Int. J. Nonlinear Anal. Appl. In Press, 1–9
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.27238.3540

Prediction of DMUs Performance Using Improved SVMs Method Based on DEA

Abedeh Moradi[a], Mohammadreza Farahmand[b,*], Leila Karamali[c]

[a]*Department of Mathematics, Yadegar-e-Emam Khomeini (RAH) Branch, Islamic Azad University, Tehran, Iran*

[b]*Department of Computer, Abarkouh Branch, Islamic Azad University, Abarkouh, Iran*

[c]*Department of Mathematics, Yadegar-e- Emam Khomeini (RAH) Branch, Islamic Azad University, Tehran, Iran*

Abstract

Data Envelopment Analysis (DEA) is an effective method for measuring the efficiency of Decision Making Units (DMUs). DMUs have been generated based on the production function, cost function, and efficiency for evaluation and selection. Prediction of DMUs' Performance (PDMUP) is a problem to select the best among DMUs based on input and output data of the DMUs. PDMUP includes evaluating ranking and efficiency for DMUs. Improving the accuracy and computation time in measuring the efficiency of PDMUP has been one of the main challenges. Hence, in this paper, a method is proposed using DEA and Support Vector Machine (SVM) for predicting DMUs. This method has been compared using datasets in earlier research for large DMUs. The experimental results of comparisons between PDMUP method and early methods demonstrate that the method can significantly improve the accuracy and reduce the computation time in predicting the efficiency of large DMUs.

Keywords: Data Envelopment Analysis, Support Vector Machine, Decision Making Units, Efficiency.
2020 MSC: Primary xxxxx (mandatory); Secondary xxxxx, xxxxx (optionally)

1 Introduction

Data Envelopment Analysis (DEA) is an effective method for obtaining the efficiency of Decision Making Units (DMUs) [10, 11, 15]. DMUs have been generated based on the production function, cost function, and efficiency for evaluation and selection [24, 19]. Conventional DEA methods such as Charnes, Cooper and Rhodes (CCR) and Banker, Charnes and Cooper (BCC) models have been proposed [10, 5]. One of the important issues in DEA literature has been ranking efficient units. Hence, methods such as Additive (ADD), Enhanced Russell Measure (ERM), and Slack-Based Measure (SBM) have been proposed to classify the DMUs into efficient and inefficient units, which are not able to complete ranking for DMUs [10, 18, 27, 12]. In most of these DEA models, the best DMUs have an efficiency score of unity, and multiple DMUs are usually efficient.

DEA-determined efficiency is identical to the Productivity Index used in Engineering and Economics and, as such, uses DMU inputs and outputs to calculate the performance level of the DMUs being considered [22]. As per the classical definition of efficiency, DEA uses a ratio of a weighted output to a weighted input and permits each DMU to select any weights it wants to use for each input and output [28].

*Corresponding author
Email address: mmrezafarahmand@gmail.com (Mohammadreza Farahmand)

Received: May 2022 *Accepted:* June 2022

Recently, an integration of DEA and Neural Network (NN), DEA-NN, for supplier evaluation has been proposed based on incomplete information on criteria and multiple inputs and outputs data [8, 17]. In their work, they highlighted that DEA often failed to work effectively as it is associated with homogeneity and accuracy assumptions. The main contribution of the method is its ability to work with incomplete information on evaluation attributes, which is a common problem in real-life situations. An integrated method of NNs and DEA has also been proposed for a supplier evaluation system with several inputs and output data [26]. They commented that DEA still requires more computer resources in comparison with NNs, which are suitable tools for simulation and learning. They found that developing NNs can be useful to reduce the time of the DEA process. In addition, it paves the way for prediction or checking of suppliers' efficiency. In another work, an integrated method of DEA, Decision Tree (DT), and NNs has been presented for evaluation of suppliers' performance [33]. 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 (e.g. [15, 20]). 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 [15].

To overcome the computation time in DEA, several researchers tackled the problem in large data sets using different DMUs [2, 3, 4, 6, 14]. As an example, Dulá [14] discussed a DEA computation study involving DEA problems with up to 100000 DMUs. Initial results from his study showed that the computation time for large DMUs is very high when using LP. He improved the LP algorithm and calculated the computational time for DMUs with different numbers of attributes. The computational time was obtained around 14 hours for 50000 DMUs with 20 attributes.

