

Variable selection in data envelopment analysis: A random forest approach with data augmentation**Tzu-Pu Chang^{a*}**^a*Department of Finance, National Yunlin University of Science and Technology, Taiwan***CHRONICLE***Article history:*

Received December 14 2025

Received in Revised Format

December 29 2025

Accepted February 20 2026

Available online February 20
2026

Keywords:

*Data envelopment analysis**Variable selection**Machine learning**Random forest**Permutation importance**Data augmentation***ABSTRACT**

Variable selection is an important step in data envelopment analysis (DEA) when the number of decision making units (DMUs) is insufficient. This research thus proposes a two-stage variable selection method integrating a well-known supervised machine learning technique, the random forest algorithm. In the first stage, a baseline DEA model with full input and output variables is implemented and each DMU can be determined as being efficient or inefficient. In the second stage, a random forest is trained to learn how to classify efficient or inefficient DMUs well in high dimensions. Accordingly, the importance of each variable can be calculated based on permutation importance indices in random forest. This paper further discusses two issues about data augmentation in order to improve the robustness of permutation importance when the number of DMUs is quite small.

© 2026 by the authors; licensee Growing Science, Canada

1. Introduction

Data envelopment analysis (DEA) is a well-known non-parametric method that applies linear programming to evaluate the efficiencies of decision making units (DMUs) in terms of multiple variables (i.e., inputs and outputs) (Endri et al., 2022). Apart from abundant empirical applications, researchers still endeavor to answer an essential question: how to select a relevant input/output combination (Dobrzanski, 2020). This question arises, because we can easily collect a dataset with a small number of DMUs and a large number of inputs and outputs in empirical studies. However, from the methodology perspective, the larger the number of variables is, the less discriminative the DEA model will be (Jenkins & Anderson, 2003; Ekiz and Tuncer Şakar, 2020; Karagiannis & Karagiannis, 2023). To achieve an appropriate level of discrimination, several studies in the literature have suggested some criteria about the relationship between the number of DMUs and input/output variables. For details, please refer to Lee and Cai (2020).

To deal with the above-mentioned problem, variable selection (or variable reduction) becomes a required step in the DEA model when the number of DMUs is insufficient. Therefore, many variable selection approaches have been proposed up to the present. From a different perspective, Lee and Choi (2010) classify variable selection approaches into two categories: approaches that do not relate and approaches that relate to the impact on DEA results. Nataraja and Johnson (2011) consider that all methods can be viewed as statistics-based or not. Moreover, according to our viewpoint, those approaches can be categorized into two groups: “plug-in” methods and “add-on” methods. The former group directly models a certain variable reduction method and plugs it into the objective function or constraints in a linear programming problem, such as Ueda and Hoshiai (1997), Adler and Golany (2002), Benítez-Peña et al. (2020), and Lee and Cai (2020). The latter group proposes some measures (no matter statistically or not) and adds them in multiple steps after performing DEA models to detect variable importance, such as Pastor et al. (2002), Jenkins and Anderson (2003), Ruggiero (2005), and Wagner and Shimshak (2007).

Following the second category, the present paper proposes a novel procedure to select relevant input/output variables. Specifically, this paper suggests a two-stage method, which takes full variables in the baseline DEA model in the first stage

* Corresponding author

E-mail changtp@yuntech.edu.tw (T.-P. Chang)

ISSN 1929-5812 (Online) - ISSN 1929-5804 (Print)

2026 Growing Science Ltd.

doi: 10.5267/j.dsl.2026.2.008

and then implements a random forest algorithm to calculate variable importance in the second stage. Random forest, formally proposed by Breiman (2001), is a popular supervised machine learning (ML) technique in many research fields. This study trains a random forest classifier to recognize efficient and inefficient DMUs determined by the baseline DEA model.

The core idea of the proposed procedure is straightforward. In the first stage, we run a base DEA model including all inputs and outputs. Hence, the efficiency score for each DMU is obtained, and efficient DMUs are determined in this stage. Once all efficient and inefficient DMUs are defined, we intuitively address a typical classification issue: whether or not the binary target variable is efficient, and all inputs and outputs are features in this supervised learning task (the second stage). Obviously, a well-trained classifier must completely separate efficient and inefficient DMUs in high dimensions, because at least the efficient frontier determined in the first stage is the decision boundary of the classifier. Finally, we aim to determine the degrees of importance of all variables in the classifier and further find out which variables are more important in the baseline DEA model.

Among a growing number of supervised ML techniques, there are two crucial characteristics of random forest for why we choose it. First, random forest has been proven to perform well with a “small n and large p ” dataset (Chen & Ishwaran, 2012). This is equivalent to the problem of “fewer DMUs and more input/output variables” in our study. Second, after executing a random forest classifier, the importance of each variable can be computed directly and quickly (Breiman, 2001; Strobl et al., 2008). We note that the variable importance in random forest is not identical to the variable importance in DEA. We shall discuss how to interpret the different meanings between them in the next section. To the best of our knowledge, no study in the literature performs ML techniques to select variables in the DEA model, except Lee and Cai (2020). In fact, Lee and Cai (2020) use a LASSO technique and plug it into a linear programming problem, but our paper applies a two-stage approach to determine the variable importance clearly. Results suggest that the proposed procedure is simpler for researchers to apply.

