Pattern recognition of financial institutions' payment behavior[+]

Carlos León[a,d,e], Paolo Barucca[b], Oscar Acero[a], Gerardo Gage[c], Fabio Ortega[a]

Abstract

We present a general supervised methodology to represent the payment behavior of financial institutions starting from a database of transactions in the Colombian large-value payment system. The methodology learns a feedforward artificial neural network parameterization to represent the payment patterns through 113 features corresponding to financial institutions' contribution to payments, funding habits, payments timing, payments concentration, centrality in the payments network, and systemic impact due to failure to pay. The representation is then used to test the coherence of out-of-sample payment patterns of the same institution to its characteristic patterns. The performance is remarkable, with an out-of-sample classification error around three percent. The performance is robust to reductions in the number of features by unsupervised feature selection. Also, we test that network centrality and systemic impact features contribute to enhancing the performance of the methodology definitively.

JEL Classification: C45, E42, G21

Keywords: payments, neural networks, feature selection, machine learning

---

[+] Opinions and statements in this article are the sole responsibility of the authors and represent neither those of Banco de la República nor of its Board of Directors nor of the institutions the authors are affiliated to. We thank Serafín Martínez, Raúl Morales, and Deisy Zambrano for their contribution to the development of the methodology. We thank XXXX for their comments and suggestions. Any remaining errors are our own.

[a] Banco de la República, Colombia. (O. Acero is currently at Stratio, Colombia).
[b] University College London, United Kingdom.
[c] Centro de Estudios Monetarios Latinoamericanos (CEMLA), México.
[d] Tilburg University, The Netherlands.
[e] Corresponding author, e-mail: cleonrin@banrep.gov.co / carlosleonr@hotmail.com.

Introduction

The amount of digital transactions occurring in financial systems is constantly growing. Identifying usual and unusual patterns of behaviour is crucial to monitor the safe and efficient functioning of payment systems. Machine learning methodologies have demonstrably been able to learn complex patterns and provide accurate data representations. In the context of payment systems, this is leading to the development of automated monitoring tools that enhance financial authorities' supervision and oversight methods.

Starting from a large-value payments systems dataset, Triepels, et al. (2017) and Sabetti and Heijmans (2020) have shown that accurate representations can be learnt to identify payment networks that deviate from their norm. This is accomplished by an entirely data-driven unsupervised neural network methodology, an autoencoder architecture, that can consistently recognise a typical, yet potentially complex, pattern of financial transactions. With a simpler approach, León (2020) accomplishes a similar representation based on principal component analysis and clustering methods. However, to the best of our knowledge, no representation of individual financial institutions' behaviour in payment systems has been reported.

In such representations, it is now considered crucial to integrate systemic features about the whole financial network. This concurs with the systemic approach to the monitoring of financial systems that emerged since the Global Financial Crisis, when evidence showed the importance of risk propagation channels that amplify losses in unforeseen manners (see Haldane & May, 2011). This systemic approach has emphasised how financial relationships create networks that link institutions that may not be considering their mutual interdependence and risk exposures. It is opposed to a purely institution-centric approach, where institutions are evaluated individually, neglecting network effects.

The recent data-driven machine learning approach applied to payment systems can give a new perspective to the supervision and oversight of financial systems. Not only machine learning methodologies can be trained to represent financial institutions' payment behavior but can also take into account financial institutions' interrelations in payments networks as determinants of that behavior. For financial authorities, this is the first step towards an automated and systemic detection of individual financial institutions' anomalous behavior in payment systems.

We use a feedforward artificial neural network to represent the payment behavior of individual financial institutions starting from a dataset of transactions in the Colombian large-value payment system. We employ characteristic payment patterns through features corresponding to financial institutions' contribution to payments, funding habits, payments timing, payments concentration, centrality in the payments network, and systemic impact due to failure to pay. We use an individual

representation of institutions to test the coherence of out-of-sample payment patterns of the same institution to its characteristic patterns. That is, we address financial institutions' payment behavior problem as a supervised classification problem. We test whether classification performance is robust to reductions in the number of features by unsupervised feature selection. Also, we measure to what extent the systemic approach, in the form of network centrality and systemic impact features, contributes to enhancing the performance of the classification methodology.

1. Literature review

Our choice of an artificial neural network for this classification problem is based on three main strengths highlighted in related literature. First, given enough hidden layers and enough training samples, artificial neural networks can closely approximate any function, thus they can deal with non-linear relationships between factors in the data (see Bishop, 1995; Han & Kamber, 2006; Fioramanti, 2008; Demyanyk & Hasan, 2009; Eletter, et al. 2010; Sarlin, 2014; Hagan, et al. 2014). Second, artificial neural networks make no assumptions about the statistical distribution or properties of the data (see Zhang, et al., 1999; McNelis, 2005; Demyanyk & Hasan, 2009; Nazari & Alidadi, 2013; Sarlin, 2014). Third, artificial neural networks have proven to be very effective classifiers, even better than the state-of-the-art models based on classical statistical methods (see Wu, 1997; Zhang, et al., 1999; McNelis, 2005; Han & Kamber, 2006).

Artificial neural networks have been applied to classification and anomaly detection problems in the financial domain. They have been used in credit card fraud detection (see Aleskerov, et al., 1997; Ghosh & Reilly, 1994; Dorronsoro, et al., 1997), which presents the challenge of providing a prompt classification to be able to act and stop the on-going fraudulent activity. They have been used in anti-money laundering (see Brause, et al., 1999), where the absence of labeled data about account holders and transactions is a significant challenge. In auditing, financial ratios and artificial neural networks have been used together to identify potential tax-evasion cases (see Wu, 1997). In credit risk, artificial neural networks have been used to improve loan decisions by classifying potential borrowers' credit quality (see Angelini, et al., 2008; Eletter, et al., 2010; Nazari & Alidadi, 2013; Bekhet & Eletter, 2014) and to classify firms according to their likelihood of bankruptcy or failure (see Tam & Kiang, 1990; Tam, 1991; Salchenberger, et al., 1992; Wilson & Sharda, 1994; Olmeda & Fernández, 1997; Zhang, et al., 1999; Atiya, 2001; Brédart, 2014). Fioramanti (2008), Sarlin (2014), and Holopainen and Sarlin (2016) use artificial neural networks in macro early-warning systems. Also, artificial neural networks have been applied at an institutional level to classify banks as domestic or foreign (see Turkan, et al., 2011) and Islamic or conventional (see Khediri, et al., 2015), and to classify banks' balance sheets into their corresponding bank (see León, et al., 2017).

In recent work related to the oversight of payments systems, Triepels, et al. (2017) and Sabetti and Heijmans (2020) developed methodologies based on autoencoders to detect anomalous payments networks. Triepels, et al. (2017) worked on large-value payments networks corresponding to the Dutch partition of the Eurosystem payments network (TARGET2). Sabetti and Heijmans (2020) worked on data from the Automated Clearing Settlement System (ACSS), a Canadian retail payment system. From a simpler approach, León (2020) developed a methodology to detect anomalous payment networks in the Colombian large-value payments system (CUD) based on a mixture of principal component analysis and clustering methods. These three works share a common aim, namely to detect anomalous payments networks; they do not work on the individual behavior of financial institutions participating in the payment systems but on the overall structure of the payments networks. Further, they also share an unsupervised approach and the construction of synthetic anomalous payments to test out-of-sample performance.

