IFC-Bank of Italy Workshop on "Machine learning in central banking"

19-22 October 2021, Rome / virtual event

Classifying payment patterns with artificial neural networks: an autoencoder approach[1]

Luis Gerardo Gage and Raúl Morales-Resendiz,
Centre for Latin American Monetary Studies (CEMLA)

John Arroyo and Jeniffer Rubio, Banco Central del Ecuador;

Paolo Barucca, University College London

[1] This presentation was prepared for the Workshop. The views expressed are those of the authors and do not necessarily reflect the views of the Bank of Italy, the BIS, the IFC or the central banks and other institutions represented at the event.

An extended version is published in the Latin American Journal of Central Banking 1 (2020) 100013.

Classifying payment patterns with artificial neural networks: an autoencoder approach[1]

Jeniffer Rubio[2], Paolo Barucca[3], Gerardo Gage[4], John Arroyo[5] and Raúl Morales-Resendiz[6]

Abstract

Payments and market infrastructures are the backbone of modern financial systems and play a key role in the economy. One of their main goals is to manage systemic risk, especially in the case of systemically important payment systems (SIPS) serving interbank funds transfers. We develop an autoencoder for the *Sistema de Pagos Interbancarios* (SPI) of Ecuador, which is the largest SIPS, to detect potential anomalies stemming from payment patterns. Our work is similar to Triepels-Daniels-Heijmans (2018) and Sabetti-Heijmans (2020). We train four different autoencoder models using intraday data structured in three time-intervals for the SPI settlement activity to reconstruct its related payments network. We introduce bank run simulations to feature a baseline scenario and identify relevant autoencoder parametrizations for anomaly detection.

The main contribution of our work is training an autoencoder to detect a wide range of anomalies in a payment system, ranging from the unusual behavior of individual banks to systemic changes in the overall structure of the payments network. We also found that these novel techniques are robust enough to support the monitoring of payments' and market infrastructures' functioning, but need to be accompanied by the expert judgement of payments overseers.

Keywords: Market Infrastructure, Neural Network, Anomaly Detection, Autoencoder, Artificial intelligence, Retail Payments, Machine Learning.

JEL classifications: C45, E42, E58.

[1] Authors are responsible for the statements found in this research article, and do not represent those of the institutions to which the authors are affiliated. Any errors are their own responsibility. A special acknowledgement to Dr. Serafín Martínez-Jaramillo, CEMLA, and Dr. Ronald Heijmans, De Nederlandsche Bank, for their valuable comments and suggestions

[2] Banco Central del Ecuador (BCE)
[3] University College London (UCL)
[4] Centro de Estudios Monetarios Latinoamericanos (CEMLA), México.
[5] Banco Central del Ecuador (BCE)
[6] Centro de Estudios Monetarios Latinoamericanos (CEMLA), México.

1. Introduction

Financial market infrastructures (FMIs) underpin the financial system and the economy by enabling multilateral transactions under certain rules and common platforms. They entail by design financial and operational risks related to interbank funds transfers. Given their systemic importance, central banks need to be able to monitor their activity and to identify anomalous events, and for these purposes artificial neural networks result purposeful.

According to the CPMI-IOSCO Principles for Financial Market Infrastructures (PFMI), payment systems and FMIs should be properly designed to support their participants to manage and mitigate risks more efficiently, and have better liquidity management. Their performance is key to support the financial system's health. Indeed, when payment processing rules or arrangements are not clear or comprehensive, payments and FMIs participants could take unnecessary. Likewise, if platforms are not resilient, the entire network could be endangered by cyber threats. In fact, poorly designed and operated payment systems and FMIs can contribute to exacerbate systemic crises. Contagion risk could impact the overall stability of the financial system. Therefore, the payment systems and FMIs must be robust and reliable, available even in times of stress (CPMI-IOSCO, 2012).

Monitoring payments and financial market infrastructures is one of the primary objectives for central banking to ensure that the above events take place. By overseeing the functioning of FMI and SIPS they can identify and address systemic risk events. Central banks have long worked in establishing an appropriate monitoring and risk management framework for SIPS, and other prominent payment systems and FMI. Oversight is also thought as a relevant task that fosters FMI good performance. This task is supported by several quantitative and qualitative tools, including international standards such as the PFMI, risk policies, Business Intelligence and other software tools for liquidity and collateral monitoring, business continuity plans, among others. Yet understanding the complexity of FMI and SIPS requires a powerful and well-designed toolkit. (CPMI-IOSCO, 2012)

In light of this challenging task, we present an application of artificial neural networks for outlier detection, tailored to payment systems and FMIs. In our work, we focus on the major FMI in Ecuador, the *Sistema de Pagos Interbancarios* (SPI), managing both wholesale and retail payment transactions. The SPI is a hybrid payments system performing Real-Time Gross Settlement and Deferred Net Settlement features.

For the purpose of our work, payment systems can be efficiently represented as directed networks, where institutions are nodes and payment flows from an institution to another one constitute edges. There are payment flows or patterns of payment flows, i.e. network patterns, that can pose a systemic risk in case a particular participant, or a set of participants, is unable to settle their transactions in a specific time interval. From an oversight point of view, it is crucial to be able to determine a normal pattern in a payment network as well as to identify risky events arising from anomalous patterns.

In our work, we introduce and test a pattern recognition tool for anomaly detection based on an autoencoder architecture. An autoencoder is an unsupervised feed-forward neural network that aims to reconstruct the input data at the output layer by passing through a lossy compression process that creates a lower-dimensional representation. This makes the model learn the most important features inherent to the data. Once the autoencoder is trained and has learned the usual patterns, the anomaly detection is done through flagging those instances that have high reconstruction errors, which perhaps is an indicative of abnormal patterns.

We trained four autoencoder models to identify common and anomalous payment patterns of financial entities (participants) in the SPI payments network. We present the results of the models that consist of anomalies between normal and unknown patterns of payment flows. These results can significantly contribute to oversight experts, to establish an alerting system and to anticipate potential risks in the SPI. The detailed set up, training and testing of our models is further explained in the methodology section where we also provide information on the different architectures, dataset partitions' and data preprocessing taken.

Our work is related to Triepels-Daniels-Heijmans (2018), Triepels-Heuver (2019), and Sabetti-Heijmans (2020). They also developed autoencoder approaches for both wholesale and retail payment systems, by training different models using daily payment flows, in some cases to identify financial stress in entities that have gone bankrupt.

The main contribution of our work relates to training autoencoders able to detect a wide range of anomalies in the SPI, ranging from spotting the anomalous behavior of individual banks to detecting changes in the overall activity of the payments network. Our work highlights that these novel techniques are robust enough to support payments' and market infrastructures' oversight, and ultimately to monitor financial stability, but need to be always in tandem with the expert judgement of central banking overseers.

The remainder of the work is structured as follows. Section 2 surveys relevant literature that is closely related to our work. Section 3 describes the *Sistema de Pagos Interbancarios* and also provides a statistical analysis of the data. Section 4 introduces the methodology and autoencoder setup. Section 5 presents key results on the autoencoder performance and the bank run simulations. In Section 6 we discuss how our work can be further advanced.

2. A brief survey of literature

Detecting outliers in a dataset is an old challenge in statistics (Edgeworth, 1887). Although the definition of anomaly can vary across different disciplines, the underlying statistical definition of anomaly is the same, i.e. a subset of data that behaves according to different patterns with respect to those identified as the normal ones. This general definition suits perfectly the logic of many known machine learning algorithms. Not surprisingly, these algorithms have been developed and applied within many domains, e.g. intrusion detection, fraud detection, fault detection, as well as medical anomaly detection (Chandola, 2008).

Hodge and Austin (2004) described three fundamental approaches for outlier detection. *Type 1* where the outliers are determined without any prior knowledge, this approach is analogous to unsupervised learning, where the learning algorithm is provided with an unlabeled dataset and it aims to find hidden patterns within the data. *Type II* requires pre-labeled data targeted as normal or abnormal, this approach is analogous to supervised learning, where the learning algorithm objective is to fit a function that reproduces the behavior of pre-labeled data in order to make predictions on unseen data. And, *Type III* that requires also pre-labeled but it only models normality - and in few cases abnormality - helping to define a boundary for normality, this approach is analogous to semi-supervised learning, which is in the middle ground between supervised and unsupervised learning and can use both labeled and unlabeled data during the learning process. Hodge and Austin described different techniques from three fields: statistics, neural networks and machine learning. One can also find hybrid systems that combine techniques from these fields. As our work analyzes a dataset that *a priori* do not contain labels on what is and what is not an anomaly, our methodology can be classified as *Type I* at first glance, i.e. analogous to unsupervised learning. But given that after the fitting process we create bank run simulations, which can be considered in a way as anomalies, to test the capacities of the autoencoder to identify them, it is rather appropriate to state that our methodology falls within *Type III*, i.e. analogous to semi-supervised learning.

A challenge for outlier detection is associated with the presence of high dimensionality in the data. There are different approaches (Barnett and Lewis, 1994; Arning et al., 1995) to dimensionality reduction, including clustering methods (Knorr and Ng., 1998). One way to tackle high dimensionality is to make parsimonious projections in lower spaces and then proceed with the anomaly detection as proposed in Aggarwal (2001). One of the main properties of the data in this paper is that the number of features is close to the number of observations, thus high dimensionality needs to be taken under consideration; remarkably, an advantage of the autoencoder is that - by design - the encoding creates a lower representation of the data that learns the most relevant features.

Autoencoders have been previously used for outlier detection. Hawkins et al., (2002) developed a methodology for an autoencoder with two different datasets, one dataset for network intrusion detection and the other for breast cancer identification. In the first case all outliers were identified and, in the second case over 75% out of the total, showing the robustness and transferability of the methodology.

Another case in which an autoencoder is implemented to detect anomalies is found in Williams et al. (2002). Their results were compared with three techniques: i) the Donoho-Stahel estimator, ii) an outlyingness estimator proposed in Hadi (1994) based on both the means and covariances of the variables and the Mahalanobis distance, iii) and a model of mixture-models clustering. The comparison was done fitting and testing the techniques on many datasets, where each dataset contained labels that identified the abnormal instances; the datasets relates to information from different areas such as breast cancer, internet intrusions and other topics. The results show that for small datasets, the compared techniques show a good level of performance, with clustering being the one that presented the most difficulties in detection, but in the case of longer datasets, the autoencoder showed better performance identifying anomalies.

In the context of finance, applications of anomaly detection are found in Aleskerov et al. (1997), Ghosh and Reilly, (1994), Dorronsoro et.al. (1997), and Baruse et al. (1999), and are mainly concerned with credit card fraud detection. Nevertheless, the application of unsupervised machine learning methodologies for payment systems' oversight is relatively new among central banks and relevant authorities. In a recent series of papers (Triepels-Daniels-Heijmans, 2018; Sabetti-Hejmans, 2020), the autoencoder architecture has been shown to be effective for learning patterns of normal transaction data and to detect anomalous payments, using the autoencoder reconstruction error. In Triepels-Daniels-Heijmans (2018) two autoencoders with one hidden layer were trained - one used a linear activation and the other one used a sigmoid activation in the hidden layer. They used data from TARGET2 (the RTGS for the Eurosystem) settlement system to reconstruct liquidity-related information and also introduced a bank run simulation. The paper reported that the data presented relevant features of the payments network enabling the autoencoders to detect changes in the payments flow behavior. Sabetti-Heijmans (2020) compared the performance between one hidden layer and two hidden layers autoencoders, using data from the Canadian ACSS (a Canadian retail payment system). They found that the one hidden layer autoencoder had lower validation error compared with the two hidden layers autoencoder, but the two hidden layers autoencoder displayed a lower variance that can lead to better results when using testing data.

Our work follows the general approach of Triepels-Daniels-Heijmans (2018) and Sabetti-Heijmans (2020), and it contributes to the literature by analysing a new dataset for the Ecuador SPI and presenting a detailed review of alerts, illustrating the ample range of anomalies that can be detected by the autoencoder.

3. The *Sistema de Pagos Interbancarios*

3.1 SPI main features

The Central Bank of Ecuador (BCE) must provide the physical and electronic means of payment necessary for the proper functioning of the country's economy. In this respect, the BCE is the owner and operator of several payment systems, which as a whole are known as the Central Payment System (SCP). The SCP entails the interbank funds transferring system for large value payments and it also supports settlement of private retail payment systems and securities clearing and settlement systems. Thus, the SCP represents the most relevant payment infrastructure for Ecuador.

The underlying system that makes up the SCP is the *Sistema de Pagos Interbancarios* (SPI). The SPI settles 60% of the total payments in the SCP, for which reason, this system is deemed as the major SIPS in Ecuador. In light of its importance, our work focuses on the SPI activity. The relevance of the SPI is paramount. It provides an infrastructure for different types of participants. Within this universe, there are banks with a higher activity and make payments with the rest of the SPI participants. Some SPI participants only make transactions with few entities in the payments network. It also channels all Government payments as well as 98% of wholesale and retail payments from the private sector. On average, the SPI processes 300,000 transactions per day totaling USD 450 million.

In terms of the clearing and settlement mechanism, the SPI makes the settlement of payments in three daily time-intervals. Each time interval represents an intraday settlement period for the interbank payments ordered by the financial entities in the SPI. These time-intervals are carried out at three different hours (08:30, 11:00 and, 16:30). Each time-interval is exclusive of the other, the net amounts between financial entities are cleared and settled at the end of each time interval. The SPI only settles the operations of financial institutions that have liquidity in their accounts at the BCE to cover their net debit position at the time of the settlement of each of these time intervals, otherwise the financial institution is excluded from the process. The latter in order to avoid liquidity risks for the entire SPI participants.

The SPI as any other payments and market infrastructure is subject to operational and financial risks. Technological advances such as malicious intruders or operational events experienced by a single participant, can both represent a major risk for the SPI and its participants, with undesirable negative effects in the financial system and, ultimately, the economy.

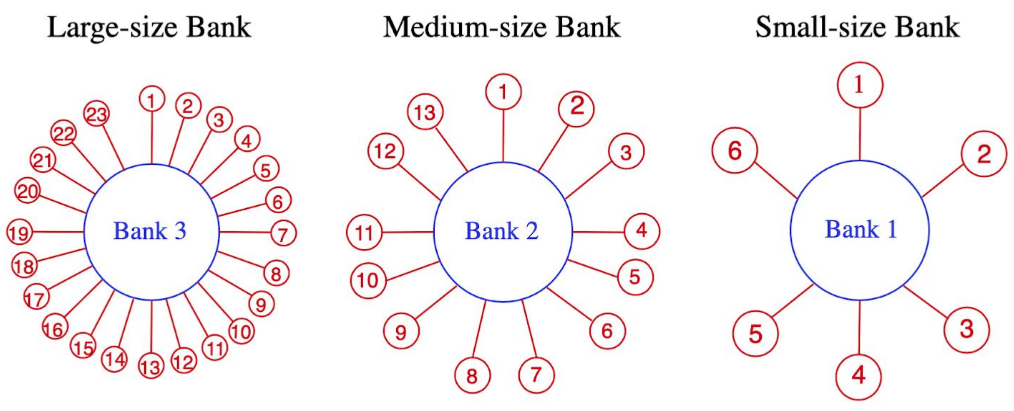
Developing an automated oversight tool to detect atypical payments or payment behavior is a significant contribution to better identify malicious activity in SPI and other prominent payment infrastructures. Such an alert system should be also able to allow the monitoring authorities and the own operator to understand normal behavior of financial institutions as they participate in the SPI. For such purposes, we work on the autoencoder feedforward neural network to anticipate and identify potential risks in this systemically important market infrastructure of Ecuador.

