Quantitative Finance

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/rquf20

Design of adaptive Elman networks for credit risk assessment

Marco Corazza, Davide De March & Giacomo di Tollo

To cite this article: Marco Corazza, Davide De March & Giacomo di Tollo (2021) Design of adaptive Elman networks for credit risk assessment, Quantitative Finance, 21:2, 323-340, DOI: 10.1080/14697688.2020.1778175

To link to this article: https://doi.org/10.1080/14697688.2020.1778175

Published online: 28 Jul 2020.

Quantitative Finance, 2021
Vol. 21, No. 2, 323–340, https://doi.org/10.1080/14697688.2020.1778175

Design of adaptive Elman networks for credit risk assessment

MARCO CORAZZA†, DAVIDE DE MARCH‡ and GIACOMO DI TOLLO*†

†Department of Economics, Ca' Foscari University of Venice, Sestiere Cannaregio 873, Venezia 30121, Italy
‡EvoSolutions SRL, Via Daniele Manin 60, Mestre (VE) 30174, Italy

(*Received 9 July 2018; accepted 22 May 2020; published online 28 July 2020*)

The banks' need of quantitative approaches for credit risk assessment is becoming more and more evident, due to the introduction of the Basel agreements. To this extent, we define an Elman network approach to determine the insolvency of a borrower, and compare its performances with classical neural networks approaches for bankruptcy prediction. Then, we devise an adaptive procedure to select the best network topology, by performing a multi-objective analysis to take into account different compromises between conflicting criteria. We apply our procedure to different real and partly-artificial-case scenarios composed of Italian SMEs by using predictors coming from balance-sheet ratios, credit-history ratios and macro-economic indicators; then, we compare our approach to other ones proposed in the literature and to a standard logistic regression tool used by practitioners; last, given the recent research interest towards the use of qualitative predictors in credit risk assessment, we also apply our approaches on qualitative data. The results show that the Elman networks are effective in assessing credit risk and robust with respect to criteria and data, confirming the applicability of neural networks to bankruptcy prediction. Our contribution adds to the discussion of the ongoing debate about comparing neural networks to standard techniques: in particular, we find Elman Networks to lead to lower classification errors than standard feed-forward networks, whilst results from the comparison to logistic regression vary with respect to the error class considered. As for the data, we remark that the use of macro-economics indicators does not lead to particular improvement in classification accuracy, except when used to improve results coming from the use of qualitative variables only.

Keywords: Credit risk management; Optimisation; Elman network; Adaptive tool; Italian SMEs

JEL Classification: C45, G32, H81

1. Introduction

Credit risk refers to the likelihood that a borrower will not fulfil her/his lending obligations. In order to allocate financial resources (and to define the lending interest rates accordingly), banking institutions are interested in predicting the non-performing position and the default of potential borrowers. This context has led to many efforts to develop quantitative approaches for credit risk (Altman and Saunders 1998, Ravi *et al.* 2008), especially after the worldwide structural increase in the number of bankruptcies and the decline of assets values due to the global crisis.

We can partition the quantitative approaches proposed in the literature into:

- *classification approaches*, in which the credit risk analysis is formulated as a binary classification problem, in which the output of the classification task is a binary variable indicating whether the firm goes into default or not;
- *probability estimation approaches*, in which the goal it to determine the firm's probability of insolvency (West 1985).

It is important, when discussing classification approaches, to distinguish between *credit scoring* and *bankruptcy prediction*: in credit scoring, the objective is to determine whether the credit customer is good or not (Hand and Henley 1997); in bankruptcy prediction instead, the objective it to predict whether the customer will be solvent (i.e. *in-bonis*) or not (i.e. *default*) (Duma *et al.* 2013). When facing bankruptcy prediction, banks are interested in correctly classifying *in-default* firms, since their mis-classifications would lead to grant the credit to an unsafe customer (Chi and Tang 2006). There are also approaches willing to assign the firm to one out of several rating classes (Altman *et al.* 1994, Pacelli and

*Corresponding author. Email: giacomo.ditollo@unive.it

Azzollini 2011), or to predict *when* a firm is likely to go into default (Topaloglu and Yildirim 2009, Dellepiane *et al.* 2015). A broad discussion about these methods is out of the scope of this paper, and we refer the interested reader to Tong *et al.* (2012) and to Baesens *et al.* (2004).

In this paper we are introducing an artificial neural network approach to the problem of *bankruptcy prediction*: we are using a specific *recurrent* neural network topology referred to as an Elman network (Elman 1990), which has been successfully applied to many fields (Gao *et al.* 1996, Stagge and Sendhoff 1997, Hongmei *et al.* 2006), but whose applications to bankruptcy prediction are improvable.

We are applying our Elman network approach to real balance, credit-history and macroeconomic indicators relating to a sample of small and medium Italian businesses. The interest in small and medium-sized businesses (SMEs) has been triggered for two reasons: first, it has been shown that lending to SMEs has a strong positive impact on banks' profitability (Dermine and de Carvalho 2006); then, the default models developed for large firms are not suited to small businesses, since their features are structurally different, and often the set of data relating to the latter is smaller (Eales and Bosworth 1998).

For the above mentioned reasons, the Basel II Capital accord (Basel Committee on Banking Supervision 2006) allows banks to implement a separate model for bankruptcy prediction of SME borrowers (i.e. firms for which reported sales for the consolidated group they belong to is less than 50 million euros). The Basel III accord (Basel Committee on Banking Supervision 2010) introduces measures to strengthen the banks' capital adequacy and to reduce liquidity risk. In this direction the use of neural networks is gaining more and more attention, due to the idea that 'traditional techniques cannot match the fine resolution across the entire range of account profiles that a neural network produces' (Jost 1993). According to the aforementioned documents, banks should use *robust* procedures and processes to adequately compute their capital requirements, and for credit risk management in general (please, notice that *robustness* has the double meaning of *capacity to operate on unseen / missing / incorrect data* and *flexibility to internal policies*). Moreover, great emphasis is given to the necessity for such procedures and processes to have satisfying out-of-sample performances.

In compliance with the aforementioned requirements, we want to develop an approach that has to be:

(i) capable to correctly classify *in-bonis* and *default* firms;
(ii) able to operate with missing and noisy data;
(iii) capable to operate with new data;
(iv) flexible with regards to policies defined by banks' internal regulations;
(v) able to deliver satisfactory out-of-sample performances.

The non-linear nature of neural networks is apt to develop an approach able to satisfy the aforementioned points (i), (ii), (v). On top of this, we want to create a general tool which is able to adapt itself, in order to satisfy the aforementioned points (iii) and (iv). Most state-of-the-art approaches define the network topology before the experimental analysis, either by exploiting previous experience (Liu *et al.* 2007), or by resorting to parameter tuning (Berni *et al.* 2013). In our approach we are developing a *robust* tool able to determine the best network topology with regard to data at hand (point (iii)) and to the criteria to be optimised, which are determined by internal policies and regulations (point (iv)).

With particular reference to the data sets at hand, we want to stress that the two following key issues in bankruptcy prediction are given by the actual data and the actual set of predictors used in the experimental phase.

- As for the the data at hand, these are often provided by financial institutions (Zhao *et al.* 2015, Xiao *et al.* 2016), hence their choice is constrained by the availability and the operational procedures adopted by banks, when not imposed by the law. Apart from a small number of cases (Barboza *et al.* 2017 and the references therein), most of the contributions tackle sets of data of limited sizes, hence a full experimental validation and comparison is not generally possible. In this direction, our paper contributes to the ongoing discussion about the use of intelligent techniques on bankruptcy prediction over data sets of small sizes. Furthermore, we are testing our approach on bigger sets of data constituted both by *artificial instance*s that take into account the features of our original small set of data and by *data set*s used in other contributions on this topic. The corresponding results will be reported in Sections 7.2 and 7.4.
- As for the set of the predictors used, in the literature there is a huge amount of contributions relying on accounting-based variables (such as for our original small data set) and on market-based ones (Ravi Kumar and Ravi 2007, Demyanyk and Hasan 2010), but recent studies have started to investigate the use of another class of indicators, related both to quantitative and qualitative businesses features (Liang *et al.* 2016, Lahmiri and Bekiros 2019) and to country characteristics (Ioannidis *et al.* 2010, Doumpos *et al.* 2017). For instance, many contributions find relationships between macroeconomic variables and bankruptcy (Ali and Daly 2010, Yurdakul 2014), whereas other studies report that the location of the business has an impact on the likelihood of bankruptcy (Filipe *et al.* 2016). Following this direction, in our contribution we also consider the use of macroeconomic indicators (see Section 3 for details). Regarding the qualitative variables, the original small set of data does not contain them, hence it is not possible to include such indicators in the experimental analysis. But, since we are interested in checking our approach over this class of variables, we have also applied it over qualitative data from the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Qualitative_Bankruptcy), already used in Lu *et al.* (2015). The corresponding results will be reported in Section 7.4.

The goals of our paper are:

- to implement a robust neural network approach which is able to identify the best Elman network with respect to data and banks' internal policies;
- to perform an experimental analysis for the bankruptcy classification problem solved by Elman networks;
- to assess whether the proposed tool is able to deal with a multi-objective classification to simultaneously minimise more than one error at a time.

We are evaluating our classification tool with respect to misclassification of *in-bonis* firms (i.e. `misbo`), to misclassification of *default* firms (i.e. `misdef`), and to overall misclassifications (i.e. `overall`). We are also introducing a multi-objective classification approach able to consider simultaneously more than one error to form a Pareto frontier of network models optimised with respect to two classes of errors (Jin and Sendhoff 2008). This approach, referred to as *adaptive Elman network* approach, will be used to identify the trade-off curve between the two classes of errors. In this way, the user will be able to choose the Elman network to use, depending on the errors she/he wants to minimise and on their relative importance.

The remainder of the paper is organised as follows: the related literature is discussed in Section 2. Then, the set of data at hand is presented in Section 3, and the recurrent neural network approach is outlined in Section 4. Section 5 introduces the pre-processing operations we have performed in order to prepare data for the experiments, which are described in Section 6, before discussing their results in Section 7. Section 8 concludes and outlines guidelines for further research.

2. Related works

Many classification models heve been proposed for bankruptcy prediction: early approaches analysed and combined multiple ratios and features to predict potential distress, and to identify the hyperplane that linearly separates default and non-default firms (Beaver 1966, Altman 1968). The shortcoming of these approaches is that they rely on linear separability and normality assumptions, and later approaches have proposed different models to overcome these shortcomings: multiple discriminant analysis, logistic regression (Ohlson 1980), factor analysis (West 1985), just to name the most used ones. An overview of statistical approaches for bankruptcy prediction can be found in Balcaen and Ooghe (2006).

In the last decades, methods stemming from Artificial Intelligence have been applied to bankrupcty prediction: support vector machines (SVM) (Min and Lee 2005, Shin *et al.* 2005, Bellotti and Crook 2009, Sun and Li 2012), Bayesian networks (Sarkar and Sriram 2001, Lili and Shenoy 2017), k-nearest neighbour (Chen *et al.* 2011), kernel extreme learning machines (Zhao *et al.* 2016), kernel Fisher discriminant analysis (Van Gestel *et al.* 2004a), genetic algorithms (Yobas *et al.* 2000), ant colony optimisation (Martens *et al.* 2010), decision trees (King *et al.* 1995), fuzzy logic (Gorzalczany and Rudzinski 2016) and data-mining techniques aimed to extract rules (Shin and Lee 2002, Kim and Han 2003). Amongst these approaches, there is a wide variety of literature about neural networks for credit risk assessment and bankruptcy prediction (Piramuthu 1999, Wu and Wang 2000, Atiya 2001, Pang *et al.* 2002, Rong-Zhou *et al.* 2002, Hamid and Iqbal 2004, Ravi *et al.* 2008), and their comparisons with other approaches such as logit models and discriminant analysis (Ravi *et al.* 2008). The outcome of such comparisons is far from being univocal: some works suggest that neural networks outperform *traditional* approaches such logistic regression, probit and discriminant analysis (Odon and Sharda 1990, Salchenberger *et al.* 1992, Desai *et al.* 1996, Fan and Palaniswami 2000, Galindo and Tamayo 2000); others suggest the contrary (Altman *et al.* 1994, King *et al.* 1995, Yang *et al.* 1999). In many works, it is not even possible to draw unambiguous conclusions (Coats and Fant 1993, Boritz and Kennedy 1995): Li *et al.* (2006) show that multilayer perceptrons are outperformed by SVM, but only on a small sample size; Huang *et al.* (2004) report unrelevant differences when comparing a back-propagation neural network to SVM.