On the other hand, for a massive data set with high dimensions of inputs and outputs for DMUs, DEA needs huge computer resources in terms of memory and CPU time [15]. This can be a main shortcoming of conventional DEA for efficiency evaluation of DMUs for real-world problems in both public and private sectors. The use of the DEA method is cumbersome since it involves determining optimum efficiency using an LP approach, especially when the size of DMUs is large. It is further complicated when the LP model is infeasible due to too small or too big values of inputs. Some of the methods also suffer from obtaining complete ranking. The main problems with DEA based methods for large DMUs raking are time consuming and insufficient accuracy.

Support Vector Machine (SVM) is a powerful technique in prediction methods for solving classification and regression. This technique is a generative learning system based on recent advances in statistical learning theory and is used in real-world applications (e.g., [31, 30, 1, 29, 13]). SVM is adaptive and can handle small training and testing sets for research on credit grade assessment [32]. The real examples show that the SVM method is better than NNs and able to overcome the drawbacks of NNs [20]. Several machine learning techniques, including SVM for regression, have been used to forecast the distorted demand at the end of a supply chain [7]. A comparative study of supplier selection based on SVM and Radial Basis Function Neural Network (RBFNN) also showed that SVM is superior, giving more accurate results than the RBFNN algorithm [23]. In another research by Yeh et al. [34] 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 data set 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 Jiang et al. [21]. In addition, Farahmand et al. [16] 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.

As can be found in the above literature, accuracy and computational complexity in the classification of large DMUs have been two of the main challenges in the field. Besides that, requiring huge computer resources that spend large amount of memory and CPU time for large DMUs is one of the main issues connected to the conventional DEA. In addition, the approaches used in prior research have some drawbacks, which include uncontrolled convergence, non-generalization, and instability. Thus, in this paper, a new combination of DEA and Support Vector Classification (SVC) for Binary-class and Multi-class classification, DEA-(B\M)SVC, is proposed for PDUMP, and its performance is tested to evaluate the efficiency with a large number of DMUs.

The remainder of this paper is organized as follows. The proposed DEA-BSVC Method is presented in Section 2. Section 3 shows experimental results and performance comparisons. Finally, conclusions are presented in Section 5.

2 Proposed DEA-BSVC method for PDUMP

In this section, an effective method is proposed based on DEA and SVM for binary-class classification, DEA-BSVC, in large sets of DMUs based on efficient and inefficient DMUs. Assume that there are n DMUs with $\mathrm{DMU}_j = (X_j, Y_j)$, $(j = 1, 2, \ldots, n)$, where vectors X_j and Y_j denote the observed data, where $X_j = (x_{1j}, x_{2j}, \ldots, x_{mj})$ is a column vector of observed inputs and $Y_j = (y_{1j}, y_{2j}, \ldots, y_{sj})$ is a column vector of observed outputs for each DMU, where m and s are the number of inputs and outputs, respectively. Suppose $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \in \mathbb{R}^M \times \{-1, 1\}$, where $\mathbb{R}^M$, M and $\{-1, 1\}$ are a dot product space, attributes $(M = m + s)$ and binary-class. The primal objective function for SVC is as follows:

$$\begin{aligned} &\min_{w,b} \quad \frac{1}{2} w^T w \\ &\text{s.t.} \quad \begin{cases} y_i - \left(w^T \varphi(x_i) + b\right) \le \varepsilon, \\ \left(w^T \varphi(x_i) + b\right) - y_i \le \varepsilon, \\ i = 1, 2, \ldots, n. \end{cases} \end{aligned} \tag{2.1}$$

where $\varphi(x)$ is kernel function, w is margin and (x_i, y_i) belong to the training set. In fact, SVC uses a bound to set based on tolerance and error number. The dual optimization problem of SVC model is obtained as follows [30]:

$$\begin{cases} \max\limits_{\alpha} \quad -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \varphi(x_i)\varphi(x_j)(\alpha_i \alpha_j) + \sum_{i=1}^{n} \alpha_i \\ \text{subject to:} \quad \sum_{j=1}^{n} \alpha_j = 0, \qquad \alpha_i \in [0, C]. \end{cases} \tag{2.2}$$

and the estimate function is as follows:

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i K(x, x_i) + b \right). \tag{2.3}$$

he proposed method consists of four main stages. In stage 1, DEA is performed to obtain the efficiency of DMU data. In stage 2, a pre-processing proposes for input and output factors of DMUs and the efficiency of large DMUs data. Stage 3 selects the best parameters of the SVM using a K-fold cross-validation model. In Stage 4, the efficiency of large DMUs data is utilized for classification using the best SVM method. In the proposed method combine one common methods of DEA with SVM and the method call DEA-SVM. In this section, the proposed method is used for classification of large DMUs based on two classes. To evaluate the accuracy of the proposed method, the classification data obtained from DEA-SVM are compared with the DEA-NN method.

