Iranian Journal of Operations Research
Vol. 13, No. 1, 2022, pp. 13-30

Evaluation Efficiency of Large-Scale Data Set: Cerebellar Model Articulation Controller Neural Network

D. Modhej[۱,*], A.Dahimavi[۲]

Data Envelopment Analysis (DEA) is a nonparametric approach for evaluating the relative efficiency of a homogenous set of Decision Making Units (DMUs). To evaluate the relative efficiency of all DMUs, DEA model should be solved once for each DMU. Therefore, by increasing the number of DMUs, computational requirements are increased. The Cerebellar Model Articulation Controller (CMAC) is a neural network that resembles a part of the brain known as cerebellum. The CMAC network with a simple structure is capable of estimating nonlinear functions, system modelling and pattern recognition. Meanwhile, the CMAC approach has fast learning convergence and local generalization in comparison to other networks. The present paper is concerned with assessing the efficiency of DMUs by the CMAC neural network for the first time. The proposed approach is applied to a large set of 600 Iranian bank branches. The efficiency results are analyzed and compared with the Multi-layer Perceptrons (MLP) network outcomes. Based on the results, it can be seen that the DEA-CMAC results tend to be similar to those of DEA-MLP in terms of accuracy. In addition, the Mean Squared Error (MSE) in DEA-CMAC decreases much faster than that in DEA-MLP. The DEA-CMAC model takes 1008 and 1107 iterations to reach MSE errors of ۲.۰۳×۱۰^{-٤} *and of* ٦.۰۱×۱۰^{-٤}*, respectively, while the DEA-MLP model takes 1190 iterations keeping the MSE error stable at* ۲.۰۷×۱۰^{-۱}*. Moreover, DEA-CMAC requirements for CPU time are far less than those needed by DEA-MLP.*

***Keywords**: Data Envelopment Analysis; Cerebellar Model Articulation Controller; Neural Networks; Efficiency; Bank Branch.*

.
Manuscript was received on 12/07/2021, revised on 01/08/2022 and accepted for publication on 03/11/2022.

۱. Introduction

Efficiency measurement is an important task in management issues, and it has become an appealing research area recently. Evaluating efficiency methods can be divided into two bundles of parametric and non-parametric approaches. The parametric approach attempts to estimate the frontier production function via diverse frontiers including a combinative error. As the commonly used parametric approaches, one can adduce to the followings: Thick Frontier Approach (TFA) which was developed by Berger and Humphrey [8], Stochastic Frontier Approach (SFA) of Aigner et al. [1] and Distribution Free Approach (DFA) which was engendered by Berger [7]. Unlike the parametric approach, in the non- parametric methods no stringent functional form is imposed to determine the input-output relation. The commonly used non-parametric approaches for efficiency measurement

* Corresponding Author.
[۱]Assistant Professor, Departments of Mathematics, Sosangerd Branch, Islamic Azad University, Sosangerd, Iran, Email: modhej83@gmail.com.
[۲]Postdoctoral graduate of Water Resources, Khozestan Water and Power Organization, Email: adeldahimavi@yahoo.com.

are Data Envelopment Analysis (DEA), introduced by Charnes et al. [11] and Free Disposal Hull (FDH), proposed by Deprins et al. [14]. FDH can be regarded as a general case of DEA model because it relaxes the convexity assumption of the efficiency boundary from DEA model.

DEA is the most commonly used techniques in the performance measurement. Since the advent of the basic concept of DEA in 1978, there has been a broad spectrum of theoretical developments and applications in DEA literature.

The main goal of DEA is to evaluate the productivity and efficiency of a homogenous set of Decision Making Units (DMUs). A DMU is designated as any entity which converts common multiple resources (inputs) into common multiple outcomes (outputs). As a result, DEA could be used to gain the efficiency and productivity of hospitals, banks, airlines, university departments, schools, and manufacturers. The first innovation in DEA was presented in the CCR paper (after Charnes, Cooper, & Rhodes in 1978).Subsequently Banker et al. [5] proposed variable returns to scale version of the CCR model named the BCC (after Banker, Charnes, and Cooper) model. DEA mathematically traces a piecewise linear frontier which lies on top of the observational data set and classifies DMUs as efficient or inefficient. Those DMUs that rest on the frontier are efficient, and those that are under the envelopment surface are recognized as inefficient DMUs. To evaluate the relative efficiency of all DMUs, DEA model should be solved once for each DMU. Therefore, by increasing the number of DMUs, computational requirements are increased.

Artificial Neural Networks (ANNs) are computational algorithms which are based on the human thinking paradigm. ANNs are made up of processing units (neurons) which are linked by weighted connections. These connections motivate the estimation of non-linear models by using a training data set. Since ANNs allow modelling of nonlinear processes, they have been widely used for solving many problems such as predication, classification, regression and recognition in recent years. Various approaches have been proposed aiming at contributing to the use of ANNs in DEA.

Cerebellar Model Articulation Controller (CMAC) is a kind of neural network. The CMAC neural network simulates a special part of the human brain known as the cerebellum. At first, Albus [2] designed the CMAC network to control a robot’s arms. The most noticeable advantages of the CMAC network are its simple structure, local generalization, and the fast convergence speed. Accordingly, CMAC has been extensively used in many applications such as nonlinear function approximation (Miller et al., [23]), pattern recognition (Chen and Gu [12]), equalization (Reay [28]), and system modeling (Hagood and Mcfarland [17]). The CMAC is considered as an associative memory, type of feed forward neural network. The operation of the CMAC can be described as a look-up table (LUT) in which it quantizes and scales the input vector into discrete states to look up global address.

This paper seeks an alternative approach using CMAC network to assess the DEA efficiency scores. This is the first study introducing CMAC network in DEA methodology. With augmenting the number of DMUs, the traditional DEA methods become complicated and time-consuming. Therefore, there is great deal of motivation to exert fast algorithm such as CMAC network in DEA models with many DMUs. The research is also concerned with the comparison of applying Multi-layer Perceptrons (MLP) network and CMAC network as tools for assessing the efficiency of DMUs. The rest of the paper is outlined as follows: In the section to follow, relevant literature is reviewed. Section 3 is devoted to a brief description of the DEA technique. A detailed explanation of MLP and CMAC networks are explained in section 4. DEA-CMAC methodology for assessing the efficiency of DMUs is introduced in section5.Section 6 reports results of the application of DEA-CMAC approach to a really large set of Iranian bank branches. Finally, the paper is concluded in Section7.