3.2 The dataset

Since the SPI implementation in 2002, the total amount and number of transactions settled in this system have grown year after year. In 2018, the SPI settled over USD 100 billion with a corresponding number of transactions of almost 70,000, representing a daily average of USD 400 million and 300 thousand transactions. For the purpose of our work, we used transaction information from 24 financial institutions in 2018, which represent around 90% of the amount channeled by SPI by the private sector.

As seen in Figure 1, the typical flows for a large, medium, and small bank can significantly vary. In our work, we consider a subnetwork of payments, i.e. the maximum number of flows that a financial institution can have is 24, reflecting the fact that it can send payments to the rest of the 23 banks and to itself.

Figure 1. Payment connections in the SPI for large-, medium- and small- banks



For the purposes of this investigation, 741 time intervals corresponding to 247 working days of the year 2018 were used. In our analysis, each payment flow (i.e. connection) represents the exchange of interbank payments between bank A and bank B. If we analyze the frequency of participation of a bank in the SPI, on average, the analyzed payments flows, i.e. interbank payments by SPI participants, take place in nearly half of the 741 time intervals, that is 376 time intervals. A fraction of 25% of the analyzed flows appeared more than 731 times, in effect these are particularly recurrent transactions for 2018. Conversely, 25% of payment flows only took place in 56 times intervals. A small set of unique payment flows occurred only once or up to 15 times.

The average volume of payments per time interval amounts to USD 76 million. The maximum amount for 2018 time-interval was USD 212 million and the minimum, USD 31 million. Nearly 75% of the intervals registered payments for over USD 60 million, each.

Considering that there are interbank payment flows (i.e. payments between Bank A and Bank B) among all 24 banks, there can be a total of 576 (24 banks x 24 banks) possible connections. However, the SPI dataset shows that for 101 connections there was no single exchange. Therefore, we only analyzed a total of 475 connections for 2018. We can observe that there is an average of 241 connections for each time interval along the year. The lowest number of connections for a time interval was 47 payment flows, while the maximum, 298 flows. It is worth mentioning that in 95% of the intervals there were more than 207 flows. The average payment per flow was USD 2 million over 2018, while the maximum value for a payment connection reached USD 4 billion.

Figure 2. Payment flows in the SPI

A. Average value **B. Largest flows (USD millions)**

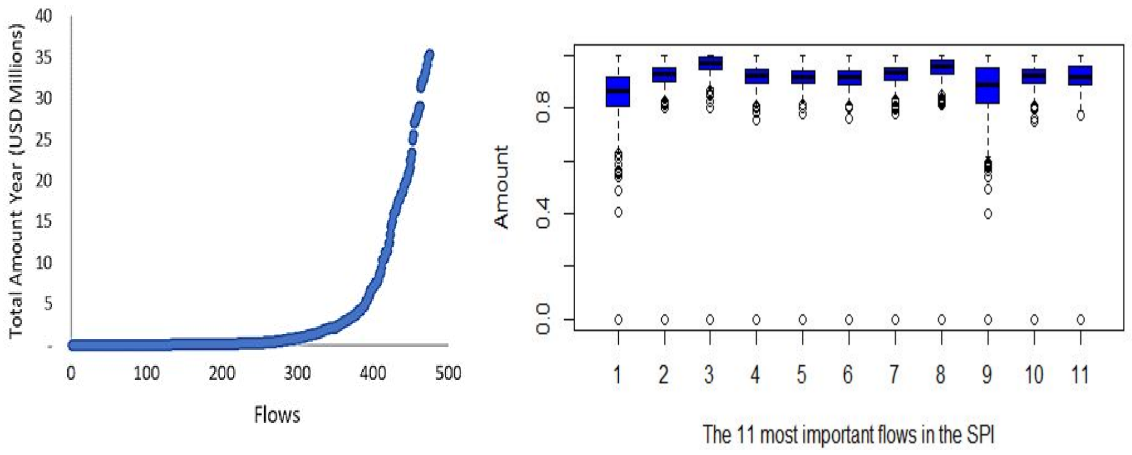


Figure 2A represents the average amount of all the flows of the SPI in 2018. The 75% of the connections amounted to over USD 130 billion, 25% of the flows amounted to more than USD 20 million. Figure 2B is a boxplot of the 11 most important flows in the SPI for payments made in the year. These flows represent around 50% of the total amount of payments made by the SPI in 2018.

The SPI dataset includes large and small, as well as more and less frequent, payment flows for 2018. However, for anomaly detection, each connection can be important regardless of the magnitude or frequency of interbank payments. There are payment flows that can pose a systemic risk in case a particular participant is unable to settle them in a specific time interval. From a payments system oversight point of view, it is crucial to be able to determine a normal behaviour and to identify risky events arising from anomalous behaviours. With this goal in mind, in the following sections, we introduced and tested a pattern recognition tool for anomaly detection based on an autoencoder architecture.

4. Methodology

In this section we begin by stating the general anomaly detection framework in the context of a payment system. First we define the basic structures that will be used, such as the set of participants, the time intervals, and the matrix that represents the interactions between participants. Once the above has been defined, the next step consists of the setup of the anomaly detection task, which will be based on the measurement of the reconstructions' quality made by a compression model.

The section continues with a detailed description of the autoencoder, as the compression selected model, providing details of its operation and why it can be used to detect abnormal patterns in the data. The section concludes with a discussion on the preprocessing of the data that is made prior to the training of the models, to then continue with a review of the adjustment and testing process of the models to finally introduce the bank run simulations.

4.1. Definition of the general framework for the anomaly detection task

Following Triepels (2018), let $B = \{b_1, b_2, ..., b_n\}$ be the set of SPI participants that settles transactions between them. Now let us consider $T = \{t_1, t_2, ..., t_m\}$ an ordered set of m time intervals where each $t_i = [\tau_{i-1}, \tau_i)$ having that i ranges from 1 to m, and where τ_i are specific timestamps delimiting time intervals. In our case each time interval represents one of the three intervals taking place in a SPI working day.

Then, we define the structure for liquidity transmission among institutions within different time intervals. Let $a_{ij}^{(k)}$ be total amount of liquidity transferred from institution b_i to institution b_j within the time interval t_k . The liquidity matrix $A^{(k)}$ accounts for the liquidity transferred between all the institutions within the interval t_k :

$$A^{(k)} = \begin{bmatrix} a_{11}^{(k)} & \cdots & a_{1n}^{(k)} \\ \vdots & \ddots & \vdots \\ a_{n1}^{(k)} & \cdots & a_{nn}^{(k)} \end{bmatrix}$$

The diagonal elements $a_{ii}^{(k)}$ indicate the total amount of liquidity that b_i transfers between its own accounts. The elements of $A^{(k)}$ can be interpreted as the weights of a network where nodes are institutions and edges are liquidity flows (payments). In order to feed the model with a simple data structure, every $A^{(k)}$ is mapped to a liquidity vector $\overline{a}^{(k)}$ with the form:

$$\overline{a}^{(k)} = [a_{11}^{(k)}, ..., a_{n1}^{(k)}, ..., a_{1n}^{(k)}, ..., a_{nn}^{(k)}]^T ,$$

where $\overline{a}^{(k)}$ is a n^2 column vector that consists of $A^{(k)}$ appended columns.

The data we study contains information on the liquidity vectors for a series of periods at a payments system. We aim to reconstruct the vectors by compressing and decompressing the dataset under use. For this purpose, we implemented a lossy compression, which generates a particular type of representation that allows the data to not be exactly learnt and some of the information to be lost. When this type of compression is implemented, the relevant patterns present in the data are learned. Once the compression model learns the common patterns, that is, the most observed, and a new liquidity vector is fed for its reconstruction; if the reconstruction is bad, this is explained by the fact that vector information differs from the normal patterns that the model learned, indicating the possibility of a potential anomaly. The quality of the reconstructions will be measured through the reconstruction error, in other words, the differences between values yielded by the lossy compression and the real vector values.

More formally, given a lossy compression model, let RE be the non-negative function that measures the reconstruction error of liquidity vector $\overline{a}^{(k)}$; $RE : D \rightarrow [0, \infty)$ where D is the set of liquidity vectors for all time intervals. Our main objective is to find all the liquidity flows (i.e. payment connections) in a particular time interval corresponding to reconstruction errors greater than a given threshold $\varepsilon > 0$, i.e., given a set of liquidity vectors D we aim to find the set $F = \{\overline{a}^{(k)} \in D \mid RE(\overline{a}^{(k)}) \geq \varepsilon\}$. It is noteworthy that there is no rule or methodology to follow in order to set the value for ε, instead it has to be set according to the particular characteristics of the data and prior knowledge on the respective payments system.

4.2. Autoencoder modeling

For our work, we select the autoencoder as the lossy compression model. The autoencoder falls in the category of artificial neural networks techniques. The basic unit of a neural network is the neuron or node, which can be both fed directly with the data or fed through other connected neurons, and depending on the type of neuron it will be defined how the information will be processed to generate an output. As can be seen in Figure 3, the neurons of one layer connect with those of the next but never between them. All neural networks have an input layer, an output layer, and at least one hidden layer (the case of having more than two hidden layers it's considered as deep learning). The types of neurons are:

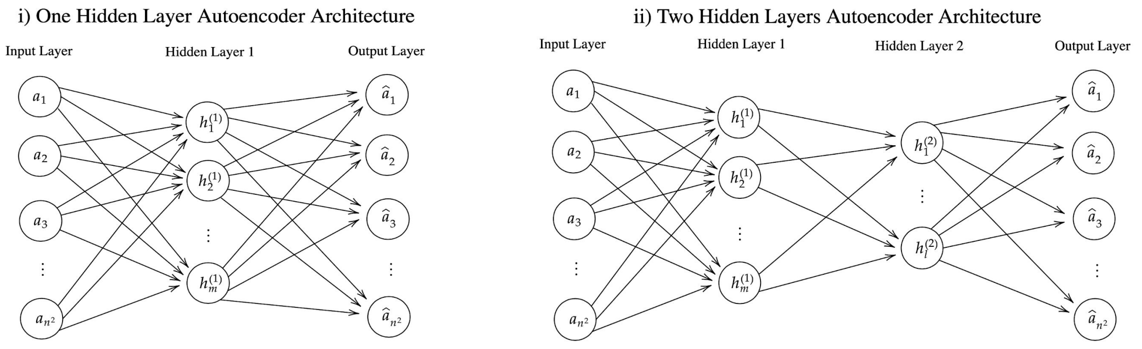
- *Input neuron*: They are fed with the data directly and this conforms the output that feeds the next layer.
- *Hidden neurons*: Each of them is fed by all the neurons of the previous layer ($x_i's$) and multiplied by a set of weights ($w_i's$). The output of these neurons is generated by first computing a weighted sum of the weights and the outputs of the previous layer and then applying a function f to it called activation function, i.e., the output is given by $f(\sum_i w_i x_i + b)$. The term b refers to the bias that data poses.
- *Output neurons*: It follows the same process as hidden neurons, but the output it generates rather than feeding other neurons is the resultant final prediction.

The weights $w_i's$ decide how much of the information in each neuron should be transmitted to the next layer; these are the parameters that will be learnt during the training process. The *activation function* f, has the purpose to learn non-linear relationships between the components of the data.

We utilized two different activation functions, Rectified Linear Unit (ReLU), namely $ReLU(x) = max(0, x)$, and the hyperbolic tangent (Tanh). The latter maps the value to the interval (-1, 1) and falls within the category of sigmoidal functions (s-shape) which provides a simple model for the firing of a real neuron. An issue regarding sigmoidal functions is that derivatives can become very small far from zero, affecting the learning process which is based on gradient methods; ReLU overcomes this issue - being linear for positive values - and also its computation is simpler, but if during the learning process the weighted sums gets below zero, then most of the neurons in the neural network will go to zero, potentially leading to non sensitivity and poor fitting[7][8].

The autoencoder is made up of two components, the encoder and the decoder. The encoder is the initial part of the autoencoder and it has the task to create an accurate lower-dimensional representation of the data. The second part of the autoencoder, the decoder, is in charge to carry out the reconstruction of the data. The encoder goes from the input layer to the layer with the lowest number of neurons, which is commonly called bottleneck given that is the layer of the network where data is the most compressed. The encoder can be represented as a function $h = f(X)$, where X represents the input data[9]. On the other hand, the decoder goes from the *bottleneck* to the output layer, it can be represented by the function $r = g(h)$. The autoencoder final objective is to find f and g such that $X \approx g(f(X))$.

Figure 3. Autoencoders Architectures



In Figure 3 we can observe the architecture for two autoencoders, one hidden layer (left panel) and two hidden layers (right panel), where $m < n^2$ and $l < m$; this tells us that the input data

[7] An alternative to overcome the issues that arise from the use of ReLU as activation function is to use the Leaky ReLU that has the same value for the non-negative values but for a negative variable (x) it assigns the correspondent value 0.01x. The use of this ReLU variation is left for future work.

[8] A further description on feed-forward neural network, components and learning process can be found in (Goodfellow et. al. 2016).

[9] Input data is represented by the liquidity flows.

will be compressed through a projection from a n^2-dimensional space to a m-dimensional space, for the one hidden layer, having an extra compression from m dimensions to l dimensions in the case of having two hidden layers. Adding one more layer to the autoencoder compresses the bottleneck, forcing the neural network to learn a lower dimensional representation. Yet adding many layers increases the number of parameters and can cause the neural network to overfit the training data, thus failing to generalize. In our study we trained both one and two hidden layers autoencoders.

The learning of the autoencoder, and in general for neural networks, is achieved by the minimization of a cost or loss function with respect to the weights and biases (mentioned above). For our work, this function will correspond to the reconstruction error (RE) of our lossy compression. More precisely, the reconstruction error will be given by the mean of the squared differences between the liquidity vectors and its reconstruction, in other words the the Mean Squared Error (MSE), for all time intervals, i.e. the loss function will depend on the set D:

$$RE_D = \frac{1}{m} \sum_{k=1}^{m} (\overline{a}^{(k)} - g(h(\overline{a}^{(k)})))^2$$

The autoencoder's weights are learnt through mini-batch backpropagation[10]. We strived to fine-tune the autoencoder's hyper-parameters to improve its performance. This is accomplished through cross-validation, which performs an exhaustive search within a predefined set of hyper-parameters for multiple data partitions, where the performance of each hyper-parameter setting is evaluated. We also carry out pre-processing of the data, which can lead to a significant improvement in the performance. This is discussed in more detail in the next subsection.

