

J Supercomput  
DOI 10.1007/s11227-015-1387-y

---

## Evaluation efficiency of large-scale data set with negative data: an artificial neural network approach

Mehdi Toloo · Ameneh Zandi · Ali Emrouznejad

© Springer Science+Business Media New York 2015

**Abstract** Data envelopment analysis (DEA) is the most widely used methods for measuring the efficiency and productivity of decision-making units (DMUs). The need for huge computer resources in terms of memory and CPU time in DEA is inevitable for a large-scale data set, especially with negative measures. In recent years, wide ranges of studies have been conducted in the area of artificial neural network and DEA combined methods. In this study, a supervised feed-forward neural network is proposed to evaluate the efficiency and productivity of large-scale data sets with negative values in contrast to the corresponding DEA method. Results indicate that the proposed network has some computational advantages over the corresponding DEA models; therefore, it can be considered as a useful tool for measuring the efficiency of DMUs with (large-scale) negative data.

**Keywords** Artificial neural network · Data envelopment analysis (DEA) · SORM-DEA · Negative data · Levenberg–Marquardt (LM) · LM-DEA

---

M. Toloo (✉)

Department of Business Administration, Faculty of Economics, Technical University of Ostrava, Sokolská třída 33, 701 21 Ostrava 1, Czech Republic  
e-mail: mehdi.toloo@vsb.cz; m\_toloo@yahoo.com

A. Zandi

Young Researchers and Elite Club, Central Tehran Branch, Islamic Azad University, Tehran, Iran  
e-mail: zandi\_a@yahoo.com

A. Emrouznejad

Operation and Information Management, Aston Business School, Aston University, Birmingham B4 ET, UK  
e-mail: a.emrouznejad@aston.ac.uk

## 1 Introduction

Data envelopment analysis (DEA) is a non-parametric linear programming method for assessing the efficiency and productivity of decision-making units (DMUs) which is originated by Charnes et al. [5]. DEA is widely applied to measure the efficiency of many different kinds of entities engaged in many different activities in many different contexts in many different countries. In managerial applications, DMUs may include banks, department stores and supermarkets, and extend to car makers, hospitals, schools, public libraries and so forth. In engineering, DMUs may take such forms as airplanes or their components such as jet engines [6].

In standard DEA models, it is usually assumed that all measures (inputs and outputs) are non-negative; nevertheless in some applications, negative measures are inevitable which persuade many researchers to develop techniques for measuring efficiency of DMUs (e.g. [9, 17, 21, 27]). Indeed, in most of these studies, data transformations are used to turn all negative values to positive values. The most recent development in this area is semi-oriented radial measure (SORM) method which is proposed by ([8, 9]). The method enables us to evaluate the efficiency score of DMUs with positive and negative measures. Kazemi Matin et al. [16] addressed some issues in target setting with SORM and validated their improved model by an application in bank sector. Hatami-Marbini et al. [15] dealt with interval data without sign restrictions in DEA.

Artificial neural network (ANN) is an intelligence that follows a similar process to the human brain. Human brain as a central processing unit with parallel structure approximately includes  $10^{11}$  connected neurons. Natural neurons, the simplest unit of structured nerves system, are responsible for transferring information. ANN consists of neurons which summarize input signals and can be considered as data processors. The first model of artificial neuron was created by McCulloch and Pitts [18]. Since ANNs functionality is equivalent to human brain in experiencing and extending that experience, it can be concluded that ANNs are different with computer programs that are based on explicit statements. Nowadays, ANNs are powerful tools used by various researches in solving complex problems such as approximation, pattern recognition, and clustering and have a wide range of applications in industries. For instance, a big deal of research is devoted to information fusion for early warning and situation assessment in critical infrastructure monitoring and surveillance applications (for a deeper discussion, see Flammini et al. [11]).

Recent years have seen a great variety of applying ANNs as a good method to assist in estimating efficiency score of DMUs, especially in bank industry. For example, Wu et al. [30] integrated DEA and neural networks (NNs) to examine the relative branch efficiency of a large Canadian bank. Mostafa [19] dealt with modeling the efficiency of top Arab banks with DEA and neural network (for more details, we refer the readers to [7, 10, 13, 22]). Athanassopoulos and Curram [2] firstly combined neural networks and DEA for classification and/or prediction.

This paper uses Levenberg–marquardt (LM) algorithm as faster back propagation (FBP) learning algorithm in supervised neural network to develop an algorithm for estimation of efficiency of DMUs in large-scale negative measures without using any transformation in negative data. We also show that the proposed method offers considerable computational savings.

This paper is laid out as follows. Firstly, the SORM-DEA model which considers negative data is explained in Sect. 2. Section 3 reviews neural networks and related training algorithms used in this paper. A new suggested method and designed algorithm for dealing with negative data are developed in Sect. 4. Section 5 illustrates the proposed supervised neural network for assessing the efficiency of large positive/negative data sets. Conclusion and further remarks are given in Sect. 6.

## 2 Semi-oriented radial measure (SORM)

In traditional DEA models, it is assumed that all measures (inputs and outputs) are non-negative, while in many applications this is not the case. Hence, some researchers developed various approaches to enable DEA to deal with negative data.