۲. Literature review

Athanassopoulos and Curram [3] is the first literature idea of combination neural networks and DEA. They gained more effective results for DEA compared to ANN for classifying and predicting the efficiency of bank branches. Nevertheless, they obtained similar results for ANN to DEA in the ranking of bank branches. Wu et al. [36] combined DEA and ANN to examine the branch efficiency of a large Canadian bank. Misiunas et al. [24] proposed deploying DEA to pre-process the data to remove outliers and hence, preserve monotonicity as well as to reduce the size of the dataset used to train the ANN. The methodology is implemented with a complex, large, US-based nationwide healthcare dataset. Their results proved that the accuracy of the ANN can be maintained while the size of the training dataset is significantly reduced. Shokrollahpour et al. [32] used ANN-DEA approach to guide weaker performers on how to improve their performance to different efficiency ratings for the future. Their methodology is implemented via the branches in one of the Iranian commercial banks. Modhej et al. [25] determined the best possible values of inputs by ANN for a large number of DMUs when their output levels are changed and their efficiency values remain unchanged. The results demonstrated that the integrated ANN-inverse DEA predicts reliable input levels for DMUs. Olanrewaju et al. [27] analyzed the total energy efficiency in a Canadian industrial sector by using ANN. Their method is validated by its application to determine the efficiency computation and an analysis of historical data as well as the prediction and optimization capability of the Canadian industrial sector. A novel hybrid approach that aims to present an optimal level of input parameters of a power plant integrating DEA, ANN and inverse problem is proposed by Jahangoshai Rezaee and Dadkhah [20]. Their method is composed of three steps. First, DEA model is applied to identify and determine the days that operated efficiently. Then, the efficient vectors are entered into the ANN. Afterward, the weights and biases of the trained network are extracted for use in the inverse neural network. Bashiri et al. [6] put forward an integrated approach based on DEA and ANN to optimize a multi-response optimization problem based on the Taguchi method for the processes where controllable factors are the smaller-the-better (STB)-type variables and the analyzer desires to find an optimal solution with smaller amount of controllable factors. Emrouznejad and Shale [15] proposed ANN- DEA approach to evaluate efficiency and productivity of large data sets with many input/output variables and many DMUs. Their algorithm is applied to five large data sets and compared with results obtained by conventional DEA. The results indicated that the ANN-DEA prediction for efficiency score appears to be a good estimate for the majority of DMUs. An analysis of error shows that the larger the dataset the smaller error. Karamali et al. [21] examined the capability of ANNs in sensitivity analysis of DEA. Their methodology provided a platform for simulating the level of some parameters against the remaining parameters for generating different scenarios, as being in demand for managers. Mostafa [26] dealt with modeling the efficiency of top Arab banks with DEA and ANN. Results indicated that the predictive accuracy of ANN models is quite similar to that of traditional statistical methods. The study showed that the ANN models have a great potential for the classification of banks' relative efficiency due to their robustness and flexibility of modeling algorithms. Revuelta et al. [29] developed a hybrid prediction model whose accuracy relative to several alternative configurations has been validated through a battery of clustering techniques. Using hospital admission data from a cohort of hospitalized transplant patients, their hybrid DEA—ANN model extrapolated the progression towards severe COVID-19 disease with an accuracy of 96.3%, outperforming any competing model, such as logistic regression (65.5%) and random forest (44.8%). Zhong et al. [37] constructed a relative effective frontier through the super-efficiency SBM model, and machine learning algorithms to construct a regression model to eventually establish an absolute effective frontier. They found BPNN had the best performance, and they finally established the Super SBM-DEA-BPNN model. Their model not only provides a stable performance evaluation tool but also facilitates comparison, which has good application significance for organizations. Tsolas et al. [33] developed a two-stage hybrid model that employs ANN via integration with DEA, which is used as a preprocessor, to investigate the ability of the DEA-ANN approach to classify the sampled branches of a Greek bank into predefined efficiency classes. Their proposed modeling approach integrates the DEA context with ANN and advances benchmarking practices to enhance the decision-making process for efficiency improvement. Shahi et al. [31] used the measurement capabilities of the DEA models to train the ANN models for the best performance modeling of sawmills in Ontario. Based on their results, the trained ANN models demonstrate promising results in predicting the relative efficiency scores and the optimal combination of the inputs and the outputs for three categories (large, medium and small) of sawmills in Ontario. Koronakos and Sotiropoulos [22] employed ANNs to estimate the efficiency scores of the milestone DEA models. The ANN employed in their research estimates the DEA efficiency scores accurately. They validated their approach by conducting a series of experiments based on different data generation processes and number of inputs and outputs. The results of the experimentation showed that the proposed approach effectively approximates the CCR and the BCC efficiency scores. Costa and Markellos [13] proposed an ANN approach to measure performance of public transport services. They analyzed the London Underground efficiency with time series data and explained that the ANN approach is superior to traditionally applied techniques since it is both nonparametric and stochastic and offers greater flexibility. Azadeh et al. [4] proposed an algorithm to assess the impact of personnel efficiency attributes on total efficiency through DEA, ANN and rough set theory (RST). The integrated approach was successfully applied to 102 branches of a large private bank and the personnel attributes impact total efficiency of bank branches was evaluated.

By augmenting the number of DMUs, the traditional DEA methods become complicated and time-consuming. Therefore, using fast algorithms, such as a CMAC network, in DEA models with many DMUs will be effective. The research focuses on the comparison of applying the MLP and CMAC network tools for assessing the efficiency of DMUs.

۳. Data Envelopment Analysis

DEA is a systematic analytic tool to establish the production frontier and is also used to assign the best practical unit among a set of comparable units. Because of different characteristics of production frontiers and due to different empirical axioms, abundant DEA models have been proposed among which the CCR and BCC models are the two basic models most frequently used by the researchers. The original CCR model is conducted under the constant return to scale (CRS) assumption whereupon the change of inputs leads to the same proportional change in outputs. While the BCC model allows variable returns to scale (VRS), videlicet increase, decrease or constant returns to scale at each point on the production frontier may be hold. In this study, the BCC model is used since it includes almost all of the attributes of the CCR model. In particular, the BCC model is more flexible than the CCR model. With respect to the orientations, CCR and BCC models can be partitioned into input-oriented and output-oriented models. The input-oriented models are configured to reduce the inputs proportionally while the outputs are held constant. In contrast, output-oriented models are used to increase the outputs proportionally while the inputs are kept at their current levels.

In light of our discussion, suppose there is a set of n peer DMUs, $\{DMU_j: j = 1, \dots, n\}$ is each using m inputs to produce s outputs and also assume $\boldsymbol{X_j} = \left(x_{۱\square}, x_{۲\square}, \dots, x_{mj}\right)^T, \boldsymbol{Y_j} = \left(y_{۱\square}, y_{۲\square}, \dots, y_{sj}\right)^T$ be the input and output vectors for DMU_j, respectively, such that $\boldsymbol{X_j} \geq \boldsymbol{0}, \boldsymbol{X_j} \neq \boldsymbol{0}$ and $\boldsymbol{Y_j} \geq \boldsymbol{0}, \boldsymbol{Y_j} \neq \boldsymbol{0}$.The input-oriented BCC model can be formulated as follows:

$$
\begin{aligned}
&(P_I)\theta^* = min\theta \\
&s.t. \quad \sum_{j=١}^{n} \lambda_j x_{ij} \leq \theta x_{io}\,, \quad i = ١, \dots, m \\
&\qquad \sum_{j=١}^{n} \lambda_j y_{rj} \geq y_{ro}\,, \quad r = ١, \dots, s \\
&\qquad \sum\nolimits_{j=١}^{n} \lambda_j = ١ \\
&\qquad \lambda_j \geq ٠ \qquad\qquad j = ١, \dots, n
\end{aligned}
\tag{1}
$$

Where θ^* is the input-oriented efficiency of DMU_o and λ_j is the intensity variable for DMU_j. If $\theta^* = ١$, we say DMU_o is (at least) weakly efficient.