Overall, it has to be reported that neural networks are generally superior to other methods with respect to the type I error (i.e. `misdef`) (see Desai *et al.* 1996, 1997): their features of being robust with respect to noisy data and of grasping non-linear relations over data make them apt to be used when the distribution of the variables at hand is unknown, which is generally the case when facing balance-sheet ratios and other economics and/or financial indicators.

Nevertheless, credit classification techniques are often evaluated with respect to accuracy (i.e. correct classifications and proper modelling of the decision-making process), interpretability (i.e. capacity of providing an explanation about the classification made), and computational times (Hand and Henley 1997, Zhu *et al.* 2013). Despite good reported accuracy and computational times, neural networks' black-box nature makes them not acceptable for regulatory issues, since one must be able to explain why an applicant is refused credit (interpretability, see Tong *et al.* 2012). For this reason, their use in the industry appears to be limited: the most used tool for credit scoring and bankruptcy prediction is still logistic regression (Thomas *et al.* 2002), whose formulation is apt for binary classification, is robust enough to deviations from multivariate Gaussian distributed classes (Efron 1975), and shows better generalisation skills than other classical tools such as least-squares regression (Lim *et al.* 2000, Van Gestel *et al.* 2004b), which are easier to interpret, but less flexible. Some works attempt to combine black boxes' robustness and interpretability either by extracting rules (Martens *et al.* 2007), or by integrating complex approaches such SVM with logistic models (Van Gestel *et al.* 2005).

In this context, applications of recurrent neural networks in credit risk assessment are in their infancy, although this kind of networks are used in other problem arising from the financial and economic areas (Quek *et al.* 2006). Up to the authors' knowledge, there exists only a couple of contributions on the topic of interest for our contribution. Loukeris and Eleftheriadis (2013) apply recurrent neural networks to bankruptcy prediction performing experiments using several types of such networks. Nonetheless they report `overall` error with respect to the total of elements in the data set (rather than with respect to the test set), which does not allow us to compare their work with other approaches from the literature. Nadali *et al.* (2020) apply recurrent neural networks for predicting insolvency of SMEs through the use of a large data set composed by 722,160 firms over thirty variables for five years. The main results are that the recurrent neural networks outperform the multilayer perceptron ones, and that the accuracy scores are in line with, but not better than, those found in literature.

3. Data set

Our prevailing interest is in developing an approach satisfying the research goals presented in Section 1 through the use of data sets of *small* size. The main reason for this choice is that, for this kind of investigations, banking institutions generally provide small data sets as they fear that from data sets of large size is possible 'to infer' something about their commercial, internal, and other policies (Zhao *et al.* 2015, Xiao *et al.* 2016). However, it remains necessary to check our Elman network approach also when applied to *large* data sets, in order to test its learning capabilities in different size-based scenarios. Therefore, in the following we will perform our analyses using both two small sets of real data provided by a bank and two large sets of artificial data and of artificial and real data. Please, notice that the large data sets have been (partly) artificially generated due to the impossibility to obtain large real data sets by the bank. Details on these data sets are given in what follows.

Regarding the small real data sets, we have collected data from 76 small and medium firms (SMEs) that are clients of an Italian bank: this scenario is relevant in practical terms, since SMEs represent an important feature of the Italian economy. The sample period covers three years: for each firm we use annual data between 2001 and 2003. Furthermore, the sample is rather diversified and distributed across industries, and includes missing and noisy values. Please notice that, with respect to the sample period, we have chosen one that is not affected by crises, as we mean to develop a bankruptcy predictor for normal economic-financial periods, in which bankruptcies are less easy to detect. Please notice also that the cardinality of the set of data considered in this paper is similar to those used in many contributions listed in Section 2 and that, as stated above, it is representative of the size of the data set that banks, or analogous companies, typically provide for analyses and studies.

For each firm we have 8 financial ratios (indices) drawn from balance sheets and 7 credit-position ratios (indices) computed by comparing the credit positions with respect to the supplying bank ('Andamentale' ratios) and the overall Italian Banking System ('Centrale dei Rischi' ratios). We have data for 76 firms over three years, hence for each attribute we should have 228 entries. Nevertheless, for each attribute we have found missing and incorrect data, hence, for each attribute, the number of correct entries is smaller than its theoretical value. This information will be important to devise pre-processing operations, as we will see in Section 5. Statistical descriptions of the data set can be found in table 1, in which q indicates that the index value is expressed by its *quantity*, v indicates that the index value is expressed by its monetary *value*; s refers to *short-term* and *m-l* refers to *medium-long term*. All these data have been used by Angelini *et al.* (2008) and constitute the first small real data set (hereinafter indicated with $sDS1$).

In the second small real data set (hereinafter indicated with $sDS2$), for each firm we consider the same indices described above plus 4 more macroeconomic indicators: the year-over-year variation rate of the Gross Domestic Product (GDP%); the Government Debt to Gross Domestic Product ratio (GD/GDP); the year-over-year Inflation rate (I%); and the year-over-year Unemployment rate (U%). In such a way, we check the prediction capability of indicators catching important aspects of the real economy.†

The sample firms are split into two groups: *in-bonis* and *default*, consisting, respectively, of firms repaying their loan obligations or not at the end of the period. In each of the two small real sets of data we have 48 *in-bonis* and 28 *default* firms. Our task is to apply techniques that learn models from these data to correctly classify to which of the two categories a new unseen firm belongs to at the end of the third year.

Concerning the large artificial data sets, we have generated them having sizes an order of magnitude larger than those of the small real data sets. In particular, starting from the first small real data set, we have generated the first large artificial data set (hereinafter indicated with $lDS1$) as follows:

- for each single year of the sample period, i.e. 2001, 2002 and 2003, we have collected the related data from the 76 SMEs;
- we have split each of the so-created three data subsets in the *in-bonis* firm group and in the *default* firm group;
- for each of these two groups, we have detected the *minimum* and the *maximum* of every index, i.e. $m_{i;y,g} = \min_f \{\text{Index}_{i;y,g,f}\}$ and $M_{i;y,g} = \max_f \{\text{Index}_{i;y,g,f}\}$, respectively, where i indicates the index, y denotes the year, g stands for the *in-bonis/default* firm group, and f designates the firm;
- for each index, we have uniformly randomly generated in the interval $[m_{i;y,g}, M_{i;y,g}]$ a number of artificial observations which is equal to 10 times the cardinality of the *in-bonis/default* firm group g of the year y.

Please, notice that by doing so, we have produced a large artificial set of data conditionally both to the *in-bonis/default* firm group and to the year, thus preserving in some sense the distributional properties of the starting small real data set. As to say that also the large artificial data set might be concerned to customers of the Italian bank. The building of the second large artificial data set (hereinafter indicated with $lDS2$) is

† The values of these indicators have been retrieved from the Italian Istituto Nazionale di Statistica (National Institute of Statistics) on January 20th, 2020.

Table 1. Main statistics of 'Balance Sheet', 'Andamentale' and 'Centrale dei Rischi' indices: *q* indicates that the index value is expressed by its *quantity*, *v* indicates that the index value is expressed by its monetary *value*, *s* refers to *short-term* and *m-l* refers to *medium-long term*.

Index	Min	Mean	Max	Correct entries
Balance Sheet Ratios:				
Cash flow/Total debt	− 1.07	0.04	0.52	225
Turnover/Inventory	0.00	77.95	1877.00	193
Current Liability/Turnover	0.00	11.35	1277.50	222
Equity/Total assets	− 10.69	0.03	1.60	215
Financial costs/Total debts	0.00	0.05	0.72	225
Net working capital/Total assets	− 78.75	− 0.82	1.01	215
Trade accounts receivables/Turnover	0.00	0.51	13.84	222
Value added/Total assets	− 0.22	0.24	1.13	215
Andamentale ratios:				
Utilised credit line/Accorded credit line	0.00	0.81	14.54	219
Unsolved effects (q)/Under usual reserve effects (q)	0.00	0.48	4.48	39
Unsolved effects (v)/Under usual reserve effects (v)	0.00	0.52	5.15	39
Centrale dei Rischi ratios:				
Transpassing s/Accorded credit line s	− 2.74	0.21	18.95	156
Transpassing m − l/Accorded credit line m − l	0.00	0.11	3.99	207
Utilised credit line s/Accorded credit line s	− 0.06	0.80	19.95	157
Utilised credit line m − l/Accorded credit line m − l	0.00	0.85	4.99	208

analogous to that of the second small real data set, in the sense that for each firm we consider the same artificial values of the indices generated for the first large artificial data plus the real values of the aforementioned 4 macroeconomic indicators.

For each of the considered data sets, let $\mathbf{x}$ be a vector of index ratios, or of index ratios and macroeconomic indicators, in a period, as those listed above, and y the corresponding actual class of the firm at the end of the period. The value of y will be 0 if the firm is able to repay the credit, i.e. *in-bonis*, and 1 if it is not, i.e. *default*. All techniques we consider in this article produce a model $f(\mathbf{x})$ to predict the category of a firm. Let $\hat{y}$ be the predicted value produced by the model, that is, $\hat{y} = f(\mathbf{x})$. Let $j = 1, \ldots, m$ be an index to indicate the specific firm (case) from the data used to assess the model and let $I(\cdot)$ be the indicator function that is 1 if the argument is true and 0 otherwise. To assess the classification performance, we look at three criteria:

- *Type I error* (misdefault) the fraction of wrongly classified *default*, that is, firms that are classified to be *in-bonis* when actually they are in *default*:

$$\texttt{misdef} = \frac{\sum_{j=1}^{m} I\{\hat{y}_j = 0 \wedge y_j = 1\}}{\sum_{j=1}^{m} I\{y_j = 1\}}. \quad (1)$$

- *Type II error* (misbonis) the fraction of wrongly classified in-bonis, that is, firms that are classified to be *default* when actually they are *in-bonis*:

$$\texttt{misbo} = \frac{\sum_{j=1}^{m} I\{\hat{y}_j = 1 \wedge y_j = 0\}}{\sum_{j=1}^{m} I\{y_j = 0\}}. \quad (2)$$

- *Overall* misclassifications, indicating the fraction of wrongly classified firms, no matter if they are *in-bonis* or *default*

$$\texttt{overall} = \frac{\sum_{j=1}^{m} I\{(\hat{y}_j = 1 \wedge y_j = 0) \vee (\hat{y}_j = 0 \wedge y_j = 1)\}}{\sum_{j=1}^{m} I\{(y_j = 0) \vee (y_j = 1)\}}. \quad (3)$$

Furthermore, we will assess the classification performance of our networks by drawing Receiver Operating Characteristic (ROC) curves, that plot the true positive rate against the false positive rate for different possible thresholds of a classification exercise, and are used to evaluate the goodness of classification and diagnostic / prediction tests (Fawcett 2006).

4. Neural networks for credit risk assessment

Artificial neural networks (ANNs) are learning strategies inspired by biological neural networks: they are designed in order to grasp non-linear relationships between inputs and outputs. They are composed of elementary units (*neurons*) which are connected to each other via weighted and oriented arcs (*synapses*). *Neurons* may be organised in several architectures, and for any given architecture, one have to determine synapses' weights via a learning algorithm, that iteratively modifies the synapses's weights until a termination condition is met (Kirkpatrick *et al.* 1983, Rumelhart *et al.* 1986, Mitchell 1996).

