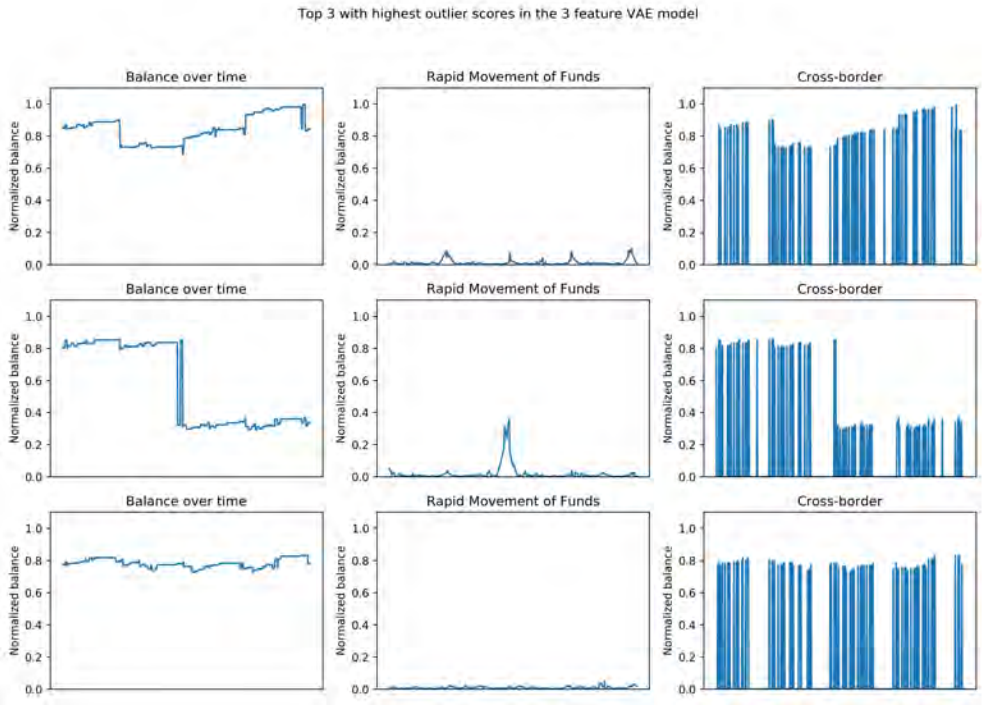

Figure 44: Three windows with the highest outlier score, for three feature combination VAE model

For this feature combination also the median outlier scores were determined. For those cases we noticed that the the balance changes very little, the RMoF behavior shows no high peaks and almost no cross-border transactions occur. This matches with the expectation of windows that are in the middle. Some small activity is happening but not something that points to fraudulent behavior. This figure can be found in the Appendix.

The last figure for the three feature model is that of the three windows with the lowest outlier scores. This displays rhythms as expected, flat lines for RMoF and cross-border transactions. And the balance decreases very slowly over time. The three windows with the lowest outlier scores all have the same contract number. From this figure it can be clearly seen that no fraudulent behavior

is present in these time windows.

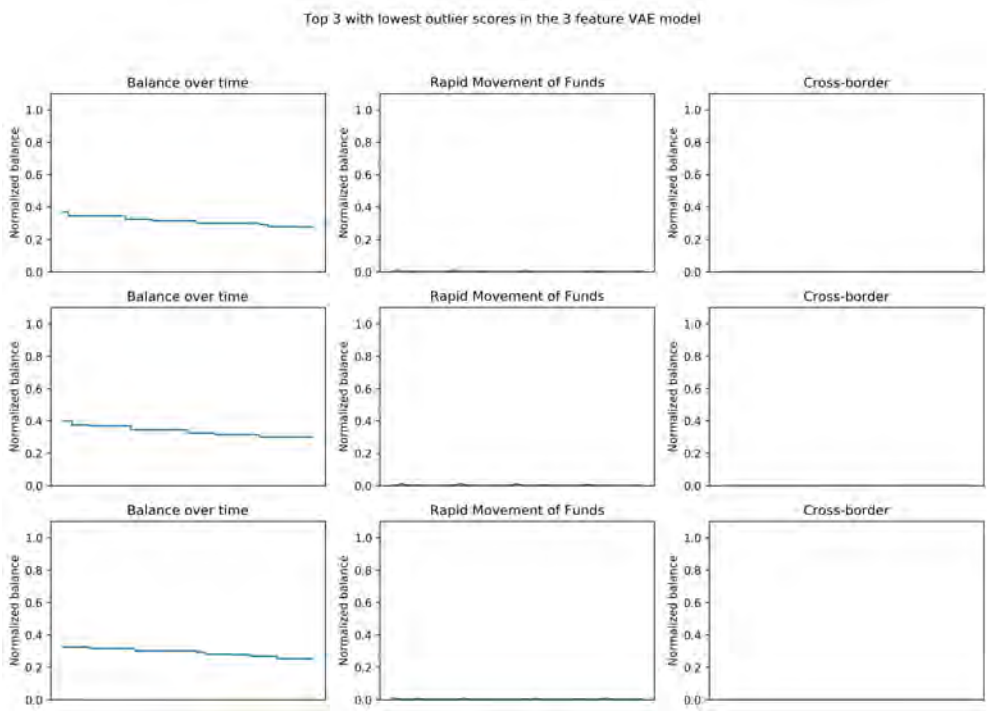

Figure 45: Three windows with the lowest outlier score, for three feature combination VAE model

Four features (13 dimensions)

The next model which was investigated is the four feature combination with 13 dimensions. The input for the VAE is balance over time, Rapid Movement of Funds (RMoF), cross-border and the top 5 debit and credit counterparties. In Figure 46 the results of the recall score on the test data can be seen. This combination obtains the lowest recall score yet, of 0.17. This performance is conform the results of the train data. Here this feature combination obtained a recall score of 0.21. So a slight decrease has occurred. The standard deviation is rather small, which is an advantage because the value do not deviate much from the mean.

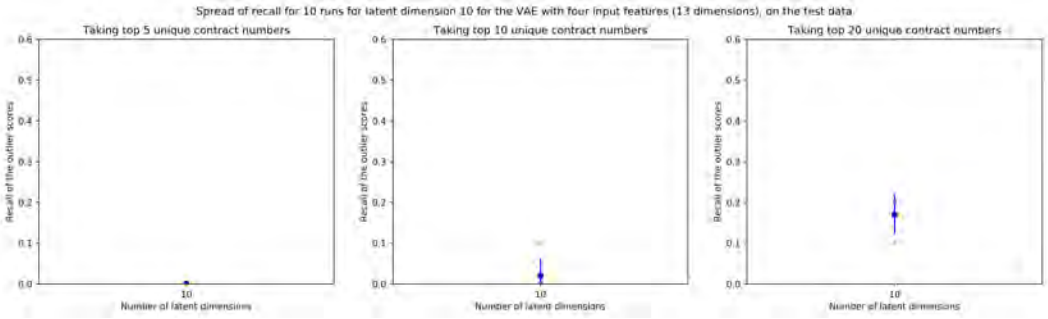

Figure 46: Recall on outlier scores for latent dimension 10 for VAE with four features (13 dimensions) on the test data, taking top 5, 10 and 20 unique contract numbers

	Top 5	Top 10	Top 20
10 latent dimensions	0.00	0.02	0.17

Table 21: Mean recall scores over ten runs on test data, for VAE with four features (13 dimensions)

As can be seen in Figure 46, the lowest mean recall score yet is observed. Therefore, the decision was made to present the three windows with the highest, median and lowest outlier scores in the Appendix for the interested reader.

Four features (5 dimensions)

The fifth feature combination that was investigated was again four features, only now 5 dimensions are taken. Now the input for the VAE model is balance over time, Rapid Movement of Funds (RMoF), cross-border and the first credit and debit counterparty. The obtained recall scores are shown in Figure 46. These score show a improvement compared to the four feature combination with 13 dimensions. The calculated recall score on the test data is 0.26, which is almost the same as on the train set. There a recall score of 0.27 was obtained. However, it is still below the highest score yet of 0.40 and below the score of the one feature model, which was 0.34. From the figure it can be seen that the standard deviation is larger compared to the 13 dimensions.

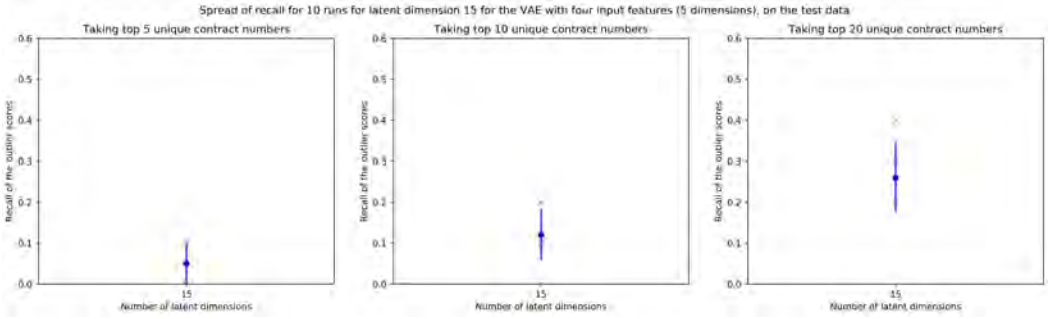

Figure 47: Recall on outlier scores for latent dimension 15 for VAE with four features (5 dimensions) on the test data, taking top 5, 10 and 20 unique contract numbers

	Top 5	Top 10	Top 20
15 latent dimensions	0.05	0.12	0.26

Table 22: Mean recall scores over ten runs on test data, for VAE with four features (5 dimensions)

This feature combination scores below the one and three feature models. Therefore, the decision was made to show the windows with the highest, median and lowest scores in the Appendix for the interested reader.

Five features

The last VAE model which was investigated, is the five feature combination. The input for the VAE model is balance over time, Rapid Movement of Funds (RMoF), cross-border, top 5 credit and debit counterparties, and cash. In Figure 48 the recall scores for this model are presented. This feature combination demonstrates the lowest recall scores on the test set. Namely a mean score of 0.14 for the top 20 unique contract numbers. Considering that the train data had a mean recall score of 0.26, it has dropped significantly. A possible explanation for the low recall scores is that the investigated number of latent dimensions values were set too low in the grid search of this project. The input for the VAE with five features has as input 14 dimensions, therefore it is possible that a higher number than 25 is needed for the number of latent dimensions.

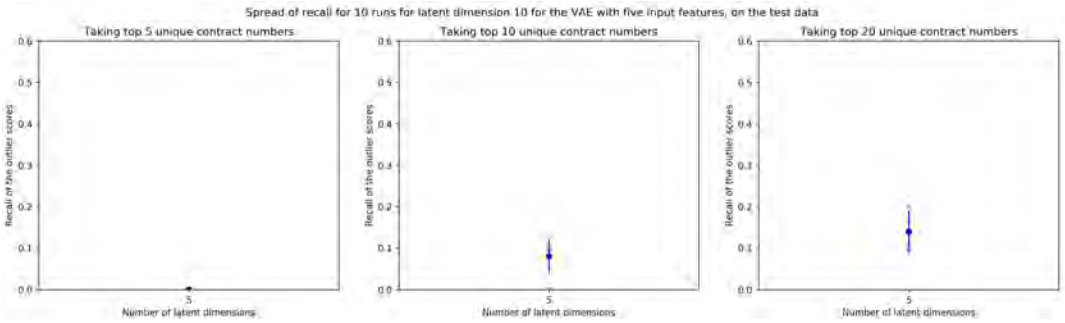

Figure 48: Recall on outlier scores for latent dimension 5 for VAE with five features on the test data, taking top 5, 10 and 20 unique contract numbers

	Top 5	Top 10	Top 20
5 latent dimensions	0.00	0.08	0.14

Table 23: Mean recall scores over ten runs on test data, for VAE with five features

It can be concluded that the five feature model obtains the lowest mean recall score of all the investigate feature models. So, the decision was made to present the highest, median and lowest three windows in the Appendix for the interested reader.

8.3.2 Analysed with fraud investigator perspective

In this section the five contract numbers with the highest outlier scores per feature combination are discussed with fraud investigators. This is to determine if the results make sense from there perspective. Also it can be investigated if new fraud cases or patterns can be found using a VAE model. The findings with the investigators are summarized below.

One feature model The contract number with the highest outlier score does not show any particular signs of fraud. Another contract number does also show no signs of fraud. Both accounts are used rarely and have some large transactions, which is the most likely reason they are indicated as outlier by the model. The other three contract numbers do show more fraudulent behavior. All accounts have very few counterparties and most of the time have the same counterparties, which is conform the behavior of VAT/carousel fraud. Also, the money is quickly flowing through the accounts. The investigators stated that a more indepth analysis would be needed for these three contract numbers to determine whether they are committing fraud or not, since they are not flagged as fraud by the bank.

Two features model Two contract numbers display suspicious behavior. In one account money flows in via a non-EU country and is almost immediately transferred to an EU country. The other account transfers money quickly to one EU country after it received the money. Both accounts also have a relative large flow of money for a relative small company. The investigators stated that more investigation is needed into these accounts, but based on this first investigation they display signs of fraudulent behavior. Another contract number does have weakly payments to the same companies, also the counterparties are constantly the same. So this contract number shows some possible signs of fraudulent behavior. In the last two contract numbers no suspicious behavior was found.

Three features model One contract number has many cross-border transactions. Which was also visible in Figure 44. After some investigation from the fraud investigator, it was concluded that this company is not fraudulent. It is a large specialist company which has clients all around

Europe. Another contract number certainly had committed fraud, which was not yet labeled in the data set but was known by the bank. A third contract number does large investments into real estate in European countries. Since it deal with such high numbers of transactions, the investigators would advice to inspect this contract number further. The last two contract numbers do not display any fraudulent behavior.

Four features (13 dimensions) model Out of the five contract numbers, only one contract number is definitely fraud, but not labeled as such in the data. All the other contract numbers do not display any fraudulent behavior. The four accounts deal with a large number of counterparties, also the balance stays relatively the same. The high number of counterparties could be a possible explanation for these accounts to be labeled as outliers.

Four features (5 dimensions) model One contract number was no longer a client of the bank. Therefore, it was not possible to analysis this anymore. Two other contract numbers did not display any signs of fraudulent behavior. The last two contract numbers could potentially be fraudulent. It are companies which have all the components for it, but a more indepth investigation is needed.

Five features model All the contract numbers from this model do not display any sign which could point to fraudulent behavior.

From the analysis with the fraud investigators, it can be concluded that the most relevant observations were made in the one, two and three feature models. This is in line with the outcome of the outlier detection methods. There the one and three feature combinations models performed best.

8.3.3 Statistical test

In the previous sections it was observed that for the three feature VAE model the highest mean recall score was obtained. The one feature VAE model had the second highest mean recall score. The *Kolmogorov-Smirnov (KS) test* will be performed to investigate if there is a significant difference between the models. The obtained p-values can be found in Table 24. Where model 1 indicates the one feature model, model 2 the two feature model, model 3 the three feature model, model 4 the four feature model with 13 dimensions, model 5 the four feature model with 5 dimensions, and model 6 the five feature model.

	model 1	model 2	model 3	model 4	model 5	model 6
model 1	1	0.0524	0.4175	1.0825e-05	0.0524	1.0825e-05
model 2	0.0524	1	0.0524	0.4175	0.7869	0.4175
model 3	0.4175	0.0524	1	1.0825e-05	0.0524	1.0825e-05
model 4	1.0825e-05	0.4175	1.0825e-05	1	0.4175	0.7869
model 5	0.0524	0.7869	0.0524	0.4175	1	0.0524
model 6	1.0825e-05	0.4175	1.0825e-05	0.7869	0.0524	1

Table 24: p-values for the Kolmogorov-Smirnov test on the six feature combination VAE models

The models 1 and 3 had the highest mean recall scores, so it is of interest to investigate their p-values. What can be observed from the table is that model 1 and model 3 have a p-value of 0.4175. The p-value is above α, which was set to 0.05. So, it can be concluded that there is insufficient evidence to reject the null hypothesis. This indicates that these two models most likely do not deviate much from one another. When comparing model 1 with the other model options, p-values of 0.05 or lower are found. The same holds for model 3 with the other model options. So comparing these two model with the other models results in rejecting the null hypothesis.

8.4 Explainability and interpretability with the β-VAE

In Chapter 8.2 the decision was made to investigate the β values 2 and 5 for a higher epoch number and on the test data. The β-VAE had as input three features, namely balance over time, RMoF and cross-border. The reason for the three feature input is because this feature combination had the highest mean recall score on the train data. The results of the recall score on the test data are presented in Figure 49. From this table it can be observed that $\beta = 2$ obtains a higher recall score than in the three feature VAE model, namely a score of 0.43. The standard deviation is not that small, but it shows that a few times a score of 0.50 is obtained. Which is relatively high. The value of 5 has a score of 0.27, which is lower compared to the recall score retrieved in the train data. So, based on the recall scores the value of 2 would detect fraudulent behavior more accurate. It even detects fraud more accurate than the three feature VAE model, based on the mean recall scores. However, that was not the goal of the β-VAE. We wanted to investigate if the β-VAE would improve the explainability and interpretability of the latent space of the VAE. Therefore, the three windows with the highest and lowest outlier scores are determined and plotted, for both β values. Based on these figures it will investigated if we can observe an increase in explainability and interpretability.

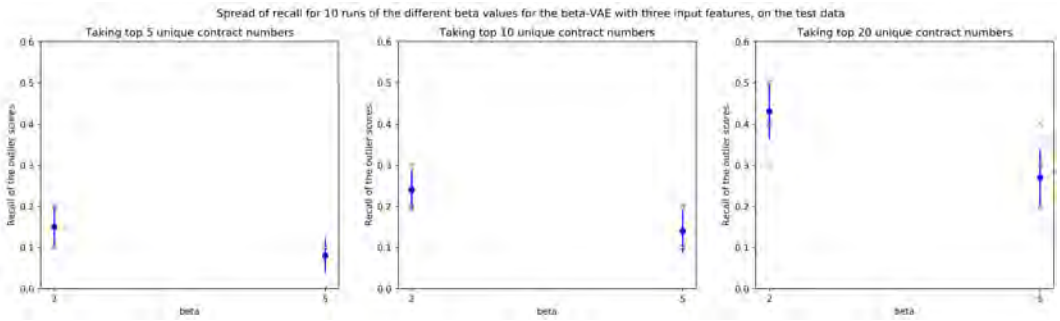

Figure 49: Recall on outlier scores for different β's values for β-VAE with three features on test data, taking top 5, 10, 20 unique contract numbers

	Top 5	Top 10	Top 20
$\beta = 2$	0.15	0.24	0.43
$\beta = 5$	0.08	0.14	0.27

Table 25: Mean recall scores over ten runs on test data, for β-VAE with three features

Begin with the three windows with the highest outlier scores for β equals 2. These are presented in Figure 50. Since the β-VAE has as input three features, a total of three rhythms are displayed per window. Which are the balance over time, RMoF and cross-border of a time span of 28 days. What can be observed from the figure is that the highest three windows are from the same contract number and also relatively close in date. The first two windows have overlap. The next thing what can be noticed is that cross-border is very dominate. Which could be a good thing, since a higher number cross-border transactions is a big indicator for VAT/carousel fraud. However, the balance over time and RMoF do not display the desired rhythm for VAT/carousel fraud. Which is regularly high peaks which return to around zero. As stated in the three feature combination VAE model, the most likely reason why these windows are labeled as high outlier, is that the height of the cross-border peak is set equal to the height of the normalized balance. Perhaps it would be more accurate to set the height of the peak equal to the normalized mutation amount. Then the goal would be to observe many regular high peaks in RMoF and cross-border. These would point to fraudulent behavior. What was observed in the results of the β-VAE was that the contract number for the three windows with highest outlier scores dominated the highest outlier

scores. The first fifty windows with the highest outlier scores were all from the same contract number. This was not observed in the three feature model. This could perhaps indicate that the model does disentangle in some manner and consistently labels this contract number with high outlier scores.