The remainder of this paper runs as follows. Section 2 illustrates the proposed procedure for variable selection. Section 3 applies this procedure to two datasets published in the literature and discusses some further issues in this procedure. Section 4 concludes this paper.

2. Procedure for Variable Selection

As mentioned above, the proposed two-stage procedure consists of running a baseline DEA model and random forest classifier in the first and second stages, respectively. We briefly introduce this variable selection method as follows.

2.1 First stage: Baseline DEA model

Assume there are n DMUs, and each DMU $_j$ ($j=1, 2, \dots, n$) uses m inputs x_{ij} ($i=1, 2, \dots, m$) to produce s outputs y_{rj} ($r=1, 2, \dots, s$). Hence, in this case the total number of variables is $p = m + s$. The first stage runs a DEA model with a full dataset that includes n DMUs, m inputs, and s outputs. We then obtain the baseline efficiency score for each DMU. It is noteworthy that the proposed procedure is a two-stage approach and does not depend on any form of a DEA model used in the first stage. In other words, researchers can adopt a preferable DEA model with either input or output orientation, or with either constant or variable returns to scale technology. With respect to this baseline model, it should present the worst discrimination and the largest number of efficient DMUs. We further label all DMUs as being efficient or inefficient regardless of their efficiency scores in order to train a random forest classifier in the next stage.

2.2 Second stage: Classification for efficient/inefficient DMUs using random forest

Random forest, also known as random decision forest, is a tree-based machine learning algorithm based on an ensemble learning technique; i.e., bootstrap aggregation (bagging). Briefly, this algorithm builds T decision trees from T bootstrap samples, which are sampled with replacement. In each bootstrap sample the sample size is always equal to the size of the original dataset, and the dimension of features (k) is chosen from the random subset of the original set of features. It is noted that, for each tree, unsampled data are denoted as out-of-bag (OOB) data, which we use for cross-validation and OOB error estimation. Finally, random forest aggregates and chooses the majority class of all trees as the final decision. Turning back to our issue, random forest creates T bootstrap samples with a sample size of n (number of DMUs) and q features ($q < p$) to grow each decision tree in parallel (see Figure 1). In addition, q is often claimed to be the square root of p , meaning that $(p - q)$ variables (either input or output, or both) are randomly dropped out in each bootstrap sample. This randomness property is quite critical for our proposed variable selection method, because we are able to further compute the importance of each input/output variable z_h ($h=1, 2, \dots, m, m+1, m+2, \dots, m+s$). We describe it as follows.

