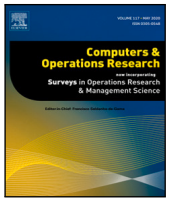
Computers & Operations Research 163 (2024) 106485

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor



Evaluating different methods for ranking inputs in the context of the performance assessment of decision making units: A machine learning approach

Daniel Valero-Carreras [a], Raul Moragues [a], Juan Aparicio [a,b,*], Nadia M. Guerrero [a]

[a] Center of Operations Research (CIO). Miguel Hernandez University of Elche (UMH), 03202 Elche (Alicante), Spain
[b] Valencian Graduate School and Research Network of Artificial Intelligence (valgrAI), Valencia, Spain

ARTICLE INFO

Keywords:
Data envelopment analysis
Support vector frontiers
Ranking inputs

ABSTRACT

In the context of assessing the performance of decision-making units (companies, institutions, etc.), it is important to know the contribution or importance of each input to the generation of products and services in the production process. Identifying the degree of relevance of each input is a challenge from both an applied and a methodological point of view, especially within the field of non-parametric techniques, such as Data Envelopment Analysis (DEA), where the mathematical expression of the production function associated with the data generating process is not specified. This means that there is no specific coefficient to be estimated for each input, which makes it difficult to determine a ranking of importance of this type of variable compared to parametric methods, where a target function dependent on some parameters must be previously specified. Within this challenging context associated with the non-parametric approach to estimating technical efficiency, in this paper, we adapt several methods for identifying the importance of features used together with the Support Vector Machine technique in order to determine an importance ranking of the inputs in a productive process. The different adaptations developed in this article are computationally checked through a simulated experiment.

1. Introduction

A problem which has attracted the interest of many researchers from a variety of perspectives is that of measuring the technical efficiency of firms in the context of a group of units which use the same types of inputs to produce comparable outputs. The main objective is to know which companies are making an optimal use of their resources and, for those that can improve, to obtain insights about how and how much to enhance their efficiency. Within this context, the technical efficiency can be measured in terms of a production process by which the firms transform inputs into outputs. In this regard, the technology is the set made up of all feasible input–output bundles. The technical efficiency of a firm can then be determined as the distance from the vector of inputs and outputs representing the firm to the border of the technology or production frontier.

Many proposals have been put forward in the literature to estimate technical efficiency. They are often classified into parametric and non-parametric techniques. In the first family, the production frontier is assumed to have a particular functional (parametric) form and the brunt of the work is in estimating the parameters involved in this functional form, and later statistical inference tools are applied to obtain information about the underlying production process. One well-known example is Stochastic Frontier Analysis, introduced in Aigner et al. (1977) and Meeusen and van Den Broeck (1977). On the other hand, non-parametric techniques are more data-driven and do not need to assume any functional form for the production frontier. Some of the most well-known non-parametric methods for estimating frontiers are Data Envelopment Analysis (DEA), see Charnes et al. (1978) and Banker et al. (1984) and Free Disposal Hull (FDH), introduced in Deprins et al. (1984). In this paper, we focus our attention on the non-parametric approach.

The non-parametric techniques require fewer assumptions about the probability distribution related to the production process. In particular, the Free Disposal Hull (FDH) estimator determines a production frontier relying only on enveloping all observed data from above with a surface which satisfies free disposability of inputs and outputs, yielding a stepwise production frontier using the data as a basis for the steps.

* Corresponding author at: Center of Operations Research (CIO). Miguel Hernandez University of Elche (UMH), 03202 Elche (Alicante), Spain.
E-mail addresses: dvalero@umh.es (D. Valero-Carreras), rmoragues@umh.es (R. Moragues), j.aparicio@umh.es (J. Aparicio), nguerrero@umh.es (N.M. Guerrero).

https://doi.org/10.1016/j.cor.2023.106485
Received 10 May 2023; Received in revised form 18 October 2023; Accepted 20 November 2023
Available online 25 November 2023

Free disposability of inputs and outputs only requires that, if a certain bundle of inputs and outputs is feasible, so is any other bundle which produces the same outputs with more inputs, as well as any bundle that produces fewer outputs with the original inputs. Furthermore, FDH satisfies the minimal extrapolation postulate, which chooses the smallest set among those sets which satisfy free disposability of inputs and outputs, and contains the observations. This results in an estimation which fits perfectly to the data, and may not generalize well to unseen data, i.e., in machine learning terms, the FDH estimator overfits the available data. This causes bad estimations of the actual production frontier, particularly when the sample size is not big enough. Other estimators, such as DEA, assume further microeconomic axioms such as convexity of the technology, and DEA presents the same problems as FDH with respect to overfitting.

One weakness of DEA, FDH and related methods is that the inclusion of additional variables only results in higher efficiency for the observed units, regardless of whether the variables in question are relevant or not to the production process. Model misspecification with inappropriate variables can result in inconsistent efficiency scores (Smith, 1997), and reinforces the need for identifying which variables are more important to the estimators. Preventing irrelevant variables from being included in the model is essential, since they affect the rates of convergence of the estimators, as pointed out by Simar and Wilson (2001). When considering the correlation between potential variables and estimated efficiency scores, Ruggiero (2005) observed that the addition of variables that are highly correlated to preliminary estimates of efficiency increases the quality of models, whereas those that are not correlated have the opposite effect in the models, resulting in increased prediction errors. These results are in line with the statistical learning theory, where the choice of which variables to include as predictors is extremely important to the quality of the models, see e.g. James et al. (2013). In particular, when the number of variables is large compared to the number of available data, FDH and DEA characterize a large number of units as efficient, and cannot distinguish among them. This problem is often called the curse of dimensionality, see for example, Charles et al. (2019). These arguments indicate the need for algorithms that measure the relative importance of variables in a production process (for example, the inputs as explanatory variables of the output), and ranking them accordingly, so that better models can be constructed which balance the complexity of the models with the quality of the estimators.

This has led to lots of attention in the area towards the problem of model misspecification, and the selection of the variables that are most relevant to the production process. One of the first proposals used the Efficiency Contribution Measure (ECM) of each variable (Pastor et al., 2002), via a hypothesis test determining whether an input is relevant or not. Other approaches used regression-based analysis, such as Ruggiero (2005), bootstrapping methodology (Simar and Wilson, 2000a), or enriched the DEA optimization programs using binary variables to model which inputs are selected (Peyrache et al., 2020; Benítez-Peña et al., 2020), as well as statistical methods (Araújo et al., 2014). Another family of approaches performs aggregations of the available variables, creating new variables which inform of the characteristics of the data, but losing interpretability. Among these, we can highlight various uses of Principal Component Analysis in Ueda and Hoshiai (1997) and Adler and Golany (2002), or using Akaike's information criteria in Li et al. (2017). If further information is available, such as price-based information, there are contributions such as Zelenyuk (2020) for price-based aggregation of variables. In addition, there have been comparisons of these methods in the literature, e.g. Sirvent et al. (2005), Adler and Yazhemsky (2010), Nataraja and Johnson (2011), and Chen and Han (2021).

Regarding the inherent overfitting problem present in DEA and FDH, numerous papers have tried to solve this drawback by endowing FDH and DEA with inferential capability. For example Banker and Maindiratta (1992) and Banker (1993) showed that DEA could be interpreted as a Maximum Likelihood estimator, and Simar and Wilson (1998, 2000a) and Simar and Wilson (2000b) proposed a methodology to establish confidence intervals for the efficiency score of each observation by adapting the bootstrapping methodology first introduced by Efron (1979) to the context of DEA and FDH. Another recent stream in the literature on efficiency frontiers is already starting to make use of the similarities of DEA-like methods and machine learning techniques. For example, Kuosmanen and Johnson (2010, 2017) showed that DEA could be interpreted as non-parametric least-squares regression subject to particular constraints on the shape of the frontier and sign-constraints on the residuals, leading to the introduction of Corrected Concave Non-parametric Least Squares. Another contribution in this direction is Parmeter and Racine (2013), where the authors use non-parametric kernel frontier estimators providing a smooth approximation to the production frontier. Other techniques being adapted in this research line are, for example, the use of quadratic and cubic splines with shape constraints introduced in Daouia et al. (2016). Olesen and Ruggiero (2022) have also adapted hinging hyperplanes to estimate production functions in a flexible non-parametric representation of the production function. More recently, the links between machine learning have been made more explicit via the adaptation of classical algorithms based on decision trees such as Classification And Regression Trees (CART) (Breiman et al., 1984) and Random Forest (Breiman, 2001) being adapted in Esteve et al. (2020) and Esteve et al. (2023), or probabilistic regression trees in Tsionas (2022). Guerrero et al. (2022) have modeled the problem of technical efficiency measurement as the implementation of the Structural Risk Minimization principle, as introduced in Vapnik (1991). Other approaches have adapted Support Vector Regression (Valero-Carreras et al., 2021, 2022; Liao et al., 2023). In particular, Valero-Carreras et al. (2021) proposed Support Vector Frontiers (SVF), a procedure based on Support Vector Regression for the estimation of technical efficiency. In this paper, we consider SVF as the basis for our contribution, and endow this technique with input ranking procedures.

The problem of ranking the importance of variables, as well as that of identifying which of the available features are most relevant to obtain sound models, is an important task not only affecting the measurement of efficiency, but it is also an area of high interest in any data-based modeling problem, such as those involving machine learning methodologies. In both contexts, it refers to measuring how important each variable is to predict the output values, according to the effect that the presence or absence of a variable has on the predictions of a model. In particular, in nonparametric methodologies, where by their nature we do not have p-values associated with statistical significance of the predictors, these measures play an analogous role. It could be expected that as the number of predictors increases, the accuracy of the fitted models should improve, but this is not always true. Furthermore, it will eventually result in overfitting, where the model may be too attuned to the particular data with which it was trained and not to unseen data, thus performing very well on this data, but failing to generalize to unseen data which was not used in the fitting process. This is equivalent to saying that the frontier method is not able to claim anything about the Data Generating Process (DGP) that is behind the data. Measuring the importance of variables and ranking them accordingly is a way to distinguish among them and identify which variables are more appropriate for describing the underlying DGP. Thus, variable ranking methods can serve as a useful complement to decision makers to value the most important variables and identify which variables should be included or could be discarded from the production process. These processes can enable practitioners to obtain better models which use those variables that are more relevant to the process being studied. Additionally, the determination of a ranking of inputs is a way of opening the black box associated with the production process. It is a manner of getting information about which inputs are more relevant to explain the observed level of output of the assessed companies (Decision Making Units in a general production framework).

