A Combined Data Envelopment Analysis and Support Vector Regression for Efficiency Evaluation of Large Decision Making Units

Mohammadreza Farahmand*, Mohammad Ishak Desa, Mehrbakhsh Nilashi

Faculty of Computing, Universiti Teknologi Malaysia, Malaysia
E-email: mmreza_farahmand@yahoo.com

Abstract Data Envelopment Analysis (DEA) is a method for measuring efficiencies of Decision Making Units (DMUs). While it has been widely used in many industrial and economic applications, for large DMUs with many inputs and outputs, DEA would require huge computer resources in terms of memory and CPU time. Several studies have attempted to overcome this problem for large datasets. However, the approaches used in the prior researches have some drawbacks which include uncontrolled convergence and non-generalization. Support Vector Regression (SVR) as a generalization from Support Vector Machine (SVM) is a powerful technique based on statistical learning theory for solving many prediction problems in the real-world applications. Hence, in this paper, a new combination of DEA and SVR, DEA-SVR, method is proposed and evaluated for large scale data sets. We evaluate and compare the proposed method using five large datasets used in earlier research. Experimental results demonstrates that the proposed method outperforms the recent most promising combined method of DEA and back-propagation neural networks, DEA-NNs, in terms of accuracy in efficiency estimation.

Keywords: Support Vector Machines, Support Vector Regression, Neural Networks, Data Envelopment Analysis, Decision Making Units.

1. Introduction

Data Envelopment Analysis (DEA) is a method for obtaining the efficiency of Decision Making Units (DMUs). While it has been widely used in industrial and economic applications, DEA for a large datasets with many inputs and outputs would require huge computer resources in terms of memory and CPU time [1]. An application of DEA with undesirable factor has been applied for evaluation of branch efficiency in the Taiwanese bank [2]. DEA approach with clustering algorithm have been proposed for Canadian bank branch network with about 1000 branches with 3 inputs and 4 outputs data, but it was not used for efficiency prediction [3]. Researchers in the previous researches showed that Neural Networks (NNs) produce better performance than statistical methods. However, the NNs method has natural drawbacks, such as local optimization solution, uncontrolled convergence, and non-generalization. This is because the performance of the NNs method is data dependent and it consumes a lot of memory and processing time to run [4]. Support Vector Machines (SVMs) are adaptive to deal with training and testing of small data [5]. The real examples show that SVM method is better than NNs and able to overcome the drawbacks of NNs [4]. The combination of DEA and NN, DEA-NN, has been used for measuring efficiency of DMUs by several researchers but the drawbacks in terms of local optimization solution, uncontrolled convergence, and non-generalization, remain [1][4][8][9]. In the DEA-NN method, it is difficult to find rules for finding the best number of hidden layers, and the network design is a trial and error process which may affect the accuracy of the resulting trained DEA-NN model. The initial values of the weights may also affect the resulting accuracy [1]. Several machine learning techniques including SVM for regression have been used to forecast the distorted demand at the end of a supply chain in [6]. The method of SVM for regression, SVR, has also been proposed to forecast the demand of beers for retailers in [7]. In addition, the method has also effectively been used for large data set problems for example in [8][9], in which DEA, Analytic Hierarchy Process (AHP), and other statistical methods have difficulties to find appropriate solution in an acceptable computational time. The hybrid manifold learning and SVM method has been proposed for the prediction of business failure, and the method achieves good general performance on many business failure prediction problems for small size of datasets [10]. A combination of SVM and Back Propagation Neural Network (BPNN) has been proposed in evaluating the enterprise financial distress in [11]. A Comparative study of supplier selection based on SVM and Radial Basic Function Neural Network (RBFNN) also showed that SVM is more superior, giving more accurate results, than RBFNN algorithm [12]. In another research by [13] who has proposed a hybrid of SVM and Rough Set Theory (RST), RST-SVM, for binary class classification indicated that RST-SVM has better performance in terms of accuracy compared to RST-NNs, the hybrid of RST and neural networks. In DMUs performance classification, DEA-SVM approach has been proposed and tested on a small DMU dataset of 23 suppliers, and also indicated that it performed better than Logistic Regression (LR), Naïve Bayes (NB) and Decision Tree (DT) approaches as well as the DEA-NNs [14].

According to the above literature, DEA has been widely used for applications with inputs and outputs; however, for large datasets with few inputs and outputs, it requires huge computer resources in terms of memory and CPU time. In addition, the approaches used in the prior researches have some drawbacks which include

uncontrolled convergence, non-generalization. Thus, in this paper, a new combination of DEA and Support Vector Regression (SVR), DEA-SVR, is proposed and tested its performance to evaluate efficiencies of large number of DMUs.

The remainder of this paper is organized as follows. Section 2 and 3 introduce the DEA and SVR method for proposed model, respectively. Section 4 provides research methodology and hybrid proposed model. Section 5 presents the experimental results and performance comparisions and finally, conclusions is presented in Section 6.

2. DEA Method

DEA is a powerful method conventionally used for measuring the efficiency of DMUs using linear programming techniques [15]. DEA requires several inputs and outputs to be considered at the same time to measure DMU efficiency which is defined as:

$$\text{Efficiency} = \frac{\text{Weighted sum of outputs}}{\text{Weighted sum of inputs}} \qquad \forall \text{ DMUs} \tag{1}$$

Assuming a set of observed DMUs $\{DMU_j | j = 1,2, \dots, n\}$ associated with m inputs $\{x_{ij} | i = 1,2, \dots, m\}$ and s outputs $\{y_{rj} | r = 1,2, \dots, s\}$.The two common DEA models, CCR and BCC models, are represented as follows [16]:

CCR Model. The CCR model is developed based on the assumption of Constant Returns to Scale (CRS) which can be applied in the production frontier with multiple input and output data. The CCR model is represented as follow [16]:

$$\min \theta$$
$$subject\ to: \begin{cases} \sum_j \lambda_j x_{ij} \le \theta x_{ip}, & \forall i \\ \sum_j \lambda_j y_{rj} \ge y_{rp}, & \forall r \\ \lambda_j \ge 0, & \forall j \end{cases} \tag{2}$$

where, θ is the efficiency for each unit p, λ_j is the dual variables for the benchmarks of inefficient units and other variables are as previously defined. Assuming θ^* as the optimum solution of the model, it can be said that if θ^* =1 then the unit is technical efficiency otherwise the unit is inefficient.