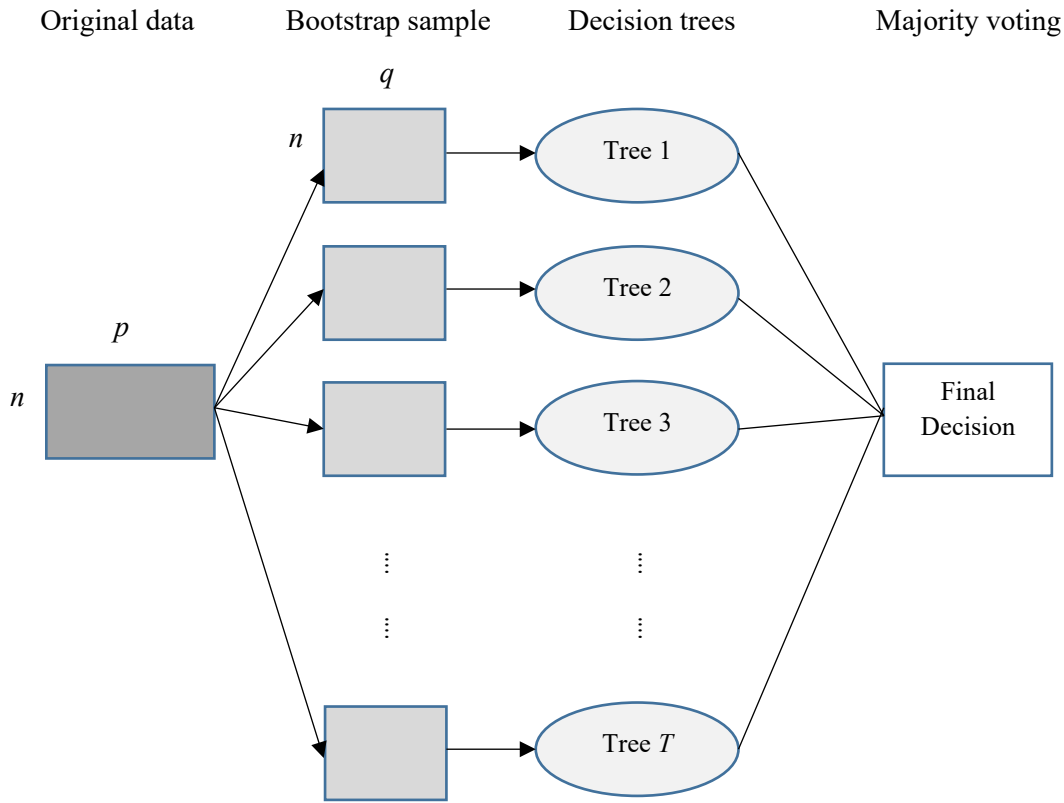


Fig. 1. The concept of a random forest algorithm

2.2.1 Variable importance in random forest

There are two widely-used variable importance measures in random forest: one is mean decrease in impurity based on the Gini index of node impurity; the other one is mean decrease in accuracy, also called permutation importance, based on classification accuracy of OOB data (Strobl et al., 2008; Verikas et al., 2011). This paper adopts permutation importance to evaluate variable importance, because this measure can be broken into the accuracy of each class (i.e., efficient or inefficient). By definition, permutation importance is the difference in prediction accuracy before and after permuting a particular variable z_h in the OOB data (Breiman, 2001). Hence, the permutation importance of the h^{th} variable is calculated by:

$$PI_h = \frac{1}{T} \sum_{t=1}^T (R_{t,h} - R_t), \tag{1}$$

where R_t^{oob} is the OOB error rate in the t^{th} tree before permuting z_h ; and $R_{t,h}^{oob}$ is the OOB error rate in the t^{th} tree after permuting z_h . Moreover, the permutation importance for the efficient and inefficient groups of the h^{th} variable is measured by:

$$PI_h^{eff} = \frac{1}{T} \sum_{t=1}^T (R_{t,h}^{eff} - R_t^{eff}) \tag{2}$$

$$PI_h^{ineff} = \frac{1}{T} \sum_{t=1}^T (R_{t,h}^{ineff} - R_t^{ineff}), \tag{3}$$

where the superscripts *eff* and *ineff* denote efficient and inefficient DMUs in the OOB data, respectively. Accordingly, permutation importance ranges from -1 to +1. The permutation importance computed by Eqs. (1-3) denotes raw importance in the literature. Scaled importance is obtained by dividing by its standard error and converted to a normalized z-score. However, Diaz-Uriarte and de Andrés (2006) and Strobl et al. (2008) claim that raw importance has better statistical properties than the scaled version. Notice that the higher the permutation importance is, the more important the variable is. A negative permutation importance means that the prediction accuracy increases when a certain variable is permuted, indicating that this variable is extremely less important for classifying DMUs as efficient and inefficient DMUs obtained from the baseline DEA model.

2.2.2 Variable importance in the baseline DEA model

From different managerial thoughts, this paper applies two distinct strategies suggested in the literature to select which variables should be omitted or retained. Strategy-A seeks to use fewer input or output variables that can produce a similar pattern of efficiency scores to the baseline DEA, such as Jenkins and Anderson (2003) and Sharma and Yu (2015). Strategy-

B retains input or output variables that show a huge influence on the baseline DEA model; e.g., Wagner and Shimshak (2007) and Li and Liang (2010). Hence, we propose two criteria in correspondence with two strategies as follows.

Strategy-A: Aims to select a reduced set of variables that yields an efficiency score pattern similar to the baseline model (Jenkins & Anderson, 2003). This is achieved by choosing variables with high PI_h , PI_h^{eff} , and PI_h^{ineff} .

Strategy-B: Aims to retain variables that exert a substantial influence on the model to improve discrimination (Wagner & Shimshak, 2007). This involves selecting variables with low PI_h and low PI_h^{eff} .

Strategy-A is straightforward, because a variable with high permutation importance forms a better classification boundary than other variables. As aforementioned, the efficient frontier determined in the baseline DEA is one possible classification boundary in random forest. Therefore, we claim that the most important variable in random forest predicts the efficient frontier better, implying that this variable is an essential input or output to determine the efficiency score in the baseline DEA model.

A positively low permutation importance indicates that the variable is irrelevant to the baseline DEA model, while negative permutation importance represents that the variable can distort the ranking of DMUs. Therefore, strategy-B chooses those inputs and outputs with negative permutation importance that can vastly impact the result of the baseline DEA model. In addition, negative permutation importance must be due to negative PI_h^{eff} or PI_h^{ineff} , or both, whereas we suggest that the negative PI_h^{eff} is necessary, but PI_h^{ineff} is not for strategy-B. A variable with negative PI_h^{eff} shows that the misclassification rate for the efficient group decreases (accuracy rate increases) after permuting this variable. It also means that only using a variable with negative PI_h^{eff} can predict less efficient and more inefficient DMUs in random forest. Hence, we suggest that retaining this kind of variable improves the discrimination of DEA models.