Similarly, the output-oriented BCC model is the following optimization problem:

$$
\begin{aligned}
&(P_O)\varphi^* = max\varphi \\
&s.t. \quad \sum_{j=١}^{n} \lambda_j x_{ij} \leq x_{io}\,, \quad i = ١, \dots, m \\
&\qquad \sum_{j=١}^{n} \lambda_j y_{rj} \geq \varphi y_{ro}\,, \quad r = ١, \dots, s \\
&\qquad \sum_{j=١}^{n} \lambda_j = ١ \\
&\qquad \lambda_j \geq ٠ \qquad\qquad j = ١, \dots, n
\end{aligned}
\tag{2}
$$

Likewise, φ^* is the output-oriented efficiency of DMU_o and λ_j is the intensity variable for DMU_j. If $\varphi^* = ١$, then DMU_o is called (at least) weakly efficient.

It is noteworthy that the selection of the input-oriented or the output-oriented model is relevant to the application as well as the characteristics of the problems.

٤. Artificial Neural Network

ANNs can learn from training samples like human brains, and they have the capability to decide based on the data from past training. ANNs can be applied to approximate complex relationships between sets of variables. Therefore, ANNs have extensive applications in many areas such as weather forecast, air traffic control, medical research, economics, and finance. In this paper MLP network and CMAC network are utilized for assessing the efficiency of DMUs in DEA models.

٤,١. Structure of MLP Neural Network

MLP is one of the most popular and widely used ANN types. MLP includes three different layers:

The input layer simultaneously receives inputs of the ANN; each input needs one node. This layer shifts the data to the linkage layer, so it is not used to perform any calculations. The output layer refers to the estimation of the network and has neurons as outputs of the ANN. One or more hidden layers with an arbitrary number of computational nodes lie between the input and the output layers. While each data set is given to the ANN, hidden layers manage the internal mapping and let the ANN to learn and generalize new data by the formerly learned data sets. Therefore choosing the number of hidden layers and the number of nodes in each hidden layer is important. It is noteworthy that a network with few hidden layers and hidden nodes inhibits identifying the structure of training patterns. On the other hand, owing to extra calculations, the large number of hidden layers and nodes in each hidden layer is associated with longer training time. In most cases, the best structure of hidden layers is determined through a trial and error process. MLP is feed-forward due to the connections between neurons which are in one direction from the input layer to the output layer without any feedback. An example of an MLP is shown in Figure 1. The illustrated MLP has three input neurons, two outputs and two hidden layers with four neurons in each.

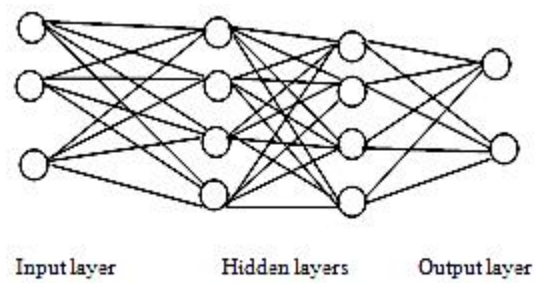

Fig. 1. The structure of MLP

There are two different modes of learning ANNs: supervised and unsupervised. For a supervised learning algorithm, the outputs are known for the given inputs. For an unsupervised learning algorithm, no outputs are specified for a set of inputs. The MLP network is utilized in a supervised manner. Back Propagation (BP) of Rumelhart et al. [30] is the most popular learning algorithm for a training MLP. Celebi and Bayraktar [9] reminded that the popularity of BP is because of its high level of accuracy and low level of complexity. BP algorithm takes training datum of input patterns and the related set of desired output patterns along with small arbitrary weights.

Following the propagation of inputs in the input layer and directly passing them through the first hidden layer, the weighted inputs are summed up in each node and the result is transferred to an activation function in order to transmit an output from the node. The consequence can be an input to the second hidden layer (if there is any), and so on. Finally, the outcome of the last hidden layer is used as the input for the output layer and the network's prediction is provided by transforming the sum of weighted inputs into an activation function. In case there is a difference between the desired output and the output produced by the network, the connection weights should be altered and adjusted so as to minimize the Mean Squared Error (*MSE*) as follows:

$$MSE = \frac{\sum_{p=1}^{P} \sum_{i=1}^{n_1} \left(y_{i-observed}^{p} - y_{i-prediction}^{p}\right)^2}{P n_1} \tag{3}$$

Where n_1 is the number of nodes in the output layer and P is the number of learning samples (input/output pairs). This error will be distributed in the backward direction from the output layer through each hidden layer down to the first hidden layer. This leads to the so-called Back-Propagation algorithm. Figure 2 illustrates a computational neuron of BP algorithm in which $x_۱, x_۲, x_۳, \ldots, x_n$are inputs of the neuron and $w_۱, w_۲, w_۳, \ldots, w_n$ respectively show their weights. Output of the neuron is then computed as:

$$y = f(\sum_{j=۱}^{n} x_i w_i) \quad (4)$$

where f is the activation function. The activation function is chosen such that it can be non-decreasing and continuously differentiable.

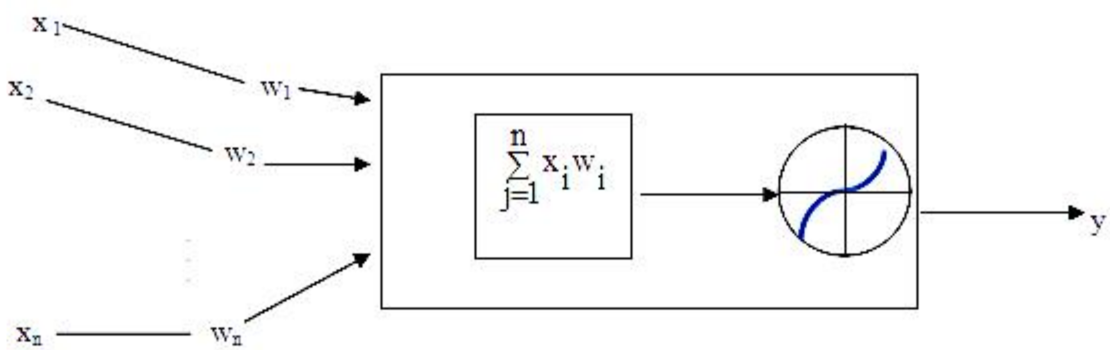

Fig. 2. Operation of neuron in BP algorithm

٤,٢. Structure of CMAC neural network

CMAC is a special neural network which models the cortex's structure of the cerebellum. CMAC-based neural networks present attractive features in comparison with MLP such as simpler calculations, fast learning capability, and better generalization ability. These privileges are because of the following:

The training of the MLP network is done using back propagation algorithm which is a global training scheme. Here, all weights of all neurons from all layers are updated at each training step. However, the CMAC network has a local training property to the effect that only the network nodes which affect the output will participate in the weight correction routine whereby the storage space requirement and the computational costs are reduced.