BCC Model. The BCC model, on the other hand, is based on Variable Returns to Scale (VRS). The condition considered in BCC and CCR models are the same except that the convexity condition of $\sum_j \lambda_j = 1, \lambda_j \ge 0$, $\forall j$. The BCC model is represented as follows [16]:

$$\min \theta$$
$$subject\ to: \begin{cases} \sum_j \lambda_j x_{ij} \le \theta x_{ip}, & \forall i \\ \sum_j \lambda_j y_{rj} \ge y_{rp}, & \forall r \\ \sum_j \lambda_j = 1, & \forall j \\ \lambda_j \ge 0, & \forall j \end{cases} \tag{3}$$

where, θ is the efficiency for each unit p, λ_j is the dual variables for the benchmarks of inefficient units and other variables are as previously defined.

The different between CCR and BCC is in the dual variable that is presented in the dual problem with the constraint $\sum_j \lambda_j = 1, \lambda_j \ge 0, \ \forall j$ which does not appear in the CCR model. Also, the feasible region of the BCC model is a subset for feasible region of the CCR model. Notice that θ^*_{BCC} is not less than θ^*_{CCR}, so the feasible region of BCC model is a subset for feasible region of CCR model. Some of the important parameters such as type of data, translation and returns to scale for CCR and BCC models are given in Table 1 [16]. In this table "Semi-P" is semi positive i.e. nonnegative with at least one positive element in the data for each DMU, and Free is negative, zero or positive data. In this study, CCR-I and BCC-I models with type of input-oriented have been considered due to its simplicity and using the same types of data.

Table 1 Some of DEA models based on important parameters (I, O, X and Y are input oriented, output oriented, input data and output data respectively)

Models	CCR-I	CCR-O	BCC-I	BCC-O
Data	X: Semi-P Y: Free	X: Semi-P Y: Free	X: Semi-P Y: Free	X: Free Y: Semi-P
Trans. Invariance	X: No Y: No	X: No Y: No	X: No Y: Yes	X: Yes Y: No
Returns to Scale	CRS	CRS	VRS	VRS

3. SVR method

SVM is a powerful technique based on statistical learning theory and one of the important techniques in classification and regression [17]. Compared with other classification methods, SVM does not require any parameters. Thus, SVM is sometimes called non-parametric methods. Also, SVM can support large-scale problems. SVM is a commonly used tool for solving classification problems in science and engineering but still has some limitations [18]. The SVR is the same as SVM classification except that the output of SVR is a real number and the output of SVM classification is a finite integer number [19][20]. In SVR, a margin of tolerance ε is set in approximation to the SVM which have already requested from the problem [21]. However, the main idea of SVR is to minimize the error and finding the hyperplane which maximizes the margin, keeping in mind that the error is tolerated (see Figure 1).

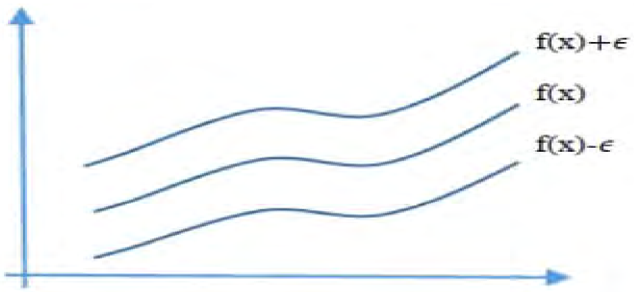


Figure 1 Prediction function with tolerance ε

a) Primal objective function, ν- SVR:

There are two SVR models for non-separable data points, namely ε-SVR and ν-SVR. In this paper, we used ν-SVR model due to its higher prediction accuracy in relation to the ε-SVR. The parameter ε of the ε -insensitive loss is useful if the desired accuracy of the approximation can be used beforehand. In the other hand, we just want the prediction to be as accurate as possible, without having to commit ourselves to a specific level of accuracy a priori. We used a modification of the ε-SVR algorithm, is called ν-SVR, which automatically computes ε [23][24]. To estimate the prediction function as in Eq. (9) from empirical data, at each point x_i, and an error ε, everything above ε is captured in slack variables ξ_i and ξ_i^* which are penalized in the objective function via a regularization constant C, chosen a priori. The size of ε is traded off against model complexity and slack variables via a constant $\nu \geq 0$ [21].