The most recent development in this area is the SORM which is proposed by Emrouznejad et al. [9]. The method can handle DMUs for the case when some measures can take negative values. Consider a set of  $n$  observed DMUs,  $\{DMU_j : j = 1, \dots, n\}$ , using  $m$  inputs,  $\{x_{ij} : i = 1, \dots, m\}$ , to produce  $s$  outputs,  $\{y_{rj} : r = 1, \dots, s\}$ . Banker et al. [3] formulated the following input-oriented DEA model, named BCC, to assess a DMU under variable returns to scale (VRS) assumption.

$$\begin{aligned}
 \theta_o^* &= \min \theta \\
 \text{s.t. } &\sum_{j=1}^n \lambda_j x_{ij} \leq \theta x_{io}, \quad i = 1, \dots, m \\
 &\sum_{j=1}^n \lambda_j y_{rj} \geq y_{ro}, \quad r = 1, \dots, s \\
 &\sum_{j=1}^n \lambda_j = 1 \\
 &\lambda_j \geq 0, \quad j = 1, \dots, n.
 \end{aligned} \tag{1}$$

where  $DMU_o$  for  $o \in \{1, \dots, n\}$  is the DMU being evaluated and  $0 < \theta_o^* \leq 1$  is the efficiency score of  $DMU_o$ . The model must be solved  $n$  times, each for one DMU, to obtain the efficiency score of all DMUs. Indeed, Model (1) looks for a feasible activity that guarantees at least the output level  $\mathbf{y}_o = (y_{1o}, \dots, y_{so})$  of  $DMU_o$  in all components while reducing the input vector  $\mathbf{x}_o = (x_{1o}, \dots, x_{mo})$  proportionally (radially) to a value as small as possible. If  $\theta_o^* = 1$ , then  $DMU_o$  is BCC-efficient and otherwise is BCC-inefficient. Note that we only discuss the case of VRS in this paper, since the assumption of constant returns to scale (CRS) is not possible in technologies with negative data (Portela et al. [23]).

In linear programming problem, if a variable, say  $x$ , is unrestricted in sign, then it can be replaced by  $x' - x''$  where  $x' > 0$  and  $x'' > 0$  (See Bazaraa et al. [4] page 4). Since standard DEA models are able to handle non-negative data set, the aim of SORM method is to utilize a variant of the mentioned manipulation to the standard BCC model: Let  $I$  and  $R$  be the set of indexes of non-negative inputs and outputs, respectively. Mathematically,  $I = \{i : x_i \geq 0_n, i = 1, \dots, m\}$  and  $R = \{r : y_r \geq 0_n, r = 1, \dots, s\}$ . Furthermore, let  $l$  and  $k$  be the set of indexes of an input and an

output with at least one negative data, or equivalently,  $l \in I^c$ ,  $k \in R^c$ . Emrouznejad et al. [9] substituted  $x_l$  and  $y_k$  with two non-negative values as:  $x_l^1, x_l^2$  and  $y_k^1, y_k^2$  as below:

$$\begin{aligned} x_{lj}^1 &= \begin{cases} x_{lj}, & x_{lj} > 0 \\ 0, & x_{lj} \leq 0 \end{cases}; & x_{lj}^2 &= \begin{cases} 0, & x_{lj} > 0 \\ -x_{lj}, & x_{lj} \leq 0 \end{cases} \\ y_{kj}^1 &= \begin{cases} y_{kj}, & y_{kj} > 0 \\ 0, & y_{kj} \leq 0 \end{cases}; & y_{kj}^2 &= \begin{cases} 0, & y_{kj} > 0 \\ -y_{kj}, & y_{kj} \leq 0 \end{cases} \end{aligned} \quad (2)$$

Note that  $x_{lj} = x_{lj}^1 - x_{lj}^2$  and  $y_{kj} = y_{kj}^1 - y_{kj}^2$  for all  $j$  might be negative.

The following model is proposed to measure the efficiency score of  $DMU_o$ :

$$\begin{aligned} &\min \theta \\ &\text{s.t.} \\ &\sum_{j=1}^n \lambda_j x_{ij} \leq \theta x_{io}, \quad i \in I \\ &\sum_{j=1}^n \lambda_j x_{ij}^1 \leq \theta x_{io}^1, \quad i \in I^c \\ &\sum_{j=1}^n \lambda_j x_{ij}^2 \geq \theta x_{io}^2, \quad i \in I^c \\ &\sum_{j=1}^n \lambda_j y_{rj} \geq y_{ro}, \quad r \in R \\ &\sum_{j=1}^n \lambda_j y_{rj}^1 \geq y_{ro}^1, \quad r \in R^c \\ &\sum_{j=1}^n \lambda_j y_{rj}^2 \leq y_{ro}^2, \quad r \in R^c \\ &\sum_{j=1}^n \lambda_j = 1 \\ &\lambda_j \geq 0, \quad \forall j \end{aligned} \quad (3)$$

As can be extracted, there are  $|I| + |R| + 2(|I^c| + |R^c|) + 1$  constraints and  $n + 1$  variables in model (3). In fact, with the required substitution  $|I^c| + |R^c|$ , additional constraints must be considered and hence, in a large-scale data set with negative values, more computer resources are needed to solve model (3). More precisely, to solve the SORM model (3) by the simplex method, we must firstly put it to the standard form which needs imposing  $|I| + |R| + 2(|I^c| + |R^c|)$  more slack and surplus variables. Empirically, the simplex method to solve the SORM model requires roughly on the order of  $|I| + |R| + 2(|I^c| + |R^c|) + 1$  to  $3(|I| + |R| + 2(|I^c| + |R^c|) + 1)$  iterations and in each iteration it needs  $(n + 1)(|I| + |R| + 2(|I^c| + |R^c|)) + n + 1$  multiplications and  $(n + 1)(|I| + |R| + 2(|I^c| + |R^c|))$  additions (for deeper discussion about the number of elementary computations in DEA models, see Toloo et al.