For this reason, a lot of attention has been paid in the area of machine learning to the problem of model specification, in identifying which variables are more important to a given process, and to rank them accordingly. These serve to obtain less complex estimators, which usually lead to a better generalization, as well as reducing the computational resources used Guyon et al. (2002). In doing so, they yield insights into the underlying process that generated the data (Louppe et al., 2013).

In general, the methodologies used to determine the relevance of the variables in the data set can be classified into three main categories: filter, wrapper and embedded methods. Filter methods use a statistical criterion to establish the relevance of a feature before executing the learning process, for example, the Fisher Criterion Score. They use the structure of the data, not necessarily that of the estimator. Wrapper methods, on the other hand, evaluate a specific subset of variables by training and testing a specific model tailored to a specific problem. They evaluate the variations in a given model and measure which of the features available are more important to its performance. Embedded methods are mechanisms, built-in in the learning algorithm, which perform feature selection, for example the pruning algorithm for decision trees (Breiman et al., 1984).

With the objective of endowing the recent methodology for technical efficiency measurement known as Support Vector Frontiers (SVF) with capability for ranking the importance of inputs, and given that SVF is based on Support Vector Machines by Vapnik (1998), we focus our attention on the main approaches in the literature on machine learning that combine SVM and the determination of the relevance of variables. In particular, Guyon et al. (2002) proposed the most widely-used method for variable selection using SVM, known as SVM-Recursive Feature Elimination (SVM-RFE). This methodology makes some simplifying assumptions to reduce the computation in large databases, and uses the variation of the objective function of the dual program as a measure of importance. Another methodology, introduced by Alonso-Atienza et al. (2012), uses bootstrapping to rank input variables. Sanz et al. (2018) proposed a methodology for evaluating the estimator on a set of pseudosamples in order to evaluate the importance of features. Other authors like Maldonado and Weber (2009) and Chen and Lin (2006) have suggested filter methods for feature selection based upon the combination of SVM and the metric F-Score. Meanwhile, other authors have proposed embedded methods like Becker et al. (2009) and Becker et al. (2011), who developed a penalized version of the SVM.

In this paper, we adapt the SVM-methodologies introduced by Alonso-Atienza et al. (2012) using bootstrapping, the simplified approach in Guyon et al. (2002), and the pseudosamples approach of Sanz et al. (2018) to the context of Support Vector Frontiers by Valero-Carreras et al. (2021) and Valero-Carreras et al. (2022) for the ranking of the importance of the input variables to the efficiency of a production process, allowing also the selection of the most relevant inputs when avoiding misspecification of the model is one of the research objectives. We also adapt a methodology based on the random permutation of the values of a variable inspired by Breiman (2001).

The paper is structured as follows. Section 2 introduces the topics of FDH and SVF, and describes the methods for ranking variables in SVM (based on Permutation Feature Importance, Bootstrap Resampling, Pseudosamples, and Dual Objective Variation). We adapt these methods to the context of Support Vector Frontiers and Free Disposal Hull in Section 3. The performance of these algorithms is evaluated through a computational experience in Section 4, where they are compared in a variety of scenarios. Finally, Section 5 presents some conclusions.

2. Background

This section provides a brief review of one of the most widely used methods for estimating production frontiers, the Free Disposal Hull (FDH, Deprins et al., 1984), and an adaptation of the Support Vector Regression machine learning technique (SVR) for frontier estimation, such as Support Vector Frontiers (SVF). Also, in the second half of this section, a review of different techniques using feature selection and ranking using Support Vector Machines (SVM) is made.

2.1. Frontier estimation

The measurement of the efficiency of companies, organizations, etc., is a topic of interest for economists when establishing their improvement strategies. Beginning with Farrell (1957), different techniques have been developed for the estimation of production frontiers that allow establishing a series of measures for the evaluation of efficiency. In this section, a small review is made of two techniques that are relevant in our subsequent investigation: FDH and SVF.

2.1.1. Free Disposal Hull

In microeconomics, given a data set $\Omega = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$, where n is the number of Decision Making Units (DMUs), it is assumed that each company consumes a series of resources $\mathbf{x}_i = (x_i^{(1)}, \dots, x_i^{(m)}) \in \mathbb{R}_+^m$ to generate a series of products and/or services $\mathbf{y}_i = (y_i^{(1)}, \dots, y_i^{(s)}) \in \mathbb{R}_+^s$. The technical efficiency of a DMU can be calculated based on the frontier of the production possibility set or technology T, defined as:

$$T = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}_+^{m+s} : \mathbf{x} \text{ can produce } \mathbf{y}\} \tag{1}$$

The technology frontier, also called the efficient frontier of T, is defined as:

$$\delta(T) := \{(\mathbf{x}, \mathbf{y}) \in T : \hat{\mathbf{x}} < \mathbf{x}, \hat{\mathbf{y}} > \mathbf{y} \Rightarrow (\hat{\mathbf{x}}, \hat{\mathbf{y}}) \notin T\} \tag{2}$$

Regarding this technology, different properties are assumed, such as the free disposability of inputs and outputs and deterministicness. Free disposability is defined as follows: if $(\mathbf{x}, \mathbf{y}) \in T$, then $(\mathbf{x}', \mathbf{y}') \in T$ whenever $\mathbf{x}' \geq \mathbf{x}$ and $\mathbf{y}' \leq \mathbf{y}$. On the other hand, deterministicness implies that a production frontier must envelop the entire data set. Given a technology, different measures can be used to calculate the degree of technical efficiency of a DMU: see the case of the input-oriented or output-oriented radial measures (Banker et al., 1984) or the Directional Distance Function (Chambers et al., 1998), to mention just a few.

Free Disposal Hull is a non-parametric technique developed by Deprins et al. (1984) used for the estimation of a technology, from a data sample, which allows the evaluation of the technical efficiency. This method generates a step-wise frontier that complies with deterministicness, the principle of free disposability and the principle of minimal extrapolation (i.e., among all the possible technologies, we select the smallest one). The FDH technology is defined as follows:

$$\hat{T}_{FDH} = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}_+^{m+s} : \mathbf{y} \leq \mathbf{y}_i, \mathbf{x} \geq \mathbf{x}_i, i = 1, \dots, n\} \tag{3}$$

This technique is very interesting (Kerstens and Van De Woestyne, 2014; Kerstens et al., 2022), since it is based on few assumptions, but, due to the principle of minimal extrapolation, it causes overfitting to the data on the estimated frontiers. This weakness has necessitated the introduction of other techniques in an attempt to solve this problem.

2.1.2. Support Vector Frontiers

Support Vector Frontiers (Valero-Carreras et al., 2021, 2022) is an adaptation of the machine learning method Support Vector Regression (SVR), for frontier estimation. The idea behind Support Vector Frontiers is to adapt SVR to the estimation of production functions. SVR estimates the values of a response variable (output) as a function of a set of covariables (inputs). It allows the estimation of a variety of functions via a transformation of the input space ϕ into a higher dimensional space. The standard SVR estimation estimates an "average" of the output values and this is adapted by SVF so that the estimated outputs envelop the observed data from above. Furthermore, restrictions are added which ensure that the estimated production function satisfies free disposability of inputs and outputs. This method is introduced in Valero-Carreras et al. (2021) in the case of a single output, and is extended to the multi-output case in Valero-Carreras et al. (2022). This non-parametric method is characterized by creating technologies that

obtain a production frontier, complying with the axioms of deterministicness and free disposability, while reducing the overfitting of the frontier to the data.

In particular, Model (4), from Valero-Carreras et al. (2021, 2022), represents the program to be solved to obtain the production frontier in a multi-input multi-output scenario. The objective function tries to balance the compromise between minimizing the empirical error and the generalization error through a hyperparameter called C, which is determined through cross-validation. Constraint (4a) ensures that the axiom of deterministicness is fulfilled. Constraint (4b) relates the empirical error and the margin (ϵ) of the problem. The margin is another hyper-parameter that is obtained by cross validation and helps to give robustness to the model. Constraint (4c) guarantees the postulate of free disposability. Finally, the last constraint (4d), ensures that the errors are non-negative.

$$\min_{\mathbf{w}^{(r)},\xi_i^{(r)},b_0} \sum_{r=1}^{s} \left\|\mathbf{w}^{(r)}\right\|_2^2 + C \sum_{r=1}^{s}\sum_{i=1}^{n} \xi_i^{(r)} \tag{4}$$

$$s.t. \quad \mathbf{w}^{(r)}\boldsymbol{\phi}_{SVF}^{G}(\mathbf{x}_i) + b_0 - y_i^{(r)} \geq 0, \qquad i = 1,\dots,n,\ r = 1,\dots,s \tag{4a}$$

$$\mathbf{w}^{(r)}\boldsymbol{\phi}_{SVF}^{G}(\mathbf{x}_i) + b_0 - y_i^{(r)} \leq \epsilon + \xi_i^{(r)}, \qquad i = 1,\dots,n,\ r = 1,\dots,s \tag{4b}$$

$$\mathbf{w}^{(r)} \geq 0 \qquad r = 1,\dots,s \tag{4c}$$

$$\xi_i^{(r)} \geq 0, \qquad i = 1,\dots,n,\ r = 1,\dots,s \tag{4d}$$

To guarantee that the frontier obtained is step-wise, a transformation of the data set ($\boldsymbol{\phi}_{SVF}^{G}$) is used Valero-Carreras et al. (2021). This transformation is a vector of ones and zeros that is obtained by analyzing the position of an observation $\mathbf{x}_i$ based on a grid G. This grid is made up of cells $C_{l_1\dots l_m}$, where $l_j \in \{1,\dots,d\}$, that are delimited by a set of nodes. The endpoints of the grid are defined by the minimum and maximum values of each input variable, and the nodes of the grid are equidistant partitions in the space of the inputs defined by a hyperparameter d that is obtained through cross-validation. This transformation function is invariant to changes of scales in the inputs, so that the model is also invariant to rescaling.