3. Sample Datasets

In this section we employ the proposed procedure to two datasets published in the DEA literature. We apply the R language to perform a random forest technique, and a random seed 1234 is set for reproducible purposes. At the end of this section, we shall discuss two issues about the methodology of random forest and the relation to existing selection approaches.

3.1 Sample of academic departments

Sinuany-Stern et al. (1994) provide a dataset including 21 departments with four outputs and two inputs at Ben-Gurion University. This dataset is widely used in related literature, such as Jenkins and Anderson (2003) and Wagner and Shimshak (2007). In the first stage, we apply an input-oriented and constant returns to scale DEA model as the baseline in accordance with the literature. In the second stage, we run a random forest algorithm and set $T = 100$ as well as $q = 3$.

Table 1 presents the permutation importance, including PI_h , PI_h^{eff} , and PI_h^{ineff} , of each input or output variable. With respect to output variables, graduate students have the highest PI_h , followed by contact hours, showing that graduation students are the most important output for the baseline DEA model. Hence, we can choose graduate students as the only one output in terms of Strategy-A. This result is consistent with Jenkins and Anderson (2003), who find that omitting graduate students' output would lose most of the information and conclude that graduate students are the most important output in order to retain maximum information. However, for Strategy-B, we suggest that publications should be retained, because it shows the lowest PI_h and PI_h^{eff} . This result is the same as Wagner and Shimshak (2007), who indicate that publications are the core output since it substantially impacts efficiency scores. The same criteria can be applied to select input variables; i.e., operation cost and salaries are selected by Strategy-A and Strategy-B, respectively. Considering the one-input and one-output model, the selection from Strategy-B is totally the same as the core model in Wagner and Shimshak (2007).

Table 1

Permutation importance for 21 academic department cases

Variable	PI_h	PI_h^{eff}	PI_h^{ineff}
O1: Grants	0.0028	-0.0112	0.0031
O2: Publications	0.0027	-0.0175	0.0133
O3: Graduate students	0.0340	0.0428	0.0331
O4: Contact hours	0.0304	0.0537	0.0224
I1: Operation cost	0.0238	0.0025	0.0310
I2: Salaries	-0.0036	-0.0167	0.0033

Graduate students (publications) and contact hours (grants) empirically present quite similar permutation importance indices in Table 1. Hence, it is more appropriate to select a two-output and one-input DEA model regarding this dataset. Table 2 exhibits a correlation matrix using Spearman's rank correlation among five input-output combinations. On the one hand, the result exactly supports our argument that the ranking of DMUs obtained by Strategy-A is highly correlated with the ranking

of the baseline. The suggested two-output (graduate students and contact hours) and one-input combination (operation cost) presents a significantly positive correlation coefficient of 0.7387 and determines only two efficient DMUs, indicating that this combination uses less variables to generate a similar result and has higher discriminative power in comparison to the baseline DEA model. On the other hand, it is reasonable that the ranking of DMUs obtained by Strategy-B is irrelevant to that of the baseline. Choosing a two-output (publications and grants) and one-input (salaries) combination, the ranking of DMUs shows a negative but insignificant correlation coefficient. In addition, only two DMUs are denoted as efficient by Strategy-B, implying that this specification is more discriminative than the baseline DEA model.

Table 2

Spearman's rank correlation coefficients for five specifications

Specification	Full	(O3, O4, I1)	(O3, I1)	(O1, O2, I2)	(O2, I2)
Baseline	—				
Full					
Strategy-A	0.7387***	—			
(O3, O4, I1)	[0.0001]				
Strategy-A	0.4657**	0.4964**	—		
(O3, I1)	[0.0334]	[0.0221]			
Strategy-B	-0.0126	-0.5289**	-0.0465	—	
(O1, O2, I2)	[0.9568]	[0.0137]	[0.8414]		
Strategy-B	-0.1522	-0.5413**	-0.3291	0.7627***	—
(O2, I2)	[0.5100]	[0.0113]	[0.1452]	[0.0001]	
No. of efficient DMUs	7	2	1	2	1
Range	0.436	0.987	1.000	0.827	0.827
Standard deviation	0.147	0.316	0.237	0.225	0.193

Notes: *, **, and *** denote significant levels at 0.1, 0.05, and 0.01, respectively. P-value is presented in brackets.

Aside from the number of efficient DMUs, Table 2 also lists two dispersion measures, range and standard deviation, in order to represent the discriminative power of each combination. For Strategy-A, (O3, O4, I1) combination shows quite a large range and standard deviation in comparison with the full specification, indicating that the discriminative power of this combination is much higher than the baseline. The range of (O3, I1) combination is equal to one, because one DMU's graduate student is zero and the efficiency score is zero. However, the standard deviation of (O3, O4, I1) combination is relatively larger than (O3, I1) combination, implying that a two-output and one-input DEA model is more suitable for Strategy-A. Regarding Strategy-B, both two dispersion measures of (O1, O2, I2) combination are larger than the full specification. It means that this combination is also discriminative indeed. Notice that Strategy-A and Strategy-B come from distinct aspects so that we do not directly compare their discriminative powers with each other.