4.3. Model fitting, selection and testing

In this subsection we describe the procedure to fit the autoencoder, which involves the preprocessing and partition of the data, and the training and validation steps.

Before fitting the model, we pre-processed the data, this corresponds to a log-transformation that was followed by a min-max standardization. The former have the purpose to reduce the skewness in the payments flows, while the latter maps the values to the interval $[0, 1]$, to give the same degree of importance to all the bilateral transactions and avoid the autoencoder to be unbalanced toward the transactions with highest value. In such a pre-processing, let V be a feature (in our case we have 576 features, each one corresponding to the flow of liquidity between one institution to another) the log-transformation of V is done by applying the natural logarithm to it, this is computed for all the features.

After this step, we continued with the min-max standardization, for this end, we first found the maximum and minimum values for feature V, to then transform each feature element by:

[10] Backpropagation is a learning mechanism, based on gradient descent, which is widely used for the training of neural networks. The mini-batch indicates that the updating of the parameters learned is done after passing not the complete dataset or a single instance (that currently are another types of backpropagation), but a portion of the whole dataset, to the network; a further insight can be found in (Goodfellow et. al. 2016).

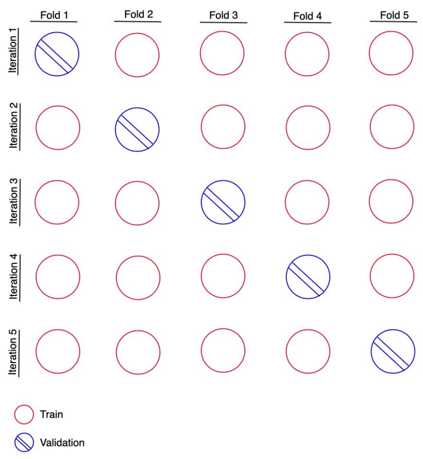
$$minmax(x_i) = \frac{x_i - min(V)}{max(V) - min(V)}$$

Where x_i is an element of V. The aforementioned steps led to smaller training and validation reconstruction errors than when the data was fed in its original form.

Once the data is preprocessed, we make a partition of it. It's a common practice in machine learning to separate the whole data into different subsets. In our work, we first randomly divide the data in two parts, the first, which will be used to perform cross-validation, is the larger and contains 80% of the original data. The remainder 20% of the data, commonly called the test set, is used in order to evaluate the performance of the model with data not observed during training or validation.

The cross-validation is performed once the data is divided in subsets, to assess the performance of the different hyper-parameters, with the ultimate goal of setting the best model configuration. This is mainly guided by using as selection criterion the validation MSEs. More specifically, cross-validation as a re-sampling technique is useful to evaluate the effectiveness of machine learning models. For our work, we implemented K-fold cross-validation with 5 folds. As mentioned above, the implementation is performed on the 80% of the whole dataset. In sum, the cross-validation process can be divided into the following: first we randomly divide the data into 5 disjoint groups, or folds; then, a hyper-parameters configuration is chosen (e.g. an autoencoder with one hidden layer with 100 neurons); in the next step which corresponds to the first iteration, the model is trained in four groups and the validation is done in the remaining group; and, the iteration ends with the computation of training and validation MSEs. The following iterations redo the same process, but as can be seen in Figure 4, the difference lies in the training and validation sets that are used.

Figure 4. 5-fold Cross-validation



Once all the iterations are completed, the mean of both training and validation MSEs is computed and this will be the performance of the model with the previously hyper-parameters chosen. The above process is carried out for each element of the set of hyper-parameters to be tested.

The determination of the best configuration of a model is made based on the validation MSEs. Generally, the model that yields the lowest MSE is selected, but it may be the case that the best model is too complex, ie, that the number of neurons in the hidden layers is very large, and one will prefer to choose a model with less complexity where the difference between their MSEs is not significant.

After performing cross-validation and the best hyper-parameters configuration is selected, the model is re-trained in the 80% of the data. In the last step the model is fed with unobserved data, which corresponds to the test set, and the instances that show highest reconstruction errors are identified.

4.4. Bank run simulations

Given that the SPI dataset presents a small amount of anomalies that could pose a level of uncertainty about its abnormality, we perform a series of bank run simulations similar to Triepels-Daniels-Heijmans (2018) and test whether the autoencoder was able or not to flag them as anomalies. The simulations were done by randomly choosing an institution b_i, then all its outgoing flows for a given period are modified according to:

$$a_{ij}^{(k)} \to a_{ij}^{(k)} + (B(k) \cdot E(k))$$

Where k is the time period, $B(k) \in \{0, 1\}$ was sampled from a $Bernoulli(p)$ random variable and decides whether extra liquidity will be added or not to the current period, $E(k) \in [0, \infty)$ was sampled from a $Exponential(\lambda)$, which decides how much extra liquidity will be added to the payment flow. The parameters p and λ determines the intensity of the bank run, the greater, the more intense the bank run will be. The autoencoder is expected to be less able to reconstruct the payment networks arising from these simulations, indicating that they are displaying anomalous behavior, in this case a bank run.

5. Results

In this section we first present the models we trained, highlighting the changes for each model related to the network architecture and the activation function. This is followed by a summary of the performance and results of every model. Next, we show the analysis of the anomalies that were detected in all or the majority of the models and that were found relevant for oversight purposes.

5.1. Results from autoencoder's fitting and performance

We analyze different setups for the autoencoder by varying the number of hidden layers, the number of neurons in each layer and the activation functions, this led us to the definition of four different models:

- ➔ Model 1: One hidden layer with TanH as the activation function
- ➔ Model 2: Two hidden layers with TanH as the activation function
- ➔ Model 3 One hidden layer with ReLU as the activation function

➔ Model 4: Two hidden layers with ReLU as the activation function

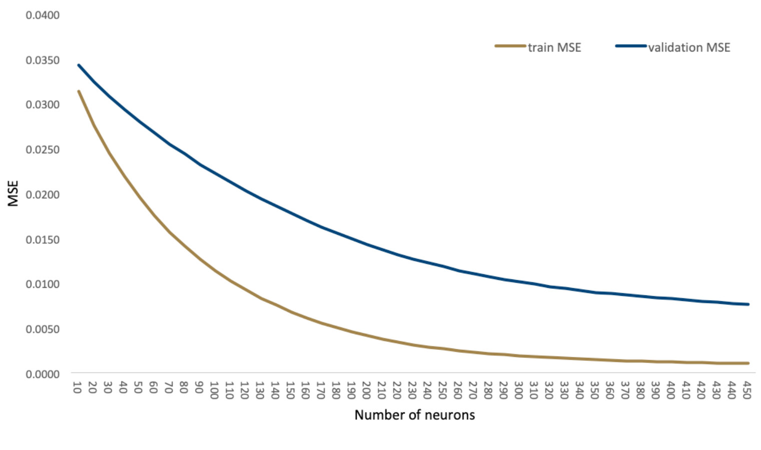
As described in subsection 4.3, each of the models were evaluated using 5-fold cross-validation to determine the optimal number of neurons for the hidden layers, the values proven for each model are summarised in Table 1. It is noteworthy that the autoencoder performs a compression, this can be observed in the reduction of number of neurons between the input layer and the hidden layers. In the case of one hidden layer we have gone from a 576-dimensional space[11] to a new space whose dimensions can range from 10 to 450. In the case of two hidden layers we have a double reduction, first a reduction similar to the case of one hidden layer takes place (now dimensions ranges from 10 to 400), then the autoencoder is forced to generate a smaller space with 8,16 or 32 dimensions. This compression enables the model to learn only the most important characteristics inherent in the data.

Table 1. Summary of the configurations evaluated for each model

	Activation Function	Neurons in input layer	Neurons in first hidden layer	Neurons in second hidden layer	Neurons in output layer
Model 1	TanH	576	(10, 20, 30,..., 450)	----------------	576
Model 2	TanH	576	(10, 20, 30,..., 400)	(8, 16, 32)	576
Model 3	ReLU	576	(10, 20, 30,..., 450)	----------------	576
Model 4	ReLU	576	(10, 20, 30,..., 400)	(8, 16, 32)	576

Figure 5 shows the Model 1 (with one hidden layer) validation and training errors. We set the final configuration at 300 neurons, which is the point where the error has no substantial reduction.

Figure 5. One Hidden Layer with TanH performance for different number of neurons

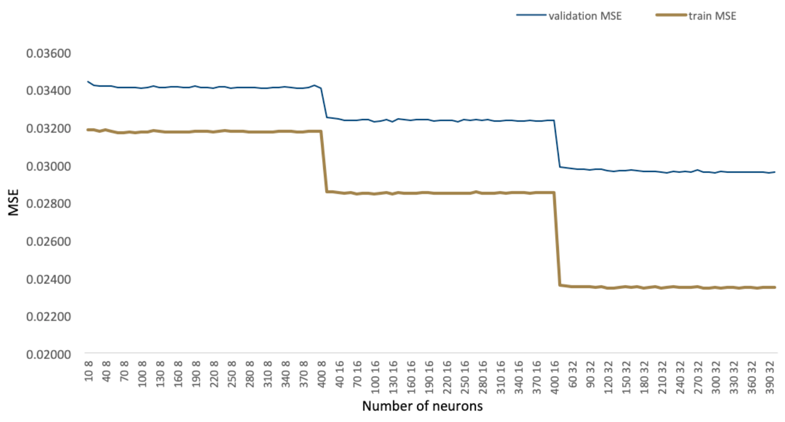


For the case of Model 2, Figure 6 shows a stepwise decrement of the errors that relates to the number of neurons used in the second hidden layer, where it can be said that having 32 neurons in the second hidden layer is the best decision. For the first hidden layer, here it is observed a

[11] The 576 dimensional space corresponds to all our features which are the 24x24 participants' settlement interactions.

continuous decreasing behavior, where the inflection point corresponds to 220 neurons. In light of this training results, the final configuration of Model 2 is set to have 220 neurons in the first layer and 32 neurons in the second layer.

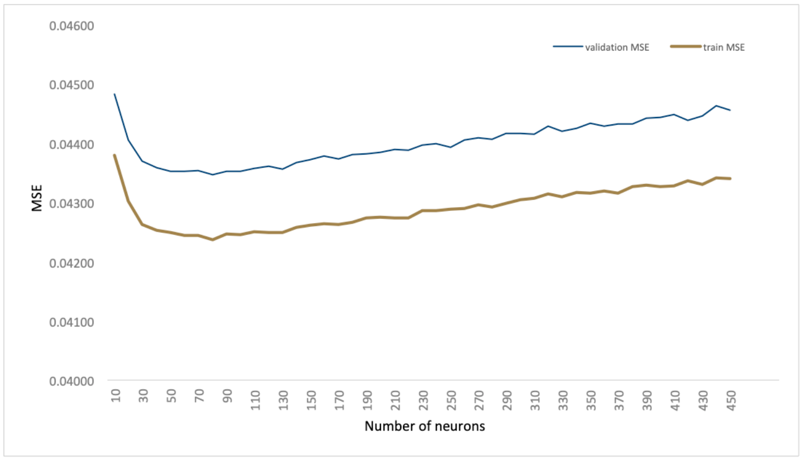
Figure 6. Two Hidden Layers with TanH performance for different number of neurons



In Figure 7 is shown the performance of Model 3, which corresponds to one hidden layer with ReLU activations. It can be observed that from 10 to around 80 neurons, the training and validation errors present a decreasing trend, but for the rest of the neurons the errors only increase, for that reason we decided to set the final configuration of the model using 80 neurons in the hidden layer.

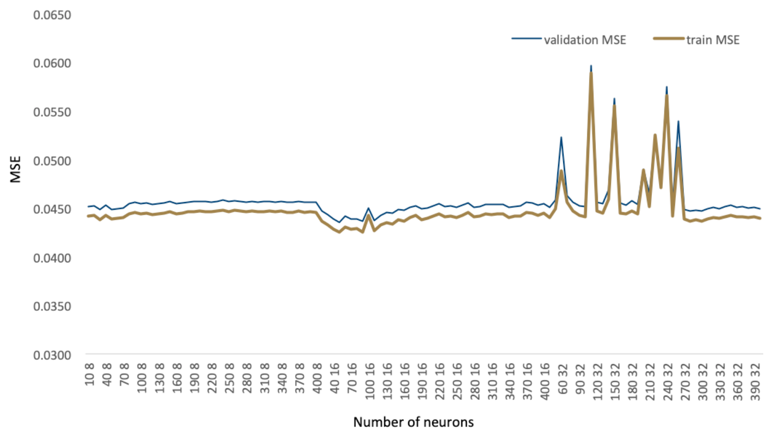
For Model 4, Figure 8 shows its performance. It can be noted that the behavior of the errors is stable for the configurations where the second hidden layer has 8 neurons, presenting only small variations. Then, for the configurations with 16 neurons in the second layer, a decrease in errors is observed, reaching the minimum with 110 neurons in the first layer; after this point a slight increase in errors is observed. In the case of 32 neurons in the second layer, the performance begins to be unstable, it is possible to observe peaks where the errors have large increases with respect to the rest of the configurations and where even the training and validation errors are the same[12].

Figure 7. One Hidden Layer with ReLU performance for different number of neurons



[12] This is not desirable given that the training data is the one used to adjust the model, thus it should show smaller errors than the validation data which is not used for the training.

Figure 8. Two Hidden Layers with ReLU performance for different number of neurons



Once the optimal configuration for each model was selected we proceed to re-train the models on the whole set used for cross-validation. It follows the feeding of the models with unseen data. The behavior of the reconstruction errors for the liquidity vectors belonging to the test set is expected to be stable and low and that only a few of them are above the average. This is given by our assumption that most of the data has a normal behavior and that only a few instances will perform abnormally; if the above does not happen, it means that our model did not perform correctly the reconstruction. This will imply that more data need to be used for training or that the chosen configuration is not adequate.

Figure 9 shows the results of the test set reconstruction for Model 1 and Model 2. It can be observed that both models identified six time intervals where the reconstruction error is higher than the average, but the rest of observations present a stable reconstruction with the difference that Model 1 shows higher variance and that Model 2 has bigger overall reconstruction errors.