The CMAC network is based on associative memory which generally acts as a look- up table technique. Figure 3 shows the block diagram of the CMAC network. The input vector is quantized into discrete states considering the input states of the CMAC approach (represented by S in Figure 3).Several memory addresses (associative memories) are used to retrieve information. The output is computed by adding the retrieved data from the associated memory locations. The correct output will be accorded by regulating the contents of the memory cells.

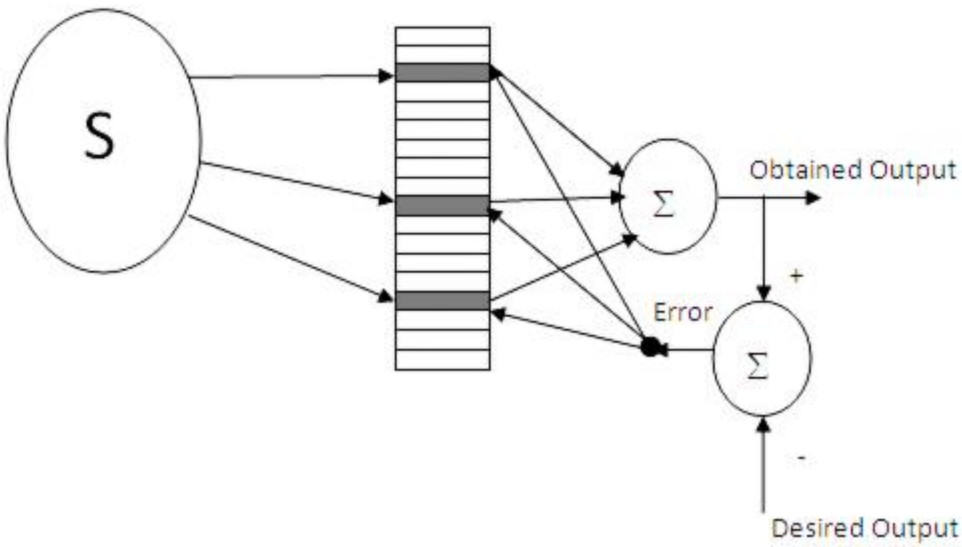

Fig. 3. Block diagram of the CMAC network

The CMAC model is a feed forward neural network which performs three subsequent mappings as follows:

$\boldsymbol{S} \rightarrow \boldsymbol{A}$: Quantized input state to association memory
$\boldsymbol{A} \rightarrow \boldsymbol{A'}$: Association memory to physical memory
$\boldsymbol{A'} \rightarrow \boldsymbol{Y}$: Physical memory to CMAC output vector

The mapping process $\boldsymbol{S} \rightarrow \boldsymbol{A}$ is nonlinear and projects the input state into an association vector$a \in \boldsymbol{A}$. In the conventional CMAC model, the association vector is binary in which the active associative cell takes one and the inactive one adopts zero. Henceforth, we refer to one and zero elements of the association vector as active and inactive positions, respectively. Let ρ bits of the association vector be active and the others be inactive. In that case, ρ is less adequate than the length of the association vector. Thereby, the computational requirements will be reduced significantly. Whereas the associative memory is extremely large to be realized in practice, it is referred to as conceptual memory.

In the next mapping($\boldsymbol{A} \rightarrow \boldsymbol{A'}$), the conceptual memory $\boldsymbol{A}$ is compressed into a reasonable size memory $\boldsymbol{A'}$ (physical space) through hash coding. Hash coding is a random compression technique which is applied for high input dimension along with a huge amount of ρ. It is performed by mapping all elements of the association vector into a smaller compressed physical one to reduce memory size.

In the last mapping$\boldsymbol{A'} \rightarrow \boldsymbol{Y}$, the output response of the CMAC is formed by summing up the adjustable stored data (weights) in the physical space, that is,

$$y(s) = \sum_{j=١}^{c} W_j \tag{5}$$

where$\boldsymbol{W} = (W_{١}, W_{٢}, \dots, W_C)$is the vector of adjustable weights in the physical space.

It is noteworthy that the similar inputs, due to common memory locations, will be mapped by the mapping $\boldsymbol{S} \rightarrow \boldsymbol{A}$ into similar binary vectors in $\boldsymbol{A}$. Hence, CMAC should produce similar outputs to nearby points in the input space. This explains the generalization capability of CMAC structure. However, distant inputs as highly disjoint memory locations will be mapped into dissimilar vectors of $\boldsymbol{A}$. Therefore, CMAC will produce nearly independent outputs for dissimilar inputs. This signifies that CMAC can be used to classify or recognize input patterns.

Figure 4 explains the layout of the two-dimensional CMAC module (2D CMAC). The input variables$X = (x_1, x_2)$ are quantized into quantization regions called blocks. These blocks are labelled as A, B and C on x_1 and a, b and c on $x_2$which compose the first layer. By shifting each block a small distance, different blocks are yielded in new layers. For instance, the blocks on the second layer are labelled as D, E, F, d, e and f and similarly G, H, I, g, h and i lie on the third layer. The area formed by these blocks in the input state such as Ac, Ab, Dd, Ef, Ed, Gg, Hi, Ig, etc. are called hypercubes. Each hypercube is a memory location that stores and retrieves the weights of the CMAC networks. Total hypercubes are the same as the association vector. In most CMAC schemes, only the blocks on the same layers can be used to create hypercubes. With this kind of hypercube composition, each state will be covered by one hypercube from each layer. An exemplar input state (2, 2) is covered by Ee, Bb and Hg (one hypercube from each layer) as shown in Figure 4. Therefore, information for this state will be distributively stored in these hypercubes.

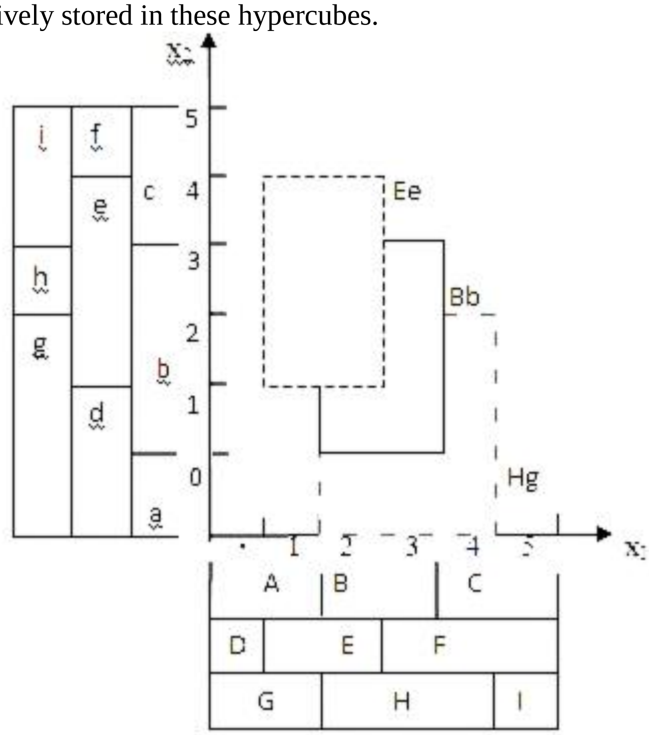

Fig.4. The memory cells of two dimensional CMAC

The CMAC communicates the hypercubes to physical memory locations. Because of the absence of memory space, hash mapping may assign several hypercubes into the same memory location.