Recurrent neural networks (RNNs) represent an extension of the ordinary feed-forward layered structure (Rumelhart *et al.* 1986): they have been successfully used to tackle time dependent data (e.g. sequence processing, pattern recognition, natural language processing) and to provide long-range and multi-step-ahead predictions (Su *et al.* 1992, Zhang and Morris 1998, Zhang *et al.* 1998). In a feed-forward network data flows in a network in which no cycles are present. In recurrent networks instead, a cycle is added to the data flow: this cycle may be implemented in different ways, for instance by connecting hidden units to themselves, or by using output values to feed back the hidden neurons (Jordan 1986).

In this paper, we are using a topology in which the cycle is defined by imposing that activation values of (some) hidden neurons have to be used as a part of the network's input. The resulting network is a feed-forward network, enhanced by connecting the hidden neurons to an extra-layer (referred to as *context* layer): both input and context neurons activate the hidden neurons; then hidden neurons activate both output and context neurons. In the error back-propagation phase, recurrent synapses weights are kept constant and not subject to learning: context units will contain the former activation values of hidden neurons, and will represent a short-term memory representing information about the former input. This network topology is referred to as *Elman* network (Elman 1990), and it can be modelled via state-transition of finite automata, allowing a broader analysis of its behaviour (Noda 1992): it has been successfully used in human recognition, and it shows a better dynamic performance and non-linear mapping ability than other neural networks, due to its feedback connections (Liu *et al.* 2015). As for the learning algorithm, standard algorithms like Back-Propagation (Rumelhart *et al.* 1986) can be applied.

In our application we are testing two recurrent topologies obtained by adding a context layer to the feed-forward networks defined by Angelini *et al.* (2008), which will be referred to as *ancestors*: the first topology (referred to as *Standard Elman*) is obtained by adding a context layer to a standard feed-forward network (see figure 1 for an example); the latter (referred to as *Ad-hoc Elman*) is obtained by adding a context layer to a four layered feed-forward network in which the input neurons are first grouped in triads, and each group is connected to one neuron of the following layer. We are considering two versions of the *Ad-hoc Elman*: in the first the context layer is connected with the first hidden layer; in the latter the context layer is connected with the second hidden layer.

In the experimental phase we are first using Elman and ancestor networks as stand-alone solvers, by running experiments with all topologies and by identifying the best topology after the experiments in Sections 7.1 and 7.2. Then, we are proposing an adaptive algorithm that identifies the best topology to meet the users' needs, by minimising one single objective (Section 7.4) or multiple objectives (Section 7.5) in term of classification errors.

Let us develop this last point with respect to two practical scenarios, assuming that a bank could be interested in deciding whether to grant the credit to a firm: in the first scenario this could be made by minimising the `overall` error (equation (3)), without considering differences between `misbo` (equation (2)) and `misdef` (equation (1)); in the second scenario, since it is generally more important to reduce `misdef` than `misbo`, the bank can be willing to take into account more information about the different behaviours remarked when facing either `misbo`, or `misdef`. We are introducing the neural network approach for each of the aforementioned scenarios in Sections 4.1 and 4.2, by introducing a tool based on Elman network that takes into account either `overall` error, or a joint use of `misbo` and `misdef`.

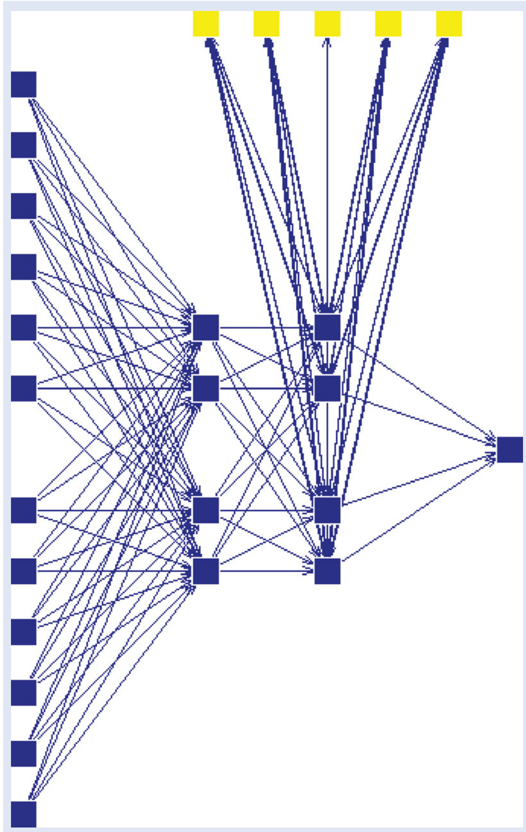

Figure 1. *Standard Elman* network designed for the processing of $sDS1$ and $lDS1$. Context units are shown in yellow.

4.1. Single objective classification

Defining the topology of a network is a non-trivial task: as reported by Liu *et al.* (2007), there are no established procedures to determine the cardinality of the hidden layers. In order to determine the number of hidden neurons, we use an adaptive procedure to identify a topology which minimises the `overall` (see equation 3) error calculated for each of the data set at hand. The algorithm starts with a network with one neuron in the hidden and in the context layer, and iteratively add one neuron until no improvement on the `overall` error has been identified over the last K iterations on the test set (K is to be defined by the user). The adaptive procedure can be extended to any arbitrary number of hidden layers, but for the sake of this paper we set the maximal number of hidden layers equal to two. The procedure can be summarised in the pseudocode of Algorithm 1.

4.2. Multi-objective classification

In this subsection we define a multi-objective adaptive neural network approach which uses more than one error metric to identify the trade-off curve between the predicted `misdef` error and the predicted `misbo` error calculated over a test set: our aim is to provide the user with a set of solutions that are non-dominated with respect to the criteria taken into account. The Pareto front is made out of networks ($ElNet_p$) that simultaneously optimise two contradictory criteria (`misdef` and

Algorithm 1 Single-objective adaptive Elman network for credit risk

Require: Observational data set
Ensure: The optimal single objective Elman network topology
  for b in $1, \ldots, \#$ sub-sampling runs **do**
    $X_b \leftarrow$ Dataset at bth sub-sample
    $Xtrain_b \leftarrow$ Training set at bth sub-sample run
    $Xtest_b \leftarrow$ Test set at bth sub-sample run
    for i in $1, \ldots, \#$ of hidden layer **do**
      $k \leftarrow 0$
      $j \leftarrow 1$
      while $k \leq K$ **do**
        train Elman net $ENet_{ij}$ on $Xtrain_b$
        calculate $Overall_{ij}$ error with equation (3) on the $Xtest_b$
        if $Overall_{ij} \leq$ **all** $Overall_{ij}$ **then**
          $BestENet \leftarrow ENet_{ij}$
          $Neurons \leftarrow (i,j)$
        end if
        if $Overall_{ij} > Overall_{Neurons}$ **then**
          $k++$
        else
          $k \leftarrow 0$
        end if
        $j++$
      end while
    end for
    $BestENet_b \leftarrow BestENet$
    $Neurons_b \leftarrow Neurons$
  end for
  return BestEnet, Overall, Neurons

Algorithm 2 Multi-objective adaptive Elman network for credit Risk

Require: Observational data set
Ensure: The optimal multi-objectives Elman network topology
  for b in $1, \ldots, \#$ sub-sampling runs **do**
    $X_b \leftarrow$Dataset at bth sub sample
    $Xtrain_b \leftarrow$ Training set at bth sub sample
    $Xtest_b \leftarrow$ Test set at bth sub sample
    for i in $1, \ldots, \#$ of hidden layer **do**
      $k \leftarrow 0$
      $j \leftarrow 1$
      while $k \leq K$ **do**
        train Elman net $ENet_{ij}$ on $Xtrain_b$
        calculate $Overall_{ij}$ error with equation (3) on the $Xtest_b$
        calculate $Err1_{bij} \leftarrow \texttt{misbo}_{ij}$ from equation (2)
        calculate $Err2_{bij} \leftarrow \texttt{misdef}_{ij}$ from equation (1)
        if $\exists (g,d) | ENet_{gd} \succ ENet_{ij}$ with $g < i \; d < j$ **then**
          $k++$
        else
          $k \leftarrow 0$
        end if
        $j++$
      end while
    end for
    for all i,j **do**
      identify $ENet_{ij}$ on the Pareto Front of $Err1_{bij}$ and $Err2_{bij}$
    end for
  end for
  return BestEnet, Err1, Err2, Overall, Neurons

`misbo`) and that, for each network d ($d \neq p$), show the following properties (Deb 2001):

(i) the solution provided by $ElNet_p$ is no worse than that provided by $ElNet_d$ with respect to all errors;
(ii) the solution provided by $ElNet_p$ is strictly better than that provided by $ElNet_d$ with respect to at least one error.

The algorithm starts with a one hidden layered network with one hidden neuron and one context neuron, and iterates the search until non-dominated networks have not been found over the last user-defined K iterations. For each set of data, the adaptive procedure records all the test set misclassification errors and identifies the Elman network topologies belonging to the Pareto front. The procedure can be summarised in the pseudocode of Algorithm 1.

5. Data pre-processing

A careful examination of data is important to understand their features, to reveal anomalies and to choose a proper representation. For these reasons, based on the methodology discussed in Bishop (2005), and on the procedure devised by Angelini *et al.* (2008), we apply the data pre-processing steps that are explained in what follows.

Removal and replacement As we can see from the last column of table 1, firms in the original small data set provided by the Italian bank exhibit missing and incorrect values, either due to inconsistency, or to computational errors (i.e. division by zero). Instead of removing all these firms, we adopt a mixed approach: first we erase indices containing more than 30% of missing and wrong values (i.e. *Unsolved effects (q) / Under usual reserve effects (q)*, *Unsolved effects (v) / Under usual reserve effects (v)*, *Transpassing s / Accorded credit line s*, *Utilised credit line s / Accorded credit line s*); then, we replace missing values due to computational errors with the upper limit of the normalised interval (see below), and wrong values due to inconsistency with the index' arithmetical mean. Please, notice that a preliminary analysis has been made to choose the threshold for indices removal: we have let the threshold vary from 10% to 50%, and for each threshold results were similar. Thresholds smaller than 10% would have let us remove *Turnover / Inventory*, triggering a performance deterioration for all networks taken into account.

Normalisation Normalising the inputs of a neural network is an important pre-processing step in general, and particularly when dealing with bankruptcy prediction (Khashman 2010). Here, in order to avoid loss of useful information we use a logarithmic transformation to feed each input node with data belonging to an homogeneous range:

$$\bar{x}_i = \log_u (|\min(0, x_{\min})| + x_i + 1)\,, \qquad (4)$$

where x_i and $\bar{x}_i$ represent respectively the value before normalisation and the normalised value of input x for firm i; x_{min} represents the minimum value of input x. In order to have $\bar{x}_i \in [0, 1]$, the value u is obtained by adding 1 to the maximum value belonging to the index.

Sequence of application Preliminary experiments performed by using the methodology described in Section 4 have shown that different sequences of application of the aforementioned pre-processing operations do not lead to significant performances differences (the lack of significance is assessed by a Wilcoxon test). Hence, in what follows we will detail experiments in which the pre-processing operations are executed in the following order:

(i) removal of indices with less than 30% data;
(ii) removal of missing and incorrect values;
(iii) replacement of missing values due to missing or wrong data with the average of the corresponding index;
(iv) replacement of missing values due to computational errors with the upper bound of the corresponding index;
(v) data normalisation.

We have performed both a Pearson correlation analysis and a Spearman ranked-based correlation analysis over the post-normalised variables, in order to detect possible high correlations amongst the variables of our small original set of data so avoiding highly-correlated variables from the network's input. Pairwise Pearson correlation values amongst variables stay in the range $[-0.31, 0.72]$; pairwise Spearman ranked-based correlation values amongst variables stay in the range $[-0.34, 0.64]$. We have concluded that these values are not as high as to justify the removal of any indicators from the predictors set. Hence, in what follows, the network inputs will be composed by the 11 quantitative indicators resulting by the difference between all the indicators reported in table 1 and those that have been removed in the paragraph *Removal and replacement*. Last, before entering into the experimental analysis of our approach, we want to report that our original database contained a qualitative indicator labelling the industry sector of the borrower. This attribute assumed more than 50 different values over the set of data, whose total number of observations counts up to 76. In this situation, the ratio between possible values of the attribute and total observations is too high to grasp useful information from its use (see Haykin 1994 for a detailed explanation). Hence will not report experiments carried out by including this qualitative indicator in the set of predictors.