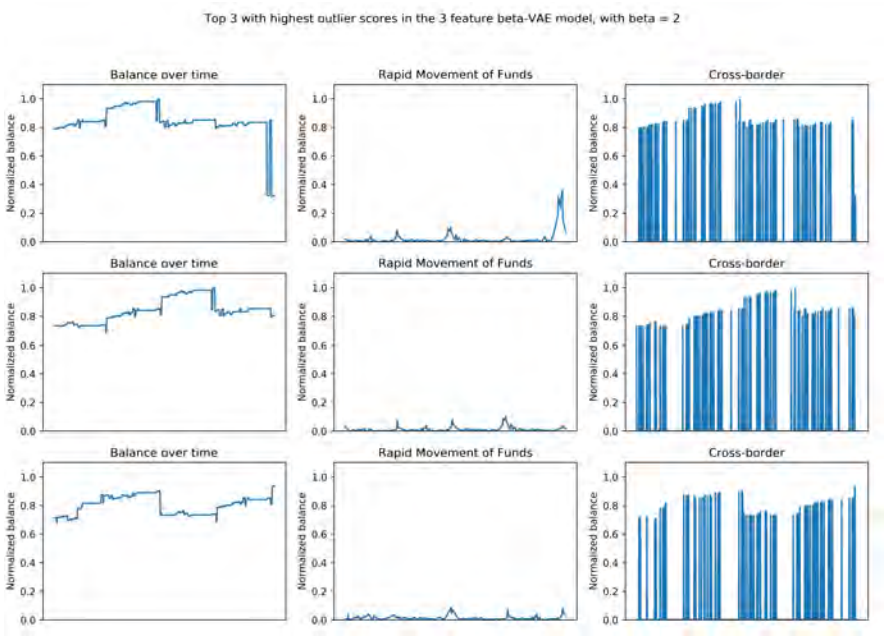

Figure 50: Three windows with the highest outlier score, for three feature combination beta-VAE model, with beta = 2

The three windows with the lowest outlier scores are also plotted. The expectation is that these three windows have an almost flat signal for all three features. The rhythms for the three windows can be seen in Figure 51. As can be observed from this figure, all three features display an almost flat line. Which is in line with the expectation.

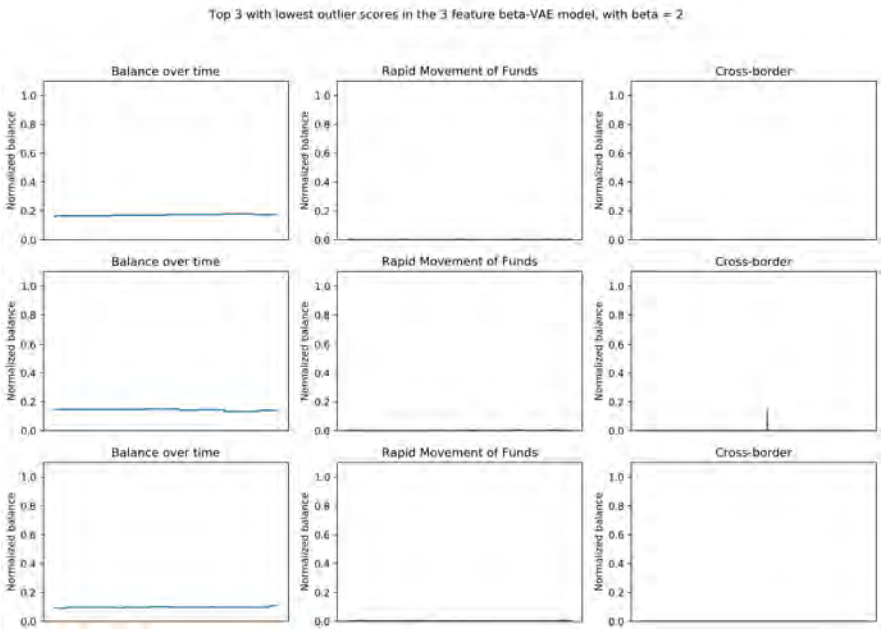

Figure 51: Three windows with the lowest outlier score, for three feature combination beta-VAE model, with beta = 2

Next the three windows with the highest outlier scores for the β equals 5 are determined and shown in Figure 52. When investigating these figures it can be seen that other contract numbers and windows are labeled as highest outlier compared to the value of 2 for the β. In the figure it can be noticed that for two windows some relatively high peaks for RMoF are occurring, while for the other window a lot of cross-border transactions are occurring. All three windows do not necessary display the desired VAT/carousel fraud rhythm in the balance over time.

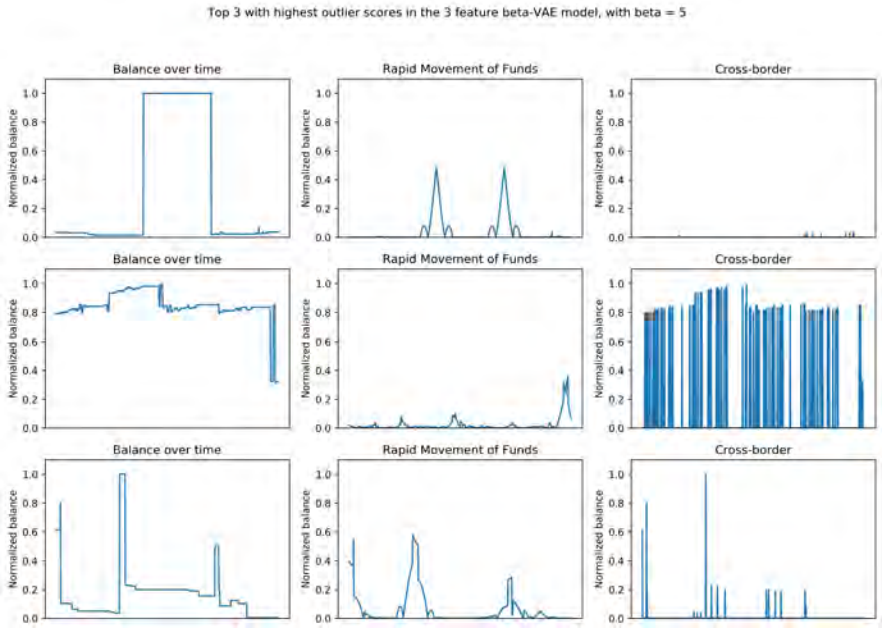

Figure 52: Three windows with the highest outlier score, for three feature combination beta-VAE model, with beta = 5

Lastly, the three windows with the lowest outlier scores are presented in Figure 53. These results are in line with the expectation, as an almost flat line is observed in every window for all three features.

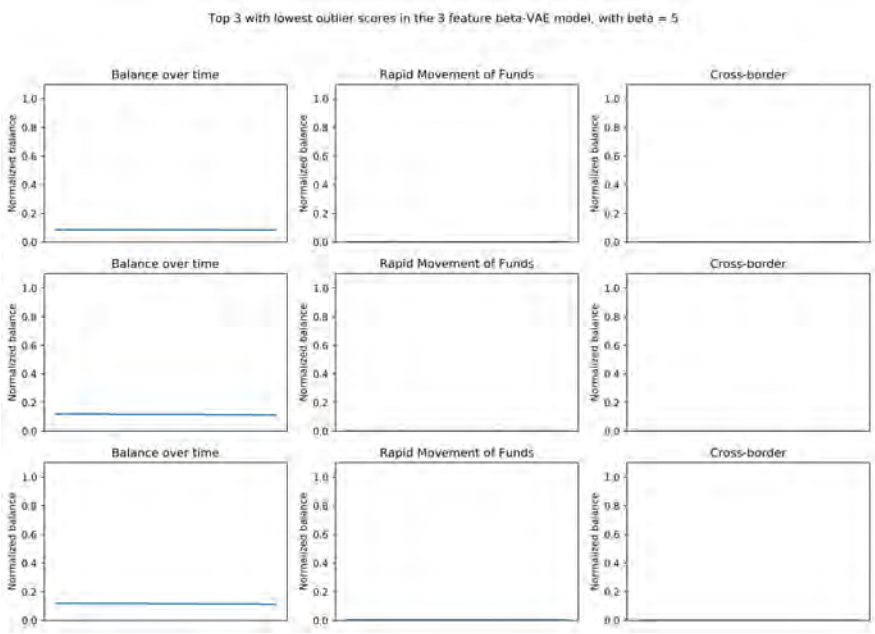

Figure 53: Three windows with the lowest outlier score, for three feature combination beta-VAE model, with beta = 5

It can be concluded that based on these results that the recall score increases when taking a value of 2 for the β. However, the goal was to improve the explainability and interpretability of the latent space of the VAE. The only clear pattern which could be observed in the highest outliers was that a high number of cross-border transactions took place, for β equals 2. Also, that consistently the same contract number which displayed these high peaks of cross-border was labeled with high outlier scores. However, it is difficult to state based on these results if the explainability and interpretability has improved. Maybe, the disentanglement process of the β-VAE can be better observed when taking clusters in the latent space. Then the clusters obtained in the VAE model can be compared with the clusters in the β-VAE model. The expectation is that a clear shift can be observed. Also domain experts and investigators can help in investigating and labeling these clusters, to explore if this increases the explainability. It can be stated that future research on this topic is needed to explore all the possibilities.

9 Discussion

The goal of this project was to use an unsupervised machine learning method to find abnormal behavior in accounts which in turn could point to fraudulent behavior. The research question was defined into two parts:

1. Is using multivariate analyses for an outlier detection in the latent space of a VAE an improvement compared to univariate analyses?;
2. How can implementing the β-VAE improve the interpretability and explainability of the underlying representation of the VAE?

To answer the first research question, a total of six VAE models were investigated. Each model consists of a different feature combination. In the first research question it is investigated if the performance of the VAE would increase when taking more than one feature as input. The evaluation metric applied to investigate the performance was the recall. The results of the performance of the six VAE models was presented in Chapter 8. From these results it was observed that the one feature VAE model had a recall score of 0.34, which is relatively high. Only the three feature VAE model obtained a higher recall score, namely a score of 0.40. All the other models got scores below 0.34. Also the statistical test showed that the one and three feature VAE model are significantly not that different, but are very different from the other models. The three feature model uses the following features: balance over time, RMoF and cross-border. That this feature combination works best was to be expected according to domain knowledge. This is especially true since RMoF and cross-border transactions are big indicators for VAT/carousel fraud. These recall scores are promising because it is difficult to obtain high recall scores for imbalanced data. A possible explanation of the drop in recall scores for the four and five feature models would be that the number of dimensions as input for the VAE model increases significantly. Therefore, it is possible that a value of 25 latent space dimensions is too low and a higher recall score could be obtained when investigating higher values for this parameter. The top three windows with the highest outlier score mostly displayed a rhythm of dropping from one to zero or vice versa. These are outliers, but these do not necessarily display the desired behavior for VAT/carousel fraud. The three lowest windows for all feature combinations demonstrated the expected behavior, which was that almost no activity was observed. Solemnly based from a data science perspective, it can be stated that using a multivariate analyses for an outlier detection in the latent space of a VAE is indeed an improvement compared to univariate analyses. When looking at the results from a fraud investigator perspective, some promising results were observed. However, some contract numbers which did not display fraudulent behavior had high outlier scores as well. When increasing the number of features, the number of discovered fraud cases decreases. Although this method shows promising results, there is room for improvement.

This project investigated the possibilities of applying an unsupervised machine learning method to detect VAT/carousel fraud. The results are promising and the methods can most likely be implemented after further investigations. What must be highlighted from this project is the high accuracy obtained in the RMoF feature. The results from this feature demonstrated that for the top 20 contract numbers in the RMoF feature an accuracy of above 80% was obtained, which is extremely high. With this feature, some fraud cases that were not yet labeled by the bank were found. This feature makes it possible to detect all different kinds of fraud, not only VAT/carousel fraud. This is because most fraud types include a form of RMoF. An advantage of implementing a 'simple' technique would be that we can better understand what the algorithm is doing and explain the obtained results. Which makes it easier to incorporate such method in decision making, which is eventually the desired result. For example, the RMoF feature could be used with other risk indicators to determine whether a contract number has potential of being fraudulent. There could be even more potential in the RMoF feature than what is investigated in this project. A more

indepth investigation of the parameter settings for the smoothing filter can make the detection of RMoF even more accurate.

So, perhaps in future investigations of the bank regarding fraud topics it might be valuable to apply this RMoF feature. Not to state that the complex deep learning methods, such as the VAE, are not valuable. But if the bank wants to implement a metric in a short period of time that can spot RMoF which in turn could point to fraud. The RMoF feature might be a good solution. In the mean time a more indepth investigation into the VAE and other unsupervised machine learning methods can be conducted.

One of the big problems regarding deep learning methods is the lack of interpretability and explainability of the underlying representation of the methods. This is what the second research question tries to address. In order to overcome this problem, the β-VAE is implemented. It was decided to implement the β-VAE for the highest recall scoring VAE model, which was the three feature VAE model. From the results in Chapter 8 it was obtained that the value of 2 for the β displays the highest mean recall score, namely a score of 0.43. This score was even higher than the mean recall score obtained in the three feature VAE model, there a score of 0.40 was retrieved. The value of 5 for the β showed promising results on the train data, but the performance dropped significantly when executed on the test data. However, the second research question does not necessarily want to increase the recall score but wants to improve the interpretability and explainability of the VAE. To try to answer this question the three highest outlier scores for both values were investigated. It can be argued if this is the best method of investigation an increase in explainability and interpretability. Most likely the use of cluster can better determine if the β-VAE indeed disentangles the latent space more compared to the VAE. For the value 5 not a very clear pattern could be observed when taking the three windows with the highest outlier scores. The value of 2 did display a clear pattern, namely the feature cross-border was very dominate. In the results the first fifty windows were all from the same contract number, which showed the pattern of very high peaks in the feature cross-border. This could perhaps indicate that the β-VAE model disentangles the latent space more and therefore consistently labeled this contract number as an outlier. The balance over time for this contract number is not the pattern desired for VAT/carousel fraud. A possible explanation for this would be that the height of a peak for a cross-border transaction is set equal to the height of the normalized balance, then the information regarding the mutation amount is lost. So, if the height of the peak indicates the normalized mutation amount a better insight can be obtained on the mutation amount. Then the model needs to strive for high regular peaks in RMoF and cross-border transactions for VAT/carousel fraud detection. It is difficult to state based on investigation of the windows with the highest outlier scores if the interpretability and explainability has improved. Some small indications for this are occurring, such as the consistently labeling one contract number as outlier which displays a clear patterns in cross-border transactions. Perhaps in future research, it can be investigated if the disentanglement process of the β-VAE can be better observed when taking clusters in the latent space. The clusters obtained in the VAE model can be compared with the clusters in the β-VAE model. The expectation is that a clear shift can be observed. To summarize, this first investigation into the β-VAE shows some promising results but more extensive future research on this topic is needed to explore all the possibilities.

Added value of the project

This project tried to contribute to filling the research gap on exploring the possibilities of applying unsupervised machine learning methods for the detection of VAT/carousel fraud. To the best of our knowledge, there is only one research paper that uses an unsupervised machine learning technique for the detection of VAT/carousel fraud. The authors implemented the nearest neighbour algorithm as a detection system [120]. The data used by the authors is different then the data used in this project. Therefore, it is hard to compare the results of the authors with the results from this project. The results from this paper obtained high scores in their evaluation metrics, which where hit curves and lift rates. Therefore, it might be of interest to implement the nearest

neighbour algorithm for further research on this topic, using this projects data set and to compare the results.

The results of this project was valuable for the bank. First of all, applying a smoothing filter to detect RMoF has not been done before. The currently used RMoF detection technique of the bank does not obtain as high of an accuracy as the RMoF feature in this project. Next, this project is among the first to explore the possibilities of applying unsupervised machine learning methods for the detection of VAT/carousel fraud. Since millions of euros are lost yearly due to this type of fraud, it is valuable for banks to detect fraudulent activity as quickly as possible. This could be achieved with the use of unsupervised machine learning techniques. Another added value of this project would be that the VAE can perhaps be implemented for the detection of other types of fraud. The features will most likely have to be altered per fraud type.

This project demonstrated that unsupervised machine learning methods, such as the VAE, can be used to detect VAT/carousel fraud. The recall is not yet very high, but this is common for highly imbalanced data sets. The overall picture demonstrates promising results, but further research into the VAE, the features, and other unsupervised machine learning methods is needed.

Limitations

There were some limitations in the project which need to be addressed. The first one is the relatively small number of labeled data. This is always a problem when dealing with fraud data and is also the reason why this project investigated unsupervised methods. However, it would be very valuable to obtain more labeled data, such that the methods can be better evaluated and trained. The second limitation is that the data set is labeled by business number. It is possible that there are multiple contract numbers within one business number and it is therefore possible that not all these contract numbers are committing fraud but are labeled fraudulent anyway. The third limitation is that the data set is rather small. For future research, a larger data set would be preferred. The last limitation is that the clusters on which the method were trained got memory errors quickly. This last limitation must be taken into account for future research, especially when dealing with more complex and longer methods.

Future research

There are many future research possibilities on the topic of VAT/carousel fraud detection. According to our results, the three feature combination performed best. When altering the feature cross-border, perhaps even higher recall scores can be obtained. Currently, the height of a cross-border peak is set equal to the height of the normalized balance. When doing it this way, information regarding the mutation amount is lost. Therefore, it might be interesting to set the height of the cross-border peaks equal to the normalized mutation amount. Then, the height of the peak indicates the amount of mutation. The goal is to obtain windows where regular high peaks for RMoF and cross-border are occurring. These windows will in turn most likely point to fraudulent behavior.

Another research topic could be to investigate other feature combinations. In this project the decision was made to make feature combinations based on domain knowledge. It could be that other feature combinations work better. For example, the RMoF and cross-border combination can be investigated, since these are very strong indicators for VAT/carousel fraud. Also, in future research more features can be created and investigated.

