Computers & Operations Research 143 (2022) 105765

Contents lists available at ScienceDirect

Computers and Operations Research

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

Multi-output Support Vector Frontiers

Daniel Valero-Carreras, Juan Aparicio *, Nadia M. Guerrero

Center of Operations Research (CIO), Miguel Hernandez University of Elche (UMH), 03202 Elche, Alicante, Spain

ARTICLE INFO

Keywords:
Support Vector Machines
Efficiency
Computational experience
Data Envelopment Analysis

ABSTRACT

In this paper, we show that both Free Disposal Hull (FDH) and Data Envelopment Analysis (DEA), which are well-known modern techniques for efficiency measurement, can be seen as particular cases of a more general model based upon Support Vector Regression (SVR) within machine learning. Our approach is based on the adaptation of SVR in a multi-response framework for dealing with standard microeconomic assumptions, such as free disposability and convexity of the underlying technology. This adaptation allows us to introduce a more robust notion of technical efficiency, linked to the concept of ε-insensitivity in standard SVR. Due to computational reasons, we also introduce a simplified version of the initial approach, whose validity is checked through simulation. By resorting to a computational experience, we also show that the new approach, called multi-output Support Vector Frontiers, outperforms FDH and DEA with respect to mean squared error and bias, avoiding the overfitting problem associated with the assumption of the principle of minimal extrapolation in the case of FDH and DEA. We finally show how to implement some usual efficiency measures under the new approach and illustrate their performance through an empirical example.

1. Introduction

The measurement of technical efficiency through the previous estimation of a production function or a technology, which underlies the generation of the observations (also called Decision Making Units – DMUs –), has been a topic of interest in economics, management science, manufacturing engineering and operations research in the last decades. For the single-output case, the Data Generating Process (DGP) that is usually assumed and governs the generation of the observations (firms, schools, farms, etc.) may be briefly described as follows. An unknown production function $f(\boldsymbol{x}) : \mathbb{R}_+^m \rightarrow \mathbb{R}_+$ represents the maximal output that is producible from an input vector $\boldsymbol{x}$. Nevertheless, if we assume that technical inefficiency can take place in the production process, then obtaining the maximal producible output from a certain input bundle $\boldsymbol{x}$ could not be possible for a DMU. So, the actual observed output for this DMU would coincide with $y = f(\boldsymbol{x}) - u$, where $u \geqslant 0$ is defined as a random variable representing technical inefficiency. Notice then that $y \leqslant f(\boldsymbol{x})$ for any observation under the deterministic scenario, where statistical noise (the effects of weather, strikes, luck, etc.) is not considered. The assumption of 'deterministicness' implies that the production function always envelops all the data from above, something that contrasts with the standard regression problem, where the target function associates a response variable with some predictors in average terms. In the multi-output multi-input case, other axioms are usually assumed, as convexity and free disposability in inputs and outputs. In this context, a technology or production possibility set is the set of all the input–output bundles that are producible. Convexity means that given two feasible input–output bundles, any convex combination of them is also producible. In the case of dealing with a production function in the single-output production context, the convexity assumption of the technology is translated to concavity of the production function. Regarding free disposability, it states that if a certain input–output bundle is producible, then any input–output bundle that presents a greater value for inputs and a lower value for outputs is also feasible (i.e., doing it worse is always possible). Graphically, in the case of producing only a single-output, free disposability is related to the estimation of a non-decreasing production function. Under the DGP described above, for each observation we observe y rather than $f(\boldsymbol{x})$, but we are really interested in determining a certain measure of the deviation between y and $f(\boldsymbol{x})$, which represents technical inefficiency for the evaluated DMU. This is the challenge faced by researchers: we observe y but we need to have an accurate estimation of $f(\boldsymbol{x})$ to provide a measure of technical inefficiency for DMUs.

Different parametric and non-parametric methods have been introduced to cope with this estimation problem in the literature. On the one hand, good representatives of parametric methodologies for estimating

* Corresponding author.
E-mail address: j.aparicio@umh.es (J. Aparicio).

https://doi.org/10.1016/j.cor.2022.105765
Received 24 June 2021; Received in revised form 1 February 2022; Accepted 22 February 2022
Available online 8 March 2022
0305-0548/© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

production functions are, among others, Aigner et al. (1977) and Meeusen and van den Broeck (1977), who independently proposed the Stochastic Frontier Analysis (SFA) methodology, which takes measurement errors and other sources of statistical noise for estimating a parametrically defined production function into account. However, as is usual in the field of regression models, the treatment of the multi-response case is not trivial. Additionally, at an initial stage, these techniques need to specify the mathematical expression of the production function, stated in terms of a list of parameters to be estimated. On the other hand, we have non-parametric techniques for estimating the technical efficiency. Nowadays, two famous non-parametric approaches for estimating production functions are Free Disposal Hull (FDH) by Deprins et al. (1984) and Data Envelopment Analysis (DEA) by Charnes et al. (1978) and Banker et al. (1984). In the case of FDH, the estimator of the production function is a step function, whereas in the case of DEA, the estimator is a piece-wise linear function. These two techniques are related: the convex hull of the production possibility set estimated by Free Disposal Hull coincides with the technology provided by Data Envelopment Analysis (Daraio and Simar, 2005). From an axiomatic point of view, FDH is a desirable technique since it relies only on a few technological assumptions. Free disposability in inputs and outputs is one of them. Moreover, FDH is assumed to be deterministic (no statistical noise is assumed). In other words, the technology estimated by FDH always contains all the observations. However, many possible estimators may fulfill free disposability and deterministicness. Hence, a third axiom must be assumed to finally get the FDH estimate. We are referring to the so-called minimal extrapolation principle, which is linked to the well-known Occam's razor problem-solving principle. Invoking the minimal extrapolation principle, the most conservative estimator would be that related to a function that envelops the data from above and is as close as possible to the observations. Regarding DEA, this technique assumes convexity of the technology as well as deterministicness, free disposability and minimal extrapolation. Both FDH and DEA are data-driven approaches and, therefore, they do not assume restrictive premises on the DGP. The same happens with respect to machine learning approaches, which nowadays represents a mainstream in the statistics literature. However, neither FDH nor DEA can be considered proper machine learning techniques. The main objective of FDH and DEA seems to be the description of the extreme behavior of a group of DMUs, under the fulfillment of certain production assumptions. And, indeed, both techniques suffer from overfitting (the model describes the data sample instead of telling something about the underlying Data Generating Process). The minimal extrapolation principle seems to be the main culprit of this shortcoming. In practice, one of the effects of overfitting is that many assessed DMUs are evaluated as being technically efficient, especially in the case of FDH. In contrast, usual machine learning techniques (as for example, Classification and Regression Trees or Support Vector Machines) incorporate tools to avoid the overfitting problem. Specifically, they consider the concept of generalization error, which allows more accurate estimations of the target function to be given.

Some authors have modified standard FDH and DEA techniques so that they work as inferential methods rather than as mere descriptive tools. For example, Banker and Maindiratta (1992) and Banker (1993) related Data Envelopment Analysis to maximum likelihood. Simar and Wilson (1998) and Simar and Wilson (2000a, 2000b) adapted bootstrapping to FDH and DEA. Kuosmanen and Johnson (2010) and Kuosmanen and Johnson (2017) introduced the Corrected Concave Nonparametric Least Squares. Unfortunately, despite the importance of machine learning techniques in the current literature, there have been few attempts to adapt Free Disposal Hull and Data Envelopment Analysis to the field of machine learning. In this regard, Esteve et al. (2020) is one of the few exceptions, adapting Classification and Regression Trees (CART) to the efficiency analysis scenario. More recently, Valero-Carreras et al. (2021) have introduced a new technique called Support Vector Frontiers (SVF), based on Support Vector Machines (Vapnik, 1995, 1998), which allows the estimation of production functions fulfilling the usual microeconomic postulates. See also Hong et al. (1999), Yang and Dimitrov (2017), Badiezadeh et al. (2018), Zhu et al. (2018), Zhu (2019) and Pendharkar (2021).

However, in the contribution by Valero-Carreras et al. (2021), the Support Vector Frontiers technique was exclusively defined for working in the single-output production context, which represents a huge limitation from a practical point of view. In practice, production processes are complex enough to consider yielding multiple type of outputs from the consumption of multiple kinds of resources. One trivial possibility for extending the single-output model for dealing with the multi-output framework would be to apply the single-output methodology, treating each output in an independent way, i.e., fitting as many models as the number of outputs. However, this strategy would neglect the existing relationships among the different outputs, which could lead to worse estimations of the multi-dimensional target surface (this issue will be illustrated later in our computational experience). For this reason, in this paper, we adapt the SVF technique by Valero-Carreras et al. (2021) to the multi-output multi-input production context by resorting to a model that considers all outputs at the same time. In particular, we prove that the technology estimated by the multi-output SVF technique satisfies deterministicness, free disposability in inputs and outputs and, additionally, if desired, convexity. Moreover, we show that FDH and DEA can be understood as particular cases of the more general multi-output Support Vector Frontiers technique. Furthermore, a more robust notion of technical efficiency is introduced, based upon the margin concept in Support Vector Regression. We also show how to implement the new approach through linear and mixed-integer linear programming. However, due to the huge number of constraints that the optimization program can have, depending on the size of the problem, we introduce a simplified version of the original. We check the performance of this simplification and the SVF technique, against standard FDH and DEA, through a computational experience based on simulated data. Our results show that the new approach outperforms both FDH and DEA regarding Mean Squared Error and Bias. We also show how some well-known technical efficiency measures (different distances to the technological frontier) may be implemented under the new approach. Finally, the new method is applied to a real dataset.

The structure of the paper follows. We briefly revise the basis of Free Disposal Hull, Data Envelopment Analysis and Support Vector Frontiers for single-output production processes. We extend Support Vector Frontiers to the multi-output production framework in Section 3, fulfilling standard microeconomic assumptions. Performance of the new approach, against FDH and DEA, is analyzed through a computational experience in Section 4. Section 5 is devoted to showing how some well-known measures of technical efficiency can be implemented through the new technique. In Section 6, we show an empirical illustration. Finally, Section 7 concludes.

2. Background: FDH, DEA and SVF

In this section, we briefly review the grounds related to FDH and DEA. Additionally, we also review a new methodology called Support Vector Frontier (SVF), which was recently introduced for dealing with a production context based on multiple inputs and a single output.