6. Experimental setting

We have implemented all neural networks in SNNS† and trained them with *back-propagation through time* (Werbos 1988) without Kalman filter. For each network, the input are the values corrisponding to indices retained after data pre-processing (see Section 5). In particular, the networks for the processing of $sDS1$ and $lDS1$ have 33 input nodes, and the networks for the processing of $sDS2$ and $lDS2$ have 45 input nodes. All networks have a single output node.

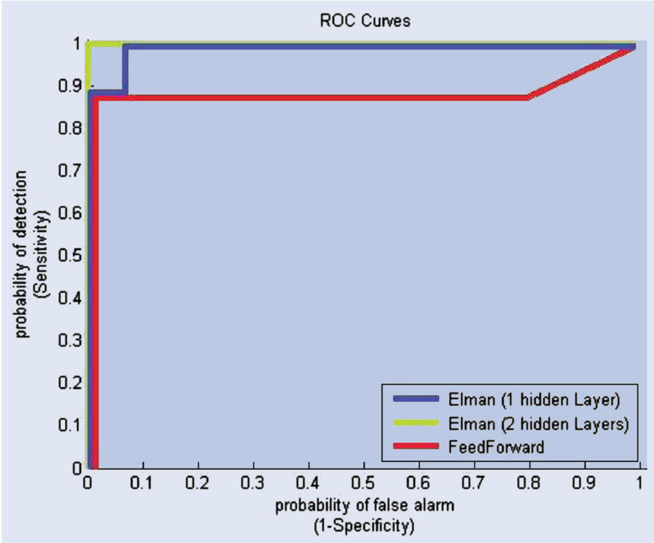

Figure 2. ROC curves for different neural networks coming from the processing of $sDS1$: bigger area under the ROC curves means better classification.

In order to map this output value to one out of two mutually exclusive classes, we have to detect a threshold in order to split the output range in two partitions (Angelini *et al.* 2008, Ravi *et al.* 2008). This could cause some robustness problem with respect to the choice of the threshold value, hence we have performed preliminary experiments on 10 different training-test set partitions, finding that either the network's output is rather close to the desired output, or it shows a value close to the opposite possible value. This phenomenon is confirmed by analysing the ROC curves for different neural network approaches shown in figure 2. ROC curves are plots of the *true positive rate* (probability of detection) against the *false positive rate* (probability of false alarm) for the different possible thresholds of the classification problem, where *false positive rate* indicates the ratio between *negatives incorrectly classified* and *total negatives*, and *true positive rate* indicates the ratio between *true negatives* and *false positives* + *true negatives*. As an example, we have drawn the ROC curves relative to a specific set of data, without lack of generality. Please, notice that the bigger the area under the ROC curve, the more robust (hence, better) the classifier with respect to different thresholds. In our case, the same classification output is obtained by several thresholds, and many thresholds are associated with the uppest-left point. For this reason, as suggested by other works (Tam and Kiang 1992, Desai *et al.* 1996), we have chosen a threshold value of 0.5 (this value is always associated to the uppest-left point in our ROC curves): if $y < 0.5$ then the firm is classified as *in-bonis*; otherwise, the firm is classified as *default*.

† Stuttgart Neural Networks Simulator, developed at the University of Stuttgart (Zell *et al.* 1991).

7. Results and experimental analysis

In this section we are reporting the results obtained by our experiments: first, we have run experiments over different networks and compared their performances; then, we have introduced the strategy proposed in Section 4.1 in order to adaptively search for the best topology, by using a single objective classification approach; last, we have introduced the strategy proposed in Section 4.2 in order to search for the best topology by using a multi-objective classification approach. Furthermore, in Section 7.3, we have compared our results with those coming from the application of the logistic regression, which is currently referred to as the main credit risk assessment prediction model in the industry.

In all experiments we sample two disjoint sets of observations out of the 76 firms when using $sDS1$ and $lDS1$, and out of the 760 firms when using $sDS2$ and $lDS2$: the *training set* (used to learn the parameters of the neural network and of the logistic regression) and the *test set* (used for assessing their performances). For the training set, we sample 53 firms when using $sDS1$ and $lDS1$, and 530 firms when using $sDS2$ and $lDS2$, and we include the remaining firms in the test set. The whole procedure, including sampling, learning of parameters and assessment, is repeated 50 times, each time with a different sub-sample. For each sub-sample the performance of the models on the corresponding test set is assessed on the basis of `misbo`, `misdef` and `overall` over the third year prediction. Finally, we compute the average and standard deviation statistics over the 50 repetitions.

In the following tables we report the number of neurons used in the hidden layer, the average and the standard deviation (expressed in percentage terms) over 50 runs of the sub-sampling techniques for the `misbo`, `misdef` and `overall` classification errors.

7.1. *Stand alone Elman networks applied to sDS1 and to sDS2*

In this section, we are using *Standard* and *Ad-hoc* Elman neural networks as stand alone solvers, and comparing their performances with the ancestor networks' ones. Networks have been trained by minimising the `overall` error.

In tables 2 and 3 we report the resulting statistics of some amongst the best performing topologies for the two classes of ancestor networks with different number of hidden neurons applied to $sDS1$ (columns labelled *No macro*) and to $sDS2$ (columns labelled *Macro*). We observe that the results for the *Standard* ancestor networks are not strongly affected by the number of hidden neurons, and that increasing this number does not imply better performances. The average `overalls` for these networks are between 11.57% and 14.43% in the case in which the macroeconomic indicators are not considered as additional predictors (hereinafter 'No-macro case'), and are between 11.51% and 14.86% in the case in which these indicators are considered (hereinafter 'Macro case'). *Ad-hoc* ancestor networks perform better exhibiting an average `overall` of 7.30% in the former case, and of 7.44% in the latter case.

Results of *Standard* and *Ad-hoc* Elman networks, both applied to $sDS1$ (columns labelled *No macro*) and to $sDS2$ (columns labelled *Macro*), are reported, respectively, in tables 4 and 5. In the latter, in the first column, the number before colon represents the number of context units and the number after colon represents the hidden connected layer. *Standard Elman* networks perform better than the *Standard* ancestor ones. In the No-macro case, the average `overalls` of the Elman models range in the interval [9.04%, 12.00%], whose upper extreme is lower than the most values of the average `overall` obtained by the *Standard* ancestor topology reported in table 2. Likewise, in the Macro case, the average `overalls` of the Elman models ranges in the interval [9.22%, 13.08%], whose upper extreme is lower than the most values of the average `overall` obtained by the *Standard* ancestor topology presented in table 2. The same holds when we compare the *Ad-hoc* Elman network with its ancestor, both in the No-macro case, and in the Macro case: to exemplify, the average `overalls` for 22 different combinations of context nodes is between 6.17% and 8.00%, which is in 13 cases better than the 7.3% shown by the ancestor *Ad-hoc* model reported in table 2. The dominance is even clearer if we look at `misdef` performances, which are of particular interest for banks. Both in the No-macro case, and in the Macro case, in all the experiments the average `misdefs` of *Ad hoc* networks are lower than 9.50% obtained by both their ancestors presented in table 3. Finally, we do not observe any significant difference between connecting the context layer to the first and the second hidden layer. By comparing the aforementioned tables we can conclude that Elman topologies perform better than their ancestor version, with respect to all types of errors.

As for the use of macroeconomic indicators as additional predictors, in the great majority of the performed experiments, the results from all the considered Elman networks are slightly worse than those coming from the corresponding networks in which such indicators are not used as inputs. In other terms, the macroeconomic indicators do not seem be particularly informative for SMEs' credit risk assessment. Likely, this is due to the fact that SMEs, even if 'close' to bankruptcy, react quickly to the dynamics changes of real economy and reflect them just as quickly in the balance ratios, so that making superfluous *ad hoc* sources of information.

7.2. *Stand alone Elman networks applied to lDS1 and to lDS2*

As we have stated in Section 3, our main interest is in developing an approach that satisfies the research goals presented in Section 1 through the use of data sets of small size. However, it remains necessary to check our Elman network approach also when applied to data sets of large size, in order to test its learning capabilities in different size-based scenarios. So, in this section, we carry out the same analyses performed in the previous section but applied to $lDS1$ and to $lDS2$. We recall that, as for the 'Balance Sheet', the 'Andamentale' and the 'Centrale dei Rischi' ratios, $lDS1$ and $lDS2$ are constituted by artificial data generated in such a way to preserve the distributional properties of the starting small real data set. Networks

Table 2. Experiments with *Standard* ancestor networks applied to *sDS*1 (columns labelled *No macro*) and to *sDS*2 (columns labelled *Macro*): mean and standard deviation of the errors calculated on the test set over 50 different samples.

	No macro		Macro		No macro		Macro		No macro		Macro	
H	Mean misbo	SD misbo	Mean misbo	SD misbo	Mean misdef	SD misdef	Mean misdef	SD misdef	Mean overall	SD overall	Mean overall	SD overall
25	10.53	11.59	10.95	11.82	14.00	11.32	13.25	11.10	11.74	8.96	11.51	8.51
26	12.13	11.87	12.01	12.23	13.00	12.10	14.25	12.35	12.43	9.50	12.55	9.41
27	12.80	12.83	13.06	12.96	14.00	11.12	14.75	11.12	13.22	9.89	13.62	9.40
28	10.27	9.84	10.06	10.04	14.00	9.45	14.50	10.45	11.57	7.93	11.69	8.17
29	10.93	10.41	11.15	10.20	12.75	9.32	12.25	9.57	11.57	8.58	11.80	8.67
30	12.13	11.87	12.13	12.23	13.25	11.10	12.50	11.45	12.52	8.94	12.27	9.30
31	13.20	10.89	12.80	10.35	14.75	12.87	14.75	12.11	13.74	8.84	13.33	8.84
32	11.60	12.03	12.06	12.15	12.25	11.10	12.00	11.04	11.83	9.58	11.71	9.29
33	13.60	12.34	13.46	12.46	16.00	11.55	16.00	12.09	14.43	9.43	14.86	9.05

Note: In this table and in the following ones, *H*, when present, stays for *Hidden neurons*.

Table 3. Experiments with *Ad-hoc* ancestor networks applied to *sDS*1 (columns labelled *No macro*) and to *sDS*2 (columns labelled *Macro*): mean and standard deviation of the errors calculated on the test set over 50 different samples.

	No macro		Macro		No macro		Macro		No macro		Macro	
H	Mean misbo	SD misbo	Mean misbo	SD misbo	Mean misdef	SD misdef	Mean misdef	SD misdef	Mean overall	SD overall	Mean overall	SD overall
11 + 11 (2 layers)	6.13	7.59	6.31	7.89	9.50	10.58	9.50	11.01	7.30	6.59	7.44	6.45

have been trained with back propagation, by minimising the `overall` error.

In tables 6 and 7 we present the statistics coming from some of the best performing topologies for the *Standard* and *Ad-hoc* ancestor networks with different number of hidden neurons applied to *lDS*1 (columns labelled *No macro*) and to *lDS*2 (columns labelled *Macro*). Unlike the findings coming from the previous experiments, we observe that the results for the *Standard* ancestor networks are partly affected by the number of hidden neurons: increasing of this number generally implies better performances. The average `overalls` for such networks vary in the interval $[2.76\%, 4.69\%]$ in the No macro case, and vary in the interval $[2.98\%, 5.64\%]$ in the Macro case. *Ad-hoc* ancestor networks perform worst exhibiting an average `overall` of 5.80% in the former case, and of 6.04% in the latter case.