We aim at the individual behavior of financial institutions. Our approach is a supervised methodology to represent the payment behaviour of financial institutions in the Colombian large-value payment system. In this vein, our work is related to León, et al. (2017), who used a supervised methodology to represent banks' characteristic financial structure through features corresponding to their balance sheets; it is a pattern recognition problem based on banks' balance sheet data. León, et al. (2017) showed that an artificial neural network can accurately classify out-of-sample banks by learning the main features of their balance sheets. Similarly, our work is a pattern recognition problem based on financial institutions' individual and systemic behavior in the large-value payment system.

2. Methods

The base case model is an artificial neural network pattern recognition method on a set of 113 features that capture the behavior of 26 banking institutions participating in the Colombian large-value payment system during 2019. The objective is to train the artificial neural network to classify banking institutions based on their behavior as captured by the selected 113 features.[1]

In this section, we discuss the methodologies we employ for pattern recognition and dimensionality reduction. First, we introduce the feedforward artificial neural network architecture and explain how it can be applied to our dataset. Second, starting from transactional data, we describe the feature selection procedure that defines the set of features for each institution in our

[1] Focusing on banking institutions not only allows for an easier and more tractable presentation of results but also acknowledges that banks are the most contributive type of institution from several viewpoints (e.g. contribution to sent and received payments, centrality in the payments network). For illustrative purposes, the main results for all financial institutions are reported in Appendix 1.

system. Third, we present principal component analysis and describe how we apply it to our data for the dimensionality and noise reduction of the set of features.

2.1. The feedforward artificial neural network architecture

We employ a feedforward artificial neural network to classify financial institutions.[2] We consider a set of financial institutions each characterized by a set of distinctive features. In our case, $p_{t,v}$ is the *v* feature of the institution corresponding to example *t*, defining a matrix *P* of size $T \times V$, where *V* is the number of features per example and *T* is the number of examples; an example corresponds to a financial institution on a certain date. The feedforward artificial neural network is trained to recognize the pattern of features of each of the financial institutions within the training set.

We define two nonlinear functions with a neural network structure, ϕ and ψ, representing respectively the encoder and the decoder functions. Namely,

$$\phi : \Re^V \rightarrow \Re^H \text{ and } \psi : \Re^H \rightarrow \Re^N \qquad [1]$$

where *H* is the size of the hidden variables set in which the initial features are encoded; and *N* is the number of classified financial institutions. The efficiency of the classification is evaluated according to the *cross-entropy* (CE),

$$CE(A^{(t)}, Q^{(t)}) = CE(\psi(\phi(Q^{(t)})), Q^{(t)}) = -\sum_{t=1}^{T} \psi(\phi(Q^{(t)}))\, log(Q^{(t)}) \qquad [2]$$

where $A^{(t)}$ and $Q^{(t)}$ are the predicted and target (i.e. observed) classifications, respectively, of example *t*.

In our case, we have a feature matrix *P* that is a two-dimensional 6369×113 matrix that contains 113 leading indicators or features ($V = 113$) for the banking institutions that participate in the large-value payment system during 2019. For each of the 6369 examples ($T = 6369$), we have the corresponding identity of the financial institution, i.e. it is a supervised machine learning model. Therefore, P is the input to our artificial neural network, with the element $p_{t,v}$ corresponding to the v feature of the t example. As usual, to avoid issues related to the scale of features across different financial institutions and days, matrix *P* is row normalized.

[2] In a feedforward artificial neural network, the connections between nodes do not form cycles or feed-back loops. Thus, the outputs can be expressed as deterministic functions of the inputs, and so the whole network represents a multivariate non-linear functional mapping (see Bishop, 1995).

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,V} \\ p_{2,1} & p_{2,2} & & \\ \vdots & & \ddots & \vdots \\ p_{T,1} & & \cdots & p_{T,V} \end{bmatrix} \quad [3]$$

The target matrix Q contains the actual target class. It is a binary array with $T = 6369$ rows (i.e. the same as P) and Ncolumns that identifies to which class (i.e. financial institution identity) each row belongs to; in each row of Q there is a single column with a value of 1 matching the identity of the financial institution, and the rest are zeros. For instance, $q_{t,n} = 1$ corresponds to features in example t belonging to financial institution n.

$$Q = \begin{bmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,N} \\ q_{2,1} & q_{2,2} & & \\ \vdots & & \ddots & \vdots \\ q_{T,1} & & \cdots & q_{T,N} \end{bmatrix} \quad [4]$$

Training the feedforward artificial neural network will attain a prediction matrix A, which has the same dimensions as Q. However, unless complete certainty is achieved, it is a non-binary array. In our case, the neural network architecture achieves for each row in A the probability associated with each class.[3] Therefore, elements in each row of A are in the [0,1] range, and the sum of each row equals 1.

Regarding the architecture, we select a standard two-layer network, with one hidden layer and one output layer. This architecture is the simplest and most commonly used in economic and financial applications (Zhang, et al., 1999; McNelis, 2005; Witten, et al., 2011). Figure 1 depicts the architecture, where superscripts identify the layer (i.e. first or second); $p_{v,t}$ are the elements of the input matrix P; Y is the number of neurons in the hidden layer; r are the net input vectors that result from the weighted sum (Σ) involving the weights and bias terms that change while training, w and b, respectively. About activation or transfer functions in the hidden and output layer, $f^1$and f^2, respectively, the first accommodates a customary log-sigmoid function, whereas the second is

[3] This is achieved by using a *softmax* transfer function, as explained next.

a *softmax* function.[4] The number of neurons in the output layer is the number of financial institutions, N; the number of neurons in the hidden layer, Y, may be adjusted. As before, because this is a classification problem, the cross-entropy fitness measure is preferred.

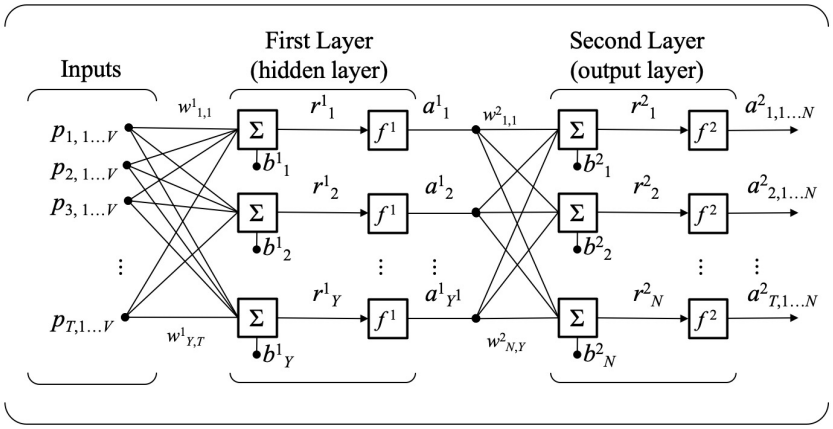

Figure 1. Neural network architecture.

Our work is aimed at out-of-sample classification. Therefore, when training the neural network, we should avoid overfitting.[5] The overfitting problem may be described as the model's ability to succeed at fitting in-sample but to fail at fitting out-of-sample (see Shmueli, 2010; Varian, 2014). Overfitting occurs when the parameters in the artificial neural network, namely w and b, depend too strongly on the details of the particular examples used to produce them (Witten, et al., 2011). In this sense, the role of artificial neural networks is to provide general non-linear mappings between a set of input variables and a set of output variables. The goal is not to memorize the training data, but to model the underlying generator of the data (Bishop, 1995).