2.1. Free Disposal Hull (FDH)

Let $\Omega = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$ be a set of n Decision Making Units (DMUs), where each DMU consumes $\mathbf{x}_i = \left(x_i^{(1)}, ..., x_i^{(m)}\right) \in \mathbb{R}_+^m$ amounts of inputs for the production of $\mathbf{y}_i = \left(y_i^{(1)}, ..., y_i^{(s)}\right) \in \mathbb{R}_+^s$ amounts of outputs. Throughout this paper, we will use bold for denoting vectors, and non-bold for scalars. In general terms, the so-called production possibility set or technology is defined as follows:.

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

A certain relevant part of the boundary of the technology, defined as $\partial(T) := \{(\mathbf{x}, \mathbf{y}) \in T : \widehat{\mathbf{x}} < \mathbf{x}, \widehat{\mathbf{y}} > \mathbf{y} \Rightarrow (\widehat{\mathbf{x}}, \widehat{\mathbf{y}}) \notin T\}$, is called the efficient frontier of T. Technical inefficiency is defined as the distance from $(\mathbf{x}, \mathbf{y}) \in T$ to $\partial(T)$. There are a lot of possible distances to be selected to measure technical inefficiency. One of the most famous technical efficiency measures is the output-oriented radial measure by Charnes et al. (1978) and Banker et al. (1984). This measure determines the efficiency score for an evaluated point $(\mathbf{x}_k, \mathbf{y}_k)$ by maintaining inputs constant and equiproportionally increasing all its outputs:.

$$\phi(\mathbf{x}_k, \mathbf{y}_k) = max\{(\mathbf{x}_k, \phi_k \mathbf{y}_k) \in T\} \tag{2}$$

Given a data set $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$, it is necessary to estimate T, in a first stage, to determine the value of $\phi(\mathbf{x}_k, \mathbf{y}_k)$. To do that, one of the possible techniques to be applied is the Free Disposal Hull (FDH). This approach was first formulated by Deprins et al. (1984):.

$$\widehat{T}_{FDH} = \{(\mathbf{x}, \mathbf{y}) \in R_+^{m+s} : \mathbf{y} \leqslant \mathbf{y}_i, \mathbf{x} \geqslant \mathbf{x}_i, i = 1, ..., n\} \tag{3}$$

$\widehat{T}_{FDH}$ is derived from the data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$ assuming: (1) $(\mathbf{x}_i, \mathbf{y}_i)$ must belong to the technology for all $i = 1, ..., n$ (deterministicness); (2) free disposability and (3) the minimal extrapolation principle. Then, the efficiency score $\phi(\mathbf{x}_k, \mathbf{y}_k)$ is determined by substituting the general T by $\widehat{T}_{FDH}$ in. By this replacement, $\phi(\mathbf{x}_k, \mathbf{y}_k)$ can be calculated through mixed-integer linear optimization programming, as follows:

$$\phi_{FDH}(\mathbf{x}_i, \mathbf{y}_i) = \begin{array}{lllr} max & \phi & & (4.0) \\ s.t. & & & \\ & \sum_{k=1}^{n} \lambda_k x_k^{(j)} \leqslant x_i^{(j)}, & j = 1, ..., m & (4.1) \\ & \sum_{k=1}^{n} \lambda_k y_k^{(r)} \geqslant \phi y_i^{(r)}, & r = 1, ..., s & (4.2) \\ & \sum_{k=1}^{n} \lambda_k = 1, & & (4.3) \\ & \lambda_k \in \{0, 1\}, & k = 1, ..., n & (4.4) \end{array} \tag{4}$$

2.2. Data Envelopment Analysis (DEA)

Data Envelopment Analysis (DEA) is another non-parametric method used to measure the efficiency of a set of DMUs. This model was introduced by Charnes et al. (1978) and Banker et al. (1984) (see also Pastor et al., 2012, Aparicio et al., 2015, Aparicio et al., 2016, Aparicio et al., 2018). As FDH, it allows us to estimate a technology through a data-driven approach. In contrast with FDH, DEA needs stronger statements (convexity). The DEA estimator of the technology is as follows (Banker et al., 1984):.

$$\widehat{T}_{DEA} = \left\{ \begin{array}{c} (\mathbf{x}, \mathbf{y}) \in R_+^{m+s} : y^{(r)} \leqslant \sum_{k=1}^{n} \lambda_k y_k^{(r)}, \forall r = 1, ..., s, \\ x^{(j)} \geqslant \sum_{k=1}^{n} \lambda_k x_k^{(j)}, \forall j = 1, ..., m, \sum_{k=1}^{n} \lambda_k = 1, \lambda_k \geqslant 0, \forall k = 1, ..., n \end{array} \right\} \tag{5}$$

As Daraio and Simar (2005) pointed out, the convexification of the FDH estimator yields the DEA estimator.

The DEA estimator of the efficiency score $\phi(\mathbf{x}_k, \mathbf{y}_k)$ is as follows:.

$$\phi_{DEA}(\mathbf{x}_i, \mathbf{y}_i) = \begin{array}{lllr} max & \phi & & (6.0) \\ s.t. & & & \\ & \sum_{k=1}^{n} \lambda_k x_k^{(j)} \leqslant x_i^{(j)}, & j = 1, ..., m & (6.1) \\ & \sum_{k=1}^{n} \lambda_k y_k^{(r)} \geqslant \phi y_i^{(r)}, & r = 1, ..., s & (6.2) \\ & \sum_{k=1}^{n} \lambda_k = 1, & & (6.3) \\ & \lambda_k \geqslant 0, & k = 1, ..., n & (6.4) \end{array} \tag{6}$$

Mathematically speaking, the only difference between models and is that the decision variables λ_k, $k = 1, ..., n$, are non-negative in and binary in.

From an axiomatical point of view, Date Envelopment Analysis (DEA) and Free Disposal Hull (FDH) frontiers are defined from the same set of postulates except for one: convexity. DEA assumes free disposability, convexity, the condition that each observation must belong to the technology and minimal extrapolation (Banker et al. 1984). In contrast, FDH relies only on free disposability, the condition that each observation must belong to the technology and minimal extrapolation (Deprins et al., 1984); ignoring convexity. Geometrically speaking, the postulates that each technique satisfy are responsible for the shape of the corresponding frontier. While DEA yields a piece-wise linear frontier, FDH produces a step function. As Daraio and Simar (2005) pointed out, FDH may be considered the 'skeleton' of DEA since the convex hull of the frontier estimated by FDH coincides with the DEA frontier. From a practical viewpoint, and although convexity is an assumption widely used in economics, it is not always valid. The production possibility set might admit increasing returns to scale (i.e., outputs increase at a faster rate than inputs, which cannot be graphically modelled by convexity), or there might be lumpy goods (i.e., fractional values of inputs or outputs do not exist) (see Aragon et al., 2005). From a statistical perspective, Simar and Wilson (2000a,b, p. 56) state that, if the technology is non-convex, then there is no choice and one must use FDH, since DEA is inconsistent. If the technology is convex, then DEA is preferred over FDH since it offers a faster rate of convergence.

As for the limitations of DEA and FDH, some of them were previously highlighted in, for example, Dyson et al. (2001); as the possible heterogeneity of the units under evaluation or the different effect caused on the efficiency scores by the input/output set selected. From a statistical perspective, Simar and Wilson (1998) pointed out some weaknesses of nonparametric frontier estimators as DEA and FDH. Regarding specifically the content of our paper, we would like to highlight that Data Envelopment Analysis and Free Disposal Hull have already been criticized for their deterministic and non-statistical nature, being labelled as pure descriptive tools of the data sample with little inferential power (their inferential power is exclusively based on the property of consistency and the increase of the sample size as opposed to the fundamentals of the method) (see the recent paper by Esteve et al., 2020). Indeed, both DEA and FDH suffer from an overfitting problem because of the application of the minimal extrapolation principle, which places the estimator of the production frontier as close to the observations as possible. This limitation is the reason why DEA and FDH, considered as data-driven approaches, are not able to provide a suitable estimation of the actual production frontier from which the observed units were drawn. In other words, DEA and FDH were not defined to correctly describe the unknown Data Generating Process (DGP), which is behind the generation of the observations, but to depict the behaviour of exactly the n observed DMUs.

Regarding the DGP in frontier analysis, for the single-output case, it can be assumed (Aragon et al., 2005, Daraio and Simar, 2007) that an unknown production function $f(\boldsymbol{x}) : \mathbb{R}^m_+ \to \mathbb{R}_+$ represents the maximal output that is producible from an input vector $\boldsymbol{x}$. If we assume that technical inefficiency can take place in the production process, then the actual observed output for a DMU would coincide with $y = f(\boldsymbol{x}) - u$, where $u \geqslant 0$ is defined as a random variable representing technical inefficiency. Under the DGP described above, for each DMU we observe y instead of $f(\boldsymbol{x})$, but we are really interested in determining a certain measure of the deviation between y and $f(\boldsymbol{x})$, which represents the actual technical inefficiency for the corresponding assessed unit. This is the challenge faced by researchers: we observe y but we need to have an accurate estimation of $f(\boldsymbol{x})$ to provide a real measure of technical inefficiency. In the multi-output framework, Daraio and Simar (2007, Chapter 3) describe the Data Generating Process (DGP) which lies behind every productive process, introducing the statistical foundation of nonparametric frontier models. It is assumed that we observe a sample of an identically and independently distributed random input–output vector with an unknown joint distribution with a certain support. In the production framework, the technology, i.e., everything that is feasible to be observed, coincides with the support of the joint distribution. Our paper works within this context, avoiding overfitting by the adaptation of a machine learning technique and trying to identify the DGP that lies behind the production process. This identification allows to determine technical efficiency with respect to the actual production frontier. This contrasts with DEA and FDH, which are descriptive tools of the behavior in terms of efficiency of the set of n observed units.

2.3. Support Vector Frontier (SVF)

Support Vector Frontier (see Valero-Carreras et al., 2021) (SVF) is a machine learning technique, based on Support Vector Machines (SVM), used for the estimation of production functions, i.e., it was defined for dealing wihe single-output multiple-input production context. Let us introduce in this section some important notation used throughout the paper.