The results coming from the application of *Standard* and *Ad-hoc* Elman networks both to *lDS*1 (columns labelled *No macro*) and to *lDS*2 (columns labelled *Macro*) are presented in tables 8 and 9, respectively. Please, notice that in table 9, in the first column, the number before colon represents the number of context units and the number after colon represents the hidden connected layer. The *Standard Elman* networks perform slightly better than the *Standard* ancestor ones. In fact, both in the No-macro case, and in the Macro one, the variation intervals of the average `overalls` obtained from the two kinds of Elman are more or less overlapped, but the standard deviations of the `overalls` associated to the *Standard Elman* networks are definitely smaller than the standard deviations of the `overalls` associated to the *Standard* ancestors networks reported in table 6. A similar situation holds when we compare the *Ad-hoc* Elman network with its ancestor, both in the No-macro case, and in the Macro one. In fact, the average `overall` from the latter kind of Elman network is very close to the upper extreme extreme of the variation interval of the average `overalls` from the former kind of Elman

Table 4. Experiments with *Standard Elman* networks applied to *sDS*1 (columns labelled *No macro*) and to *sDS*2 (columns labelled *Macro*): mean and standard deviation of the errors (expressed in percentage terms) calculated on the test set over 50 different samples.

	No macro		Macro		No macro		Macro		No macro		Macro	
H	Mean misbo	SD misbo	Mean misbo	SD misbo	Mean misdef	SD misdef	Mean misdef	SD misdef	Mean overall	SD overall	Mean overall	SD overall
4	9.47	10.70	9.19	11.34	11.85	9.64	13.04	9.63	10.70	8.39	10.27	9.00
5	11.20	11.62	12.66	12.78	12.75	9.82	13.90	11.09	12.00	9.28	13.08	10.20
6	8.53	9.62	8.70	9.43	9.52	10.30	9.42	9.79	9.04	8.19	9.22	8.58
10	8.27	9.58	9.10	10.92	10.61	9.91	11.78	9.31	9.48	7.69	10.62	7.28
15	9.73	10.01	10.12	10.11	11.77	10.40	13.06	10.72	10.78	8.93	12.18	8.92
20	9.60	11.28	9.89	12.86	10.54	9.73	10.43	11.09	10.09	8.49	10.90	9.07
25	8.40	9.88	8.40	10.67	11.50	10.49	11.39	11.44	10.00	8.44	10.80	8.00
31	8.13	8.66	9.19	9.01	12.42	10.74	13.17	11.07	10.35	7.90	11.59	8.59
32	8.53	9.43	8.70	9.24	12.22	10.16	12.46	10.87	10.44	8.00	10.02	8.70
33	8.40	9.02	8.15	10.19	13.02	10.67	13.93	12.06	10.78	8.06	10.35	7.72

Table 5. Experiments with *Ad-hoc* Elman networks applied to *sDS*1 (columns labelled *No macro*) and to *sDS*2 (columns labelled *Macro*): in the first column, the first number before colon represents the number of context units and the second number after colon represents the hidden connected layer.

	No macro		Macro		No macro		Macro		No macro		Macro	
H	Mean `misbo`	SD `misbo`	Mean `misbo`	SD `misbo`	Mean `misdef`	SD `misdef`	Mean `misdef`	SD `misdef`	Mean `overall`	SD `overall`	Mean `overall`	SD `overall`
1 : 2	5.73	6.74	6.05	6.54	7.77	7.75	7.65	8.48	6.78	5.90	6.51	5.70
2:2	5.47	6.56	5.71	6.43	6.84	8.13	7.32	7.52	6.17	6.09	6.46	5.77
3:2	5.73	7.38	5.36	7.70	7.60	7.88	8.46	8.78	6.70	6.16	6.90	5.72
4:2	6.40	8.08	6.01	7.88	8.15	8.36	8.44	7.83	7.30	6.11	7.50	5.98
5:2	6.13	7.72	6.60	8.21	9.08	7.97	9.88	8.25	7.65	5.73	7.60	6.03
6:2	6.80	8.02	6.80	9.02	9.13	9.36	8.66	9.63	8.00	7.42	8.20	6.77
7:2	5.87	5.97	6.49	6.67	7.47	7.31	8.17	7.62	6.70	5.14	7.44	4.79
8:2	6.00	7.41	6.10	7.58	9.20	8.03	9.37	8.80	7.65	5.46	7.60	5.23
9:2	5.33	6.46	5.29	5.94	7.30	7.57	6.81	7.67	6.35	5.21	6.09	4.66

Table 6. Experiments with *Standard* ancestor networks applied to *lDS*1 (columns labelled *No macro*) and to *lDS*2 (columns labelled *Macro*): mean and standard deviation of the errors calculated on the test set over 50 different samples.

	No macro		Macro		No macro		Macro		No macro		Macro	
H	Mean `misbo`	SD `misbo`	Mean `misbo`	SD `misbo`	Mean `misdef`	SD `misdef`	Mean `misdef`	SD `misdef`	Mean `overall`	SD `overall`	Mean `overall`	SD `overall`
25	8.42	4.07	9.27	4.44	4.28	2.82	4.75	2.92	4.69	3.21	5.64	3.35
26	7.45	3.79	7.46	3.67	4.28	2.01	4.68	2.10	4.65	2.32	4.68	2.33
27	6.98	2.95	8.20	3.10	4.13	2.01	4.45	2.44	4.32	1.93	4.50	2.17
28	6.74	4.30	6.98	4.86	4.13	1.91	5.03	2.18	4.32	2.10	4.84	2.43
29	6.98	4.55	7.47	4.83	3.66	2.62	3.53	2.78	3.81	3.21	3.61	3.43
30	6.98	3.54	7.91	4.30	4.13	2.09	4.51	2.24	4.20	2.65	4.48	2.99
31	6.18	2.52	6.79	2.90	2.96	2.42	3.14	2.64	3.01	1.73	3.22	1.93
32	5.89	4.04	6.27	4.28	2.43	1.86	2.32	2.07	2.93	1.51	2.98	1.52
33	6.03	2.01	5.96	1.93	3.66	2.70	3.49	2.65	2.76	1.78	3.00	1.94

networks, even though the standard deviation of `overall` from the Ad-hoc ancestor network presented in table 7 is definitely smaller than the standard deviations of the `overalls` from *Ad-hoc Elman* networks. Such a slight dominance of *Ad-hoc Elman* networks with respect to the *Ad-hoc* ancestor networks holds if we look at `misdef` performance, which is of particular interest for banks. Finally, with reference to the *Ad-hoc Elman* networks (see table 9), we do not observe any significant difference between connecting the context layer to the first and the second hidden layer.

By comparing the aforementioned tables we can conclude that Elman topologies perform slightly better than their ancestor versions, with respect to all types of errors. We can notice that Elman topologies generally perform better than their ancestor also when applied to data sets of large size, although not so evidently as the case of their application to small sets of data. Reasonably, this is due to the fact that larger data sets provide to the networks, whatever their architectures are, more information for 'inferring' the unknown relationships between inputs the output. Therefore, the importance of the 'inferential' role played by the topologies in such a context is reduced. This considerations, jointly with the fact that banking institutions provide small sets of data, confirms the need to develop specific network topologies for the processing of sets of data of small size.

Some interesting remarks can be drawn with regards to macroeconomics indicators. Their use do not lead to a significant improvement of the network performances when using ancestor networks. Furthermore, when using Elman networks, their introduction over *sSD*1 obtained poorer performances, and the smaller standard deviation found over the overall error is not enough to compensate poor classification of default firm, which are less accurate than the No-macro case (see tables 4 and 5). The experiments over *lSD*1 confirm this finding, indicating even worst performances with respect to `misbo` and `misdef` (see tables 8 and 9). All these results suggest that their inclusion in the set of predictors does not lead to an improvement of performances, validating a former

Table 7. Experiments with *Ad-hoc* ancestor networks applied to *lDS*1 (columns labelled *No macro*) and to *lDS*2 (columns labelled *Macro*): mean and standard deviation of the errors calculated on the test set over 50 different samples.

	No macro		Macro		No macro		Macro		No macro		Macro	
H	Mean `misbo`	SD `misbo`	Mean `misbo`	SD `misbo`	Mean `misdef`	SD `misdef`	Mean `misdef`	SD `misdef`	Mean `overall`	SD `overall`	Mean `overall`	SD `overall`
11 + 11 (2 layers)	5.61	1.91	5.84	2.11	7.44	0.93	7.79	1.13	5.80	0.51	6.04	0.71

Table 8. Experiments with *Standard Elman* networks applied to *lDS*1 (columns labelled *No macro*) and to *lDS*2 (columns labelled *Macro*): mean and standard deviation of the errors (expressed in percentage terms) calculated on the test set over 50 different samples.

	No macro		Macro		No macro		Macro		No macro		Macro	
H	Mean `misbo`	SD `misbo`	Mean `misbo`	SD `misbo`	Mean `misdef`	SD `misdef`	Mean `misdef`	SD `misdef`	Mean `overall`	SD `overall`	Mean `overall`	SD `overall`
4	7.64	2.70	8.49	2.98	6.54	0.86	6.63	0.91	7.06	0.63	6.78	0.61
5	8.04	2.71	7.84	2.55	4.13	0.56	4.17	0.55	5.05	0.21	5.13	0.22
6	6.43	3.68	6.68	4.12	2.31	0.42	2.58	0.44	3.39	0.16	3.21	0.18
10	6.46	2.45	7.89	2.94	3.32	0.56	3.66	0.60	5.05	0.23	5.63	0.21
15	4.82	3.44	5.25	3.85	3.10	0.43	3.23	0.44	4.03	0.15	4.34	0.16
20	4.19	3.25	4.73	3.74	2.33	0.51	2.53	0.57	3.28	0.19	3.60	0.20
25	4.65	0.98	4.85	0.97	2.07	0.41	2.11	0.39	2.31	0.14	2.18	0.16
31	4.60	0.97	4.47	0.95	3.93	0.51	4.14	0.49	4.03	0.26	3.91	0.24
32	4.41	1.46	4.50	1.51	1.51	0.86	1.45	0.89	3.05	0.72	3.11	0.78
33	5.05	1.74	4.88	1.82	1.46	0.25	1.60	0.25	4.53	0.11	4.43	0.11

Table 9. Experiments with *Ad-hoc* Elman networks applied to *lDS*1 (columns labelled *No macro*) and to *lDS*2 (columns labelled *Macro*): in the first column, the first number before colon represents the number of context units and the second number after colon represents the hidden connected layer.

	No macro		Macro		No macro		Macro		No macro		Macro	
H	Mean `misbo`	SD `misbo`	Mean `misbo`	SD `misbo`	Mean `misdef`	SD `misdef`	Mean `misdef`	SD `misdef`	Mean `overall`	SD `overall`	Mean `overall`	SD `overall`
1 : 2	3.71	3.35	3.94	3.49	5.80	5.78	5.50	5.82	6.07	3.27	6.31	3.36
2 : 2	3.33	2.97	3.71	2.91	4.66	6.47	4.79	6.45	3.16	3.12	3.47	2.92
3 : 2	6.67	5.24	7.14	5.13	4.84	4.35	4.99	4.13	4.27	4.10	4.10	3.88
4 : 2	6.72	5.07	6.69	4.76	4.95	4.53	5.05	4.63	4.34	3.08	4.89	3.39
5 : 2	7.46	4.70	7.05	4.46	5.55	6.43	5.81	5.86	5.34	5.55	4.94	5.42
6 : 2	6.41	5.47	7.91	5.20	4.61	5.34	5.12	6.22	5.43	3.48	5.32	3.56
7 : 2	5.43	3.39	5.74	3.58	7.25	6.07	8.94	6.51	6.24	3.01	6.56	3.69
8 : 2	6.72	5.66	7.06	6.46	7.41	7.23	7.90	7.86	5.91	3.84	6.86	4.01
9 : 2	3.74	3.37	3.89	3.65	5.19	5.01	5.68	5.28	5.74	3.14	6.91	3.61
10 : 2	7.39	6.05	8.42	7.04	5.75	5.40	6.06	6.27	5.08	4.92	6.14	5.63

statement by Barboza *et al.* (2017), that do not use macroeconomic indicators claiming that firm-based measures produced already relevant outcomes.

