Anomaly Detection in RTGS Systems: Performance Comparisons Between Shallow and Deep Neural Networks

Luca Arciero[1], Giuseppe Bruno[1], Sabina Marchetti[1], Juri Marcucci[1]

Via Nazionale, 91, Rome. Italy

Abstract

Anomaly detection is a critical application in many fields, such as fraud detection in finance or the timely discovery of banking liquidity shortages. However, reaching a high level of accuracy (i.e. low false alarms along with few misses) is a very challenging task. The aim of this paper is to use deep neural networks belonging to the family of Autoencoders, in order to evaluate their performance in the timely discovery of anomalous payment patterns in the Italian component of the Interbank Payment System TARGET2. The paper extends the work by Triepels et al. (2017) using deeper autoencoders. We find that increasing the depth of the Autoencoder allows an accurate evaluation of the network of participating institutions. Furthermore Autoencoders can easily detect changes in the liquidity flows between participating banks.

Keywords: Deep Neural Networks, RTGS, Bank run, Autoencoder

1. Introduction

The liquidity crisis that followed the recent Great Financial Crisis has spurred an increasing interest of policy makers in quantitative analysis of large value payment flows among banks for financial stability purposes. To this purpose, data from payments systems are considered one of the key sources of macro prudential quantitative

analysis. These data are usually available almost in real-time and with high quality.

Our application focuses on TARGET2 payments data. TARGET2 is the second generation of the Trans-European Automated Real-Time Gross Settlement (RTGS) system. It is owned by the Eurosystem and operated by Banca d'Italia, Bundesbank and Banque de France on behalf of the Eurosystem itself.

Both central banks and commercial banks can submit payment orders to the platform, with no upper limit for their value. These are processed and settled in central bank money, debiting and crediting the settlement accounts (*Payments Module* accounts) that each participant has to hold at one of the Eurosystem central banks. The TARGET2 system has gained a leading position in the Payment Systems landscape both in Europe and in the rest of the world, thanks to the over 5,000 credit institutions using it to initiate transactions in euro, either on their own behalf or on behalf of their own customers.

The platform settles real-time payments on an individual basis (gross settlement), provided that sufficient liquidity is available in the participant banks' accounts. Due to its gross settlement feature, TARGET2 eliminates credit risk among participants. In fact, regardless of some participants' potential insolvency, transactions deemed as final and irrevocable according to the system rules cannot be either voided or reversed. These are legally enforceable and binding on third parties. Unlike Deferred Net Settlement (DNS) systems that only settles net balances, resulting from the daily cumulative streams of incoming and outgoing payments, RTGS systems require their participants to maintain sufficient liquidity on their accounts throughout the whole business day, in order to execute their payments without undue delays. In this context, being TARGET2 a network that interlinks all participants, if one participant fails to fulfill its obligations, this can endanger the liquidity position of its recipients and, in turn, it can cause the latter to be unable to meet its own payment obligations

versus a third participant in time. This could cause a domino effect in the network.

One of the main goals of TARGET2 is actually that of minimizing systemic risk in the financial system. This is defined as the risk of a financial system collapse, induced by a cascading failure of one or more entities, via the interlinkages among financial institutions. Failures arise from disruptions to the payment systems, credit flows and destruction of asset values, as well as bank runs, whose spillover effects on creditor banks expose the whole system to collapse. Both monitoring and proactive timely detection of anomalies, such as illiquidity circumstances or even bank runs, play a crucial role in the assessment of the build-up of systemic risk.

As an example of the materialization of such risk, consider the events from the US System in 2008. On September 15th Lehman Brothers filed for *Chapter 11* bankruptcy, due to an intraday liquidity shortage. This prevented it from providing enough liquidity to its clearing banks to fund its payment obligations. In a scenario stressed by the sub-prime mortgage crisis, the inability of the firm to settle its payments was perceived as a situation of potential insolvency. As a consequence, credit lines were withdrawn by its counterparties, leading to the subsequent bank's collapse, see for example Ball et al. (2013). Following the bankruptcy of Lehman Brothers, uncertainty about the financial soundness of major banks worldwide gave rise to a dramatic shrinkage in the activity of several financial markets. In particular, this affected the money market, that represents one of the available sources for payment system participants to cope with their liquidity needs. Injection of unconventional liquidity from central banks contributed to secure the Payment Systems from the liquidity risk. Stress tests carried out by the Eurosystem, simulating extreme shocks to the value of eligible collateral with a decrease of the intraday credit lines and payment capacity of the TARGET2 participants, showed resilience of the system to stress scenarios (see European Central Bank (2017)). Similarly, Banca d'Italia has

run stress tests to assess the ability of the Italian banks to withstand liquidity shocks in the TARGET2 settlement system. Such tests indicated that banks had enough liquidity to cope with further freezings in the money market or with a blockage of inflows from their main counterparty, see Banca d'Italia (2010, 2012).
At the time of writing, liquidity risk in TARGET2 appears to still be extremely low. Nonetheless, the end of the Quantitative Easing program will likely be accompanied by an increased sensitivity of participants to intraday liquidity imbalances. This will deserve a careful monitoring to promptly detect idiosyncratic illiquidity conditions, that could potentially materialize in systemic risk.

According to Gerlach (2009), three main approaches are currently used for measuring systemic risk:

1. Indicators of financial soundness or financial stability;
2. Measures on the state of single institutions; e.g. the intraday liquidity monitoring tools developed by the Basel Committee on Banking Supervision (2013);
3. Gauging interlinkages and dependencies between financial institutions.

However, all these methods are affected by data availability issues, that undermine their effective real-time application.

The aim of the present paper is to extend previous work by Triepels et al. (2017). To do so, we implement and gauge the performances of a special type of Artificial Neural Network, called *Autoencoder*, to model the behavior of payment flows, while detecting anomalous deviations from the patterns learned from the past. The rest of the paper is organized as follows: first, we review the literature on payments data and the state of the art of their modelling in Sec. 2. In Sec. 3 we describe the Italian TARGET2 data we used for our application, and describe all the steps to our methodology, based on Autoencoders. We discuss our main findings in Sec. 4,

and state some final remarks in Sec. 5. All figures and tables may be found in the Appendix.

2. Related Work

A host of papers on payments data can be found in the literature in which the authors study the properties and the behavior of banks in settlement systems. For example, looking at the Federal Reserve's Fedwire Funds Transfer service, McAndrews and Rajan (2000) show that the highest daily concentration of funds-transfer value occurs in the late afternoon. This should be the result of an attempt of banks and customers to coordinate the submission of payments, aimed to benefit incoming transactions to settle with outgoing payments. For the Italian payments system, Arciero and Impenna (2007) detect regular intraday and daily patterns of payments settled in the Italian RTGS system related to operational deadlines such as end of reserve maintenance, tax collection, and settlement of monetary policy operations, while Massarenti et al. (2012) identify intraday patterns of interbank payments in TARGET2. Additionally, Arciero et al. (2016) study how to identify euro-wide unsecured loans with maturities below one year, based on payment data from TARGET2.

Another thread of literature stresses the role of network interconnectedness in causing financial contagion and systemic risk. In this vein, Allen and Gale (2000) explain the origin of financial contagion from the lack of complete links among financial institutions, while Huang and Xu (2000) focus on financial crises, based on the interplay of corporate sector and the interbank liquidity market. León and Pérez (2014) study the Colombian financial market infrastructures by proposing two centrality graph measures. Finally, Berndsen et al. (2016) analyze the consequences of interactions between the network of Financial Institutions and the Market Infrastructure where exchanges take place. The insights achieved from those studies

were eventually employed to develop indicators of liquidity and systemic risk: see for example Heijmans et al. (2014) on the problem of classification of banks' liquidity problems using payment information from Large Value Payment Systems.

Anomaly detection was also successfully applied on other types of financial data, such as stock market data, tackling violation of securities laws. The latter ones include detection of unprofitable trades by brokers, and abnormal stock price changes caused by stock price manipulation; see, respectively Ferdousi and Maeda (2006) and Kim and Sohn (2012). Stock market data were also combined with options data and news data to detect trades that were made based on information that was not available to the general public; see for example Donoho (2004). Multiple techniques were applied to detect such *insider* transactions, including decision trees and ANN. Another related application involves credit card fraud detection. In this case, anomaly detection is applied on credit card transactions to find out suspicious spending patterns. Many techniques were proposed for this challenging task: Ghosh and Reilly (1994), Maes et al. (2002) suggest Neural and Bayesian networks; Zaslavsky and Strizhak (2006), Quah and Sriganesh (2008) employ Self-Organizing-Maps; Sánchez et al. (2009) adopt Association Rules; and Srivastava et al. (2008) propose Hidden Markov models.