Following Vapnik (1998, p. 464), given a set of knots $T_j = \left\{t^{(j)}_{l_j} : l_j = 1, ..., k_j\right\}$ for each input dimension j, a grid G is defined as the disjoint subsets $C_{1...1}, ..., C_{k_1...k_m}$ such that $C_{l_1...l_m} := \left\{\boldsymbol{x} \in \mathbb{R}^m_+ / t^{(j)}_{l_j} \leqslant \boldsymbol{x}^{(j)} < t^{(j)}_{l_j+1}, j = 1, ...m\right\}$, $l_1 \in \{1, ..., k_1\}, ..., l_m \in \{1, ..., k_m\}$ with $t^{(j)}_{k_j+1} := \infty$, $\forall j = 1, ..., m$. Each subset $C_{l_1...l_m}$ is a cell of the grid G. Additionally, for each cell $C_{l_1...l_m}$, $\boldsymbol{a}_{l_1...l_m} = \left(t^{(1)}_{l_1}, ..., t^{(m)}_{l_m}\right)$ and $\boldsymbol{b}_{l_1...l_m} = \left(t^{(1)}_{l_1+1}, ..., t^{(m)}_{l_m+1}\right)$ represent the lower extreme knot-point and the upper extreme knot-point, respectively. In this way, $C_{l_1...l_m} = \left\{\boldsymbol{x} \in \mathbb{R}^m_+ : a^{(j)}_{l_1...l_m} \leqslant \boldsymbol{x}^{(j)} < b^{(j)}_{l_1...l_m}, j = 1, ...m\right\}$. Additionally, an activation function for each cell $C_{l_1...l_m}$ is defined as $L_{l_1...l_m}(\boldsymbol{x}) = \prod_{j=1}^m B\left(x^{(j)} - t^{(j)}_{l_j}\right)$, where.

$$B\left(x^{(j)} - t^{(j)}_{l_j}\right) = \begin{cases} 1, & if \quad x^{(j)} - t^{(j)}_{l_j} \geqslant 0 \\ 0, & if \quad x^{(j)} - t^{(j)}_{l_j} < 0 \end{cases}, \forall l_j = 1, \ldots, k_j, \forall j = 1, \ldots, m$$

When a cell $C_{l_1...l_m}$ is activated, then all the cells $C_{s_1...s_m}$ with $s_j \leqslant l_j$, are also activated, then $L_{s_1...s_m}(\boldsymbol{x}) = 1$. Given an input bundle $\boldsymbol{x}$ and a grid G, $\phi^G_{SVF}(\boldsymbol{x})$ is a vector of ones and zeros defined as:.

$$\boldsymbol{x} \to \phi^G_{SVF}(\boldsymbol{x}) = (L_{11...11}(\boldsymbol{x}), L_{11...12}(\boldsymbol{x}), \ldots, L_{11...1k_m}(\boldsymbol{x}), L_{11...21}(\boldsymbol{x}), L_{11...22}(\boldsymbol{x}), \ldots, L_{11...2k_m}(\boldsymbol{x}), \ldots, L_{k_1k_2...k_m}(\boldsymbol{x})) \quad (7)$$

Finally, we refer any reader interested in the details of the single-output SVF approach to review the content of Valero-Carreras et al. (2021).

3. A multi-output version of Support Vector Frontiers

In this part of the paper, we introduce an extension of Support Vector Frontiers (Valero-Carreras et al., 2021), defined only for single-output scenarios, by adapting it to the more usual multi-output production context. This adaptation will be mainly based on Vazquez and Walter (2003), who introduced a generalization of Support Vector Regression for dealing with multiple response variables. Additionally, and due to computational issues, we also present a simplified version of the original multi-output SVF approach.

Leaving all other weaknesses aside (pointed out at the end of Section 2.2), our paper focuses its attention on solving the problem of overfitting associated with standard DEA and FDH. The problem of overfitting, mainly caused by the principle of minimal extrapolation, is the reason why DEA and FDH determine the degree of 'relative' technical efficiency for each unit under evaluation, i.e., the degree of efficiency measured in comparative terms with respect to the performance of exactly the n observed units in the data sample. In contrast, in our paper, we aim to identify 'absolute' technical efficiency, that is, efficiency measured with respect to the underlying production frontier associated with the (unknown) data generating process from which the data was drawn. To avoid overfitting, Support Vector Regression (SVR), which is the base of Support Vector Frontiers (SVF), is rooted in the principle of structural risk minimization (see Vapnik, 1995, 1998). SVR aims to minimize the bound on the generalization error (i.e., the error made by the estimator on data outside and inside the data sample) rather than directly minimizing the empirical error, which is what DEA and FDH do (see Kuosmanen and Johnson, 2010). The SVR model depends on two parameters to be estimated: C, which balances between correctly predicting outside or inside the data sample, and ε, a margin added to the estimator for endowing the model with robustness. These parameters are usually tuned by applying a cross-validation procedure, based on folds. In V-fold cross-validation, where V usually equals 5 or 10, the data Ω is randomly divided into $\Omega_1, \ldots, \Omega_V$ disjoint subsamples with the same sample size or as close as possible. Let the v-th fold (subsample) be $\Omega^{(v)} = \Omega - \Omega_v$. Then, the process to tune the parameters C and ε is based on considering a grid of possible values for them and solving the SVR model for each one using the data in $\Omega^{(v)}$; additionally, the error made in predicting the data in Ω_v

is computed. This step is repeated with each of the folds considered in the analysis. Finally, the best values for C and ε are determined as those that minimize the predicting error. All these relatively complex steps have been adapted in our approach with the goal of estimating technical efficiency and avoiding overfitting in the context of Data Envelopment Analysis and Free Disposal Hull.

3.1. The multi-output SVF extension

The multi-output SVF is an extension of SVF to adapt this method to the multi-output production scenario. This adaptation is based on Vazquez and Walter (2003), who extended SVR by considering the so-called Cokriging method, which is a multi-output version of Kriging that exploits the correlations due to the proximity in the space of factors and outputs. In model, we show the changes made in model (9) in Valero-Carreras et al. (2021) to extend the single-output framework to the multi-output production context.

$$
\begin{array}{llll}
\underset{w_r,\xi_{ri}}{Min} & \sum_{r=1}^{s}\left\|w^{(r)}\right\|_1 + C\sum_{r=1}^{s}\sum_{i=1}^{n}\xi_i^{(r)} & & (8.0)\\
s.t. & w^{(r)}\phi_{SVF}^{G}(x_i) - y_i^{(r)} \geqslant 0, & i=1,\ldots,n,\ r=1,\ldots,s & (8.1)\\
 & w^{(r)}\phi_{SVF}^{G}(x_i) - y_i^{(r)} \leqslant \varepsilon + \xi_i^{(r)}, & i=1,\ldots,n,\ r=1,\ldots,s & (8.2)\\
 & W_{l_1\ldots s_j\ldots l_m}^{(r)} \leqslant W_{l_1\ldots l_m}^{(r)}, & \forall l_1,\ldots,l_m / l_j\in\{1,\ldots,k_j\},\forall j\in\{1,\ldots,m\}\ \ \forall s_j = l_j-1,\forall j=1,\ldots,m,\forall r=1,\ldots,s & (8.3)\\
 & \xi_i^{(r)} \geqslant 0, & i=1,\ldots,n,\ r=1,\ldots,s & (8.4)
\end{array}
\qquad (8)
$$

In model, $y_i^{(r)}$, $r=1,\ldots,s$, are the components of the output vector $y_i = \left(y_i^{(1)},\ldots,y_i^{(s)}\right)$. Moreover, $\phi_{SVF}^{G}(x_i)$ denotes the application of the transformation function ϕ_{SVF}^{G}, defined in, on the input vector $x_i = \left(x_i^{(1)},\ldots,x_i^{(m)}\right)$; enabling the treatment of multi-output multi-input production processes. Additionally, there is a vector $w^{(r)}$ associated with each output dimension r, $r=1,\ldots,s$. Each vector $w^{(r)}$ has a component for each cell of the grid G, $w_{l_1\ldots l_m}^{(r)}$, defined in the input space. Also, new deviation variables to the margin $\xi_i^{(r)}$ were defined, for each observation i in each output dimension r. Regarding the constraints, they are the same as for the SVF single-output model but replicated for each output dimension, where $W_{l_1\ldots l_m}^{(r)} := \sum_{\substack{s_1=1,\ldots,l_1\\ \vdots \\ s_m=1,\ldots,l_m}} w_{s_1\ldots s_m}^{(r)}, \forall r=1,\ldots,s$. Model tries to minimize the objective function by finding a balance between its two components: the regularization term $\sum_{r=1}^{s}\left\|w^{(r)}\right\|_1$ and the empirical error $\sum_{r=1}^{s}\sum_{i=1}^{n}\xi_i^{(r)}$, through the parameter C. In the constraints, ε, known as the model margin, is a parameter related to robustness of the results. The best model will be obtained by cross-validation as in the univariate case, tuning the parameters $C\in\mathbb{R}_+$ and $\varepsilon\in\mathbb{R}_+$. Once the best model has been obtained, the prediction is defined as $\hat{y}_{SVF}^{(r)}(x) = w^{*(r)}\phi_{SVF}^{G}(x)$, $r=1,\ldots,s$. In this way, we define the estimator of the technology for the multi-output multi-input production context as follows:.

$$\hat{T}_{SVF} := \left\{(x,y)\in R_+^{m+s} : y \leqslant \hat{y}_{SVF}(x)\right\} \qquad (9)$$

The estimator satisfies the axioms of deterministicness and free disposability, as we show next. Before doing so, it is necessary to state a technical result.

Lemma 1. *Let $x' \geqslant x$, then $\hat{y}_{SVF}(x') \geqslant \hat{y}_{SVF}(x)$.*

Proof. By the way we defined the estimation of each component of the output vector, we have that if $x' \geqslant x$, then $\hat{y}_{SVF}(x') \geqslant \hat{y}_{SVF}(x)$, due to $\hat{y}_{SVF}(x')$ is a non-decreasing function (see Theorem 1, Valero-Carreras et al., 2021). ■.

Proposition 1. *$(x_i,y_i)\in\hat{T}_{SVF}$, for all $i=1,\ldots,n$.*

Proof. By constraint (8.1), we have $\hat{y}_{SVF}^{(r)}(x_i) = w^{*(r)}\phi_{SVF}^{G}(x_i) \geqslant y_i^{(r)}$. Finally, by, we have that $(x_i,y_i)\in\hat{T}_{SVF}$ for all $i=1,\ldots,n$. ■.