Fig. 1 shows a DEA-SVM algorithm for classification of large DMUs. In this figure, $V_{\mathrm{DMU}_j} = (X_j, Y_j, \theta_j)$ (for $j = 1, 2, \ldots, n$) are vectors belonging to $\mathbb{R}^M \times \{-1, 1\}$, where $M = m + s$. In addition, this figure is divided into three phases: input layers, hidden layers, and output layer. Input layers are considered for DMU vectors as input SVC. The second phase denotes hidden layers that calculate by kernel function and output layers classify input variable as introduced by Eq. (2.3) in $f(x)$.

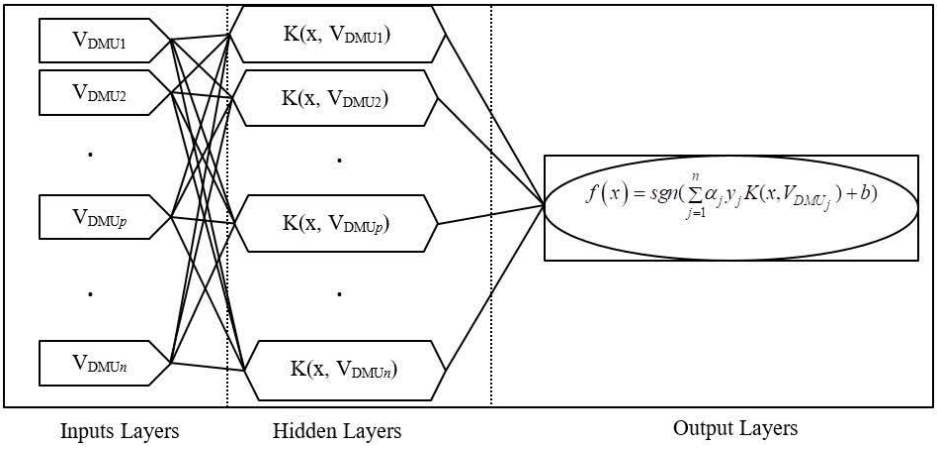


Figure 1: A DEA-SVM algorithm for classification of large DMUs efficiency

The data used for training the SVM includes input, output and efficiency of DEA. The input and output of DEA consider for DMUs as the SVM input and the efficiency as the output of SVM. The efficiencies of DMUs compute using DEA models and then use them as input to DEA-SVM method. The SVM algorithm in DEA is trained by iteratively processing a training sample, comparing the network's prediction of efficiency scores for each sample of DMUs with actual known efficiency scores. DMUs set "G" and V_{DMU_j} set (input data SVC set) "G*" are defined as follows:

$$G = \left\{ \mathrm{DMU}_j = (X_j, Y_j) \;\middle|\; X_j = (x_{1j}, x_{2j}, \ldots, x_{mj}),\quad Y_j = (y_{1j}, y_{2j}, \ldots, y_{sj}),\qquad j = 1, 2, \ldots, n \quad \right\}. \tag{2.4}$$

$$G^* = \left\{ V_{\mathrm{DMU}_j} = (X_j, Y_j, \theta_j) \;\middle|\; \theta_j \text{ is efficiency of } \mathrm{DMU}_j,\ j = 1, 2, \ldots, n \right\}. \tag{2.5}$$

where m and s are number of inputs data and outputs data, respectively. In addition, the so called DEA-SVM based efficiency prediction model was also developed for classification process of new DMUs. The algorithm for this method is illustrated in this section. A DMU is considering to be categorized for two classes as efficient (weak and strong efficiency) and inefficient DMUs sets that call "E" and "I" respectively, as follows:

1. If $\theta_j = 1$, then DMU_j belongs to E.
2. If $\theta_j < 1$, then DMU_j belongs to I.

where θ_j is efficiency score for DMU_j. Also, E^* and I^* define based on E and I as follows:

$$E^* \cup I^* = \left\{ V_{\mathrm{DMU}_j} = (\mathrm{DMU}_j, \theta_j) \;\middle|\; \mathrm{DMU}_j = (X_j, Y_j) \text{ and } \theta_j \text{ is efficiency of } \mathrm{DMU}_j,\ j = 1, 2, \ldots, n \right\}.$$

where, $E^* \cup I^* = G^*$ and $E^* \cap I^* = \varnothing$.