In our case, we avoid overfitting through early stopping. Instead of allowing the algorithm to attain the minimal in-sample error (at the expense of out-of-sample error), we stop the minimization process before the complexity of the solution inhibits its generalization capability. As we stop

[4] The log-sigmoid function in the hidden layer is of the form $f(x) = 1/(1 + e^{-x})$. Regarding the output layer, the softmax function is interesting for our classification purposes as its outcome can be interpreted as the probabilities associated with each class; this is particularly convenient as we use cross-entropy as our error measure (see Bishop, 1995; Hagan, et al., 2014). After dropping the superscript corresponding to the second layer, the softmax function in the output layer of Figure 1 is of the form $f(r_n) = e^{r_n}/\sum_{j=1}^{N} e^{r_j}$.

[5] The training of this feedforward artificial neural network is performed by back-propagation. That is, training involves an iterative procedure for minimizing the classification error, with adjustments made in the parameters in a backwards sequence (see Bishop, 1995).

training before the minimum in-sample is reached, then the network will be less likely to overfit (Hagan, et al., 2014).

An intuitive and customary early stopping criterion is cross-validation. For this purpose, the data is divided into three independent datasets: a training dataset, a validation dataset, and a test dataset, which are randomly selected with an approximate 70%, 15%, 15% allocation, respectively (see Hagan et al., 2014); with this partition, training, validation, and test datasets comprise 4459, 955, and 955 examples, respectively. The training dataset is used to train the artificial neural network, i.e. to minimize the difference between the prediction and the actual target value, measured with cross-entropy. As the neural network is trained, the validation dataset is used simultaneously to check how the estimated parameters fit out-of-sample data. As we expect the validation error to start increasing as overfitting arises (Hastie et al., 2013), the validation dataset is used to determine when to stop training. After training, the error obtained on the test dataset is used to evaluate the out-of-sample performance of the artificial neural network, i.e. its generalization capability.

Regarding the number of neurons in the hidden layer, Y, we tried several options in the 20-110 range with 10 neuron increments. As shown in the results (Section 3), the classification error in the test data set becomes stable when using 60 neurons or more.

2.2. Feature Selection

In payments systems, financial institutions send and receive funds to and from each other through time, for different payment concepts (e.g. (un)secured lending, securities purchases, foreign exchange transactions, third-party transfers, central bank repos, etc.). For each payment concept, we can create a time series of payment networks. Let N be the number of financial institutions, D the number of days, and M the number of payment concepts, the $N \times N \times D \times M$ hypermatrix X accommodates all transactions between financial institutions. Each element $x_{p,q,d,m}$, corresponds to the value of funds sent by financial institution p to financial institution q on day d under the type of payment m.

Hypermatrix X contains the behavior of each financial institution. However, the behavior of each financial institution is manyfold, as sender and receiver of payments for each type of payment. The behavior of financial institution p as a sender of payments is contained in elements $x_{p,1\dots N,1\dots D,1\dots M}$, whereas its behavior as a receiver is contained in elements $x_{1\dots N,p,1\dots D,1\dots M}$. Therefore, a traditional manner to detect an anomaly in the behavior of a financial institution involves determining what is to be regarded as normal from its interactions with all other financial institutions for all payment types during a sample period. Nevertheless, this may turn computationally burdensome: the behavior of each financial institution at day d depends on $2 \times M \times (N - 1)$ relations and the

behavior of all financial institutions is a $N \times M \times \big(2 \times (N-1)\big)$ problem. There is a dimensionality problem.

In the Colombian case, the large-value payment system is non-tiered, with all kinds of financial institutions participating directly. There are 111 financial institutions in the large-value payment system during 2019, comprising commercial banks (26), other banking institutions (22), securities broker-dealer firms (19), insurance firms (13), private pension funds (4) and investment funds (26).[6] And there are 20 types of payments worth studying because of their contribution to total payments and to the study of payments behavior ($M = 20$).[7] Therefore, the behavior of a single financial institution on day d consists of $4400 = 2 \times M \times (N-1)$ elements, whereas the behavior of all financial institutions on day d is contained in about half a million elements. As we focus on banks, the behavior of a single bank on day d consists of $1000 = 2 \times M \times (N-1)$ elements, whereas the behavior of all banks on day d is contained in 26000 elements.

Reducing the dimensionality is convenient to detect individual behavior anomalies. Payments systems have been extensively studied (McAndrews & Rajan, 2000; Becher, et al., 2008; Bernal, et al., 2012; Diehl, 2013; Denbee, et al., 2014; Martínez & Cepeda, 2018), and—thus—some measures to monitor financial institutions are already well-established as leading indicators of their behavior. We select several types of measures, which may be conveniently classified depending on their aim.[8] First, measures intended to assess financial institutions' contribution to payments, such as the amount of net and gross payments, sent and received. Second, those intended to study how financial institutions fund their payments, such as initial balance at central bank's accounts, the contribution of liquidity recycling (i.e. using received funds to make payments), and contribution of liquidity savings mechanisms (e.g. multilateral offsetting). Third, those intended to capture the timing of payments during the day, in the form of aggregated hourly payments from 7:00 to 20:00. Fourth, measures aimed at assessing the concentration by payment types and by counterparties. Fifth, those aimed at evaluating the importance in the payments network, measured by different network centrality metrics (degree, strength, authority, hub, SinkRank, PageRank).[9] Sixth, those intended to gauge the systemic footprint in case of failure, in the form of the simulated decrease in overall payments due to inability to make discretionary payments—those payments

[6] Other institutions participate directly in the large-value payment system, such as the Central Bank, the Ministry of Finance, information processors, and financial infrastructures. They have been excluded as our aim is to represent financial institutions' payment behavior.

[7] About 200 types of payments have been identified as uninformative or of limited informational content about financial institutions behavior, such as fees and fines paid to financial infrastructures, accounting adjustments, and taxes. They are not excluded but considered in an aggregated manner.

[8] The exact features, their description and calculation, are not revealed for confidentiality reasons.

[9] See Kleinberg (1998), Brin and Page (1998), Newman (2010), Soramaki and Cook (2013), and León, et al. (2018) for a comprehensive review of these centrality measures. We use the Financial Network Analytics (FNA) platform to calculate these centrality measures.

that financial institutions can deliberately delay or refuse to make.[10] We refer to the last two types as the network features and the simulation-based features, respectively, which pertain to the systemic approach to the monitoring of financial systems.

Based on these six classes of measures, the set of 113 leading indicators of financial institutions payment behavior is created from hypermatrix X. Using this set of leading indicators reduces the dimensionality from 4400 elements per financial institution on day d to 113. If all financial institutions are considered, the input is contained in a two-dimension matrix P with 24234 rows and 113 columns; if financial institutions are limited to banking institutions (i.e. the most contributive type of financial institution), P is a 6369-row and 113-column input matrix.[11]

Although working on 113 features is suitable in our case, such a non-small number of features may contain potentially redundant or noisy data. Further reducing the number of elements may contribute to test the robustness of the chosen features and the classification model. Instead of subjectively discarding leading indicators, we implement principal component analysis (PCA) dimensionality reduction on the 113 selected features.[12]

2.3. Principal component analysis dimensionality reduction

PCA is an unsupervised method for feature selection that performs an orthogonal transformation of the data to find high-variance directions while discarding low-variance ones (Mehta, et al., 2019). PCA finds a mapping from original features into a new lower-dimensional set of features, with minimum loss of information.