3.2 Hotel chain sample

The second dataset, provided by Ragsdale (2004), contains only eight DMUs (hotel chain) with two outputs and six inputs, showing a more serious problem of being low discriminative than the sample of academic departments above. For comparable purposes, the baseline adopts an input-oriented and constant returns to scale model. Moreover, in the second stage, we run a random forest algorithm and set $T = 100$ as well as $q = 3$.

Table 3 shows three permutation importance indices for each input and output variable. Because the number of DMUs is very small, we only consider a one-input and one-output combination herein. In terms of Strategy-A, we select one input and one output with the largest permutation importance; i.e., convenience (I4) and value (O2), respectively. This result is not found in the literature, because related studies only discuss the selection of inputs regardless of outputs, such as Jenkins and Anderson (2003) and Sharma and Yu (2015). However, Sharma and Yu (2015) suggest that selecting I2 and I4 presents the minimum reduction in efficiency scores, which is similar to our result (I2 is the second important input variable in random forest).

Table 3

Permutation importance for 8 hotel chain cases

Variable	PI_h	PI_h^{eff}	PI_h^{ineff}
O1: Satisfaction	0.0033	0.0033	0.0067
O2: Value	0.0917	0.1000	0.0833
I1: Service	0.0183	0.0217	0.0250
I2: Climate control	0.0250	0.0250	0.0300
I3: Price	-0.0117	-0.0150	-0.0133
I4: Value	0.0317	0.0333	0.0450
I5: Room comfort	-0.0083	-0.0133	-0.0100
I6: Food quality	-0.0033	-0.0100	0.0050

According to Strategy-B, our result suggests that the combination of price (I3) and satisfaction (O1) is the best choice with the lowest permutation importance indices. This result echoes the work of Wagner and Shimshak (2007) who consider the same combination as their core model for the hotel chain sample. Table 4 presents the correlation matrix among full variables, (O2, I4) combination, and (O1, I3) combination. Once again, the finding is in accordance with our consideration that Strategy-

A chooses a combination that uses fewer variables to generate a similar ranking of DMUs in comparison to the baseline. Although the correlation coefficient is moderately high, the p-value is not lower than the significance level due to the small sample size. As mentioned above, Strategy-A and Strategy-B come from quite different concepts, resulting in a negative correlation between the rankings determined by these two strategies. Moreover, the result indicates that both strategies exhibit higher discriminative power than the baseline DEA model.

Table 4
Spearman's rank correlation coefficients for three specifications

Specification	Full	(O2, I4)	(O1, I3)
Baseline	—		
Full			
Strategy-A (O2, I4)	0.5708 [0.1395]	—	
Strategy-B (O1, I3)	0.4059 [0.3184]	-0.2619 [0.5309]	—
No. of efficient DMUs	4	1	1
Range	0.143	0.884	0.242
Standard deviation	0.063	0.292	0.071

Note: P-value is presented in brackets.

In terms of dispersion measures shown in Table 4, the full model presents the lowest range and standard deviation among three specifications. Strategy-A, the (O2, I4) combination, represents the highest range and standard deviation with 0.884 and 0.292, respectively. Based on this specification, only one DMU is efficient and one has a moderate efficiency score (0.613). The other six DMUs have very low efficiency scores (less than 0.25), indicating that Strategy-A can determine a similar ranking and better discriminative power than the baseline. Strategy-B, the (O1, I3) combination, also has a larger range and standard deviation than the baseline. However, the differences of dispersion measures among (O1, I3) combinations and full variables is slight.

4. Data Augmentation for Robustness

The proposed two-stage approach is easy to implement, because some free and open source languages, such as R and Python, provide corresponding packages or libraries. For R users, the “randomForest” package is the most used tool to execute the random forest algorithm. For Python users, the most famous library is “scikit-learn (sklearn)”. These two languages also provide DEA-related tools, such as deaR in R and pyDEA in Python. In order to obtain robust results, this subsection further discusses two relevant issues as follows.

4.1 Small sample size and SMOTE

The number of DMUs is relatively small empirically. Although random forest performs well with a “small n and large p” dataset, there are no criteria for the minimum number of n . When n is quite small, like the aforementioned hotel chain sample ($n = 8$), the result of permutation importance may be unstable when different random seeds are used. One solution is to “produce” more DMUs, yet the difficulty is that we do not know the true data generating process (DGP) of the sample DMUs. To this end, we therefore adopt a synthetic minority oversampling technique (SMOTE), which is a widely used oversampling method in data science (Chawla et al., 2002), to synthesize more DMUs based on the observed data. In the following, we first describe the procedure of SMOTE and then demonstrate how to use SMOTE to generate efficient and inefficient DMUs from the baseline DEA model.