Figure 9. Errors corresponding to the reconstruction of test set for Model 1(left) and Model 2 (right)

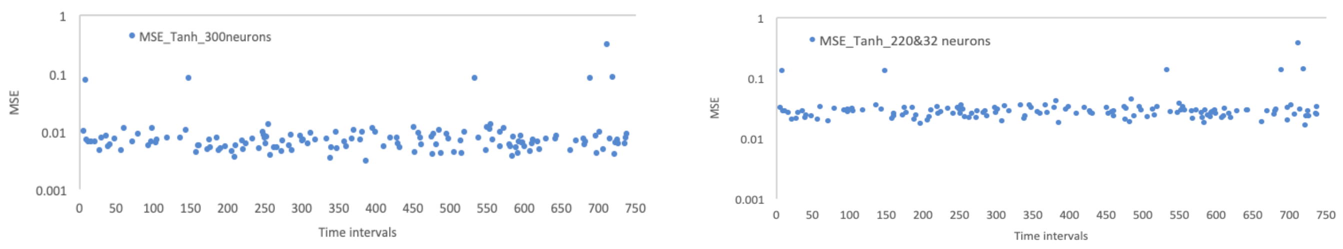
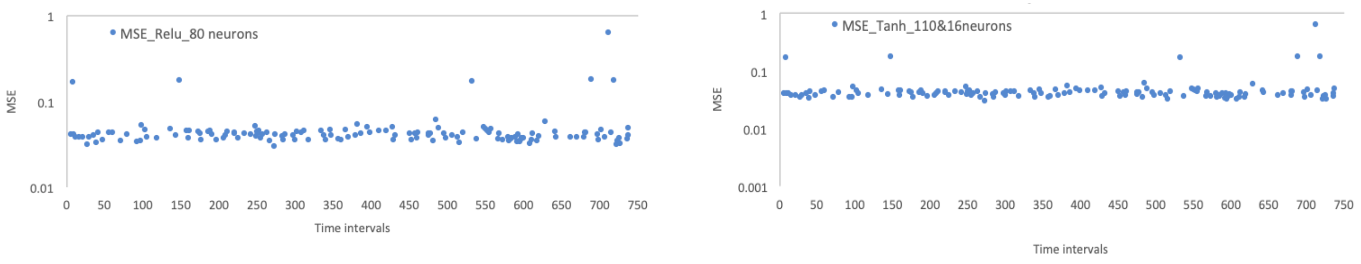


Figure 10 presents the reconstruction errors for the test set related to Model 3 and Model 4, it can be observed that both models have a similar behavior and that six time intervals present larger reconstruction errors as in the case of the previous models.

Figure 10. Errors corresponding to the reconstruction of test set for Model 3 (left) and Model 4 (right)



Figures 9 and 10 show that there are six time intervals where all the models had difficulties in carrying out the reconstruction of the corresponding liquidity vectors. This result indicates, on one hand, that our methodology is robust to changes in the architectures of the autoencoders, and on the other, that these intervals are potential anomalies. We will delve into the latter at the end of this section.

5.2. Bank runs simulations testing

After the fitting and testing of the four models, we performed bank run simulations to see if the autoencoder was able to flag them as anomalies. The bank run was done in the last 90 time intervals (from 651 to 741) for 2018, on one SPI participant. The simulation consisted of stressing institution b_i outflows toward the rest of participants. More specifically, we introduce the bank run as a random value with exponential distribution and a probability of occurrence for the selected b_i outflows, sampled from a Bernoulli distribution. Following Triepels-Daniels-Heijmans (2018), the corresponding parameters p for the *Bernoulli* and λ for the *Exponential* sampled variables are defined as follows:

$$p(x) = p_s + (p_e - p_s)(\tfrac{x}{d})^{r}$$

$$\lambda(x) = \lambda_s + (\lambda_e - \lambda_s)(\tfrac{x}{d})^{r}$$

Where the subscripts s and e represents starting and ending values for each parameter, $x \in \{1, 2, ..., 90\}$ indicates the time interval, while d is the total number of time intervals, 90, finally r is a rate that controls the increase of $\frac{x}{d}$. This parametrization increases the value of p and λ as time passes, leading to a more intense liquidity adding to the end of the period under the simulation.

Moreover, in Figure 11 and Figure 12, the MSE highlights the simulated bank run in the final time intervals. In these time intervals, the MSE rapidly changed as the payment network unexpectedly began to change as well. Below, we present the MSE of the final liquidity matrices, emphasizing that the high outgoing liquidity flows of the stressed b_i could not be accurately reconstructed, resulting in a high reconstruction error during bank runs. This is observed in all proposed architectures, having that for the TanH models when two layers are used, the autoencoder makes stricter penalizations. It can be explained because the error related to the simulations is larger than the case of one layer. On the other hand, the ReLU autoencoders show a very similar behavior, showing larger errors for both the simulations and the rest of the time intervals.

Figure 11. Bank run simulations results for Model 1 and Model 2

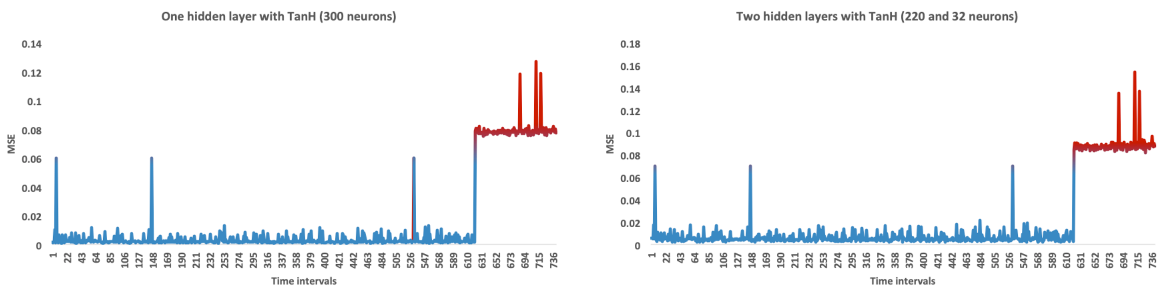
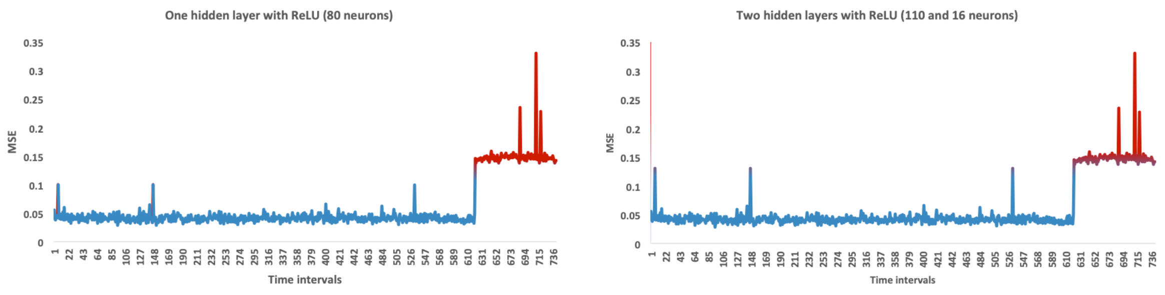


Figure 12. Bank run simulations results for Model 3 and Model 4



With the above it can be confirmed that we have achieved a compression model that is capable of learning the common patterns inherent to the data, and thus is able to recognize anomalous behavior. The next step, once the robustness of the autoencoder has been tested, is to deepen on the analysis of the anomalies detected within the test set in order to have a deeper understanding on autoencoder capacities as an oversight tool.

5.3. Alert analysis

Besides identifying intervals with anomalous patterns of payments with the autoencoder, we investigated which of the flows (one or several) caused these anomalies. In this subsection, we present the main results obtained from the models presented in the previous section, analyzed from the point of view of the system overseer to confirm that the alerts could be considered real anomalies. It is worth noting that there is a concordance in the alert results for all the trained models that indicate unusual payment patterns of systemic importance within the SPI.

The table 2 shows the top ten time-intervals that can be classified as anomalies in the SPI considering our four autoencoder models, the classification criterion is based on the set F defined in subsection 4.1, that is, the anomalous instances will be those that are greater than ε, which in our case is equal to the 90th percentile of the validation MSE of each model. The first 6 alerts (711,718,147,688,532,7) coincide in all models. The time intervals 484, 381, 628, 549, 97 are repeating alerts on some models. While the time intervals 254, 549, 394 are alerts that only Model 1 indicated.

Table 2. Top 10 of time intervals with highest errors for each model

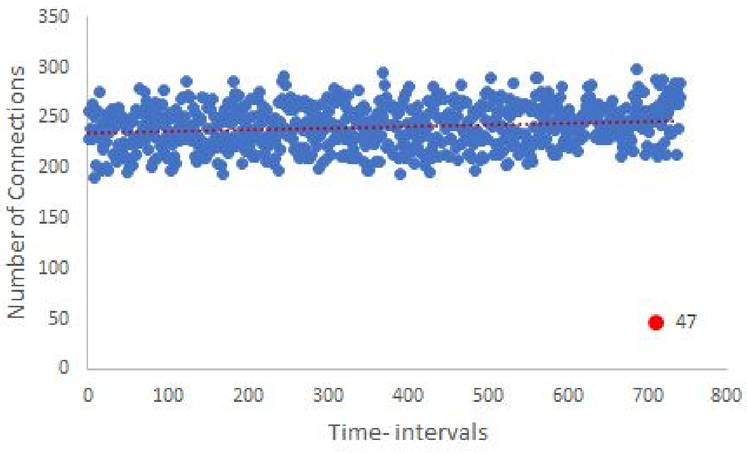
Rank	**Model 1**		**Model 2**		**Model 3**		**Model 4**	
	Time interval	MSE	Time interval	MSE	Time interval	MSE	Time interval	MSE
1	**711**	0.31	**711**	0.39	**711**	0.64	**711**	0.64
2	**718**	0.08	**718**	0.14	**688**	0.18	**688**	0.18
3	**147**	0.08	**532**	0.14	**147**	0.18	**147**	0.18
4	**688**	0.08	**688**	0.14	**718**	0.18	**718**	0.18
5	**532**	0.08	**147**	0.14	**532**	0.17	**532**	0.17
6	**7**	0.08	**7**	0.13	**7**	0.17	**7**	0.17
7	**554**	0.01	**484**	0.05	**484**	0.06	**484**	0.06
8	**254**	0.01	**381**	0.04	**628**	0.06	**628**	0.06
9	**549**	0.01	**549**	0.04	**381**	0.05	**381**	0.05
10	**394**	0.01	**345**	0.04	**97**	0.05	**97**	0.05
Median		0.007		0.028		0.041		0.041
75th percentile		0.008		0.032		0.045		0.045
90th percentile		0.011		0.034		0.049		0.050

It can be underscored that the resulting alerts show the time intervals in which there are fewer than normal payment connections. In particular, we find both intervals for which major banks do not operate and intervals for which medium or small banks do not channel payments to large banks, among other possible patterns. Results show that the autoencoder was able to alert on individual or systemic unusual patterns.

Below, we present the most significant alerts that the autoencoder identified. They are useful examples of how this analysis can support the oversight of the *Sistema de Pagos Interbancarios.*

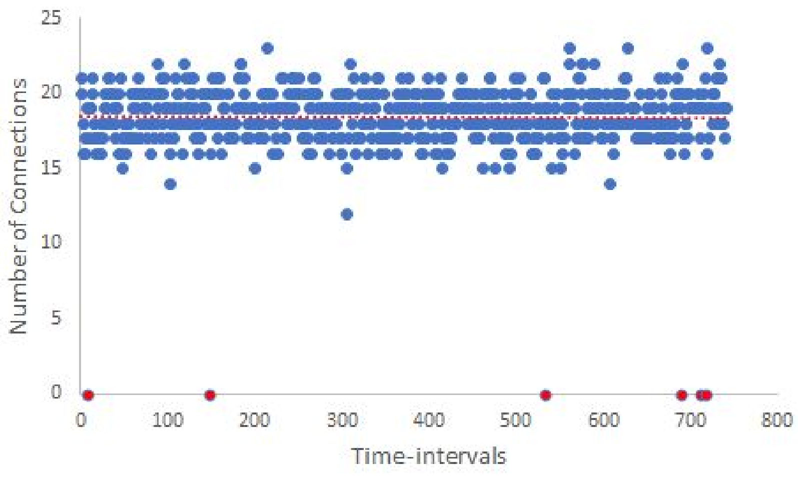
Systemic alert: The most relevant alert was given in time interval 711 (December 2018). This can be explained by an unusual low participation of banks in the SPI. In this case, only 47 payment flows (3 ordering banks) occurred, much lower than the average flows in each time interval, that is 240 payment connections. In Figure 13, we present the flows and amounts for this time-interval.

Figure 13. Alert for unusual low payment connections



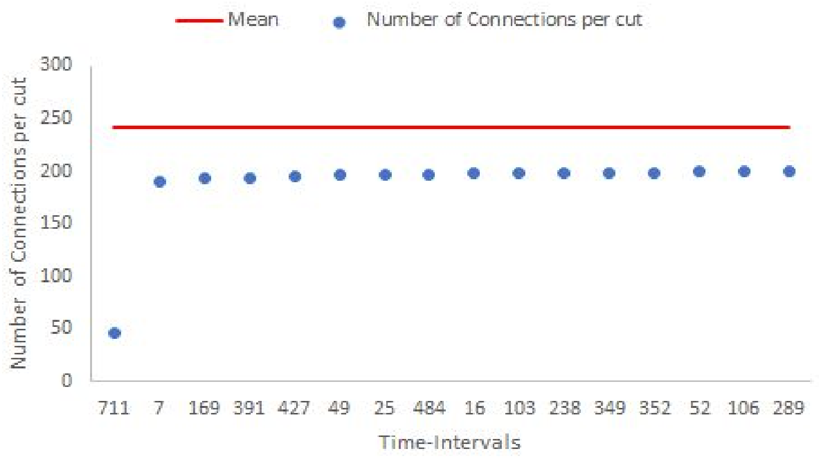
Individual alert for a SPI large participant: This large bank channelized 22% of the total in the SPI and sent and received 11% out of the total operations in 2018. The activity of this large bank is stable along a year time period; it participated in 735 of the 741 time intervals of the year. All the models indicate that there are 6 time-intervals for which the bank did not participate in the SPI at all. As a relevant fact, this bank had 17 payment connections on average over 2018. For example, it can be seen that in intervals 717 and 719, this large bank registered 23 and 24 connections - above the average - but in intervals 7, 147, 532, 688,711, 718, this SPI participant does not have a single connection. Figure 14 shows the activity of this "big bank" in all time intervals of the year.

Figure 14. Alerts for a large bank at times of no payment connections



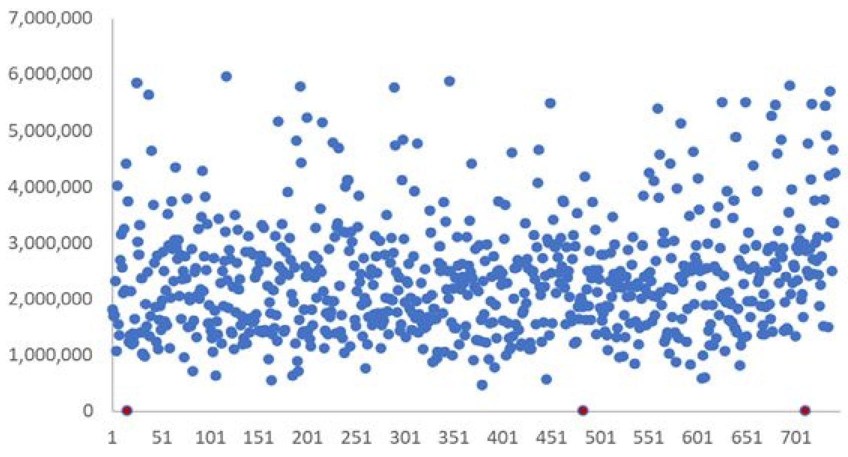
Alerts of low number of participants (and payment connections) in the SPI. In 5% of the time intervals, the number of payment flows is considerably low in comparison with the average 241 connections. The most significant alert is detected in interval 711 with only 47 payment connections. It is worth mentioning that this time interval was associated in the model with a higher alert. Figure 15 depicts the anomalous behavior for such intervals as well as others below the average, all ranked according to their number of connections. These alerts also involve time-interval 484 and 394.