In our case, PCA feature selection works on the 113×113 covariance matrix from input matrix P, namely $C = P'P$. As in [5], PCA decomposes C into eigenvector and eigenvalue matrices, Γ and Λ, respectively.

$$C = \Gamma\Lambda\Gamma' \qquad [5]$$

[10] In this case, we use the Financial Network Analytics (FNA) platform to perform a simulation of payments under the assumption of a financial institution being unable to make discretionary payments during the day. The amount of payments left unsettled with respect to the observed payments is a measure of the systemic footprint of that financial institution in case of failure. This simulation is performed for each financial institution, for each day in the sample. For more details, we refer to the FNA documentation in https://fna.fi/technologies/#Simulation.

[11] Not all financial institutions participate every day in the large-value payment system during the period under analysis. Thus, when all financial institutions are considered, the number of rows is 24234 instead of $26973 = N \times T$; when only banks are considered, the number of rows is 6369 instead of $6370 = N \times T$.

[12] References on PCA for dimensionality reduction are Vishwanathan, et al. (2010), Sree and Venkata (2014), Alpaydin (2014), Ding and Tian (2016), and Mehta, et al. (2019).

Λ is a diagonal matrix, with diagonal elements corresponding to the eigenvalues of C, such that the eigenvalues are in decreasing order of magnitude. Γ is a matrix containing the eigenvectors as columns paired to the eigenvalues, such that the *i*-th column contains the *i*-th eigenvector. The first principal component, corresponding to the first column in Γ, lies in the direction of maximum variance of the samples, whereas the second column corresponds to the direction of maximum variance in the remaining data, and so on (Ding & Tian, 2016).

By choosing a subset of eigenvectors, Γ^*, it is possible to map the original set of features into a lower-dimensional set, X^*. As in [6], X^* is attained by calculating a projection of X. If Γ^* is a subset that contains the first i eigenvectors of Γ, X^* is a mapping of the 113 features into i features. The retained variance in this mapping is calculated as the contribution of the corresponding eigenvalues in Λ^* to the sum of eigenvalues in Λ.

$$X^* = X' \Gamma^* \qquad [6]$$

We select the number of eigenvectors to attain X^* by choosing a minimum retained variance target. When a 0.90 minimum retained variance target is chosen, the number of features in $X^*$is 26. Therefore, the number of features decreased from $V = 113$ to $V^* = 26$. When the PCA-based feature selection method is implemented, matrix Pchanges—but Q remains unaltered. Based on [6], with a 0.90 minimum retained variance target, P^* is the 26-column projection of 113-column matrix P.

$$P^* = \begin{bmatrix} p^*_{1,1} & p^*_{1,2} & \cdots & p^*_{1,V^*} \\ p^*_{2,1} & p^*_{2,2} & & \\ \vdots & & \ddots & \vdots \\ p^*_{T,1} & & \cdots & p^*_{T,V^*} \end{bmatrix} \qquad [7]$$

3. Results

We present the three main experiments for pattern recognition we performed. First, in the base case experiment, we use the full set of our 113 selected features to train and validate the artificial neural network varying the number of neurons in the hidden layer. Second, we remove the network and simulation-based features and repeat the training and validation to compare the performance with the full model. Third, we explore the possibility of dimensionality and noise reduction in this approach, by redefining our full set of features with PCA, i.e. identifying a smaller set of linear

combinations of our initial features that maximize the fraction of explained variance in the feature space.

The out-of-sample classification performance of the full model is remarkable. In the base case scenario, which corresponds to the classification of 26 banks based on 113 features, the out-of-sample mean classification error attained when using 60 or more neurons is about 3 percent, and errors do not cluster in any particular bank. When we discard features related to a systemic approach to financial institutions' behavior (i.e. network and simulation methods), the mean classification error increases around 29 percent, which verifies the contribution of such features to the performance of the model. When PCA-feature selection is implemented, we reduce the number of features from 113 to 26 but the mean classification error doubles with respect to the base case scenario. We present each of these three cases in detail next.

3.1.1. Base case scenario

As artificial neural networks' results are dependent on initialization parameters, we run 100 independent training processes for several choices of the number of neurons in the hidden layer. We report the mean classification error attained for each choice, for the training, validation, and test sets. The overall performance attained with the selected architecture is displayed in Table 1 and Figure 2. The lowest mean classification error in the test set, 2.80 percent, is attained with 90 neurons in the hidden layer.

Set	Number of neurons in the hidden layer									
	20	30	40	50	60	70	80	**90**	100	110
Training	1.87 (4.60)	0.99 (0.52)	0.84 (0.41)	0.84 (0.38)	0.80 (0.37)	0.80 (0.82)	0.77 (0.36)	**0.74 (0.34)**	0.81 (0.46)	0.86 (0.85)
Validation	4.94 (4.47)	3.46 (0.69)	3.17 (0.63)	3.02 (0.60)	2.90 (0.62)	2.94 (0.95)	2.80 (0.55)	**2.64 (0.58)**	2.70 (0.65)	2.87 (1.07)
Test	5.20 (4.47)	3.65 (0.72)	3.37 (0.67)	3.25 (0.69)	3.08 (0.58)	2.96 (0.88)	2.88 (0.50)	**2.80 (0.50)**	2.89 (0.62)	3.07 (1.18)

Table 1. Mean classification error for different choices of the number of neurons in the hidden layer. Calculated on 100 independent training processes; standard deviation is reported in parenthesis. The lowest mean classification error in the test set is in bold.

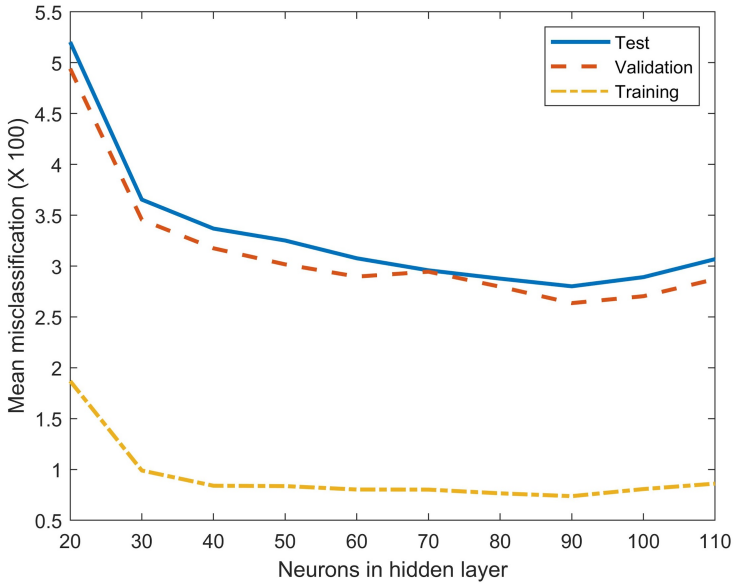

Figure 2. Mean classification error for different choices of the number of neurons in the hidden layer. Calculated on 100 independent training processes.

For the test set, Figure 3 shows the boxplot for our choices of the number of neurons. The top and bottom of each box are the 25th and 75th percentile of the samples, respectively; the line inside the box is the median, and the whiskers mark a +/-2.7 standard deviation limit that reveals outliers (shown with +). As expected, for low numbers of neurons the classification error is higher and more dispersed. Based on Figure 3 and Table 1, the lowest mean classification error and dispersion occur in the 80-90 neurons range.