3. Data and Methods

Our application is aimed to detect anomalies arising in the Italian TARGET2 payments system. Following the definition in Chandola et al. (2009), anomalies are patterns in data that do not conform to a well defined notion of *usual behavior*. Since our setting is unsupervised, prior labels indicating anomalousness of data points are unavailable and we resort to a type of Artificial Neural Networks called Autoencoder.

We considered the Italian TARGET2 payment activity over 583 working days (spanning over a period of almost 27 months from January 2017 to April 2019). Records were collected every 15 minutes during working days.[1] Overall, $T^{tot} = 24,192$ observations were recorded.

Our empirical application was restricted to the payments among the $N = 20$ largest Italian financial institutions, henceforth *banks*. Symbol X_{ij} is used to denote the flow of payments from bank i to bank j, with $i, j = 1, \ldots, N$, yielding N^2 possible flows. At each time t, the matrix of payments $\mathbf{x}^t$ is observed:

$$\mathbf{x}^t = \begin{bmatrix} 0 & x_{12}^t & \ldots & x_{1N}^t \\ x_{21}^t & 0 & \ldots & x_{2N}^t \\ \vdots & \vdots & \vdots & \vdots \\ x_{N1}^t & x_{N2}^t & \ldots & 0 \end{bmatrix}.$$

Element x_{ij}^t corresponds to the value of X_{ij} at time t, for each pair (i, j) and $t = 1, \ldots, T^{tot}$. In general, matrix x^t is not symmetric, i.e. $x_{ij}^t \neq x_{ji}^t$, and sparse. As a further remark, note how all elements on the diagonal were set to zero, to exclude circular flows from our analysis. As a consequence, the effective number of variables decreases from N^2 to $m = N \cdot (N - 1)$, i.e. from 400 to 380.

Any Artificial Neural Network (ANN) is composed of an input, an output and one or more internal hidden layers. In turn, each layer includes one or more nodes (*neurons*) connected to the nodes of other layers by means of directed edges. Formally, let $\mathbf{w} = [w_1, \ldots, w_m]^T$ be a vector of $m > 0$ *weights*, and $b \in \mathbb{R}$ be a *bias* term. Each node belonging to a given layer processes the values $\mathbf{x} = [x_1, \ldots, x_m]^T$ coming

[1] TARGET2 activity starts at 7:00 a.m. and stops at variable closing times, so that a working day includes between 37 and 44 observations.

from previous layers in two steps:

1. Linear aggregation of its inputs: $y_{agg} = \sum_{i=1}^{m} w_i \cdot x_i + b$,
2. Activation function: $y_{out} = f(y_{agg})$.

Suppose the ANN is composed of H hidden layers, for a given sequence of activation functions $\{f_1, \ldots, f_H\}$, the output $\widehat{\mathbf{x}}$, is determined as follows:

$$\begin{aligned}
\mathbf{x}^{(1)} &= f_1\left(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1\right) \\
\mathbf{x}^{(2)} &= f_2\left(\mathbf{W}_2\mathbf{x}^{(1)} + \mathbf{b}_2\right) \\
&\ldots \\
\widehat{\mathbf{x}}^{(H)} &= f_H\left(\mathbf{W}_H\mathbf{x}^{(H-1)} + \mathbf{b}_H\right)
\end{aligned}$$

$\mathbf{W}_h$, $h = 1, \ldots, H$, is a matrix of weights in $\mathbb{R}^{n_h \times (n_{h-1})}$, mapping elements from a n_{h-1}-dimensional layer toward the n_h nodes of the successive layer. It is easy to see the vector of weights $\mathbf{w}$ mentioned above is just a column of matrix $\mathbf{W}_h$, when $n_{h-1} = m$. Analogously $\mathbf{b}_h \in \mathbb{R}^{n_h}$ is a vector of bias terms. We collectively refer to the weights and biases as *network coefficients*. In our setup, we fix the activation function across all layers, this means $f_h = f$ for any h.

The network architecture suitable for outlier detection with the data described above is called Autoencoder (AE). It is a special type of ANN with a multi-layer symmetric structure, used for unsupervised applications. An AE is trained to reconstruct input data, corresponding to payments matrices in our setup. The reconstruction error obtained is used as *outlier score*, successively passed to some rule to assess occurrence of an anomaly. Formally, any matrix of payments $\mathbf{x}^t$ substantially different from those employed for the training and the validation phase[2] will be marked as anomalous.

[2]We assume that the training set is devoid of anomalies.

AEs were first introduced by Hinton and Salakhutdinov (2006) as a non-linear generalization of Principal Component Analysis (PCA). The AE architecture foresees the same number of neurons in the input layer and the output layer. Furthermore, the (innermost) hidden layer contains a sufficiently small number of neurons to trade-off reconstruction accuracy with training time. Training of an AE yields a minimal representation of input data, in a completely unsupervised framework. An example of a *shallow* (i.e. with a single hidden layer) AE is sketched in Fig. 1.

We now outline the three main steps required to employ an Autoencoder:

i) SCALING: Input data to an AE is scaled to fit into the unit range $[0, 1]$, see for example Bishop (1995). We apply Min-Max scaling to map all observations therein, based on three different scaling criteria:

   1) System level (**Sys**) - Scaling accounts for the overall flows at each time step: input value $x_{i,j}^t$ is mapped in the unit range according to the following rule,

$$\tilde{x}_{i,j}^t = \frac{x_{i,j}^t - \min_{i,j}(x_{i,j}^t)}{\max_{i,j}(x_{i,j}^t) - \min_{i,j}(x_{i,j}^t)} \quad \forall i, j, \forall t$$

   2) Outflow level (**Out**) - Scaling is restricted to each bank's overall outflows:

$$\tilde{x}_{i,j}^t = \frac{x_{i,j}^t - \min_{j}(x_{i,j}^t)}{\max_{j}(x_{i,j}^t) - \min_{j}(x_{i,j}^t)} \quad \forall j \neq i, \forall t$$

   3) Inflow level (**In**) - Scaling is restricted to each bank's overall incoming payments flows:

$$\tilde{x}_{i,j}^t = \frac{x_{i,j}^t - \min_{i}(x_{i,j}^t)}{\max_{i}(x_{i,j}^t) - \min_{i}(x_{i,j}^t)} \quad \forall i \neq j, \forall t$$

   All terms used for Min-Max scaling are stored for the final unscaling phase to generate the estimated values in the original scale.

ii) ENCODE AND DECODE: Scaled data are processed by a deep AE. These networks are composed of an odd number of hidden layers that mediate the flow of information toward the latent dimensions and outwards, in a symmetric manner. We consider an AE with three hidden layers. Let m be the number of neurons in the input and the output layers, k the number of neurons in the first and the third hidden layer and finally l be the number of neurons in second hidden layers (bottleneck layer). We assume $m > k$ and $l = k/2$. The encoding process maps the input in a progressively lower dimensional space data from $\mathbb{R}^m$ to $\mathbb{R}^k$, to $\mathbb{R}^l$, with $m > k > l$. The decoding phase follows the reciprocal path to restore the original dimension.

iii) UNSCALING: Scaling coefficients stored in the first step are employed to recover the output generated by the AE in the original scale.

The outlined procedure is also synthesized by Fig. 2.

We have used three different approaches for the scaling task: Sys-, Out- and In-scaling. In particular, Sys-scaling of the training data is expected to capture deviations in the flows that affect the overall network of payments. However, this approach might end up being too conservative since relatively large deviations in the flows involving small banks would likely be unseen by the anomaly detection task. On the other hand Out- (In-) scaling tackles this situation weighting each bank's outflows (inflows) irrespective of its relative size. Deviations in payments from smaller banks are more likely to be marked as anomalies by the detection task even if these anomalies did not threat the whole payment system.

Estimate of the network coefficients (*training*) and choice of the hyper-parameters (*tuning*) of the AE are based on, respectively, an expanding window of observations and a three-months period of data points. The first set of data is called the training

set whereas the second is the validation set. The training set corresponds to a 1-year window, ranging from April 2017 to March 2018, progressively expanded by one month until February 2019 is reached. Tuning of the trained model is based on its performance on the validation set which goes from January to March 2017. Once the model is fully calibrated, it is employed on the test set [3], which extends over a six weeks period, from the beginning of March to the first half of April 2019. The partitioning of our dataset in three time windows is shown in Fig. 3.