[29]). Furthermore, the model must be solved  $n$  times to obtain the efficiency score of all DMUs.

The main contribution of this study is to propose a combined algorithm using ANN for assessing the efficiency of DMUs with negative data. As will be seen subsequently, the proposed method without the need of substituting a negative data with two non-negatives deals with DEA with negative data. As a result, the designed algorithms significantly reduce the number of required computations.

### 3 Artificial neural networks

Artificial neural network is an information processing and computing system that utilizes an enormous amount of simple linking artificial nerves to simulate the capability of biological neural network (Freeman and Skapura [12]). All ANNs have at least three common components: (1) Neuron or processing unit, (2) connected weights and (3) distinct layers including neurons that are connected via weights. ANNs are also constructed with different layers: input, hidden, and output. Input layer contains the first layer of neurons, the last layer of neurons forms output layer, and the rest of the layers (if exists) are named hidden layers [25].

There are three main steps for solving problems in ANNs: training, validation, and testing. In the training step, firstly, the weights of ANNs are randomly initialized. Then, the output of the network is calculated and compared to the desired value. In sequel, the error of the network is computed and used to adjust the weights of the output layer. In a similar fashion, the network error is also propagated backward and used to update the weights of the previous layer. Validation indicates the ability of the network for presenting a valid response for new inputs that does not belong to training data set. Testing also improves the performance of the mentioned network.

There are numerous categories of approaches for training a network. One of the most conventional types of training methods for feed forward networks is Back-Propagation (BP) [26]. Several improvements for BP were developed, such as the Quick-Propagation (QP) algorithm and resilient error BP. In general, much better results can be obtained using second-order methods such as LM algorithm.

The LM Method was derived by modifying the ordinary least squares norm and is a combination of the Gauss–Newton and Steepest Descent methods. Based on the criterion of ordinary least squares, the iterative formula has the following expression (Press et al. [24]):

$$W_{k+1} = W_k + \left[ J_k^T J_k + \mu I \right]^{-1} J_k^T E(W) \quad (4)$$

where  $k$  is the number of iterations,  $E$  is the total error for all patterns,  $W$  refers to the weights,  $\mu$  is a positive scalar called learning rate,  $I$  is a diagonal matrix, and  $J$  is the sensitivity coefficient matrix defined as  $J(W) = \frac{\partial E^T(W)}{\partial W}$

The purpose of the matrix term  $\mu I$  in Eq. (4) is to damp oscillations and instabilities due to the ill-conditioned character of the problem, by making its components larger than those of  $J^T J$ , if necessary. The learning rate  $\mu$  is decreased after each successful

step (reduction in performance function) and otherwise is increased. In this way, the performance function is always reduced in each iteration of the algorithm. If the value of  $\mu$  is very large, then the LM algorithm becomes steepest decent or BP. It should be noted that if  $\mu = 0$ , then the LM algorithm is identical to the Gauss–Newton method. For a deeper discussion about the LM optimization method for neural network training, we refer the readers to Hagan and Menhaj [14] and Nelles [20].

#### 4 The proposed ANN–DEA approach

Emrouznejad and Shell [7] proposed a combined ANN and DEA algorithm to measure the efficiency of a set of large-scale data, based on an error BP training algorithm. The ANN input corresponds to measures in DEA, *i.e.*  $\{\mathbf{x}_1, \dots, \mathbf{x}_m, \mathbf{y}_1, \dots, \mathbf{y}_s\} \in R^{(m+s) \times q}$ , and the ANN output corresponds to estimated efficiency scores  $\tilde{\theta}_1, \dots, \tilde{\theta}_q$  where  $q$  is the number of utilized DMUs. Furthermore, let  $\theta_j$  and  $\tilde{\theta}_{pj}$  be the actual efficiency score (obtained by DEA models) and estimated efficiency score via  $p$ th pattern (obtained by ANN) of DMU $_j$ , respectively. Consequently, the error can be calculated as below which must be minimized:

$$e_p = \frac{1}{n} \sum_{j=1}^n (\theta_j - \tilde{\theta}_{pj})^2 \quad (5)$$

It should be noted here that initial weights in the ANN are selected randomly. In an ANN with error BP training algorithm, the input of ANN flows directly across the network to compute  $\tilde{\theta}_{pj}$ . If  $e_p \leq \varepsilon$ , then an acceptable estimation of the efficiency scores is achieved via ANN where  $\varepsilon$  is a given small enough number. Otherwise, the next pattern can be calculated by the following updated weights:

$$\Delta \mathbf{w} = -\frac{\mu \partial e_p}{\partial \mathbf{w}_p} \quad (6)$$

where  $\mathbf{w}_p$  is the weight matrix of  $p$ th pattern.

Error BP training algorithm is a gradient descent method. The main disadvantage of this algorithm is low speed of its convergence and (in some cases) the pause of learning of the network due to trapping the network in local minimum point [1]. In this method, value of learning rate is constant during the training step, thus performance of the training algorithm is very sensitive to this value. Practically, it is almost impossible to select a suitable value for the learning rate before the training step. Consequently, a trial and error approach can be applied to estimate an appropriate learning rate.

In this study, we apply LM method which has been resulted by imposing changes in BP. Indeed, we combine LM and DEA methods and create a supervised neural network to estimate the efficiency score of DMUs with negative large-scale data.