The optimal hyperparameters (C^*, ϵ^* and d^*) are obtained through a cross-validation process. For this, it is necessary to establish a grid of hyperparameters to test the best combination of them. Once the best hyperparameters have been obtained and the model has been solved, an optimal solution of the model (4), ($\mathbf{w}^*, \xi^*, b_0^*$), is obtained.[1] The estimation of the technology associated with the SVF technique is:

$$\hat{T}_{\text{SVF}} = \{(\mathbf{x},\mathbf{y}) \in \mathbb{R}_+^{m+s} : y^{(r)} \leq \mathbf{w}^{(r)}\boldsymbol{\phi}_{\text{SVF}}^{G}(\mathbf{x}) + b_0, r = 1,\dots,s\}. \tag{5}$$

As can be seen in Fig. 1, where the solid line represents the production frontier generated by the SVF method, the dashed line the frontier generated by FDH and the dotted line a simulated theoretical frontier, SVF generates a step-wise frontier that does not overfit the data as much as that of FDH and is closer to the theoretical frontier. This is due to the generalizing power of SVM, which makes it possible to effectively predict the value of an observation that is outside the training set and to the fact that it does not comply with the principle of minimal extrapolation, which forces other techniques such as the FDH to fit their estimated frontiers as closely as possible to the data.

Both SVF and FDH estimate stepwise production functions. FDH is a fast method, fits perfectly to the data and may not generalize well to unseen data (Esteve et al., 2020). Another characteristic of FDH and DEA is that, given an input profile, these methods do not produce a unique prediction of a vector of outputs. Instead, these methods estimate a whole set of producible output vectors. This difficulty is particularly notable in the multi-output case, but it can be overcome when there is a single output.

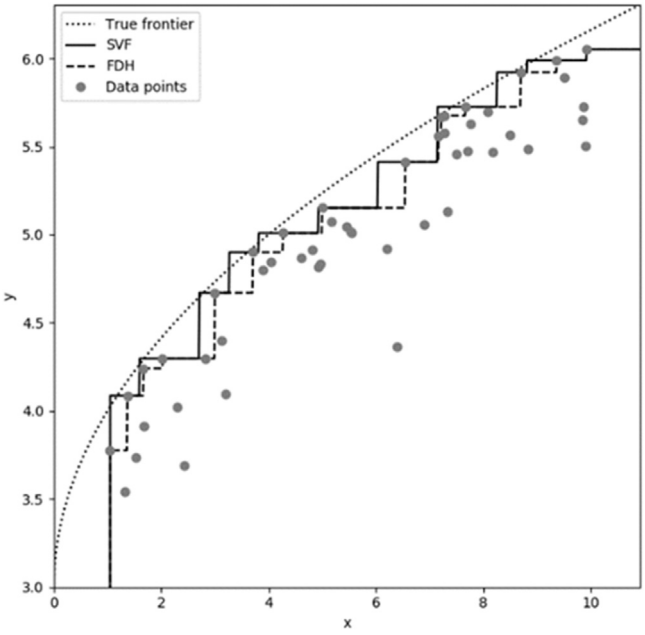


Fig. 1. Comparison between the frontiers generated by FDH and SVF.

We can observe in the simulation results in Valero-Carreras et al. (2021) that SVF estimates a production function which is closer to the unobserved theoretical frontier than that estimated by FDH. In particular, the MSE between the estimated and true frontiers is improved by between 14.3% and 34.4% when estimated by SVF compared to the FDH estimation. These improvement values are increased to improvements ranging from 38.7% to 78.4% on average when comparing the Convexified SVF (CSVF) with DEA. Additionally, the bias is reduced in these experiments by between 8% and 19.9% when comparing SVF with FDH. Furthermore, when comparing CSVF with DEA, these improvements range from 27.3% to 56.1%. In a multi-output computational experiment in Valero-Carreras et al. (2022), similar results were obtained, with SVF outperforming FDH with improvements in MSE of between 24% and 50%. Additionally, CSVF outperforms DEA with enhancements in MSE ranging from 16% to 24% in these experiments. These results indicate that SVF can estimate frontiers which more closely resemble the theoretical frontier, allowing for better out-of-sample estimations. Thus, it can be of particular relevance when only a sample of the data is available and the objective is to infer properties from the underlying production process, instead of only the particular units available. However, a limitation of SVF is its computational complexity. We refer the reader to Valero-Carreras et al. (2021, 2022) for further details about these comparisons. We summarize the characteristics of both SVF and FDH in Table 1.

2.2. Ranking and feature selection with Support Vector Machines

One of the main problems of machine learning is to identify those predictor variables that are really important to improve the accuracy of the model when predicting the values of a response variable. This is because the inclusion of certain features in the problem can cause an overfitting of the model or the appearance of unnecessary noise. In addition, the use of thousands of variables can be computationally unapproachable despite advances in computation and memory storage.

[1] Since the estimate of the boundary passes through the origin of coordinates, any solution of the SVF model will have $b_0^* = 0$.

Table 1
Comparison between SVF and FDH.

Method	SVF	FDH
Merits	Adds robustness ϵ-insensitive efficiency Reduces overfitting Estimates the theoretical production frontier	Fast and straightforward
Weaknesses	Computational cost	Fits perfectly to the available data May not generalize well to unseen data Curse of dimensionality
Applicability	Can generalize to unseen data Can estimate out-of-sample	Wide range of application
Solving difficulty	High computational complexity Transformation function is costly Cross-validation too	Low difficulty Mixed Integer Linear Programming

Due to this, different feature selection and ranking techniques have been implemented using machine learning methods, such as SVM. In this section, a literature review is made of some techniques that have been developed to be used in the context of SVM-based models and that are going to be adapted to the SVF context in Section 3.

2.2.1. SVM-Permutation Feature Importance

Support Vector Machine-Permutation Feature Importance (SVM-PFI) is a method for establishing the importance of variables in a data set, which was introduced by Breiman (2001) explains the fundamentals of the Random Forest technique and argues that in numerous data analyses it is necessary to know the interaction that occurs between the variables in order to improve the predictive accuracy of the model. To do this, he establishes a method of evaluating variables based on the decrease in the precision of the prediction.

From a data set $\mathbf{V} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}$ represents a vector of predictor variables and y is a response variable of the problem, the first step is to randomly select a training set $\mathbf{V}_{train}$ and a test set $\mathbf{V}_{test}$ from the original dataset. Afterwards, the model is trained with the $\mathbf{V}_{train}$ data using the predictive technique that is going to be used, such as Random Forest or SVM, and the error E_0 is calculated with $\mathbf{V}_{test}$. One of the metrics that can be used is the mean square error (MSE), $E_0 = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$ where $\hat{y}_i$ is the model prediction and y_i the observed value. Next, for each u-th variable in the problem, the data values of this u-th variable are randomly shuffled creating a new data set, $\mathbf{V}^u_{train}$, and the model is trained. As a test, the remaining data set $\mathbf{V}_{test}$ is used and the error is calculated as: $E_u = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i^{(u)} - y_i)^2$, where $\hat{y}_i^{(u)}$ is the prediction of the model trained on $\mathbf{V}^u_{train}$.

The variation of the model prediction error with respect to the model error with the original data, $\Delta E_u = \frac{E_u - E_0}{E_0}$, is the tool used to determine the predictive power of the variable.

Finally, a ranking is established with the variation of errors obtained. The variables associated with a greater reduction in the prediction error of the model are those that are most important.

2.2.2. SVM-Bootstrap Resampling

Support Vector Machine-Bootstrap Resampling (SVM-BR) is a technique developed by Alonso-Atienza et al. (2012), used for feature selection although it can also be used for ranking tasks. This method is based on the combination of SVM with Bootstrap resampling techniques (Efron, 1979).

Specifically, the SVM-BR approach is based on the use of Bootstrap to measure the variation of the SVM performance when it is trained with the complete data set and, on the other hand, when it is trained with a subsample in which the number of characteristics of the problem has been reduced. By solving the optimization program associated with the SVM, we obtain some coefficients $(\mathbf{w}^*, b_0^*)$ that depend on the data set $\mathbf{V}$ and some hyperparameters.

A bootstrap sample $\mathbf{V}' = \{(\mathbf{x}'_1, y'_1), \dots, (\mathbf{x}'_n, y'_n)\}$ is a new dataset created by selecting at random observations of $\mathbf{V}$ with replacement. A sample $\mathbf{V}$ can be seen in resampling terms as: $\mathbf{V}' = (\mathbf{V}'_{in}, \mathbf{V}'_{out})$, where $\mathbf{V}'_{in}$ and $\mathbf{V}'_{out}$ are the subset of samples included and excluded in the resampling. The coefficients $(\mathbf{w}^*, b_0^*)$ are obtained by solving the corresponding optimization program using only the information contained in the data set $\mathbf{V}'_{in}$, while the prediction error (based on the MSE) R^* is calculated using the data $\mathbf{V}'_{out}$.

Given a set of B independent bootstrap samples, $\{\mathbf{V}'(1), \dots, \mathbf{V}'(B)\}$, we can calculate the risk density function from the histogram generated by the errors of all the samples $R^*(b)$, where $b = 1, \dots, B$.

Alonso-Atienza et al. (2012) use a reduced subset $\mathbf{V}_u$ of observations in which the variable u is removed and they create a bootstrap sample with their respective subsets: $\mathbf{V}'_{u,in}$ and $\mathbf{V}'_{u,out}$. From there, the error is calculated as: $R^*_u(b)$. Finally, they use the difference between the error of the complete model with the model when removing the variable u, giving rise to the measurement: $\Delta R^*_u(b) = R^*_u(b) - R^*(b)$ where a higher value of $\Delta R^*_u(b)$, indicates a greater importance of the variable u.