The training process is performed over a maximum of 2,000 epochs[4]. Network coefficients are initialized as follows: Xavier random initialization is used for the weights, while the biases are set to zero [5]. The training process consists in an optimization aimed at determining the set of weights $\mathbf{W}$ and biases $\mathbf{b}$ which minimize the mean reconstruction error (MRE), i.e. the average squared distance between the original data and their estimated output from the AE:

$$MRE_T = \frac{1}{T}\sum_{t=1}^{T} RE(\mathbf{x}^t) = \frac{1}{2T}\sum_{t=1}^{T} \|\mathbf{x}^t - \hat{\mathbf{x}}^t\|_2^2 \,. \tag{1}$$

$\mathbf{x}^t$ and $\hat{\mathbf{x}}^t$ are, respectively, the vectors of all observations and their estimates produced by the AE at time t, where $t = 1, \ldots, T$.

As already mentioned, the choice of the main hyper-parameters of the AE is based

[3] Quite often called hold-out set.

[4] An epoch is one complete presentation of the data set to be learned to a learning machine. The value of 2,000 results from an empirical trade-off between training accuracy and small level of overfitting.

[5] The right weights initialization is relevant to minimize the training time (see Xavier and Bengio (2010)). Xavier initialization consists in assigning the weights from a Gaussian distribution with zero mean and a constant variance equal the inverse of the layer width: $\frac{1}{n_h}$. In this way the layer variance remains $\sigma_h = 1$. This assures a smooth training preventing the error from exploding or vanishing to zero.

on its out-of-sample performance in the validation set. These include the sort of activation function and the cardinality of the innermost hidden layer (l). In our experiments, we have chosen the *identity* activation function which has outperformed both the *hyperbolic tangent* and the *sigmoid* activation functions. We base the choice of the optimal number of neurons in the innermost hidden layers on a trade-off between the reconstruction error performance and overfitting/underfitting, and set $l = 50$, hence $k = 100$. The results for the three different kind of scaling are shown in Fig. 4.

Once the model has been trained and tuned with each one of the three kind of scaling approaches - Sys, Out and In -, anomaly detection is performed on the unscaled predictions. We consider a deviance-based score for detecting anomalies. Formally, let MRE^{val} and $\sigma_{RE^{\text{val}}}$ be the mean reconstruction error and its standard deviation in the validation set, our rule marks as anomalous any observation $\mathbf{x}^t$ for which:

$$RE(\mathbf{x}^t) \geq MRE^{\text{val}} + \alpha\sigma_{RE^{\text{val}}} \tag{2}$$

for each t in the test set, and $\alpha > 0$.

We assume that systemic failures, as well as one-time disruptions, likely occur as fuzzy sequences of perturbations, and there is not a unique rule to identify and characterize them. Additionally, single perturbations manifest in time with varying magnitudes, some of which negligible. For the anomaly detection task, no accuracy measure may be consistently applied to evaluate the performance of our models without formulating specific assumptions. Here we adopt a threshold-based rule, Eq. (2), to label anomalous data points, irrespective of the extent to which these exceed the threshold value α. To account for such extent, in sec. 4 we shall consider different values of the threshold value, to gain the impression of how sensitive a model

is to deviations in data. However, on the basis of our experience we suggest to use $\alpha \leq 3$, for good practice.

Finally, to further test the ability of our AEs to detect anomalies in a supervised framework, we introduce suitably crafted perturbations in our dataset.

For a fixed number of time steps d, we augment the outflows of bank i based on a stochastic procedure. Let d be the number of time steps composing the *perturbation window*, starting at t_0. At any $t' \in [t_0, t_0 + d]$ anomalies occur according to some scheme. We introduce anomalies according to the following:

$$\breve{x}_{ij}^{t'} = x_{ij}^{t'} + \delta_{ij}^{t'} a_{ij}^{t'}\,, \tag{3}$$

$\delta_{ij}^{t'} \in \{0, 1\}$ and $a_{ij}^{t'} \in \mathbb{R}$ being, respectively, a binary term denoting occurrence of the anomaly and its size.

We account for three main scenarios, namely with extreme, size increasing and memoryless one-shot anomalies.

Due to the haziness in the classification of anomalies, in a complex environment like TARGET2, in the first scenario these are introduced as deliberately *extreme*, to pin down a wide acceptable measure of performance to enable statistically sound comparisons across methods and datasets. This is relevant, also in light of the growing economic impact of classification technology (ranging from fraud/anomaly detection to medical diagnosis). In the field of classification, performances are usually expressed as weighted combinations of False Positive and False Negative Rate, so as to mirror the inherent conflict between two antagonistic objectives: minimizing the number of False Positives (e.g., false anomalies) as well as that of False Negatives (e.g., missed true anomalies). In our empirical applications we have chosen the very

popular measure of precision and recall [6], which enjoy a wide acceptance among scholars in different fields.

Roughly, we split the test set in two parts, with the fraction of anomalies ranging from 40% to 60%. Anomalous input data are derived from matrix $\mathbf{x}^t$, transformed from sparse to dense: its zero values are replaced by random realizations of a Gaussian variable with mean 10^7 [7] and standard deviation 10^5, while the others are reduced to one twentieth of their original amount.

In the second scenario, anomalies are introduced following a perturbation scheme analogous to the one proposed by Triepels et al. (2017). We consider a fixed number of time periods ($d = 209$), corresponding to around a full working week. $\delta_{ij}^{t'}$ from Eq. (3) is the realization of a Bernoulli variable, indexed by time-varying parameter $\pi(t')$. This latter is defined as follows:

$$\pi(t') = \underline{\pi} + (\overline{\pi} - \underline{\pi}) \left(\frac{t'}{d}\right)^2 . \tag{4}$$

It holds $\underline{\pi} \leq \pi(t') \leq \overline{\pi}$, with $\underline{\pi} = 0.1$, $\overline{\pi} = 0.8$.

Size of the perturbations is determined in turn as a realization of an exponential process whose rate $\lambda(t')$ follows the same dynamics as that in eq. (4):

$$\lambda(t') = \underline{\lambda} + (\overline{\lambda} - \underline{\lambda}) \left(\frac{t'}{d}\right)^2 .$$

$\underline{\lambda} \leq \lambda(t') \leq \overline{\lambda}$.

We fix $\underline{\lambda} = 10^4$, $\overline{\lambda} = 10^7$ and $\overline{\lambda} = 5 \cdot 10^7$, increasingly yielding, respectively, mild and strong perturbations.

One-shot anomalies are only introduced on the last day of the same working week

[6] defining tp = true positive, fp = false positive and fn = false negative, precision = $\frac{tp}{tp+fp}$ while recall = $\frac{tp}{tp+fn}$

[7] This is approximately the median of the values in the perturbation window different from zero

according to a Bernoulli process with constant probability $\pi = 0.5$, and $a_{ij}^{t'} = 9 \cdot x_{ij}^{t'}$, for each t' in the window. The two perturbation processes based on stochastic generation of anomalies are depicted in Fig. 5, while accuracy of the AE in the anomaly detection task is discussed throughout sec. 4.1.

4. Results

The training of our AEs is based on the three different scaling techniques earlier described. AE performances are shown by graphs showing the relation with a increasing size of the training set by a color-coded scale. The number of daily detected anomalies is length-coded. Each day presents a number of anomalies for each one of the values taken into account. Higher and denser segments mean a larger number of anomalies. Fig. 6 depicts the anomaly detection performance of the AE trained on a progressively expanding window of sys-scaled data - from 12 to 23 months, with monthly incremental steps. There we use the values $\alpha = 1, 2, 3$. Analogous results for Out- and In-Scaled data are reported by Fig. 7 and 8, respectively.

Overall, performance of the models proved robust to different sizes of the expanding window used for training, see Sec. 3. Nonetheless, shorter windows, i.e. fewer data points, of observations tend to slightly over-report anomalies when $\alpha \leq 1$. Robustness to changes in the number of records in the Training Set was supported by preliminary analysis in the frequency domain, where high-frequency components of data was assessed. The intuition is the following: any high-frequency periodic phenomenon produces several representations of the period's phases. In such a framework, training of the AE is based on repetitions of a limited range of patterns, making the marginal contribution of additional data points less relevant.

A further remark concerns sensitivity of the anomaly detection task to changes in

parameter α, whose value ought to reflect the attitude toward deviations from the expected patterns as anomalies: the larger α, the higher is confidence about stability of TARGET2's payment flows. Detection of anomalous behavior at the system level is almost invariant to changes of α, when $\alpha > 1$. See also fig. 9 for clarity.