Proposition 2. *The set $\hat{T}_{SVF}$ meets free disposability.*

Proof. Let $(x,y)\in\hat{T}_{SVF}$ and let $(x',y')\in R_+^{m+s}$ such that $x' \geqslant x$ and $y' \leqslant y$. First, $y \leqslant \hat{y}_{SVF}(x)$ holds since $(x,y)\in\hat{T}_{SVF}$. Second, by Lemma 1, if $x' \geqslant x$, then $\hat{y}_{SVF}(x') \geqslant \hat{y}_{SVF}(x)$. Hence, $y' \leqslant y \leqslant \hat{y}_{SVF}(x) \leqslant \hat{y}_{SVF}(x')$, which implies that $(x',y')\in\hat{T}_{SVF}$ by. ■.

Additionally, let us show a natural result: the single-output SVF model (Valero-Carreras et al., 2021) is a particular case of the multi-output SVF model.

Proposition 3. *If $s=1$, then model is equivalent to model (9) in Valero-Carreras et al. (2021).*

Proof. It is evident from the formulations of models in this paper and (9) in Valero-Carreras et al. (2021). ■.

As happens with the single-output SVF model, each observation in the sample belongs to a cell $C_{l_1\ldots l_m}$ in a grid defined in an input space. As we mentioned above, each cell is associated with a lower extreme knot-point $(a_{l_1\ldots l_m})$. Then, we can prove that the technology estimated from the multi-output SVF model, i.e., $\hat{T}_{SVF}$, may be rewritten as an FDH-type technology defined from the 'virtual' (in the sense of not being necessarily observed) input–output points $\left\{\left(a_{l_1\ldots l_m}, \hat{y}_{SVF}(a_{l_1\ldots l_m})\right)\right\}_{\substack{l_1=1,\ldots,k_1\\ \vdots\\ l_m=1,\ldots,k_m}}$. This result is formally established in the following proposition.

Proposition 4. *$\hat{T}_{SVF} = \{(x,y)\in R_+^{m+s} : \exists l_1,\ldots,l_m$, with $l_1\in\{1,\ldots,k_1\},\ldots,l_m\in\{1,\ldots,k_m\}$ such that*

$$y \leqslant \hat{y}_{SVF}(a_{l_1 \ldots l_m}), x \geqslant a_{l_1 \ldots l_m} \}. \quad (10)$$

Proof. We use $\widehat{T}_{SVF}^{FDH}$ for denoting the RHS of expression (10). Let $(x', y') \in \widehat{T}_{SVF}$. Then, by definition, $y' \leqslant \hat{y}_{SVF}(x')$. If we consider $x' \in C_{l'_1 \ldots l'_m}$, then $\hat{y}_{SVF}(x') = \hat{y}_{SVF}(x)$ for all $x \in C_{l'_1 \ldots l'_m}$. Additionally, we have that $a_{l'_1 \ldots l'_m} \leqslant x'$. In this way, we have that $\hat{y}_{SVF}(a_{l'_1 \ldots l'_m}) = \hat{y}_{SVF}(x')$ and $x' \geqslant a_{l'_1 \ldots l'_m}$. Consequently, $\exists l'_1, \ldots, l'_m$, with $l'_1 \in \{1, \ldots, k_1\}, \ldots, l'_m \in \{1, \ldots, k_m\}$ such that $y' \leqslant \hat{y}_{SVF}(a_{l'_1 \ldots l'_m})$ and $x' \geqslant a_{l'_1 \ldots l'_m}$. Consequently, $(x', y') \in \widehat{T}_{SVF}^{FDH}$. Let now $(x', y') \in \widehat{T}_{SVF}^{FDH}$. Then $\exists l_1, \ldots, l_m$, with $l_1 \in \{1, \ldots, k_1\}, \ldots, l_m \in \{1, \ldots, k_m\}$ such that $y' \leqslant \hat{y}_{SVF}(a_{l_1 \ldots l_m})$ and $x' \geqslant a_{l_1 \ldots l_m}$. By Lemma 1, $\hat{y}_{SVF}(x') \geqslant \hat{y}_{SVF}(a_{l_1 \ldots l_m})$. Then we have $y' \leqslant \hat{y}_{SVF}(a_{l_1 \ldots l_m}) \leqslant \hat{y}_{SVF}(x')$, which implies that $(x', y') \in \widehat{T}_{SVF}$ by definition of $\widehat{T}_{SVF}$. ■.

Notice that the right hand side in coincides with the definition of an FDH-type technology, as, when the set of observations $\{(x_i, y_i)\}_{i=1}^n$ is substituted by the set of virtual points $\{(a_{l_1 \ldots l_m}, \hat{y}_{SVF}(a_{l_1 \ldots l_m}))\}_{l_1 = 1, \ldots, k_1; \ \vdots; \ l_m = 1, \ldots, k_m}$. The last proposition has an important application. First, Free Disposal Hull (FDH) can be seen as a particular case of the multi-output SVF model, when the lower extreme knot-points and their corresponding output estimations linked to SVF coincide with the data. Nevertheless, in general, the production possibility set estimated by FDH is always a subset of the technology derived from the SVF technique. It is due to the fact that both technologies satisfy deterministicness and free disposability, but only the technology related to FDH meets minimal extrapolation. Second, the result may also be utilized to show how to determine any measure of technical efficiency using $\widehat{T}_{SVF}$. For example, for the output-oriented radial measure, the efficiency score $\phi(x_i, y_i)$ may be estimated by plugging $\widehat{T}_{SVF}$ in place of T. Using the parallelism established between the FDH technology and $\widehat{T}_{SVF}$, the output-oriented radial score can be determined through an optimization problem as in where, in the constraints, the original data is substituted by virtual input–output bundles:.

$$
\begin{aligned}
\phi_{SVF}(x_i, y_i) = \quad & \max \quad \phi && (11.0)\\
& s.t. \\
& \sum_{l_1 = 1, \ldots, k_1; \ \vdots; \ l_m = 1, \ldots, k_m} \lambda_{l_1, \ldots, l_m} a^{(j)}_{l_1 \ldots l_m} \leqslant x_i^{(j)}, \quad j = 1, \ldots, m && (11.1)\\
& \sum_{l_1 = 1, \ldots, k_1; \ \vdots; \ l_m = 1, \ldots, k_m} \lambda_{l_1, \ldots, l_m} \hat{y}^{(r)}_{SVF}(a_{l_1 \ldots l_m}) \geqslant \phi y_i^{(r)} \quad r = 1, \ldots, s && (11.2)\\
& \sum_{l_1 = 1, \ldots, k_1; \ \vdots; \ l_m = 1, \ldots, k_m} \lambda_{l_1, \ldots, l_m} = 1, && (11.3)\\
& \lambda_{l_1, \ldots, l_m} \in \{0, 1\}, \quad \forall l_j = 1, \ldots, k_j, \forall j = 1, \ldots, m && (11.4)
\end{aligned}
\quad (11)
$$

Seeking robustness in our results, we can also adapt the definition of ε-insensitive technically efficiency from the single-output SVF context (see Valero-Carreras et al., 2021) to the multi-output framework by incorporating the margin ε to the constraints of model. Regarding the application of the notion of ε-insensitive technical efficiency, this is a more robust definition of efficiency than the traditional one. Geometrically speaking, the method builds a band of points in the input–output space that is used to determine whether a unit can be identified as technically efficient, i.e., located within the band, or not. In that framework, technical inefficiency can be defined as the 'distance' from

Table 1
Number of restrictions in model.

Restriction	Number of restrictions
8.1	sn
8.2	sn
8.3	$\approx md^m$

Table 2
Calculation of the MSE statistic and bias for each method.

Method	MSE	Bias
SVF	$\sum_{e=1}^{50}\sum_{i=1}^{n}(\phi(x_i,y_i) - \phi_{SVF}(x_i,y_i))^2/50n$	$\sum_{e=1}^{50}\sum_{i=1}^{n}\lvert\phi(x_i,y_i) - \phi_{SVF}(x_i,y_i)\rvert/50n$
FDH	$\sum_{e=1}^{50}\sum_{i=1}^{n}(\phi(x_i,y_i) - \phi_{FDH}(x_i,y_i))^2/50n$	$\sum_{e=1}^{50}\sum_{i=1}^{n}\lvert\phi(x_i,y_i) - \phi_{FDH}(x_i,y_i)\rvert/50n$
DEA	$\sum_{e=1}^{50}\sum_{i=1}^{n}(\phi(x_i,y_i) - \phi_{DEA}(x_i,y_i))^2/50n$	$\sum_{e=1}^{50}\sum_{i=1}^{n}\lvert\phi(x_i,y_i) - \phi_{DEA}(x_i,y_i)\rvert/50n$
CSVF	$\sum_{e=1}^{50}\sum_{i=1}^{n}(\phi(x_i,y_i) - \phi_{CSVF}(x_i,y_i))^2/50n$	$\sum_{e=1}^{50}\sum_{i=1}^{n}\lvert\phi(x_i,y_i) - \phi_{CSVF}(x_i,y_i)\rvert/50n$

the assessed input–output bundle to this band.

In particular, when the aim is to assess the efficiency level of an observation belonging to the data sample, then it is enough to incorporate the information associated with the lower bound of the estimation of the outputs. This is due to the fact that, for any observation $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, we have that $\widehat{y}^{(r)}_{SVF}(\boldsymbol{a}_{l_1\ldots l_m}) \geqslant \widehat{y}^{(r)}_{SVF}(\boldsymbol{x}_i) \geqslant y^{(r)}_i$, with $C_{l_1,\ldots,l_m}$ such that $\boldsymbol{x}_i \in C_{l_1,\ldots,l_m}$. And, consequently, $\widehat{y}^{(r)}_{SVF}(\boldsymbol{a}_{l_1\ldots l_m}) + \varepsilon \geqslant y^{(r)}_i$, with $\varepsilon \geqslant 0$, trivially holds. In this scenario, the model to be solved would be the following:.