The proposed LM-DEA algorithm is summarized as follows:

```

LM-DEA algorithm
0. Initialize the weights and parameters  $\mu$  and  $0 < \beta < 1$ . (we suggest  $\mu = 0.01$ ,  $\beta = 0.1$ )
1. while terminating condition is not satisfied
2.   For each training sample of DMUs is samples
3.     compute MSE
4.     solve  $F(W) = E^T E$ .
5.     solve  $\Delta W = [J^T J + \mu I]^{-1} J^T E$ 
6.     while  $(F(W + \Delta W) \geq F(W))$ 
7.        $\mu = \frac{\mu}{\beta}$ 
8.        $\Delta W = [J^T J + \mu I]^{-1} J^T E$ 
9.        $W = W + \Delta W$ 
10.    endwhile
11.     $\mu = \mu\beta$ 
12.  endfor
13. endwhile
    
```

### 5 LM-DEA in practice

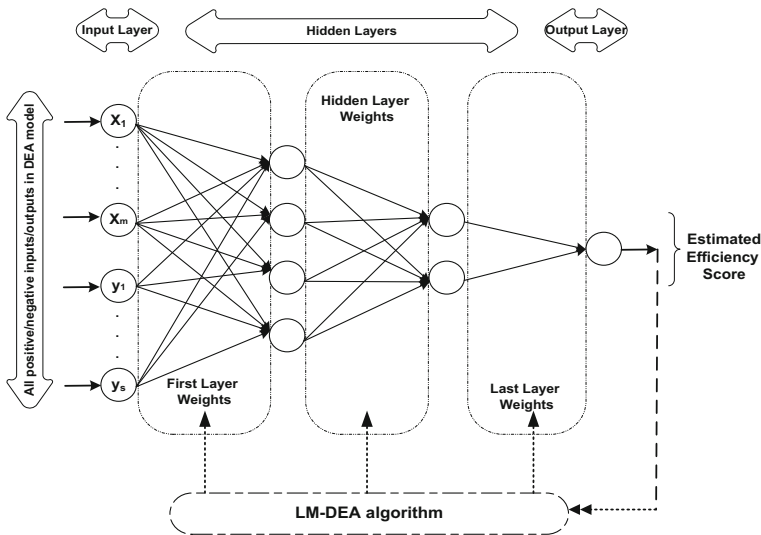
To show the applicability of the proposed LM-DEA algorithm, a data set from one of the largest private bank in Iran which involves 3,200 branches has been used. As it is exhibited in Table 1, there are 29 measures (19 inputs and 10 outputs).

Here, the resources originate from people’s deposits in the form of current accounts, savings, short-time and long-term investment accounts, absorbed by bank either as

**Table 1** Inputs and outputs measures

Input measures		Output measures
1. Area (m <sup>2</sup> )	11. Short-term Loans (IRR)	1. Profit derived from Loans (IRR)
2. Staff	12. long-term Loans (IRR)	2. Penalty derived from Loans (IRR)
3. Current account (IRR) <sup>a</sup>	13. Interest-free Loans (IRR)	3. Loans Commission (IRR)
4. Saving account (IRR)	14. Home Loans (IRR)	4. Guarantees Commission (IRR)
5. Short-term investment accounts (IRR)	15. Public sector Loans Based on Budget Notes (IRR)	5. L/Cs Commissions (IRR)
6. Long-term investment accounts (IRR)	16. Private sector Loans Based on Budget Notes (IRR)	6. Other Commissions (IRR)
7. Securities deposits (IRR)	17. Loans out of managed funds (IRR)	7. Deferred charges (IRR)
8. State current deposits (IRR)	18. Foreign Exchange Loans (USD)	8. Deferred charges based on Budget Notes (IRR)
9. Foreign currency account (USD)	19. Growth Resources (IRR)	9. Doubtful charges (IRR)
10. Other accounts (IRR)		10. Doubtful charges based on Budget Notes (IRR)

<sup>a</sup>Iran currency



**Fig. 1** Suggested topology

national or foreign exchanges. The growth resources' measure for some branches is positive, meanwhile for the others is negative and the charges are always negative. Due to the existence of negative values, the SORM model of Emrouznejad et al. [9] can be applied to measure the efficiency score of these DMUs. From the data set, we have  $I^c = \{19\}$  and  $R^c = \{7, 8, 9, 10\}$ . As a result, accordingly, 5 more constraints must be imposed on model (3) and to solve the resulting model by the simplex method for all 3,200 branches at most  $3,200 \times 105 = 336,000$  iterations with  $3,200 \times 112,035 = 358,512,000$  multiplications and  $3,200 \times 108,834 = 348,268,800$  additions are required. We should mention here again that the proposed LM-DEA algorithm evaluates the efficiency score of all branches without any changes in the data set which implies a simpler, more reliable, more succinct and more practical approach.

We suggest a topology for the LM-DEA model to measure the efficiency of DMUs with negative data which is exhibited in Fig. 1.

In this study, all neurons in each layer are connected to all other neurons of the next layer via a supervised feed-forward network. Considering the assumed topology, each neuron obtains their inputs only from the previous layer and transforms its output to the next layer.

To find the optimal network architecture, numerous combinations have been evaluated. These combinations included different number of hidden layers, different number of neurons in each layer, and different type of transfer functions. We converged to a configuration consisting of a two hidden layers that uses hyperbolic tangent sigmoid transfer function. The output layer has only one neuron with a sigmoid transfer function. This configuration has been proven to be a suitable mapper, provided that the hidden layer has enough neurons. On the one hand, if there are too few neurons, the network will not be flexible enough to model the data well and, on the other hand, if there are too many neurons, the network may over fit the data. Typically, the number



**Table 2** Details of trained LM-DEA

Item	Value	Item	Value/type
No. of data used for training	1,900	Input layer	29
No. of data used for validation	600	Hidden layer	2
No. of data used for testing	700	Output layer	1
No. of max iterations	300	Hidden layer activation functions	Tansig
No. of neurons in 1 <sup>st</sup> hidden layer	4	Output layer activation function	Logsig
No. of neurons in 2 <sup>nd</sup> hidden layer	2		

of neurons in the hidden layer is chosen by a trial and error approach, i.e. selecting a few alternatives and then running simulations to find out the one with the best results. We have considered four and two neurons for the first and second hidden layers, respectively.