While our application is aimed to timely detect dynamics that could materialize systemic risk, finer grained information may also be exploited, to assess the state of single banks. Let us restrict our analysis to the first two weeks of April 2019. Performance of the daily anomaly detection performed by AEs trained on different scalings (Out and In) is depicted in Fig. 10, with respect to a finer grid of values of parameter α. It clearly emerges that extending the training period causes a sharp drop in the number of anomalies, especially, for higher values of threshold parameter α. This reflects the ability of the autoenconder to rightly address some spikes to the usual behavior, even if they occur on TARGET2 due to, e.g., the contemporaneous occurrence of settlements of some less frequent monetary policy operations, and other large interbank payments. The output produced by the AEs allows to delve into the nature of deviations from *usual behavior* of payments data. As an example, Fig. 11 represents contributions to the general RE by each bank's cumulative outflows and inflows. Focusing on larger values of α, we found that most singularities appear as related to payments settled by banks on behalf of their customers. Those constitute a frequently executed kind of payments via TARGET2, for pairs of banks which exchange customer payments quite rarely. These findings suggest a potential usage of the autoencoder for wholesale payment fraud detection, as it proved to be able to recognize "unusual or uncharacteristic payment patterns (e.g., in terms of timing, value, volume or location)" as required by the Committee on Payments and Market Infrastructures (2018). Besides these customer payments, the autoencoder succeeded in detecting singularities caused by a bank suffering a major outage which

prevented it from submitting payments to TARGET2 for several hours in a day.
In real-time, granular information on single payments flows allows straightforward assessment of those acting as drivers for the impending systemic failure, if any. Retrospectively, it may help the qualitative as well as quantitative reconstruction of the chain of events that led to liquidity crises.

4.1. Supervised Anomaly Detection

Table 1 summarizes the accuracy performance of our AE with extreme anomalies, according to different values of threshold parameter α. The scenarios considered are designed so as to guarantee a balance between anomalies and original values. In Table 1 we show the AE performances in terms three standard figures for classification tasks: precision, recall and F1-score. The latter is a summarizing measure of the other two, obtained as their harmonic mean.
In the case of Sys-scaling we record values of precision and recall systematically higher than 99.0%. When we consider the other two kind of scaling, Out and In, we achieve somewhat lesser values for the classification performances. Nonetheless they stay always above 92.0%.
The least value obtained for F1-score is 94.8%. However, the values from Table 1 provide us with evidence of a satisfactory classification mechanism. As a remark, this exercise does not account for the ordering of observations: it is aimed to evaluate its accuracy in properly discriminating between anomalies and expected labelled data points, out of haziness. Nonetheless, the fraction of original observations may themselves contain slightly anomalous records. In fact, further analysis on the anomalous component only provided increased values of the accuracy measures.

Fig. 12 summarizes the performance of the AE on the whole perturbed week, according to different threshold values of α (see Eq. 2).

As perturbations were restricted to the outflows (of a single bank), the AE trained with In-Scaled data revealed very sensitive to deviations from the expected behavior. Additionally, the daily number of unexpected values in the whole system detected by the Autoencoder grew from 26.19% on Monday, to 78.57% (+230.00% compared to original data) on Thursday, to 97.62% (+241.67%) on Friday, with fixed $\alpha = 2$ and strong perturbations. Analogously, mild perturbations yielded +50% and +191.67% anomalous data points in the last two days of the week.

The case of abrupt perturbations concentrated all anomalies on a single day of the week, without any forewarning in the data nor preparing dynamics. The perturbed outflows were proportional to the original payments flows, and yielded reporting of 14.29% (+20.08%) and 40.48% (+41.69%) of critical data points from the AEs trained on Sys- and In-Scaled data, respectively, with $\alpha = 2$.

It is now worth stressing two critical points. As a first, while the case of extreme anomalies would serve the purpose to make the classification task fully supervised, albeit the inherent fuzziness in the original component of the sample, the perturbations introduced in the other two scenarios did not consider the spillover effects originating from deviations in the payments flows. Although this may be judged overly simplistic, it provided us with insights on the way models trained on data scaled differently behave in specific situations, that are rather likely to occur in payments data. Overall, our models proved efficient in timely detecting gradual introduction of anomalies even though the perturbation scenarios described in Sec.4 would only affect 5% of all features, i.e. 19 out of 380 components of a payment flow.

By considering the shown results, we can infer that the AEs represent a robust method for a reliable assessment for anomalies detection on TARGET2.

5. Concluding Remarks

With the help of a deep neural network architecture, in this work we have extended a method to detect both idiosyncratic and system-level anomalies in a RTGS system. The procedure is based on the training of an Autoencoder to reconstruct a set of known liquidity vectors. The outcome of our empirical application provides the following results:

1. A deep Autoencoder with three hidden layers detects anomalies by comparing the reconstruction error in the test set with the Mean Recostruction Error in the training set;
2. Besides the extension of the package available for the R software, the Autoencoder has also been ported in Python achieving a reduction in the training time of one order of magnitude;
3. The preliminary results seem to show good timely properties for system wide stability conditions for the payment system.

Furthermore, we have shown that the reconstruction error made by a well-trained Autoencoder after the reduction and reconstruction phase for the liquidity vectors shows very clearly anomalous changes in the payment flows among banks.

We plan to further improve our work by extending the sample of analysis to years which have witnessed more instabilities in terms of liquidity, and trying to evaluate recurrent neural networks which also embed time dependencies.

References

Allen, F. and Gale, D. Financial Contagion. *Journal of Political Economy*, 1:1–33, 2000.

Arciero, L. and Impenna, C. L'andamento infragiornaliero dei pagamenti nel sistema di regolamento lordo (BI-REL). *Banca Impresa Società*, 3:429–450, 2007.

Arciero, L., Heijmans, R., Heuver, R., Massarenti, M., Picillo, C., and Vacirca, F. How to measure the unsecured money market: The Eurosystem's implementation and validation using TARGET2 data. *International Journal of Central Banking*, 12(1):247–280, March 2016.

Ball, A., Denbee, E., Manning, M., and Wetherilt, A. Intraday liquidity: risk and regulation. Bank of England Financial Stability Paper 11, Bank of England, 2013.

Banca d'Italia. Simulation of the effects on TARGET2-Banca d'Italia of a shock in the interbank market. Financial Stability Report 1, Banca d'Italia, 2010.

Banca d'Italia. The intraday liquidity risk of banks connected to TARGET2-Banca d'Italia. Financial Stability Report 4, Banca d'Italia, 2012.

Basel Committee on Banking Supervision. Monitoring tools for intraday liquidity management. BCBS Report 2048, Bank for International Settlement, Basel, CH, 2013.

Berndsen, R., León, C., and Renneboog, L. Financial stability in networks of financial institutions and market infrastructures. *Journal of Financial Stability*, 35:120–135, 2016.

Bishop, C. M. *Neural Networks for Pattern Recognition*. Oxford university press, 1995.

Chandola, V., Banerjee, A., and Kumar, V. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

Committee on Payments and Market Infrastructures. Reducing the risk of wholesale payments fraud related to endpoint security. CPMI Report 178, Bank for International Settlement, Basel, CH, 2018.

Donoho, S. Early detection of insider trading in option markets. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429. ACM, 2004.

European Central Bank. Stress-testing of liquidity risk in TARGET2. European Central Bank, Occasional Paper 183, European Central Bank, 2017.

Ferdousi, Z. and Maeda, A. Unsupervised Outlier Detection in Time Series Data. In *Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on*, pages x121–x121. IEEE, 2006.

Gerlach, S. Defining and measuring systemic risk. *European Parliament papers*, pages 1–9, 2009.

Ghosh, S. and Reilly, D. L. Credit card fraud detection with a neural-network. In *System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on*, volume 3, pages 621–630. IEEE, 1994.

Heijmans, R., Heuver, R., Levallois, C., and van Lelyveld, I. Dynamic Visualization of Large Transaction Networks: The Daily Dutch Overnight Money Market. *DNB WP*, March 2014. URL `https://ideas.repec.org/p/dnb/dnbwpp/418.html`.

Hinton, G. E. and Salakhutdinov, R. R. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507, 2006. doi: 10.1126/science.1127647.

Huang, H. and Xu, C. Financial Institutions, Financial Contagion and Financial Crises. *IMF Working Paper*, 92:3–32, 2000.

Kim, Y. and Sohn, S. Y. Stock fraud detection using peer group analysis. *Expert Systems with Applications*, 39(10):8986–8992, 2012.

León, C. and Pérez, J. Assessing financial market infrastructures systemic importance with authority and hub centrality. *Journal of Financial Market Infrastructures*, 2:67–87, 2014.

Maes, S., Tuyls, K., Vanschoenwinkel, B., and Manderick, B. Credit card fraud detection using bayesian and neural networks. In *Proceedings of the 1st international naiso congress on neuro fuzzy technologies*, pages 261–270, 2002.