7.3. *Single objective classification applied to sDS*1 *and to sDS*2

In this subsection (and in Section 7.5), we are generating different Elman networks topologies without choosing one fixed architecture over the 50 repetitions: the topology may change from one sub-sample to the others, as formalised by Algorithm 1. In this way we create a general approach to make the tool robust with respect to several and different data; in what follows, we refer to this approach as 'Adaptive Elman'. Networks have been trained with back propagation, by minimising the `overall` error. Please, notice that on the basis of the findings reported in the above sections, here we address our attention only on the applications to data sets of small size (i.e. *sDS*1 and to *sDS*2).

We report in table 10 the results obtained with the adaptive Elman tool for minimising the `overall` error applied to *sDS*1 (columns labelled *No Macro*) and to *sDS*1 (columns labelled *Macro*). When using `overall` error as the single objective to be minimised, the adaptive network with one hidden layer shows low average misclassification error: 8.26% with a standard deviation equal to 6.16% in the No-macro case, and 6.42% with a standard deviation equal to 6.93% in the Macro case. This shows that the Adaptive Elman tool is very flexible and it is able to correctly predict unseen companies credit risk. Moreover, this approach leads to a very low `misdef` error, which is of particular interests for banks. The performances of Elman networks with two hidden layers are even more robust both without and with the macroeconomic indicators: variability of both `misdef` and `overall` errors decreased compared with the corresponding quantities obtained through the hidden single-layered Elman networks. There is a strong evidence that both methods outperform a standard tool such as the logistic regression.† Furthermore, we remark that their computational times are lower than a brute-force approach consisting of running experiments over all candidate networks to determine which is the most performing.

We want to summarise the results obtained by the different approaches applied to *sDS*1 through showing in figure 3 their positions in the ROC space, in which each discrete classifier is associated to a single point (*false positive rate*, *true positive rate*). Please, notice that the results coming from their applications to *sDS*2 are equivalent. To compute the

† Please, notice that in table 10 we report the results from the logistic regression only in the case in which the macroeconomic indicators are not considered. In fact, such indicators show high linear dependence among them and, as known, the presence of multicollinearity makes unreliable the results from a regressive analysis.

Table 10. Adaptive Elman network tool for a single objective classification on the test set applied to *sDS*1 (rows labelled *No macro*) and to *sDS*2 (rows labelled *Macro*): statistics (expressed in percentage terms) over 50 repetitions of sub-sampling cross validation.

Method	Mean misbo	SD misbo	Mean misdef	SD misdef	Mean overall	SD overall
One hidden layer Elman network (No macro)	8.80	8.88	7.25	10.13	8.26	6.16
One hidden layer Elman network (Macro)	9.45	7.73	7.51	9.38	6.42	6.93
Two hidden layer Elman network (No macro)	8.27	6.40	5.75	9.18	7.40	4.33
Two hidden layer Elman network (Macro)	8.02	5.72	5.60	8.82	7.13	5.01
Logistic regression (No macro)	25.27	15.08	14.26	10.61	18.78	9.95

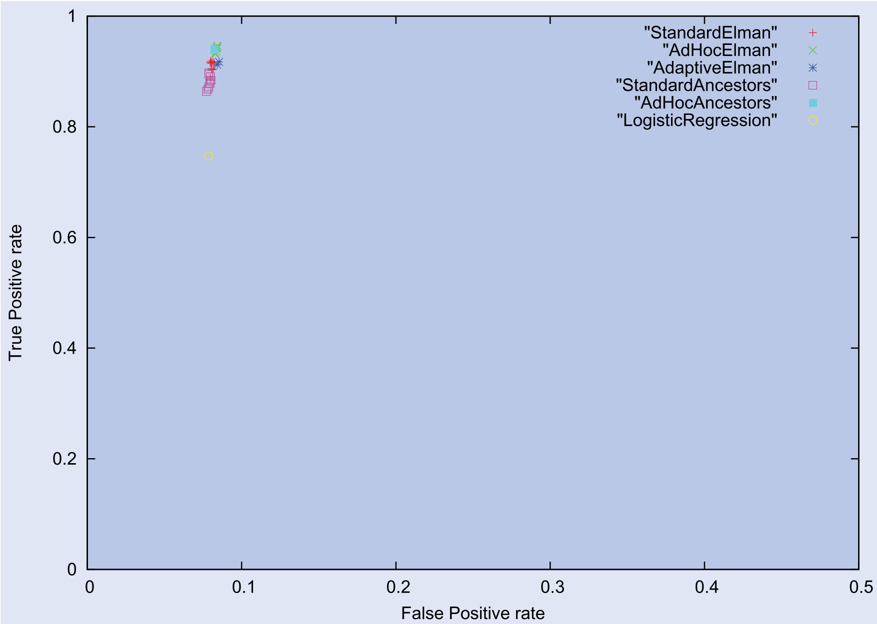

Figure 3. ROC space showing ancestors', Elman's and logistic regression's performances when applied to *sDS*1. Point $(0, 1)$ represents perfect classification.

needed quantities, mean values reported in tables 2–10 have been used. By recalling that point $(0, 1)$ represents perfect classification and that a point in ROC space is better than another if it lies in the northwest of the latter (*true positive* rate is higher, *false positive* rate is lower, or both) of the first (Fawcett 2006), we see that our neural network approaches can be preferred to logistic regression. We also remark that Elman approaches show better classification accuracy than their ancestors, and that Adaptive Elman performances are comparable to those of the other Elman approaches, with the advantage of being more general and to need lower computational times.

7.4. *Comparison with other approaches*

In this section we are assessing the robustness of our Adaptive Elman approaches, by applying them on sets of data used in other studies; we have appropriately modified the number of input nodes of our approaches in order to make them comparable with the state-of-the-art.

The first comparison is made with Zhao *et al.* (2016), which introduce a classification tool based on Kernel Extreme Learning Machines (KELM) and compare it with Support Vector Machine (SVM), Extreme Learning Machines (ELM), Random Forest (RF) and a hybrid approach obtained combining Particle Swarm Optimization and *k*-Nearest Neighbour (PSOFKNN). The set of data reports 30 financial ratios from 240 Polish firms (128 non-bankrupt and 112 bankrupt) over two years (Pietruszkiewicz 2008). 10-fold cross-validation has been employed for the analysis.

Then, in table 11 we compare the results from the single-objective adaptive Elman tool with those presented in Zhao *et al.* (2016) by using the performance measures used therein: accuracy (ACC), Type I error (i.e. misdef), Type II error (i.e. misbo), and area under the ROC curve (AUC). Please, notice that we have retrieved the macroeconomic indicators relating both to the countries and to the years considered in the papers in order to be able compare the No-macro case with the Macro one. We remark that our approach, both without and with the macroeconomic indicators as additional predictors, offers better classifications of default firms, but with higher standard deviation. Furthermore, standard deviations of misclassifications obtained on this data set are lower than the ones outlined in Sections 7.1 and 7.3, reasonably due to the bigger size of the set of data.

Finally, as we have stated in Section 1, we want also to check our approach over the class of the qualitative variables, since there is an ongoing debate about their inclusion in the set of predictors used by intelligent methods to bankruptcy prediction. To this goal, we have applied the single-objective adaptive Elman tool to the data set *Quality*

Table 11. Comparisons of the results from the Adaptive Elman approach applied to *sDS*1 and to *sDS*2 with the results in Zhao *et al.* (2016).

Method	Mean	SD	Mean	SD	Mean	SD	Mean	SD
	ACC		Type I error		Type II error		AUC	
Adaptive Elman approach (No macro)	0.8021	0.1710	0.1152	0.0892	0.2132	0.1358	0.8452	0.0739
Adaptive Elman approach (Macro)	0.7781	0.1943	0.1324	0.1232	0.2308	0.1217	0.8036	0.0130
KELM by Zhao *et al.* (2016)	0.8250	0.0473	0.1515	0.0837	0.1948	0.1192	0.8268	0.0458
SVM by Zhao *et al.* (2016)	0.7667	–	0.2232	–	0.2422	–	–	–
ELM by Zhao *et al.* (2016)	0.7250	–	0.2857	–	0.2656	–	–	–

Note: In the rows, 'No macro' means the macroeconomic indicators are not considered as additional predictors, and 'Macro' means the macroeconomic indicators are considered.

Table 12. Comparisons of the results from the Adaptive Elman approach applied to *sDS*1 and to *sDS*2 with the results in Lu *et al.* (2015).

Method	Mean	SD	Mean	SD	Mean	SD	Mean	SD
	ACC		Type I error		Type II error		AUC	
Adaptive Elman approach (No macro)	0.9158	0.0273	0.0700	0.0340	0.1018	0.1041	0.8174	0.0562
Adaptive Elman approach (Macro)	0.9401	0.0120	0.0250	0.0180	0.0821	0.0920	0.9633	0.0372
SVM by Lu *et al.* (2015)	0.5714	–	–	–	–	–	–	–
GA-SVM by Lu *et al.* (2015)	0.9841	–	–	–	–	–	–	–
PSO-SVM by Lu *et al.* (2015)	0.9441	–	–	–	–	–	–	–
SPSO-SVM by Lu *et al.* (2015)	0.9920	–	–	–	–	–	–	–

Note: In the rows, 'No macro' means the macroeconomic indicators are not considered as additional predictors, and 'Macro' means the macroeconomic indicators are considered.

Bankruptcy from the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Qualitative_Bankruptcy), that have been used in Lu *et al.* (2015). This data set includes 250 firms (143 non-default and 107 default), and for each of them it reports qualitative data about 6 attributes, whose possible values are: *positive* (P), *average* (A) and *negative* (N).

First, we have converted qualitative information into numerical values: $P = 1$, $A = 0.5$, $N = 0$; then, we have compared the results coming from our approach with those coming from the approaches proposed by Lu *et al.* (2015), i.e. SVM and hybrid algorithms that combine SVM with Genetic Algorithms (GA-SVM), with Particle Swarm Optimization (PSO-SVM) and with Switching Particle Swarm Optimization (SPSO-SVM). Accordingly to Lu *et al.* (2015), the set of data has been split into 124 training samples (71 non-default and 53 default) and 126 test samples (54 non-default and 72 default). 10-fold cross validation has been used for the analysis. Results are reported in table 12.

We remark that our approach offers slightly worse ACC than the hybrid algorithms both in the No-macro case, and in the Macro one (please, notice that only ACC is reported therein). Likely, this may be due to the specific partition into training and test set chosen by Lu *et al.* (2015). In fact, when allocating bigger portions to the training set, the ACC obtained by our approach raises. Anyway, our approach performs well in classifying defaults. Indeed, standard deviations of misclassifications on this set of data are lower than those reported in Sections 7.1 and 7.3, thus confirming a higher reliability in the evaluation when using bigger sets of data. We also remark that, concerning the use of macroeconomic indicators as additional predictors, all the results from experiments in which they are considered (i.e. the Macro case) are better than those from the experiments in which they are not considered (i.e. the No-macro case). This contradicts what presented in the previous sections, in which the use of macroeconomic indicators generally lead to worsen the results. Reasonably, this is due to the conversion of the qualitative information into numerical values. In fact, since this conversion is substantially ordinal (i.e. $P = 1$, $A = 0.5$ and $N = 0$), it is not able to capture the 'intensity' of how much positive is positive, of how much average is average, and of much negative is negative. In other terms, the cardinal dimension of the phenomenon measured by the variable is missing. This dimension, i.e. the aforementioned intensity, is recovered by the use of macroeconomic indicators that are, precisely, cardinal quantities.

7.5. *Multi objective classification*

The multi-objective Elman classification tool (Algorithm 2) can search for the best compromise between `misdef` and `misbo`, identifying all the network models that can be considered optimal with respect to both the errors (i.e. the Pareto front). The network models on the Pareto front represent Elman network topologies which are non-dominated, and the user has to choose which of them fits her/his preferences or compromise amongst criteria (di Tollo *et al.* 2011, Zemella *et al.* 2011).