Here, we emphasize again that measures in DEA with negative values are directly considered as input of the network without any modification. The input is passed layer through layer until the final output of ANN is calculated, and it is compared to the actual value of efficiency score of DMUs to find the error. The error is then propagated back to the input adjusting the weights and biases in each layer. Three subsets of available data are defined as training ( $D_{train}$ ), validation ( $D_{valid}$ ), and testing ( $D_{test}$ ) sets. The sets  $D_{train}$  and  $D_{valid}$  are used for training and selecting the best performing models. Performance of the aggregation algorithms for forecasts combination is tested by the set  $D_{test}$ .

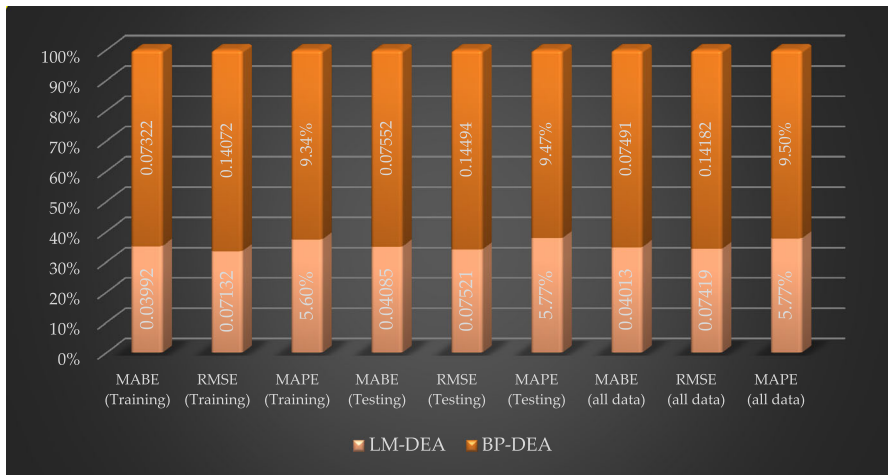
After training network and obtaining the maximum performance of simulation, the ability of network for simulation of data is evaluated. To do this, for training, validation, and testing steps we randomly considered 60, 20, and 20% of all data set, respectively. The DEA-SORM efficiency scores have been obtained using GAMS software and the LM-DEA algorithm has been implemented in MATLAB [28]. Table 2 summarizes the details of suggested network for evaluating the efficiency score with LM-DEA algorithm.

The prediction performance of the LM-DEA algorithm was evaluated by means of statistical and graphical comparisons. We assess the validity of the prediction models by statistical quantities (7)–(10):

$$MABE = \frac{1}{N} \sum_{j=1}^N \left( \left| \tilde{\theta}_{pj} - \theta_j \right| \right) \tag{7}$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{j=1}^N \left( \theta_j - \tilde{\theta}_{pj} \right)^2} \tag{8}$$

$$MAPE = \frac{100}{N} \left( \sum_{j=1}^N \left| \frac{\theta_j - \tilde{\theta}_{pj}}{\theta_j} \right| \right) \tag{9}$$



**Fig. 2** Evaluation criteria to compare LM-DEA and BP-DEA models

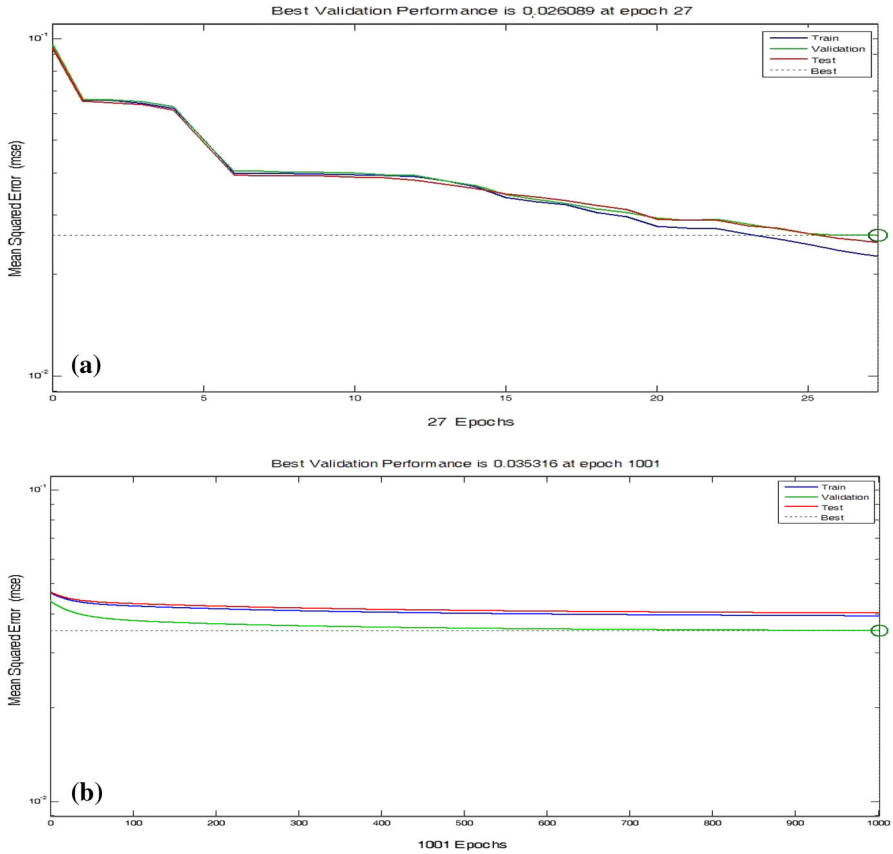
$$R = \frac{\sum_{j=1}^N (\theta_j - \bar{\theta}_j) (\tilde{\theta}_{pj} - \bar{\theta}_{pj})}{\sqrt{\sum_{j=1}^N (\theta_j - \bar{\theta}_j)^2 \sum_{j=1}^N (\tilde{\theta}_{pj} - \bar{\theta}_{pj})^2}} \quad (10)$$