Figure 15. Alerts of low number of participants (and payment connections)



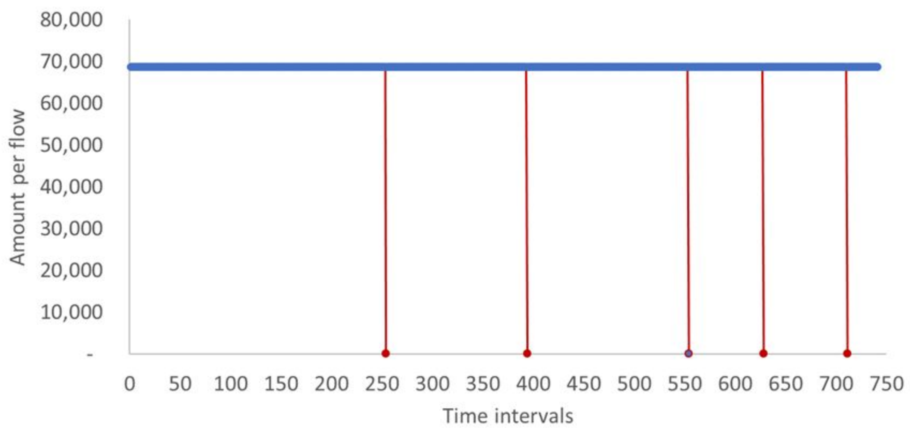
Problem with communication provider and impact on a large bank: The 484 time interval identified by the autoencoder is mainly explained by the non-participation in the SPI of a large bank as well as six other banks (1 medium and 6 little ones) that usually operate. According to the records of operation of the SPI, at this time-interval (see Figure 16) there was a problem of intermittent connection with certain banks with a provider of communication channels, which prevented them from sending operations to the Central Bank of Ecuador in this time interval.

Figure 16. Alert for problem with communication provider



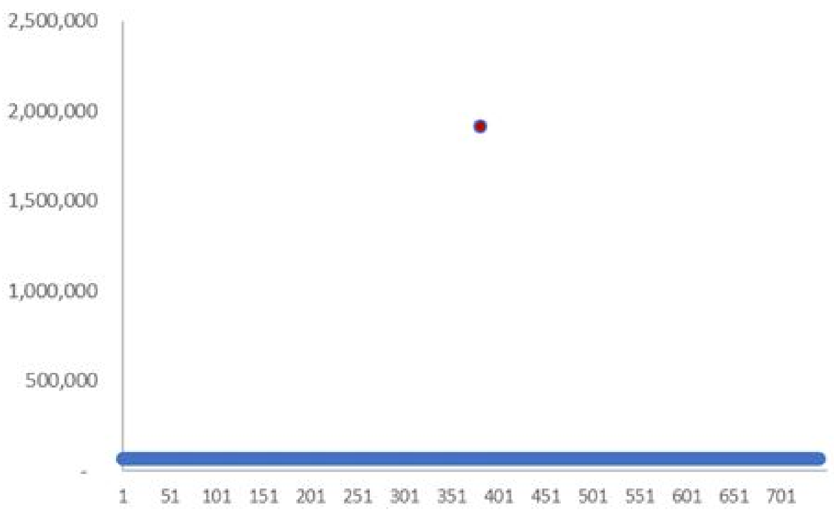
Individual alert for a SPI "average Joe" participant: An alert of a medium bank is found in three intervals (254, 554, 394, 628, 711). The autoencoder detected that in 99.5% of the 2018 intervals, this "average Joe" bank has payment connections with the 5 largest banks participating in the SPI. The alert refers to three intervals for which this bank does not participate in the system, including its regular connections with the 5 largest banks. Figure 17 demonstrates that the "average-Joe" SPI participant was absent in such time-intervals.

Figure 17. Alert for a medium size bank with no payments connections



Unusual payment amounts: In the cases the autoencoder detects events in which the entities send unusual amounts in most cases it indicates the maximum or very low amounts with respect to its normal behavior. These behaviors are evident in the time interval analyzed 381, 549, 345. For example, an entity that on average sends payments for an amount of USD 64,000, the autoencoder alerts when this entity sends USD 1.9 million, the latter being the maximum payment made by the SPI. As shown in Figure 18, it may constitute a significant alert to be analyzed further.

Figure 18. Alert for amounts of payments different from the usual ones



The above alerts provided by the autoencoder models are useful for the oversight and monitoring of the SPI because they allow us to detect payment patterns. The whole SPI dataset cannot be analyzed manually due to the quantity of information that is processed daily, so this methodology provides a powerful tool to equip oversight experts with the ability to identify both normal and anomalous payment patterns.

All in all, interpreting the autoencoder alerts requires the expert judgement of payments oversight teams who have the best understanding of how the system operates, to determine whether the alerts are relevant and if they constitute evidence for pattern changes that could represent a risk.

Finally, the autoencoder can be used as a tool to detect possible operational problems that can cause uncertainty within the system. This is in line with Klee (2010) who used an algorithm to identify outliers in the payment patterns of financial institutions in Fedwire Funds that stem from operational outages. Operational problems cause uncertainty regarding end-of-day Fed account positions that can impact rates in the Federal Funds market. The magnitude of the effects depend on the severity of the difficulty, the time it occurs, and the volume of payments made by the affected participant.

6. Discussion of results

The application of machine learning techniques to support monitoring of financial transactions in SIPS does not replace human decision making but rather it provides new tools to test both simple and complex hypotheses on a large scale over big datasets. This is a critical development for SIPS and FMIs oversight. Autoencoder models can process long time-series of payment networks and based on volume distribution and network structure, they suggest arbitrarily small subsets of transaction periods for further evaluation. In effect, the generality of the autoencoder representation -given by the non-linear decomposition into adaptive hidden units- allows us to extract common patterns in the data and single out uncommon patterns of transactions. The detailed analysis of the properties of this subset of flagged transactions demonstrated the complexity of this automated monitoring procedure, as alerts were generated for a number of different reasons, related to volume, number of connections, or absence of specific institutions.

The autoencoder for anomaly detection is a methodology originally proposed for a real-time payment system (RTGS) by Triepels-Daniels-Heijmans (2018), making the model to identify patterns in real time transactions. In our model, given the SPI does not fully perform as an RTGS but rather as a hybrid, our analysis is more similar to Sabetti-Heijmans (2020) that performs an autoencoder for a DNS system. This confirms the versatility and validity of applying an autoencoder to detect anomalies in payment systems. As we underlined in previous sections, the autoencoder needs to be applied in tandem with the thoughtful review of a payment systems oversight team, to verify the real causes of the alert.

The construction and training of the model is a careful process involving numerous validations and tests that must be carried out, but once the model has been trained its daily application in detecting anomalies in SIPS and FMI operations can take only minutes. This is accompanied by the important and opportune access to data, we were able to use the BCE dataset while fine-tuning the model, as the Central Bank is responsible for operating and overseeing the SPI. This is important as noted by León (2020) who indicates that the application of new methods in payment systems should consider low computational costs and ease in data collection.

This document contributes to identify evidence of incidents in the payments flow of a respective system, and thereby provides new tools for payments oversight, and it ultimately sets the basis for early warning tools. We were able to detect anomalies in the payments flows processed by banking entities in the major SIPS Ecuador, the *Sistema de Pagos Interbancarios*. We proposed four models to test autoencoder robustness, where we selected the best architectures by performing cross-validation. These models were trained on real banks behavior stemming from the payments network of the SPI for 2018, which makes our work novel and relevant. All the autoencoders we tested identified relevant anomalous patterns, finding problems in the reconstruction of the data and flagging specific payment networks as anomalous. In order to evaluate the models, a bank run simulation was performed, altering a major SPI participant's outgoing payment flows exponentially over a period of time.

In our approach, we were able to identify alerts that could affect the system, such as: 1) non-participation of systemically important participants, 2) a low number of connections (payment flows), 3) medium-size banks not sending payments to systemically important participants, among others. Additionally, the bank run simulation shows the ability of the autoencoder to detect risk events that may be generated in the SPI in the absence of common patterns. Importantly, we deepened the analysis of the alerts that the autoencoder signaled as potential anomalies by relying on the expert judgement of overseers of the SPI.

Further studies in machine learning for anomaly detection in payment systems can improve the accuracy, reliability, and speed of the methodology leading to financial alerts that are more spot-on, consistent, and that can be performed in real-time. Notwithstanding, these techniques should not be intended to replace the knowledge in payment systems oversight teams but rather become part of their toolkit.

References

Aggarwal, Charu C., and Philip S. Yu. "Outlier detection for high dimensional data." Proceedings of the 2001 ACM SIGMOD international conference on Management of data. 2001.

Aleskerov, Emin, Bernd Freisleben, and Bharat Rao. "Cardwatch: A neural network based database mining system for credit card fraud detection." Proceedings of the IEEE/IAFE 1997 computational intelligence for financial engineering (CIFEr). IEEE, 1997.

A. Arning, R. Agrawal, P. Raghavan. A Linear Method for Deviation Detection in Large Databases. KDD Conference Proceedings, 1995.

V. Barnett, T. Lewis. Outliers in Statistical Data. John Wiley and Sons, NY 1994.

Ben-Gal, Irad. "Outlier detection." Data mining and knowledge discovery handbook. Springer, Boston, MA, 2005. 131-146.

Brause, Rüdiger, T. Langsdorf, and Michael Hepp. "Neural data mining for credit card fraud detection." Proceedings 11th international conference on tools with artificial intelligence. IEEE, 1999.

Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." ACM computing surveys (CSUR) 41.3 (2009): 1-58.

Committee on Payment and Market Infrastructures and Technical Committee of the International Organization of Securities Commissions (CPMI-IOSCO). 2012a. "Principles for Financial Market Infrastructures." (April).

Dorronsoro, Jose R., et al. "Neural fraud detection in credit card operations." IEEE transactions on neural networks 8.4 (1997): 827-834.

Ghosh, Sushmito, and Douglas L. Reilly. "Credit card fraud detection with a neural-network." System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on. Vol. 3. IEEE, 1994.

Goodfellow I., Bengio Y., and Courville A. "Deep Learning" MIT Press, 2016.

Hadi, A.S. A modification of a method for the detection of outliers in multivariate samples. Journal of the Royal Statistical Society, B, 56(2), 1994.

Hawkins, Simon, et al. "Outlier detection using replicator neural networks." International Conference on Data Warehousing and Knowledge Discovery. Springer, Berlin, Heidelberg, 2002.

Hodge, Victoria, and Jim Austin. "A survey of outlier detection methodologies." Artificial intelligence review 22.2 (2004): 85-126.

Klee, E. (2010). Operational outages and aggregate uncertainty in the federal funds market. *Journal of Banking & Finance*, *34*(10), 2386-2402.

León-Rincón, C. E., (2020). Detecting anomalous payments networks: A dimensionality reduction approach. *Latin American Journal of Central Banking* 1 (1-4).

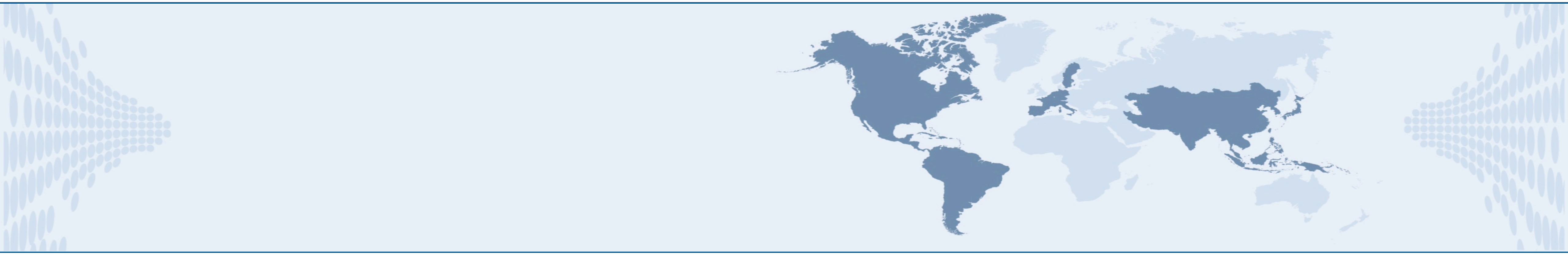
E. Knorr, R. Ng. Algorithms for Mining Distance-based Outliers in Large Data Sets. VLDB Conference Proceedings, September 1998.

Sabetti, Leonard, and Ronald Heijmans. "Shallow or deep? Detecting anomalous flows in the Canadian Automated Clearing and Settlement System using an autoencoder." (2020).

Triepels, Ron, Hennie Daniels, and Ronald Heijmans. "Anomaly Detection in Real-Time Gross Settlement Systems." ICEIS (1). 2018.

Williams, G. J., Baxter, R. A., He H. X., Hawkins S. and Gu L. (2002) "A Comparative Study of RNN for Outlier Detection in Data Mining" IEEE International Conference on Data-mining (ICDM'02), Maebashi City, Japan, CSIRO Technical Report CMIS-02/102.

CEMLA
CENTER FOR LATIN AMERICAN MONETARY STUDIES



Classifying payment patterns with artificial neural networks: An autoencoder approach

Jeniffer Rubio, Paolo Barucca, Gerardo Gage, John Arroyo, Raúl Morales-Resendiz

Introduction

Payments and market infrastructures are the backbone of modern financial systems and play a key role in the economy by enabling multilateral transactions under certain rules and platforms. One of their main goals is to manage systemic risk, especially in the case of systemically important payment systems serving interbank funds transfers. Thus central banks need to be able to monitor their activity and to identify anomalous events.

Thanks to the availability of high data volumes and to the increment in computation capabilities it is posible to develop tools that automatically performs tasks as the identification of anomalous patterns.

In this vein, we developed a methodology based on a unsupervised neural network, the autoencoder, to detect a diverse set of anomalies arising within the *Sistemas de Pagos Interbancarios* (SPI) from Ecuador. It was found that the methodology is robust enough to support the monitoring of payment systems, but need to be acompained by the expert judgement of payments overseers.

Methodology

The data we study contains information on the liquidity vectors, we aimed to reconstruct the vectors by compressing and decompressing the dataset under use. For this purpose, we implemented a lossy compression, which generates a particular type of representation that allows the data to not be exactly learnt and some of the information to be lost.

When this type of compression is implemented, the relevant patterns present in the data are learned. Once the compression model learns the common patterns and a new liquidity vector is fed for its reconstruction; if the reconstruction is bad, this is explained by the fact that vector information differs from the normal patterns that the model learned, indicating the possibility of a potential anomaly.