Furthermore, other normalization techniques can be investigated. The current technique has a downside that the information regarding the height of the balance is lost. In future research this component could be preserved, since this would be valuable for the model.

Future research could also investigate different parameter settings for the VAE and β-VAE. In this project unfortunately only a few parameter settings could be investigated, due to time constraints. Therefore, it is extremely likely that the chosen values are not optimal for the parameters. For example, the use of Bayesian hyperparameter search could result in more optimal values for the different parameters.

Another potential research topic would be to spot fraud in clusters in the latent space of the VAE. In this project the assumption was made that fraud can be spotted using outlier detection techniques, such as Isolation Forest. However, it is possible that not all VAT/carousel fraud can be spotted with outlier detection techniques only. So, it might be interesting to investigate the latent space of the VAE on the clusters. Perhaps the latent space of the VAE shows very distinct clusters and the fraud cases would be clustered together. Then, other cases near this cluster could be investigated as well.

Lastly, there are some future research suggestions for the β-VAE. A more indepth analysis for the optimal value of the β could be done. In the paper by Higgins et al. [52] the authors proposed a disentangled metric. This metric could be used to obtain the best value for β depending on your data set. For future research on the β-VAE it would be interesting to dive into this. Due to time constraints this could not be investigated in this project. Another research topic for the β-VAE would be to investigate the latent space on clusters. The expectation is that the disentanglement could be visually observed in clusters.

10 Conclusion

In this project the possibilities of applying an unsupervised machine learning model to detect VAT/carousel fraud were explored. The goal was to use an unsupervised machine learning method to find abnormal behavior in accounts which in turn could point to fraudulent behavior. The following two research question were defined:

1. Is using multivariate analyses for an outlier detection in the latent space of a VAE an improvement compared to univariate analyses?;
2. How can implementing the β-VAE improve the interpretability and explainability of the underlying representation of the VAE?

To answer the first research question, a total of six VAE models were investigated. It is investigated if the performance of the VAE would increase when taking more than one features as input. From the results in Chapter 8, it was observed that the three feature VAE model obtains the highest recall score, namely a score of 0.40. The one feature VAE model had a recall score of 0.34. It can be concluded that using a multivariate analyses for outlier detection in the latent space of a VAE is indeed an improvement compared to univariate analyses. Also, from a fraud investigators perspective the three feature VAE model showed more promising results than the one feature VAE model. The overall picture demonstrates promising results, but further research into the VAE, the features, and other unsupervised machine learning methods is needed.

The second research question investigates if the implementation of the β-VAE improves the interpretability and explainability of the underlying representation of the VAE. The β-VAE was implemented for the highest recall scoring VAE model, which was the three feature VAE model. The value of 2 for the β displays an improvement in the mean recall score compared to the three feature VAE model. Also, some pattern in the windows with the highest outlier scores could be observed. A contract number with regularly high peaks in cross-border transactions was labeled consistently with high outlier scores. This could indicate that the model more clearly disentangles the different features. This first investigation into the β-VAE shows some promising results. However, future research into the β-VAE is needed to explore all the possibilities. For example, the disentanglement process of the β-VAE might be more clearly observed when taking cluster in the latent space instead of applying an outlier detection technique.

The main goal of this project was to contribute to filling the research gap on applying unsupervised machine learning methods for the detection of VAT/carousel fraud. The quick detection of this type of fraud is valuable for banks, since millions of euros are lost yearly due to this type of fraud. Also, the development of the RMoF is valuable for the bank. The currently employed RMoF techniques by the bank have a lower accuracy then the technique employed in this project. The RMoF feature can be implemented for all types of fraud and can help the bank to detect fraudulent behavior more quickly. Lastly, there is the potential to use the VAE as detection system for other types of fraud.

Some suggestions for future research are to investigate other feature combinations and to develop new features. Another suggestion is to apply a different normalization technique to the data which preserves the height of the balance. A third suggestion is to implement the Bayesian hyperparameter search to find the optimal parameter values for the VAE and β-VAE. A fourth suggestion would be to execute a more indepth analysis for the optimal value for β by implementing the disentangled metric proposed by Higgings et al. [52]. Lastly, we suggest to investigate if more VAT/carousel fraud cases can be spotted when clustering the latent space of the VAE, instead of applying an outlier detection technique to the latent space of the VAE.

References

[1] Adamov, A.Z. "Machine Learning and Advanced Analytics in Tax Fraud Detection". In: *IEEE 13th International Conference on Application of Information and Communication Technologies (AICT)* (2019).

[2] Ahmed, M., Mahmood, A. N., and Islam, M. R. "A survey of anomaly detection techniques in financial domain". In: *Future Generation Computer Systems* 55 (2016), pp. 278–288.

[3] Alam, M. *Isolation Forest: A Tree-based Algorithm for Anomaly Detection*. Last accessed on 03-02-2022. URL: https://towardsdatascience.com/isolation-forest-a-tree-based-algorithm-for-anomaly-detection-4a1669f9b782.

[4] Albashrawi, M. "Detecting Financial Fraud Using Data Mining Techniques: A Decade Review from 2004 to 2015". In: *Journal of Data Science* 14.3 (2016), pp. 553–569.

[5] An, J. and Cho, S. "Variational Autoencoder based Anomaly Detection using Reconstruction Probability". In: *SNU Data Mining Center, Special Lecture on IE* (2015).

[6] Aruchamy, V. *How To Normalize Data Between 0 And 1*. Last accessed on 20-01-2022. URL: https://www.stackvidhya.com/how-to-normalize-data-between-0-and-1-range/.

[7] Belastingdienst. *Btw-tarieven: welke tarieven zijn er, en wanneer moet u ze toepassen?* Last accessed on 13-12-2021. URL: https://www.belastingdienst.nl/wps/wcm/connect/bldcontentnl/belastingdienst/zakelijk/btw/btw_berekenen_aan_uw_klanten/btw_berekenen/btw_tarief/btw_tarief.

[8] Belastingdienst. *Voorbeeld van btw-carrouselfraude*. Last accessed on 03-11-2021. URL: https://www.belastingdienst.nl/wps/wcm/connect/bldcontentnl/belastingdienst/zakelijk/btw/btw_aangifte_doen_en_betalen/btw-fraude/btw_carrouselfraude/voorbeeld_van_btw_carrouselfraude.

[9] Bengio, Y. "Practical Recommendations for Gradient-Based Training of Deep Architectures". In: *Neural Networks: Tricks of the Trade* (2012), pp. 437–478.

[10] Bengio, Y., Simard, P., and Frasconi, P. "Learning long-term dependencies with gradient descent is difficult". In: *IEEE Transactions on Neural Networks* 5.2 (1994), pp. 157–166.

[11] Bhulai, S. "Recurrent neural networks". In: *Advanced Machine Learning Lecture 10, slides* (2020).

[12] Bishop, C.M. "Pattern Recognition and Machine Learning". In: Springer-Verlag New York Inc., 2011. Chap. chapter 5.

[13] Borselli, F. "Pragmatic Policies to Tackle VAT Fraud in the European Union". In: *International VAT Monitor* 5 (2008), pp. 333–342.

[14] Brital, A. *AutoEncoders Explained*. Last accessed on 26-01-2022. URL: https://medium.com/@AnasBrital98/autoencoders-explained-da131e60e02a.

[15] Brownlee, J. *How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification*. Last accessed on: 15-04-2022. URL: https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/#:~:text=We%5C%20can%5C%20calculate%5C%20the%5C%20recall%5C%20as%5C%20follows%5C%3A%5C%201,%5C%2F%5C%20%5C%2895%5C%20%5C%2B%5C%205%5C%29%5C%203%5C%20Recall%5C%20%5C%3D%5C%200.95.

[16] Buchanan, B. "Money laundering - a global obstacle". In: *Research in International Business and Finance* 18 (2004), pp. 115–127.

[17] Burgess, C.P. et al. "Understanding disentangling in β-VAE". In: *31st Conference on Neural Information Processing Systems (NIPS)* (2017).

[18] Castellón González, P. and Velásquez, J. D. "Characterization and detection of taxpayers with false invoices using data mining techniques". In: *Expert Systems with Applications* 40.5 (2013), pp. 1427–1436.

[19] Chen, J. *Money Laundering*. Last accessed 24-11-2021. URL: https://www.investopedia.com/terms/m/moneylaundering.asp.

[20] Chen, J. *Neural Network*. Last accessed on 22-01-2022. URL: https://www.investopedia.com/terms/n/neuralnetwork.asp.

[21] Chen, J. *What is Fraud?* Last accessed on 07-12-2021. URL: https://www.investopedia.com/terms/f/fraud.asp.

[22] Chen, Z. et al. "Machine learning techniques for anti-money laundering (AML) solutions in suspicious transaction detection: a review". In: *Knowledge and Information Systems* 57 (2018), pp. 245–285.

[23] CISI. "The Background and Nature of Financial Crime". In: Charted Institute for Securities and Investment. Chap. 1.

[24] Comply Advantage. *What Is Financial Crime And How It Can Affect You?* Last accessed on 01-12-2021. URL: https://complyadvantage.com/insights/financial-crime/#:~:text=Financial%5C%20crime%5C%20is%5C%20defined%5C%20as%5C%20crime%5C%20that%5C%20is,forms%5C%2C%5C%20and%5C%20they%5C%20happen%5C%20all%5C%20over%5C%20the%5C%20world.

[25] Consumer Protect. *12+ Most Common Types of Fraud & Schemes*. Last accessed on 08-12-2021. URL: https://www.consumerprotect.com/crime-fraud/most-common-types-of-fraud-schemes/.

[26] De Roux, D. et al. "Tax Fraud Detection for Under-Reporting Declarations Using an Unsupervised Machine Learning Approach". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2018), pp. 215–222.

[27] DeltaNet. *What is VAT Fraud?* Last accessed on 03-11-2021. URL: https://www.delta-net.com/compliance/fraud-awareness/faqs/what-is-vat-fraud.

[28] Domingues, R. et al. "A comparative evaluation of outlier detection algorithms: Experiments and analyses". In: *Pattern Recognition* 74 (2018), pp. 406–421.

[29] European Commission. *Single market*. Last accessed on 13-12-2021. URL: https://ec.europa.eu/info/policies/single-market_en.

[30] European Commission: Taxation and Customs Union. *VAT Gap*. Last accessed on 03-11-2021. URL: https://ec.europa.eu/taxation_customs/vat-gap_en.

[31] European Commission: Taxation and Customs Union. *What is VAT?* Last accessed on 13-12-2021. URL: https://ec.europa.eu/taxation_customs/what-vat_en.

[32] Europol. *Europol helps Spanish authorities break up a €26.5 million VAT fraud scheme*. Last accessed on 15-03-2022. URL: https://www.europol.europa.eu/media-press/newsroom/news/europol-helps-spanish-authorities-break-%5C%e2%5C%82%5C%ac265-million-vat-fraud-scheme.

[33] Europol. *Money Laundering*. Last accessed on 26-11-2021. URL: https://www.europol.europa.eu/crime-areas-and-trends/crime-areas/economic-crime/money-laundering.

[34] Europol. *MTIC (Missing Trader Intra Community) fraud*. Last accessed on 13-01-2022. URL: https://www.europol.europa.eu/crime-areas-and-statistics/crime-areas/economic-crime/mtic-missing-trader-intra-community-fraud.

[35] Financial Action Task Force. *Report on Money Laundering and Terrorist Financing Typologies 2003-2004*. Financial Action Task Force: Paris 2004.

[36] Financial Action Task Force. *Terrorist Financing*. Last accessed on 06-12-2021. URL: https://www.fatf-gafi.org/publications/fatfgeneral/documents/terroristfinancing.html.

[37] Financial Action Task Force. *What is money laundering?* Last accessed on 24-11-2021. URL: https://www.fatf-gafi.org/faq/moneylaundering/#d.en.11223.

[38] FIOD Belastingdienst, W. Visser. *Onderzoek naar miljoenfraude met btw*. Last accessed on 15-03-2022. URL: https://www.fiod.nl/onderzoek-naar-miljoenenfraude-met-btw/.

[39] Formula Search Engine. *Savitzky–Golay filter*. Last accessed on 21-02-2022. URL: https://en.formulasearchengine.com/wiki/Savitzky%5C%E2%5C%80%5C%93Golay_filter.

[40] FSMA2000. *Financial Services and Markets Act 2000*. Section 6(3).

[41] GeeksforGeeks. *Introduction to Recurrent Neural Network*. Last accessed on 14-03-2022. URL: https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/.

[42] GeeksforGeeks. *Kolmogorov-Smirnov Test (KS Test)*. Last accessed on: 25-04-2022. URL: https://www.geeksforgeeks.org/kolmogorov-smirnov-test-ks-test/.

[43] GeeksforGeeks. *Long Short Term Memory Networks Explanation*. Last accessed on 02-02-2022. URL: https://www.geeksforgeeks.org/long-short-term-memory-networks-explanation/.

[44] GeeksforGeeks. *Variational AutoEncoders*. Last accessed on 28-01-2022. URL: https://www.geeksforgeeks.org/variational-autoencoders/.

[45] Government of Canada: Financial Transactions and Reports Analysis Centre of Canada. *What is terrorist financing?* Last accessed on 06-12-2021. URL: https://www.fintrac-canafe.gc.ca/fintrac-canafe/definitions/terrorist-terroriste-eng.

[46] Gradeva, K. "VAT Fraud in Intra-EU Trade". In: (2014).

[47] Great Learning Team. *Introduction to Autoencoders? What are Autoencoders Applications and Types?* Last accessed on 26-01-2022. URL: https://www.mygreatlearning.com/blog/autoencoder/.

[48] Gu, S., Kelly, B., and Xiu, D. "Autoencoder asset pricing models". In: *Journal of Econometrics* (2020).

[49] Han, K. et al. "Variational autoencoder: An unsupervised model for encoding and decoding fMRI activity in visual cortex". In: *NeuroImage* 198 (2019), pp. 125–136.

[50] He, H. and Ma, Y. "Imbalanced Learning: Foundations, Algorithms, and Applications 1st Edition". In: Wiley-IEEE Press, 2013. Chap. p. 55.

[51] Heidenreich, H. *What are the types of machine learning?* Last accessed on: 26-04-2022. URL: https://towardsdatascience.com/what-are-the-types-of-machine-learning-e2b9e5d1756f.

[52] Higgins, I. et al. "β-VAE: Learning Basic Visual Concepts With A Constrained Variational Framework". In: *conference paper at ICLR* (2017).

[53] Hilal, W., Gadsden, S. A., and Yawney, J. "Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances Expert Systems With Applications". In: *Expert Systems With Applications* 193 (2022).

[54] Hochreiter, S. and Schmidhubber, J. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[55] Hodge, V.J. and Austin, J. "A Survey of Outlier Detection Methodologies". In: *Artificial Intelligence Review* 22.2 (2004), pp. 85–126.

[56] IBM Cloud Education. *Neural Networks*. Last accessed on 22-01-2022. URL: https://www.ibm.com/cloud/learn/neural-networks.

[57] IBM Cloud Education. *Recurrent Neural Networks*. Last accessed on 14-03-2022. URL: https://www.ibm.com/cloud/learn/recurrent-neural-networks.

[58] IBM Cloud Education. *Supervised Learning*. Last accessed on 22-01-2022. URL: https://www.ibm.com/cloud/learn/supervised-learning.

[59] Ieracitano, C. et al. "A Novel Statistical Analysis and Autoencoder Driven Intelligent Intrusion Detection Approach". In: *Neurocomputing* (2019).

[60] International Complianc eAssociation. *What is Financial Crime?* Last accessed on 01-12-2021. URL: https://www.int-comp.org/careers/your-career-in-financial-crime-prevention/what-is-financial-crime.

[61] Investopedia Team. *Value-Added Tax (VAT)*. Last accessed on 13-12-2021. URL: https://www.investopedia.com/terms/v/valueaddedtax.asp.

[62] Irwin, A.S.M. and Choo, K-K.R. and Liu, L. "An analysis of money laundering and terrorism financing typologies". In: *Journal of Money Laundering Control* 15.1 (2011), pp. 85–111.

[63] Jakubowski, J. et al. "Anomaly Detection in Asset Degradation Process Using Variational Autoencoder and Explanations". In: *Sensors* 22.291 (2022).

[64] Jordan, J. *Introduction to autoencoders*. Last accessed on 26-01-2022. URL: https://www.jeremyjordan.me/autoencoders/.

[65] Jordan, J. *Variational autoencoders*. Last accessed on 28-01-2022. URL: https://www.jeremyjordan.me/variational-autoencoders/.

[66] Kagan, J. *Combating the Financing of Terrorism (CFT)*. Last accessed on 06-12-2021. URL: https://www.investopedia.com/terms/c/combating-financing-terrorism-cft.asp.

[67] Kalchbrenner, N., Danihelka, I., and Graves, A. "Grid Long Short-Term Memory". In: *Conference paper at ICLR* (2016).

[68] Keen, M. and Smith, S. "VAT Fraud and Evasion: What Do We Know and What Can Be Done?" In: *National Tax Journal* 59.4 (2006), pp. 861–887.

[69] Kingma, D.P. and Welling, M. "Auto-Encoding Variational Bayes". In: *arXiv:1312.6114v10* ().

[70] KYC Lookup. *What is Financial Crime, types of financial crimes and Who commits Financial Crimes*. Last accessed on 01-12-2021. URL: https://www.kyclookup.com/uncategorized/what-is-financial-crime/.

[71] Law Insider. *Terrorist financing definition*. Last accessed on 06-12-2021. URL: https://www.lawinsider.com/dictionary/terrorist-financing.

[72] Legal Dictionary. *Fraud*. Last accessed on 07-12-2021. URL: https://legaldictionary.net/fraud/.

[73] Lesouple, J. et al. "Generalized Isolation Forest for Anomaly Detection". In: *Pattern Recognition Letters* 149 (2021), pp. 109–119.

[74] Li, Z., Chen, W., and Pei, D. "Robust and Unsupervised KPI Anomaly Detection Based on Conditional Variational Autoencoder". In: *IEEE 37th International Performance Computing and Communications Conference (IPCCC)* (2018).

[75] Lin, S. et al. "Anomaly Detection for Time Series Using VAE-LSTM Hybrid Model". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020).

[76] Liu, F.T., Ting, K.M., and Zhou, Z.-H. "Isolation Forest". In: *2008 Eighth IEEE International Conference on Data Mining* (2008), pp. 413–422.

[77] Liu, Y. et al. "Self-adversarial variational autoencoder with spectral residual for time series anomaly detection". In: *Neurocomputing* 458 (2021), pp. 349–363.

[78] Ljubas, Z. *Spain Losses Millions in a 'Carousel Fraud' Case*. Last accessed on 15-03-2022. URL: https://www.occrp.org/en/daily/14433-spain-losses-millions-in-a-carousel-fraud-case.

[79] Lookman-Sithic, H. and Balasubramanian, T. "Survey of Insurance Fraud Detection Using Data Mining Techniques". In: *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* 2.3 (2013), pp. 62–65.