In the case of evaluating a new observation, if $\phi_{SVF-}(\boldsymbol{x},\boldsymbol{y}) \leqslant 1$ and $\phi_{SVF+}(\boldsymbol{x},\boldsymbol{y}) \geqslant 1$, then the model signals this unit as ε-insensitive technically efficient since it is located between the upper and lower bounds. This type of situation can happen when the unit to be evaluated is not in the learning sample. For example, technological improvements over time can yield observations in period time t that may be located outside the technology corresponding to a previous period t'.

Additionally, to provide an estimation of the technology satisfying

$$
\begin{aligned}
\phi_{SVF-}(\boldsymbol{x}_i,\boldsymbol{y}_i) = \quad & \max \quad \phi^- && (12.0)\\
& s.t. \\
& \sum_{\substack{l_1=1,\ldots,k_1\\ \vdots \\ l_m=1,\ldots,k_m}} \lambda_{l_1,\ldots,l_m} a^{(j)}_{l_1\ldots l_m} \leqslant x^{(j)}_i, \quad j=1,\ldots,m && (12.1)\\
& \sum_{\substack{l_1=1,\ldots,k_1\\ \vdots \\ l_m=1,\ldots,k_m}} \lambda_{l_1,\ldots,l_m} \left(\widehat{y}^{(r)}_{SVF}(\boldsymbol{a}_{l_1\ldots l_m}) - \varepsilon^*\right) \geqslant \phi^- y^{(r)}_i \quad r=1,\ldots,s && (12.2)\\
& \sum_{\substack{l_1=1,\ldots,k_1\\ \vdots \\ l_m=1,\ldots,k_m}} \lambda_{l_1,\ldots,l_m} = 1, && (12.3)\\
& \lambda_{l_1,\ldots,l_m} \in \{0,1\}, \quad \forall l_j=1,\ldots,k_j, \forall j=1,\ldots,m && (12.4)
\end{aligned}
\tag{12}
$$

For the observations, and given the optimal value of model, denoted as $\phi_{SVF-}(\boldsymbol{x}_i,\boldsymbol{y}_i)$, we have two possibilities. If $\phi_{SVF-}(\boldsymbol{x}_i,\boldsymbol{y}_i) \rangle 1$, then the observation is located below the lower bound and it is ε-insensitive technically inefficient. Otherwise, i.e., if $\phi_{SVF-}(\boldsymbol{x}_i,\boldsymbol{y}_i) \leqslant 1$, the observation is located between the margins and, consequently, the model identifies it as ε-insensitive technically inefficient.

In contrast, in the case of assessing a new observation $(\boldsymbol{x},\boldsymbol{y})$, we have to consider the upper bound as well as the lower bound. In this way, model must be used for evaluating $(\boldsymbol{x},\boldsymbol{y})$, getting the value $\phi_{SVF-}(\boldsymbol{x},\boldsymbol{y})$. Additionally, model must be also determined with the aim of taking into account the information linked to the upper bound.

the convexity assumption, we can directly convexify $\widehat{T}_{SVF}$, giving rise to the extension of the Convexified Support Vector Frontiers (CSVF) technique for dealing with the multi-output production context. In this way, the production possibility set estimated by CSVF could be defined as $\widehat{T}_{CSVF} := conv(\widehat{T}_{SVF})$. As we show next, invoking Proposition 4, it is possible to rewrite $\widehat{T}_{CSVF}$ in terms of the virtual set of points $\left\{\left(\boldsymbol{a}_{l_1\ldots l_m}, \widehat{\boldsymbol{y}}_{SVF}(\boldsymbol{a}_{l_1\ldots l_m})\right)\right\}_{\substack{l_1=1,\ldots,k_1\\ \vdots \\ l_m=1,\ldots,k_m}}$, in the same way that the DEA technology can be obtained from the convexification of the FDH technology.

$$
\begin{aligned}
\phi_{SVF+}(\boldsymbol{x}_i,\boldsymbol{y}_i) = \quad & \max \quad \phi^+ && (13.0)\\
& s.t. \\
& \sum_{\substack{l_1=1,\ldots,k_1\\ \vdots \\ l_m=1,\ldots,k_m}} \lambda_{l_1,\ldots,l_m} a^{(j)}_{l_1\ldots l_m} \leqslant x^{(j)}_i, \quad j=1,\ldots,m && (13.1)\\
& \sum_{\substack{l_1=1,\ldots,k_1\\ \vdots \\ l_m=1,\ldots,k_m}} \lambda_{l_1,\ldots,l_m} \left(\widehat{y}^{(r)}_{SVF}(\boldsymbol{a}_{l_1\ldots l_m}) + \varepsilon^*\right) \geqslant \phi^+ y^{(r)}_i \quad r=1,\ldots,s && (13.2)\\
& \sum_{\substack{l_1=1,\ldots,k_1\\ \vdots \\ l_m=1,\ldots,k_m}} \lambda_{l_1,\ldots,l_m} = 1, && (13.3)\\
& \lambda_{l_1,\ldots,l_m} \in \{0,1\}, \quad \forall l_j=1,\ldots,k_j, \forall j=1,\ldots,m && (13.4)
\end{aligned}
\tag{13}
$$

$$\widehat{T}_{CSVF}=\left\{\begin{array}{c}(\boldsymbol{x},\boldsymbol{y})\in R_{+}^{m+s}:y^{(r)}\leqslant\sum_{\substack{l_1=1,\ldots,k_1\\ \vdots\\ l_m=1,\ldots,k_m}}\lambda_{l_1,\ldots,l_m}\widehat{y}_{SVF}^{(r)}(\boldsymbol{a}_{l_1\ldots l_m}),\forall r=1,\ldots,s,\\ x^{(j)}\geqslant\sum_{\substack{l_1=1,\ldots,k_1\\ \vdots\\ l_m=1,\ldots,k_m}}\lambda_{l_1,\ldots,l_m}a_{l_1\ldots l_m}^{(j)},\forall j=1,\ldots,m,\sum_{\substack{l_1=1,\ldots,k_1\\ \vdots\\ l_m=1,\ldots,k_m}}\lambda_{l_1,\ldots,l_m}=1,\lambda_{l_1,\ldots,l_m}\geqslant0,\forall l_1,\ldots,l_m\end{array}\right\} \tag{14}$$

Regarding the fulfillment of the usual microeconomic assumptions, convexity is trivially satisfied, while deterministicness and free disposability must be checked.

Proposition 5. $(x_i,y_i)\in\widehat{T}_{CSVF}$

Proof. Given that $\widehat{T}_{SVF}\subset\widehat{T}_{CSVF}$, *by* Proposition 1, *we have that* $(x_i,y_i)\in\widehat{T}_{CSVF}$. ■.

Proposition 6. $\widehat{T}_{CSVF}$ *meets free disposability.*

Proof. Let $(\boldsymbol{x},\boldsymbol{y})\in\widehat{T}_{CSVF}$. By, there are $\lambda_{l_1,\ldots,l_m}\geqslant0$, $l_1=1,\ldots,k_1$, …, $l_m=1,\ldots,k_m$, with $\sum_{\substack{l_1=1,\ldots,k_1\\ \vdots\\ l_m=1,\ldots,k_m}}\lambda_{l_1,\ldots,l_m}=1$, such that $y^{(r)}\leqslant\sum_{\substack{l_1=1,\ldots,k_1\\ \vdots\\ l_m=1,\ldots,k_m}}\lambda_{l_1,\ldots,l_m}\widehat{y}_{SVF}^{(r)}(\boldsymbol{a}_{l_1\ldots l_m})$, $\forall r=1,\ldots,s$, $x^{(j)}\geqslant\sum_{\substack{l_1=1,\ldots,k_1\\ \vdots\\ l_m=1,\ldots,k_m}}\lambda_{l_1,\ldots,l_m}a_{l_1\ldots l_m}^{(j)}$, $\forall j=1,\ldots,m$. Let now $(\boldsymbol{x}',\boldsymbol{y}')\in\mathbb{R}_{+}^{m+s}$, with $\boldsymbol{x}'\geqslant\boldsymbol{x}$ and $\boldsymbol{y}'\leqslant\boldsymbol{y}$. Then, $y'^{(r)}\leqslant y^{(r)}\leqslant\sum_{\substack{l_1=1,\ldots,k_1\\ \vdots\\ l_m=1,\ldots,k_m}}\lambda_{l_1,\ldots,l_m}\widehat{y}_{SVF}^{(r)}(\boldsymbol{a}_{l_1\ldots l_m})$, $\forall r=1,\ldots,s$, $x'^{(j)}\geqslant x^{(j)}\geqslant\sum_{\substack{l_1=1,\ldots,k_1\\ \vdots\\ l_m=1,\ldots,k_m}}\lambda_{l_1,\ldots,l_m}a_{l_1\ldots l_m}^{(j)}$, $\forall j=1,\ldots,m$. This implies that $(\boldsymbol{x}',\boldsymbol{y}')\in\widehat{T}_{CSVF}$, as we wanted to prove. ■.

As in the case of the SVF, given an input–output bundle, CSVF can be applied to determine the output-oriented radial measure of technical efficiency as well as its robust versions, based upon the ε-insensitive efficiency notion, simply by adding the constraint $\sum_{\substack{l_1=1,\ldots,k_1\\ \vdots\\ l_m=1,\ldots,k_m}}\lambda_{l_1,\ldots,l_m}=1$ to programs (11), (12) and (13).

Finally, it is worth mentioning that Data Envelopment Analysis may be seen as a particular case of the CSVF technique when the set of virtual points $\left\{(\boldsymbol{a}_{l_1\ldots l_m},\widehat{\boldsymbol{y}}_{SVF}(\boldsymbol{a}_{l_1\ldots l_m}))\right\}_{\substack{l_1=1,\ldots,k_1\\ \vdots\\ l_m=1,\ldots,k_m}}$ coincides with the set of observations $\left\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\right\}_{i=1}^{n}$. Nevertheless, in general, the production possibility set estimated by DEA is always a subset of the technology derived from CSVF. The reason behind this result is that both technologies satisfy deterministicness and free disposability, but only the technology related to DEA fulfills the principle of minimal extrapolation, which, at the same time, relates DEA to overfitting.

3.2. Simplified versions of SVF and CSVF techniques

The multi-output SVF model is a hard optimization program with respect to computational resources and time employed. This is due to the huge number of parameters and constraints that it presents when the dimensionality and sample size of the problem increase. Regarding the number of constraints associated with model, the following table shows how many restrictions are linked to each type of constraint in the approach.