2.2.3. SVM-RFE Pseudosamples

SVM-RFE Pseudosamples is a method proposed by Sanz et al. (2018) that aims to evaluate the importance of the variables of a dataset based on the variation of the median absolute deviation (MAD) when a pseudosample of the data is created. This technique consists of an extension of the ideas proposed by Postma et al. (2011) and Krooshof et al. (2010), which evaluate the importance of the variables through the transformations of the data that are produced when using SVR.

In order to obtain a ranking of the importance of the predictor variables of the model, an SVM model is first trained with the entire initial set of data, $\mathbf{V}$. From there, optimal coefficients $(\mathbf{w}^*, b_0^*)$ and hyperparameters are obtained. Next, m data subsamples $(\mathbf{V}_m)$ are created, where m is equal to the number of predictor variables in the dataset. These matrices of size $q \times p$ will be formed by some pseudosamples in which the values of the column of the variable to be evaluated will be formed by the q quantiles of that column. The rest of the values in the other columns will be made up of the mean or median of these variables, although since the data is considered to be normalized they are replaced by 0.

Once the pseudosamples have been created, the solutions of the initially trained model $(\mathbf{w}^*, b_0^*)$ are used to obtain the prediction of the response variable associated with each of them $(\hat{y}_{q_p})$. Then, the variability of the pseudo-sample predictions is calculated through the median absolute deviation: $\mathrm{MAD}_p = median(|\hat{y}_{q_p} - median(y_p)|)$.

Finally, the variables that have a lower MAD value will be eliminated as they are considered to have a lower contribution to the model. The process is repeated until a certain stopping criterion is met.

2.2.4. SVM-RFE Dual Objective Variation

Another approach in the literature, introduced in Guyon et al. (2002), involves the change in the dual objective function when a variable is either included or excluded in the production process.

The dual problem of an SVM has an objective value J, and this method computes the change in the estimated cost function of the

dual problem caused by removing one feature, i.e., by forcing every term involving the variable in question to be 0, while keeping the dual variables constant. We first solve the original model with cross-validation in order to obtain the best hyperparameters for the model, before focusing our attention on the dual.

In order to make this method tractable computationally, instead of solving the dual model with each feature removed, the problem, in its dual form, is solved just once per step with all the features, obtaining values for the dual variables and the objective value J of the dual problem. Then, the solution for the dual variables obtained is kept constant and these values are used to calculate the effect on J of removing a single feature u, denoted by $J(u)$. The feature u which results in the smallest change in the objective value $DJ(u)$ is deemed the least important in the ranking, and the process is iterated without this variable, until every variable is ranked.

3. Determining a ranking of inputs in a production context through Support Vector Frontiers

This section shows a series of methods developed to determine the ranking of importance of the inputs in a production process. To do this, the techniques shown in Section 2.2 have been adapted to the context of frontier estimation through the SVF and FDH techniques.

3.1. SVF-Permutation Feature Importance

Support Vector Frontiers-Permutation Feature Importance (SVF-PFI) is an adaptation through SVF of the method established by Breiman (2001) and used by other authors such as Esteve et al. (2023). With this algorithm, the predictive power of the SVFs in predictive contexts is combined with the variable ranking method established by Breiman. This technique is based on the study of the change in the prediction of a model when the value of the variables of the original data set $\mathbf{V}$ is modified.

As demonstrated in Algorithm 1, a training set $\mathbf{V}_{train}$ and a test set $\mathbf{V}_{test}$ are first selected from the original dataset. Next, a model is trained and an optimal solution $(\mathbf{w}_0^*, b_0^*)$[2] of (4) is obtained. Then, the error of the initial model is calculated as $E_0 = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$, where $\hat{y}_i = \mathbf{w}_0^* \boldsymbol{\phi}_{SVF}^G(\mathbf{x}_i) + b_0^*$.

Afterwards, a copy of the previously selected training set is made and for each variable the data is shuffled creating a new dataset: $\mathbf{V}_{train}^u$. An SVF model is trained, the prediction $(y^{(u)})$ is made using the test set $(\mathbf{V}_{test})$, and its error is calculated as $E_u = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i^{(u)} - y_i)^2$, $\forall u = 1, \ldots, m$.

Finally, we establish a score that indicates the percentage of error variation that occurs when permuting the variable: $S_u = 100 \cdot \frac{E_u - E_0}{E_0}$. Those variables that have a higher score indicate that they are more important, since variations in their values cause large disturbances in the model's prediction.

3.2. SVF-Bootstrap Resampling

Based on the ideas of Alonso-Atienza et al. (2012), we have developed a method that uses Bootstrap Resampling for feature selection and ranking adapted for SVF.

From an initial data set $\mathbf{V}$, bootstrap resampling is applied to create B subsamples $\mathbf{V}'(b)$. Each of the subsamples $\mathbf{V}'(b)$ is composed of a set of training data, those data that have been included in the resampling $(\mathbf{V}'_{in})$; and a set of test data, those that have been excluded from resampling $(\mathbf{V}'_{out})$.

Once the subsample is generated, the training set is used to train an SVF model. This model will generate optimal hyperparameters: (C^*, ϵ^*, d^*) and an optimal solution: $(\mathbf{w}^*, b_0^*)$. The test set is used to obtain the subsample error: $R^*(\mathbf{w}^*, \mathbf{V}'_{out})$.

Next, the method is iterated eliminating one by one each of the inputs of the problem from the data set $\mathbf{V}'$, generating the subset $\mathbf{V}'_u$. For the test set resulting from eliminating the variable u, the prediction error is calculated.

Next, the error of the variable is calculated by subtracting the error of the solution from the original set, $R^*(b)$, to the value of the error of the solution of the set without the variable u, $R_u^*(b)$.

Once all the errors of all the variables of all the pseudosamples have been obtained, the error values obtained are classified according to their sign. If the number of times in which the value is positive is large, this indicates that the variable is very important in the dataset. If, on the contrary, the value of the error of that variable is always negative, it will indicate that the examined variable is not useful or has little importance.

Once the B pseudosamples have been determined, the relative frequency in which each variable is positive is calculated and the one with the highest value is considered to be the most important. Algorithm 2 describes the steps of this method.

Algorithm 1 SVF-PFI algorithm implementation

```
procedure GET_RANKING_PFI(V,C,ϵ,d)
    ranking, S ← [ ]
    V_train, V_test ← create_train_test(V)
    w*_0, C*, ϵ*, d* ← SVF(V_train, C, ϵ, d)
    E_0 ← calculate_error(V_test, w*_0)
    for u ← 1 to n_inp do
        V^u_train ← permute_var(V_train, u)
        w*_u ← SVF(V^u_train)
        E_u ← calculate_error(V_test, w*_u)
        S_u ← 100 · (E_u − E_0)/E_0
    end for
    ranking = order_score_asc(S)
    return ranking
end procedure
```

3.3. SVF-Pseudosamples

SVF-Pseudosamples is the methodology that we propose for the ranking of inputs that we develop based on the ideas of Sanz et al. (2018). This feature ranking technique establishes an order of the feature importance based on the value of the Median Absolute Deviation (MAD). The steps followed are shown in Algorithm 3.

First, a stopping criterion is established. In our case, since our objective is to obtain a complete ranking, we must define as a stopping criterion that the number of inputs in the problem is greater than 1. In this way, all the inputs are evaluated and the method obtains a complete ranking of the inputs. As long as the stopping criterion is not met, an SVF model is executed with the set of variables available at that moment. Also, an optimal solution of the model is obtained $(\mathbf{w}^*, b_0^*)$. A matrix is created with the pseudosamples of the variable to be evaluated $(\mathbf{V}_u)$. To do this, we follow the method established by Sanz et al. (2018) that uses q quantiles of the variable to be evaluated and leaves the other columns with zeros or the mean of the column, in case the data is not normalized. We calculate its prediction $(\hat{y}_{uq})$ and evaluate the value of the MAD as $\text{MAD}_u = \text{median}(|\hat{y}_{uq} - \text{median}(y_u)|)$, where y_u is the value of the outputs in the original dataset. The variable with the lowest MAD is eliminated from the data set and the same process is repeated until the stopping criterion is met.

Finally, we obtain a ranking of all the inputs that indicates the importance of each of them in the model.

[2] In the SVF case, as noted in Section 2.1.2, $b_0^* = 0$ because the frontier estimator passes through the origin of coordinates.

Algorithm 2 SVF-BR algorithm implementation

procedure GET_RANKING_SVF_BR(V,C,ϵ,d)
  ranking ← [ ]
  E ← [ ][ ]
  for $b \leftarrow 1$ to B **do**
    $\mathbf{V}'_{in}, \mathbf{V}'_{out}$ ← create_bootstrap_sample($\mathbf{V}$)
    $\mathbf{w}^*, C^*, \epsilon^*, d^*$ ← SVF($\mathbf{V}'_{in}, C, \epsilon, d$)
    R^* ← calculate_error($\mathbf{w}^*, \mathbf{V}'_{out}$)
    for $u \leftarrow 1$ to m **do**
      $\mathbf{V}'_{u,out}$ ← drop_var($\mathbf{V}'_{out}, b, u$)
      R^*_u ← calculate_error($\mathbf{V}'_{u,out}, \mathbf{w}^*$)
      $E_{bu} \leftarrow R^*_u - R^*$
    end for
  end for
  p ← [ ]
  for $u \leftarrow 1$ to m **do**
    pos ← 0
    neg ← 0
    for $b \leftarrow 1$ to B **do**
      if $E_{bu} < 0$ **then**
        neg ← neg + 1
      else
        pos ← pos + 1
      end if
    end for
    $p_u \leftarrow \frac{pos-neg}{B}$
  end for
  ranking ← sort_desc(p_u)
  return ranking
end procedure

Algorithm 3 SVF-Pseudosamples algorithm implementation

procedure GET_RANKING_PSEUDOSAMPLES(V,C,ϵ,d)
  ranking ← [ ]
  while !=stop_criteria **do**
    $\mathbf{w}^*, C^*, \epsilon^*, d^*$ ← SVF($\mathbf{V}, C, \epsilon, d$)
    for $u \leftarrow 1$ to m **do**
      $\mathbf{V}_u$ ← calculate_pseudosample()
      y_u ← prediction_SVF($\mathbf{w}^*, \mathbf{V}_u$)
      MAD$_u$ ← calculate_MAD($\mathbf{V}, y_u$)
    end for
    var_to_drop ← order_MAD_asc(mad)
    ranking ← add_var_to_ranking($ranking, var\_to\_drop$)
    data ← drop_var($data, var\_to\_drop$)
  end while
  ranking ← reverse($ranking$)
  return ranking
end procedure