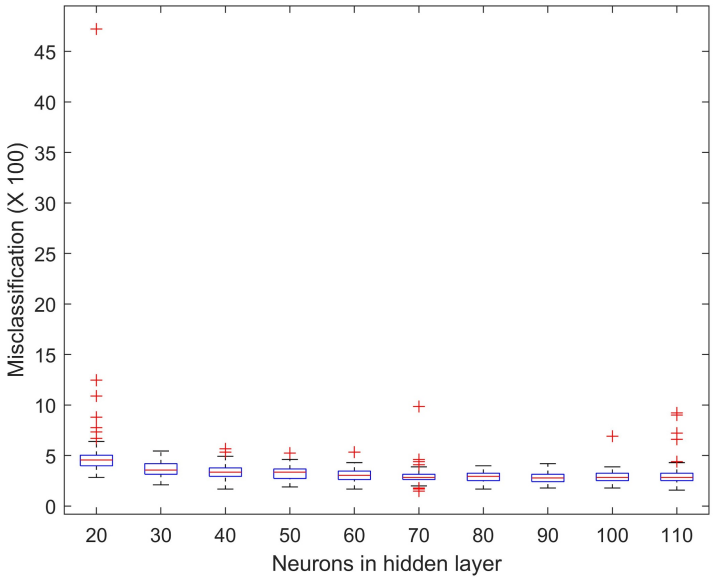

Figure 3. Boxplot of classification error for the test set, for different choices of the number of neurons in the hidden layer. Calculated on 100 independent training processes.

Table 2 shows the lowest classification error for each choice of neurons for the test set; that is, the best of the 100 independent runs for each number of neurons. The lowest classification error, 1.47 percent, is attained in a run with 70 neurons.

Set	Number of neurons in the hidden layer									
	20	30	40	50	60	**70**	80	90	100	110
Test	2.83	2.09	1.68	1.88	1.68	**1.47**	1.68	1.78	1.78	1.57

Table 2. Lowest classification error for different choices of the number of neurons in the hidden layer. The overall lowest classification error is in bold.

Besides the classification error, we report the confusion matrix.[13] The confusion or misclassification matrix is a square table that relates the target class (in rows) with the output class achieved by the model (in columns). For a classifier to have good accuracy, most of the predictions must be represented along the diagonal of the confusion matrix (i.e. predicted class matches observed class), with the rest of the entries (i.e. below or above the diagonal) being close to zero (Han & Kamber, 2006).

Figure 4 exhibits the confusion matrix corresponding to the lowest classification error attained reported in Table 2. As expected from a good classifier, most predictions match the target classification in the diagonal of the confusion matrix, whereas a few (14 out of 955) are outside the diagonal. Further, as not only predictions outside the diagonal are scarce but they are not clustered in any bank in particular, it is noticeable that the model attains a good degree of generalization for all banks in the sample.

[13] We also report the receiver operating characteristic (ROC) curve (see Appendix 1). Performance shown by the ROC curve concurs with reported classification error and the confusion matrix.

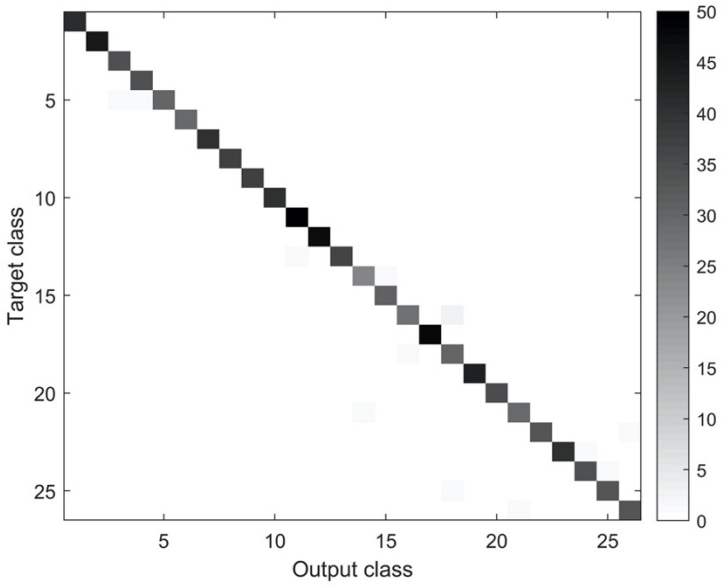

Figure 4. Confusion matrix of lowest classification error. The lowest classification error was achieved in a run with 70 neurons.

3.1.2. The contribution of network and simulation-based features

One of the most important traits of our dataset is the usage of features corresponding to a systemic approach to financial institutions' payment behavior, namely network and simulation-based features. There are ten (out of 113) of those features. Table 3 exhibits the mean classification error attained for each choice of the number of neurons, for the training, validation, and test sets after removing these ten features.

Set	Number of neurons in the hidden layer									
	20	30	40	50	60	70	**80**	90	100	110
Training	2.11 (0.93)	1.49 (0.65)	1.21 (0.51)	1.30 (0.98)	1.10 (0.42)	1.03 (0.43)	**1.09 (0.41)**	1.07 (0.40)	1.08 (0.46)	1.13 (0.73)
Validation	5.76 (1.16)	4.70 (0.79)	4.10 (0.60)	3.94 (1.15)	3.74 (0.66)	3.60 (0.68)	**3.67 (0.64)**	3.57 (0.72)	3.53 (0.83)	3.60 (0.89)
Test	6.00 (1.25)	4.65 (0.72)	4.19 (0.77)	4.15 (1.18)	3.81 (0.61)	3.84 (0.67)	**3.61 (0.57)**	3.71 (0.64)	3.66 (0.72)	3.79 (0.92)

Table 3. Mean classification error for different choices of the number of neurons in the hidden layer, excluding network and simulation-based features. Calculated on 100 independent training processes; standard deviation is reported in parenthesis. The lowest mean classification error in the test set is in bold.

According to Table 3, the lowest mean misclassification in the test set is achieved when using 80 neurons, 3.61 percent. As the lowest mean classification in the test set in Table 1 is 2.80 (i.e. the base case scenario), the gain in classification performance from including ten network and simulation-based features is about 22.44 percent. That is, by increasing the number of features in 9.71 percent the mean classification error decreases 22.44 percent.[14] Then, it is fair to say that network and simulation-based features enhance the classification performance in a definitive manner.

Interestingly, the converse is not true. If we remove all features that are not related to network and simulation-based methods, the classification performance is poor. The lowest mean classification error is about 43 percent when using 110 neurons. Therefore, network and simulation-based features enhance the performance of the model but are not sufficient by themselves.

3.1.3. PCA feature selection

When PCA feature selection is implemented on the dataset, we work with 26 features instead of 113. Table 4 shows the results attained when feature selection is implemented. As expected, misclassification error increases but remains low. The lowest mean misclassification error in the test set is achieved when using 90 neurons, 6.19 percent. This is about 2.2 times the lowest mean misclassification in the base case scenario.

[14] Comparing the lowest classification error achieved in each case also shows an enhancement in classification performance from including network and simulation-based features: the lowest classification error decreases from 1.99 to 1.47 (about 26.13 percent).