Suppose $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k) \in \mathcal{H} \times \mathbb{R}$ here $\mathcal{H}$ is a dot product space and ν-SVR as the following [21] :

$$\min_{w,b} \frac{1}{2} w^t . w + C\nu\varepsilon + C.\frac{1}{k}\sum_{i=1}^{k}(\xi_i + {\xi_i}^*)$$
$$s.t \begin{cases} y_i - (w^t.\phi(x_i) + b) \leq \varepsilon + \xi_i \\ (w^t.\phi(x_i) + b) - y_i \leq \varepsilon + {\xi_i}^* \\ \xi_i, {\xi_i}^* \geq 0, \quad \varepsilon > 0, \; i = 1,2,\ldots,k \end{cases} \tag{4}$$

where $C > 0$, $\nu \geq 0$ $\varphi(x_i)$ maps x_i into a higher dimensional space, w is the margin and (x_i, y_i) is belong to the training set.

b) Dual problem, ν- SVR:

The Lagrange function for model (4) is given by:

$$L = \frac{1}{2} w^t . w + Cv\varepsilon + \frac{C}{k}\sum_{i=1}^{k}(\xi_i + \xi_i^*) - \beta\varepsilon - \sum_{i=1}^{k}\eta_i\xi_i - \sum_{i=1}^{k}\eta_i^*\xi_i^* - \sum_{i=1}^{k}\alpha_i(\varepsilon + \xi_i + y_i - w^t.\phi(x_i) - b) - \sum_{i=1}^{k}\alpha_i^*(\varepsilon + \xi_i^* - y_i + w^t.\phi(x_i) + b) \quad (5)$$

where $\alpha_i, \alpha_i^* \eta_i, \eta_i^*$ and β are multipliers.

Now, we have to obtain the saddle point of L, as follows:

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{k}(\alpha_i - \alpha_i^*) = 0$$

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{k}\phi(x_i)(\alpha_i^* - \alpha_i) = 0 \Rightarrow w = \sum_{i=1}^{k}\phi(x_i)(\alpha_i^* - \alpha_i)$$

$$\frac{\partial L}{\partial \xi_i} = \frac{C}{k} - \eta_i - \alpha_i = 0 \Rightarrow \eta_i = \frac{C}{k} - \alpha_i, \qquad \alpha_i \in [0, \frac{C}{k}] \quad (6)$$

$$\frac{\partial L}{\partial \xi_i^*} = \frac{C}{k} - \eta_i^* - \alpha_i^* = 0 \Rightarrow \eta_i^* = \frac{C}{k} - \alpha_i^*, \qquad \alpha_i^* \in [0, \frac{C}{k}]$$

$$\frac{\partial L}{\partial \varepsilon} = Cv - \sum_{i=1}^{k}(\alpha_i^* + \alpha_i) - \beta = 0 \quad (7)$$

Substituting the above derivations in L to obtain dual optimization problem of ν-SVR model:

$$\begin{cases} \max_{\alpha} \; -\frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k}\phi(x_i)\phi(x_j)(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) - \sum_{i=1}^{k} y_i(\alpha_i - \alpha_i^*) \\ \text{subject to}: \sum_{i=1}^{k}(\alpha_i - \alpha_i^*) = 0 \text{ and } \sum_{i=1}^{k}(\alpha_i^* + \alpha_i) \le C\nu, \alpha_i, \alpha_i^* \in [0, \frac{C}{k}] \end{cases} \quad (8)$$

And the prediction function is given as,

$$f(x) = \sum_{i=1}^{k}(\alpha_i - \alpha_i^*)\; K(x, x_i) + b \quad (9)$$

where,

$$K(x_i, x_j) = \phi(x_i)\phi(x_j) = Q_{ij} \quad (10)$$

K is a kernel function and Q is called kernel matrix and it is used in the SVR algorithm. In this paper, the following RBF kernel function is used:

$$K(x_i, x_j) = e^{-\gamma(x_i - x_j)^2} \quad (11)$$

where, γ is the important parameter to increase accuracy in SVM algorithm.

3.1 Performance measure

We used Mean Squared Error (MSE) and Squared Correlation Coefficient (SCC) to measure the performance of DEA-SVR [22] [25]:

$$MSE = \frac{1}{\bar{k}}\sum_{i=1}^{\bar{k}}(f(x_i) - y_i)^2 \quad (12)$$

$$SCC = \frac{\left(\bar{k}\sum_{i=1}^{\bar{k}} f(x_i)\, y_i - \sum_{i=1}^{\bar{k}} f(x_i) \sum_{i=1}^{\bar{k}} y_i\right)^2}{\left(\bar{k}\sum_{i=1}^{\bar{k}} f^2(x_i) - \left(\sum_{i=1}^{\bar{k}} f(x_i)\right)^2\right)\left(\bar{k}\sum_{i=1}^{\bar{k}} y_i^2 - \left(\sum_{i=1}^{\bar{k}} y_i\right)^2\right)} \tag{13}$$

where y_i and $f(x_i)$ are the actual and the prediction scores, respectively.

4. The Proposed Combined DEA and SVR method

In this section, the proposed combination of DEA and SVR, DEA-SVR, method for DMU's efficiency evaluation is presented. In this proposed method, we are testing the two common methods of DEA, namely CCR and BCC, and combined them individually with SVR. In other word, the DEA-SVR method studied here consist of CCR-SVR and BCC-SVR methods.

4.1 The general steps in implementing the DEA-SVR method

Figure. 2 shows the general flow of steps in implementing the DEA-SVR method for DMUs' efficiency evaluation and prediction. The general steps are as follows:

Step 1: Get the input data= $(X_1,X_2,\ldots,X_k,Y_1,Y_2,\ldots,Y_k)$ = [DMU Inputs|DMU Outputs]. The method requires a set of DMUs, each with set of DMU's input and output.

Step 2: Calculate efficiency of units using DEA with CCR and BCC models (Obtained θ_{CCR} and θ_{BCC}).

Step 3: NIO=Normalized [DMU Inputs | DMU Outputs].

Here, we used Normalize(x)=(x-min x)/(max x- min x) for normalization.

Step 4: X_{CCR}=[NIO | θ_{CCR}] for CCR-SVR, and X_{BCC}= [NIO | θ_{BCC}] for BCC-SVR.

Step 5: $XSVR_{CCR}$=[feature column number: X_{CCR}], and $XSVR_{BCC}$ = [features column number: X_{BCC}].

The number of features = the number of DMU inputs + the number of DMU outputs.