The quality of the reconstructions will be measured through the reconstruction error which is the difference between the values yielded by the lossy compression and the real vector values.

Methodology

Anomaly detection framework

- $B = \{b_1, \dots, b_n\}$ set of SPI participants
- $T = \{t_1, \dots, t_m\}$ ordered set of m time intervals
- $a_{i,j}$ total amount of liquidity transferred from institution b_i to institution b_j
- $A^{(k)}$ liquidity matrix for the k-th interval

$$A^{(k)} = \begin{pmatrix} a_{1,1}^{(k)} & \cdots & a_{1,n}^{(k)} \\ \vdots & \ddots & \vdots \\ a_{n,1}^{(k)} & \cdots & a_{n,n}^{(k)} \end{pmatrix}$$

- $\bar{a}^{(k)}$ liquidity vector ($A^{(k)}$ rearrangement)
- $RE: D \rightarrow [0, \infty)$, non-negative function that measures the reconstruction error of liquidity vector, where *D* is the set of all liquidity vectors

Our goal is to find the set $F = \{\bar{a}^{(k)} \in D \,|\, RE(\bar{a}^{(k)}) \geq \epsilon\}$, where $\epsilon > 0$

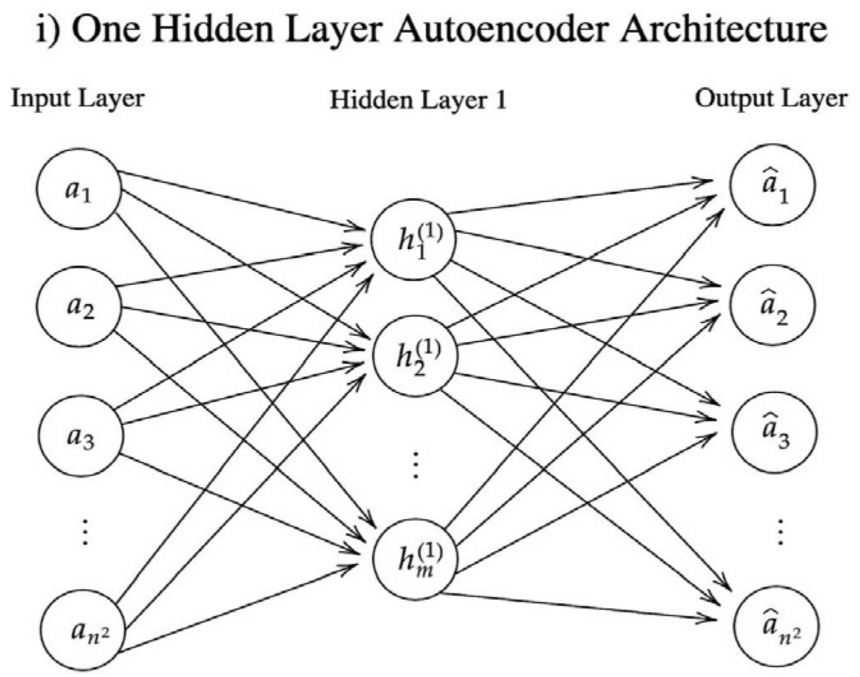
CEMLA
CENTER FOR LATIN AMERICAN MONETARY STUDIES

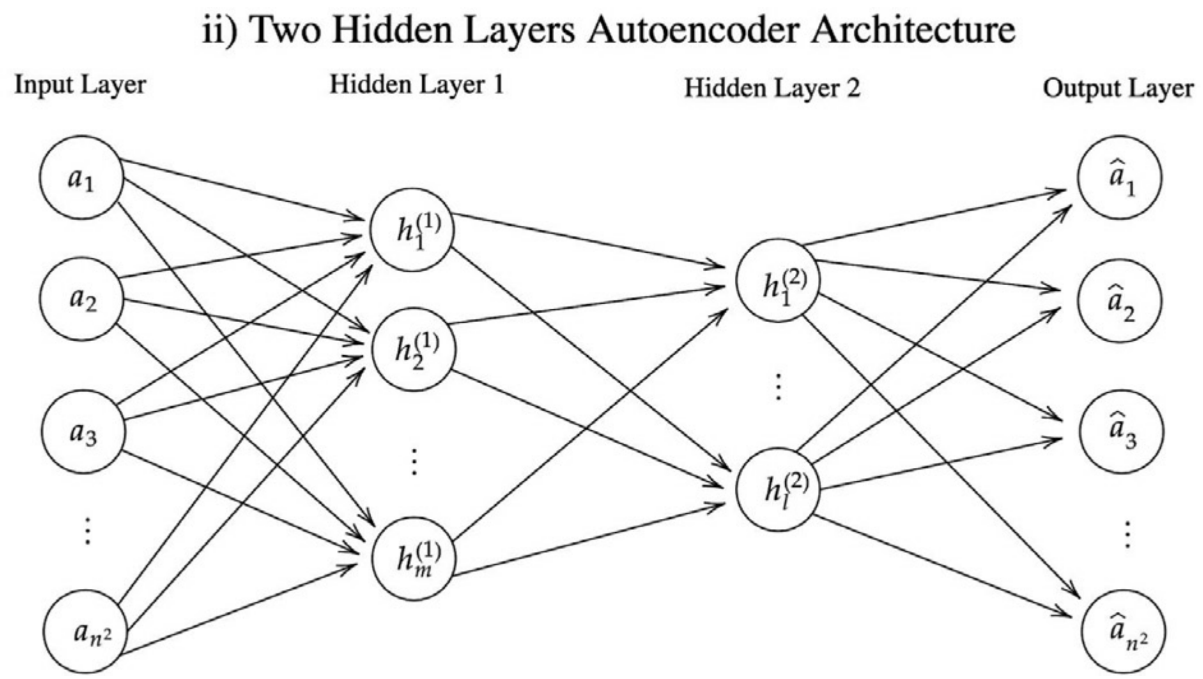
Methodology

For our work, we selected the autoencoder as the lossy compression model, which is an unsupervised neural network.

The autoencoder is made up of two components, the encoder and the decoder. The encoder is the initial part of the autoencoder and it has the task to create an accurate lower-dimensional representation of the data. The second part of the autoencoder, the decoder, is in charge to carry out the reconstruction of the data. The encoder goes from the input layer to the layer with the lowest number of neurons, which is commonly called bottleneck given that is the layer of the network where data is the most compressed. The encoder can be represented as a function $h = f(X)$, where X represents the input data. On the other hand, the decoder goes from the *bottleneck* to the output layer, it can be represented by the function $r = g(h)$. The autoencoder final objective is to find f and g such that $\mathrm{X} \approx g(f(X))$.

Methodology





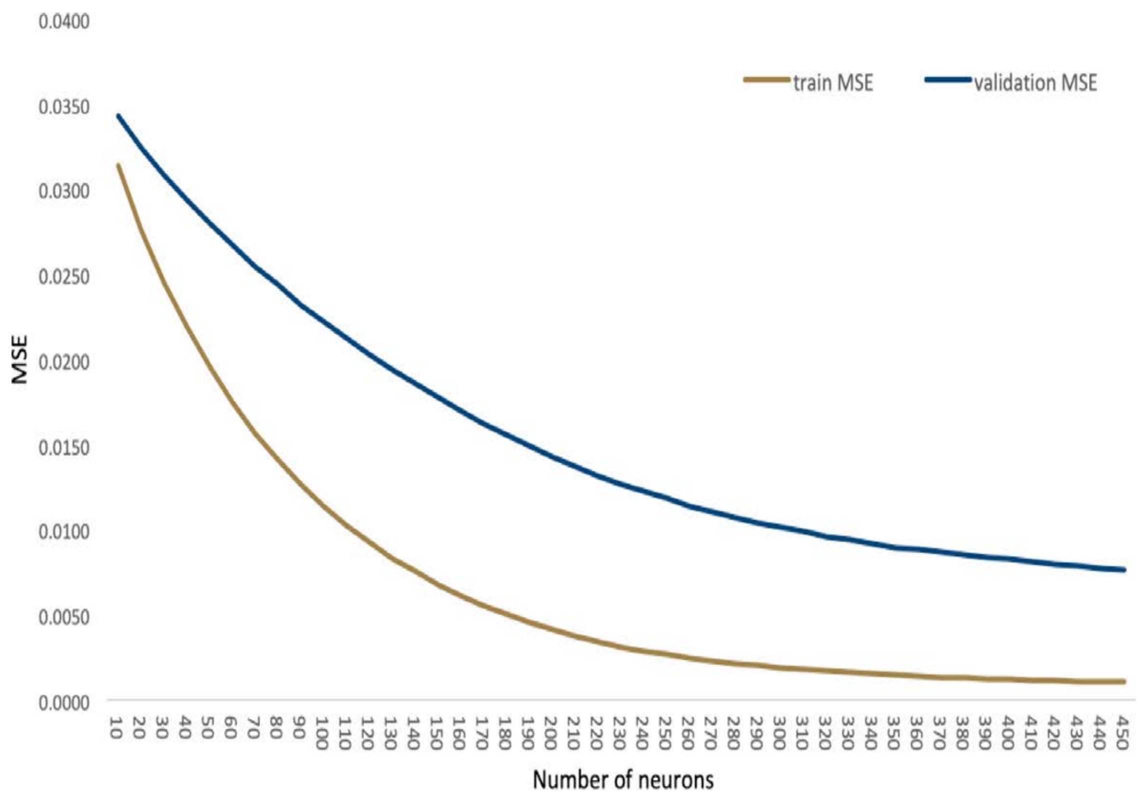
CEMLA
CENTER FOR LATIN AMERICAN MONETARY STUDIES

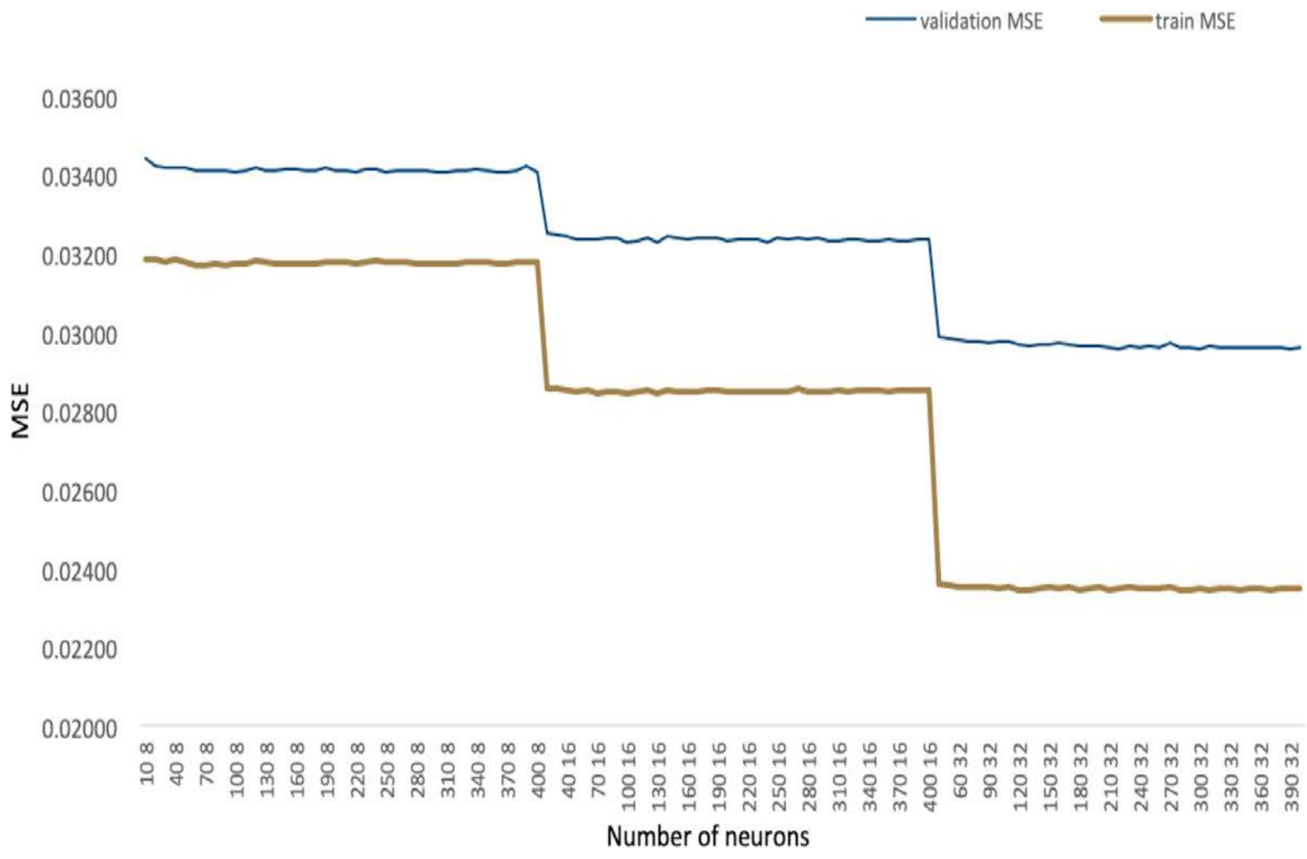
Model fitting, selection and testing

- Before fitting the model, data was pre-processed, this corresponded to a log-transformation and standardization.
- Then the data was partitioned into cross-validation and test sets in a proportion of 80% and 20%, respectively.
- We analyzed different setups for the autoencoder by varying the number of hidden layers, the number of neurons in each layer and the activation functions (TanH and ReLU), this led us to the definition of four different models
  - Model 1: Model 1: One hidden layer with TanH as the activation function.
  - Model 2: Two hidden layers with TanH as the activation function.
  - Model 3 One hidden layer with ReLU as the activation function.
  - Model 4: Two hidden layers with ReLU as the activation function.