Table 1 shows that the quantity of constraints in the SVF model exponentially augments as the number of inputs increases, especially due to constraint 8.3, where d represents the hyperparameter associated with the number of partitions into which each input dimension is split through the knots. In this regard, we propose an alternative model, called the Simplified Support Vector Frontiers (SSVF) approach, which attempts to ease the computational difficulties associated with the original multi-output SVF model. To do that, we substitute the complex constraint 8.3 by the simpler restriction 15.3, as follows:.

$$\begin{array}{llll}\underset{w,\xi_i}{Min} & \sum_{r=1}^{s}\left\|\boldsymbol{w}^{(r)}\right\|_1+C\sum_{r=1}^{s}\sum_{i=1}^{n}\xi_i^{(r)} & & (15.0)\\ s.t. & \boldsymbol{w}^{(r)}\phi_{SVF}^{G}(\boldsymbol{x}_i)-y_i^{(r)}\geqslant0, & i=1,\ldots,n,r=1,\ldots,s & (15.1)\\ & \boldsymbol{w}^{(r)}\phi_{SVF}^{G}(\boldsymbol{x}_i)-y_i^{(r)}\leqslant\varepsilon+\xi_i^{(r)}, & i=1,\ldots,n,r=1,\ldots,s & (15.2)\\ & \boldsymbol{w}^{(r)}\geqslant0, & r=1,\ldots,s & (15.3)\\ & \xi_i^{(r)}\geqslant0, & i=1,\ldots,n,r=1,\ldots,s & (15.4)\end{array} \tag{15}$$

We next prove that $\boldsymbol{w}^{(r)}\geqslant0$, for all $r=1,\ldots,s$, is a sufficient condition for guaranteeing the satisfaction of constraint 8.3.

Proposition 7. *If* $\boldsymbol{w}^{(r)}\geqslant0\quad\forall r=1,\ldots,s$, *then* $W_{l_1l_2\ldots s_j\ldots l_m}^{(r)}\leqslant W_{l_1l_2\ldots l_m}^{(r)},\forall s_j=l_j-1,\forall j=1,\ldots,m,\forall r=1,\ldots,s.$

Proof. $\boldsymbol{w}^{(r)}\geqslant0$, $\forall r=1,\ldots,s$, implies that for each $r=1,\ldots,s$, $w_{h_1\ldots h_m}^{(r)}\geqslant0$, $\forall h_j=1,\ldots,k_j$ and $\forall j=1,\ldots,m$. In this way,.

$$\sum_{\substack{h_1=1,\ldots,l_1\\ \ldots\\ h_j=1,\ldots,(l_j-1)\\ \ldots\\ h_m=1,\ldots,l_m}}w_{h_1\ldots h_m}^{(r)}\leqslant\sum_{\substack{h_1=1,\ldots,l_1\\ \ldots\\ h_j=1,\ldots,l_j\\ \ldots\\ h_m=1,\ldots,l_m}}w_{h_1\ldots h_m}^{(r)},$$

$\forall l_j=1,\ldots,k_j,\quad\forall j=1,\ldots,m$;.

and, therefore, by definition of $W_{l_1\ldots l_j-1\ldots l_m}^{(r)}$, we have.

$W_{l_1\ldots l_j-1\ldots l_m}^{(r)}\leqslant W_{l_1\ldots l_m}^{(r)},\ \forall l_j=1,\ldots,k_j,\quad\forall j=1,\ldots,m$. ■.

Once the best model has been determined by cross-validation, the prediction from the simplified approach is defined as $\widehat{y}_{SSVF}^{(r)}(\boldsymbol{x})=\boldsymbol{w}^{*(r)}\phi_{SVF}^{G}(\boldsymbol{x})$, $r=1,\ldots,s$. Moreover, by analogy with the SVF and CSVF

Table 3
Best hyperparameters for the scenario without random noise.

Method	Size	Frontier	Hyperparameter		
			C	ε	d
			MEAN (STD)	MEAN (STD)	MEAN (STD)
Original	**20**	**0%**	2.350(1.995)	0.100(0.101)	13.200(4.982)
Simplified	**20**	**0%**	0.880(0.808)	0.068(0.096)	12.520(4.691)
Original	**20**	**5%**	2.564(2.085)	0.094(0.100)	14.160(4.913)
Simplified	**20**	**5%**	0.868(0.805)	0.088(0.100)	11.840(4.880)
Original	**20**	**10%**	2.256(2.058)	0.098(0.100)	14.320(4.718)
Simplified	**20**	**10%**	0.860(0.792)	0.092(0.101)	11.800(5.018)
Original	**30**	**0%**	2.984(1.958)	0.085(0.088)	22.020(7.212)
Simplified	**30**	**0%**	0.922(0.781)	0.076(0.098)	20.100(6.393)
Original	**30**	**5%**	2.874(1.981)	0.071(0.083)	22.020(6.900)
Simplified	**30**	**5%**	0.862(0.753)	0.068(0.096)	20.340(6.918)
Original	**30**	**10%**	2.728(1.961)	0.062(0.077)	21.360(6.739)
Simplified	**30**	**10%**	0.852(0.754)	0.084(0.100)	19.800(7.146)
Original	**40**	**0%**	3.530(1.797)	0.077(0.081)	29.520(7.955)
Simplified	**40**	**0%**	0.716(0.686)	0.076(0.098)	31.040(8.224)
Original	**40**	**5%**	4.040(1.662)	0.074(0.082)	29.120(8.477)
Simplified	**40**	**5%**	0.986(0.789)	0.116(0.100)	28.960(8.497)
Original	**40**	**10%**	3.870(1.619)	0.072(0.085)	28.640(8.216)
Simplified	**40**	**10%**	0.972(0.764)	0.106(0.100)	28.400(9.116)
Original	**50**	**0%**	3.772(1.775)	0.045(0.069)	34.000(10.498)
Simplified	**50**	**0%**	0.670(0.670)	0.104(0.101)	32.600(10.264)
Original	**50**	**5%**	3.662(1.693)	0.081(0.084)	34.500(10.797)
Simplified	**50**	**5%**	0.580(0.578)	0.104(0.101)	34.700(9.763)
Original	**50**	**10%**	3.882(1.607)	0.065(0.081)	33.900(11.619)
Simplified	**50**	**10%**	0.544(0.579)	0.104(0.101)	32.800(11.525)
Original	**60**	**0%**	3.690(1.650)	0.064(0.082)	47.520(12.352)
Simplified	**60**	**0%**	0.942(0.883)	0.088(0.100)	45.960(13.477)
Original	**60**	**5%**	3.744(1.667)	0.051(0.069)	44.040(13.906)
Simplified	**60**	**5%**	0.746(0.636)	0.080(0.099)	44.280(12.176)
Original	**60**	**10%**	3.720(1.688)	0.047(0.069)	44.400(13.444)
Simplified	**60**	**10%**	0.736(0.611)	0.064(0.094)	47.760(10.565)
Original	**70**	**0%**	4.050(1.485)	0.043(0.074)	47.320(18.031)
Simplified	**70**	**0%**	0.720(0.718)	0.066(0.094)	49.560(14.549)
Original	**70**	**5%**	4.080(1.426)	0.071(0.085)	49.280(17.830)
Simplified	**70**	**5%**	0.710(0.872)	0.098(0.100)	51.240(15.839)
Original	**70**	**10%**	3.910(1.557)	0.072(0.087)	49.420(16.817)
Simplified	**70**	**10%**	0.726(0.846)	0.076(0.098)	52.500(14.849)
Original	**80**	**0%**	3.780(1.595)	0.051(0.066)	56.800(16.263)
Simplified	**80**	**0%**	0.724(0.724)	0.092(0.101)	58.720(16.766)
Original	**80**	**5%**	3.620(1.665)	0.060(0.075)	62.080(16.840)
Simplified	**80**	**5%**	0.792(0.907)	0.096(0.101)	62.400(17.332)
Original	**80**	**10%**	4.060(1.463)	0.072(0.087)	55.840(19.089)
Simplified	**80**	**10%**	0.856(0.889)	0.082(0.098)	60.480(17.864)
Original	**90**	**0%**	3.960(1.470)	0.062(0.076)	61.920(19.126)
Simplified	**90**	**0%**	0.944(1.224)	0.072(0.097)	69.660(19.054)
Original	**90**	**5%**	3.700(1.555)	0.071(0.078)	62.640(21.664)
Simplified	**90**	**5%**	0.608(0.820)	0.072(0.095)	69.300(19.435)
Original	**90**	**10%**	3.560(1.593)	0.048(0.075)	62.820(22.600)
Simplified	**90**	**10%**	0.910(1.076)	0.108(0.101)	70.020(19.439)
Original	**100**	**0%**	3.840(1.503)	0.076(0.083)	69.200(20.785)
Simplified	**100**	**0%**	0.872(0.933)	0.062(0.092)	72.600(22.298)
Original	**100**	**5%**	3.780(1.516)	0.084(0.081)	67.400(23.370)
Simplified	**100**	**5%**	1.010(1.205)	0.070(0.095)	76.400(21.358)
Original	**100**	**10%**	3.540(1.619)	0.062(0.082)	76.400(21.548)
Simplified	**100**	**10%**	0.810(0.885)	0.084(0.100)	77.000(17.871)

techniques described in the previous subsection, it is possible to define simplified versions of the estimation of the technology without assuming convexity and assuming convexity, denoted as $\widehat{T}_{SSVF} := \{(\boldsymbol{x},\boldsymbol{y}) \in R_{+}^{m+s} : \boldsymbol{y} \leqslant \widehat{\boldsymbol{y}}_{SSVF}(\boldsymbol{x})\}$ and $\widehat{T}_{CSSVF} := conv(\widehat{T}_{SSVF})$, respectively. Additionally, an estimation of the output-oriented radial efficiency measure can be derived as well as its corresponding ε-insensitive versions. We will check the accuracy of the simplified approach by comparing it to the original one through a computational experience in the next section.

4. Computational experience

In this section, we compare the estimators derived from FDH, DEA, SVF and CSVF by resorting to the well-known mean squared error and bias. Additionally, we will resort to the application of the SVF model and its simplified version, introduced in the previous section of the paper. In particular, we simulate data from a multi-output multi-input production process ($s = 2$ and $m = 2$). We follow the method proposed by Perelman and Santín (2009) to generate the data, fulfilling usual microeconomic regularity conditions.