Massarenti, M., Petriconi, S., and Lindner, J. Intraday Patterns and Timing of TARGET2 Interbank Payments. *Journal of Financial Market Infrastructure*, 2: 3–24, 2012.

McAndrews, J. and Rajan, S. The Timing and Funding of Fedwire Funds Transfers. *Economic Policy Review*, 6:17–32, 2000.

Quah, J. T. S. and Sriganesh, M. Real-time credit card fraud detection using computational intelligence. *Expert Systems with Applications*, 35(4):1721–1732, 2008.

Sánchez, D., Vila, M. A., Cerda, L., and Serrano, J. M. Association rules applied to credit card fraud detection. *Expert Systems with Applications*, 36(2):3630–3640, 2009.

Srivastava, A., Kundu, A., Sural, S., and Majumdar, A. Credit card fraud detection using hidden markov model. *IEEE Transactions on Dependable and Secure Computing*, 5(1):37–48, 2008.

Triepels, R., Daniels, H., and Heijmans, R. Anomaly detection in Real-Time Gross Settlement Systems. In *ICEIS (1)*, pages 433–441, 2017.

Xavier, G. and Bengio, Y. Understanding the Difficulty of Training deep feedforward neural networks. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 9:249–256, 2010.

Zaslavsky, V. and Strizhak, A. Credit Card Fraud Detection Using Self-Organizing Maps. *Information and Security*, 18:48, 2006.

APPENDIX: Tables and Figures

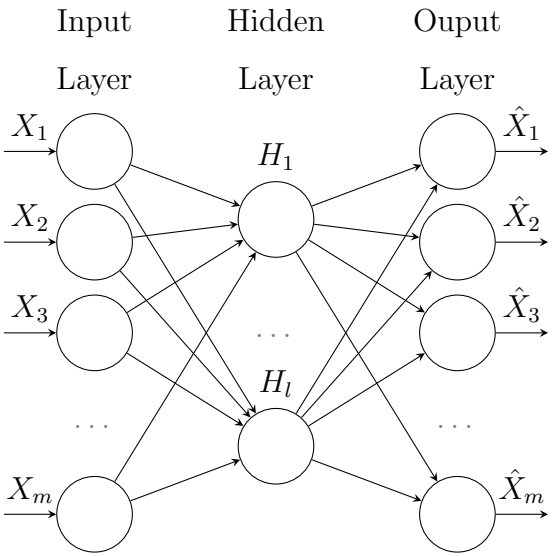

Figure 1: Basic representation of a shallow Autoencoder. Input and Output layers are constituted by $n > 0$ neurons, or nodes. Input layer is fully connected with the l nodes from Hidden layer, with $l \leq n$; these are in turn fully connected with Output layer.

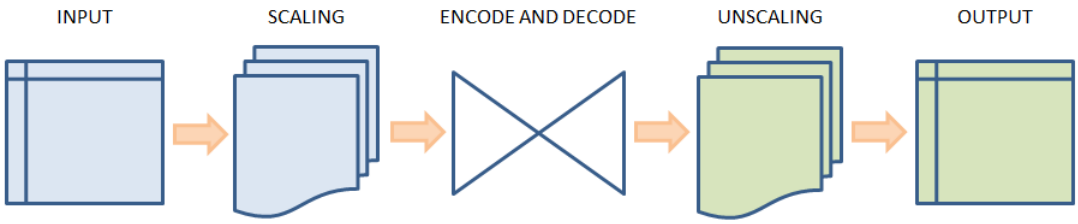

Figure 2: Autoencoder processing steps.

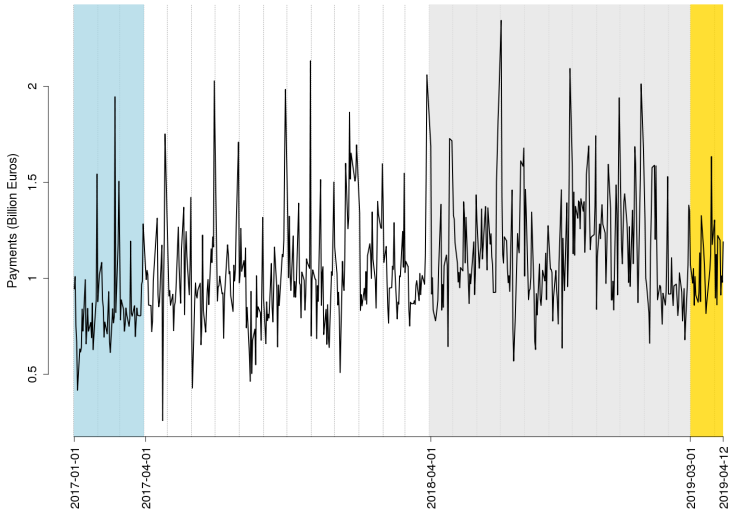

Figure 3: Gross amount of overall payments recorded between the 20 largest Italian banks from January 2017 until April 2019. Data were partitioned into Validation Set (light blue area), an expanding Training Set (white and grey area) and a Test Set (yellow area). Gray vertical lines identify months.

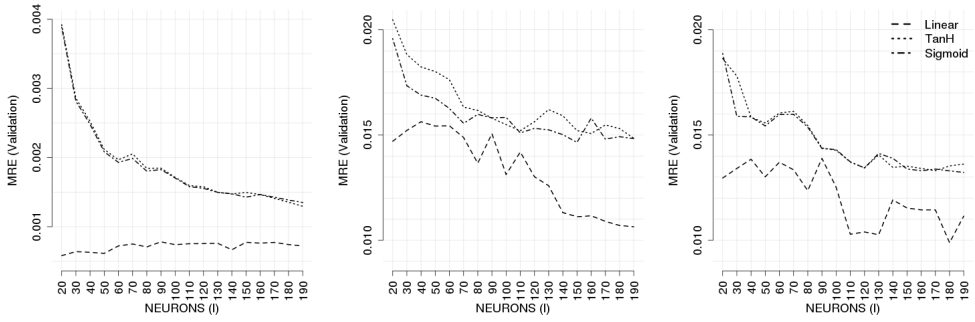

Figure 4: MRE performance on the Validation set of Autoencoders, according to varying number of neurons in the innermost layer (l) and activation function. Left-to-right: Errors are reported for the network trained with Sys-, Out- and In-scaled data.

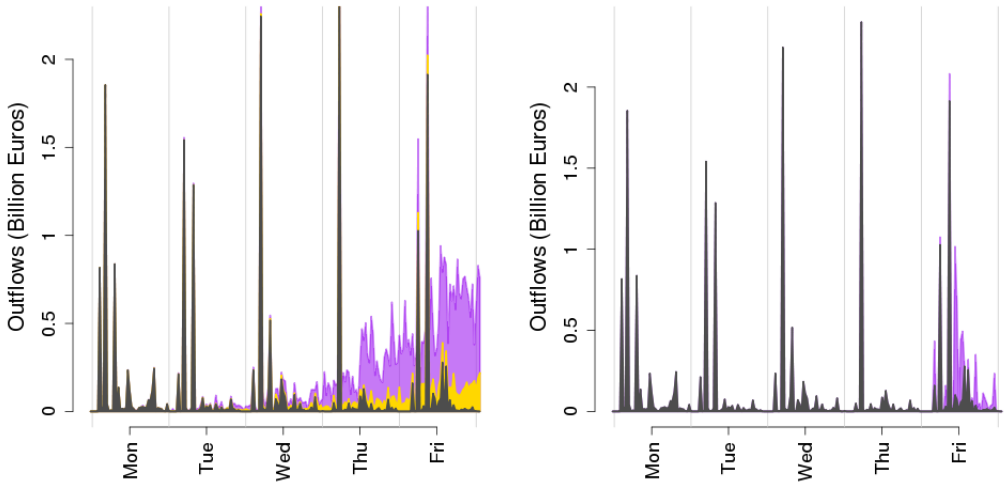

Figure 5: Left: Example of increasingly perturbed payments with respect to the outflows of bank i in the Test set. Yellow and purple areas correspond to, respectively, mild and strong perturbations of original data (black area), resulting from $\overline{\lambda} = 10^7$ and $\overline{\lambda} = 5 \cdot 10^7$, respectively. Right: Example of abruptly perturbed payments with respect to the outflows of bank i in the Test set. The purple and black area identify perturbed and original data.