Set	Number of neurons in the hidden layer									
	20	30	40	50	60	70	80	**90**	100	110
Training	4.30 (0.88)	3.73 (0.83)	3.72 (0.67)	3.62 (0.69)	3.39 (0.71)	3.36 (0.68)	3.34 (0.61)	**3.38 (0.61)**	3.32 (0.70)	3.31 (0.68)
Validation	7.05 (0.99)	6.61 (0.88)	6.32 (0.96)	6.16 (0.85)	6.03 (0.75)	5.99 (0.75)	5.80 (0.78)	**5.94 (0.74)**	5.70 (0.79)	5.79 (0.78)
Test	7.52 (0.88)	6.85 (0.76)	6.45 (0.81)	6.25 (0.76)	6.22 (0.84)	6.13 (0.73)	6.05 (0.75)	**6.19 (0.80)**	5.99 (0.85)	6.03 (0.83)

Table 4. Mean classification error for different choices of the number of neurons in the hidden layer, after feature selection. Calculated on 100 independent training processes; standard deviation is reported in parenthesis. The lowest mean classification error in the test set is in bold.

The lowest classification error after feature selection is attained in a run with 110 neurons, 3.98 percent, whereas the lowest misclassification before feature selection is 1.47 percent. Correspondingly, the confusion matrix for the lowest classification error after feature selection (in Figure 5) shows higher misclassification than when using all 113 features (i.e. before feature selection). However, classification performance is good.

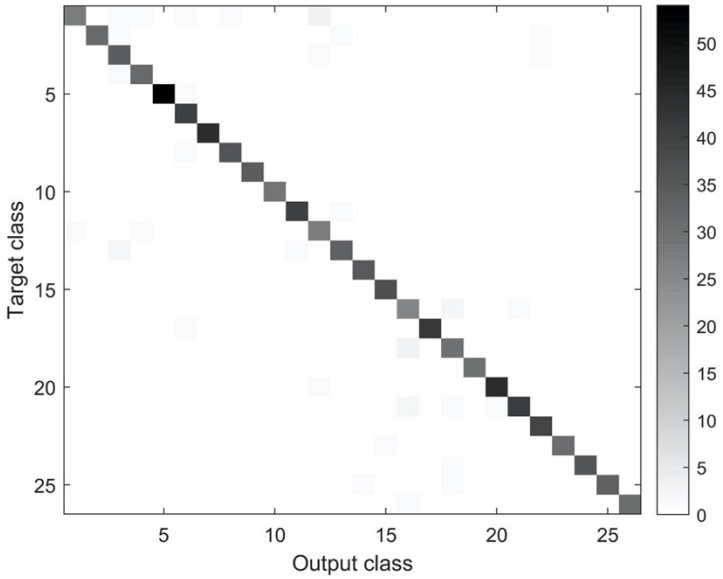

Figure 5. Confusion matrix of lowest classification error, after feature selection. The lowest classification error was achieved in a run with 110 neurons.

The decrease in performance, corresponding to a mean classification error about 2.2 times that obtained in the base case scenario, comes with some gains in computational time. Running the base case scenario (with 113 features) lasts about 1.5 hours, whereas running the lower dimension feature matrix attained with PCA feature selection procedure (26 features) lasts circa 0.4 hours.[15]

The chosen model and its architecture enable us to classify banks based on a set of features that characterize their participation in the large-value payment system. In the base case scenario, with 113 features, the out-of-sample mean classification error attained is about 3 percent, and errors do not cluster in any particular bank. That is, the out-of-sample classification performance is remarkable, and the suggested methodological approach to characterizing banks by their payment patterns is suitable.[16]

[15] Measured as the number of hours to achieve 1000 independent runs, i.e. 100 runs for each one of the 10 choices of number of neurons. The model is run on 128 Gb RAM and 3.60 Ghz CPU desktop computer (no parallel computing was used).

[16] Main results for all financial institutions are reported in Appendix 2 (Table A1, figures A3 and A4). As expected, due to non-banking institutions' less consistent and contributive participation in the large-value payment system, the performance with all financial institutions is lower—yet adequate—, with about a 11 percent out-of-sample classification error. However, evidence in Appendix 2 shows that classification error is particularly high for some financial institutions; most of them are either insurance firms, other banking institutions or investment funds.

Also, removing features related to network and simulation methods cause the mean classification error to increase around 29 percent. That is, their contribution to our classification model is noteworthy. Finally, when we implement a PCA feature selection that retains 90% of the variance in the features matrix, we can reduce the number of features from 113 to 26. As a result, the mean classification error doubles with respect to the base case model. Nevertheless, an error twice as large is around 6 percent, which is consistent with a good performance classification model. Furthermore, with feature selection, the time spent running the model decreased by 73 percent with respect to the base case model.

4. Conclusion

We build an artificial neural network parameterization that represents the characteristic payment patterns of financial institutions in the Colombian large-value payment system. The type of features that serve as building blocks of payment patterns are the contribution to payments, funding habits, payments timing, payments concentration, centrality in the payments network, and systemic impact due to failure to pay. We achieve high-performance out-of-sample classification, with about three percent error. Robustness comes in the form of stable performance after about 60 neurons, and good (yet lower) performance when implementing a PCA feature selection procedure. Additionally, we test that network centrality and systemic impact features contribute to enhancing the performance of the methodology definitively.

From a payment system monitoring perspective, the main aim of this model is to serve as a tool for anomaly detection. In our case, sizable changes in the ability of the model to classify a financial institution is a signal of a change of its behavior within the payment system. In this vein, variations in individual or joint classification performance may be used as warning signals of behavioral changes. There are some challenges related to such implementation as a monitoring tool. First, deciding on the neural network's training frequency. A frequent (e.g. daily) training may quickly convert anomalous behavior into normal, whereas sporadic training (e.g. yearly) may overlook the dynamics of the system; this is a trade-off that should be addressed. Second, deciding on a threshold to determine what a sizable change in individual classification performance is. We foresee addressing these challenges with an empirical approach that involves an iterative process that fine-tunes the outcome of the monitoring process.

Finally, as in most artificial neural network models, the importance of the features is concealed. Other machine learning methods could shed some light on the features' importance and interactions. This pending challenge will be addressed in a forthcoming research project based on random forest models, that would enable us to further understand how features drive the classification process.

References

Aleskerov, E., Freisleben, B., & Rao, B. (1997, March). Cardwatch: A neural network-based database mining system for credit card fraud detection. In *Proceedings of the IEEE/IAFE 1997 computational intelligence for financial engineering (CIFEr)* (pp. 220-226). IEEE.

Angelini, E., di Tollo, G., & Roli, A. (2008). A neural network approach for credit risk evaluation. *Quarterly Review of Economics and Finance, 48,* 733–755.

Alpaydin, E. (2014). *Introduction to Machine Learning*. The MIT Press: Cambridge.

Atiya, A.F. 2001. Bankruptcy prediction for credit risk using neural networks: A survey and new results. I*EEE Transactions on Neural Networks 12(4)*, 929–935.

Becher, C, Galbiati M, & Tudela M. (2008). The timing and funding of CHAPS Sterling payments. *Economic Policy Review*, Federal Reserve Bank of New York, September, 113-133.

Bekhet, A.H. and Eletter, S.F. 2014. Credit risk assessment model for Jordanian commercial banks: Neural scoring approach. *Review of Development Finance, 4*, 20–28.

Bernal J, Cepeda F, & Ortega F. (2012). Estimating the contribution of liquidity sources in the Colombian large-value real-time gross settlement payment: A preliminary approach. *Journal of Payments Strategy & Systems, 6(2)*, 159-182.

Bishop, C.M. (1995). *Neural Networks for Pattern Recognition*. Clarendon Press: Oxford.

Brause, R., Langsdorf, T., & Hepp, M. (1999, November). Neural data mining for credit card fraud detection. In *Proceedings 11th International Conference on Tools with Artificial Intelligence* (pp. 103-106). IEEE.