Step 6: (Feature set) FS_{CCR}= [θ_{CCR} | $XSVR_{CCR}$], and FS_{BCC}= [θ_{BCC} | $XSVR_{BCC}$].

Step 7: Select the best parameters C, γ, v and the best kernel function for SVR model based on n run programming by Do k-Fold cross validation (obtain MSE average for each run and select the best solution).

Step 8: If the results are satisfactory then stop, else go to step 7.

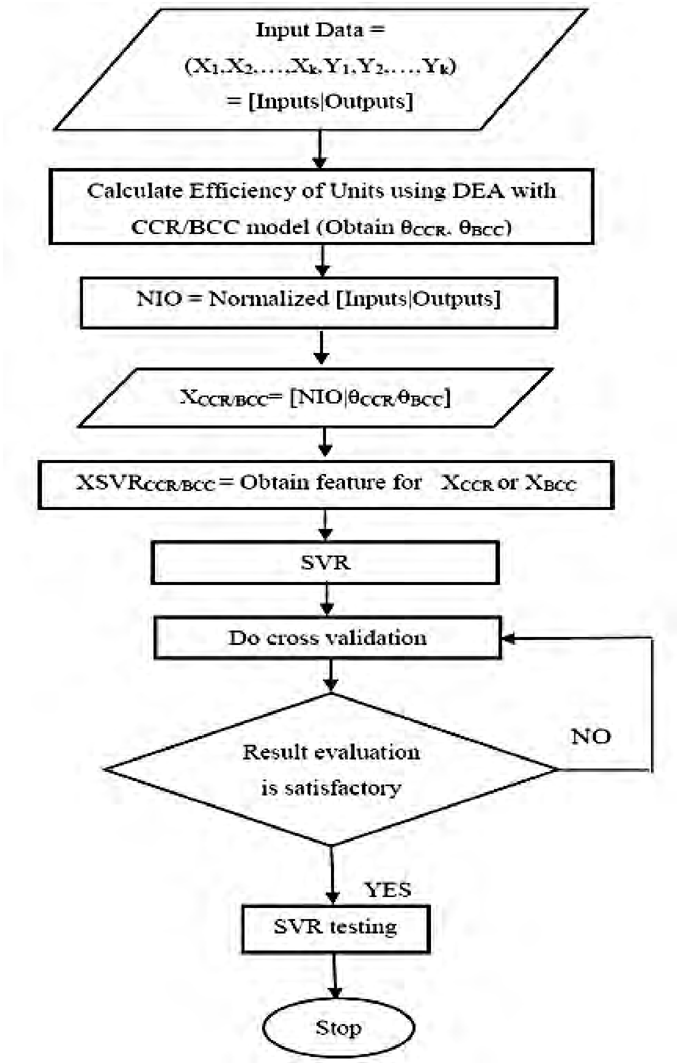


Figure 2 Framework DEA-SVR (CCR/BCC-SVR) algorithm

5. Experimental results and performance comparisons

This section explains the experiments carried out to test the performance of the proposed DEA-SVR algorithm for large DMU's data sets. The performance is in terms of accuracy which is measured by MSE and SCC. Performance comparisons between DEA-SVR and DEA-NN are also presented.

5.1 Dataset

The experimental data set for this research was taken from [1] which consist of 5 data sets. Each dataset contains 5000 units and each unit has 6 attributes of 3 inputs and 3 outputs. A sample of original dataset 1 is shown in Table 2. We used both CCR and BCC models for the DEA, and obtained the efficiency of units using a DEA software. The efficiency of units by DEA-SVR (i.e. CCR-SVR and BCC-SVR) and DEA-NN (i.e. CCR-NN and BCC-NN) were obtained using LIBSVM and MATLAB software [25].

Table 2: Sample of dataset 1

Units	Input 1	Input 2	Input 3	Output 1	Output 2	Output 3
U 1	971	99	471	1777	575	7284
U 2	190	12	4763	3312	184	2884
U 3	732	83	9019	96	1935	8575
U 4	558	58	1482	2042	1172	2766
U 5	230	20	7527	1301	892	1527
⋮	⋮	⋮	⋮	⋮	⋮	⋮
U 5000	555	34	3686	4965	857	1769

5.2 The experiments

The selection of kernel function and the related parameters is very important in SVR which affect the accuracy. In this study, we selected RBF kernel function presented in Eq. (11) for the ν-SVR model, and obtained the best value for C, γ and ν after running the model 10 trials.

5.3 The results and comparisons

Table 3 shows the comparison of MSE and SCC obtained by CCR-SVR and CCR-NN methods using 2-Fold cross validation. The optimum parameters C^* = 5000, γ^*=3.5 and ν^*= 0.2 for CCR-SVR were used. The number of hidden layers for CCR/BCC-NN method is only one and it is based on back-propagation NN algorithm as in [1]. The experiments with more than 1 hidden layers did not produce better results. It is clear from this table that CCR-SVR shows its superiority over CCR-NN. Similarly, when BCC was used instead of CCR in DEA, the performance of the hybrid of DEA and SVR is consistently superior over DEA-NN in all the datasets as shown in Table 4.