$$E^* = \{ V_j \mid \mathrm{DMU}_j \in E,\ j = 1, 2, \ldots, n \}. \tag{2.6}$$

$$I^* = \{ V_j \mid \mathrm{DMU}_j \in I,\ j = 1, 2, \ldots, n \}. \tag{2.7}$$

Also, the training and testing sets for E^* and I^* are defined as follows:

$$E^*_{tr} = \left\{ V_j^E = V_j \;\middle|\; \mathrm{DMU}_j \in E_{tr}, \theta_j = 1, j = tr_1, tr_2, \ldots, tr_k \right\}. \tag{2.8}$$

where E_{tr} and k are training set and number of training vectors respectively.

$$I^*_{te} = \left\{ V_j^I \;\middle|\; \mathrm{DMU}_j \in I_{te}, \theta_j = -1, j = te_1, te_2, \ldots, te_l \right\}. \tag{2.9}$$

where I_{te} and l are testing set and number of test vector respectively and $k + l = n$. Hence, according to SVM algorithm for two class classification, steps in implementing the DEA-SVM binary-class classification (DEA-BSVC) method is proposed in Algorithm 1 as following:

Algorithm 1: DEA-BSVC method

Step 1: Obtain $\left\{ V_{\mathrm{DMU}_j} = (X_j, Y_j, \theta_j) \;\middle|\; \theta_j \text{ is efficiency of } \mathrm{DMU}_j \right\}$, where $X_j = (x_{1j}, x_{2j}, \ldots, x_{mj})$, $Y_j = (y_{1j}, y_{2j}, \ldots, y_{sj})$, θ_j is obtained using CCR method for $j = 1, 2, \ldots, n$.
Step 2: Obtain efficient DMUs "E" and inefficient DMUs "I" sets.
where, $E = \left\{ V_{\mathrm{DMU}_j} = (X_j, Y_j, \theta_j) \;\middle|\; \theta_j = 1 \right\}$ and $I = \left\{ V_{\mathrm{DMU}_j} = (X_j, Y_j, \theta_j) \;\middle|\; \theta_j < 1 \right\}$.
Step 3: Obtain E^* and I^* sets using Eq. (2.6) and (2.7).
Step 4: Obtain training "E_{tr}" and testing "I_{te}" sets.
Step 5: Obtain E^*_{tr} and I^*_{te} using Eq. (2.8) and (2.9).
Step 6: Obtain class -1 and class 1 for new DMUs set.
Step 7: Stop.

In Fig. 2 a process stream of DEA-SVC algorithm for DMUs classification is shown.

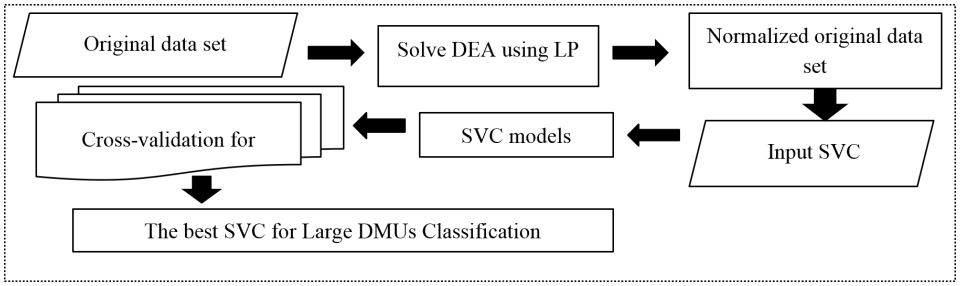


Figure 2: A process stream for classification efficiency of DEA using DEA-SVC

2.1 An illustrative example

In this section, an illustrative example corresponding to the steps proposed in Algorithm 2. The example is meant to describe the proposed on the DEA-MSVC method with step-by-step illustration. A sample of 25 DMUs as in [25] was used. The sample includes 2 Inputs and 3. This example demonstrates how to classify 5 new DMUs for comparison between three classes, thus we have:

Step 1: Obtain $G^* = \left\{V_{\mathrm{DMU}_j} = (X_j, Y_j, \theta_j) \;\middle|\; \theta_j \text{ is efficiency of } \mathrm{DMU}_j\right\}$,, θ_j is obtained using CCR method for $j = 1, 2, \ldots, 25$. Thus, $G^* = \{(X_1, Y_1, \theta_1), (X_2, Y_2, \theta_2), \ldots, (X_{25}, Y_{25}, \theta_{25})\}$.