Results shown in figure 4 and in table 13 refer to a single sub-sample out of 50 from *sDS*1; they convey that our tool is able to identify three topologies of Elman networks with one hidden layer on the `misdef` and `misbo` Pareto front. In particular, table 13 shows that each topology leads to the same `overall` error but the relative contribution of each error class (i.e. `misdef` and `misbo`) is different, hence the user may choose her/his favoured network topology with respect to her/his preferences.

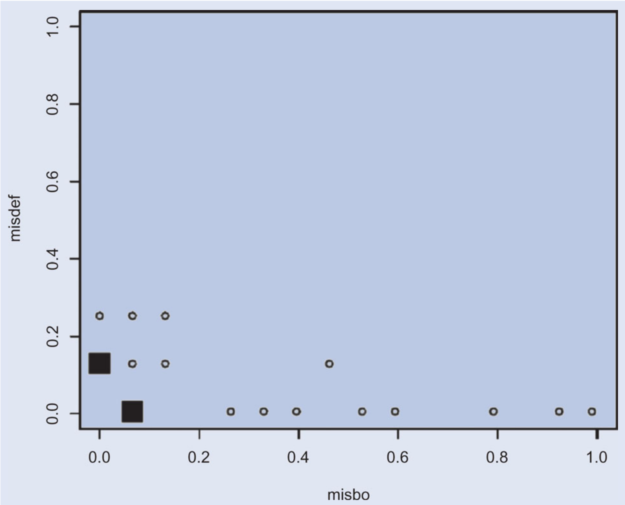

Figure 4. Elman networks on the Pareto front: example based on one test sub-sample out of 50 from *sDS*1. Black boxes identify non-dominated networks, and circles identify dominated networks. Please, notice that two of the three topologies on the Pareto front are perfectly overlapped.

Table 13. Elman networks: Pareto front from the multi-objective Elman classification tool, based on one test sub-sample out of 50 from *sDS*1.

Context neurons	misbo	misbo	misdef
2	0.00	12.50	4.43
3	0.00	12.50	4.43
16	6.60	0.00	4.43

8. Conclusions

The application of intelligent computational techniques to bankruptcy prediction and to credit risk assessment has been boosted by the introduction of the Basel agreements. Several intelligent approaches have been developed to address these issues. Among them, artificial neural networks have found a wide application due to their robustness and generalisation skills.

In this paper, we have applied a recurrent-neural-network-based approach in order to assess firms' credit risk. In particular, first we have applied recurrent networks as stand-alone tools, then we have developed an adaptive tool able to select the best recurrent network to use. Those methods have been used jointly with useful data pre-processing procedures and with a bootstrap procedure to train the network.

First, our results corroborate the main finding by Min and Lee (2005), stating that neural networks are considered to be accurate tools for credit analysis (among other things); then, they show that Elman networks can be effectively used to assess credit risk, and that they compare favourably with the state-of-the-art on this topic. Furthermore, by joining their use with an adaptive procedure able to determine their responses with respect to the data at hand, it is possible to tailor their applications to data sets having different features, always showing good performance. With respect to this point, we have developed and applied a multi-objective approach for the identification of the Pareto frontier of network models, hence providing the user with a tool which is able to represent her/his preferences with respect to different classes of risk. To the authors' knowledge, this is the first work introducing such a perspective.

Moreover, we want to stress the point that our approach compares favourably also with the state-of-the-art on the classification of *default* firms, and it is able to fine tune the network models by appropriately considering the different impacts of *type I* and *type II errors*. This consideration is of the utmost importance, as it has been shown that default firms are more difficult to detect than in-bonis firms, and because banks are more interested in correctly classifying the former than the latter.

Other interesting remarks concern the data-at-hand: our experimental analyses highlight that the use of macroeconomic indicators as additional predictors does not appear to affect the classification capabilities of our approach, thus confirming the findings of Tinoco and Wilson (2013) that point reliability problems associated with the use of these variables. This also seems to confirm the conjecture by Barboza *et al.* (2017) not to use macroeconomic indicators. They claim that their firm-based measures already produced relevant outcomes.

Further research will be devoted to develop an integrated system able to select, out of a broad set of accredited methods for credit risk assessment, the best one to apply with respect to given criteria chosen by the user.

Acknowledgements

The authors wish to thank the members of the former Institute of Cybernetics, the present Department of Software Science, of the Tallin University of Technology (Estonia) for their helpful remarks on a former draft of the paper.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Data availability statement

The data sets that support the findings of this study are available from the corresponding author, upon reasonable request.

References

Ali, A. and Daly, K., Macroeconomic determinants of credit risk: Recent evidence from a cross country study. *Int. Rev. Financ. Anal.*, 2010, **19**(3), 165–171.
Altman, E.I., Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *J. Finance*, 1968, **23**, 589–609.
Altman, E.I. and Saunders, A., Credit risk measurement: Developments over the last 20 years. *J. Bank. Financ.*, 1998, **21**, 1721–1742.

Altman, E.I., Marco, G. and Varetto, F., Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the italian experience). *J. Bank. Financ.*, 1994, **18**(3), 505–529.

Angelini, E., di Tollo, G. and Roli, A., A neural network approach for credit risk evaluation. *Quart. Rev. Econ. Finance*, 2008, **48**, 733–755.

Atiya, A.F., Bankruptcy prediction for credit risk using neural networks: A survey and new results. *IEEE Trans. Neural Netw.*, 2001, **12**(4), 929–935.

Baesens, B., Van Gestel, T., Stepanova, M. and Van den Poel, D., Neural network survival analysis for personal loan data. Technical Report, Universiteit Gent, Faculteit Economie en Bedrijfskunde, 2004.

Balcaen, S. and Ooghe, H., 35 years of studies on business failure: An overview of the classic statistical methodologies and their related problems. *Br. Account. Rev.*, 2006, **38**(1), 63–93.

Barboza, F., Kimura, H. and Altman, E., Machine learning models and bankruptcy prediction. *Expert Syst. Appl.*, 2017, **83**, 405–417.

Basel Committee on Banking Supervision, International convergence of capital measurement and capital standards a revised framework comprehensive version. Technical Report, Bank for International Settlements, 2006.

Basel Committee on Banking Supervision, Basel III: A global regulatory framework for more resilient banks and banking systems. Technical Report, Bank for International Settlements, 2010.

Beaver, W.H., Financial ratios as predictors of failure. *J. Account. Res.*, 1966, **4**, 71–111.

Bellotti, T. and Crook, J., Support vector machines for credit scoring and discovery of significant features. *Expert Syst. Appl.*, 2009, **36**(2, Part 2), 3302–3308.

Berni, R., De March, D. and Stefanini, F.M., T-optimality and neural networks: A comparison of approaches for building experimental designs. *Appl. Stoch. Models Bus. Ind.*, 2013, **29**(5), 454–467.

Bishop, C., *Neural Networks for Pattern Recognition*, 2005 (Oxford University Press: Oxford).

Boritz, J. and Kennedy, D., Effectiveness of neural networks types for prediction of business failure. *Expert Syst. Appl.*, 1995, **9**(4), 503–512.

Chen, H.L., Yang, B., Wang, G., Liu, J., Xu, X., Wang, S.J. and Liu, D.-Y., A novel bankruptcy prediction model based on an adaptive fuzzy k-nearest neighbor method. *Knowl. Based Syst.*, 2011, **24**(8), 1348–1359.

Chi, L.T. and Tang, T.C, Bankruptcy prediction: Application of Logit analysis in export credit risks. *Aust. J. Manage.*, 2006, **31**, 17–27.

Coats, P. and Fant, L., Recognizing financial distress patterns using a neural network tool. *Financ. Manage.*, 1993, **22**, 142–155.

Deb, K., *Multi-Objective Optimization Using Evolutionary Algorithms*, 2001 (John Wiley & Sons: Chichester).

Dellepiane, U., Di Marcantonio, M., Laghi, E. and Renzi, S., Bankruptcy prediction using support vector machines and feature selection during the recent financial crisis. *Int. J. Econ. Finance*, 2015, **7**(8), 182–195.

Demyanyk, Y. and Hasan, I., Financial crises and bank failures: A review of prediction methods. *Omega*, 2010, **38**(5), 315–324.

Dermine, J. and de Carvalho, L., Bank loan losses-given-default: A case study. *J. Bank. Financ.*, 2006, **30**(4), 1219–1243.

Desai, V.S., Crook, J.N. and Overstreet, G.A., A comparison of neural networks and linear scoring models in the credit union environment. *Eur. J. Oper. Res.*, 1996, **95**, 24–37.

Desai, V.S., Conway, D.G., Crook, J. and Overstreet, G., Credit scoring models in the credit union environment using neural networks and genetic algorithms. *IMA J. Math. Appl. Bus. Ind.*, 1997, **8**(4), 323–346.

di Tollo, G., Lardeux, F., Maturana, J. and Saubion, F., From adaptive to more dynamic control in evolutionary algorithms. Proceedings of the XI European Conference in Evolutionary Computation in Combinatorial Optimization, Angers, France, pp. 130–141, 2011.

Doumpos, M., Andriosopoulos, K., Galariotis, E., Makridou, G. and Zopounidis, C., Corporate failure prediction in the European energy sector: A multicriteria approach and the effect of country characteristics. *Eur. J. Oper. Res.*, 2017, **262**(1), 347–360.

Duma, I.S., Twala, B. and Marwala, T., Predictive modeling for default risk using a multilayered feedforward neural network with bayesian regularization. Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1–10, August, 2013.

Eales, R. and Bosworth, E., Severity of loss in the event of default in small business and large consumer loans. *J. Lend. Credit Risk Manage.*, 1998, **80**(9), 58–65.

Efron, B., The efficiency of logistic regression compared to normal discriminant analysis. *J. Amer. Stat. Soc.*, 1975, **70**, 892–898.

Elman, J.L., Finding structure in time. *Cogn. Sci.*, 1990, **14**, 179–211.

Fan, A. and Palaniswami, M., A new approach to corporate loan default prediction from financial statements. Proceedings of Computational Finance–Forecasting Financial Markets Conference, Calcutta, 2000.

Fawcett, T., An introduction to ROC analysis. *Pattern Recognit. Lett.*, 2006, **27**, 861–874.

Filipe, T., Grammatikos, S.F. and Michala, D., Forecasting distress in European SME portfolios. *J. Bank. Financ.*, 2016, **64**(C), 112–135.

Galindo, J. and Tamayo, P., Credit risk assessment using statistical and machine learning: Basic methodology and risk modeling applications. *Comput. Econ.*, 2000, **15**, 107–143.

Gao, X.Z., Gao, X.M. and Ovaska, S.J., A modified Elman neural network model with application to dynamical systems identification. Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Beijing, pp. 1376–1381, 1996.

Gorzalczany, M.B. and Rudzinski, F., A multi-objective genetic optimization for fast, fuzzy rule-based credit classification with balanced accuracy and interpretability. *Appl. Soft Comput.*, 2016, **40**, 206–220.

Hamid, S.A. and Iqbal, Z.S., Using neural networks for forecasting volatility of S & P 500 index futures prices. *J. Bus. Res.*, 2004, **57**, 1116–1125.

Hand, D.J. and Henley, W.E., Statistical classification methods in consumer credit scoring: A review. *J. R. Stat. Soc. Ser. A*, 1997, **160**(3), 523–541.

Haykin, S., *Neural Networks: A Comprehensive Foundation*, 1994 (Macmillan: New York).

Hongmei, L., Shaoping, W. and Pingchao, O., Fault diagnosis based on improved Elman neural network for a hydraulic servo system. 2006 IEEE Conference on Robotics, Automation and Mechatronics, Bangkok, pp. 1–6, 2006.

Huang, Z., Chen, H., Hsu, C.J., Chen, W.H. and Wu, S., Credit rating analysis with support vector machines and neural networks: A market comparative study. *Decis. Support Syst.*, 2004, **37**(4), 543–558.

Ioannidis, C., Pasiouras, F. and Zopounidis, C., Assessing bank soundness with classification techniques. *Omega*, 2010, **38**(5), 345–357.