Table 3 Comparison CCR-SVR with CCR-NN based on MSE for 2-Fold cross validation

DATA	CCR-SVR (MSE%)	CCR-NN (MSE%)	CCR-SVR (SCC)	CCR-NN (SCC)
Dataset 1	**15.57724121**	28.29623142	**0.961521214**	0.930212904
Dataset 2	**14.25318371**	22.80812213	**0.962444729**	0.93968861
Dataset 3	**15.22639321**	28.9949052	**0.961403747**	0.926623163
Dataset 4	**18.29469601**	25.1722072	**0.952030708**	0.934280176
Dataset 5	**15.98901655**	26.35747573	**0.95940308**	0.933102718

Table 4 Comparison BCC-SVR with BCC-NN based on MSE for 2-Fold cross validation with C^* = 3000, γ^*= 2 and ν^*= 0.2

DATA	BCC-SVR (MSE%)	BCC-NN (MSE%)	BCC-SVR (SCC)	BCC-NN (SCC)
Dataset 1	**15.59618271**	23.5786115	**0.960018853**	0.939645542
Dataset 2	**14.76760884**	19.97981078	**0.962188565**	0.948528197
Dataset 3	**14.93213656**	29.19744074	**0.962088283**	0.925849456
Dataset 4	**20.2955933**	25.47192812	**0.950387897**	0.937507278
Dataset 5	**15.59618271**	23.5786115	**0.955810989**	0.936065575

To show further the consistency of the DEA-SVR method's performance, a 10-Fold cross validation was tested. Table 5 and Table 6 show the MSE and SCC results obtained by CCR-SVR as compared to CCR-NN, and BCC-SVR with BCC-NN, respectively.

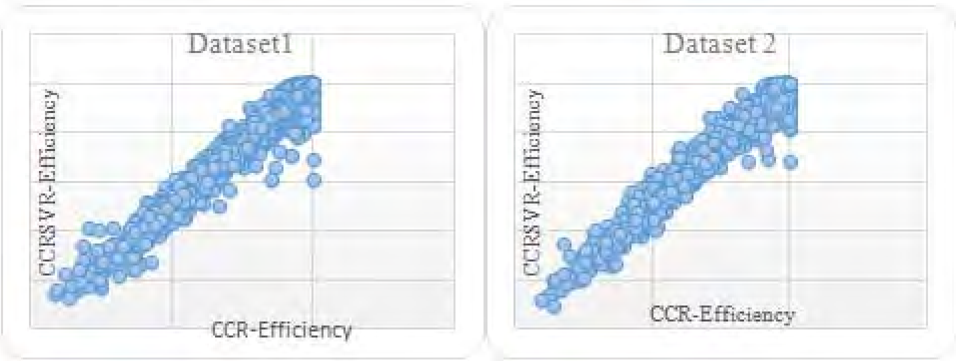
Table 5 Comparison CCR-SVR with CCR-NN based on MSE for 10-Fold cross validation with C^{*} = 5000, γ^{*}=3.5 and ν^{*}= 0.2

DATA	CCR-SVR (MSE%)	CCR-NN (MSE%)	CCR-SVR (SCC)	CCR-NN (SCC)
Dataset 1	**9.316208325**	19.72704706	**0.975197988**	0.947425955
Dataset 2	**8.690255927**	20.00818703	**0.976963441**	0.946980895
Dataset 3	**9.13459436**	21.70755195	**0.976930264**	0.945800464
Dataset 4	**12.61932849**	22.11342657	**0.963231117**	0.935649325
Dataset 5	**9.233263575**	20.19671835	**0.977305819**	0.949852873

Table 6 Comparison BCC-SVR with BCC-NN based on MSE for 10-Fold cross validation with C^{*} = 3000, γ^{*}= 2 and ν^{*}= 0.2

DATA	BCC-SVR (MSE%)	BCC-NN (MSE%)	BCC-SVR (SCC)	BCC-NN (SCC)
Dataset 1	**8.97844857**	18.62741104	**0.977685844**	0.954235309
Dataset 2	**9.949965004**	21.98891693	**0.974599614**	0.943763747
Dataset 3	**9.907717096**	17.98750091	**0.975068769**	0.954697954
Dataset 4	**13.02341674**	17.9968123	**0.9666366405**	0.953332026
Dataset 5	**12.20289311**	21.70333852	**0.971035627**	0.948935919

The CCR/BCC-SVR efficiency of experimental data sets generated by the model indicates the efficiency score as calculated by CCR/BCC-SVR are compared to the actual efficiency score of CCR/BCC efficiency as obtained from the CCR/BCC models. The graph in Figure 3 shows that the CCR-SVR predictions for efficiency score appear to be a good estimate for almost all cases as compared to the actual CCR-Efficiency.