[80] Machine Learning Tutorial. *Autoencoders*. Last accessed on 26-01-2022. URL: https://sci2lab.github.io/ml_tutorial/autoencoder/.

[81] Malhotra, P. et al. "Long Short Term Memory Networks for Anomaly Detection in Time Series". In: *23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (2015), pp. 89–94.

[82] Merriam-Webster. *Fraud*. Last accessed on 07-12-2021. URL: https://www.merriam-webster.com/dictionary/fraud.

[83] Mikulski, B. *Smoothing time series in Python using Savitzky–Golay filter*. Last accessed on 24-12-2021. URL: https://www.mikulskibartosz.name/smoothing-time-series-in-python-using-savitzky-golay-filter/.

[84] Muscat, C. *Grand Theft Europe: Malta's role in a scam worth €50 billion annually*. Last accessed on 13-01-2022. URL: https://theshiftnews.com/2019/05/07/grand-theft-europe-maltas-role-in-a-scam-worth-e50-billion-annually/.

[85] Ng, A. "Sparse autoencoder". In: *CS294A Lecture notes* (2011), pp. 1–19.

[86] Niu, X., Wang, L., and Yang, X. "A Comparison Study of Credit Card Fraud Detection: Supervised versus Unsupervised". In: *arXiv:1904.10604v1* (2019).

[87] Niu, Z, Yu, K., and Wu, X. "LSTM-Based VAE-GAN for Time-Series Anomaly Detection". In: *Sensors* 20 (2020).

[88] Oreilly. *Prior, likelihood, and posterior*. Last accessed on 09-03-2022. URL: https://www.oreilly.com/library/view/machine-learning-with/9781785889936/ff082869-751b-4de3-9a59-edff60ad4e94.xhtml#:~:text=Prior%5C%2C%5C%20likelihood%5C%2C%5C%20and%5C%20posterior%5C%20Bayes%5C%20theorem%5C%20states%5C%20the,probability%5C%20of%5C%20A%5C%20given%5C%20B%5C%2C%5C%20also%5C%20called%5C%20posterior..

[89] Ounacer, S. et al. "Using Isolation Forest in anomaly detection: The case of credit card transactions". In: *Periodicals of Engineering and Natural Sciences* 6.2 (2018), pp. 394–400.

[90] Paul, S. *"Reparameterization" trick in Variational Autoencoders*. Last accessed on 09-03-2022. URL: https://towardsdatascience.com/reparameterization-trick-126062cfd3c3.

[91] Pawar, A.D., Kalavadekar, P.N., and Tambe, S. N. "A Survey on Outlier Detection Techniques for Credit Card Fraud Detection". In: *IOSR Journal of Computer Engineering (IOSR-JCE)* 16.2 (2014), pp. 44–48.

[92] Pereira, J. and Silveira, M. "Learning Representations from Healthcare Time Series Data for Unsupervised Anomaly Detection". In: *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)* (2019), pp. 1–7.

[93] Phua, C. et al. "A Comprehensive Survey of Data Mining-based Fraud Detection Research". In: *CoRR* abs/1009.6119 (2010).

[94] Pickett, K.H.S. and Pickett, J. "Financial Crime Investigation and Control". In: John Wiley & Sons, Inc., 2002. Chap. 1.

[95] Picton, P. "Introduction to Neural Networks". In: Macmillan, 1994. Chap. Chapter 1.

[96] Pourhabibi, T. et al. "Fraud detection: A systematic literature review of graph-based anomaly detection approaches". In: *Decision Support systems* 133 (2020).

[97] Provotar, O. I., Linder, Y. M., and Veres, M. M. "Unsupervised Anomaly Detection in Time Series Using LSTM-Based Autoencoders". In: *IEEE International Conference on Advanced Trends in Information Theory (ATIT)* (2019), pp. 513–517.

[98] Reuters Staff. *FACTBOX - How carousel fraud works*. Last accessed on 10-01-2022. URL: https://www.reuters.com/article/uk-carousel-fraud-britain-factbox-sb-idUKTRE57J43U20090820.

[99] Rezapour, M. "Anomaly Detection using Unsupervised Methods: Credit Card Fraud Case Study". In: *International Journal of Advanced Computer Science and Applications (IJACSA)* 10.11 (2019).

[100] Rocca, J. *Understanding Variational Autoencdoers (VAEs)*. Last accessed on 28-01-2022. URL: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73.

[101] Ryder, N. "Financial Crime in the 21st Century, Law and Policy". In: Edward Elgar, 2011. Chap. 1.

[102] Sak, H., Senior, A., and Beaufays, F. "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling". In: *Interspeech* (2014), pp. 338–342.

[103] Savitzky, A. and Golay, M. J. E. "Smoothing and Differentiation of Data by Simplified Least Squares Procedures". In: *Analytical Chemistry* 36.8 (1964), pp. 1627–1639.

[104] Schafer, R.W. "What Is a Savitzky-Golay Filter?" In: *IEEE Signal Processing Magazine* (July 2011), pp. 111–117.

[105] Schmidhuber, J. "Deep learning in neural networks: An overview". In: *Neural Networks* 61 (2015), pp. 85–117.

[106] Schuster, M. and Paliwal, K. K. "Bidirectional recurrent neural networks". In: *IEEE Transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.

[107] Sentrient. *The 3 Stages of The Terrorism Financing Cycle Explained*. Last accessed on 06-12-2021. URL: https://www.sentrient.com.au/blog/the-3-stages-of-the-terrorism-financing-cycle-explained.

[108] Shafkat, I. *Intuitively Understanding Variational Autoencoders*. Last accessed on 27-01-2022. URL: https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf.

[109] Sharma, A. and Panigrahi, P.K. "A Review of Financial Accounting Fraud Detection based on Data Mining Techniques". In: *International Journal of Computer Applications* 39.1 (2012), pp. 37–47.

[110] Simplilearn. *What is Perceptron: A Beginners Guide for Perceptron*. Last accessed on 24-01-2022. URL: https://www.simplilearn.com/tutorials/deep-learning-tutorial/perceptron.

[111] St Pauls Chambers. *Carousel Fraud Explained*. Last accessed on 10-01-2022. URL: https://www.stpaulschambers.com/expertise/carousel-fraud/#:~:text=Carousel%5C%20Fraud%5C%20Explained%5C%20Carousel%5C%20fraud%5C%2C%5C%20also%5C%20known%5C%20as,member%5C%20states.%5C%20How%5C%20does%5C%20carousel%5C%20VAT%5C%20fraud%5C%20work%5C%3F.

[112] Stat Line. *Geregistreerde criminaliteit; soort misdrijf, regio*. Last accessed on 03-03-2022. URL: https://opendata.cbs.nl/statline/#/CBS/nl/dataset/83648NED/table?ts=1637934795993.

[113] Statology. *Kolmogorov-Smirnov Test in R (With Examples)*. Last accessed on: 25-04-2022. URL: https://www.statology.org/kolmogorov-smirnov-test-r/.

[114] Summitto. *Reconstructing a €9 million VAT fraud scheme*. Last accessed on 15-03-2022. URL: https://blog.summitto.com/posts/reconstructing_a_9_million_vat-_fraud_scheme/.

[115] Svozil, D., Kvasnicka, V., and Pospichal, J. "Introduction to multi-layer feed-forward neural networks". In: *Chemometrics and Intelligent Laboratory Systems* 39.1 (1997), pp. 43–62.

[116] Techopedia. *What Does Anomaly Detection Mean?* Last accessed on 11-01-2022. URL: https://www.techopedia.com/definition/30297/anomaly-detection.

[117] ThoughtCo. *Definition and Examples of Fraud*. Last accessed on 08-12-2021. URL: https://www.thoughtco.com/fraud-definition-and-examples-4175237.

[118] United Nations: Office on Drugs and Crime. *Money Laundering*. Last accessed on 26-11-2021. URL: https://www.unodc.org/unodc/en/money-laundering/overview.html.

[119] United States Department Of Justice: The United States Attorney's Office District of Alaska. *Financial Fraud Crime*. Last accessed on 08-12-2021. URL: https://www.justice.gov/usao-ak/financial-fraud-crimes.

[120] Vanhoeyveld, J., Martens, D., and Peeters, B. "Value-added tax fraud detection with scalable anomaly detection techniques". In: *Applied Soft Computing (pre-proof)* (2020).

[121] Weng, L. *From Autoencoder to Beta-VAE*. Last accessed on 01-02-2022. URL: https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html.

[122] Wronski, B. *Study of smoothing filters – Savitzky-Golay filters*. Last accessed on 21-01-2022. URL: https://bartwronski.com/2021/11/03/study-of-smoothing-filters-savitzky-golay-filters/.

[123] Xu, D. et al. "An Improved Data Anomaly Detection Method Based on Isolation Forest". In: *10th International Symposium on Computational Intelligence and Design (ISCID)* (2017), pp. 287–291.

[124] Xu, H. et al. "Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications". In: *International World Wide Web Conferences Steering Committee* (2018), pp. 187–196.

[125] Your Europe. *VAT rules and rates*. Last accessed on 13-12-2021. URL: https://europa.eu/youreurope/business/taxation/vat/vat-rules-rates/index_en.htm.

[126] Zaiontz, C. *Two-Sample Kolmogorov-Smirnov Test*. Last accessed on: 25-04-2022. URL: https://www.real-statistics.com/non-parametric-tests/goodness-of-fit-tests/two-sample-kolmogorov-smirnov-test/.

[127] Zhai, S. and Zhang, Z. "Semisupervised Autoencoder for Sentiment Analysis". In: *AAAI Conference on Artificial Intelligence* (2016).

[128] Zhang, C. et al. "VELC: A New Variational AutoEncoder Based Model for Time Series Anomaly Detection". In: *arXiv: 1907.01702v2* (2020).

11 Appendix

Normalized balance over time for the fraud set

Two contract numbers are not shown since only one transactions is available in those contract numbers, making it impossible to plot any figure. In the figures below the RMoF behavior of high peaks and then return back to zero is visible. However this is not visible in all contract numbers. The reason for this is that the bank classified business numbers as RMoF and not the contract numbers. Since for one business number multiple contract numbers are possible, it can be the case that some contract numbers do not show the behavior of RMoF. These are also the contract numbers were not many transactions have occurred.

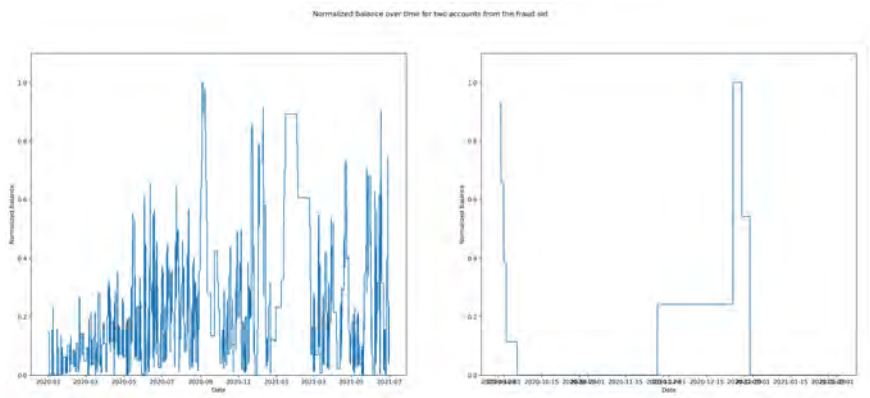

Figure 54: Balance over time for two contract numbers from fraud set

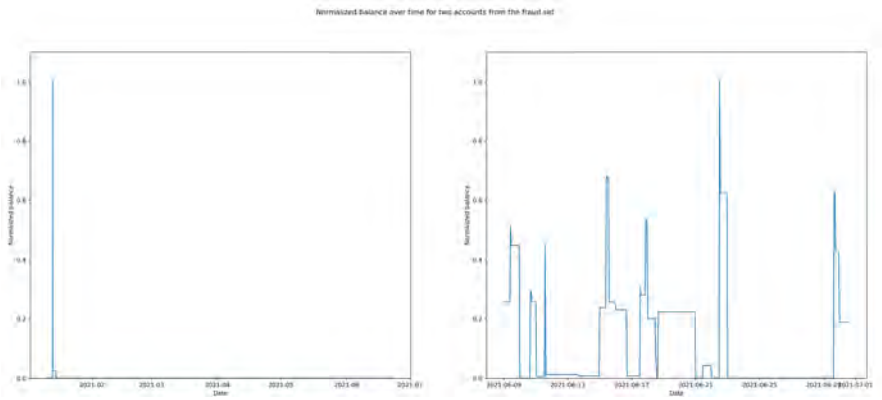

Figure 55: Balance over time for two contract numbers from fraud set

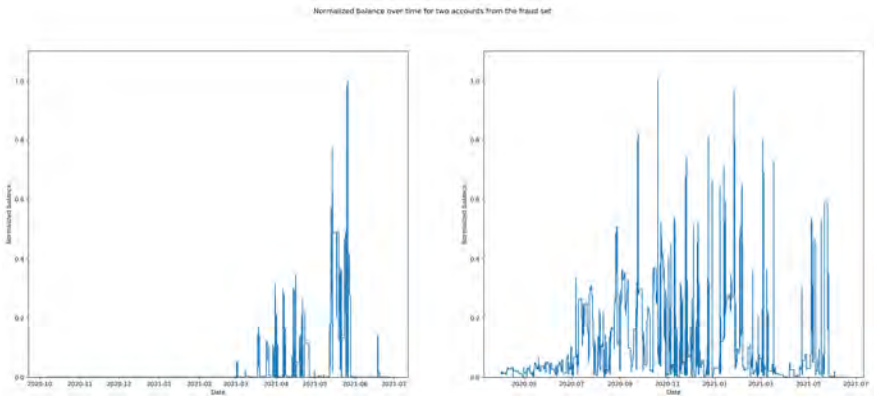

Figure 56: Balance over time for two contract numbers from fraud set

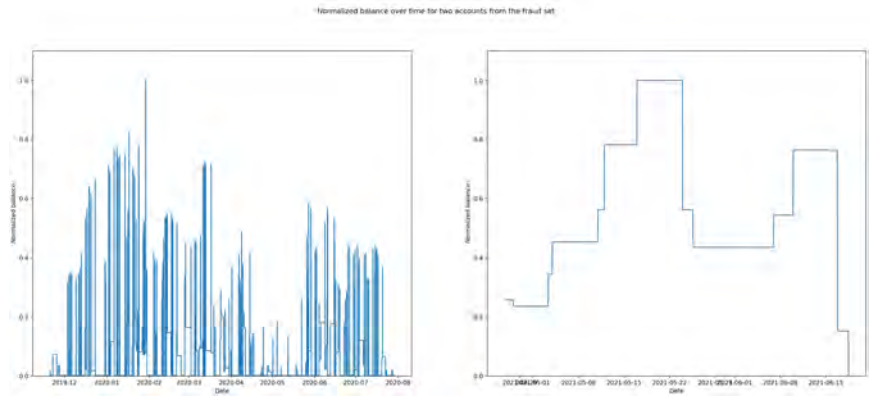

Figure 57: Balance over time for two contract numbers from fraud set

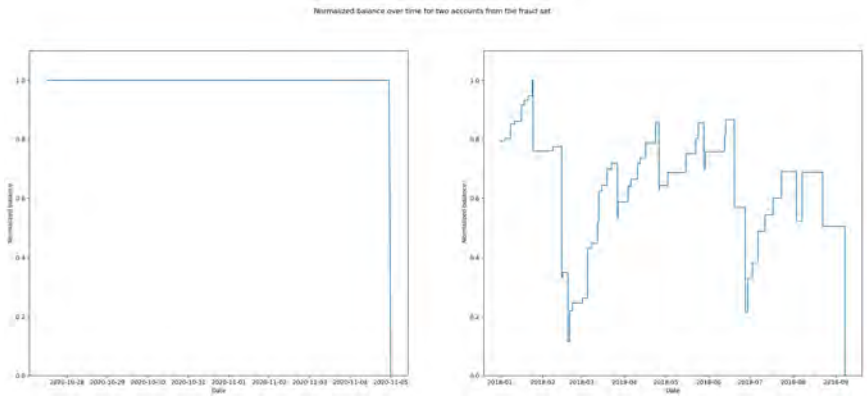

Figure 58: Balance over time for two contract numbers from fraud set

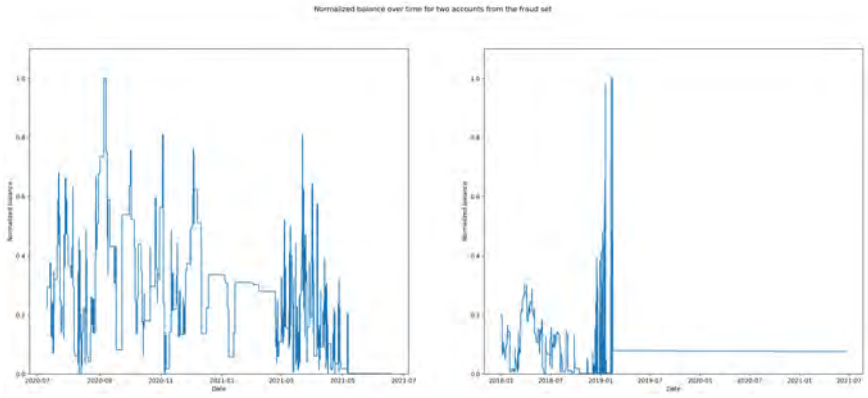

Figure 59: Balance over time for two contract numbers from fraud set

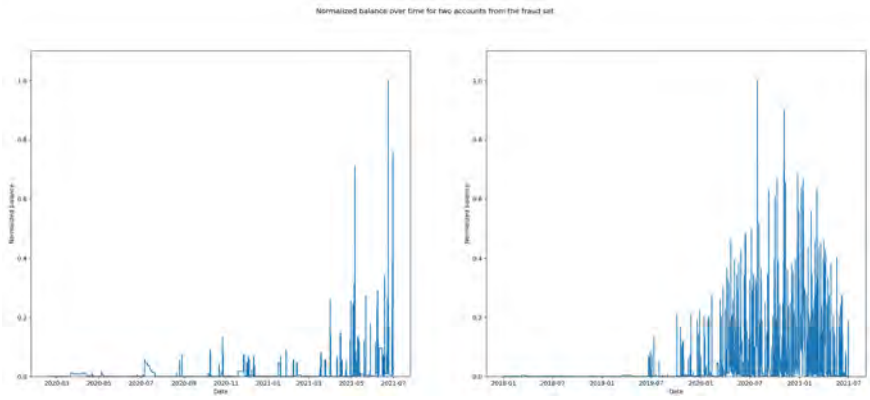

Figure 60: Balance over time for two contract numbers from fraud set

Parameters for training

The exact parameter settings per VAE model and β-VAE model can be seen below.