3.4. SVF-Dual Objective Variation

In this case, we begin by setting up and solving the SVF model (4), and we use a cross-validation procedure to obtain the optimal hyperparameters (C^*, ϵ^*, d^*), which we fix. Then, we move onto the dual model to (4). The dual of model (4) is a maximization problem with the objective function $-J$, which is equivalent to the following model, see e.g. (Smola and Schölkopf, 2004, Model(10)):

$$\min_{\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\delta}} \quad J = \frac{1}{2}\sum_{r=1}^{s}\left(\left\|\mathbf{w}^{(r)}\right\|_2^2 + \epsilon^*\sum_{i=1}^{n}\delta_i^{(r)} - \sum_{i=1}^{n}(\alpha_i^{(r)} - \delta_i^{(r)})y_i^{(r)}\right) \tag{6}$$

$$\text{s.t.} \quad \alpha_i^{(r)} \geq 0, \qquad i = 1,\dots,n,\ r = 1,\dots,s \tag{6a}$$

$$\boldsymbol{\gamma}^{(r)} \geq 0, \qquad r = 1,\dots,s \tag{6b}$$

$$0 \leq \delta_i^{(r)} \leq C^*, \qquad i = 1,\dots,n,\ r = 1,\dots,s \tag{6c}$$

In program (6), we have $\mathbf{w}^{(r)} = \sum_{i=1}^{n}(\alpha_i^{(r)} - \delta_i^{(r)})\boldsymbol{\phi}^G_{SVF}(\mathbf{x}_i) + \boldsymbol{\gamma}^{(r)}$ for $r \in \{1,\dots,s\}$. The variables $\boldsymbol{\alpha}, \boldsymbol{\delta}$ are the dual variables corresponding to constraints (4a) and (4b) of the primal program, while $\boldsymbol{\gamma}$ corresponds to constraint (4c). The variable $\boldsymbol{\beta}$ corresponding to constraint (4d) yields the upper bound for $\boldsymbol{\delta}$, and satisfies $\beta_i^{(r)} + \delta_i^{(r)} = C^*$ for each component. Therefore, it can be substituted in the dual formulation and does not appear in the model. By substituting the expression for $\mathbf{w}$ in the objective J of model (6), the expression of J purely in terms of the variables of model (6) is:

$$\begin{aligned} J = &\frac{1}{2}\sum_{r=1}^{s}\left[\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\alpha_i^{(r)} - \delta_i^{(r)}\right)\left(\alpha_j^{(r)} - \delta_j^{(r)}\right)\left\langle\boldsymbol{\phi}^G_{SVF}(\mathbf{x}_i)\cdot\boldsymbol{\phi}^G_{SVF}(\mathbf{x}_j)\right\rangle\right. \\ &+2\left(\sum_{i=1}^{n}\left(\alpha_i^{(r)} - \delta_i^{(r)}\right)\left\langle\boldsymbol{\phi}^G_{SVF}(\mathbf{x}_i)\cdot\boldsymbol{\gamma}\right\rangle\right) + \sum_{k=1}^{(d^*)^m}{\gamma_k^{(r)}}^2 \\ &\left.+\epsilon^*\sum_{i=1}^{n}\delta_i^{(r)} - \sum_{i=1}^{n}\left(\alpha_i^{(r)} - \delta_i^{(r)}\right)y_i^{(r)}\right]. \end{aligned} \tag{7}$$

We solve model (6) with the optimal hyperparameters C^*, ϵ^*, d^* from the primal and we obtain optimal values $\boldsymbol{\alpha}^*, \boldsymbol{\delta}^*, \boldsymbol{\gamma}^*$, which we will then keep fixed.

When considering the effect of removing an input variable u while keeping $\boldsymbol{\alpha}^*, \boldsymbol{\delta}^*, \boldsymbol{\gamma}^*$ constant, the only terms that will change are those involving the transformation function $\boldsymbol{\phi}^G_{SVF}(\mathbf{x}_i)$. We denote the transformation without variable u by $\boldsymbol{\phi}^G_{SVF}(\mathbf{x}_i(-u))$, and the corresponding value of J by $J(u)$. The expression for $DJ(u)$ is given by:

$$\begin{aligned} &DJ(u) = J - J(u) = \\ &\frac{1}{2}\sum_{r=1}^{s}\left[\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\alpha_i^{*(r)} - \delta_i^{*(r)}\right)\left(\alpha_j^{*(r)} - \delta_j^{*(r)}\right)\left(\left\langle\boldsymbol{\phi}^G_{SVF}(\mathbf{x}_i)\cdot\boldsymbol{\phi}^G_{SVF}(\mathbf{x}_j)\right\rangle\right.\right. \\ &\left.\left. - \left\langle\boldsymbol{\phi}^G_{SVF}(\mathbf{x}_i(-u))\cdot\boldsymbol{\phi}^G_{SVF}(\mathbf{x}_j(-u))\right\rangle\right)\right) \\ &+ 2\left(\sum_{i=1}^{n}\left(\alpha_i^{*(r)} - \delta_i^{*(r)}\right)\left\langle\left(\boldsymbol{\phi}^G_{SVF}(\mathbf{x}_i) - \boldsymbol{\phi}^G_{SVF}(\mathbf{x}_i(-u))\right)\cdot\boldsymbol{\gamma}^*\right\rangle\right)\Bigg] \end{aligned} \tag{8}$$

We remark that only those terms involving $\mathbf{x}_i$ which have $\alpha_i^{*(r)} - \delta_i^{*(r)} \neq 0$ (i.e. the Support Vectors) will appear in the final calculation. This means that we only need to consider a subset of the data, which will usually be much smaller than the whole dataset, in order to calculate this value.

In order to calculate $DJ(u)$, we now describe how $\boldsymbol{\phi}^G_{SVF}(\mathbf{x}_i)$ changes when a variable is removed. Recall that the transformation function $\boldsymbol{\phi}^G_{SVF}$ takes values $\{0,1\}$ on an m-dimensional grid of size $(d^*)^m$. When removing an input, we project the data into an $(m-1)$-dimensional space, and we project the values in each cell of the grid accordingly, removing the contribution of variable u. Let $\mathbf{x}_i$ be in cell $C_{l_1\dots l_m}$, where each $l_j \in \{1,\dots,d^*\}$. The effect of removing input u on $\boldsymbol{\phi}^G_{SVF}(\mathbf{x}_i)$ is that every cell whose index l_u is greater than 1 takes the value of 0, thus removing the effect of the variable u on the grid. The rest of the cells keep their original value.

We calculate $DJ(u)$ using Eq. (8) as u runs over every input that we are ranking. The variable that attains the smallest value of $DJ(u)$ is deemed the least important, it is ranked last among the remaining ones, and is removed from the model. The method is then iterated until every variable has been ranked. Algorithm 4 shows the main steps of the method.

Algorithm 4 SVF-DOV algorithm implementation

procedure GET_RANKING_SVF_DOV(**V**,C,ϵ,d)
  ranking ← [ ]
  DJ ← [ ]
  while !=stop_criteria **do**
    $C^*, \epsilon^*, d^* \leftarrow \text{SVF}(\mathbf{V}, C, \epsilon, d)$
    $\alpha^*, \delta^*, \sigma^* \leftarrow \text{SVF\_Dual}(\mathbf{V}, C^*, \epsilon^*, d^*)$
    for $u \leftarrow 1$ to m **do**
      $DJ_u \leftarrow$ calculate_dj($\alpha^*, \delta^*, \sigma^*, \mathbf{V}, u$)
    end for
    var_to_drop ← min(DJ)
    V ← drop_var(var_to_drop)
    ranking ← insert_var_to_ranking($ranking, var\_to\_drop$)
  end while
  return ranking
end procedure

3.5. FDH methodologies

We finally consider whether any of the methods introduced could be used as ranking methods using the FDH estimator. We choose FDH for this task since SVF estimates stepwise production functions, as does FDH. The adaptations and difficulties using DEA could be similar, but are left as future work.

The method SVF-PFI, which measures the importance of variables using the effects of permuting the variables, can be adapted to be used with a variety of estimators. It relies on an estimation using a training set, and the effect of this permutation on each variable is measured by comparing the error between the prediction values and the observed values of DMUs in the test set. We have therefore adapted it to its use with FDH. The shuffling of input values of the DMUs in the training set is performed as before, and its effect on the estimator is evaluated via the test set.

The bootstrapping methodology can be adapted to the FDH or DEA context, see for example proposals by Simar and Wilson (2000a). However, this adaptation was found unsatisfactory by Nataraja and Johnson (2011), due to reasons of computational complexity, lack of clarity regarding the number of replications, and poor performance. This is the reason why we have not considered this adaptation in our context.

The SVF-Pseudosamples algorithm relies on the estimator being able to propose an output value given an input profile, which FDH does not do directly. In the single-output case, this can be overcome (since the maximum – scalar – producible output is always unique), but, in general, it cannot. Instead, FDH (and DEA) returns a set of feasible output vectors given an input profile, but this is tricky to use as part of the algorithm to rank inputs. The version of FDH-Pseudosamples which we consider in this paper attaches to each pseudosample the output value 1 (in the single output case). The measurement of the efficiency of this pseudosample using FDH then estimates the maximum producible output given an input profile, and is used as the prediction value of FDH. However, in the multi-output case, this is not possible. For example, DMUs with different mixes of outputs can potentially yield different levels of maximum producible outputs.

Finally, we consider the SVF-DOV algorithm. This method uses the dual formulation of an SVM and interprets its objective value as a measure of the error of the model. This task cannot be directly performed with FDH. In particular, FDH is a Mixed Integer Linear Program and its dual may not be as tight as in the case of linear programs. See, e.g., Briec et al. (2004), where a local duality result is proved for FDH and the difficulties arising from the non-convexity are discussed.