The stored weights in hypercubes are adjusted in a supervised learning environment as follows (this is the same as saying that the CMAC is in training):

$$W_{new} = W_{old} + \frac{\alpha}{Ne} a(s)(\hat{y}(s) - y(s)) \tag{6}$$

where $\alpha$is the learning rate, $\hat{y}(s)$ is the desired network output assigned to the input s, and Ne is the number of hypercubes. By multiplying $\hat{y}(s) - y(s)$ in $a(s)$ the amount of error will be distributed only in the active cells of $a(s)$.

٥. The proposed CMAC network for DEA

In this section, DEA-CMAC network will be proposed for measuring the efficiency of DMUs in DEA models. Analogous to other DEA- networks, resources and outcomes in the corresponding DEA models are introduced as input variables of the DEA-CMAC approach. Thus, the efficiency score is nominated as the only output of DEA-CMAC algorithm. The DEA-CMAC approach goes through the following steps:

Step1: The configuration of the CMAC application (with its parameters) has to be built. The parameters include the following:
N_e: Number of layers
N_b: Number of blocks in each layer
$\boldsymbol{A}$: Associative cells or hypercubes
$\boldsymbol{A'}$: Physical address of N_e hypercubes from set $\boldsymbol{A}$
Max-Epoch: Maximum iterations
α: Learning rate

Step2: The inputs are normalized so that all inputs become at a comparable range. Using the following, the range of input values will be compressed between 0 and 1:

$$p_{ni} = (p_i - p_{min})/(p_{max} - p_{min}) \tag{7}$$

Where $p_i =$ The original value of the i^{th} component of input vector
$p_{ni} =$ The normalized value of p_i
$p_{min} =$ Minimum value among all the i^{th} component values
$p_{max} =$Maximum value among all the i^{th} component values

Step3: Data are partitioned into two data sets: training data and testing data (in the DEA-CMAC approach, we have training DMUs and testing DMUs). The training data is used to fit the CMAC parameters and to find the optimal weights. The testing data is used in order to validate the learned CMAC system.

Step4: The training data is imported into the CMAC approach.

Step5: This step tolerates quantizing the input vector, coding and allocating physical memory addresses via hash mapping.

Step6: The DEA-CMAC network output is calculated using equation (5).

Step7: The DEA-CMAC network is trained by using training data. For this, the error caused by the difference of the obtained output and the desired output is uniformly distributed in hypercubes. The weights in hypercubes are accorded to reduce the error as based on the *MSE* rule as follows:

$$MSE = \frac{\sum_{i=١}^{n}(y_i - \hat{y}_i)^{٢}}{n} \tag{8}$$

Here, n is the number of DMUs in the training sample, $y_i$and $\hat{y}_i$are estimated efficiency by DEA-CMAC network and actual efficiency, respectively. Note that since DEA-CMAC has only one output, the *MSE* in (3) and (8) are equal.

It is worth stressing that only those hypercubes that participate in output calculation are considered to update weights. The training process is stopped when assuring condition on *MSE* or following the pre-specified epochs. The final memory weights are saved into the application.

Step8: The DEA-CMAC network is tested by using testing data

Following the training phase, testing DMUs is immediately used to test the generalization and learning performance of the DEA-CMAC approach.

٦. The case study

In this section, MLP and CMAC networks are both utilized for assessing the efficiency of a big set of 600 Iranian bank branches. The problem of identification of the banking inputs and outputs is a controversy in the literature (Halkos and Salamouris, [18]; Weill, [35]; Fethi and Pasiouras, [16]; Wang et al., [34]). There is not consistency concerning the role of input and output selections due to different research objectives in banking. For instance, Isik and Hassan [19] involved labor, capital and loanable funds as input measures and short-term loans, long-term loans, off-balance-sheet items and other earning assets as output measures. Chang et al. [10] used two inputs, physical capital and labor and two outputs, total loans and other earning assets. Halkos and Salamouris [18] used five financial ratios as outputs with no input measures. They argued that all banks manage in the same market; consequently, the inputs are identical for all banks. Weill [35] selected Personnel expenses, other noninterest expenses and interest paid as inputs and loans and investment assets as outputs. Wang et al. [34] defined fixed assets and labors as inputs and Interest income, noninterest income and bad loans as outputs. In addition, deposits are considered as intermediate measures. Lastly, in order to consider the most relevant and acceptable items of banking system, which are commonly used for measuring efficiency in the literature, this study regards the following categories of inputs:
Input1 (personnel) includes personnel expenses.
Input2 (Payable interest) refers to interest expense and revenue.
Input3 (Deferred receivables) concerns to Instalments of deferred receivables and deferred payment credits.
And the following categories of outputs are considered:
Output1 (Facilities) consists of term loans, cash credit, overdraft, letters of credit, and bank guarantees.
Output2 (The total sum of four main deposits) refers to demand deposits, short-term investment deposits, long-term investment deposits, and foreign currency deposits.
Output3 (Received interest) represents earning assets into investment and interest income.
Outpu4 (Fee received) includes fee income and fee- based services.

Output5 (Other deposits) refers to other earning asset, Commercial deposits and Retail deposits.

Summaries of the statistical properties for inputs and outputs are given in Tables 1 and 2. It is worth stressing that all inputs and outputs are measured in terms of Iranian million Rials.

Table 1. Summery statistic of input values

Inputs			
	Personnel	Payable interest	Deferred receivables
Max	88.15	513160	1064400
Min	2.26	41.603	1.4824
Average	14.28	8395.6	18453
Standard deviation	10.17	25090	76816
Median	11.59	3535.7	2500.2

Table 2. Summery statistic of output values

outputs					
	Facilities	Sum of deposits	Received interest	Fee received	Other deposits
Max	8818600	10857000	875880	394980	5216200
Min	2087.4	9327.6	0.569	0.056	0.21
Average	124830	124790	10444	1105.8	10706
Standard deviation	454580	464040	45925	16153	212980
Median	47275	62697	3197.2	163.39	193.740

Organizations such as banks are engaged to provide services: they do not make any effort to reduce the inputs; rather they try to hold the inputs fixed and increase the outputs. In other words, increasing the outputs is the main priority in banking. Therefore, the selective model to evaluate banks' efficiency is the output- oriented version. At first, we employ a CMAC network as explained in section 4 to evaluate bank branches' performance. Inputs of the DEA-CMAC approach (including inputs and outputs of bank branches) are normalized, using values in the first and second rows in Tables 1 and 2. Forty percent of data (240 bank branches) are randomly chosen for training data and the residual data (360 bank branches) are prepared as the testing data. Efficiency values of bank branches in the training set are gained by model (2) using GAMS software on account of the fact that the CMAC approach is a supervised algorithm. The DEA-CMAC goes through the training process with the following parameters, which are found after some trials:

The number of layers is $N_e= 6$; the number of blocks in each layer is $N_b=7$; the learning rate is $\alpha = 0.6$; $MSE=0.01$ and the maximum iterations is equal to 2000.The training process is stopped whenever assessing accuracy condition on MSE or following the pre-specified iterations. The last trained weights of the hypercubes are stored in DEA-CMAC's memory. Followed by training, the network is tested with normalized testing data set.