LSTM layers	128
latent space dimensions	20
batch size	32
learning rate	0.001
epochs	300

Table 26: Parameter settings of VAE with one feature

LSTM layers	128
latent space dimensions	25
batch size	32
learning rate	0.001
epochs	300

Table 27: Parameter settings of VAE with two features

LSTM layers	128
latent space dimensions	15
batch size	32
learning rate	0.001
epochs	300

Table 28: Parameter settings of VAE with three features

LSTM layers	128
latent space dimensions	10
batch size	32
learning rate	0.001
epochs	300

Table 29: Parameter settings of VAE with four features (13 dimensions)

LSTM layers	128
latent space dimensions	15
batch size	32
learning rate	0.001
epochs	300

Table 30: Parameter settings of VAE with four features (5 dimensions)

LSTM layers	128
latent space dimensions	5
batch size	32
learning rate	0.001
epochs	300

Table 31: Parameter settings of VAE with five features

LSTM layers	128
latent space dimensions	15
β	2
batch size	32
learning rate	0.001
epochs	300

Table 32: Parameter settings of β-VAE with three features, with $\beta = 2$

LSTM layers	128
latent space dimensions	15
β	5
batch size	32
learning rate	0.001
epochs	300

Table 33: Parameter settings of β-VAE with three features, with $\beta = 5$

Results of VAE model with one feature

The three windows which contained the median outlier score are shown in Figure 61. Here the VAE with only one feature is used, namely balance over time.

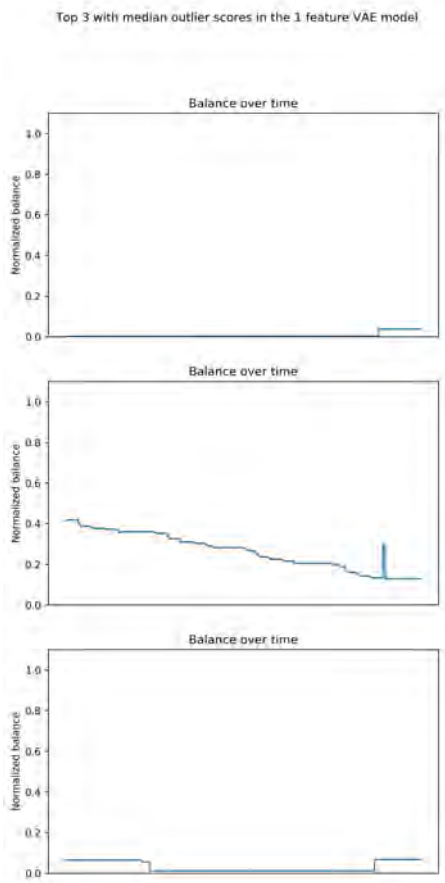

Figure 61: Three windows with a median outlier score, for one feature combination VAE model

Results of VAE model with two features

In Figure 62 the top three windows with a median outlier score are presented for the VAE with two features. The features were balance over time and Rapid Movement of Funds (RMoF). What can be observed is that the balance changes very slightly over time and in RMoF almost no peaks are visible.

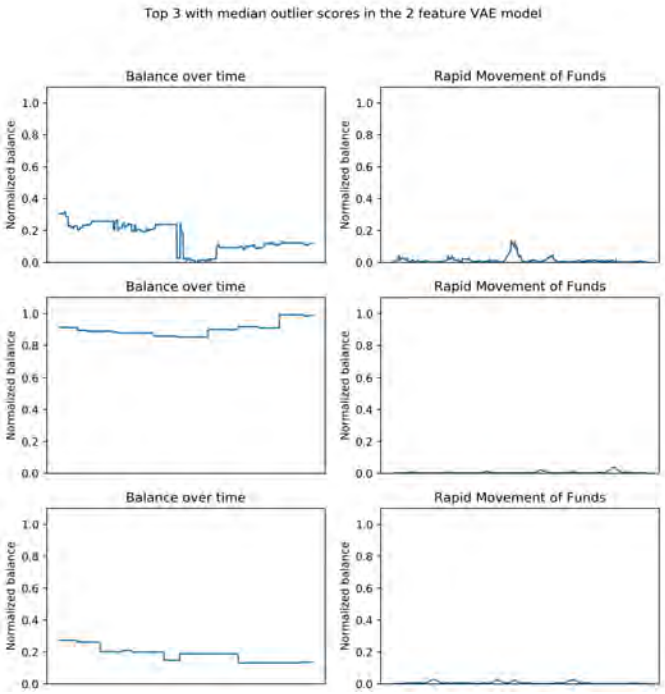

Figure 62: Three windows with a median outlier score, for two feature combination VAE model

Results of VAE model with three features

For the three feature VAE model the following median outlier scores were obtained, which are presented in Figure 63. Here it is visible that the balance changes very little, the RMoF behavior shows no high peaks and there are almost no cross-border transactions.

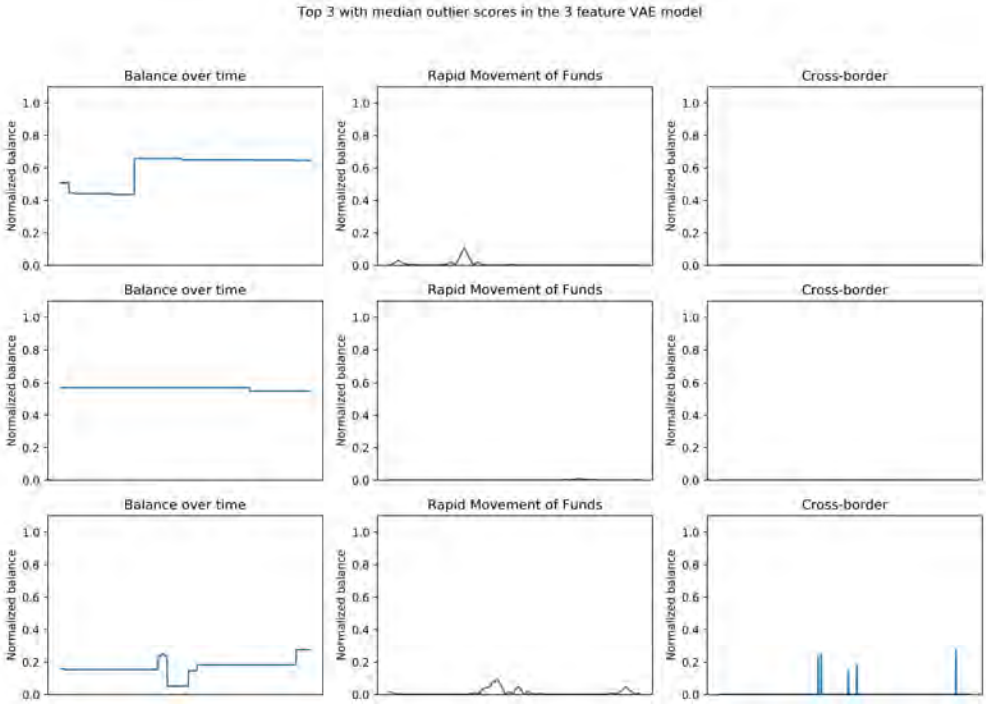

Figure 63: Three windows with a median outlier score, for three feature combination VAE model

Results of VAE model with four features (13 dimensions)

The three windows with the highest outlier scores are presented in Figure 64. This feature combination model has 13 dimensions as input, which implies that per window 13 figures can be made. Presenting 13 figures per window is too overwhelming. Therefore, the decision was made to choose the three features which display the clearest pattern. There are figures made which show all 13 features per window and these are presented later in this section. For the four feature model the RMoF and credit_5 displayed the most activity. So these were along side the balance over time plotted for the three windows with the highest outlier scores. What can be observed from the figure is that a sudden drop in the balance, leads to a high peak in RMoF. The common pattern between the three windows is visible in the rhythm of credit_5. In this feature regularly high peaks are observed. This would indicated that many transaction occur with that specific counterparty. Although, the fifth credit counterparty displays not 'normal' behavior this is not represented in the balance. For fraud one would expect regular high peaks which return to zero, but in this case the balance does not change drastically.

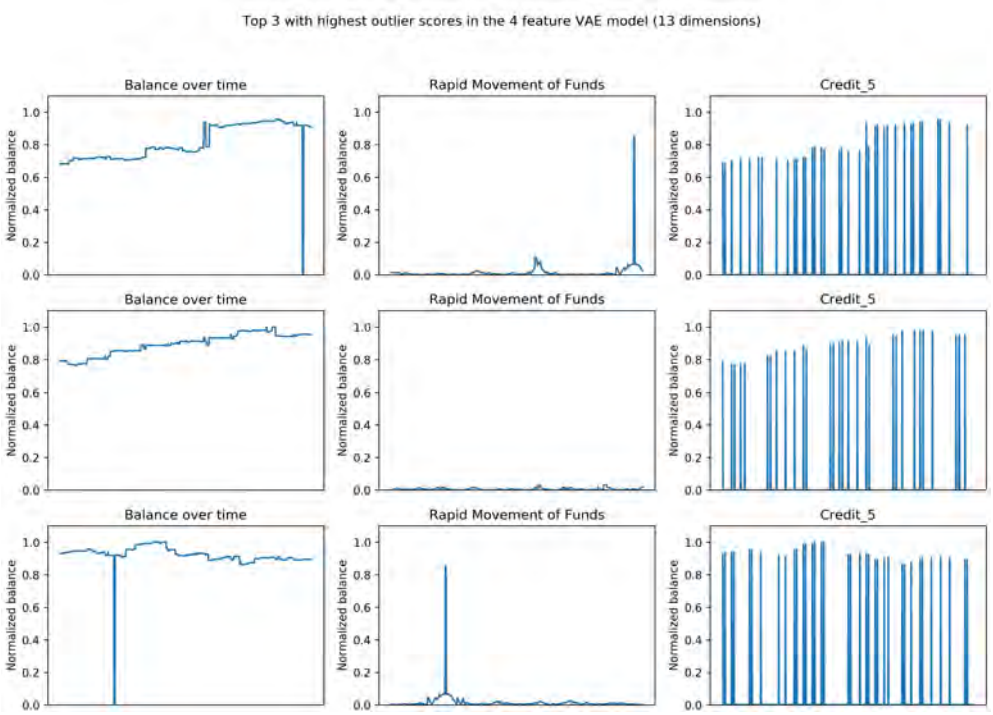

Figure 64: Three windows with the highest outlier score, for four feature (13 dimensions) combination VAE model

The median outlier scores were also decided for three windows. As was the same in the other feature combinations, these outlier scores do not display a lot of activity. The balance over time changes slightly and the other features show very little activity. For these three windows all the 13 feature rhythms are presented further down in this section.

Next, the three windows with the lowest outlier scores are retrieved. The expectation is that these three windows display an almost flat signal for all the features. This is the pattern that is observed in all the feature combinations so far. Again the features balance over time, RMoF and credit_5 are shown in the figure. Figure 65 supports the expectation of observing an almost flat line for the features. In the figure only three features are shown, an complete overview of all 13 features for the windows with the lowest outlier scores is given later in this section.

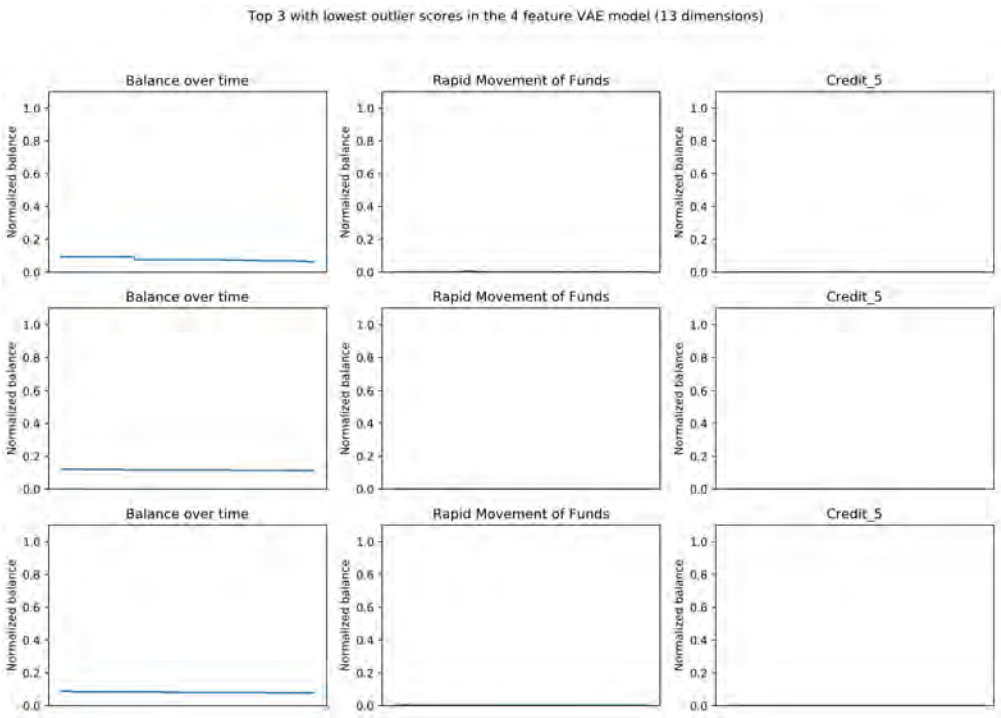

Figure 65: Three windows with the lowest outlier score, for four feature (13 dimensions) combination VAE model

The four feature VAE model with 13 dimensions has per window 13 features which can be plotted. For the top three windows with the highest outlier scores this is presented in Figures 66, 67 and 68. The three windows with the median outlier score are displayed in Figures 69, 70 and 71. Lastly, the three windows with the lowest outlier scores are shown in Figures 72, 73 and 74.

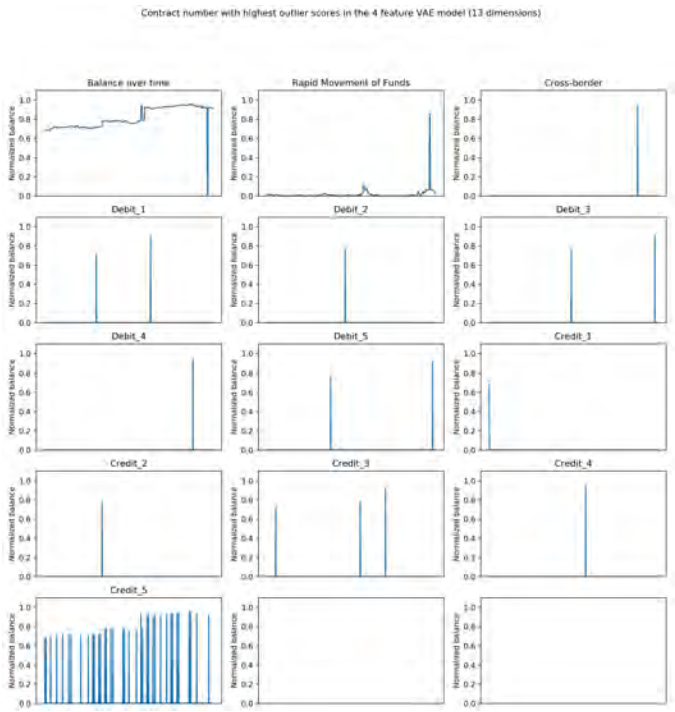

Figure 66: The windows for all features for the contract number with the highest outlier score

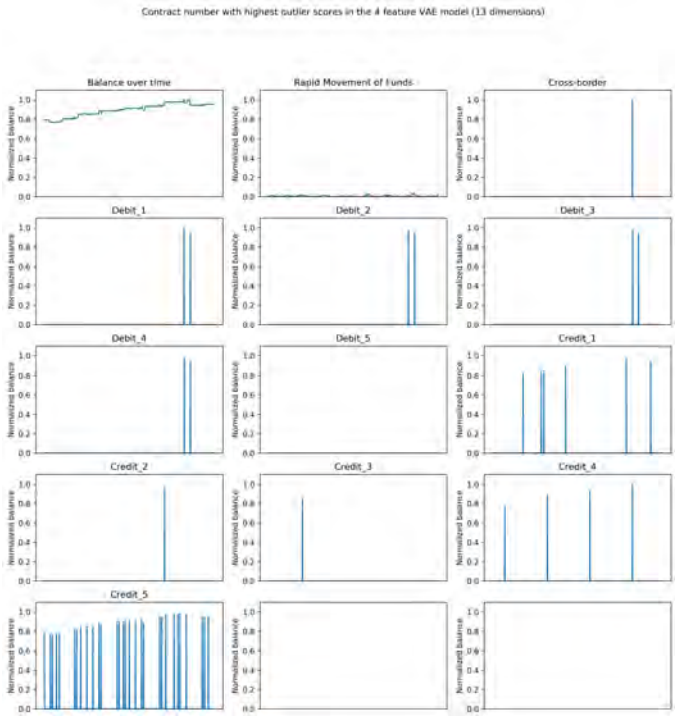

Figure 67: The windows for all features for the contract number with the second highest outlier score

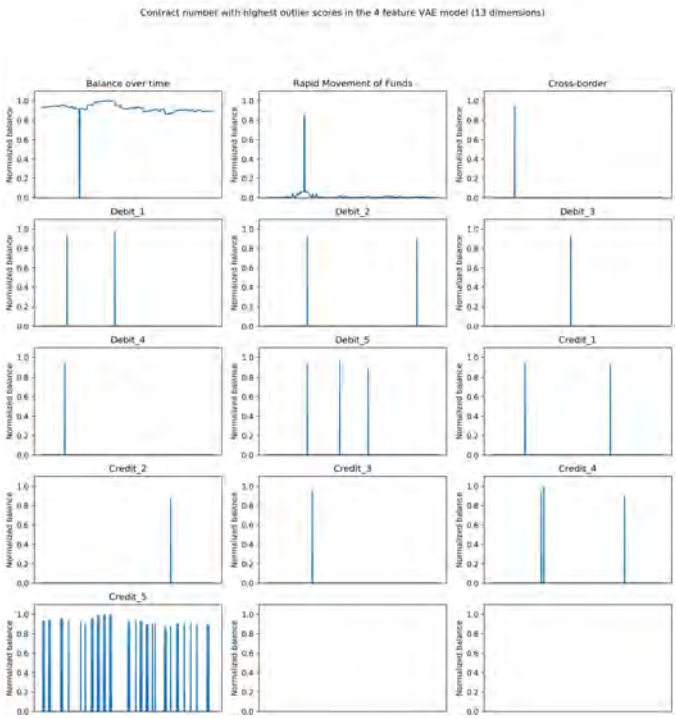

Figure 68: The windows for all features for the contract number with the third highest outlier score