The essential goal of SMOTE is to deal with imbalanced datasets through oversampling/creating the data in the minority class. By considering each datapoint in the minority class, this technique randomly chooses one datapoint from the k -nearest minority neighbors in the feature space. A random number from $[0, 1]$ is then chosen, and the synthetic data can be represented as:

$$w_{syn} = w_i + (w_j - w_i) \cdot \gamma, \quad (4)$$

where w_i is the feature vector of the datapoint under consideration, w_j is the feature vector of a datapoint randomly selected from the k -nearest neighbors (kNN), and γ is a random number ranging from 0 to 1. Hence, the synthetic data (w_{syn}) can be deemed random points along the line segment between two observed data points.

The fact is that this paper does not deal with the problem of imbalanced data. We just aim to apply the concept of SMOTE to enlarge the training data and enhance the robustness of the random forest used in the proposed second stage. As mentioned above, each DMU is classified into an efficient or inefficient class via the baseline DEA model. Therefore, we run SMOTE twice - that is, running SMOTE for the efficient (inefficient) DMUs if the efficient (inefficient) class is the minority and then running SMOTE for the other class.

The double-SMOTE theoretically will not distort the classification (efficient or not) obtained from the baseline DEA model if we let k be equal to 1. According to Lemma 1 of Tone (2010), it is intuitive that the linear combination of any two inefficient DMUs is still inefficient regarding the baseline DEA. Hence, the synthetic data from inefficient DMUs maintain inefficiency no matter what number k is. Notice that, herein, w_i and w_j in Eq. (4) denote the input and output variables rather than the efficiency scores of two DMUs. Regarding the efficient class, it is also straightforward that a random point along the line segment between two efficient DMUs is efficient under CRS technology. However, under VRS technology, the argument holds if and only if k is equal to 1. We describe this by using a graphic example shown in Fig. 2.

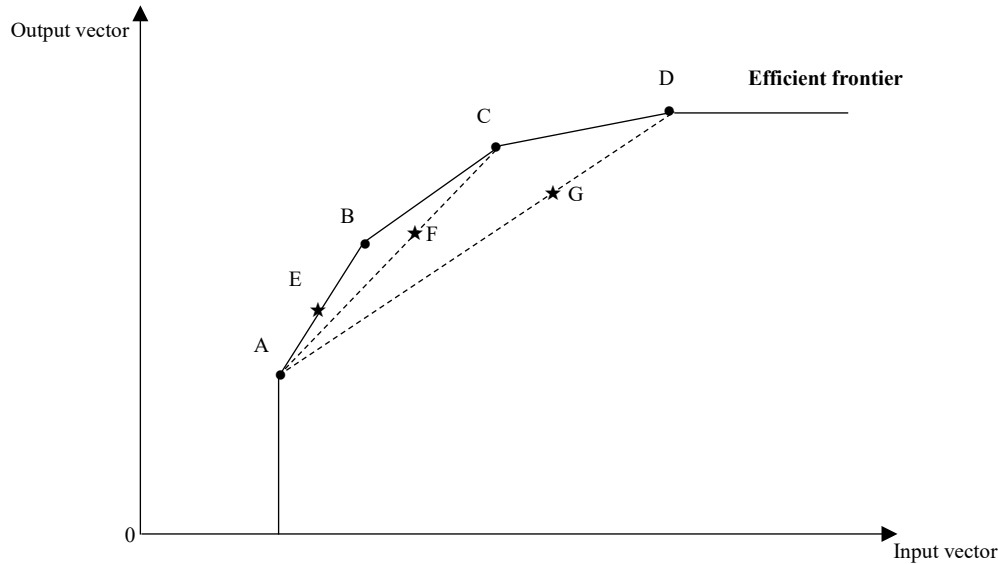


Fig. 2. A graphic example for the SMOTE algorithm in an efficient class

We assume four points (A, B, C, D) are the vertices that construct a piecewise linear frontier under VRS technology. When $k = 1$, each vertex point selects the closest point and randomly samples γ to obtain a synthetic point according to Eq. (4). For example, the nearest neighbor for point A is point B. Point E should be one of infinite random points along the line segment between points A and B. Because the efficient frontier is a piecewise linear function within a DEA model, the line segment between points A and B is part of the efficient frontier. Hence, point E on the line segment is also on the efficient frontier. When $k = 2$, each vertex point randomly chooses one of the 2-nearest neighbors and randomly samples γ to obtain a synthetic point. In terms of point A, the 2-nearest neighbors for point A are points B and C. If point C is selected by the SMOTE algorithm, then in this case any point on the line segment between points A and C (such as point F) is inefficient due to the convexity property of the DEA frontier. For the same reason, point G on the line segment between points A and D should be an inefficient point if D is chosen as one of 3-nearest neighbors of A. Therefore, if and only if $k = 1$, then the synthetic points preserve efficiency with respect to the efficient class.