4. Computational experiments

In this section, we are going to measure the performance of the methods exposed in Section 3 when establishing a ranking of the inputs of the production process. In addition, we will evaluate the computational cost. To this end, different experimental scenarios have been developed (see Table 2). In particular, a Cobb–Douglas production function has been simulated with three inputs (x_1, x_2, x_3) and one output (y) using the following formula: $y = {x_1}^a {x_2}^b e^{-I}$ and considering the variable x_3 independently, i.e., the variable x_3 is not part of the production process. The inputs have been randomly generated following a uniform distribution with a range between (1,10). In the calculation of the inefficiencies (I), different distributions have been used with different configurations (see the column "Inefficiency" in Table 2). The correlation between the variables x_1 and x_2 $(\rho_{x_1x_2})$ and the contribution to output of each input, i.e., the value of the coefficients a and b in the Cobb–Douglas production function, are shown in the other columns of Table 2.

The first scenario is the base case. From it, the characteristics are modified to analyze the behavior of the methods with different data. In experiments 2, 3, 4 and 5, the sample size is modified, setting it at 15, 20, 30 and 80 DMUs, respectively. In the next two experiments, we study the ranking by modifying the value of the correlation. To establish the correlation, the equation used is given by: $x_2 = \rho_{x_1x_2} x_1 + \upsilon\sqrt{1 - {\rho_{x_1x_2}}^2}$, where υ is a random variable obtained from a distribution $\upsilon \sim \text{Uni}(1, 10)$, following the method of Wang and Schmidt (2002). In scenario 8, we study the performance of the algorithms when the contribution of the variables to the output is similar. Finally, to test the robustness of the methods, the probability distribution of inefficiencies has been modified, considering a higher mean inefficiency, as well as an exponential distribution.

We ran 50 simulations of each experiment for each of the six previously developed methods and calculated the percentage of times that each variable reaches a position in the ranking. Due to the nature of the SVF model, it was necessary to establish a grid of hyperparameters and make a selection from them. The grid consisted of the following values: $C = [0.001, 0.01, 0.1, 1, 100, 1000]$, $\epsilon = [0, 0.001, 0.01, 0.1, 0.2, 0.5, 1]$, $d = 0.1hn$ where $h = 1, \ldots, 10$ and n is the sample size. To obtain the best hyperparameters for the SVF estimator, the original data set was divided into a training sample with 80 percent of the observations and a test sample made up of the remaining 20 percent of observations. For the SVF-BR method, 100 resamples were used, while for the SVF-Pseudosamples and FDH-Pseudosamples methods, $q = 10$ was used to create the pseudosamples. Regarding the SVF-PFI and FDH-PFI methods, the original data was split, with 80% becoming the training set while the remaining 20% of DMUs were used to test the effect of the permutation.

4.1. Results

Table 3 shows the results (in percentage) obtained by each of the algorithms evaluated. The way to obtain the percentage has been by calculating the number of repetitions in which a variable occupied a certain position in the ranking of the 50 simulations. The values 1, 2, 3 indicate the ranking position in which each of the three variables (x_1, x_2, x_3) included in the problem ended up. The SVF-BR algorithm does not appear due to high computational cost.

First, we evaluated the results associated with the scenarios in which there is variation in the number of DMUs. It can be observed that both the SVF-Pseudosamples and the SVF-DOV algorithms improve their performance as the number of DMUs increases. However, the effect in the SVF-PFI method is less pronounced. In particular, in scenarios 2 and 3, that is, when the number of DMUs is small and we are in the situation of the curse of dimensionality, we already observe satisfactory performances by SVF-Pseudosamples and SVF-DOV, which

Table 2
List of experiments and their meaning.

Experiment	Correlation	Inefficiency	Contribution	Details
1				Base case, $n = 50$
2	Independently generated	$I \sim \lvert N(0, 0.4)\rvert$	$a = 0.6,\ b = 0.3$	$n = 15$
3				$n = 20$
4				$n = 30$
5				$n = 80$
6	$x_2 = \rho_{x_1x_2}x_1 + v\sqrt{1-\rho_{x_1x_2}{}^2}$	$I \sim \lvert N(0, 0.4)\rvert$	$a = 0.6,\ b = 0.3$	Lowly correlated inputs, $\rho_{x_1x_2} = 0.2$
7				Highly correlated inputs, $\rho_{x_1x_2} = 0.8$
8	Independently generated	$I \sim \lvert N(0, 0.4)\rvert$	$a, b = 0.45$	x_1 and x_2 equal importance
9		$I \sim \lvert N(0, 0.8)\rvert$		
10	Independently generated	$I \sim \exp(0.4)$	$a = 0.6,\ b = 0.3$	Different inefficiency
11		$I \sim \exp(0.8)$		

Table 3
Ranking results.

		Sample size														
		1			2			3			4			5		
		1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
SVF-Pseudosamples	x_1	88%	12%	0%	82%	16%	2%	68%	26%	6%	92%	4%	4%	96%	4%	0%
	x_2	10%	78%	12%	10%	64%	26%	22%	56%	22%	8%	80%	12%	4%	94%	2%
	x_3	2%	10%	88%	8%	20%	72%	10%	18%	72%	0%	16%	84%	0%	2%	98%
SVF-PFI	x_1	98%	2%	0%	74%	14%	12%	52%	42%	6%	76%	22%	2%	74%	26%	0%
	x_2	2%	82%	16%	20%	52%	28%	38%	26%	36%	20%	50%	30%	24%	66%	10%
	x_3	0%	16%	84%	6%	34%	60%	10%	32%	58%	4%	28%	68%	2%	8%	90%
SVF-DOV	x_1	84%	16%	0%	68%	22%	10%	74%	24%	2%	82%	18%	0%	86%	14%	0%
	x_2	12%	72%	16%	18%	66%	16%	16%	52%	32%	18%	64%	18%	10%	86%	4%
	x_3	4%	12%	84%	14%	12%	74%	10%	24%	66%	0%	18%	82%	4%	0%	96%
FDH-Pseudosamples	x_1	82%	14%	4%	40%	16%	44%	44%	18%	38%	64%	20%	16%	88%	10%	2%
	x_2	16%	48%	36%	40%	26%	34%	38%	36%	26%	26%	30%	44%	10%	62%	28%
	x_3	2%	38%	60%	20%	58%	22%	18%	46%	36%	10%	50%	40%	2%	28%	70%
FDH-PFI	x_1	46%	26%	28%	50%	16%	34%	40%	32%	28%	38%	24%	38%	80%	18%	2%
	x_2	46%	30%	24%	32%	40%	28%	36%	30%	34%	28%	30%	42%	10%	46%	44%
	x_3	8%	44%	48%	18%	44%	38%	24%	38%	38%	34%	46%	20%	10%	36%	54%

		Correlation						Same weight			Inefficiencies								
		6			7			8			9			10			11		
		1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
SVF-Pseudosamples	x_1	92%	8%	0%	94%	6%	0%	36%	64%	0%	90%	10%	0%	86%	14%	0%	84%	12%	4%
	x_2	6%	84%	10%	6%	72%	22%	64%	36%	0%	10%	76%	14%	12%	76%	12%	16%	70%	14%
	x_3	2%	8%	90%	0%	22%	78%	0%	0%	100%	0%	14%	86%	2%	10%	88%	0%	18%	82%
SVF-PFI	x_1	98%	2%	0%	90%	10%	0%	66%	34%	0%	90%	6%	4%	90%	8%	2%	88%	8%	4%
	x_2	2%	80%	18%	10%	68%	22%	34%	64%	2%	10%	78%	12%	10%	74%	16%	12%	58%	30%
	x_3	0%	18%	82%	0%	22%	78%	0%	2%	98%	0%	16%	84%	0%	18%	82%	0%	34%	66%
SVF-DOV	x_1	76%	24%	0%	72%	28%	0%	46%	54%	0%	76%	18%	6%	84%	14%	2%	80%	14%	6%
	x_2	16%	70%	14%	24%	70%	6%	50%	44%	6%	14%	62%	24%	14%	78%	8%	18%	72%	10%
	x_3	8%	6%	86%	4%	2%	94%	4%	2%	94%	10%	20%	70%	2%	8%	90%	2%	14%	84%
FDH-Pseudosamples	x_1	88%	10%	2%	88%	6%	6%	54%	32%	14%	70%	18%	12%	68%	26%	6%	58%	26%	16%
	x_2	10%	54%	36%	12%	50%	38%	44%	38%	18%	24%	42%	34%	32%	42%	26%	28%	40%	32%
	x_3	2%	36%	62%	0%	44%	56%	2%	30%	68%	6%	40%	54%	0%	32%	68%	14%	34%	52%
FDH-PFI	x_1	54%	32%	14%	60%	24%	16%	20%	38%	42%	38%	42%	20%	56%	28%	16%	38%	40%	22%
	x_2	26%	34%	40%	34%	42%	24%	48%	28%	24%	50%	28%	22%	22%	34%	44%	32%	40%	28%
	x_3	20%	34%	46%	6%	34%	60%	32%	34%	34%	12%	30%	58%	22%	38%	40%	30%	20%	50%

correctly rank each variable in at least 60% of replications with as few as 15 DMUs.

Second, we evaluate the scenarios in which the relevant inputs in the production process are correlated. In scenario 6, the variables x_1 and x_2 are poorly correlated with each other ($\rho_{x_1x_2} = 0.2$), while in scenario 7 the variables are more correlated ($\rho_{x_1x_2} = 0.8$). In these scenarios, the SVF-DOV method performs well, since it is capable of detecting with high precision that the variable x_3 is not very relevant. Regarding the other methods, although they are correct when choosing the variable x_1 as the most relevant one in a high percentage of the simulations, they are not capable of establishing a correct ranking when placing the variable x_3 as the second most relevant in 20% of the simulations in scenario 7.

In addition, we analyze in scenario 8 how the methods work when the relevant variables have the same importance ($a = b = 0.45$). In this regard, the SVF-DOV method is the one that works best in these cases, since it practically always detects that the variable x_3 is the least relevant and, in addition, it chooses the variables x_1 and x_2 as the most relevant in a percentage of approximately 50%. The other methods have a similar performance, but have a tendency to sort the relevant variables in a particular order.