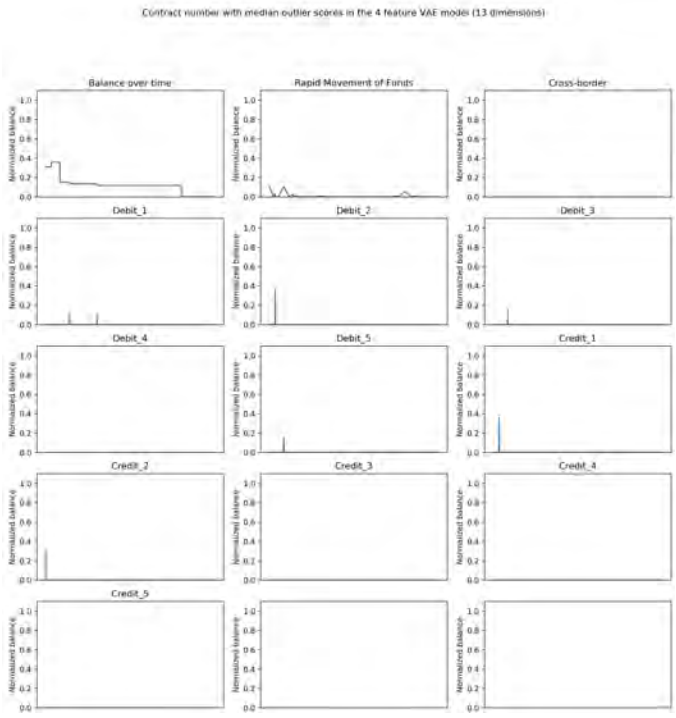

Figure 69: The windows for all features for the contract number with the median outlier score

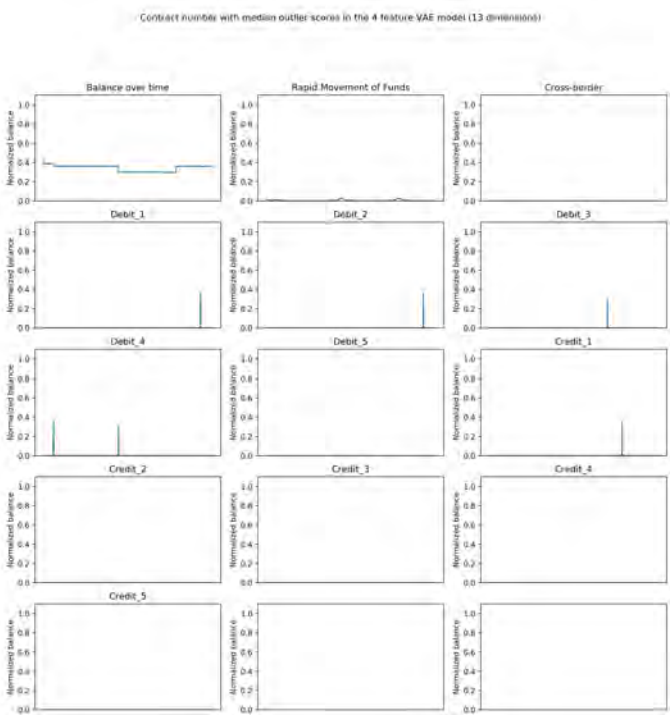

Figure 70: The windows for all features for the contract number with the second median outlier score

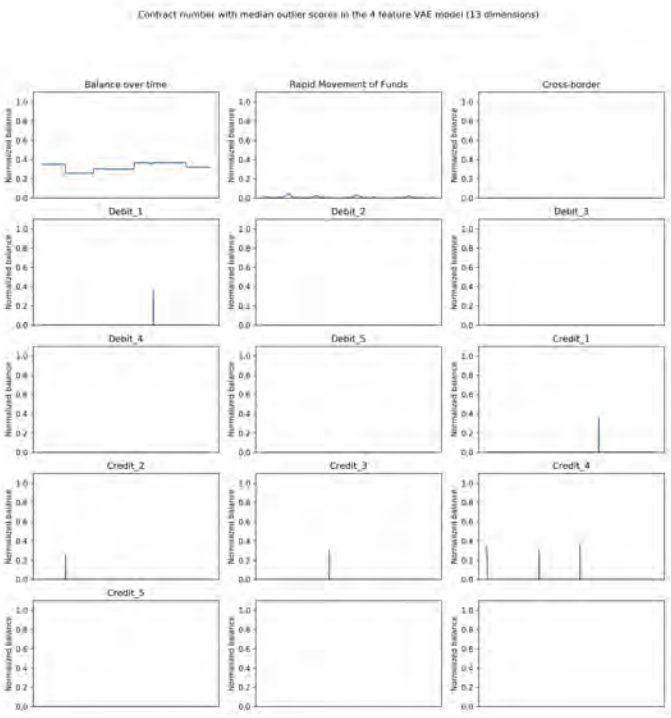

Figure 71: The windows for all features for the contract number with the third median outlier score

Figure 72: The windows for all features for the contract number with the lowest outlier score

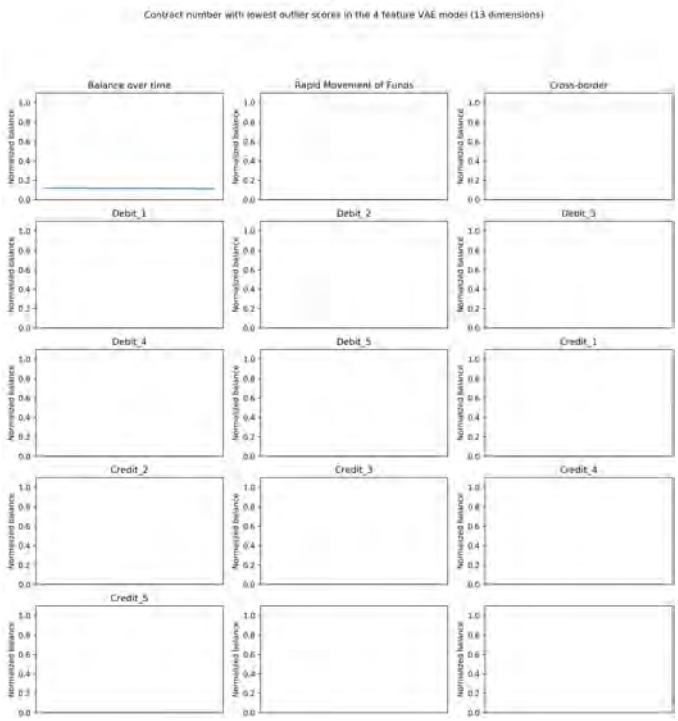

Figure 73: The windows for all features for the contract number with the second lowest outlier score

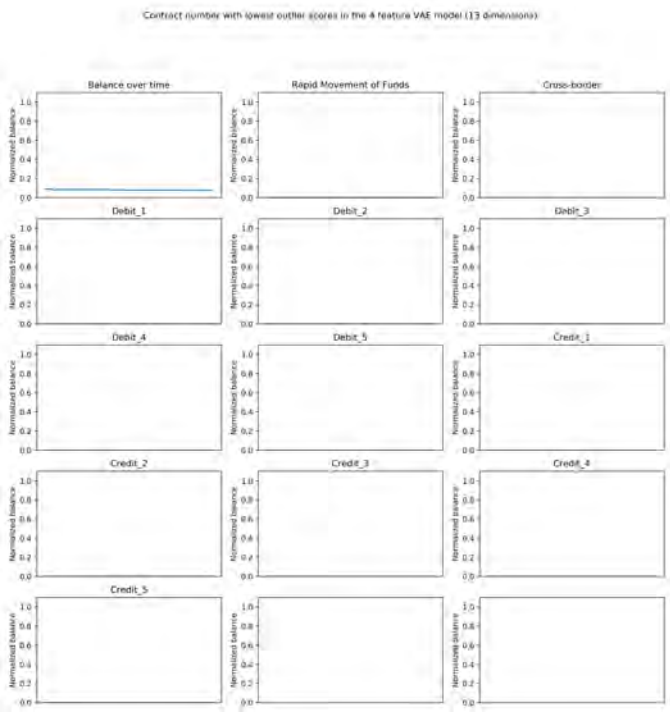

Figure 74: The windows for all features for the contract number with the third lowest outlier score

Results of VAE model with four features (5 dimensions)

The three windows with the highest outlier scores are retrieved and plotted. This model has 5 dimensions, which means that per window five figures can be made. It was decided to show only three figure per window, the features which showed the most clear pattern where chosen. This was balance over time, RMoF and cross-border. From the figure it can be noticed that the highest two windows were also detected in the top three of the three feature model. This contract number apparently makes a lot of cross-border transactions. The third windows displays much fewer cross-border transactions. Also, a drop in the balance results in a peak in the RMoF feature. Although a lot of cross-border transaction occur within these windows, the balance does not display the pattern which is expected for VAT/carousel fraud.

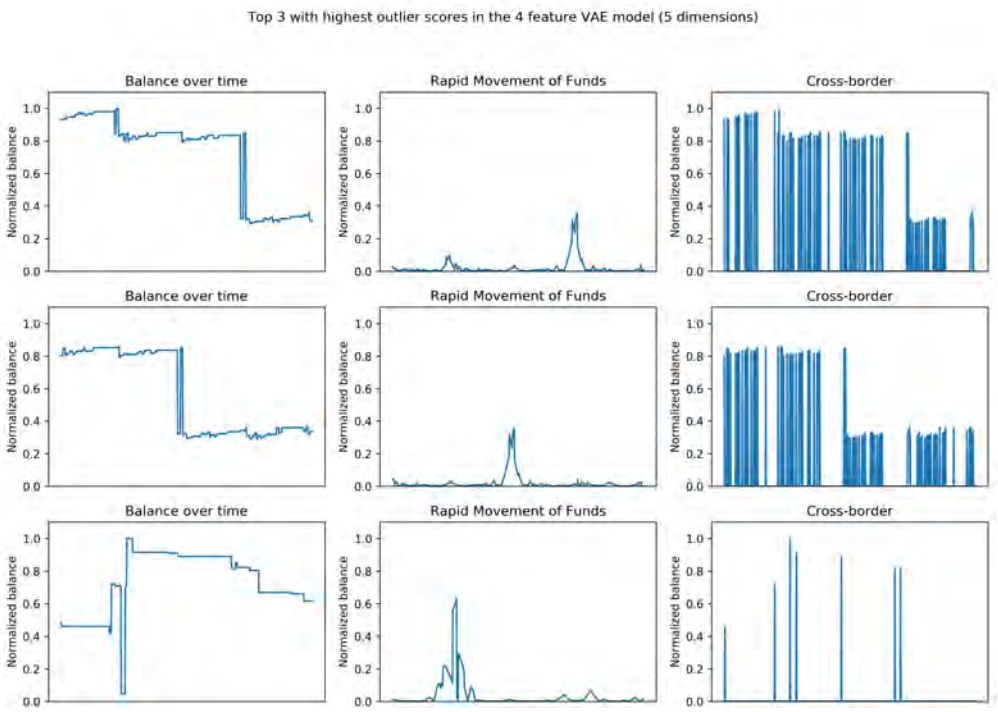

Figure 75: Three windows with the highest outlier score, for four feature (5 dimensions) combination VAE model

The three windows for the median outlier scores were also determined, these are shown further down in this section. For the three windows it was noticed that very little activity occurred in the balance and a small peak at the first debit counterparty.

The last figure for this feature combination is that of the three windows with the lowest outlier score. This is presented in Figure 76. This figure support the expectation that for low outlier scores no activity or almost no activity is visible in the different features. Later in this section figures can be found which show the rhythms of all five features per window.

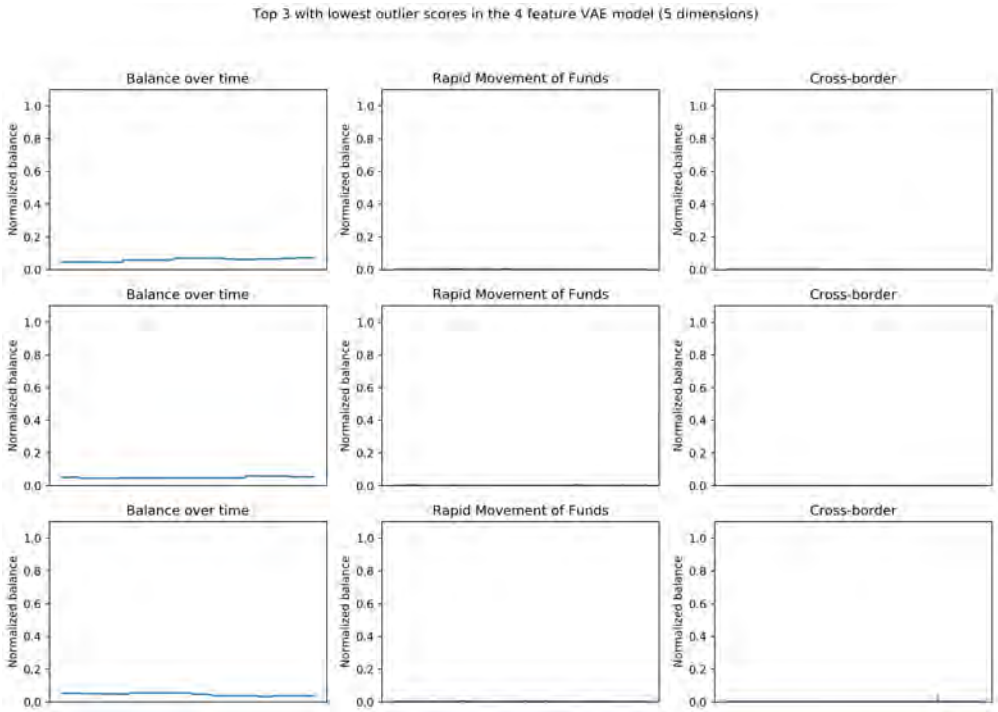

Figure 76: Three windows with the lowest outlier score, for four feature (5 dimensions) combination VAE model

The four feature VAE model with 5 dimensions has per window 5 features which can be plotted. For the top three windows with the highest outlier scores this is presented in Figures 77, 78 and are displayed in Figures 80, 81 and 82. Lastly, the three windows with the lowest outlier scores are shown in Figures 83, 84 and 85.

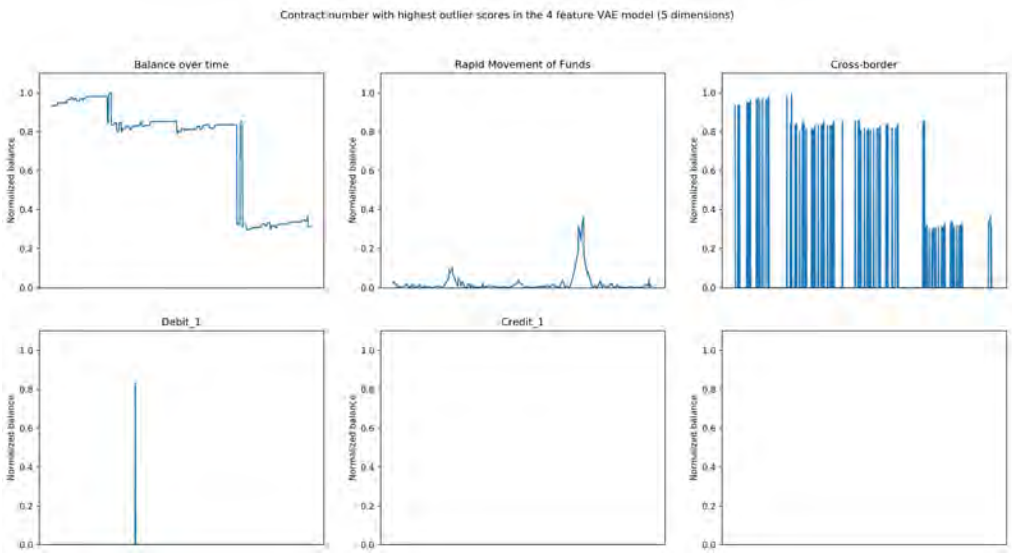

Figure 77: The windows for all features for the contract number with the highest outlier score

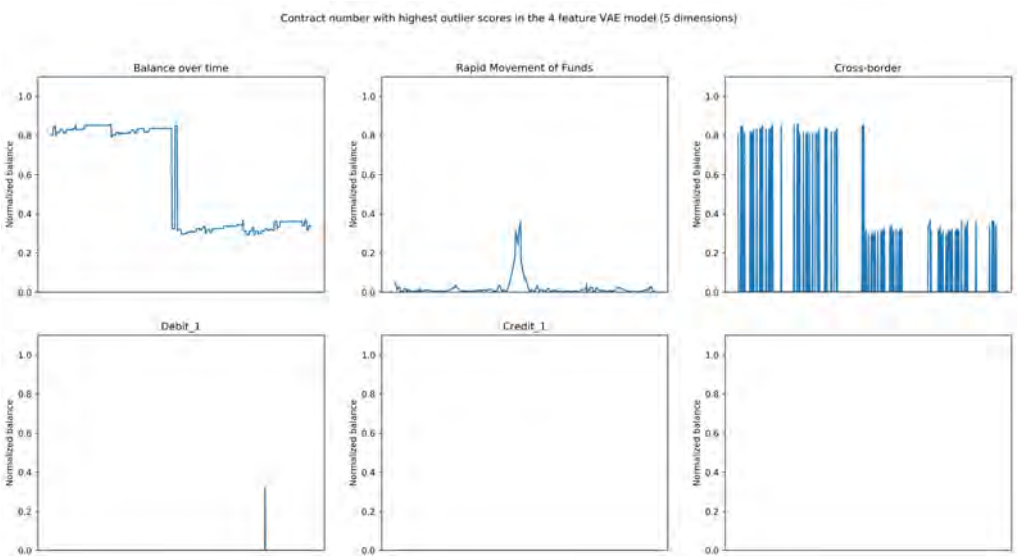

Figure 78: The windows for all features for the contract number with the second highest outlier score

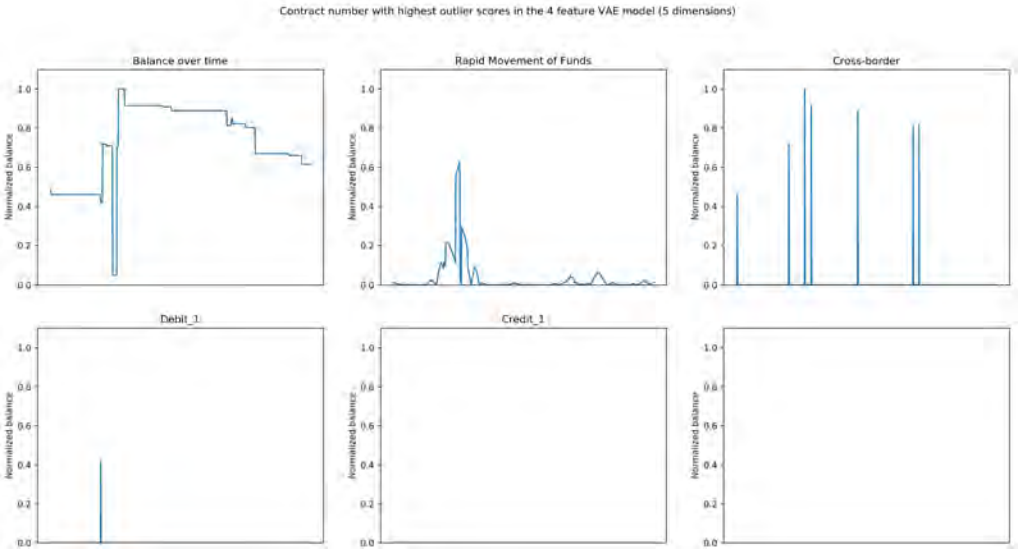

Figure 79: The windows for all features for the contract number with the third highest outlier score