Step 2: Obtain C_1, C_2 and C_3 classes using Eq. (2.10),

$$\begin{aligned} C_1 &= \left\{\hat{V}_j = (\mathrm{DMU}_j, z_1) \;\middle|\; \mathrm{DMU}_j = (X_j, Y_j), j = 1, 2, \ldots, 25\right\} \\ C_2 &= \left\{\hat{V}_j = (\mathrm{DMU}_j, z_2) \;\middle|\; \mathrm{DMU}_j = (X_j, Y_j), j = 1, 2, \ldots, 25\right\} \\ C_3 &= \left\{\hat{V}_j = (\mathrm{DMU}_j, z_3) \;\middle|\; \mathrm{DMU}_j = (X_j, Y_j),\ j = 1, 2, \ldots, 25\right\} \end{aligned} \tag{2.10}$$

Step 3: Obtain TR^* and TE^* sets using Eq. (2.11) and (2.12).

$$\begin{aligned} TR^* &= \left\{V_j^{TR} = (\mathrm{DMU}_j, \theta_j) \;\middle|\; \mathrm{DMU}_j \in TR,\ \theta_j \in \{z_1, z_2, z_3\}\right\} \\ &= \{(\mathrm{DMU}_1, z_1), (\mathrm{DMU}_2, z_1), (\mathrm{DMU}_3, z_3), (\mathrm{DMU}_4, z_1), (\mathrm{DMU}_5, z_3), (\mathrm{DMU}_6, z_3), \\ &\quad (\mathrm{DMU}_7, z_2), (\mathrm{DMU}_8, z_2), (\mathrm{DMU}_9, z_2), (\mathrm{DMU}_{10}, z_2), (\mathrm{DMU}_{11}, z_1), (\mathrm{DMU}_{12}, z_3), \\ &\quad (\mathrm{DMU}_{13}, z_3), (\mathrm{DMU}_{14}, z_3), (\mathrm{DMU}_{15}, z_2), (\mathrm{DMU}_{16}, z_1), (\mathrm{DMU}_{17}, z_2), (\mathrm{DMU}_{18}, z_2), \\ &\quad (\mathrm{DMU}_{19}, z_3), (\mathrm{DMU}_{20}, z_3), (\mathrm{DMU}_{21}, z_3), (\mathrm{DMU}_{22}, z_2), (\mathrm{DMU}_{23}, z_2), (\mathrm{DMU}_{24}, z_3), \\ &\quad (\mathrm{DMU}_{25}, z_2)\}. \end{aligned} \tag{2.11}$$

$$\begin{aligned} TE^* &= \left\{V_j^{TE} = (\mathrm{DMU}_j, \theta_j) \;\middle|\; \mathrm{DMU}_j \in TE,\ j = 26, 27, \ldots, 30,\ \theta_j \in \{z_1, z_2, z_3\}\right\} \\ &= \{(\mathrm{DMU}_{26}, z_1), (\mathrm{DMU}_{27}, z_1), (\mathrm{DMU}_{28}, z_3), (\mathrm{DMU}_{29}, z_1), (\mathrm{DMU}_{30}, z_2)\}. \end{aligned} \tag{2.12}$$

Step 4: Obtain class 1, class 2, class 3 for new DMUs set.

$$\begin{aligned} \text{Class 1} &= \{\mathrm{DMU}_{26}, \mathrm{DMU}_{27}, \mathrm{DMU}_{29}\} \\ \text{Class 2} &= \{\mathrm{DMU}_{30}\} \\ \text{Class 3} &= \{\mathrm{DMU}_{28}\} \end{aligned}$$

Step 5: Stop.

3 Experimental results and performance comparisons

This section explains the experiments carried out to test the performance of the proposed DEA-SVC algorithm for large DMU data sets. The performance is in terms of accuracy, which is measured by several evaluation methods. Performance comparisons between DEA-SVC and DEA-NN are also presented.

3.1 Data set description

The experimental data set for this section was taken from [15] which consists of 5 data sets. Each data set contains 5000 units, and each unit has 6 attributes of 3 inputs and 3 outputs. A sample of original data set 1 is shown in Table 1. The program code for DEA-SVC (i.e. CCR-BSVC and CCR-MSVC) and DEA-NN (i.e. CCR-NN) is improved by LIBSVM [9] implemented in MATLAB software.