Finally, we assess the robustness of the methods when changing the inefficiencies through different distributions and variances. In scenario 9, a half-normal distribution is used with high median inefficiency. The results show that the selection percentages of the variable x_3 as the least important variable are high in SVF-Pseudosamples and SVF-PFI, but lower in the SVF-DOV method. In scenarios 10 and 11, in which

we modify the half-normal distribution to an exponential one, it can be observed how the SVF-Pseudosamples and SVF-DOV methods perform well while the SVF-PFI performs less accurately.

Regarding the comparison with the FDH-based methodologies, overall, we observe that the SVF-based methodologies outperform them. The FDH-PFI is the worst in terms of correct ranking of the variables, with every scenario failing to rank a single variable in its correct place in over 60% of replications except for the variable x_1 in scenario 5, that is, the scenario with 80 DMUs. Meanwhile, the FDH-Pseudosamples method is slightly better, but still worse than the SVF-based proposals. This method does correctly rank x_1 as the most important in scenarios 5, 6 and 7 in over 80% of the times. These scenarios are with 80 DMUs, or when there is correlation between variables, but its performance is much worse in the remaining scenarios. It particularly struggles to distinguish between x_2 and the irrelevant variable x_3, which it confuses in around 30% of replications, except in scenario 8, where the exponent of x_2 is higher.

In fact, the only scenario where FDH-Pseudosamples ranks each variable correctly in at least 60% of replications is scenario 5, with $n = 80$, whereas the SVF-based methodologies can correctly rank all three variables in at least 70% of replications in almost every scenario considered, with exception of scenario 8, where x_1 and x_2 have the same importance, and they are confused more often by all methods.

Overall, we observe that the average percentage of correct ranking of each variable throughout every scenario follows the above discussion. FDH-PFI only achieves a 42% average correct ranking, whereas FDH-Pseudosamples achieves 55% in this metric. The methods based on SVF achieve much better results, with SVF-PFI reaching 74%, while SVF-DOV achieves 75%. The best method according to this metric is SVF-Pseudosamples, which ranks each individual variable correctly in 80% of replications.

In summary, when evaluating all the experiments, it can be observed that the SVF-Pseudosamples method performs well in scenarios in which the sample size and the distribution of inefficiencies are altered, while in scenarios where the variables are more highly correlated or when the variables have the same contribution to the output, this method does not work so well. The SVF-PFI method behaves similarly to the SVF-Pseudosamples, although it does not perform as well overall. It shows less improvement with increasing sample size, and is more affected by changes in the distribution of inefficiencies. Finally, the SVF-DOV method performs worse in scenarios in which the sample sizes and the distribution of inefficiencies are modified, but it presents good results when the variables are highly correlated or have the same weight, partly because of its ability to distinguish the variable x_3 as the least important even when the correlation between the relevant variables is high (scenario 7). Meanwhile, the FDH-based methodologies have a worse performance than the SVF-based methodologies throughout.

4.2. Computational cost

To evaluate the computational cost of each method, the execution time of each technique has been evaluated by applying different sample sizes (see scenarios 1,4,5 of Table 2). For the computation, a cluster of workstations was utilized. This cluster uses a Rocky Linux release 8.6 (Green Obsidian) operating system and has 80 cores, 755 GB of RAM and 2 Intel(R) Xeon(R) Silver 4316 CPUs @ 2.30 GHz. The jobs were launched in parallel, with 1 core being used for each simulation of the scenarios.

Table 4 shows the average execution time of each of the algorithms applied to 30, 50 and 80 DMUs. As shown in the table, the SVF-BR method could not be simulated because it has to perform 100 resamples for each scenario. For example, in sample sizes of 30 DMUs, although the algorithm presented good results, it took on average approximately 10 h for each of the 50 simulations. Moreover, the method that proved to be fastest among the SVF-based ones in most of the scenarios was the SVF-DOV method, followed by the SVF-Pseudosamples and SVF-PFI. The computational cost of the FDH-based methods is much smaller in comparison with the SVF-based methods, but their performance in the considered scenarios is much worse than the methods based on SVF.

Table 4
Average execution times.

	30	50	80
SVF-Pseudosamples	7.6 min	51.3 min	7.5 h
SVF-PFI	26.3 min	3 h	+24 h
SVF-DOV	6.4 min	53.5 min	7 h
SVF-BR	10 h	–	–

Based on the percentage of improvement in Table 2 and the execution time of each algorithm, in scenarios with low correlation, it is recommended to use the SVF-Pseudosamples model, since it is the second fastest method and obtains good results when establishing a ranking of variables. On the other hand, in the case in which the data may be highly correlated, it is recommended to use the SVF-DOV method, since it is a method that obtains good results in this type of scenario.

5. Conclusions and future work

In this paper, we have presented various methodologies to measure the relative importance of the inputs involved in a productive process by enriching the Support Vector Frontiers (SVF) methodology introduced in Valero-Carreras et al. (2021) with input ranking capability. We use this relative importance to order the inputs from least to most important, obtaining a full ranking of the inputs of a productive process.

In the context of estimation of efficiency, estimators such as Free Disposal Hull (FDH) often encounter the so-called curse of dimensionality whereby their estimations of technical efficiency suffer when the number of variables is large compared to the available data. The problem of which variables are the most relevant for an estimator and methodology has long been studied in the world of machine learning, where many methodologies have been proposed for measuring the importance of each variable and rank them accordingly. We thus adapt approaches from the Support Vector Machines literature to the SVF estimator in order to measure the importance of each input to the prediction of the outputs of a DMU. This yields a ranking of the inputs by their importance to the production process, which also could provide information about which variables should be dropped from the problem when too many variables are available in the given dataset.

We adapt the methodology called Permutation Feature Importance, which is based on the random shuffling of the values of input variables in order to ascertain the effect of such shuffling on the estimated outputs for a DMU. We denote this method by SVF-PFI. If a variable is very important, the effect of randomly shuffling this variable should be large on the estimated outputs, whereas shuffling the values of an irrelevant variable should not affect the predictions of the model. We also adapt a Bootstrap Resampling methodology, which creates bootstrap resamples of the data and uses them to evaluate the importance of each variable on the process. Another proposed algorithm is based on the study of pseudosamples, which are synthetic data using the structure of the dataset. In this method, new data is constructed using the mean values of the data in all variables except the one being evaluated, which takes quantiles of the observed values, and uses the Median Absolute Deviation (MAD) of the obtained predictions in order to evaluate the importance of each input variable in the model. We call this method SVF-Pseudosamples. Finally, we propose an approach based on Dual Objective function Variation of the Dual formulation of SVF, called SVF-DOV, which attempts to simplify the calculation by reusing parts of the solution instead of solving the whole problem every iteration. It evaluates the variation in the objective function of the dual formulation

when including or excluding the variable of interest, and uses this variation to rank the inputs. We also discuss the adaptation of the algorithms above to the FDH context, and consider the FDH-PFI and FDH-Pseudosamples methods.

In order to compare the proposed methods, we have performed computational experiments in various scenarios, considering different sample sizes, correlations among variables, the effect of having variables with the same theoretical importance, and both half-normal and exponential inefficiency distributions. We observe that SVF-Pseudosamples achieves the best results overall, although when there is high correlation between the relevant inputs as well as when exponential inefficiency distributions are considered, SVF-DOV outperforms it. On the other hand, SVF-DOV is also the quickest among the methodologies based on SVF. Meanwhile, SVF-PFI obtains results comparable to those of SVF-Pseudosamples, but takes about 3 times the computational time as SVF-Pseudosamples. Of note is that, even in scenarios with very few DMUs, where the scenarios suffer from the curse of dimensionality, the SVF-Pseudosamples and SVF-DOV methods achieve satisfactory results. Finally, SVF-BR was computationally intractable even for small sample sizes, so we omitted it from the more detailed experiment. Meanwhile, the methods based on FDH require very little computational time, but they achieve much worse performance than those based on SVF.

The findings from this paper can help decision makers to choose, among those available variables which could potentially be used in the measurement of efficiency, which to include or exclude from potential models according to their ranking in preliminary models. This could enable the selection of better models with fewer variables that capture information about the DGP while avoiding the traps associated with the curse of dimensionality, overfitting, and related issues.

Regarding interesting lines of future work, an important one could be the study and comparison of these algorithms in additional contexts, such as contexts with massive dimensions and/or number of observations. There are computational limitations in the simulated scenarios which lead us to leave this as a potential avenue for further research. Potential lines of investigation would include improvements to the models, as well as the use of new techniques such as parallelization to better handle the computations required.

CRediT authorship contribution statement

Daniel Valero-Carreras: Methodology, Software, Data curation, Visualization, Validation, Writing – original draft, Writing – review & editing. **Raul Moragues:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing. **Juan Aparicio:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Funding acquisition. **Nadia M. Guerrero:** Conceptualization, Methodology.

Data availability

Data will be made available on request.

Acknowledgments

We thank two anonymous reviewers for providing constructive comments and help in improving the contents and presentation of this paper. R. Moragues and J. Aparicio thank the grants PID2019-105952GB-I00 and PID2022-136383NB-I00 funded by Ministerio de Ciencia e Innovación/ Agencia Estatal de Investigación /10.13039/501100011033. This work was also supported by the Generalitat Valenciana under Grant ACIF/2020/155. This work was also supported by the Cátedra Santander en Eficiencia y Productividad, Miguel Hernandez University (UMH). Additionally, J. Aparicio thanks the grant PROMETEO/2021/063 funded by the Valencian Community (Spain).