In the second attempt, a MLP network (DEA-MLP) is executed to evaluate bank branches' efficiency. Details of this kind of network is pointed out by Athanassopoulos and Curram [3], Wu et al. [36], Emrouznejad and Shale [15], Fethi and Pasiouras [16]. It is noted that related literature stressed on the ability of MLP in efficiency prediction.

The DEA-MLP structure in this study is found to be 8:10:10:1, which means the number of inputs and outputs of the network are 8 and 1, respectively, and there are 10 neurons in each of the two hidden layers. Tangent hyperbolic function is used as an activation function in hidden layers and linear activation function determines the output of the neurons in the output layer. The network is trained with BP algorithm and its learning parameters are taken in accordance with the DEA-CMAC's parameters, namely the learning rate is$\alpha = ٠.٦$; *MSE=0.01* and the maximum iterations is equal to 2000.

In the Meantime, the same training and testing sets of DEA-CMAC algorithm are employed in the DEA-MLP approach. Figure 5 describes the framework of the proposed methodology.

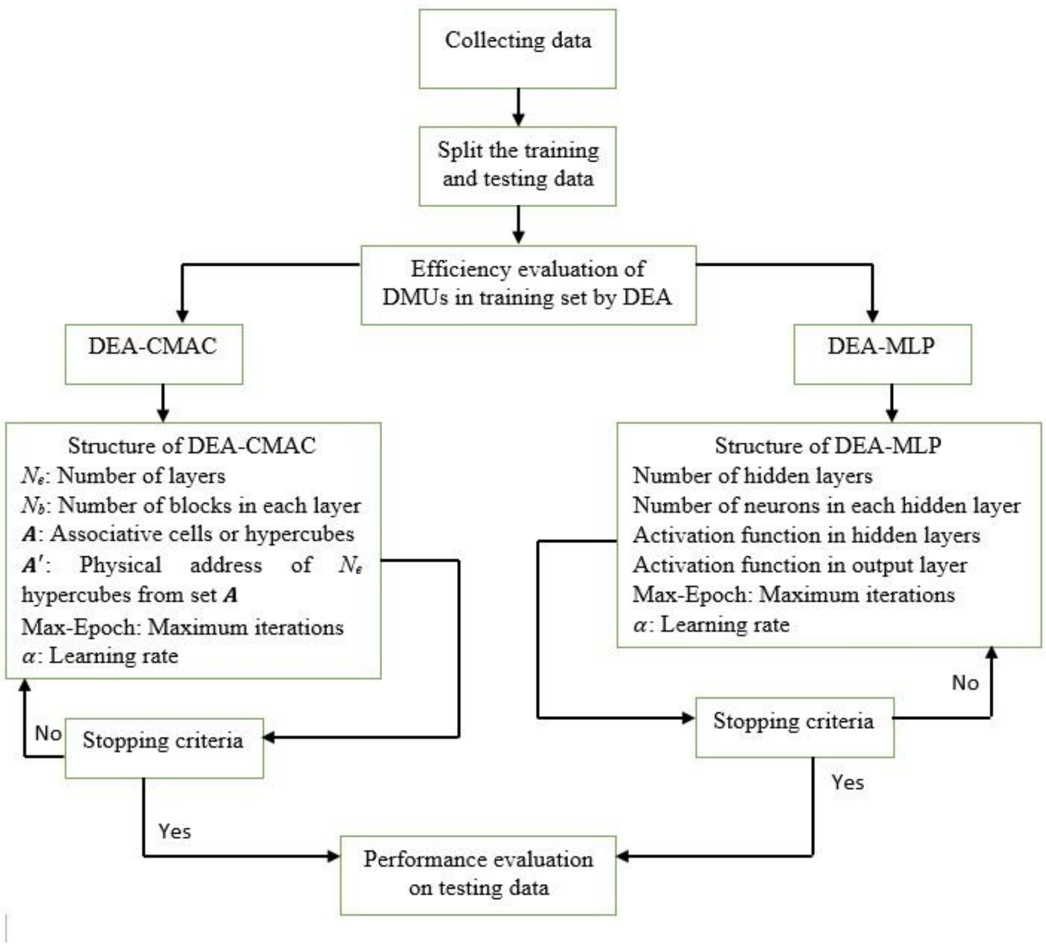

Fig.5. Framework of the proposed methodology

Ultimately, comparison of results obtained from the two approaches is analyzed through the following policies:

(1) Statistical efficiency results of DEA-CMAC and DEA-MLP are listed in Table 3.

Table 3. Statistical efficiency results corresponding to DEA-CMAC and DEA-MLP

	DEA-CMAC	DEA-MLP
Max	7.8475	7.5601
Min	1.0000	1.0000
Average	2.8394	2.8114
Standard deviation	1.2441	1.4381
Median	2.7536	2.6637

Based upon the results reported in Table 3, the difference between the max efficiency values of DEA-CMAC and DEA-MLP is 0.2874, which is the highest difference between the statistical efficiency results of DEA-CMAC and DEA-MLP. It indicates that the statistical efficiency results of DEA-CMAC are similar to those obtained through DEA-MLP.

(2) The bank branches are grouped into several categories according to their efficiency as $[١,٢), [٢,٣), [٣,٤) and [٤,\infty)$. Table 4 gives the number of branches regarding each efficiency interval of DEA-CMAC and DEA-MLP.

Table 4 .Number of branches in each efficiency interval by DEA-CMAC and DEA-MLP

	DEA-CMAC	DEA-MLP
Efficiency score interval		
$[١,٢)$	158	161
$[٢,٣)$	149	154
$[٣,٤)$	187	184
$[٤,\infty)$	106	101

From Table 4 the maximum and minimum number of branches in both DEA-CMAC and DEA-MLP are located in$[٣,٤)$ *and* $[٤,\infty)$, respectively. It can be seen that, number of branches in each efficiency interval of DEA-CMAC and DEA-MLP approaches are almost identical.

Regression analysis is carried out twice to describe the rationality of DEA-CMAC and DEA-MLP approaches. First, the regression analysis is done between the true efficiencies by normal DEA and those assessed by DEA-MLP. Subsequently, regression method is employed between DEA efficiency results and those of our current DEA-CMAC approach. Regression results are shown in Table5.

Table5. Regression analysis for efficiency prediction by DEA-CMAC and DEA-MLP

	DEA-CMAC	DEA-MLP
Slope	0.92	0.85
Intercept	0.29	0.31
R^2 coefficient	0.87	0.88

To ensure a perfect estimation, the slope would be one, the intercept would be zero and the R^2 coefficient would be one. As evident by Table 5, DEA-CMAC approach is fitted to the DEA results as well as DEA-MLP algorithm.