Brédart, X. (2014) Bankruptcy Prediction Model Using Neural Networks. *Accounting and Finance Research*, *3 (2)*, 124-128.

Brin, S. & Page, L. (1998). Anatomy of a Large-Scale Hypertextual Web Search Engine. *Proceedings of the 7th International World Wide Web Conference*.

Demyanyk, Y. & Hasan, I. (2009). Financial Crises and Bank Failures: A Review of Prediction Methods. *Federal Reserve Bank of Cleveland Working Papers Series, 09-04R*, Federal Reserve Bank of Cleveland.

Denbee, E., Garratt, R.J., & Zimmerman, P. (2014). Variations in liquidity provision in real-time payment systems. *Bank of England Working Paper Series, 513*, Bank of England.

Diehl, M. (2013). Measuring free-riding in large-value payment systems: the case of TARGET2. *Journal of Financial Market Infrastructures, 1(3)*, 31-53.

Ding, M. & Tian, H. (2016). PCA-based network traffic anomaly detection. *Tsinghua Science and Technology, 21(5)*, 500-509.

Dorronsoro, J. R., Ginel, F., Sanchez, C., & Cruz, C. S. (1997). Neural fraud detection in credit card operations. *IEEE transactions on neural networks*, *8*(4), 827-834.

Eletter, S.F., Yaseen, S.G, & Elrefae, G.A. (2010) Neuro-based Artificial Intelligence Model for Loan Decisions. *American Journal of Economics and Business Administration*, *2(1)*, 27-34.

Farmer, J.D., Gallegati, M., Hommes, C., Kirman, A., Ormerod, P., Cincotti, S., Sanchez, A., & Helbing, D. (2012). A complex systems approach to constructing better models for managing financial markets and the economy. *The European Physical Journal, 214*, 295-324.

Fioramanti, M. (2008). Predicting sovereign Debt Crises Using Artificial Neural Networks: A Comparative Approach. *Journal of Financial Stability*, *4*, 149-164.

Ghosh, S., & Reilly, D. L. (1994, January). Credit card fraud detection with a neural-network. In *System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on* (Vol. 3, pp. 621-630). IEEE.

Hagan, M.T., Demuth, H.B., Beale, M.H, & De Jesús, O. (2014). *Neural Network Design.* Martin Hagan: Oklahoma.

Haldane, A.G., & May, R.M. (2011). Systemic risk in banking ecosystems. *Nature*, *469*(7330), 351-355.

Han, J. & Kamber, M. (2006). *Data Mining: Concepts and Techniques*. Morgan Kaufmann: San Francisco.

Hastie, T., Tibshirani, R., & Friedman, J. (2013). *The Elements of Statistical Learning*. Springer: New York.

Holopainen, M. & Sarlin, P. (2016). Toward robust early-warning models: A horse race, ensembles and model uncertainty. *ECB Working Paper, 1900*, European Central Bank, April.

Khediri, K.B., Charfeddine, L., & Youssef, S.B. (2015). Islamic versus conventional banks in the GCC countries: A comparative study using classification techniques. *Research in International Business and Finance, 33*, 75–98.

Kleinberg, J.M. (1998). Authoritative Sources in a Hyperlinked Environment. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*.

León, C. (2020). Detecting anomalous payments networks: A dimensionality reduction approach. Latin American Journal of Central Banking, (*accepted paper*).

León, C., Machado, C., & Sarmiento, M. (2018). Identifying central bank liquidity super-spreaders in interbank funds networks. *Journal of Financial Stability, 35*, 75-92.

León, C., Moreno, J.F., Cely, J. (2017). Whose balance sheet is this? Neural networks for banks' pattern recognition. *Wilmott, 91*, 34-47.

Martínez, C. & Cepeda, F. (2018). Freeriding on liquidity in the Colombian large-value payment system. *Journal of Financial Market Infrastructures, 6 (4)*, 19-40.

McAndrews, J. & Rajan S. (2000). The Timing and Funding of Fedwire Funds Transfers. E*conomic Policy Review*, Federal Reserve Bank of New York, July, 17-32.

McNelis, P.D. (2005). *Neural Networks in Finance*. Elsevier: Burlington.

Mehta, P., Bukov, M., Wang, C.-H., Day, A.G.R., Richardson, C., Fisher, C.K., & Schwab, D.J. (2019). A high-bias, low-variance introduction to Machine Learning for physicists. *Physics Reports, 810*, 1-124.

Nazari, M. & Alidadi, M. (2013). Measuring Credit Risk of Bank Customers Using Artificial Neural Network. *Journal of Management Research, 5 (2)*, 17-27.

Newman, M.E.J. (2010). *Networks: An Introduction*. Oxford University Press: New York.

Olmeda, I. & Fernández, E. (1997). Hybrid classifiers for financial multicriteria decision making: The case of bankruptcy prediction. *Computational Economics, 10*, 317–355.

Sabetti, L. & Heijmans, R. (2020). Shallow or deep? Detecting anomalous flows in the Canadian Automated Clearing Settlement System using an autoencoder. *Payments Canada*.

Salchenberger, L., Mine, C., & Lash, N. (1992). *Neural networks: A tool for predicting thrift failures. Decision Science, 23*, 899–916.

Sarlin, P. (2014). On Biologically Inspired Predictions of the Global Financial Crisis. *Neural Computing and Applications*, *24 (3)*, 663-673.

Shmueli, G. (2010). To explain or to predict? *Statistical Science, 25(3)*, 289-310.

Soramaki, K. & Cook, S. (2013). SinkRank: An Algorithm for Identifying Systemically Important Banks in Payment Systems. *Economics: The Open-Access, Open-Assessment E-Journal*, *7* (2013-28), 1-27.

Sree, A. & Venkata, K. (2014). Anomaly detection using Principal Component Analysis. International. *Journal of Computer Science and Technology, 5(4)*, 124-126.

Tam, K.Y. (1991). Neural network models and the prediction of bank bankruptcy. *Omega, 19*, 429–445.

Tam, K.Y. & Kiang, M. (1990). Predicting bank failures: A neural network approach. *Applied Artificial Intelligence, 4*, 265–282.

Turkan, S., Polat, E., & Gunay, S. (2011). Classification of domestic and foreign commercial banks in Turkey based on financial performances using linear discriminant analysis, logistic regression and artificial neural network models. *Proceedings of the International Conference on Applied Economics – ICOAE 2011*, 717–723.

Triepels, R., Daniels, H., & Heijmans, R. (2017). Anomaly detection in real-time gross settlement systems. *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017), 1*, 433-441.

Varian, H.R. (2014). Big Data: New Tricks for Econometrics. *Journal of Economic Perspectives*, *28 (2)*, 3-28. doi: 10.1257/jep.28.2.3

Vishwanathan, S.V.N., Schraudolph, N.N., Kondor, R., & Borgwardt, K.M. (2010). Graph kernels. *Journal of Machine Learning Research, 11*, 1201-1242.

Wilson, R.L. & Sharda, R. (1994). Bankruptcy prediction using neural networks. *Decision Support Systems, 11(5)*, 545–557.

Witten, I.H., Frank, E., & Hall, M.A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufman Publishers: Burlington.

Wu, R.C. (1997). Neural Network Models: Foundations and Applications to an Audit Decision Problem. *Annals of Operations Research*, *75*, 291-301.

Zhang, G., Hu, M.Y, Patuwo, B.E., & Indro, D.C. (1999). Artificial Neural Networks in Bankruptcy Prediction: General Framework and Cross-validation Analysis. *European Journal of Operational Research*, *116*, 16-32.