We take the hotel chain sample as an example to confirm the robustness of our variable selection procedure. Both numbers of efficient and inefficient DMUs are four according to the baseline model (input-oriented CRS DEA as mentioned in section 3.2). This study then applies the SMOTE algorithm twice ($k = 1$ and random seed = 1234) and synthesizes 36 efficient and 36 inefficient DMUs. Hence, the total number of DMUs is 80 and we confirm that the double-SMOTE will not distort the classification. Furthermore, a random forest ($T = 500$, $q = 3$, and random seed = 1234) is executed to calculate the permutation importance indices for each input and output variable. Table 5 reports the results.

Table 5

Permutation importance for hotel chain cases after using the SMOTE algorithm

Variable	PI_h	PI_h^{eff}	PI_h^{ineff}
O1: Satisfaction	0.0342	0.0457	0.0241
O2: Value	0.2293	0.2408	0.2277
I1: Service	0.0916	0.0959	0.0915
I2: Climate control	0.0154	0.0145	0.0171
I3: Price	0.0067	0.0078	0.0055
I4: Value	0.0965	0.1168	0.0811
I5: Room comfort	0.0138	0.0106	0.0181
I6: Food quality	0.0045	0.0041	0.0048

As shown in Table 5, the importance ranking of output variables does not change after applying the double-SMOTE algorithm in comparison with Table 3. However, the least important input variable in Table 5 is I6 (food quality) with respect to the baseline DEA model, indicating that the combination for Strategy-B becomes (O1, I6) after synthesizing more synthetic

DMUs. Accordingly, the Spearman's rank correlation coefficient between full variables and (O1, I6) specifications is 0.3044, which is smaller than (O1, I3) mentioned in section 3.2. Moreover, the range and standard deviation of (O1, I6) specification are 0.873 and 0.272, implying that (O1, I6) combination has a much better discriminative power and is more appropriate for Strategy-B in comparison with (O1, I3) combination. This finding presents that a very small size of DMUs would cause a misleading selection result and supports that a robustness check using synthetic DMUs is needed.

4.2 Tomek Links for borderline DMUs

The second stage of our proposed procedure applies a random forest algorithm to classify efficient or inefficient DMUs determined in the first stage. If an inefficient DMU has a very high efficiency score (e.g., 0.99), then this DMU belongs to the inefficient class based on our proposed procedure. Hence, this subsection aims to discuss how to lower the effect of those inefficient DMUs (specific DMUs, hereafter) on the permutation importance obtained from a random forest algorithm.

We first consider that the permutation importance ranking is not distorted when the number of specific DMUs is small. As mentioned above, the random forest algorithm uses a bootstrapped sample to construct multiple decision trees and then obtains the final classification result. Theoretically, the random forest algorithm is relatively insensitive to a few specific DMUs due to the majority voting mechanism. However, when the number of specific DMUs is large (i.e., the discriminative power is extremely low), the permutation importance ranking is inevitably affected by those specific DMUs. To deal with this concern, we suggest using an undersampling method, Tomek links proposed by Tomek (1976), and dropping out some specific DMUs.

Tomek links aim to detect and remove borderline data points that may decrease the classification accuracy of any supervised learning technique. With respect to the terminology of this paper, the Tomek links method seeks the nearest neighbor for each DMU in the feature (outputs and inputs) space. Let DMU_i^{eff} and DMU_o^{ineff} be the i^{th} and o^{th} DMUs in the efficient and inefficient categories from the baseline DEA, respectively, while $d(DMU_i^{eff}, DMU_o^{ineff})$ is the Euclidean distance between these two DMUs. A pair $(DMU_i^{eff}, DMU_o^{ineff})$ is called a Tomek link if for any DMU_k (no matter what category) satisfies $d(DMU_i^{eff}, DMU_o^{ineff}) < d(DMU_i^{eff}, DMU_k)$ and $d(DMU_i^{eff}, DMU_o^{ineff}) < d(DMU_k, DMU_o^{ineff})$.

When all Tomek links are detected, we can remove all specific DMUs from the original dataset in order to enhance the classification accuracy of the random forest in the second stage. We further suggest that the SMOTE and Tomek links approaches can be applied in chorus if the number of DMUs in the original dataset is small. Taking the hotel chain sample as an example again, we perform the Tomek links approach first and remove one inefficient DMU. SMOTE is then used to generate more synthetic DMUs. Finally, the result of random forest finds that the suggested input-output combinations for Strategy-A and Strategy-B are totally consistent with the result in Table 5 and also consistent and robust even if we perform SMOTE first and then use Tomek links.

5. Conclusions

DEA is a powerful and well-accepted tool for evaluating DMUs' efficiency scores or performances in the management science field. However, empirically, researchers often confront a difficulty in selecting input and output variables. More specifically, DEA exhibits low discriminative power when the number of DMUs is insufficient and the number of input/output variables is large.