Jin, Y. and Sendhoff, B., Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Trans. Syst Man Cybern. C: Appl. Rev.*, 2008, **38**(3), 397–415.

Jordan, M.I., Attractor dynamics and parallelism in a connectionist sequential machine. Proceedings of the Eighth Annual Conference of the Cognitive Science Society, pp. 531–546, 1986.

Jost, A., Neural networks: A logical progression in credit and marketing decision system. *Credit World*, 1993, **81**, 26–33.

Khashman, A., Neural networks for credit risk evaluation: Investigation of different neural models and learning schemes. *Expert Syst. Appl.*, 2010, **37**(9), 6233–6239.

Kim, M.J. and Han, I., The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms. *Expert Syst. Appl.*, 2003, **25**, 637–646.

King, R.D., Henery, R., Feng, C. and Sutherland, A., A comparative study of classification algorithms: Statistical, machine learning and neural networks. In *Machine Intelligence*, edited by K.

Furukawa, D. Michie, and S. Muggleton, Vol. 13, pp. 311–359, 1995 (Oxford University Press, Inc.: New York, NY).

Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P., Optimization by simulated annealing. *Science*, 1983, **220**(4598), 671–680.

Lahmiri, S. and Bekiros, S., Can machine learning approaches predict corporate bankruptcy? Evidence from a qualitative experimental design. *Quant. Finance*, 2019, **19**(9), 1569–1577.

Li, S.T., Shiue, W. and Huang, M.H., The evaluation of consumer loans using support vector machines. *Expert Syst. Appl.*, May 2006, **30**(4), 772–782.

Liang, D., Lu, C., Tsai, C. and Shih, G., Financial ratios and corporate governance indicators in bankruptcy prediction: A comprehensive study. *Eur. J. Oper. Res.*, 2016, **252**(2), 561–572.

Lili, S. and Shenoy, P.P., Using bayesian networks for bankruptcy prediction: Some methodological issues. *Eur. J. Oper. Res.*, 2017, **180**(2), 738–753.

Lim, T.S., Loh, W.Y. and Shih, Y.S., A comparison of prediction accuracy, complexity and training time of thirty-three old and new classification algorithms. *Mach. Learn.*, 2000, **40**, 203–228.

Liu, Y., Starzyk, J.A. and Zhu, Z., Optimizing number of hidden neurons in neural networks. Proceedings of the 25th Conference on Proceedings of the 25th IASTED International Multi-Conference: Artificial Intelligence and Applications, pp. 121–126, 2007.

Liu, H., Zhang, J. and Lu, C., Performance degradation prediction for a hydraulic servo system based on Elman network observer and GMM–SVR. *Appl. Math. Model.*, 2015, **39**, 5882–5895.

Loukeris, N. and Eleftheriadis, I., Corporate bankruptcy detection and portfolio selection into recurrent neural networks, hybrid neuro-genetic recurrent networks, hybrid MLPs and regressions. Proceedings of the IV World Finance Conference, 2013.

Lu, Y., Zeng, N., Liu, X. and Yi, S., A new hybrid algorithm for bankruptcy prediction using switching particle swarm optimization and support vector machines. *Discrete Dyn. Nat. Soc.*, 2015, **2015**, Article ID 294930, 7.

Martens, D., Baesens, B., Van Gestel, T. and Vanthienen, J., Comprehensible credit scoring models using rule extraction from support vector machines. *Eur. J. Oper. Res.*, 2007, **183**(3), 1466–1476.

Martens, D., Van Gestel, T., De Backer, M., Haensen, R., Vanthienen, J. and Baesens, B., Credit rating prediction using ant colony optimization. *J. Oper. Res. Soc.*, 2010, **61**(4), 561–573.

Min, J.H. and Lee, Y., Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. *Expert Syst. Appl.*, 2005, **28**(4), 603–614. ISSN 0957-4174.

Mitchell, M., *An Introduction to Genetic Algorithms*, 1996 (MIT Press: Cambridge).

Nadali, L., Corazza, M., Parpinel, F. and Pizzi, C., Recurrent ANNs for failure predictions on large datasets of Italian SMEs. In *Neural Approaches to Dynamics of Signal Exchanges*, edited by A. Esposito, M. Faundez-Zanuy, F. C. Morabito, and E. Pasero, volume 151 of *Smart Innovation, Systems and Technologies*, pp. 145–155, 2020 (Springer: Vietri sul Mare).

Noda, I., A model of recurrent networks that learns the finite automaton from given input-output sequences. Proceedings of International Symbosium on Neural Information Processing Istanbul, 1992.

Odon, M. and Sharda, R., A neural network for bankrupcy prediction. Proceedings of the International Joint Conference on Neural Networks, San Diego, CA, 1990.

Ohlson, J.A., Financial ratios and probabilistic prediction of bankruptcy. *J. Account. Res.*, 1980, **18**, 109–131.

Pacelli, V. and Azzollini, M., An artificial neural network approach for credit risk management. *J. Intell. Learn. Syst. Appl.*, 2011, **3**, 103–112.

Pang, S.L., Wang, Y.M. and Bai, Y.H., Credit scoring model based on neural network. Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing, 2002.

Pietruszkiewicz, W., Dynamical systems and nonlinear Kalman filtering applied in classification. Proceedings of 2008 7th IEEE International Conference on Cybernetic Intelligent Systems, pp. 263–268, IEEE, 2008.

Piramuthu, S., Financial credit-risk evaluation with neural and neurofuzzy systems. *Eur. J. Oper. Res.*, 1999, **112**(2), 310–321.

Quek, C., Pasquier, M. and Kumar, N., A novel recurrent neural network based prediction system for trading. Proceedings of the International Joint Conference on Neural Networks, New Delhi, pp. 2090–2097, 2006.

Ravi, V., Kurniawan, H., Thai, P.N.K. and Kumar, P.R., Soft computing system for bank performance prediction. *Appl. Soft Comput.*, 2008, **8**(1), 305–315.

Ravi Kumar, P. and Ravi, V., Bankruptcy prediction in banks and firms via statistical and intelligent techniques–a review. *Eur. J. Oper. Res.*, 2007, **180**(1), 1–28.

Rong-Zhou, L., Su-Lin, P. and Jian-Min, X., Neural network credit-risk evaluation model based on back-propagation algorithm. Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing, 2002.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J., Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, edited by D.E. Rumelhart and J. L. McClelland, Vol. 1, pp. 318–362, 1986 (MIT Press: Cambridge, MA).

Salchenberger, L., Cinar, E. and Lash, N., Neural networks: A new tool for predictiong thrift failures. *Decision Sci.*, 1992, **23**, 899–916.

Sarkar, S. and Sriram, R.S., Bayesian models for early warning of bank failures. *Manage. Sci.*, 2001, **47**(11), 1457–1475.

Shin, K.S. and Lee, Y.J., A genetic algorithm applications in bankruptcy prediction modeling. *Expert Syst. Appl.*, 2002, **23**(3), 321–328.

Shin, K., Lee, T.S. and Kim, H., An application of support vector machines in bankruptcy prediction model. *Expert Syst. Appl.*, 2005, **28**(1), 127–135.

Stagge, P. and Sendhoff, B., An extended Elman net for modeling time series. In *Artificial Neural Networks — ICANN'97*, edited by W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, volume 1327 of *Lecture Notes in Computer Science*, pp. 427–432, 1997 (Springer: Berlin Heidelberg).

Su, H.T., McAvoy, T.J. and Werbos, P., Long-term prediction of chemical processes using recurrent neural networks a parallel training approach. *Ind. Eng. Chem. Res.*, 1992, **31**, 1338–1352.

Sun, J and Li, H., Financial distress prediction using support vector machines: Ensemble vs. individual. *Appl. Soft Comput.*, 2012, **12**(8), 2254–2265.

Tam, K. and Kiang, M., Managerial applications of the neural networks: The case of bank failure predictions. *Manage. Sci.*, 1992, **38**, 926–947.

Thomas, L., Edelman, D. and Crook, J., *Credit Scoring and Its Applications*, 2002 (Society for Industrial and Applied Mathematics: Philadelphia, PA).

Tinoco, M.H. and Wilson, N., Financial distress and bankruptcy prediction among listed companies using accounting, market and macroeconomic variables. *Int. Rev. Financ. Anal.*, 2013, **30**, 394–419.

Tong, E.N.C., Mues, C. and Thomas, L.C., Mixture cure models in credit scoring: If and when borrowers default. *Eur. J. Oper. Res.*, 2012, **218**, 132–139.

Topaloglu, Z. and Yildirim, Y., Bankruptcy prediction. Working Paper, Technical Report, CUNY, Department of Economics, 2009.

Van Gestel, T., Baesens, B., Suykens, J.A.K., Van den Poel, D., Baestaens, D.E. and Willekens, M., Bayesians kernel-based classification for financial distress detection. Working Paper, Technical Report, Universiteit Gent, Faculteit Economie en Bedrijfskunde, 2004a.

Van Gestel, T., Suykens, J.A.K., Baesens, B., Viaene, S., Vanthienen, J., Dedene, G., De Moor, B. and Vandewalle, J., Benchmarking least squares support vector machine classifiers. *Mach. Learn.*, 2004b, **54**, 5–32.

Van Gestel, T., Baesens, B., Van Dijcke, P., Suykens, J.A.K., Garcia, J. and Alderweireld, T., Linear and non-linear credit scoring by combining logistic regression and support vector machines. *J. Credit Risk*, 2005, **1**(4), 31–60.

Werbos, P.J., Generalization of backpropagation with application to a recurrent gas market model. *Neural Netw.*, 1988, **1**(4), 339–356.

West, R.C., A factor-analytic approach to bank condition. *J. Bank. Financ.*, 1985, **9**, 253–266.

Wu, C. and Wang, X.M., A neural network approach for analyzing small business lending decisions. *Rev. Quant. Finance Accoun.*, 2000, **15**, 259–276.

Xiao, H., Xiao, Z. and Wang, Y., Ensemble classification based on supervised clustering for credit scoring. *Appl. Soft Comput.*, 2016, **43**, 73–86.

Yang, Z., Platt, M. and Platt, H., Probabilistic neural networks in bankrupcy prediction. *J. Bus. Res.*, 1999, **44**, 67–74.

Yobas, M.B., Crook, J.N. and Ross, P., Credit scoring using neural and evolutionary techniques. *IMA J. Manage. Math.*, 2000, **11**(2), 111–125.

Yurdakul, F., Macroeconomic modelling of credit risk for banks. *Procedia—Social Behav Sci*, 2014, **109**, 784–793.

Zell, A., Mache, N., Sommer, T. and Korb, T., The SNNS neural network simulator. In *DAGM-Symposium*, edited by B. Radig, volume 290 of *Informatik-Fachberichte*, pp. 454–461, 1991 (Springer: München).

Zemella, G., De March, D., Borrotti, M. and Poli, I., Optimised design of energy efficient building façades via evolutionary neural networks. *Energy Build.*, 2011, **43**, 3297–3302.

Zhang, J. and Morris, A.J., A sequential learning approach for single hidden layer neural networks. *Neural Netw.*, 1998, **11**(1), 65–80.

Zhang, J., Morris, A.J. and Martin, E.B., Long term prediction models based on mixed order locally recurrent neural networks. *Comput. Chem. Eng.*, 1998, **22**, 1051–1063.

Zhao, Z., Xu, S., Kang, B.H., Kabir, M.M.J., Liu, Y. and Wasinger, R., Investigation and improvement of multi-layer perceptron neural networks for credit scoring. *Expert Syst. Appl.*, 2015, **42**(7), 3508–3516.

Zhao, D., Huang, C., Wei, Y., Yu, F., Wang, M. and Chen, H., An effective computational model for bankruptcy prediction using kernel extreme learning machine approach. *Comput. Econ.*, 2016, 1–17.

Zhu, X., Li, J., Wu, D., Wang, H. and Liang, C., Balancing accuracy, complexity and interpretability in consumer credit decision making: A C-TOPSIS classification approach. *Knowl. Based Syst.*, 2013, **52**, 258–267.