References

Adler, N., Golany, B., 2002. Including principal component weights to improve discrimination in data envelopment analysis. J. Oper. Res. Soc. 53 (9), 985–991.
Adler, N., Yazhemsky, E., 2010. Improving discrimination in data envelopment analysis: PCA–DEA or variable reduction. European J. Oper. Res. 202 (1), 273–284.
Aigner, D., Lovell, C.A.K., Schmidt, P., 1977. Formulation and estimation of stochastic frontier production function models. J. Econometrics 6 (1), 21–37.
Alonso-Atienza, F., Rojo-Álvarez, J.L., Rosado-Muñoz, A., Vinagre, J.J., García-Alberola, A., Camps-Valls, G., 2012. Feature selection using support vector machines and bootstrap methods for ventricular fibrillation detection. Expert Syst. Appl. 39 (2), 1956–1967.
Araújo, C., Barros, C.P., Wanke, P., 2014. Efficiency determinants and capacity issues in Brazilian for-profit hospitals. Health Care Manage. Sci. 17 (2), 126–138.
Banker, R.D., 1993. Maximum likelihood, consistency and data envelopment analysis: a statistical foundation. Manage. Sci. 39 (10), 1265–1273.
Banker, R.D., Charnes, A., Cooper, W.W., 1984. Some models for estimating technical and scale inefficiencies in data envelopment analysis. Manage. Sci. 30 (9), 1078–1092.
Banker, R.D., Maindiratta, A., 1992. Maximum likelihood estimation of monotone and concave production frontiers. J. Prod. Anal. 3 (4), 401–415.
Becker, N., Toedt, G., Lichter, P., Benner, A., 2011. Elastic SCAD as a novel penalization method for SVM classification tasks in high-dimensional data. BMC Bioinformatics 12 (1), 1–13.
Becker, N., Werft, W., Toedt, G., Lichter, P., Benner, A., 2009. PenalizedSVM: a R-package for feature selection SVM classification. Bioinformatics 25 (13), 1711–1712.
Benítez-Peña, S., Bogetoft, P., Morales, D.R., 2020. Feature selection in data envelopment analysis: A mathematical optimization approach. Omega 96, 102068.
Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.
Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. Classification and regression trees. In: Wadsworth Statistics/Probability Series, Wadsworth Advanced Books and Software, Belmont, CA, p. x+358.
Briec, W., Kerstens, K., Eeckaut, P.V., 2004. Non-convex technologies and cost functions: definitions, duality and nonparametric tests of convexity. J. Econ. 81, 155–192.
Chambers, R.G., Chung, Y., Färe, R., 1998. Profit, directional distance functions, and Nerlovian efficiency. J. Optim. Theory Appl. 98 (2), 351–364.
Charles, V., Aparicio, J., Zhu, J., 2019. The curse of dimensionality of decision-making units: A simple approach to increase the discriminatory power of data envelopment analysis. European J. Oper. Res. 279 (3), 929–940.
Charnes, A., Cooper, W.W., Rhodes, E., 1978. Measuring the efficiency of decision making units. European J. Oper. Res. 2 (6), 429–444.
Chen, Z., Han, S., 2021. Comparison of dimension reduction methods for DEA under big data via Monte Carlo simulation. J. Manage. Sci. Eng. 6 (4), 363–376.
Chen, Y.-W., Lin, C.-J., 2006. Combining SVMs with various feature selection strategies. In: Feature Extraction. Springer, pp. 315–324.
Daouia, A., Noh, H., Park, B.U., 2016. Data envelope fitting with constrained polynomial splines. J. R. Stat. Soc. Ser. B Stat. Methodol. 78 (1), 3–30.
Deprins, D., Simar, L., Tulkens, H., 1984. Measuring Labor-Efficiency in Post Offices. LIDAM Reprints CORE 571, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE).
Efron, B., 1979. The 1977 RIETZ lecture: Bootstrap methods: another look at the jackknife. Ann. Statist. 7 (1), 1–26.
Esteve, M., Aparicio, J., Rabasa, A., Rodriguez-Sala, J.J., 2020. Efficiency analysis trees: A new methodology for estimating production frontiers through decision trees. Expert Syst. Appl. 162, 113783.
Esteve, M., Aparicio, J., Rodriguez-Sala, J.J., Zhu, J., 2023. Random forests and the measurement of super-efficiency in the context of Free Disposal Hull. European J. Oper. Res. 304 (2), 729–744.
Farrell, M.J., 1957. The measurement of productive efficiency. J. R. Stat. Soc. Ser. A (Gen.) 120 (3), 253–290.
Guerrero, N.M., Aparicio, J., Valero-Carreras, D., 2022. Combining data envelopment analysis and machine learning. Mathematics 10 (6), 909.
Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. Mach. Learn. 46 (1), 389–422.
James, G., Witten, D., Hastie, T., Tibshirani, R., et al., 2013. An Introduction to Statistical Learning, Vol. 112. Springer.
Kerstens, K., Sadeghi, J., Toloo, M., Van de Woestyne, I., 2022. Procedures for ranking technical and cost efficient units: With a focus on nonconvexity. European J. Oper. Res. 300 (1), 269–281.
Kerstens, K., Van De Woestyne, I., 2014. Solution methods for nonconvex free disposal hull models: A review and some critical comments. Asia-Pac. J. Oper. Res. 31 (01), 1450010.
Krooshof, P.W., Ustun, B., Postma, G.J., Buydens, L.M., 2010. Visualization and recovery of the (bio) chemical interesting variables in data analysis with support vector machine classification. Anal. Chem. 82 (16), 7000–7007.
Kuosmanen, T., Johnson, A.L., 2010. Data envelopment analysis as nonparametric least-squares regression. Oper. Res. 58 (1), 149–160.
Kuosmanen, T., Johnson, A., 2017. Modeling joint production of multiple outputs in StoNED: Directional distance function approach. European J. Oper. Res. 262 (2), 792–801.

Li, Y., Shi, X., Yang, M., Liang, L., 2017. Variable selection in data envelopment analysis via Akaike's information criteria. Ann. Oper. Res. 253 (1), 453–476.
Liao, Z., Dai, S., Kuosmanen, T., 2023. Convex support vector regression. European J. Oper. Res..
Louppe, G., Wehenkel, L., Sutera, A., Geurts, P., 2013. Understanding variable importances in forests of randomized trees. In: Burges, C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K. (Eds.), Advances in Neural Information Processing Systems, Vol. 26. Curran Associates, Inc..
Maldonado, S., Weber, R., 2009. A wrapper method for feature selection using support vector machines. Inform. Sci. 179 (13), 2208–2217.
Meeusen, W., van Den Broeck, J., 1977. Efficiency estimation from Cobb-Douglas production functions with composed error. Internat. Econom. Rev. 18 (2), 435–444.
Nataraja, N.R., Johnson, A.L., 2011. Guidelines for using variable selection techniques in data envelopment analysis. European J. Oper. Res. 215 (3), 662–669.
Olesen, O., Ruggiero, J., 2022. The hinging hyperplanes: An alternative nonparametric representation of a production function. European J. Oper. Res. 296 (1), 254–266.
Parmeter, C.F., Racine, J.S., 2013. Smooth constrained frontier analysis. In: Recent Advances and Future Directions in Causality, Prediction, and Specification Analysis. Springer, pp. 463–488.
Pastor, J.T., Ruiz, J.L., Sirvent, I., 2002. A statistical test for nested radial DEA models. Oper. Res. 50 (4), 728–735.
Peyrache, A., Rose, C., Sicilia, G., 2020. Variable selection in data envelopment analysis. European J. Oper. Res. 282 (2), 644–659.
Postma, G., Krooshof, P., Buydens, L., 2011. Opening the kernel of kernel partial least squares and support vector machines. Anal. Chim. Acta 705 (1–2), 123–134.
Ruggiero, J., 2005. Impact assessment of input omission on DEA. Int. J. Inf. Technol. Decis. Mak. 4 (03), 359–368.
Sanz, H., Valim, C., Vegas, E., Oller, J.M., Reverter, F., 2018. SVM-RFE: selection and visualization of the most relevant features through non-linear kernels. BMC Bioinform. 19 (1), 1–18.
Simar, L., Wilson, P.W., 1998. Sensitivity analysis of efficiency scores: How to bootstrap in nonparametric frontier models. Manage. Sci. 44 (1), 49–61.
Simar, L., Wilson, P.W., 2000a. A general methodology for bootstrapping in non-parametric frontier models. J. Appl. Stat. 27 (6), 779–802.
Simar, L., Wilson, P.W., 2000b. Statistical inference in nonparametric frontier models: The state of the art. J. Prod. Anal. 13 (1), 49–78.
Simar, L., Wilson, P.W., 2001. Testing restrictions in nonparametric efficiency models. Comm. Statist. Simulation Comput. 30 (1), 159–184.
Sirvent, I., Ruiz, J.L., Borrás, F., Pastor, J.T., 2005. A Monte Carlo evaluation of several tests for the selection of variables in DEA models. Int. J. Inf. Technol. Decis. Mak. 4 (03), 325–343.
Smith, P., 1997. Model misspecification in data envelopment analysis. Ann. Oper. Res. 73, 233–252.
Smola, A.J., Schölkopf, B., 2004. A tutorial on support vector regression. Stat. Comput. 14, 199–222.
Tsionas, M.G., 2022. Efficiency estimation using probabilistic regression trees with an application to Chilean manufacturing industries. Int. J. Prod. Econ. 108492.
Ueda, T., Hoshiai, Y., 1997. Application of principal component analysis for parsimonious summarization of DEA inputs and/or outputs. J. Oper. Res. Soc. Japan 40 (4), 466–478.
Valero-Carreras, D., Aparicio, J., Guerrero, N.M., 2021. Support vector frontiers: A new approach for estimating production functions through support vector machines. Omega 104, 102490.
Valero-Carreras, D., Aparicio, J., Guerrero, N.M., 2022. Multi-output support vector frontiers. Comput. Oper. Res. 143, 105765.
Vapnik, V., 1991. Principles of risk minimization for learning theory. Adv. Neural Inf. Process. Syst. 4.
Vapnik, V., 1998. Statistical Learning Theory. In: A Wiley-Interscience publication, Wiley.
Wang, H.J., Schmidt, P., 2002. One-step and two-step estimation of the effects of exogenous variables on technical efficiency levels. J. Prod. Anal. 18.
Zelenyuk, V., 2020. Aggregation of inputs and outputs prior to Data Envelopment Analysis under big data. European J. Oper. Res. 282 (1), 172–187.