where MABE is the mean absolute bias error, RMSE is the root mean square error, MAPE is the mean absolute percentage error,  $R$  is the Spearman's correlation, and  $\bar{\theta}$  is the average of predicted values. The less value for the first three statistical quantities (error) leads to a better prediction result.

The proposed LM-DEA model indicates a good performance in the prediction of the efficiency score for large-scale data sets. Figure 2 clarifies that there is a fairly small error in obtaining prediction values obtained by the proposed network. These levels of errors are quite satisfactory for predicting the efficiency score of the bank branches. Furthermore, we utilize that the performance criteria values for the same data set to evaluate the performance of the BP-DEA model in the prediction of the efficiency score are presented in Fig. 2.

Here, three statistical quantities MABE, RMSE, and MAPE are measured for three data sets training, testing, and all data (training + testing + validation). According to Fig. 2, the maximum MABE value for LM-DEA and BP-DEA is 0.04 and 0.07, respectively, which indicates that the long-term performance of the former model is better than the latter one (which is also confirmed by RMSE and MAPE values). As a result, the suggested network effectively generates sensitive results and has a sufficient accuracy and reliability rate in estimating the efficiency score of the bank branches.

Figure 3 plots the iteration processes for training, validation, testing, and best validation performance for predicting the efficiency score using LM-DEA and BP-DEA networks. The iteration process plot shows the value of the performance function (MSE) against number of epochs. Reference to Fig. 3a shows that the LM-DEA iteration is stopped when the epoch number reaches 27 and the training error (MSE)



**Fig. 3** The iteration processes of: **a** LM-DEA model and **b** BP-DEA model

of validation sample decreases to 0.026089. According to Fig. 3b, the MSE value is 0.035316 for validation sample in BP-DEA (after 1,001 epochs).

The main advantage of LM-DEA method is attributable to its fast convergence and it has good predictive effect and high accuracy. However, the BP-DEA method has a weakness which may entrap the algorithm into local minima, as is confirmed visually by Fig. 3.

To graphically compare the LM-DEA and BP-DEA algorithms, we randomly select 100 bank branches and plot their efficiency scores in Fig. 4. To have an acceptable comparison, we consider the actual efficiency score of these DMUs which are shown in blue. Reference to the figure illustrates that the LM-DEA prediction results (shown in red)<sup>1</sup> are more accurate than the BP-DEA (shown in green).

Figure 5 indicates the linear regressions between the ANN outputs and the corresponding efficiency scores for training, validation, testing and also all data sets. In this figure, Vertical and horizontal axes represent the efficiency scores simulated by LM-DEA and SORM-DEA, respectively. The dotted line indicates the region where