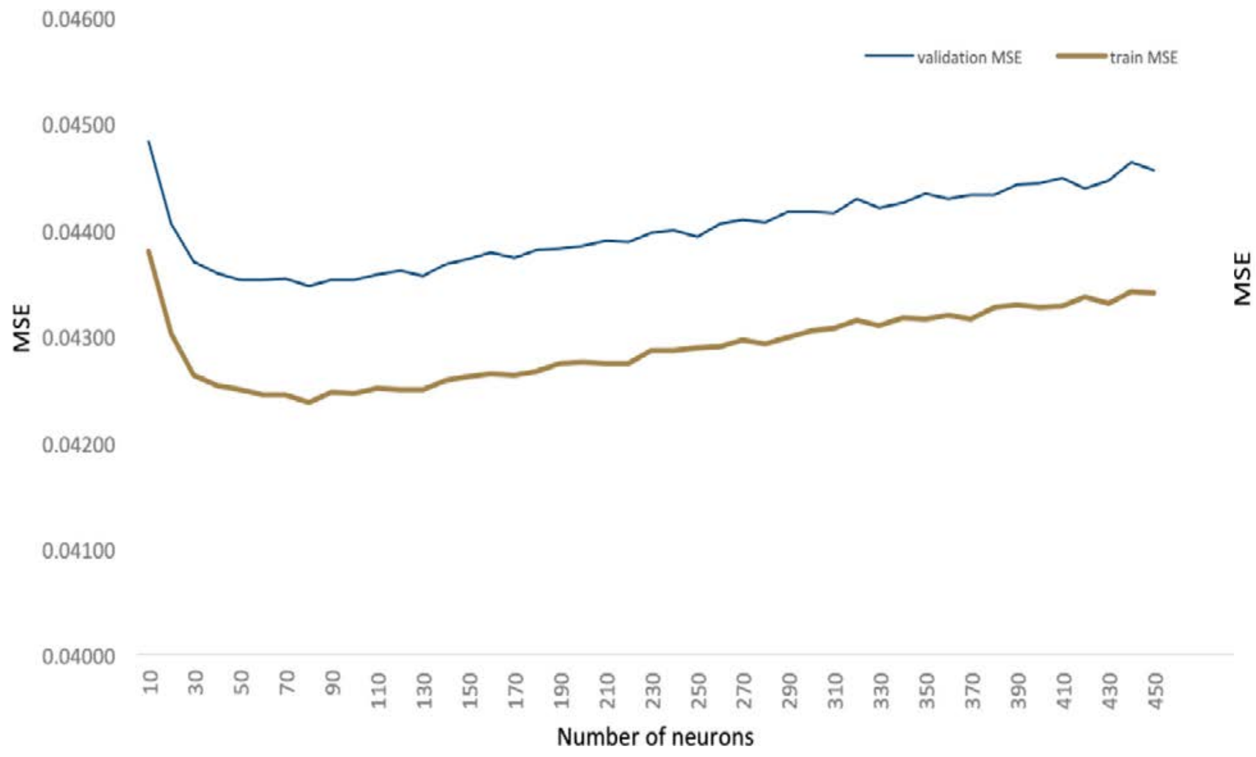
The number of neurons in each hidden layer (hyperparameters) were selected through a 5-fold crossvalidation.