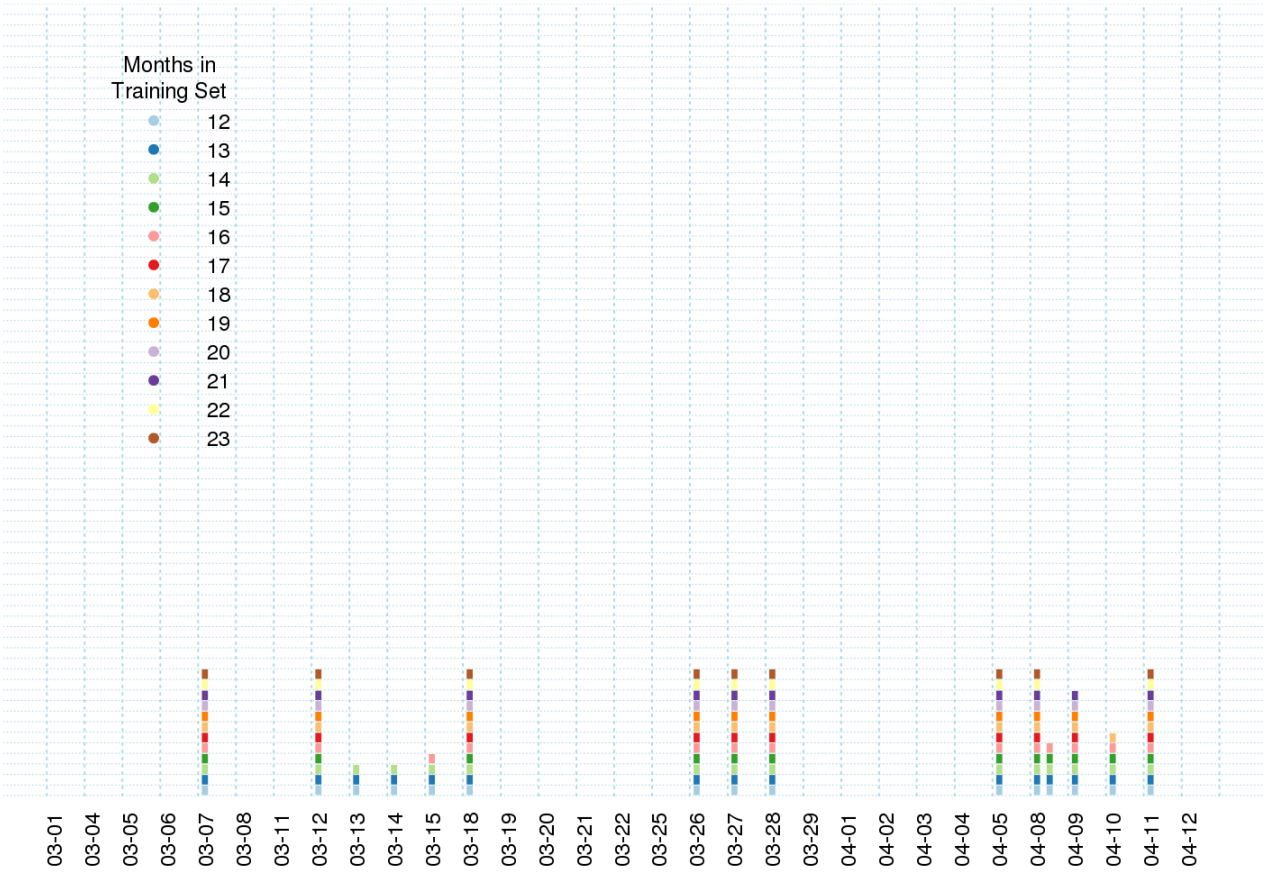

Figure 6: Daily anomalies detected in the Test Set by the Autoencoder trained on Sys-Scaled data. Each colored vertical segment depicts the number of daily detected anomalies. Length of the training period is color-coded. Each day can have up to three vertical segments referring to different values of threshold parameter $\{1.0, 2.0, 3.0\}$.

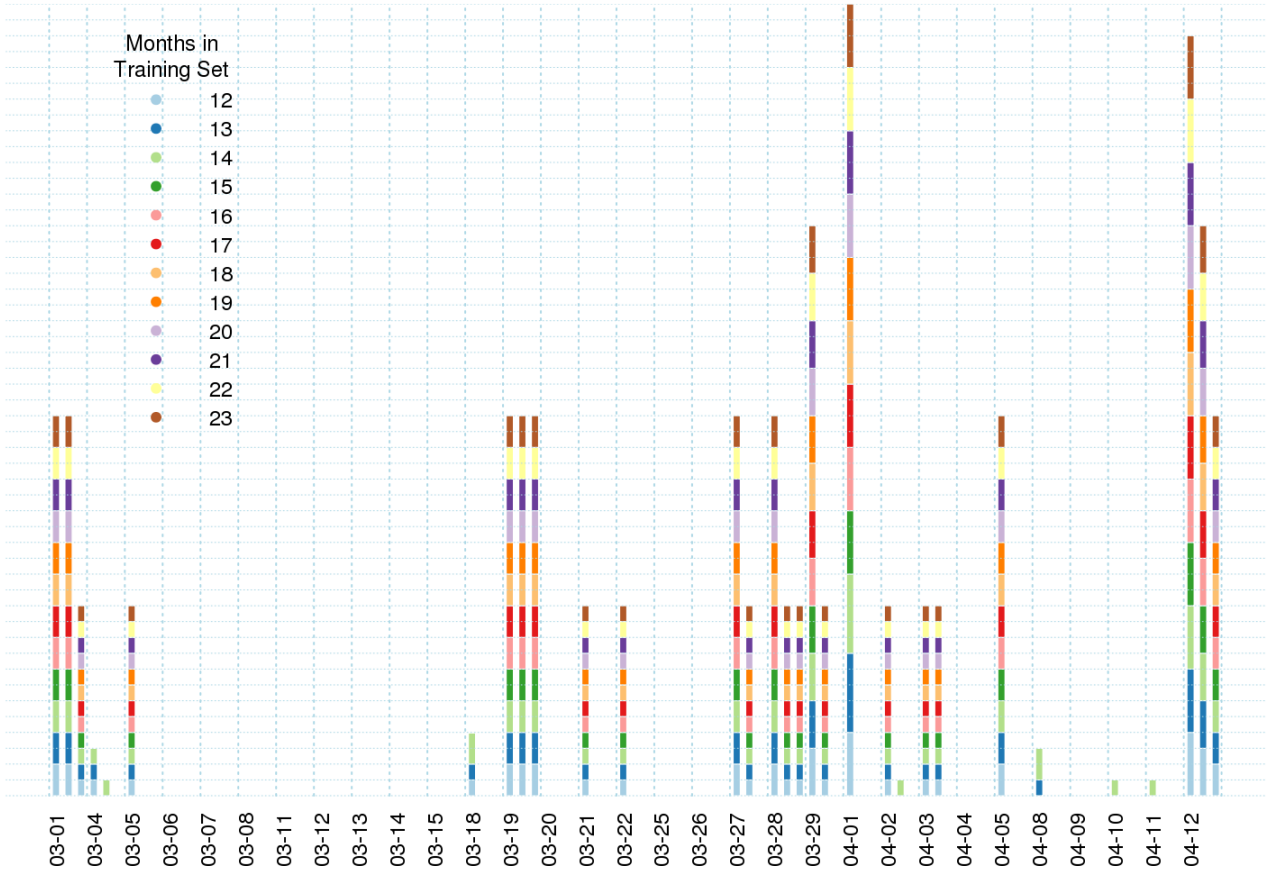

Figure 7: Daily anomalies detected in the Test Set by the Autoencoder trained on Out-Scaled data. Each colored vertical segment depicts the number of daily detected anomalies. Length of the training period is color-coded. Each day can have up to three vertical segments referring to different values of threshold parameter $\{1.0, 2.0, 3.0\}$.