(4)The performance comparison of DEA-CMAC and DEA-MLP in terms of CPU time is shown in Table6.

Table6. Comparing the performance of DEA-CMAC and DEA-MLP in terms of CPU time

	DEA-CMAC	DEA-MLP
CPU time	1.078125	5.3984375

Since DEA-CMAC requirements for CPU time are far less than those which are needed by DEA-MLP it can be a practical tool in measuring efficiency especially in large data sets.

(5)Trajectories of changes in the MSE for DEA-CMAC and DEA-MLP are demonstrated in Figure 6.

As shown in Figure 6 the MSE in DEA-CMAC decreases much faster than that in DEA-MLP. The DEA-CMAC model takes 1008 and 1107 iterations to reach MSE errors of ٢.٠٣×١٠^{-٤} and of ٦.٠١×١٠^{-٤}, respectively, while the DEA-MLP model takes 1190 iterations keeping the MSE error stable at ٢.٠٧×١٠^{-١}.

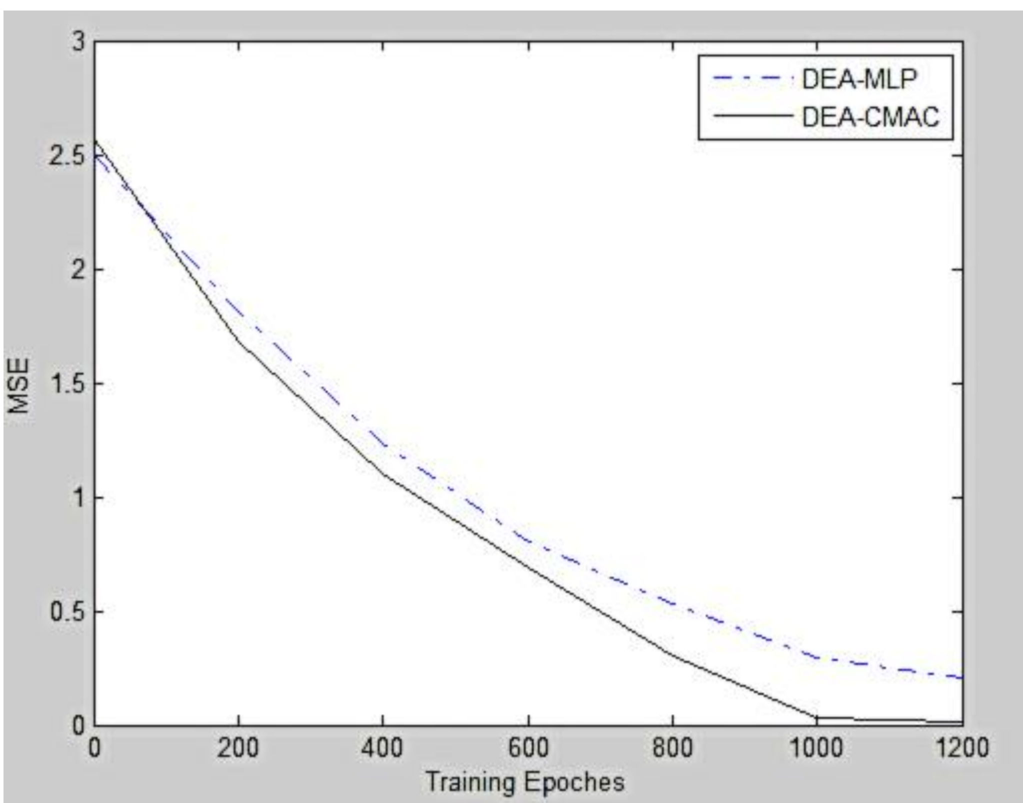

Fig.6. Trajectories of changes in the MSE in DEA-CMAC and DEA-MLP

Generally speaking, the results indicate that the proposed approach is a reliable method for efficiency evaluation together with fast convergence.

Finally, the practical implications of these findings are as follows:

- Results can be used by managers as the assessment of the bank's performance to enhance their competitiveness.
- Allow bank managements to seek banks with the best services and introduce them to investors.
- Can offer managements to analyze the underlying causes of inefficiencies.
- Allow managers to pursue the identification of practices that cause improvements in future bank's efficiency.

It is remarkable that all these can be effective for the economic growth of society.

٧. Conclusion and future works

DEA is a managerial tool for measuring the efficiency and productivity of DMUs. To evaluate the relative efficiency of all DMUs, DEA model should be solved once for each DMU. Therefore, by increasing the number of DMUs, computational requirements are increased. To overcome this drawback, diverse studies have appeared to integrate neural network approaches (including MLP, RBF, etc.) and DEA for large sets with many DMUs. This paper combines DEA with an alternative network called CMAC network for assessing the efficiency of DMUs. The advantages of using CMAC are its simplicity, generalization ability and fast learning capability. The proposed DEA-CMAC approach is used to assess the performance of a big set of Iranian banks .DEA-CMAC results tend to be similar in terms of accuracy in comparison to DEA-MLP. At the same time, it offers prominent computational saving.

Future studies can take account of ranking and benchmarking DMUs by the CMAC neural network. In addition, the efficiency evaluation of a large-scale dataset with negative data by the CMAC neural network can be addressed as an interesting topic for further research.

References

[1] Aigner, D., Lovell, C.A.K. and Schmidt, P. (1977), Formulation and Estimation of stochastic Frontier Production Function Models, *Journal of Econometrics,* 6, 21-37.

[2] Albus, J.S. (1975), A new approach to manipulator control: The cerebellar model articulation controller (CMAC), Trans. ASME, *Journal of Dynamic Systems, Measurement and Control*, 97(3), 220-227. doi:10,1115/1,3426922

[3] Athanassopoulos, A.D. and Curram, S. (1996), A comparison of data envelopment analysis and artificial neural networks as tools for assessing the efficiency of decision making units, *Journal of the Operational Research Society,* 47, 1000–1017.

[4] Azadeh, A., Saberi, M., Tavakkoli Moghaddam, R. and Javanmardi. L. (2011), An integrated data envelopment analysis–artificial neural network-rough set algorithm for assessment of personnel efficiency, *Expert Systems with Applications*, 38(3):1364–1373.

[5] Banker, R.D., Charnes, A. and Cooper, W.W. (1984), Some models for estimating technical and scale inefficiencies in data envelopment analysis, *Management Science*, 30, 1078–1092.

[6] Bashiri, M., Farshbaf-Geranmayeh, A. and Mogouie, H. (2013), A neuro-data envelopment analysis approach for optimization of uncorrelated multiple response problems with smaller the better type controllable factors, *Journal of Industrial Engineering International*, 9, 1–10. doi: 10.1186/2251-712X-9-30

[7] Berger, A.N. (1993), "Distribution-Free" Estimates of Efficiency in the U.S. Banking Industry and Tests of the Standard Distributional Assumptions, *Journal of Productivity Analysis*, ٤, ٢٦١-٢٩٢.

[8] Berger, A.N. and Humphrey, D. (1991), The Dominance of Inefficiencies over Scale and Product Mix Economies in Banking, *Journal of Monetary Economics*, 28, 117-148.