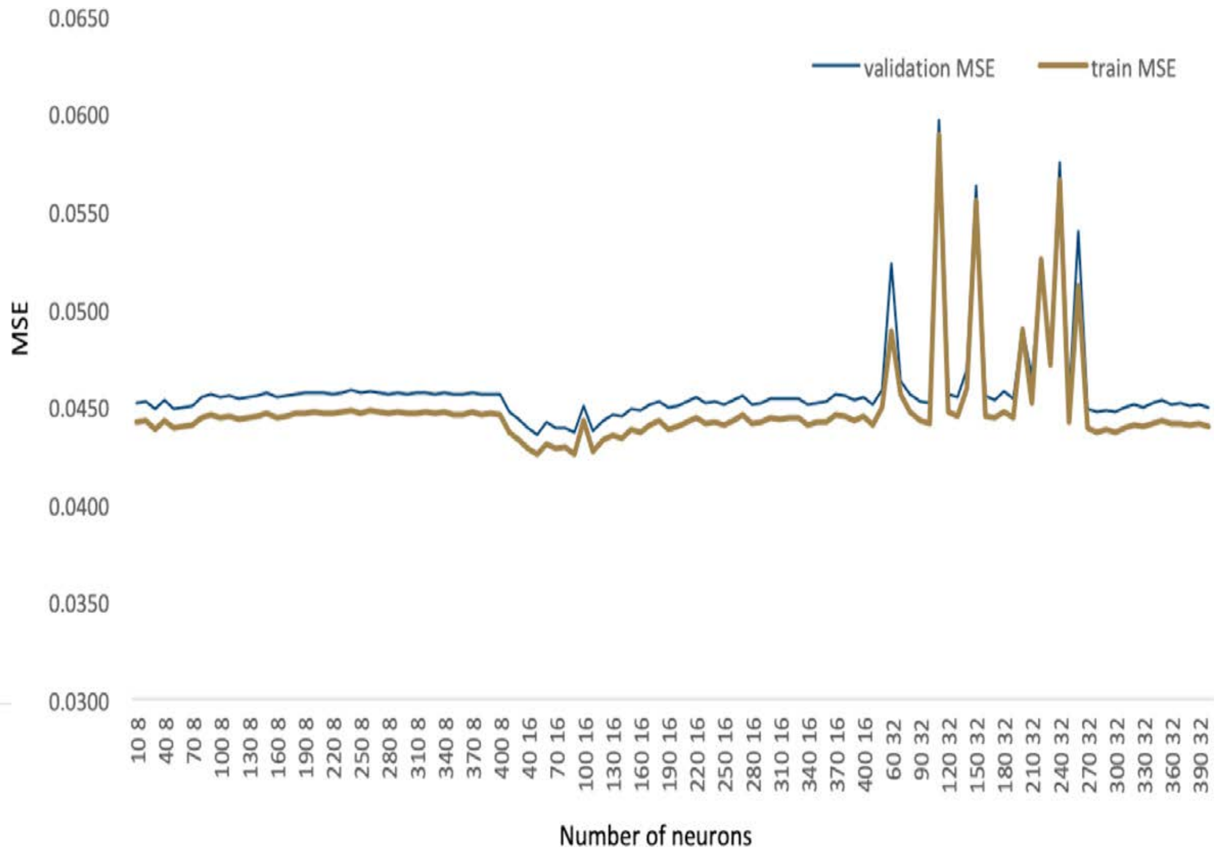
TanH models results





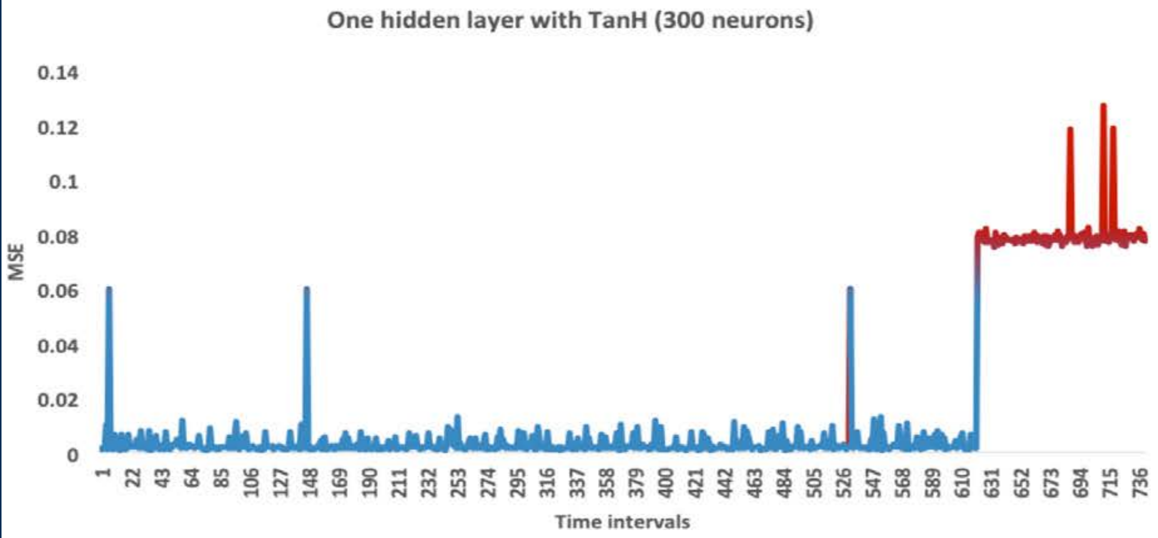
ReLU models results

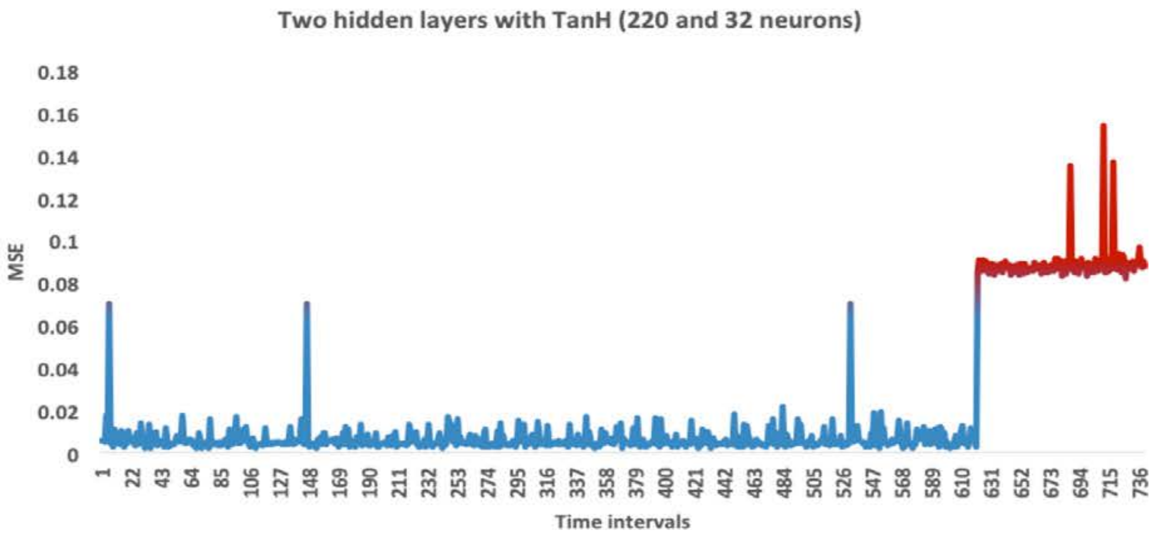


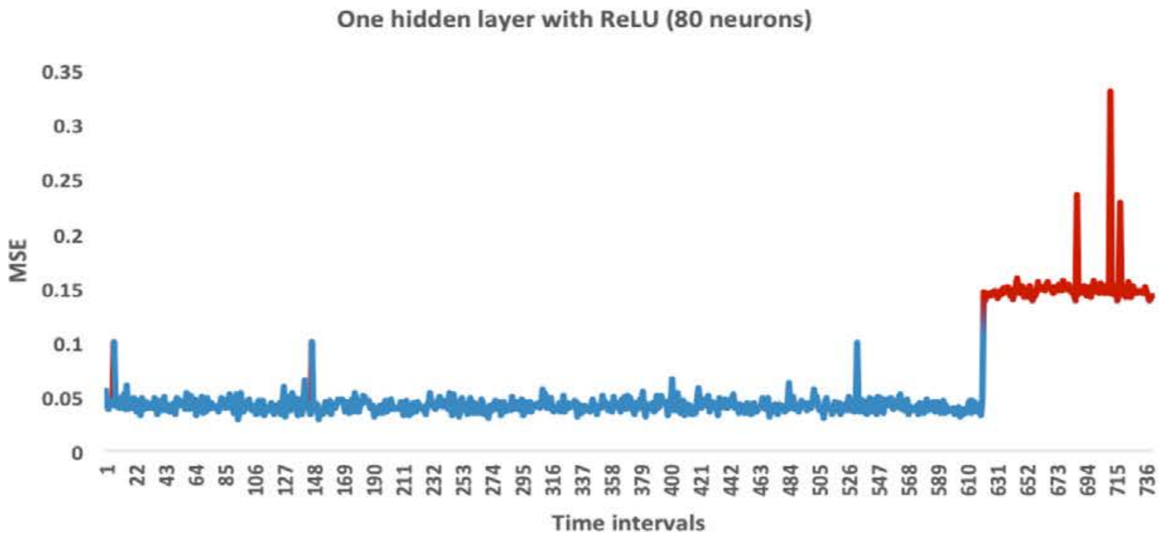


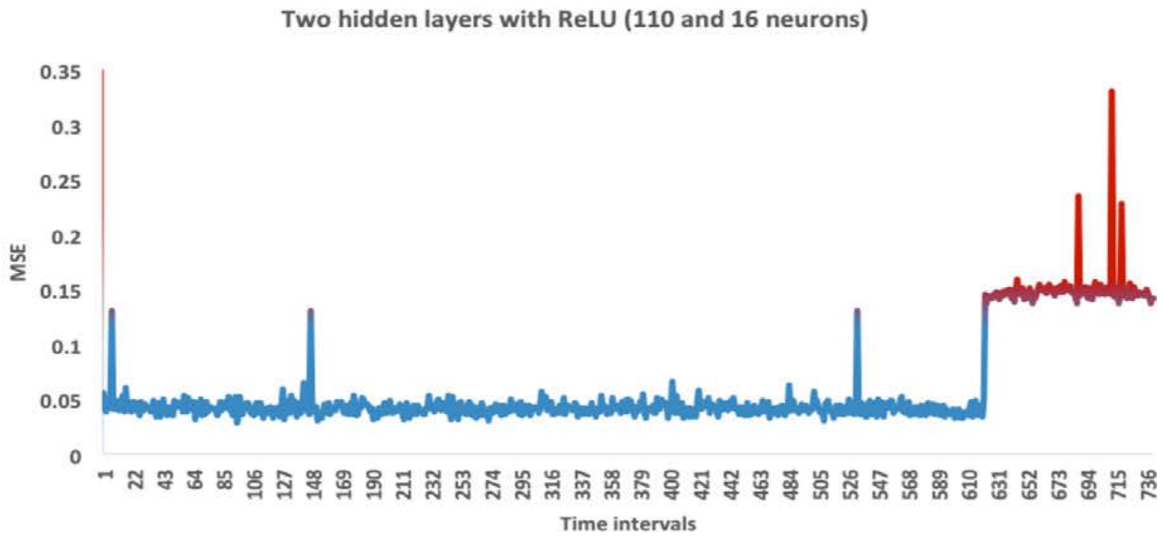
CEMLA
CENTER FOR LATIN AMERICAN MONETARY STUDIES

Bank run simulations



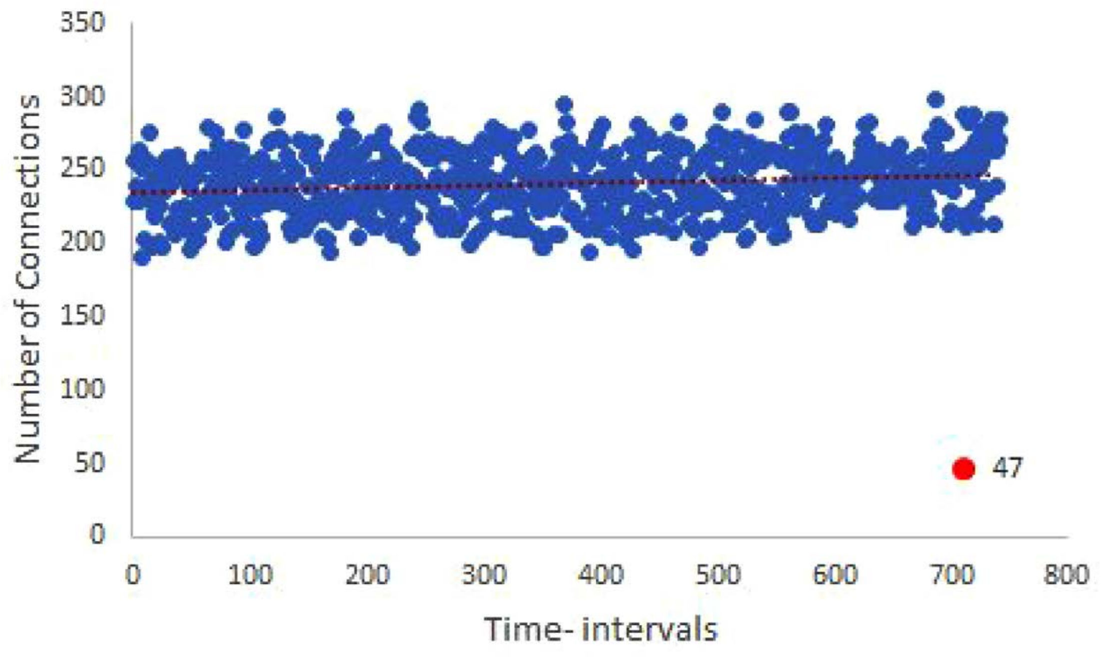


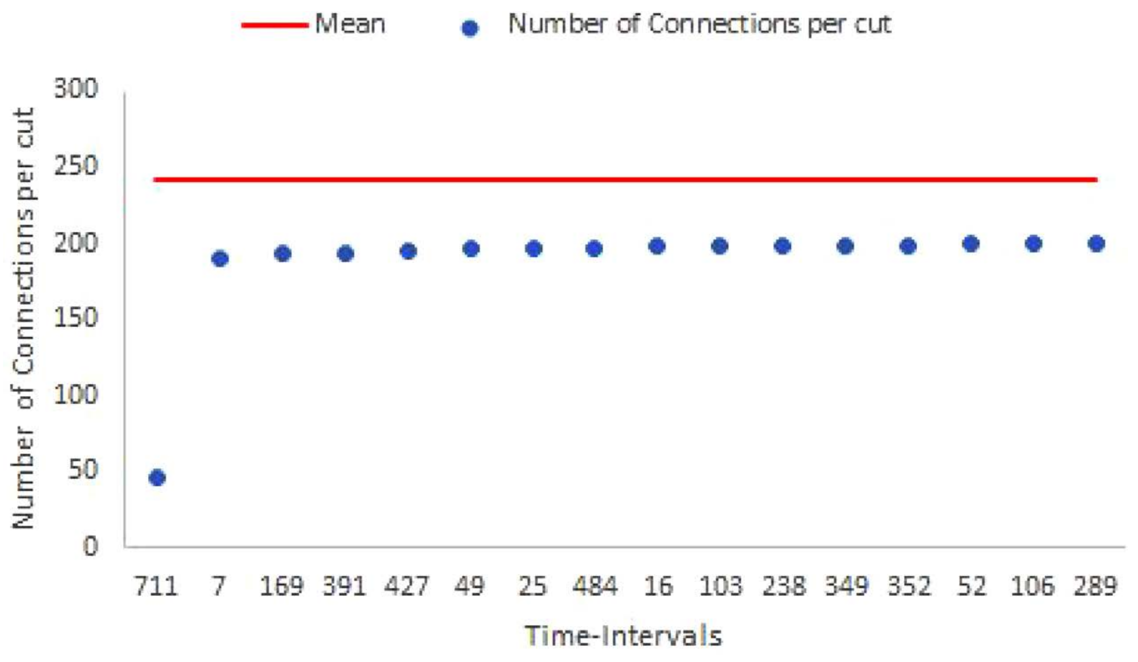




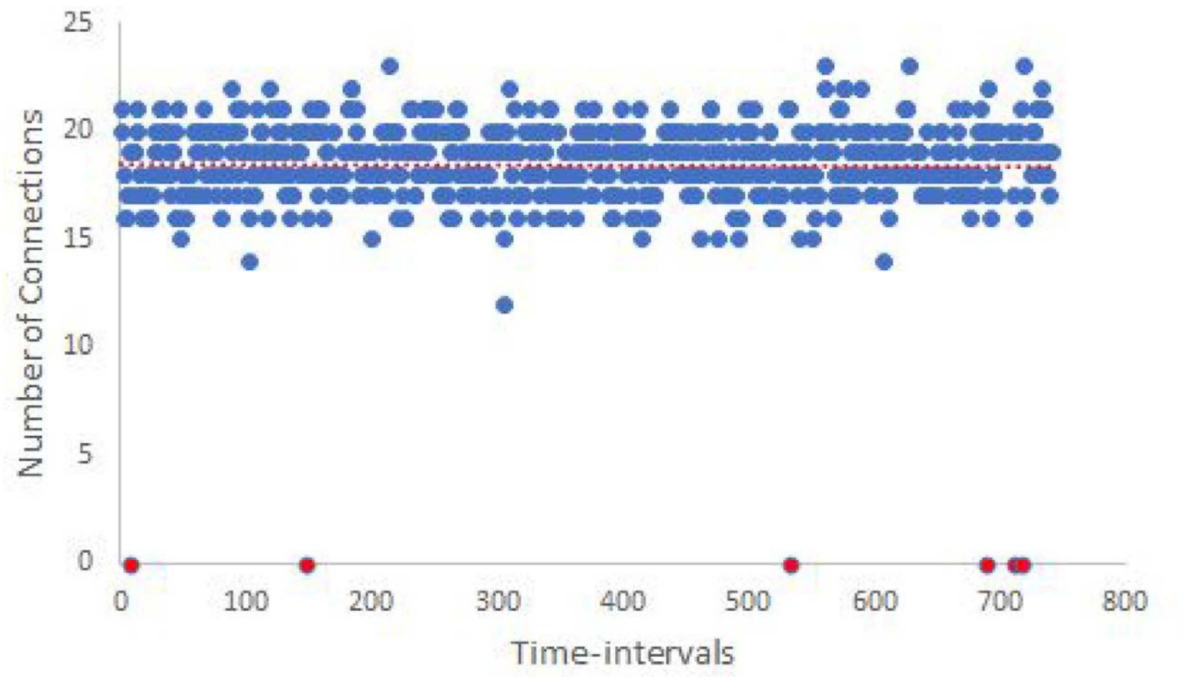
CEMLA
CENTER FOR LATIN AMERICAN MONETARY STUDIES

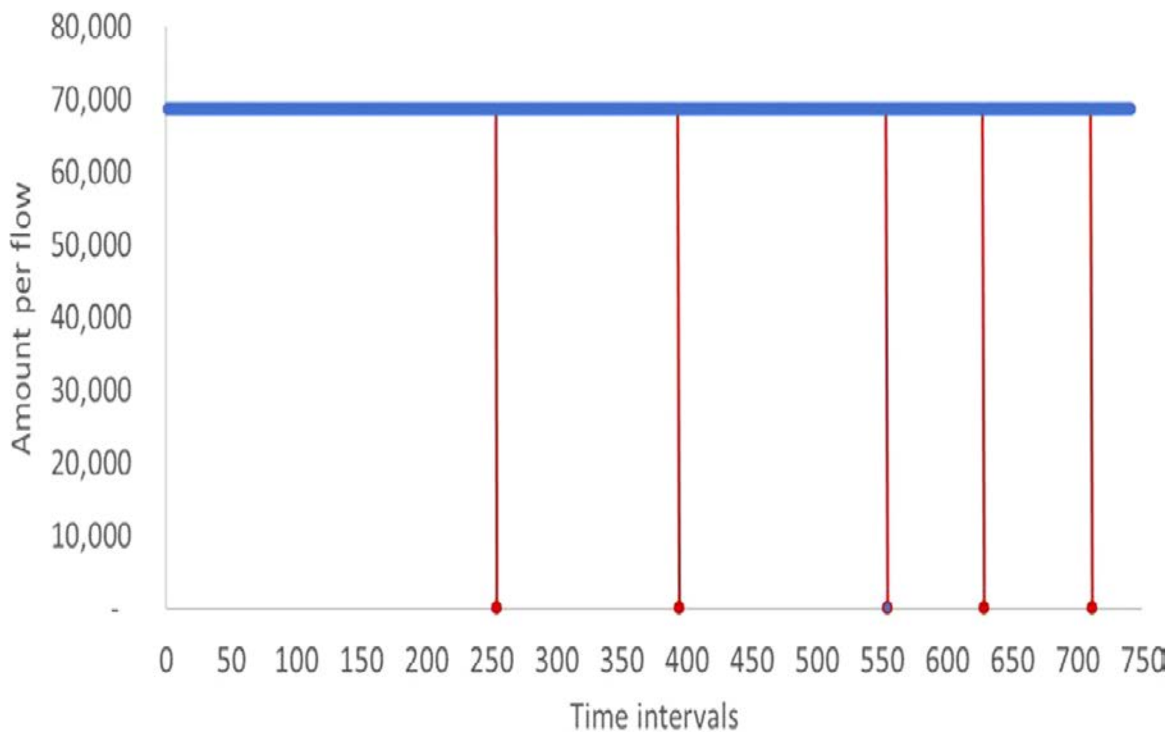
Alerts analysis





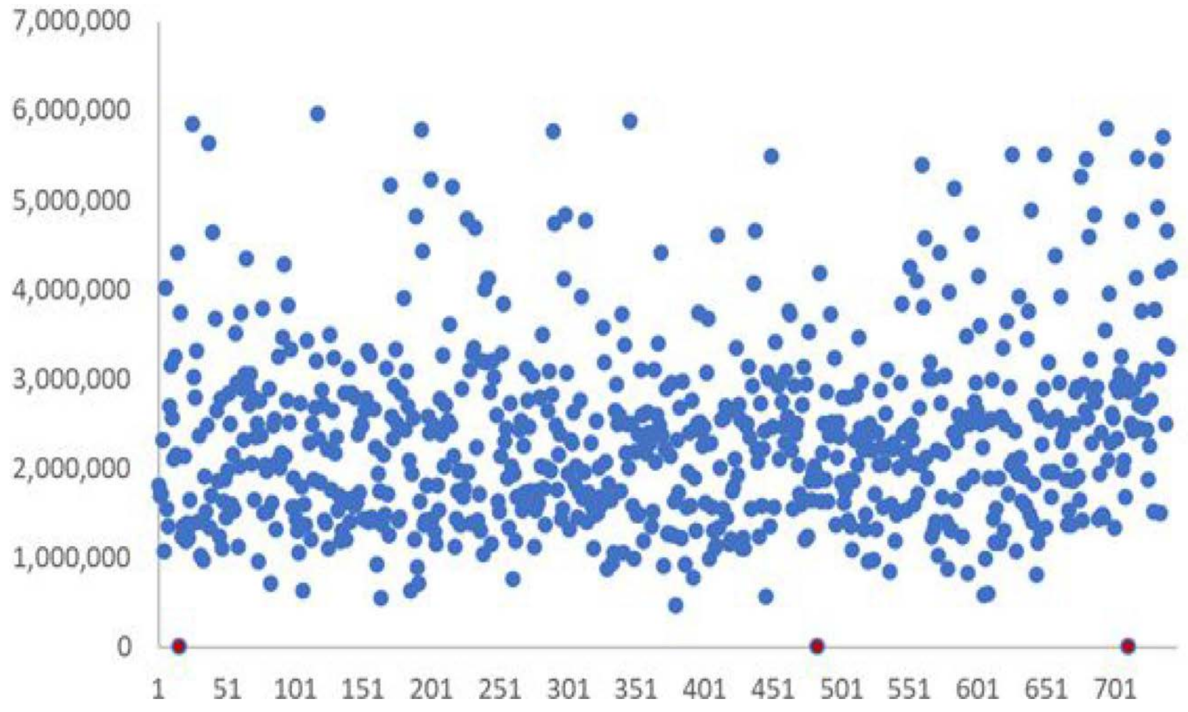
Alerts Analysis

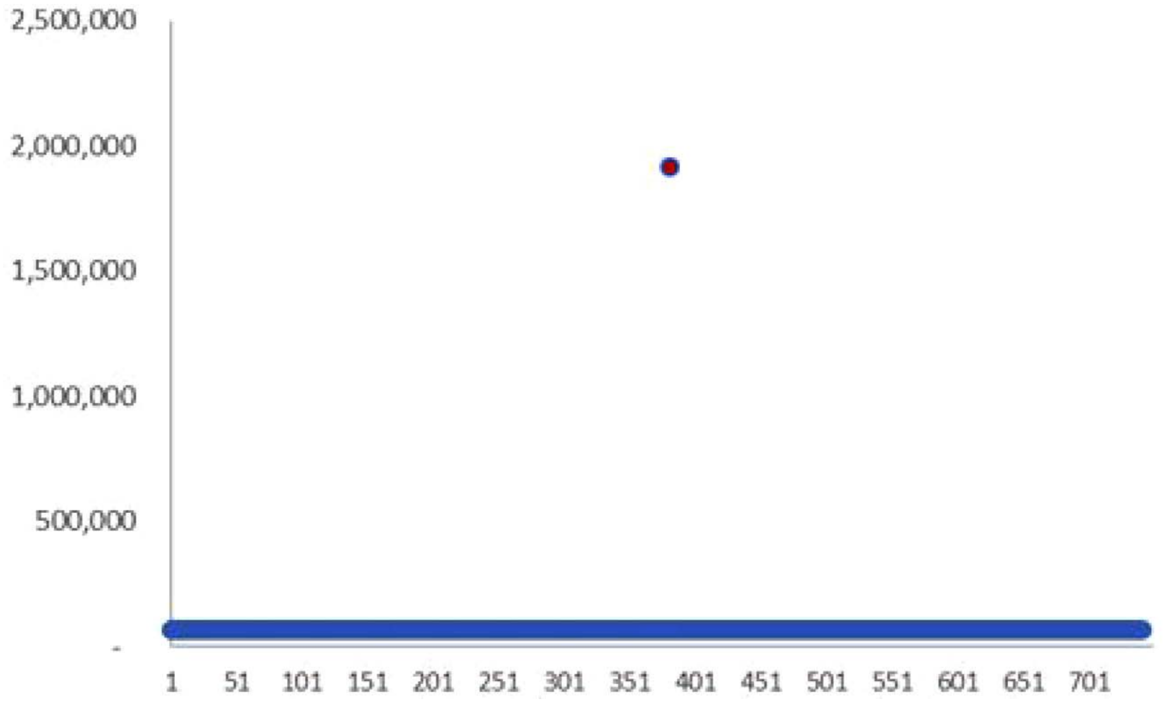




CENTER FOR LATIN AMERICAN MONETARY STUDIES

Alerts Analysis





Conclusions

- The application of machine learning techniques to support monitoring of financial transactions in SIPS does not replace human decision making but rather it provides new tools to test both simple and complex hypotheses on a large scale over big datasets .
- The generality of the autoencoder representation -given by the non-linear decomposition into adaptive hidden units- allows us to extract common patterns in the data and single out uncommon patterns of transactions.
- Further studies in machine learning for anomaly detection in payment systems can improve the accuracy, reliability, and speed of the methodology leading to financial alerts that are more spot-on, consistent, and that can be performed in real-time
- These techniques should not be intended to replace the knowledge in payment systems oversight teams but rather become part of their toolkit