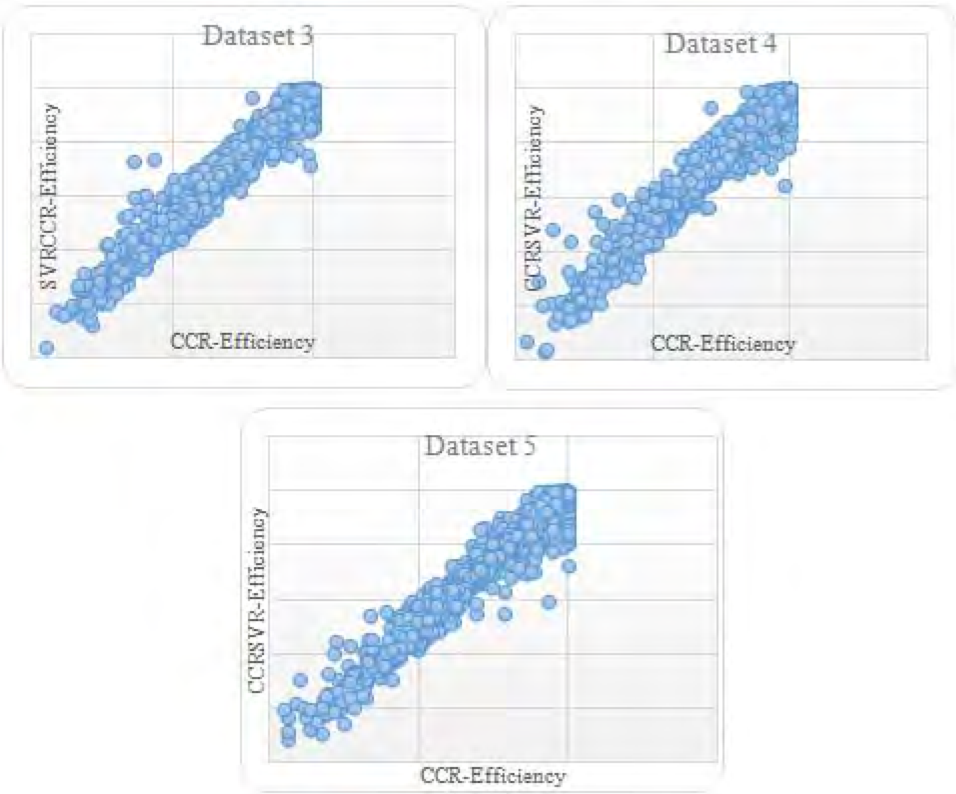


Figure 3 CCR-SVR efficiency prediction as compared with actual DEA efficiency score for the experimental dataset (the range of *x* and *y* axis is between 0 and 100)

The Error Distribution Percentage (EDP) obtained by CCR-SVR and CCR-NN models presented in [1] are compared and presented in Table 7 and Table 8 for 2-Fold and 10-Fold cross validation. The EDP of CCR-SVR model is optimized than EDP of CCR-NN in both the 2-Fold and 10-Fold cross validation.

Table 7 Comparison between CCR-SVR with CCR-NN based on EDP(%) for experimental data sets with 2-Fold Cross Validation

Data set s	Method	Error < 5	5 ≤ Error < 10	10 ≤ Error < 15	15 ≤ Error < 20	20 ≤ Error < 25	Error ≥ 25
1	**CCR-SVR**	**84.690158**	**13.3365735**	**1.255569**	**0.405022**	**0.202511**	**0.081004**
	CCR-NN	68.853787	24.706359	5.386796	0.891049	0.121507	0.040502
2	**CCR-SVR**	**85.938859**	**12.087027**	**1.490733**	**0.402901**	**0.040290**	**0.040290**
	CCR-NN	74.858985	21.071716	3.182917	0.644641	0.161160	0.080580
3	**CCR-SVR**	**86.209677**	**11.370968**	**1.774194**	**0.403226**	**0.120968**	**0.120968**
	CCR-NN	70.282258	24.112903	4.072581	1.048387	0.322581	0.161290
4	**CCR-SVR**	**83.196385**	**13.352506**	**2.588332**	**0.493017**	**0.246508**	**0.123254**
	CCR-NN	72.596549	22.514380	3.861956	0.821693	0.164338	0.041085
5	**CCR-SVR**	**85.217739**	**11.666009**	**2.237315**	**0.639233**	**0.159808**	**0.079904**
	CCR-NN	70.195765	24.730324	3.915302	1.038753	0.119856	0

Table 8 Comparison between CCR-SVR with CCR-NN based on EDP(%) for experimental data sets with 2-Fold Cross Validation

Data sets	Method	Error < 5	5 ≤ Error < 10	10 ≤ Error < 15	15 ≤ Error < 20	20 ≤ Error < 25	Error ≥ 25
1	**CCR-SVR**	**91.002045**	**7.770961**	**1.022494**	**0.204499**	**0**	**0**
	CCR-NN	78.936605	17.995910	2.249489	0.613497	0.204499	0
2	**CCR-SVR**	**91.060291**	**7.692308**	**1.247401**	**0**	**0**	**0**
	CCR-NN	76.923077	20.374220	2.079002	0.415800	0.207900	0
3	**CCR-SVR**	**90.143737**	**8.829569**	**1.026694**	**0**	**0**	**0**
	CCR-NN	77.618070	17.659138	4.106776	0.616016	0	0
4	**CCR-SVR**	**88.866799**	**9.343936**	**1.192843**	**0.397614**	**0.198807**	**0**
	CCR-NN	71.570577	25.646123	2.584493	0.198807	0	0

5	**CCR-SVR**	**91.816367**	**6.986028**	**0.998004**	**0.199601**	**0**	**0**
	CCR-NN	76.047904	20.359281	2.994012	0.598802	0	0

The EDP obtained by BCC-SVR and BCC-NN models are compared and shown in Table 9 and Table 10 for 2-Fold and 10-Fold cross validation. The EDP of BCC-SVR model is optimized than EDP of BCC-NN in both the 2-Fold and 10-Fold.

Table 9 Comparison between BCC-SVR with BCC-NN based on EDP(%) for experimental data sets with 10-Fold Cross Validation

Data sets	**Method**	**Error < 5**	**5 ≤ Error < 10**	**10 ≤ Error < 15**	**15 ≤ Error < 20**	**20 ≤ Error < 25**	**Error ≥ 25**
1	**BCC-SVR**	**86.881288**	**10.543260**	**1.931589**	**0.321932**	**0.120724**	**0.201207**
	BCC-NN	74.567404	21.488934	3.259557	0.523139	0.080483	0.080483
2	**BCC-SVR**	**87.861736**	**9.324759**	**1.929260**	**0.602894**	**0.2009646**	**0.080386**
	BCC-NN	78.135048	18.810289	2.572347	0.321543	0.120579	0.040193
3	**BCC-SVR**	**88.340081**	**9.149798**	**1.619433**	**0.526316**	**0.202429**	**0.161943**
	BCC-NN	70.485830	22.914980	5.344130	0.890688	0.161943	0.202429
4	**BCC-SVR**	**85.161290**	**10.604839**	**2.862903**	**0.685484**	**0.322581**	**0.362903**
	BCC-NN	74.677419	20.927419	3.467742	0.645162	0.120968	0.161290
5	**BCC-SVR**	**85.88**	**11**	**1.6**	**1**	**0.32**	**0.2**
	BCC-NN	72.2	22.8	3.72	0.92	0.2	0.16