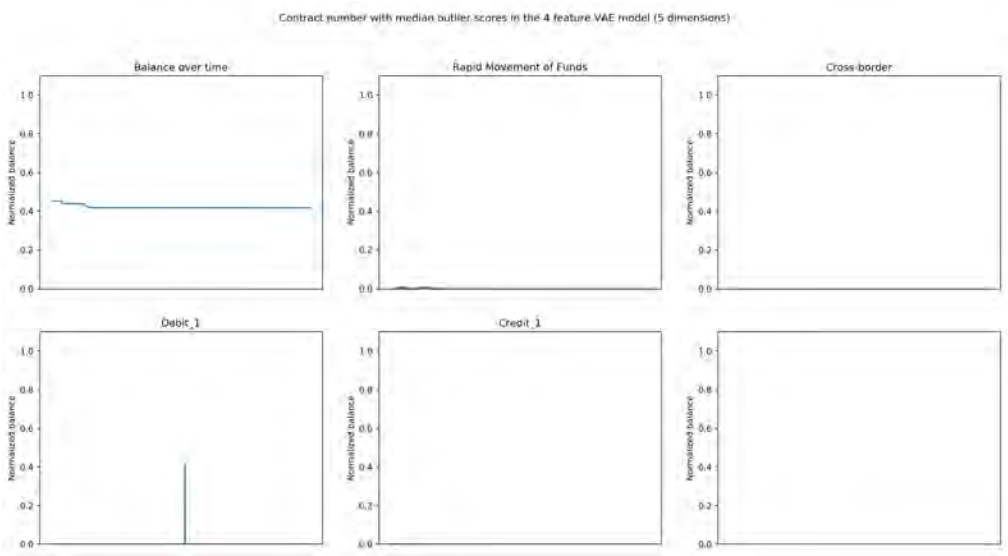

Figure 80: The windows for all features for the contract number with the median outlier score

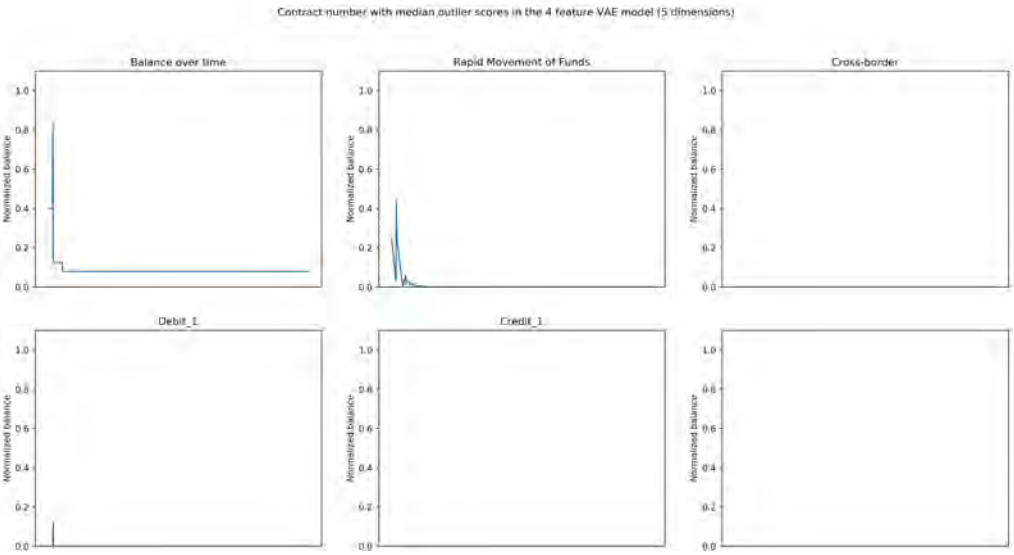

Figure 81: The windows for all features for the contract number with the second median outlier score

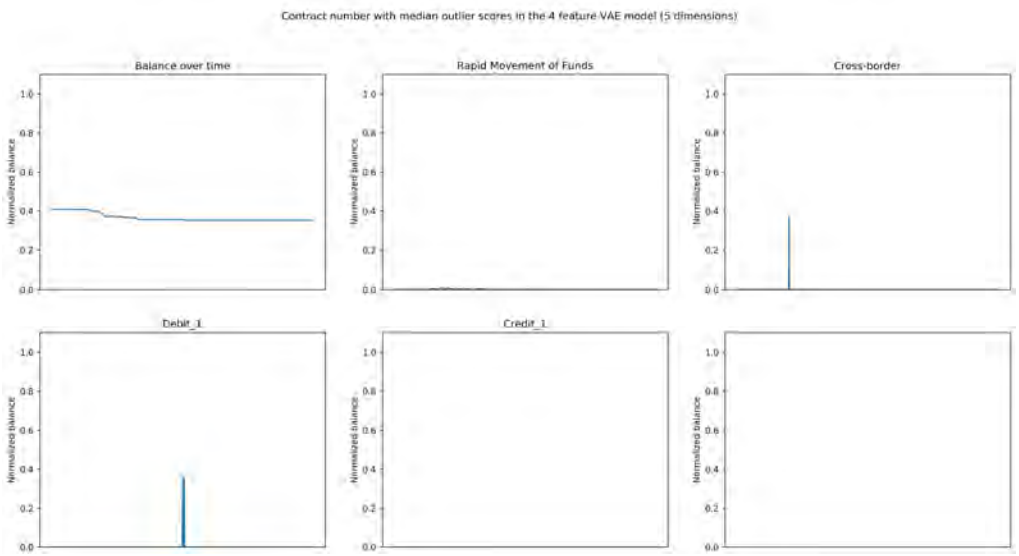

Figure 82: The windows for all features for the contract number with the third median outlier score

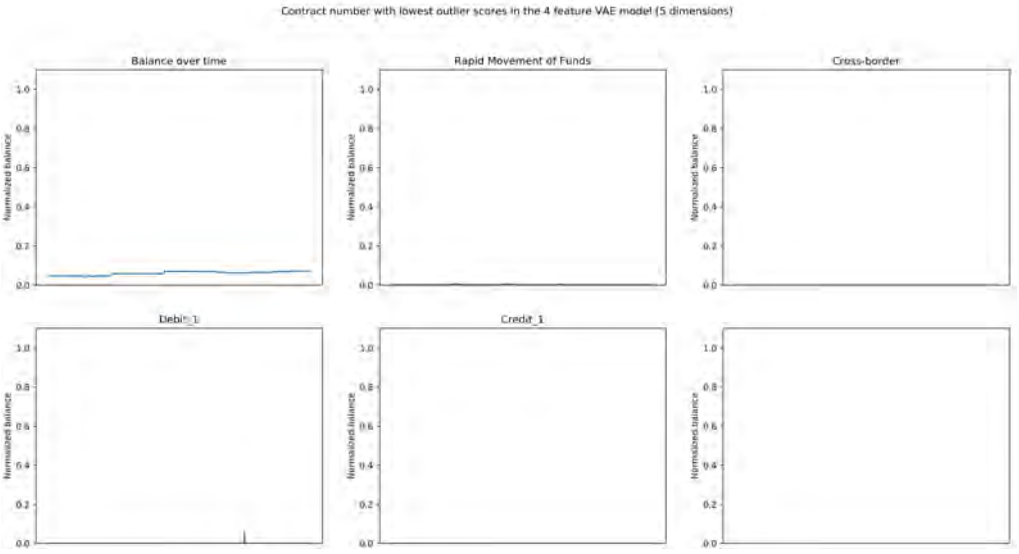

Figure 83: The windows for all features for the contract number with the lowest outlier score

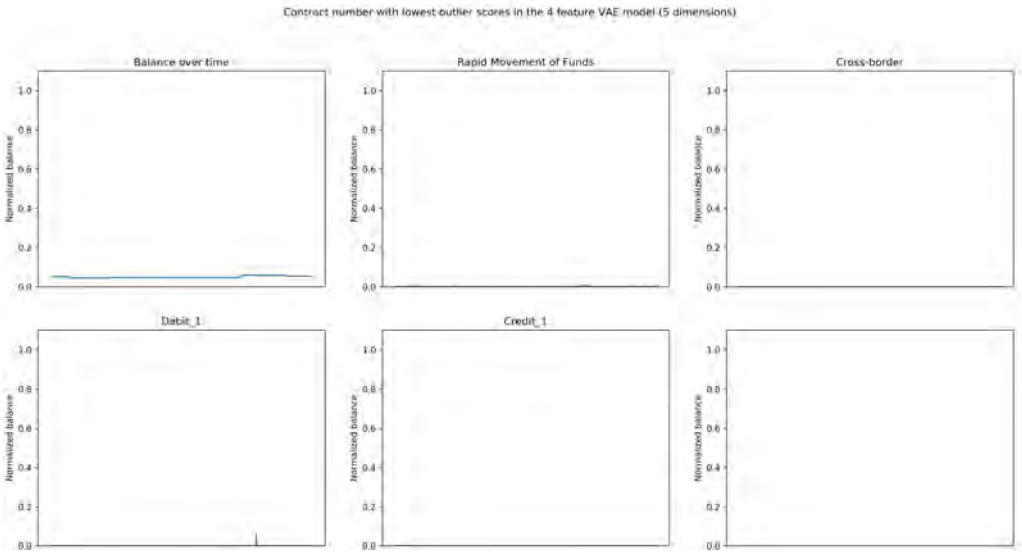

Figure 84: The windows for all features for the contract number with the second lowest outlier score

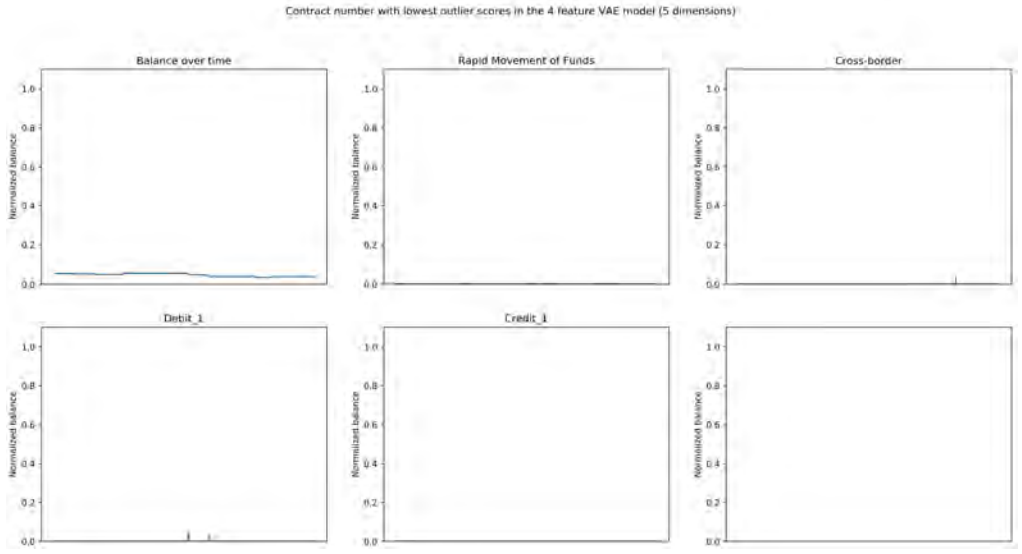

Figure 85: The windows for all features for the contract number with the third lowest outlier score

Results of VAE model with five features

The three windows with the highest outlier scores are shown in Figure 86. This model has 14 dimensions as input. Therefore, per window a total of 14 feature figures can be made. The decision was made to select the three features which display the clearest pattern, and present these for the top three windows. Further down in this section all 14 features per window are shown. For the five feature model the balance over time, RMof and credit_5 displayed the most activity. What can be observed is that for the top two windows a lot of activity is happening in the feature credit_5. In the third window this is less the case. Even though this feature displays a very regular pattern, the balance does not change drastically. This can be expected, since credit_5 display the rhythm of the fifth counterparty in terms of mutation amount within that window. Compared to the other four counterparties, less money is transferred in this feature. So, based on these features not necessarily a clear fraud pattern can be observed.

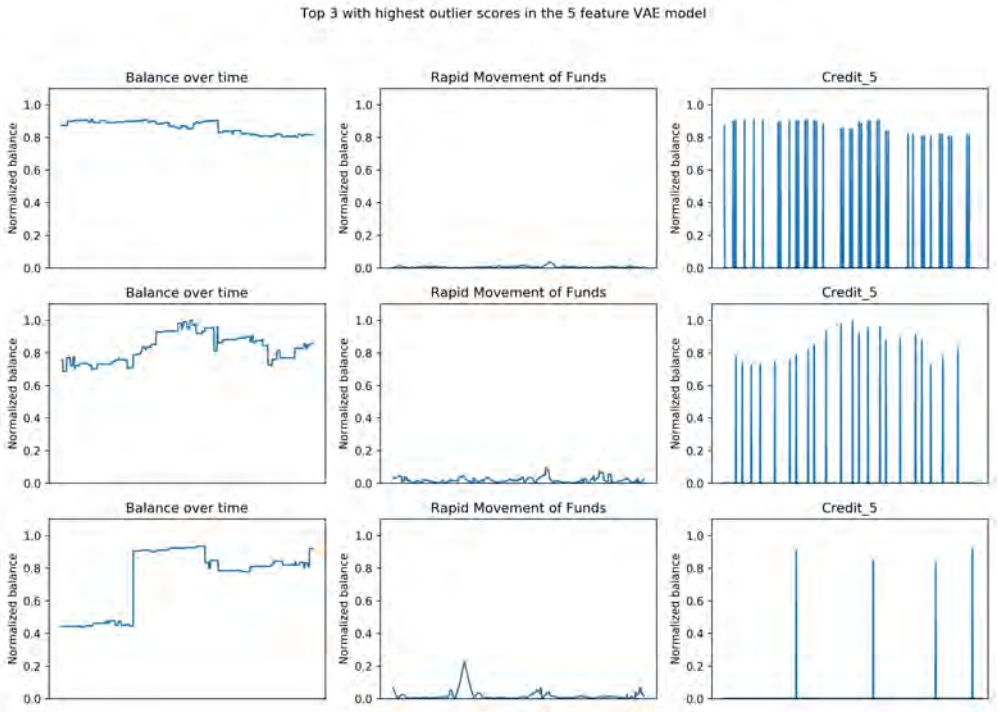

Figure 86: Three windows with the highest outlier score, for five feature combination VAE model

Next, the three windows for the median outlier scores were illustrated. The same with all the other feature combinations, these windows do not display a lot of activity. The balance over time changes slightly and the other features display almost no activity. The three windows with median outlier scores are presented later in this section for all 14 features.

Lastly, the three windows with the lowest outlier scores are plotted. The expectation is that these three windows show an almost flat signal for all 14 features. When investigating the three windows this is indeed the case. The same features as for the highest three outliers are shown in Figure 87. As can be observed from this figure, all features display an almost flat line. An complete overview of all 14 feature for the three windows is presented further down in this section.

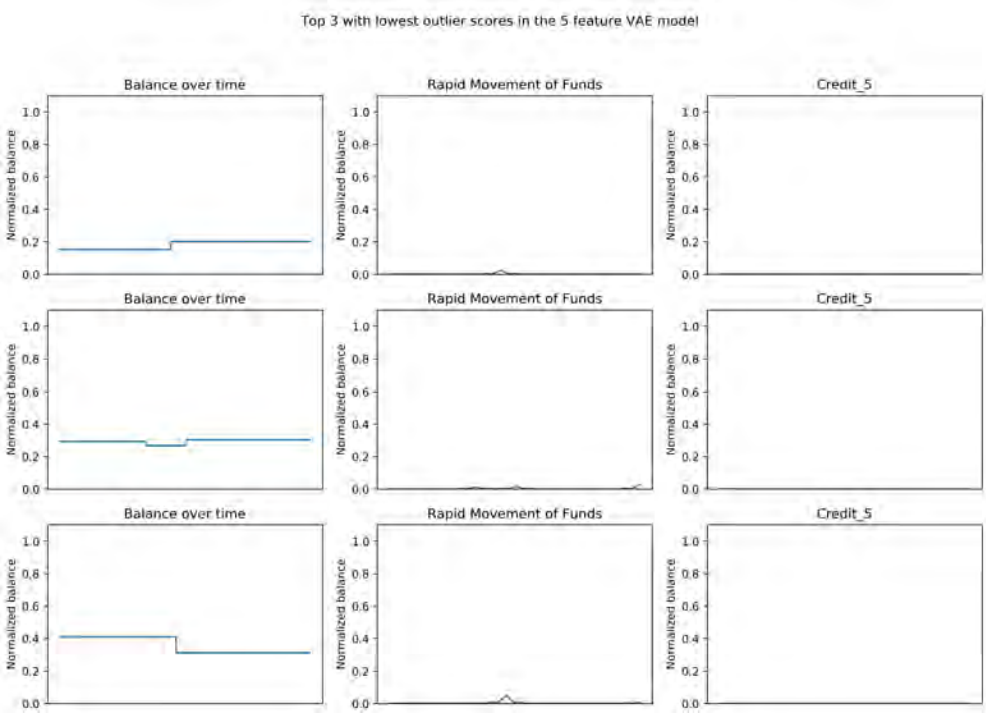

Figure 87: Three windows with the lowest outlier score, for five feature combination VAE model

The five feature VAE model has 14 dimensions, so per window 14 features can be plotted. For the three windows with the highest outlier scores this is presented in Figures 88, 89 and 90. The three windows with the median outlier score are displayed in Figures 91, 92 and 93. Lastly, the three windows with the lowest outlier scores are shown in Figures 94, 95 and 96.