[9] Celebi, D. and Bayraktar, D. (2008), An integrated neural network and data envelopment analysis for supplier evaluation under incomplete information, *Expert Systems with Applications*, 35, 1698- 1710.

[10] Chang, T.P., Hu, J.L., Chou, R.Y. and Sun, L. (2012), The sources of bank productivity growth in China during 2002–2009: a disaggregation view, *Journal of Banking & Finance*, 36, 1997–2006.

[11] Charnes, A., Cooper, W.W. and Rhodes, E. (1978), Measuring the efficiency of decision making units, *European Journal of Operational Research*, 2,429–444.

[12] Chen, H-C, and Gu, F-C. (2012), Pattern recognition with cerebellar model articulation controller and fractal features on partial discharges, *Expert Systems with Applications*, 39(7), 6575-6584. doi:10,1016/j.eswa.2011,12,044.

[13] Costa, Á. and Markellos, R. N. (1997), Evaluating public transport efficiency with neural network models, *Transportation Research Part C: Emerging Technologies*, 5(5), 301-312.

[14] Deprins, D., Simar, L. and Tulkens, H. (1984), Measuring Labor Efficiency on Post Offices. In: M. Marchand, P. Pestieau and H. Tulkens (eds.) The Performance of Public Enterprises: Concepts and Measurements. Amsterdam, North-Holland, pp 243-267.

[15] Emrouznejad, A. and Shale, E.A. (2009), A combined neural network and DEA for measuring efficiency of large scale data sets, *Computers & Industrial Engineering*, 56, 249-254. doi:10,1016/j.cie.2008,05,012

[16] Fethi, M.D. and Pasiouras, F. (2010), Assessing bank efficiency and performance with operational research and artificial intelligence techniques: a survey, *European Journal of Operational Research*, 204,189–198. doi:10,1016/j.ejor.2009,08,003

[17] Hagood, N.W. and Mcfarland, A.J. (1995), Modelling of a piezoelectric rotary ultrasonic motor, *IEEE Transactions on Ultrasonics Ferroelectrics and Frequency Control*, 42 (2), 210-224. doi:10,1109/58,365235

[18] Halkos, G.E. and Salamouris, D.S. (2004), Efficiency measurement of the Greek commercial banks with the use of financial ratios: A data envelopment analysis approach, *Management Accounting Research*, 15,201–224.

[19] Isik, I. and Hassan, M.K. (2002), Technical, scale and allocative efficiencies of Turkish banking industry, *Journal of Banking & Finance*, 26,719–766. doi: 10.1016/s0378-4266(01)00167-4

[20] Jahangoshai Rezaee, M. and Dadkhah, M. (2019), A hybrid approach based on inverse neural network to determine optimal level of energy consumption in electrical power generation, *Computers & Industrial Engineering*, 134, 52-63. doi: 10.1016/j.cie.2019.05.024

[21] Karamali, L., Memariani, A. and Jahanshahloo, G.R. (2013), ANN-DEA Integrated Approach for Sensitivity Analysis in Efficiency Models, *Iranian Journal of Operations Research*, 4(1), 14-24.

[22] Koronakos, G. and Sotiropoulos, D. N. (2020), A Neural Network approach for Non-parametric Performance Assessment, *In 2020 11th International Conference on Information, Intelligence, Systems and Applications* (IISA (pp. 1-8). IEEE.

[23] Miller, W.T., Glanz, F.H. and Kraft, L.G. (1990), CMAC: An Associative Neural Network Alternative to Back propagation, *Proceedings of the IEEE*, 78, 1561-1567.

[24] Misiunas, N., Oztekin, A., Chen, Y. and Chandra, K. (2016), DEANN: A healthcare analytic methodology of data envelopment analysis and artificial neural networks for the prediction of organ recipient functional status, *Omega*, 58, 46–54. doi: 10,1016/j.omega.2015,03,010

[25] Modhej, D., Sanei, M., Shoja, N. and HosseinzadehLotfi, F. (2017), Integrating inverse data envelopment analysis and neural network to preserve relative efficiency values, *Journal of Intelligent & Fuzzy Systems*, 32(6), 4047-4058. doi: 10,3233/JIFS-152271

[26] Mostafa, M.M. (2009), Modeling the efficiency of top Arab banks: a DEA-neural network approach, *Expert Systems with Applications,* 36,309–320

[27] Olanrewaju, O., Jimoh, A. and Kholopan, P. (2012), Integrated IDA–ANN–DEA for assessment and optimization of energy consumption in industrial sectors, *Energy*, 46,629–35. doi: 10,1016/j.energy.2012,07,037

[28] Reay, D. (1995), Nonlinear channel equalization using associative memory neural networks. In: Proc. Int. Wkshp Appl. Neural Networks Telecom In: Proceedings of the International Workshop on Applied Neural Networks and telecommunications, Stockholm. Sweden, pp.17–24.

[29] Revuelta, I., Santos-Arteaga, F.J., Montagud-Marrahi, E. *et al.* (2021), A hybrid data envelopment analysis—artificial neural network prediction model for COVID-19 severity in transplant recipients, *Artificial Intelligence Review* 54**,** 4653–4684. doi: 10.1007/s10462-021-10008-0

[30] Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986), Learning representations by back-propagation errors, *Nature*, 323, 533-536.

[31] Shahi, S.K., Dia, M., Yan, P. and Choudhury, S. (2021), Developing and training artificial neural networks using bootstrap data envelopment analysis for best performance modeling of sawmills in Ontario, *Journal of Modelling in Management*, Vol. ahead-of-print No. ahead-of-print. https://doi.org/10.1108/JM2-07-2020-0181.

[32] Shokrollahpour, E., HosseinzadehLotfi, F. and Zandieh, M. (2016), An integrated data envelopment analysis–artificial neural network approach for benchmarking of bank branches, *Journal of Industrial Engineering International*, 12,137–143. doi: 10,1007/s40092-015-0125-7

[33] Tsolas, I.E., Charles,V.and Gherman, T. (2020) , Supporting better practice benchmarking: A DEA-ANN approach to bank branch performance assessment, *Expert Systems with Applications* , 160(1): 113599 .doi: 10.1016/j.eswa.2020.113599

[34] Wang, K., Huang, W., Wu, J. and Liu, Y.N. (2014), Efficiency measures of the Chinese commercial banking system using an additive two-stage DEA, *Omega*, 44, 5-20. doi: 10,1016/j.omega.2013,09,005

[35] Weill, L. (2004), Measuring cost efficiency in European banking: A comparison of frontier techniques, *Journal of Productivity Analysis*, 21,133–152.

[36] Wu, D., Yang, Z. and Liang, L. (2006), Using DEA-neural network approach to evaluate branch efficiency of a large Canadian bank, *Expert Systems with Applications*, 31,108–115. doi:10.1016/j.eswa.2005,09,034

[37] Zhong, K., Wang, Y., Pei, J., Tang,S. and Han, Z. (2021), Super efficiency SBM-DEA and neural network for performance evaluation, *Information Processing & Management*, 58(6),102728.