Table 10 Comparison between BCC-SVR with BCC-NN based on EDP(%) for experimental data sets with 10-Fold Cross Validation

Data sets	**Method**	**Error < 5**	**5 ≤ Error < 10**	**10 ≤ Error < 15**	**15 ≤ Error < 20**	**20 ≤ Error < 25**	**Error ≥ 25**
1	**BCC-SVR**	**92.264151**	**6.981132**	**0.754717**	**0**	**0**	**0**
	BCC-NN	77.169811	20.188679	2.264151	0.377358	0	0
2	**BCC-SVR**	**91.454545**	**6.909090**	**1.454545**	**0.181818**	**0**	**0**
	BCC-NN	74.545454	20.727272	4.545454	0.181818	0	0
3	**BCC-SVR**	**91.748526**	**6.483301**	**1.178782**	**0.589391**	**0**	**0**
	BCC-NN	81.139489	15.717092	2.357564	0.392927	0.392927	0
4	**BCC-SVR**	**89.021956**	**8.582834**	**1.796407**	**0.399201**	**0.199608**	**0**
	BCC-NN	77.045908	20.758483	1.598802	0.598802	0	0
5	**BCC-SVR**	**91.085271**	**6.589147**	**1.550388**	**0.581395**	**0.193798**	**0**
	BCC-NN	76.162791	20.348837	2.713178	0.581395	0.193798	0

The Figure 4 shows the percentage of the efficiency score experimental data sets between 83.19 % and 86.2 % for CCR-SVR and percentage of the efficiency score between 68.85 % and 74.85 % for CCR-NN based on 2-Fold. Thus, CCR-SVR is better than CCR-NN.

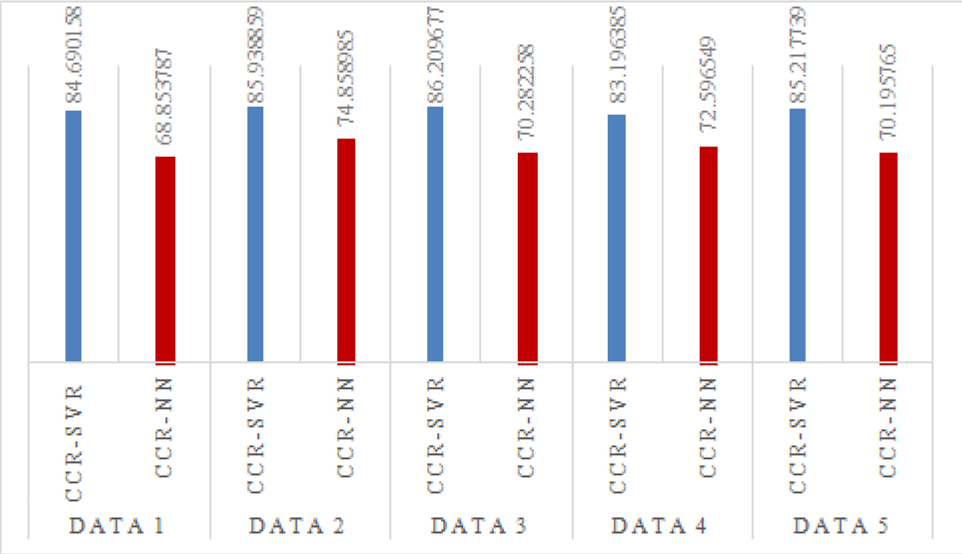


Figure 4 Comparison CCR-SVR with CCR-NN based on EDP for experimental data sets (2-Fold) for error percentage less than 5%

The Figure 5 shows the percentage of the efficiency score experimental data sets between 85.16 % and 88.34 % for BCC-SVR and percentage of the efficiency score between 70.48 % and 78.13 % for BCC-NN based on 2-Fold. Thus, BCC-SVR is better than BCC-NN.

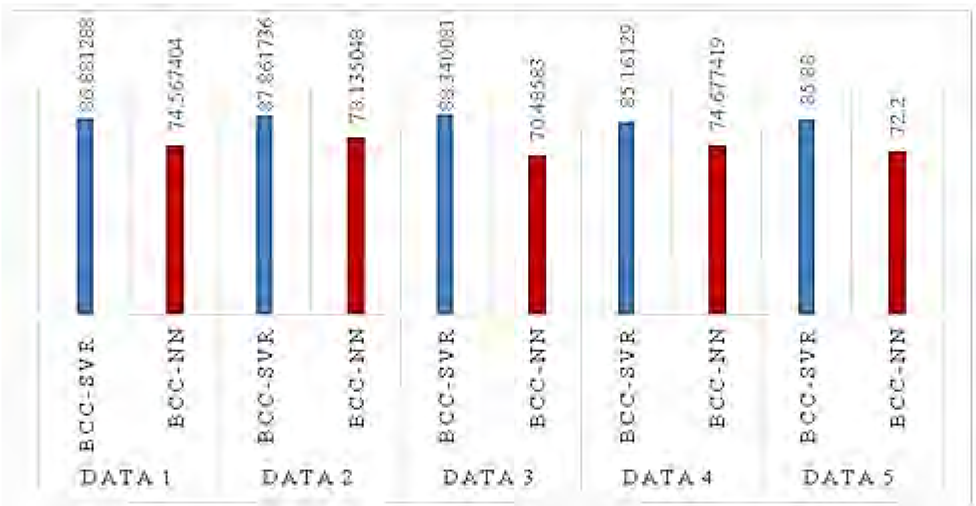


Figure 5 Comparison BCC-SVR with BCC-NN based on EDP for experimental data sets (2-Fold) for error percentage less than 5%

6. Conclusions

Data Envelopment Analysis (DEA) has been widely used in many industrial and economic applications; however, for large DMUs with many inputs and outputs, DEA needs huge computer resources in terms of memory and CPU time. Hence, in this paper, a new combined method was proposed using DEA and SVR (DEA-SVR) for efficiency evaluation of large DMUs to solve some drawbacks which include uncontrolled convergence and non-generalization. For SVR, we selected the ν-SVR using RBF kernel and obtained the best value by 2-fold and 10-fold cross-validation. The DEA-SVR method was applied to five datasets each with 5000 units, and compared the results with the combined methods by DEA and NN. Experimental results demonstrate that the proposed method outperforms the earlier developed combined method of back-propagation neural network and DEA, DEA-NN.