Table 1: Sample of large data set 1 with 5000 units, 3 inputs and 3 outputs

DMUs	$\mathbf{I}_1$	$\mathbf{I}_2$	$\mathbf{I}_3$	$\mathbf{O}_1$	$\mathbf{O}_2$	$\mathbf{O}_3$
U1	971	99	471	1777	575	7284
U2	190	12	4763	3312	184	2884
U3	732	83	9019	96	1935	8575
⋮	⋮	⋮	⋮	⋮	⋮	⋮
U5000	555	34	3686	4965	857	1769

The selection of the kernel function and the related parameters is very important in SVC,, which affects the accuracy. In this section, RBF kernel has been used, and the best values for C and γ after running the model for 10 trials.

3.2 Results of DEA-BSVC method

This section explains the experiments carried out to test the performance of the proposed DEA-BSVC algorithm for binary-class classification of large DMUs. Performance comparisons between DEA-BSVC and DEA-NN are also presented. The experimental data set is large data set 1–5 (DATA 1–5). Table 2 and Table 3 show the comparison of accuracy and average time obtained by CCR-BSVC and CCR-NN methods using 2-Fold and 5-fold cross-validation with 10 trials. The optimum parameters $C^* = 8000$, $\gamma^* = 0.07$ for CCR-BSVC were used. It is clear from this table that CCR-BSVC shows its superiority over CCR-NN.

Table 2: A comparison of accuracy and average time between CCR-BSVC and CCR-NN methods for two-class classification using DATA 1–5 with 2-fold and 10 trials

DATA	**CCR-BSVC (Accuracy %)**	**Average Time (second)**	**CCR-NN (Accuracy %)**	**Average Time (second)**
1	91.5166	0.4703	84.5792	2.4768
2	93.6825	0.3681	87.2893	4.3847
3	92.4	0.4928	83.641	3.932
4	92.24	0.4983	86.8983	3.3446
5	92.24	0.4674	84.7321	3.3453

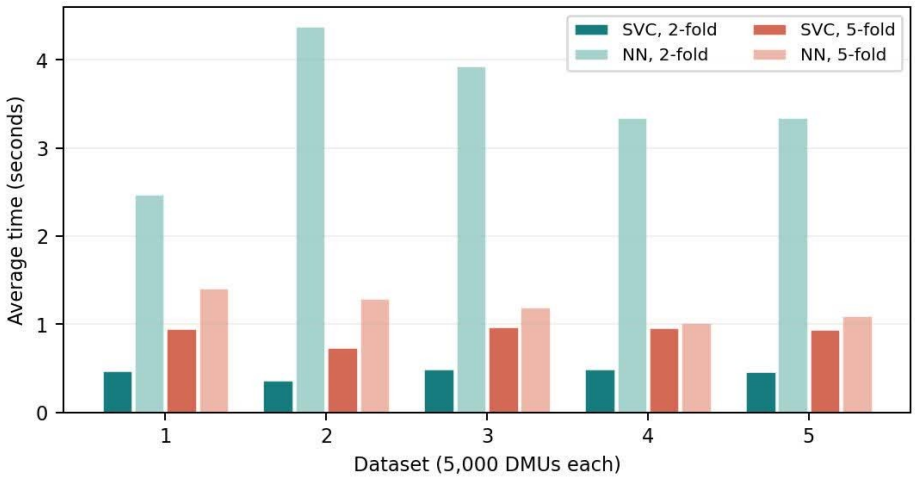


Figure 3: Computational time comparison (10 trials)

Table 3: A comparison of accuracy and average time between CCR-BSVC and CCR-NN methods for two-class classification using DATA 1–5 with 5-fold and 10-fold trials

DATA	CCR-BSVC (Accuracy %)	Average Time (second)	CCR-NN (Accuracy %)	Average Time (second)
1	92.9071	0.9536	85.8907	1.41
2	94.1059	0.7377	86.5429	1.2997
3	92.8	0.9771	85.9977	1.193
4	92	0.9676	83.8916	1.019
5	92.008	0.9464	85.5121	1.0994