<sup>1</sup> Check the online version of figure

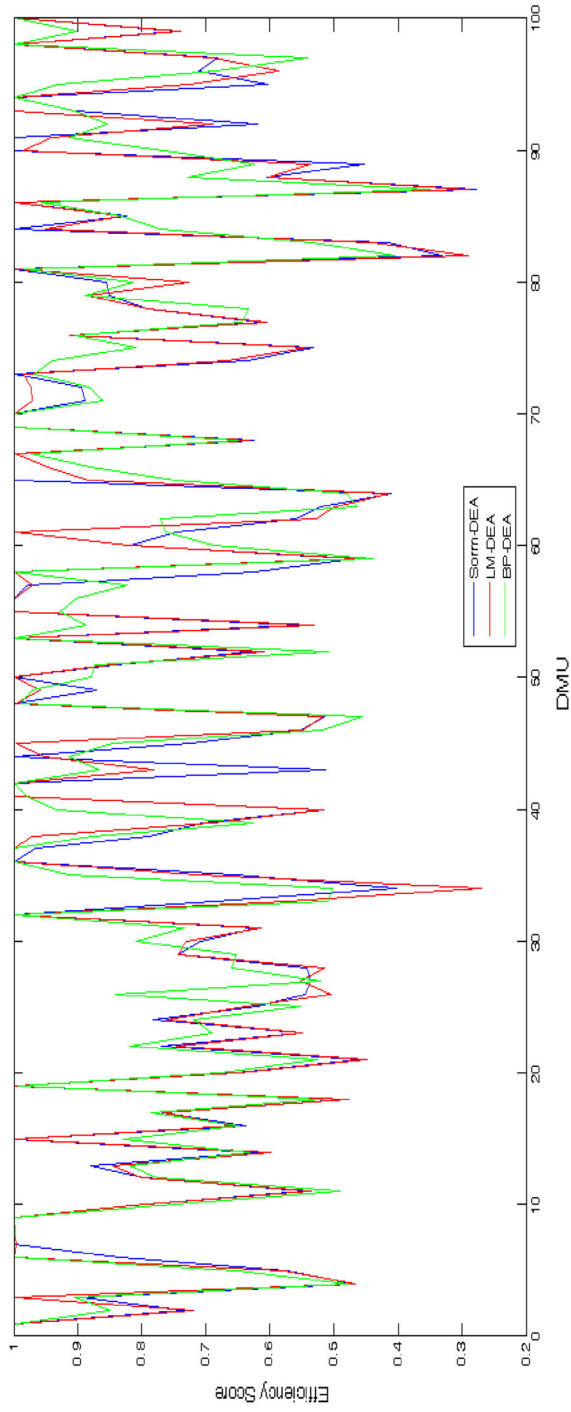
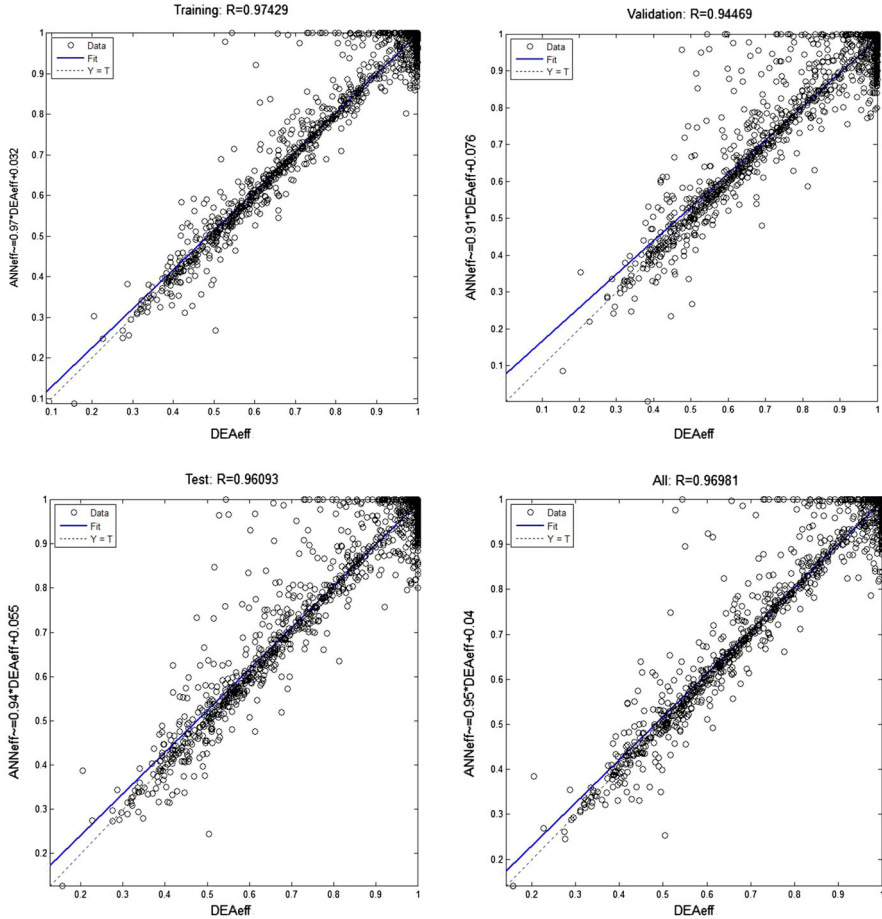
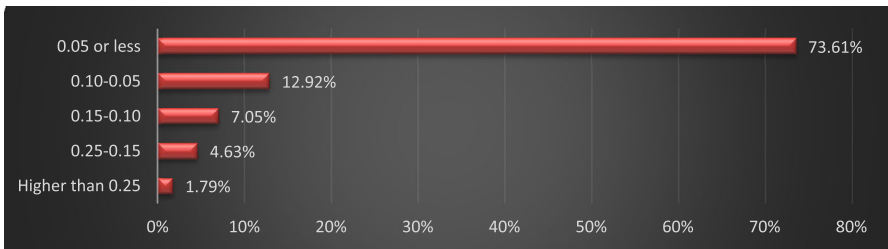


Fig. 4 Comparison of different approaches



**Fig. 5** Correlation function between LM-DEA and SORM-DEA



**Fig. 6** The distribution of error

the simulated efficiency score is exactly equal to the actual value, whereas the dashed line in blue shows their linear regression. Figure 5 indicates that the LM-DEA method significantly is succeed in predicting the efficiency scores.

An alternative method to compare LM-DEA and SORM-DEA methods is distribution analysis of MABE which is shown in Fig. 6. The figure represents the distribution analysis in five different categories which shows for 73.61 % of DMUs. The MABE is at most 0.05 and only for a small percentage of DMUs, i.e. 1.79 %, it is more than 0.25.

## 6 Conclusion

The problem of applying the conventional DEA models for measuring the efficiency score of DMUs with negative data is referenced in some studies. The main idea in most of these models is to transform the negative data to positive one which fairly increasing the number of variables and constraints. More precisely, the conducted analyses in this study showed that the SORM model needs at least  $|I| + |R| + 2(|I^c| + |R^c|) + 1$  to  $3(|I| + |R| + 2(|I^c| + |R^c|) + 1)$  iterations and in each iteration  $(n + 1)(|I| + |R| + 2(|I^c| + |R^c|)) + n + 1$  multiplications and  $(n + 1)(|I| + |R| + 2(|I^c| + |R^c|))$  additions are required. This problem is even compounded when we are dealing with large-scale data sets. In this study, we designed a new ANN-based algorithm to overcome the problem. It was shown that the proposed LM-DEA algorithm can adequately estimate the efficiency score without the need of any manipulation into the data set which is necessary in the SORM model.