To deal with the above-mentioned problem, this paper proposes a two-stage variable selection approach that performs a baseline DEA model in the first stage and then applies a famous machine learning technique, random forest, in the second stage. The proposed selection approach aims to calculate the permutation importance for each input and output variable. We further define two distinct selection strategies according to different aspects. First, we use less input and output variables to obtain a similar order among DMUs' efficiency scores. In regards to this purpose, we suggest that researchers should choose input and output variables that have higher permutation importance. Second, contrarily, we suggest that researchers should select input and output variables with lower/negative permutation importance if the managers want to keep whatever variables can significantly affect the order in the baseline DEA. Both strategies decrease the number of efficient DMUs and increase the range and standard deviation among all DMUs' efficiency scores.

This paper also discusses two extended issues for robustness purposes. One is about the extremely small size of DMUs, and the other one concerns inefficient DMUs with very high efficiency scores. With respect to these two issues, we consider that the double-SMOTE and Tomek links approaches developed in the data science field offer a robust selection result indeed. This paper also introduces machine learning and data science techniques into DEA methodology. It is suggested that other advanced machine learning or artificial intelligence methods can improve the progress of DEA methodology in the future.

References

- Adler, N., & Golany, B. (2002). Including principal components weights to improve discrimination in data envelopment analysis. *Journal of the Operational Research Society*, 53, 985–991.
- Benítez-Peña, S., Bogetoft, P., & Morales, D. R. (2020). Feature selection in data envelopment analysis: A mathematical optimization approach. *Omega*, 96, Article 102068.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357.
- Chen, X., & Ishwaran, H. (2012). Random forests for genomic data analysis. *Genomics*, 99(6), 323–329.
- Díaz-Uriarte, R., & de Andrés, S. A. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7, Article 3.
- Dobrzanski, P. (2020). The efficiency of spending on R&D in Latin America region. *Applied Economics*, 52(46), 5020–5034.
- Ekiz, M. K., & Tuncer Şakar, C. (2020). A new DEA approach to fully rank DMUs with an application to MBA programs. *International Transactions in Operational Research*, 27(4), 1886–1910.
- Endri, E., Fatmawatie, N., Sugianto, S., Humairoh, H., Annas, M., & Wiwaha, A. (2022). Determinants of efficiency of Indonesian Islamic rural banks. *Decision Science Letters*, 11(4), 391–398.
- Jenkins, L., & Anderson, M. (2003). A multivariate statistical approach to reducing the number of variables in data envelopment analysis. *European Journal of Operational Research*, 147(1), 51–61.
- Karagiannis, R., & Karagiannis, G. (2023). Nonparametric estimates of price efficiency for the Greek infant milk market: Curing the curse of dimensionality with shannon entropy. *Economic Modelling*, 121, Article 106202.
- Lee, C. Y., & Cai, J. Y. (2020). LASSO variable selection in data envelopment analysis with small datasets. *Omega*, 91, Article 102019.
- Lee, K., & Choi, K. (2010). Cross redundancy and sensitivity in DEA models. *Journal of Productivity Analysis*, 34, 151–165.
- Li, Y., & Liang, L. (2010). A Shapley value index on the importance of variables in DEA models. *Expert Systems with Applications*, 37(9), 6287–6292.
- Nataraja, N. R., & Johnson, A. L. (2011). Guidelines for using variable selection techniques in data envelopment analysis. *European Journal of Operational Research*, 215(3), 662–669.
- Pastor, J. T., Ruiz, J. L., & Sirvent, I. (2002). A statistical test for nested radial DEA models. *Operations Research*, 50(4), 728–735.
- Ragsdale, C. T. (2004). *Spreadsheet modeling and decision analysis*. South-Western.
- Ruggiero, J. (2005). Impact assessment of input omission in DEA. *International Journal of Information Technology & Decision Making*, 4(3), 359–368.
- Sharma, M. J., & Jin, Y. S. (2015). Stepwise regression data envelopment analysis for variable reduction. *Applied Mathematics and Computation*, 253, 126–134.
- Strobl, C., Boulesteix, A. L., Kneib, T., Augustin, T., & Zeileis, A. (2008). Conditional variable importance for random forests. *BMC Bioinformatics*, 9, Article 307.
- Tomek, I. (1976). Two modifications of CNN. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-6(11), 769–772.
- Tone, K. (2010). Variations on the theme of slacks-based measure of efficiency in DEA. *European Journal of Operational Research*, 200(3), 901–907.
- Ueda, T., & Hoshiai, Y. (1997). Application of principal component analysis for parsimonious summarization of DEA inputs and/or outputs. *Journal of the Operations Research Society of Japan*, 40(4), 466–478.
- Verikas, A., Gelzinis, A., & Bacauskiene, M. (2011). Mining data with random forests: A survey and results of new tests. *Pattern Recognition*, 44(2), 330–349.
- Wagner, J. M., & Shimshak, D. G. (2007). Stepwise selection of variables in data envelopment analysis: Procedures and managerial perspectives. *European Journal of Operational Research*, 180(1), 57–67.



© 2025 by the authors; licensee Growing Science, Canada. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).