Appendix 1. ROC curves[17]

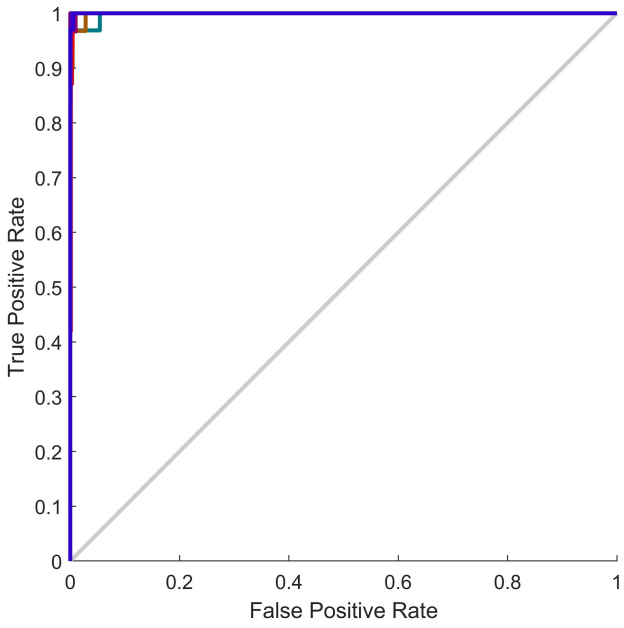

Figure A1. ROC curve of lowest classification error. The lowest classification error was achieved in a run with 70 neurons.

[17] The ROC is a curve that shows the trade-off between the true positive rate (y-axis) and the false positive rate (x-axis) for a given model (Han & Kamber, 2006). When accurate, a model is more likely to encounter true positives than false positives. Hence, a good model is expected to show a steep ROC curve (i.e. close to the y-axis). On the other hand, a poor classifier is expected to show a curve close to the x-axis, whereas a random guess corresponds to the 45-degree line in the plot.

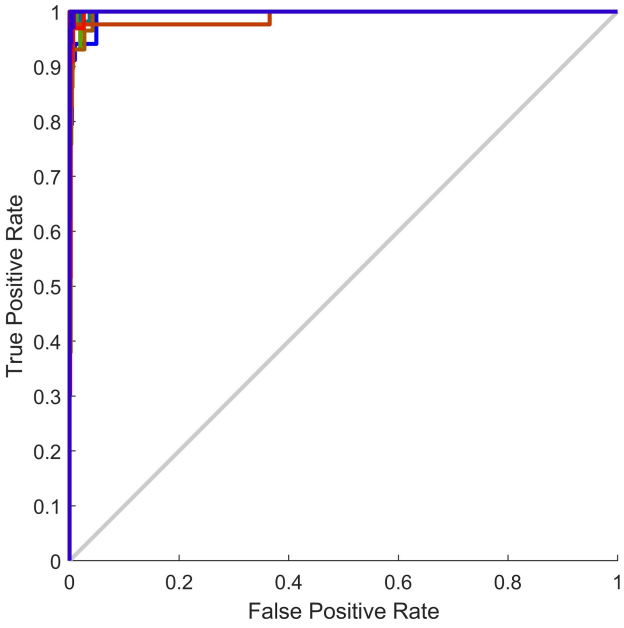

Figure A2. ROC curve of lowest classification error, after feature selection. The lowest classification error was achieved in a run with 110 neurons.

Appendix 2. Including all financial institutions (banking and non-banking)

Set	Number of neurons in the hidden layer									
	20	30	40	50	60	70	80	90	100	**110**
Training	10.51 (2.55)	9.15 (0.62)	8.73 (0.55)	8.56 (0.54)	8.49 (0.48)	8.45 (0.56)	8.39 (0.44)	8.35 (0.46)	8.35 (0.48)	**8.31 (0.42)**
Validation	13.76 (2.39)	12.25 (0.66)	11.79 (0.64)	11.52 (0.53)	11.37 (0.64)	11.21 (0.57)	11.23 (0.50)	11.06 (0.54)	11.01 (0.54)	**11.00 (0.51)**
Test	13.80 (2.29)	12.38 (0.72)	11.83 (0.73)	11.55 (0.55)	11.46 (0.64)	11.43 (0.60)	11.18 (0.61)	11.21 (0.56)	11.22 (0.44)	**11.09 (0.57)**

Table A1. Mean classification error for different choices of the number of neurons in the hidden layer, including all financial institutions. Calculated on 100 independent training processes; standard deviation is reported in parenthesis. The lowest mean classification error in the test set is in bold.

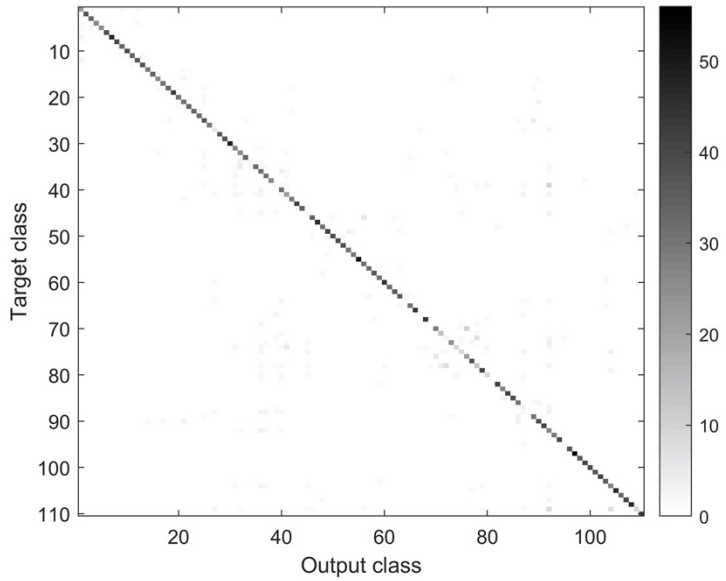

Figure A3. Confusion matrix of lowest classification error, including all financial institutions. The lowest classification error was achieved in a run with 80 neurons.

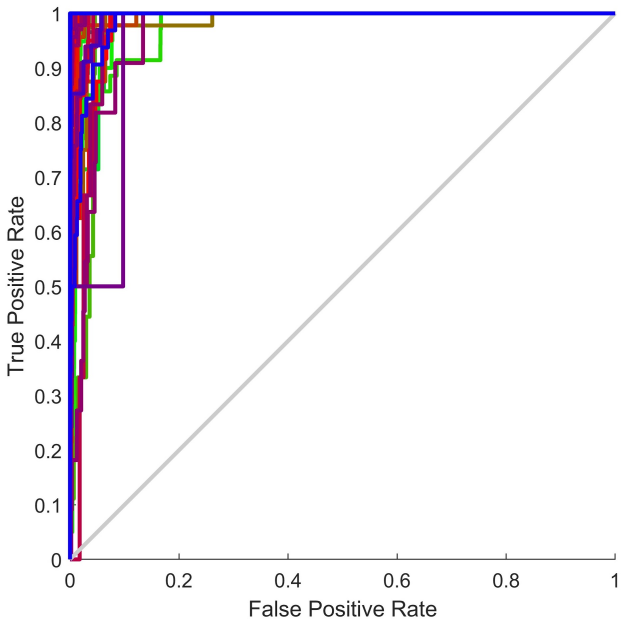

Figure A4. ROC curve of lowest classification error, including all financial institutions. The lowest classification error was achieved with 80 neurons.