4.1. Steps for generating the multi-output data set

Following Perelman and Santín (2009), the steps carried out in our computational experience are described next. First, the input data is randomly sampled using a uniform distribution over the interval [5,50] independently for each input and observation. Also, a term z is randomly generated using a uniform distribution over the interval [-1.5,1.5] for each observation, where $z = ln\frac{y_2^*}{y_1^*}$ and y_1^* and y_2^* are the output values located on the production frontier for output 1 and output 2, respectively. Second, we calculate the value for $-ln(y_1^*)$ using:.

$$\begin{aligned}-ln(y_1^*) = {} & -1 + 0.5\left(ln\frac{y_2^*}{y_1^*}\right) + 0.25\left(ln\frac{y_2^*}{y_1^*}\right)^2 - 1.5(lnx_1) \\ & -0.6(lnx_2) + 0.2(lnx_1)^2 + 0.05(lnx_2)^2 \\ & -0.1(lnx_1)(lnx_2) + 0.05(lnx_1)\left(ln\frac{y_2}{y_1}\right) \\ & -0.05(lnx_2)\left(ln\frac{y_2}{y_1}\right)\end{aligned} \tag{16}$$

From the last expression, we obtain the value of the first output on the production frontier: $y_1^* = exp(ln(y_1^*))$. Third, we determine the value for the second output on the production frontier as $y_2^* = exp(ln(y_2^*))$. As Perelman and Santin suggested, we allowed 0%, 5% and 10% of the simulated units to be on the theoretical frontier. For those DMUs that were not on the frontier, we considered a scenario where we calculated a half-normal distribution $lnD = u \sim |N(0, 0.3)|$ for generating the inefficiency term for each output. The simulated outputs capturing the technical inefficiency were calculated as:.

$\hat{y}_1 = y_1^*\frac{1}{exp(lnD)}$ and $\hat{y}_2 = y_2^*\frac{1}{exp(lnD)}$.

Finally, another scenario included random noise for each output of those DMUs that were not located on the frontier by definition. In that scenario, we estimated the output values as:.

$\hat{\hat{y}}_1 = \hat{y}_1\frac{1}{exp(v_1)}$ and $\hat{\hat{y}}_2 = \hat{y}_2\frac{1}{exp(v_2)}$, where $v_1 \sim N(0, 0.01)$, $v_2 \sim N(0, 0.01)$ for each observation.

4.2. Results of the experiments

We ran 50 trials, $e = 1, \ldots, 50$, for each combination of sample size, percentage of units on the true frontier and the consideration or not of random noise to assess the performance of each approach. Cross-validation, based upon five folds, was used for selecting the best combination of hyperparameters for each trial. In our context, the hyperparameters are C, ε and d. We set $C \in \{0.1, 0.5, 1, 2, 5\}$ and $\varepsilon \in \{0, 0.001, 0.01, 0.1, 0.2\}$. Furthermore, $d = 0.1{\cdot}h{\cdot}n$ (rounded), where $h = 1, \ldots, 10$. These values generated 250 different combinations of hyperparameters for the SVF model. The mean squared error (MSE) and the bias were used for measuring the performance of each approach (see Table 2). The same formulae were applied under the SSVF approach.

Table 3 shows the best hyperparameters in our simulations without random noise (mean and standard deviation in brackets). The results with noise were very similar. The first column indicates the type of used algorithm (original or its simplified version), while the second column shows the percentages of simulated DMUs on the frontier. Regarding the results, all the hyperparameters are usually higher in the SVF method than the SSVF method, particularly the hyperparameter C. Additionally, notice that the larger the sample size, the higher the value of hyperparameter d..

Next, Appendix Table 4 includes the scenarios simulated without random noise in the outputs while Appendix Table 5 reports the results corresponding to the scenario with random noise. The structure of both tables is the same: the first three columns indicate the method used, the sample size and the percentages of DMUs on the frontier, in this order. The tables also show the mean and the standard deviation in brackets, the fraction of trials in which SVF improves or equals the MSE of the FDH and the percentage of improvement of this method with respect to the other. The next two columns are similar to the previous ones but comparing CSVF versus DEA. Also, we include two columns that compare the ratio of trials where the results of SSVF/CSSVF are smaller or equal to SVF/CSVF. Finally, we show the same metrics for the bias results.

Regarding the results, the new approach outperforms FDH, reporting superior results in all the considered scenarios. The improvement in MSE in SVF ranged from 20% to 28% on average in simulations without noise and from 19% to 28% on average in simulations with noise. Additionally, the improvements in MSE were higher for SSVF, where they ranged from 24% to 48% on average in simulations without noise and 24% to 50% in simulations with random noise. Likewise, CSVF outperformed DEA, with enhancements for SVF ranging from 17% to 24% on average in simulations without noise and from 16% to 23% on average in simulations with noise. Under the SSVF method, the percentage of enhancement was higher than that corresponding to the SVF method, with enhancements ranging from 21% to 34% on average for simulations without noise and improvements ranging from 22% to 30% on average for simulations without noise. It is worth mentioning that the results associated with the new methodology are good, even for small sample sizes. Moreover, the fraction of trials shows how the SSVF model performed better than the SVF model in most scenarios especially when the sample size grows. These same improvement trends are repeated with respect to the bias.

Fig. 1 shows the mean of the execution time for the scenarios, aggregated by sample size. This metric was calculated as: $\left[\sum_{t=1}^{50}(CV_t + BM_t + SBM_t)\right]\Big/50$, where CV is the time associated with

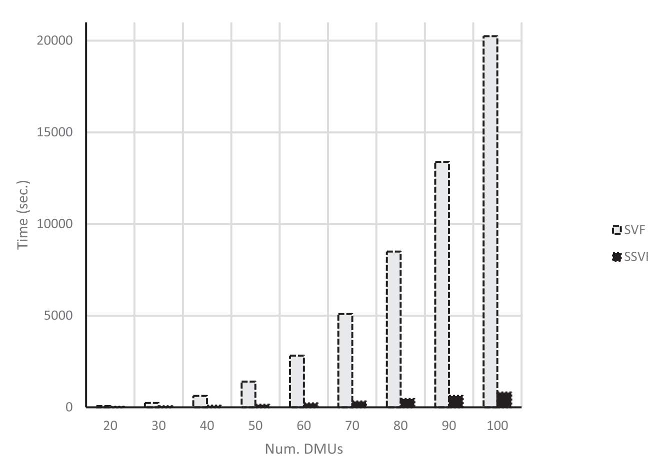

Fig. 1. Execution time in seconds of SVF and SSVF.

the cross-validation process, BM is the time related to the construction of the best model with the hyperparameters selected by the cross-validation and SBM is the spent time to solve the best model. The experiments were conducted on a PC with a 2.3 GHz Intel(R) Xeon(R) CPU E5-2650 v3 with 40 cores, 62 Gigabyte of RAM and an Ubuntu18.04.5 LTS operating system. CPLEX v12.8 was utilized for solving the optimization programs. As we can see in the figure, the execution time of the SVF method was higher than that corresponding to the SSVF method. Also, the figure illustrates how the increase of time is exponential in the SVF method when the number of DMUs augment while, in the SSVF method, the increase is less pronounced.

In summary, the simulations showed that the new methodology achieves better results than the traditional FDH and DEA, both with respect to MSE and Bias. Although the tables seem to indicate that the SSVF model achieved better results in MSE and Bias than in the SVF model, we cannot guarantee the superiority of this method over the other, due to the value of the standard deviation.

Standard Support Vector Regression (SVR), the precursor of the methodology introduced in this paper, is traditionally used with only one response variable, and the multi-response case is then dealt with by modeling each response variable independently of the others (fitting as many models as the number of response variables we have). However, as some previous authors have already shown (see, for example, Vazquez and Walter, 2003, Sanchez-Fernandez et al., 2004, and Han et al., 2012), the multi-response version of SVR makes it viable to take advantage of any existing complex relationships between the response variables to improve the quality of the results provided by the model. In this regard, we could expect to obtain better results in the case of comparing our model, in its multi-output version, with respect to the approach based on fitting as many individual models as outputs we have in the data. To check this result, Table 6 shows the MSE and bias of FDH, DEA, (simplified) SVF and (simplified) CSVF under the multi-output approach

Table 6
Comparison of FDH, DEA, (simplified) SVF and (simplified) CSVF under the multi-output and the single-output approaches.