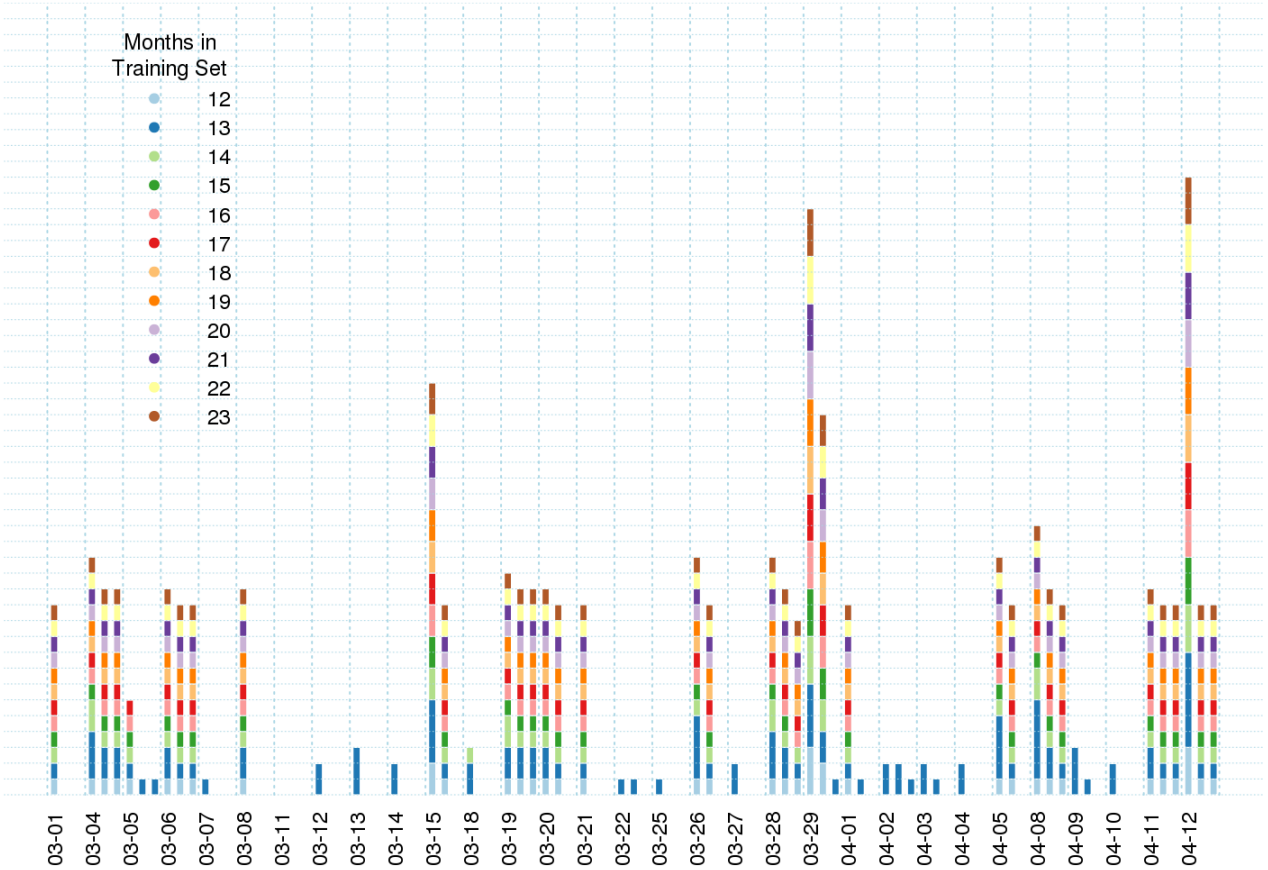

Figure 8: Daily anomalies detected in the Test Set by the Autoencoder trained on In-Scaled data. Each colored vertical segment depicts the number of daily detected anomalies. Length of the training period is color-coded. Each day can have up to three vertical segments referring to different values of threshold parameter $\{1.0, 2.0, 3.0\}$.

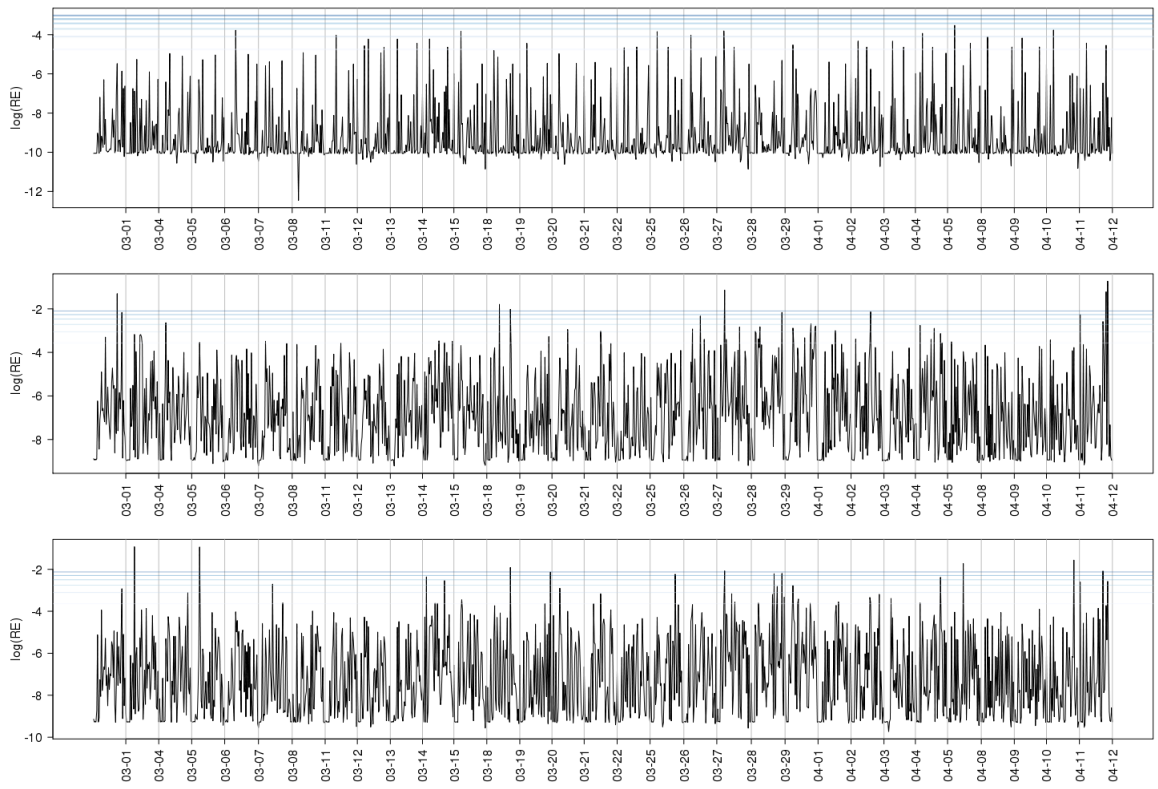

Figure 9: Reconstruction Errors produced in the Test Set by the Autoencoder trained on (top-down) Sys-, Out- and In-Scaled data, on the fully expanded Training window. Blue lines represent the threshold value with $\alpha \in [0, 3]$ (see Eq. (2)). Values are represented on a log scale, to better display changes in the RE.

	Extreme anomalies (%)	α	Sys	Out	In
Precision	50%	2	99.6%	94.5%	96.5%
		3	99.8%	95.6%	97.4%
	60%	2	99.7%	96.1%	97.7%
		3	99.9%	96.9%	98.3%
	40%	2	99.3%	92.4%	95.0%
		3	99.8%	93.9%	96.4%
Recall	50%	2	99.7%	97.2%	98.2%
		3	99.7%	97.0%	98.0%
	60%	2	99.8%	97.3%	98.2%
		3	99.8%	97.0%	98.1%
	40%	2	99.7%	97.3%	98.2%
		3	99.7%	97.0%	98.0%
F1-score	50%	2	99.7%	95.8%	97.3%
		3	99.8%	96.3%	97.7%
	60%	2	99.7%	96.7%	98.0%
		3	99.8%	97.0%	98.2%
	40%	2	99.5%	94.8%	96.6%
		3	99.7%	95.5%	97.2%

Table 1: Classification performances of the Autoencoder for the three scaling techniques.