The potential uses were illustrated by a real data set in banking industry involving 3,200 branches of a private bank in Iran. The obtained results clarified that the LM-DEA model has a very high correlation coefficients, i.e.  $R$ , between the predicted and the evaluated values of SORM-DEA model. The values of  $R$  were found as 0.97429 and 0.96093 in the training phase and testing phase, respectively. Moreover, the other statistical quantities validated the accuracy of the proposed ANN prediction model. Although the BP-DEA results also were satisfactory for predicting the efficiency score of bank branches, the LM-DEA model yielded closer values to the SORM-DEA.

Improving the accuracy of the proposed LM-DEA algorithm, considering different topology for ANN, and applying some hybrid approaches (such as Event Trees) can be considered as some interesting further research topics.

**Acknowledgments** The constructive comments and suggestions of the referees and Editor-in-Chief Professor Hamid R. Arabnia are highly appreciated. The research was supported by the Czech Science Foundation (GACR project 14-31593S), through European Social Fund within the project CZ.1.07/2.3.00/20.0296 and SP2014/111, an SGS project of Faculty of Economics, VŠB-Technical University of Ostrava.

## References

1. Adeloje AJ, De Munari A (2006) Artificial neural network based generalized storage -yield -reliability models using the Levenberg -Marquardt algorithm. *J Hydrol* 326:215–230
2. Athanassopoulos AD, Curram SP (1996) A comparison of data envelopment analysis and artificial neural networks as tools for assessing. *JORS* 47(8):100
3. Banker RD, Charnes A, Cooper WW (1984) Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Manag Sci* 30(9):1078–1092
4. Bazaraa MS, Jarvis JJ, Sherali HD (2010) *Linear programming and network flows*, 4th edn. Wiley, New York

5. Charnes A, Cooper WW, Rhodes E (1978) Measuring the efficiency of decision making units. *Eur J Oper Res* 2:429–444
6. Cooper WW, Seiford LM, Tone K (2008) Introduction to data envelopment analysis and its uses, 3rd edn. Springer, Berlin
7. Emrouznejad A, Shale E (2009) A combined neural network and DEA for measuring efficiency of large scale datasets. *Comput Ind Eng* 56:249–254
8. Emrouznejad A, Amin GR, Thanassoulis E, Anouze AL (2010b) On the boundedness of the SORM DEA models with negative data. *Eur J Oper Res* 206:265–268
9. Emrouznejad A, Latef Anouze A (2010a) A semi-oriented radial measure for measuring the efficiency of decision making units with negative data, using DEA. *Eu J Oper Res* 200:297–304
10. Fethi MD, Pasiouras F (2009) Assessing bank efficiency and performance with operational research and artificial intelligence techniques: a survey. *Eur J Oper Res* 204(2):189–198
11. Flammini F, Setola R, Franceschetti G (2013) Effective surveillance for homeland security balancing technology and social issues. Chapman and Hall/CRC, Boca Raton
12. Freeman JA, Skapura DM (1992) Neural network algorithms, applications, and programming techniques. Addison-Wesley, New York
13. Gutierrez E, Lozano S (2009) Data envelopment analysis of multiple response experiments. *Appl Math Model* 34(5):1139–1148
14. Hagan MT, Menhaj MB (1994) Training feed forward network with the Marquardt algorithm. *IEEE Trans Neural Netw* 5(6):989–993
15. Hatami-Marbini A, Emrouznejad A, Agrell PJ (2014) Interval data without sign restrictions in DEA. *Appl Math Model* 38:2028–2036
16. Kazemi Matin R, Amin GR, Emrouznejad A (2014) A modified semi-oriented radial measure for target setting with negative data. *Measurement* 54:152–158
17. Lovell CAK (1995) Measuring the macroeconomic performance of the Taiwanese economy. *Int J Prod Econ* 39:165–178
18. McCulloch W, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 5:115–133
19. Mostafa MM (2009) Modeling the efficiency of top Arab banks: a DEA-neural network approach. *Expert Syst Appl* 36:309–320
20. Nelles O (2001) Nonlinear system identification. Springer, Berlin, Heidelberg, New York
21. Pastor JT (1994) How to discount environmental effects in DEA: an application to bank branches. Working paper no. 011/94, Depto. De Estadística e Investigación Operativa, Universidad de Alicante, Spain
22. Pendharkar PC, Rodger JA, Yaverbaum GJ, Herman N, Benner M (1999) Association, statistical, mathematical and neural approaches for mining breast cancer patterns. *Expert Syst Appl* 17:223–232
23. Portela MCAS, Thanassoulis E, Simpson G (2004) Negative data in DEA: a directional distance approach applied to bank branches free. *J Oper Res Soc* 55:1111–1121
24. Press WH, Flannery BP, Teukolsky SA, Vetterling WT (1990) Numerical Recipes, Cambridge University Press, New York
25. Principe JC, Euliano NR (2000) Neural and adaptive systems: fundamentals through simulations. Wiley, New York
26. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. *Nature* 323:533–536
27. Seiford LM, Zhu J (2002) Modeling undesirable factors in efficiency evaluation. *Eur J Oper Res* 142:16–20
28. The MathWorks, MATLAB [Online]. <http://www.mathworks.com>
29. Toloo M, Masoumzadeh A, Barat M (2015) Finding an initial basic feasible solution for DEA models with an application on bank industry. *Comput Econ* 45:323–326
30. Wu D, Yang Z, Liang L (2006) Using DEA-neural network approach to evaluate branch efficiency of a large Canadian bank. *Expert Syst Appl* 31:108–115