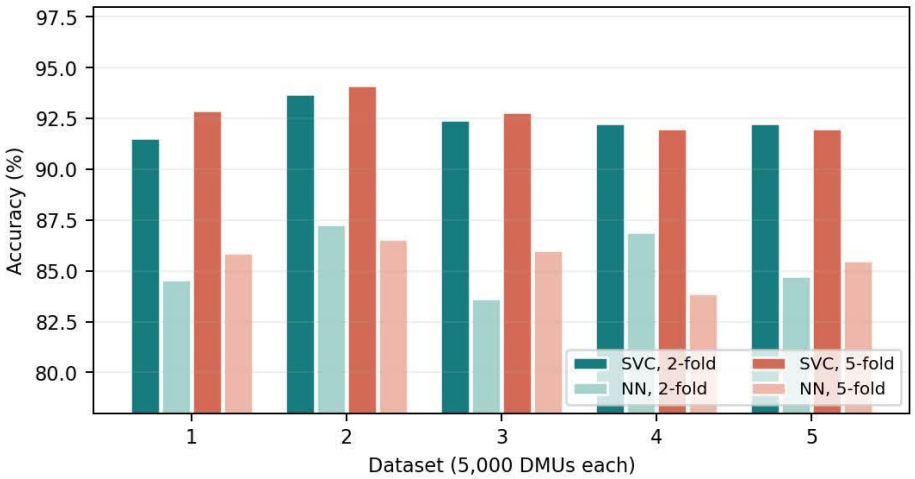


Figure 4: Five-fold and two-fold CV accuracy (10 trials)

The selection of the kernel function and the related parameters is very important in SVC, which affects the accuracy. In this study, LF kernel function was used for the SVC model, and the best values for C and γ were obtained after running the model in different trials. Fig. 5 shows the comparison of average time for 2 and 5-fold cross-validation using CCR-BSVC and CCR-NN methods. The optimum parameters $C^* = 8000$, $\gamma^* = 0.07$ for CCR-BSVC were used. It is clear from this table that CCR-BSVC shows its superiority over CCR-NN.

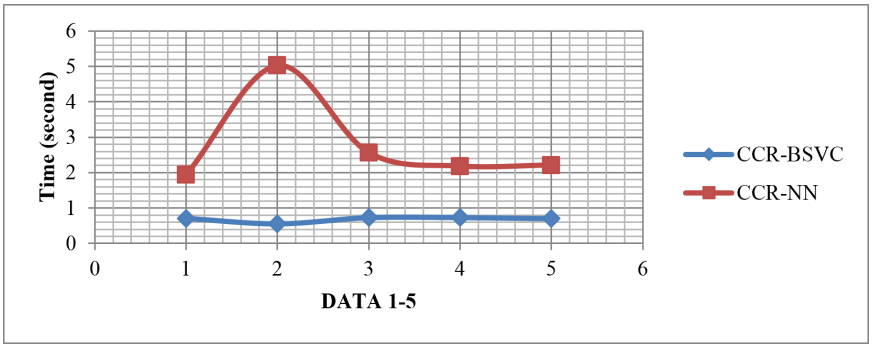


Figure 5: Comparison of average time of CCR-BSVC with CCR-NN based on two classifications and 2 and 5-fold cross-validation for DATA 1–5

4 Conclusions

This paper proposed DEA-SVC along with its parameters for binary-class (DEA-BSVC) and multi-class classification (DEA-MSVC) of PDUMP in large data sets to improve the accuracy and computation time. In DEA-MSVC, to obtain classes C_k, two methods are suggested. The experimental results on different 5 large data sets showed that the time complexity using 2-fold and 5-fold cross-validation for the proposed method is much better than the DEA-NN method. We also proposed a new effective algorithm for measuring the accuracy of classification in large data sets for PDUMP. Finally, the proposed methods tackled the computation time in efficiency evaluations of DMUs for PDMUP in classification of large DMUs.

References

[1] S. Abe, *Support vector machines for pattern classification*, Springer, London, 2010.

[2] A. I. Ali, *Streamlined computation for data envelopment analysis*, European Journal of Operational Research **64** (1993), no. 1, 61–67.

[3] A. I. Ali, *Computational aspects of dea*, Data Envelopment Analysis: Theory, Methodology, and Applications, Springer Netherlands, 1994, pp. 63–88.

[4] A. I. Ali and L. M. Seiford, *Computational accuracy and infinitesimals in data envelopment analysis*, INFOR **31** (1993), no. 4, 290–297.

[5] R. D. Banker, A. Charnes, and W. W. Cooper, *Some models for estimating technical and scale inefficiencies in data envelopment analysis*, Management Science **30** (1984), no. 9, 1078–1092.

[6] R. S. Barr, K. A. Killgo, T. F. Siems, and S. Zimmel, *Evaluating the productive efficiency and performance of US commercial banks*, Managerial Finance **28** (2002), no. 8, 3–25.

[7] R. Carbonneau, K. Laframboise, and R. Vahidov, *Application of machine learning techniques for supply chain demand forecasting*, European Journal of Operational Research **184** (2008), no. 3, 1140–1154.

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

[9] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, ACM Transactions on Intelligent Systems and Technology **2** (2011), no. 3, 27:1–27:27.