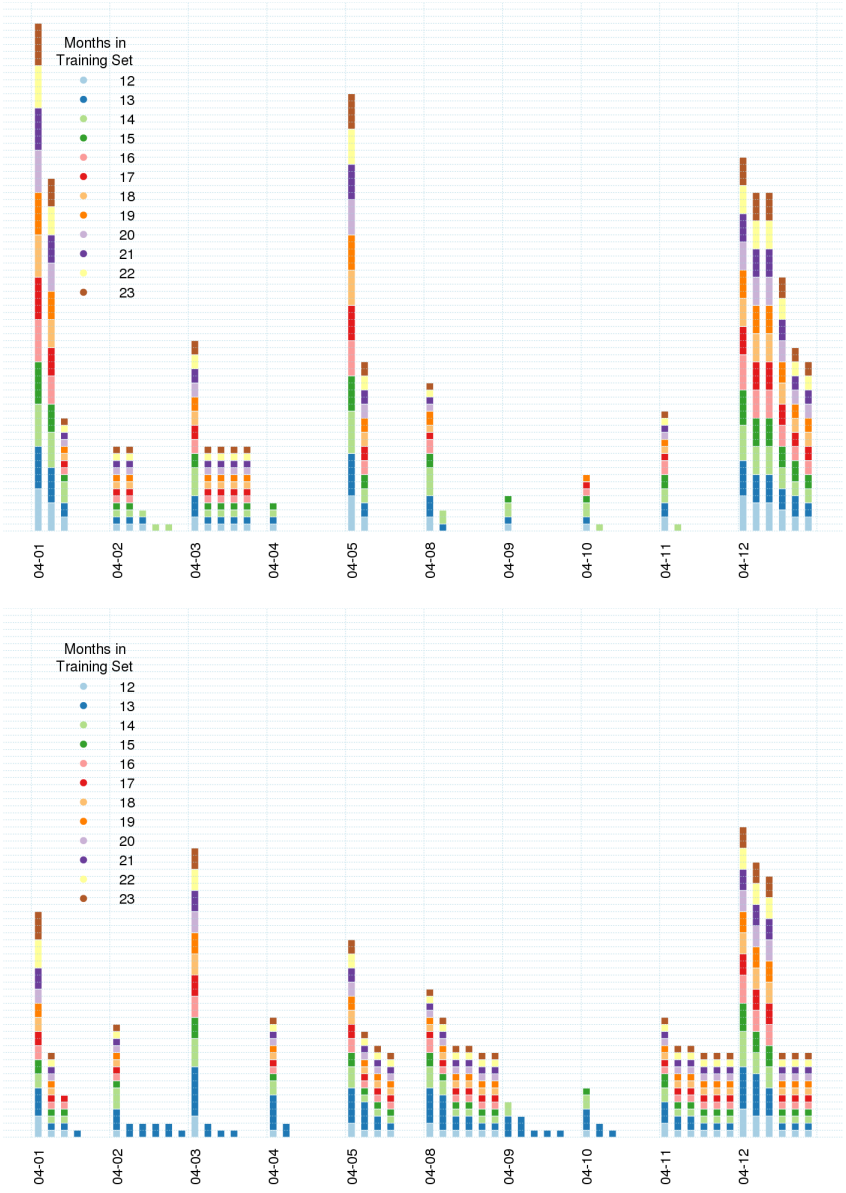

Figure 10: Daily anomalies detected in the Test Set by the Autoencoder trained on Out- (top) and In- (bottom) scaled data in the first two weeks of April 2019. Each colored vertical segment depicts the number of daily detected anomalies. Length of the training period is color-coded. Each day can have up to six vertical segments referring to different values of threshold parameter $\{0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$.

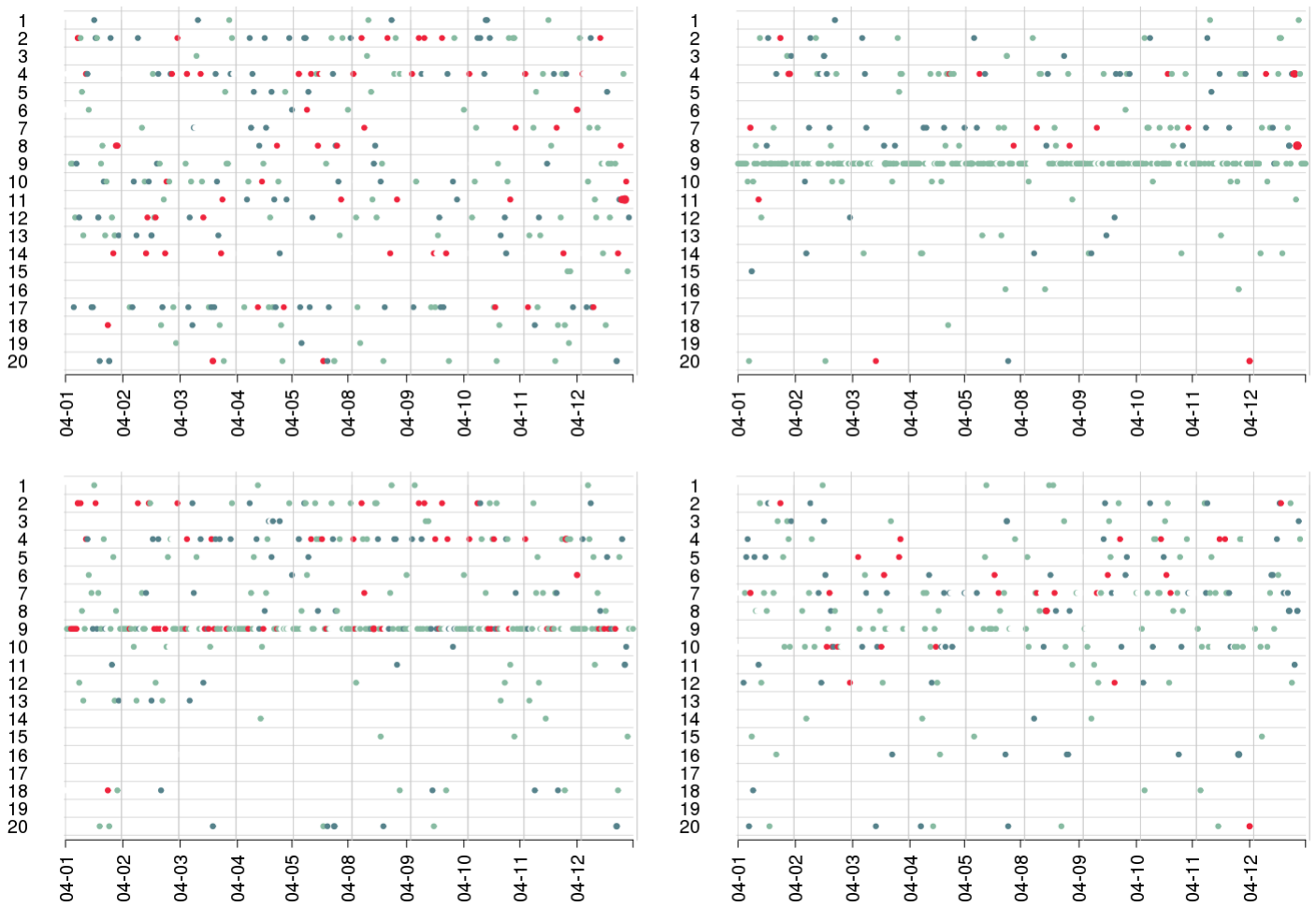

Figure 11: Bank realizing the largest contribution to the RE on each time step of the first two working weeks of April 2019 as obtained by the Autoencoder trained with Out- (top panels) and In- (bottom panels) scaled data on the fully expanded window. Left panels account for contributions to the vector of each bank's cumulative outflows; right panels consider inflows. Suppose the RE of bank i's outflow is the largest element in $\mathbf{RE}^t_{\text{out}} = [RE(1)^t_{\text{out}}, \ldots, RE(i)^t_{\text{out}}, \ldots, RE(N)^t_{\text{out}}]^T$, where $RE(i)^t_{\text{out}} = \sum_j (x^t_{i,j} - \hat{x}^t_{i,j})^2$; dot is colored light green when proportion $RE(i)^t_{\text{out}}/\|\mathbf{RE}^t_{\text{out}}\| \in (0.25, 0.50]$, it is dark green if it is in $(0.50, 0.75]$, red if it is strictly larger than 0.75, white otherwise. Labels were shuffled due to the confidential nature of data.

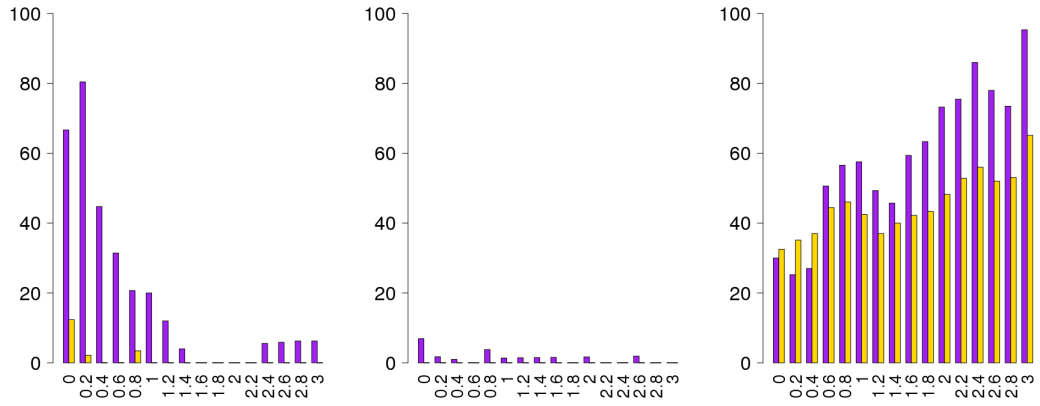

Figure 12: Left to right: Additional anomalies (%) reported by the model trained on Sys-, Out- and In-scaled data, with increasing anomalies over a week period. The increasing size of anomalies was either mild (yellow) or strong (purple).