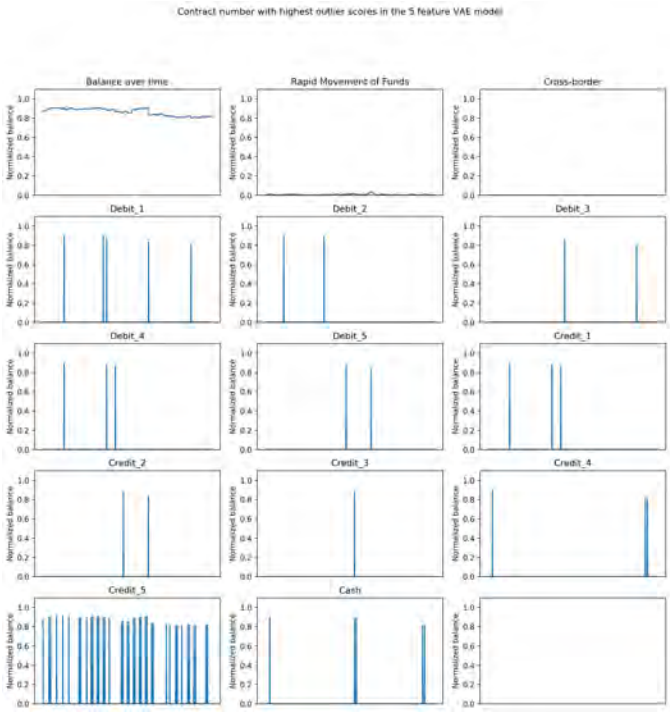

Figure 88: The windows for all features for the contract number with the highest outlier score

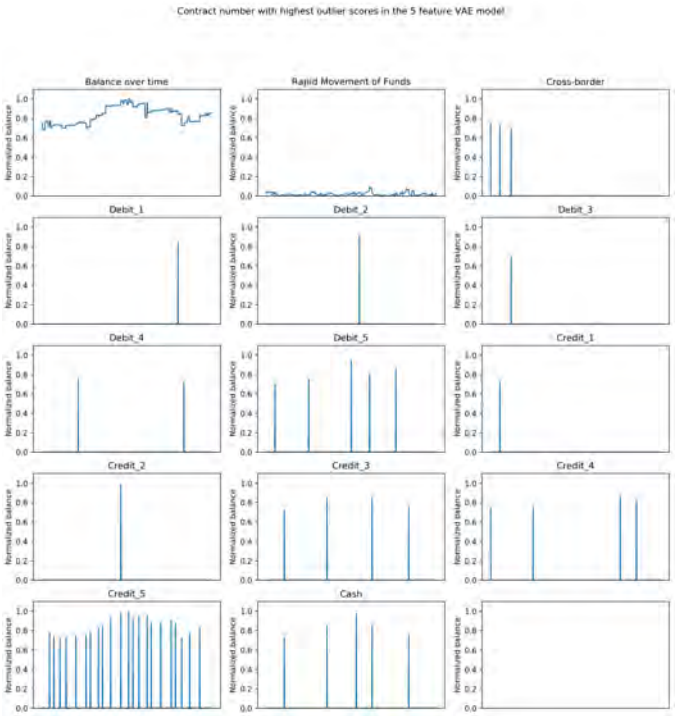

Figure 89: The windows for all features for the contract number with the second highest outlier score

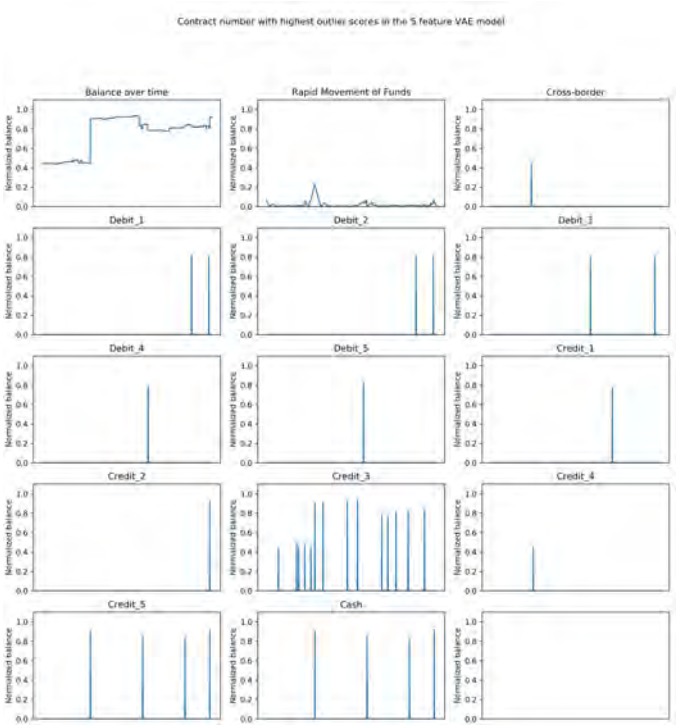

Figure 90: The windows for all features for the contract number with the third highest outlier score

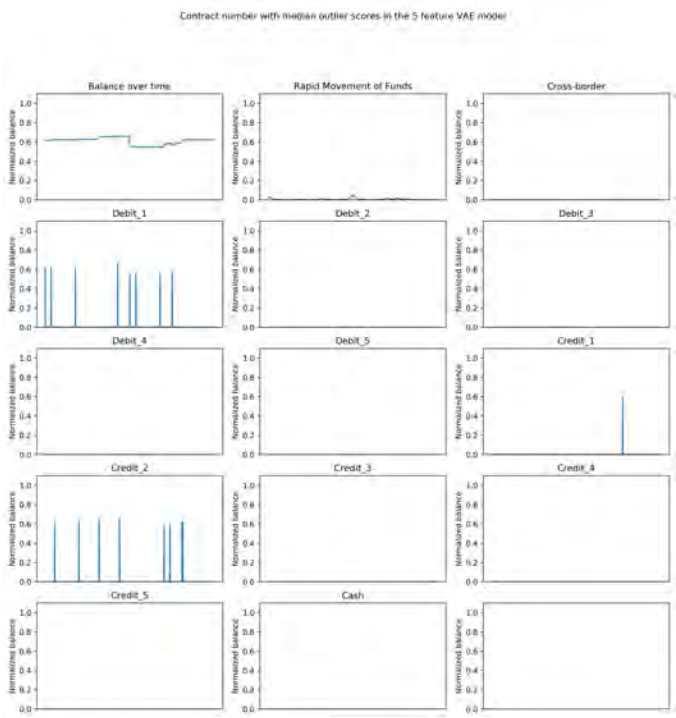

Figure 91: The windows for all features for the contract number with the median outlier score

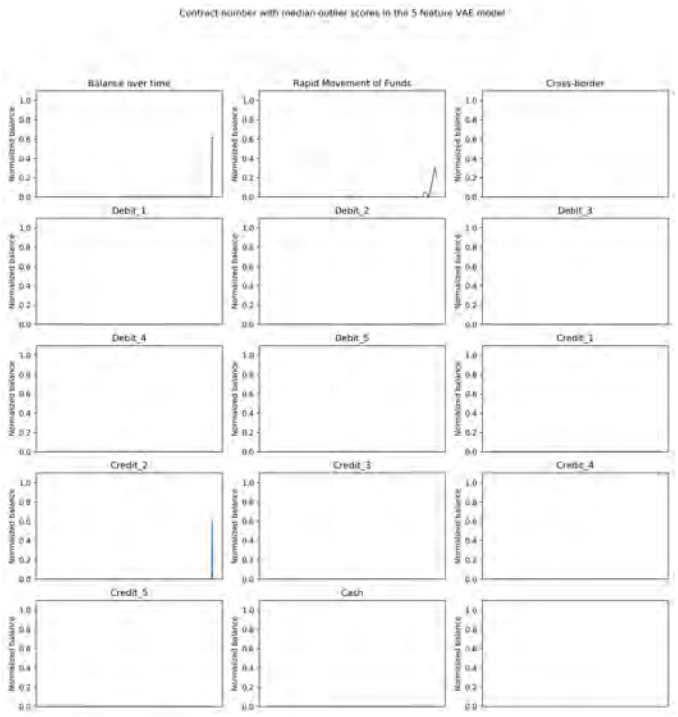

Figure 92: The windows for all features for the contract number with the second median outlier score

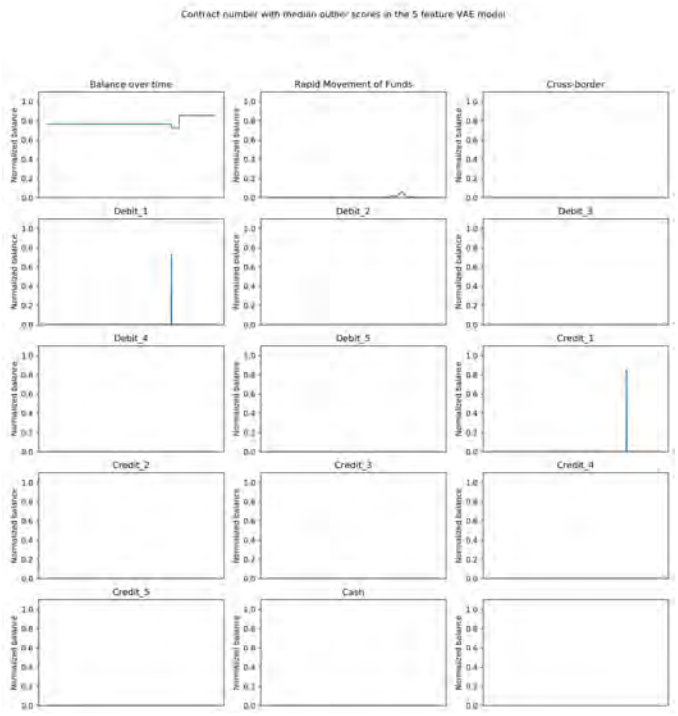

Figure 93: The windows for all features for the contract number with the third median outlier score

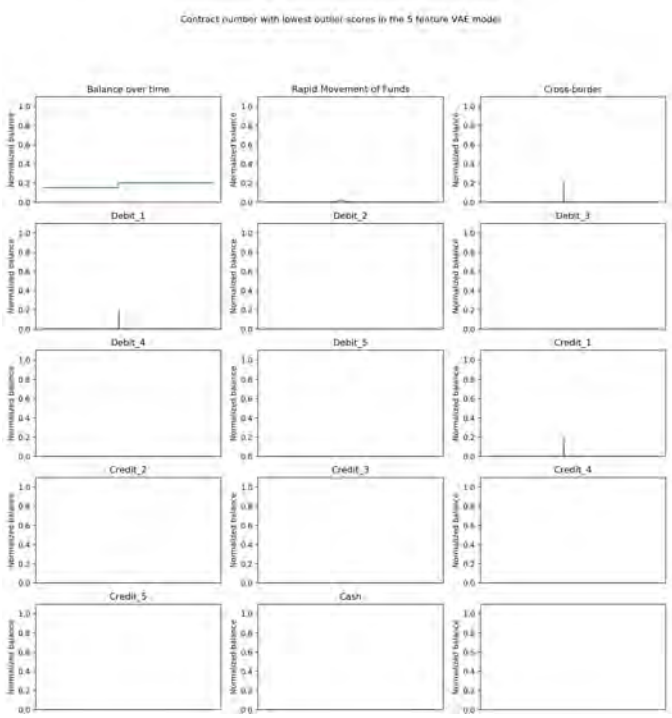

Figure 94: The windows for all features for the contract number with the lowest outlier score

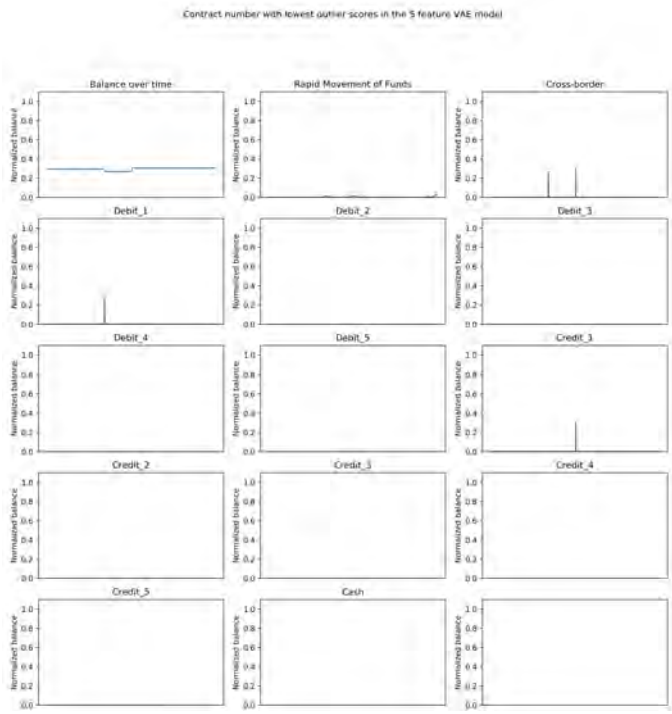

Figure 95: The windows for all features for the contract number with the second lowest outlier score

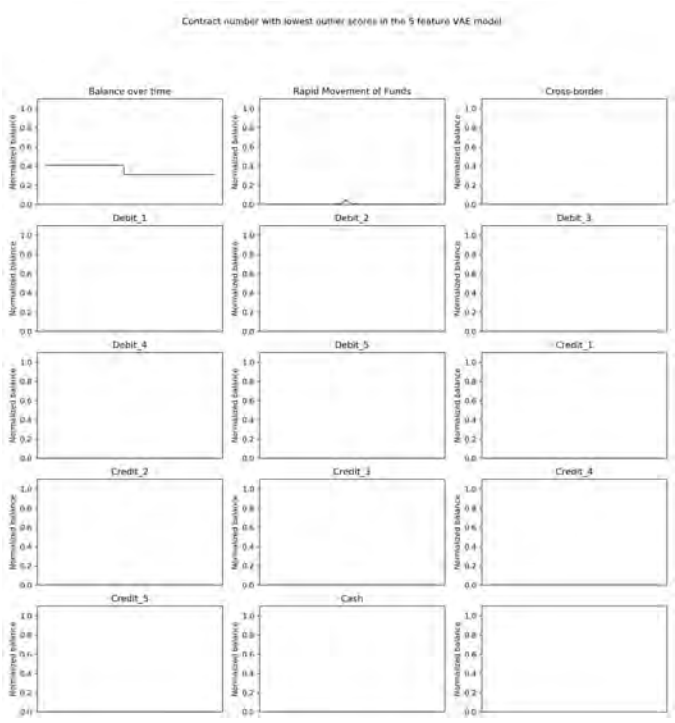

Figure 96: The windows for all features for the contract number with the third lowest outlier score

Results β-VAE model, with $\beta = 2$

For the three feature β-VAE model, with $\beta = 2$, the following median outlier scores were obtained, which are presented in Figure 97. Here it is visible that the balance changes very little, the RMoF behavior shows no high peaks and there are no cross-border transactions.

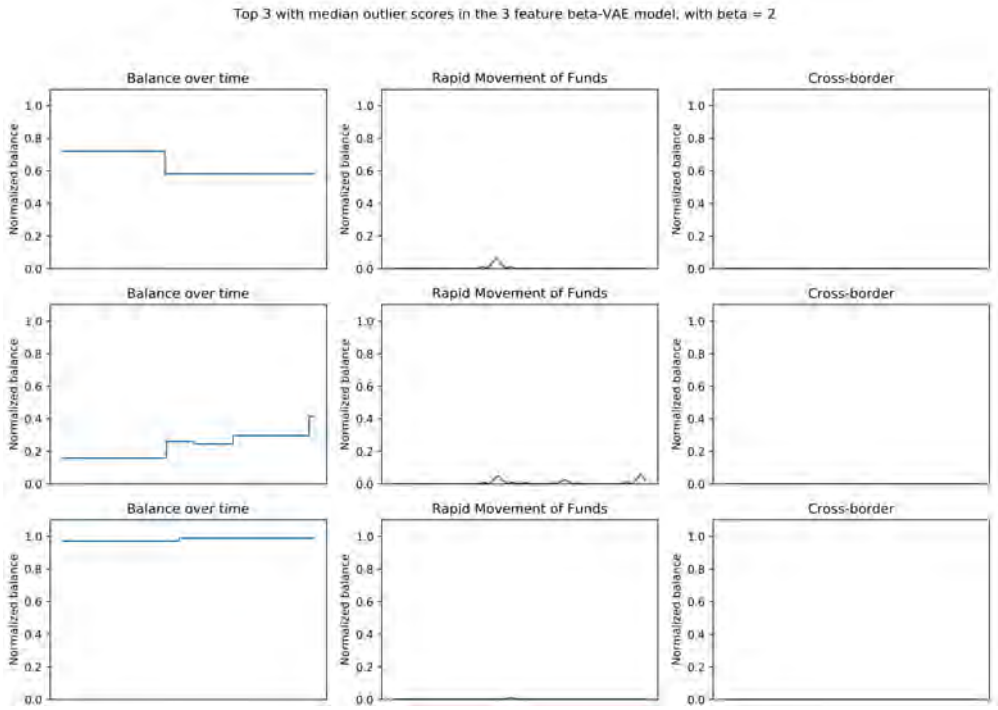

Figure 97: Three windows with a median outlier score, for three feature combination beta-VAE model, with beta = 2

Results β-VAE model, with $\beta = 5$

For the three feature β-VAE model, with $\beta = 5$, the following median outlier scores were obtained, which are presented in Figure 98. Here it is visible that the balance changes very little, the RMoF behavior shows no high peaks and there are almost no cross-border transactions.

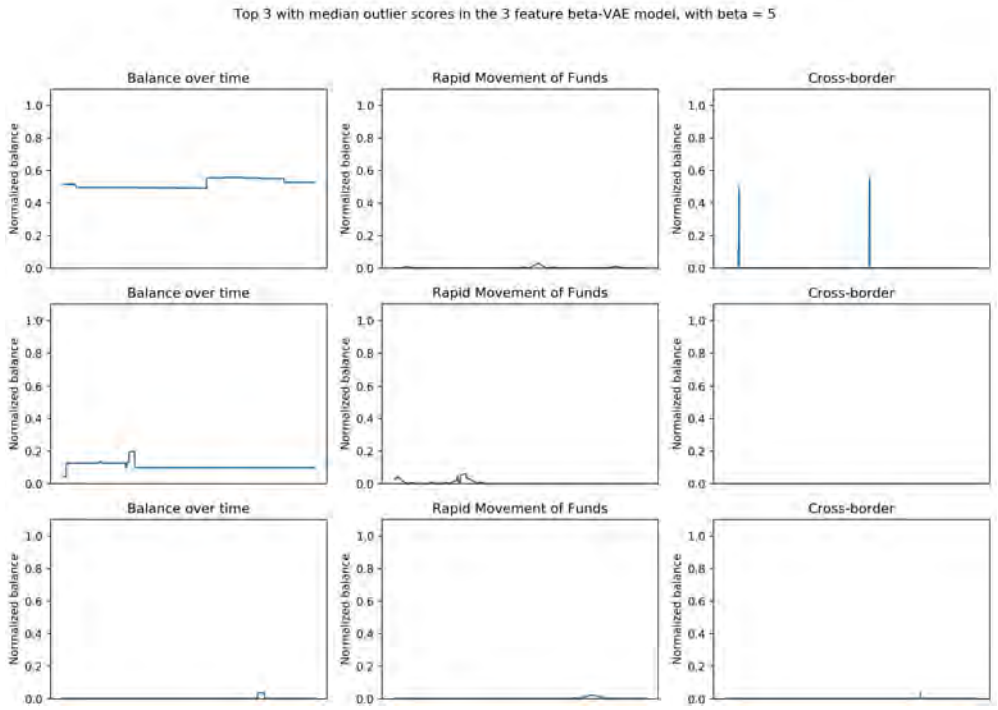

Figure 98: Three windows with a median outlier score, for three feature combination beta-VAE model, with beta = 5