References

[1] A. Emrouznejad and E. Shale, “A combined neural network and DEA for measuring efficiency of large scale datasets,” Computers & Industrial Engineering, vol. 56, no. 1, pp. 249–254, Feb. 2009.

[2] K. Chang, C. Lin, Y. Cao, and C. Lu, “Evaluating branch efficiency of a Taiwanese bank using data envelopment analysis with an undesirable factor”, African Journal of Business Management ,vol. 5, no. 8, pp. 3220–3228, 2011.

[3] J. C. Paradi, H. Zhu, and B. Edelstein, “Identifying managerial groups in a large Canadian bank branch network with a DEA approach,” European Journal of Operational Research, vol. 219, no. 1, pp. 178–187, 2012.

[4] H. Guosheng and Z. Guohong, “Comparison on neural networks and support vector machines in suppliers’ selection,” Journal of Systems Engineering and Electronics, vol. 19, no. 2. pp. 316–320, Apr-2008.

[5] L. Wen and J. Li, “Research of Credit Grade Assessment for Suppliers Based on Multi-Layer SVM Classifier,” Sixth International Conference on Intelligent Systems Design and Applications, vol. 1, pp. 207–211, Oct. 2006.

[6] R. Carbonneau, K. Laframboise, and R. Vahidov, “Application of machine learning techniques for supply chain demand forecasting,” European Journal of Operational Research, vol. 184, no. 3, pp. 1140–1154, Feb. 2008.

[7] M. Zeydan, C. Çolpan, and C. Çobanoğlu, “A combined methodology for supplier selection and performance evaluation,” Expert Systems with Applications, vol. 38, no. 3, pp. 2741–2751, Mar. 2011.

[8] D. Çelebi and D. Bayraktar, “An integrated neural network and data envelopment analysis for supplier evaluation under incomplete information,” Expert Systems with Applications, vol. 35, no. 4, pp. 1698–1710, Nov. 2008.

[9] Y. Tong and M. Yu, “An Integrated DEA-NN Method for Supplier Performance Evaluation,” 2010 International Symposium on Intelligence Information Processing and Trusted Computing, pp. 266–269, Oct. 2010.

[10] F. Lin, C. Yeh, and M. Lee, “The use of hybrid manifold learning and support vector machines in the prediction of business failure,” Knowledge-Based Systems, vol. 24, no. 1, pp. 95–101, 2011.

[11] M. Lee and C. To, “Comparison of Support Vector Machine and Back Propagation Neural Network in Evaluating the Enterprise Financial Distress”, arXiv preprint arXiv,vol. 1, no. 3, pp. 31–43, 2010.

[12] Z. Kong and J. Xue, “A Comparative Study of Supplier Selection Based on Support Vector Machine and RBF Neural Networks,” pp. 917–926, 2013.

[13] C. Yeh, D. Chi, and M. Hsu, “A hybrid approach of DEA , rough set and support vector machines for business failure prediction,” Expert Systems With Applications, vol. 37, no. 2, pp. 1535–1541, 2010.

[14] B. Jiang, W. Chen, H. Zhang, and W. Pan, “Supplier’s Efficiency and Performance Evaluation using DEA-SVM Approach,” Journal of Software, vol. 8, no. 1, pp. 25–30, Jan. 2013.

[15] A. Charnes, W. W. Cooper, and E. Rhodes, “Measuring the efficiency of decision making units,” European Journal of Operational Research, vol. 2, no. 6, pp. 429–444, Nov. 1978.

[16] K. Cooper, W. W., Seiford, L. M., and Tone, “Introduction to Data Envelopment Analysis and itsd used with DEA-Solver Software and Reference.” , : springer, 2006.

[17] V. Vapnik, “The Nature of Statistical Learning Theory,” . Springer-Verlag., 1995.

[18] D. Qu, W. Li, Y. Zhang, B. Sun, Y. Zhong, J. Liu, D. Yu, and M. Li, “Support vector machines combined with wavelet-based feature extraction for identification of drugs hidden in anthropomorphic phantom,” Measurement, vol. 46, no. 1, pp. 284–293, Jan. 2013.

[19] F. Parrella, “Online Support Vector Regression Online Support Vector Machines for Regression,” no. June, 2007.

[20] F. Ernst and A. Schweikard, “Forecasting respiratory motion with accurate online support vector regression (SVRpred),” pp. 439–447, 2009.

[21] A. J. Scholkopf, B. and Smola, Learning with Kernels. The MIT Press, 2002.

[22] S. Abe, Support Vector Machines for Pattern Classification. London: Springer London, 2010.

[23] A. J. Scholkopf, B. , Smola, R. C. Williamson, and P. L. Bartlett, “New Support Vector Algorithms ,” vol. 1245, pp. 1207–1245, 2000.

[24] A. J. Smola, A. Chalimourda, and B. Scho, “Experimentally optimal n in support vector regression for different noise models and parameter settings,” vol. 17, pp. 127–141, 2004.

[25] C. Chang and C. Lin, “LIBSVM : A Library for Support Vector Machines,” pp. 1–39, 2012.