[10] A. Charnes, W. W. Cooper, and E. Rhodes, *Measuring the efficiency of decision making units*, European Journal of Operational Research **2** (1978), no. 6, 429–444.

[11] C.-M. Chen and J. van Dalen, *Measuring dynamic efficiency: theories and an integrated methodology*, European Journal of Operational Research **203** (2010), no. 3, 749–760.

[12] W. W. Cooper, L. M. Seiford, and K. Tone, *Data envelopment analysis: A comprehensive text with models, applications, references and DEA-Solver software*, 2 ed., Springer, New York, 2007.

[13] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*, Cambridge University Press, Cambridge, 2000.

[14] J. H. Dulá, *A computational study of DEA with massive data sets*, Computers & Operations Research **35** (2008), no. 4, 1191–1203.

[15] A. Emrouznejad and E. Shale, *A combined neural network and DEA for measuring efficiency of large scale datasets*, Computers & Industrial Engineering **56** (2009), no. 1, 249–254.

[16] M. Farahmand, M. I. Desa, and M. Nilashi, *A combined data envelopment analysis and support vector regression for efficiency evaluation of large decision making units*, International Journal of Engineering and Technology (2014), 2310–2321.

[17] M. Farahmand, M. I. Desa, and M. Nilashi, *Hybrid data envelopment analysis and neural networks for suppliers efficiency prediction and ranking*, Proceedings of the International Conference on Recent Trends in Information and Communication Technologies, 2014, pp. 392–400.

[18] R. Färe and C. A. K. Lovell, *Measuring the technical efficiency of production*, Journal of Economic Theory **19** (1978), no. 1, 150–162.

[19] M. J. Farrell, *The measurement of productive efficiency*, Journal of the Royal Statistical Society: Series A (General) **120** (1957), no. 3, 253–281.

[20] H. Guosheng and Z. Guohong, *Comparison on neural networks and support vector machines in suppliers' selection*, Journal of Systems Engineering and Electronics **19** (2008), no. 2, 316–320.

[21] B. Jiang, W. Chen, H. Zhang, and W. Pan, *Supplier's efficiency and performance evaluation using DEA-SVM approach*, Journal of Software **8** (2013), no. 1, 25–30.

[22] C. Kao, *Malmquist productivity index based on common-weights DEA: the case of taiwan forests after reorgani-*

zation, Omega **38** (2010), no. 6, 484–491.

[23] Z.-J. Kong and J.-B. Xue, *A comparative study of supplier selection based on support vector machine and RBF neural networks*, International Asia Conference on Industrial Engineering and Management Innovation (IEMI2012) Proceedings, Springer, Berlin, 2013, pp. 917–926.

[24] T. C. Koopmans, *Analysis of production as an efficient combination of activities*, Activity Analysis of Production and Allocation (T. C. Koopmans, ed.), Wiley, New York, 1951, pp. 33–97.

[25] R. K. Mishra and G. Patel, *Supplier development strategies: a data envelopment analysis approach*, Business Intelligence Journal **3** (2010), no. 1, 99–110.

[26] D. Ozdemir and G. T. Temur, *DEA ANN approach in supplier evaluation system*, World Academy of Science, Engineering and Technology **54** (2009), no. 538, 343–348.

[27] J. T. Pastor, J. L. Ruiz, and I. Sirvent, *An enhanced DEA Russell graph efficiency measure*, European Journal of Operational Research **115** (1999), no. 3, 596–607.

[28] S. Samoilenko and K.-M. Osei-Bryson, *Determining sources of relative inefficiency in heterogeneous samples: methodology using cluster analysis, DEA and neural networks*, European Journal of Operational Research **206** (2010), no. 2, 479–487.

[29] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, *New support vector algorithms*, Neural Computation **12** (2000), no. 5, 1207–1245.

[30] B. Schölkopf and A. J. Smola, *Learning with kernels: Support vector machines, regularization, optimization, and beyond*, MIT Press, Cambridge, MA, 2002.

[31] V. Vapnik, *The nature of statistical learning theory*, Springer, New York, 1995.

[32] L. Wen and J. Li, *Research of credit grade assessment for suppliers based on multi-layer SVM classifier*, Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06) (Jinan, China), 2006.

[33] D. Wu, *Supplier selection: a hybrid model using DEA, decision tree and neural network*, Expert Systems with Applications **36** (2009), no. 5, 9105–9112.

[34] C.-C. Yeh, D.-J. Chi, and M.-F. Hsu, *A hybrid approach of DEA, rough set and support vector machines for business failure prediction*, Expert Systems with Applications **37** (2010), no. 2, 1535–1541.