Size	Frontier	Random noise	MSE FDH	MSE SVF MULTI	MSE SVF SINGLE	MSE DEA	MSE CSVF MULTI	MSE CSVF SINGLE	BIAS FDH	BIAS SVF MULTI	BIAS SVF SINGLE	BIAS DEA	BIAS CSVF MULTI	BIAS CSVF SINGLE
20	0	no	0.764 (0.379)	0.583 (0.359)	1.991 (1.436)	0.391 (0.237)	0.313 (0.225)	1.322 (0.924)	0.620 (0.150)	0.544 (0.154)	1,594 (0,480)	0.457 (0.128)	0.399 (0.135)	0.835 (0.272)
20	0	yes	0.783 (0.404)	0.601 (0.375)	2.341 (1.812)	0.387 (0.241)	0.305 (0.215)	1.634 (1.522)	0.627 (0.150)	0.548 (0.152)	1,700 (0,594)	0.452 (0.127)	0.392 (0.128)	0.907 (0.346)
20	5	no	0.704 (0.352)	0.532 (0.339)	2.382 (1.900)	0.343 (0.216)	0.266 (0.195)	1.461 (1.107)	0.582 (0.138)	0.503 (0.147)	1,657 (0,506)	0.414 (0.119)	0.352 (0.124)	0.882 (0.324)
20	5	yes	0.744 (0.425)	0.558 (0.414)	2.611 (2.175)	0.341 (0.222)	0.261 (0.200)	1.650 (1.261)	0.590 (0.145)	0.505 (0.159)	1,730 (0,572)	0.408 (0.122)	0.341 (0.129)	0.923 (0.336)
20	10	no	0.671 (0.356)	0.502 (0.342)	2.408 (1.860)	0.313 (0.211)	0.243 (0.191)	1.531 (1.077)	0.553 (0.140)	0.476 (0.149)	1,692 (0,507)	0.379 (0.115)	0.324 (0.122)	0.884 (0.301)
20	10	yes	0.680 (0.365)	0.504 (0.294)	2.213 (1.562)	0.302 (0.210)	0.237 (0.184)	1.647 (1.157)	0.558 (0.138)	0.480 (0.142)	1,737 (0,534)	0.369 (0.114)	0.316 (0.122)	0.870 (0.299)
30	0	no	0.681 (0.255)	0.478 (0.215)	1.873 (0.999)	0.311 (0.136)	0.233 (0.116)	1.178 (0.562)	0.592 (0.097)	0.499 (0.101)	1,585 (0,323)	0.406 (0.086)	0.343 (0.088)	0.817 (0.194)
30	0	yes	0.684 (0.243)	0.493 (0.203)	2.113 (1.386)	0.301 (0.145)	0.231 (0.132)	1.413 (0.718)	0.591 (0.094)	0.502 (0.089)	1,694 (0,373)	0.390 (0.092)	0.329 (0.089)	0.846 (0.214)
30	5	no	0.631 (0.260)	0.438 (0.219)	1.974 (0.996)	0.263 (0.134)	0.193 (0.113)	1.361 (0.625)	0.550 (0.100)	0.456 (0.098)	1,682 (0,331)	0.349 (0.079)	0.286 (0.075)	0.834 (0.191)
30	5	yes	0.646 (0.273)	0.452 (0.212)	2.174 (1.113)	0.246 (0.122)	0.181 (0.106)	1.569 (0.799)	0.549 (0.098)	0.454 (0.090)	1,782 (0,375)	0.333 (0.078)	0.272 (0.073)	0.882 (0.208)
30	10	no	0.601 (0.233)	0.402 (0.182)	2.028 (1.046)	0.241 (0.132)	0.170 (0.105)	1.431 (0.629)	0.526 (0.092)	0.428 (0.087)	1,717 (0,337)	0.326 (0.076)	0.262 (0.072)	0.852 (0.203)
30	10	yes	0.613 (0.257)	0.406 (0.197)	2.340 (1.287)	0.238 (0.136)	0.164 (0.108)	1.585 (0.701)	0.529 (0.098)	0.428 (0.083)	1,779 (0,352)	0.320 (0.081)	0.255 (0.073)	0.892 (0.193)
40	0	no	0.621 (0.174)	0.450 (0.170)	2.343 (1.149)	0.252 (0.128)	0.196 (0.125)	1.426 (0.497)	0.579 (0.080)	0.492 (0.088)	1,744 (0,253)	0.369 (0.067)	0.309 (0.073)	0.927 (0.192)
40	0	yes	0.639 (0.190)	0.456 (0.180)	2.545 (1.178)	0.242 (0.146)	0.183 (0.139)	1.773 (0.652)	0.583 (0.081)	0.488 (0.087)	1,916 (0,320)	0.346 (0.068)	0.285 (0.072)	0.968 (0.205)
40	5	no	0.557 (0.163)	0.385 (0.154)	2.276 (1.227)	0.218 (0.126)	0.165 (0.121)	1.488 (0.505)	0.536 (0.076)	0.439 (0.077)	1,776 (0,242)	0.324 (0.062)	0.263 (0.065)	0.927 (0.195)
40	5	yes	0.569 (0.180)	0.380 (0.172)	2.434 (1.218)	0.218 (0.143)	0.162 (0.135)	1.708 (0.592)	0.538 (0.081)	0.429 (0.082)	1,887 (0,288)	0.312 (0.066)	0.251 (0.061)	0.968 (0.217)
40	10	no	0.519 (0.167)	0.352 (0.155)	2.415 (1.282)	0.191 (0.124)	0.145 (0.121)	1.632 (0.598)	0.502 (0.077)	0.405 (0.078)	1,846 (0,260)	0.289 (0.056)	0.232 (0.059)	0.948 (0.208)
40	10	yes	0.525 (0.161)	0.347 (0.150)	2.700 (1.575)	0.183 (0.118)	0.142 (0.114)	1.825 (0.710)	0.502 (0.074)	0.398 (0.074)	1,918 (0,301)	0.278 (0.057)	0.228 (0.053)	0.990 (0.217)
50	0	no	0.572 (0.196)	0.389 (0.158)	2.404 (1.036)	0.209 (0.090)	0.156 (0.084)	1.472 (0.543)	0.550 (0.082)	0.447 (0.071)	1,802 (0,320)	0.331 (0.054)	0.263 (0.056)	0.968 (0.180)
50	0	yes	0.586 (0.209)	0.389 (0.164)	2.724 (1.360)	0.195 (0.085)	0.143 (0.082)	1.805 (0.684)	0.552 (0.085)	0.440 (0.075)	1,950 (0,351)	0.305 (0.055)	0.242 (0.054)	1.021 (0.220)
50	5	no	0.531 (0.196)	0.353 (0.148)	2.495 (1.024)	0.181 (0.076)	0.135 (0.068)	1.581 (0.589)	0.520 (0.085)	0.417 (0.068)	1,847 (0,313)	0.297 (0.048)	0.238 (0.044)	0.986 (0.168)
50	5	yes	0.542 (0.190)	0.353 (0.155)	2.850 (1.170)	0.174 (0.081)	0.131 (0.069)	1.872 (0.696)	0.519 (0.079)	0.408 (0.072)	1,977 (0,332)	0.275 (0.048)	0.220 (0.043)	1.043 (0.195)
50	10	no	0.483 (0.180)	0.305 (0.133)	2.622 (1.090)	0.152 (0.068)	0.115 (0.064)	1.705 (0.657)	0.479 (0.079)	0.371 (0.062)	1,895 (0,330)	0.256 (0.041)	0.204 (0.042)	1.014 (0.186)
50	10	yes	0.488 (0.200)	0.309 (0.154)	2.944 (1.071)	0.145 (0.075)	0.109 (0.065)	1.965 (0.719)	0.478 (0.083)	0.371 (0.071)	2,003 (0,370)	0.242 (0.048)	0.195 (0.045)	1.072 (0.192)
60	0	no	0.557 (0.193)	0.356 (0.140)	2.693 (1.095)	0.183 (0.090)	0.132 (0.070)	1.710 (0.787)	0.538 (0.075)	0.429 (0.065)	1,898 (0,301)	0.308 (0.054)	0.247 (0.052)	1.034 (0.151)

(continued on next page)

Table 6 (*continued*)

Size	Frontier	Random noise	MSE						BIAS					
			FDH	SVF		DEA	CSVF		FDH	SVF		DEA	CSVF	
				MULTI	SINGLE		MULTI	SINGLE		MULTI	SINGLE		MULTI	SINGLE
60	0	yes	0.571	0.345	3.085	0.176	0.124	2.010	0.539	0.419	2,041	0.288	0.229	1.101
			(0.203)	(0.128)	(1.336)	(0.099)	(0.072)	(0.762)	(0.076)	(0.064)	(0,318)	(0.062)	(0.056)	(0.192)
60	5	no	0.521	0.327	2.779	0.161	0.115	1.715	0.507	0.394	1,917	0.274	0.218	1.053
			(0.188)	(0.141)	(1.063)	(0.086)	(0.066)	(0.582)	(0.075)	(0.065)	(0,260)	(0.049)	(0.046)	(0.145)
60	5	yes	0.520	0.310	3.356	0.154	0.111	2.114	0.504	0.381	2,046	0.254	0.205	1.125
			(0.181)	(0.122)	(1.576)	(0.098)	(0.075)	(0.889)	(0.071)	(0.060)	(0,305)	(0.056)	(0.048)	(0.180)
60	10	no	0.480	0.294	2.848	0.138	0.098	1.829	0.471	0.360	1,969	0.238	0.189	1.070
			(0.191)	(0.133)	(1.122)	(0.080)	(0.059)	(0.649)	(0.074)	(0.062)	(0,279)	(0.044)	(0.040)	(0.154)
60	10	yes	0.495	0.303	3.039	0.137	0.101	2.039	0.471	0.353	2,045	0.223	0.183	1.115
			(0.215)	(0.167)	(1.504)	(0.096)	(0.077)	(0.787)	(0.077)	(0.074)	(0,293)	(0.051)	(0.043)	(0.173)
70	0	no	0.583	0.346	2.617	0.174	0.122	1.661	0.546	0.425	1,928	0.296	0.228	1.055
			(0.272)	(0.134)	(0.647)	(0.108)	(0.088)	(0.428)	(0.077)	(0.062)	(0,232)	(0.057)	(0.052)	(0.139)
70	0	yes	0.594	0.340	3.054	0.158	0.114	2.047	0.543	0.407	2,108	0.262	0.206	1.123
			(0.281)	(0.156)	(0.989)	(0.120)	(0.103)	(0.461)	(0.078)	(0.065)	(0,237)	(0.059)	(0.045)	(0.170)
70	5	no	0.539	0.305	2.759	0.147	0.106	1.831	0.506	0.382	2,006	0.250	0.193	1.085
			(0.266)	(0.127)	(0.731)	(0.104)	(0.090)	(0.521)	(0.077)	(0.063)	(0,264)	(0.051)	(0.043)	(0.157)
70	5	yes	0.536	0.299	3.193	0.135	0.099	2.219	0.503	0.374	2,152	0.232	0.189	1.149
			(0.260)	(0.124)	(0.902)	(0.090)	(0.080)	(0.738)	(0.079)	(0.066)	(0,317)	(0.053)	(0.043)	(0.168)
70	10	no	0.505	0.281	2.909	0.125	0.095	1.928	0.477	0.358	2,041	0.219	0.174	1.115
			(0.257)	(0.118)	(0.867)	(0.091)	(0.082)	(0.509)	(0.072)	(0.059)	(0,242)	(0.044)	(0.037)	(0.167)
70	10	yes	0.525	0.290	3.207	0.130	0.102	2.182	0.477	0.350	2,145	0.205	0.170	1.167
			(0.270)	(0.162)	(1.000)	(0.130)	(0.120)	(0.727)	(0.075)	(0.066)	(0,302)	(0.053)	(0.040)	(0.181)
80	0	no	0.511	0.308	2.918	0.151	0.112	1.756	0.528	0.406	1,975	0.279	0.216	1.107
			(0.155)	(0.098)	(1.048)	(0.061)	(0.057)	(0.634)	(0.068)	(0.061)	(0,228)	(0.047)	(0.045)	(0.154)
80	0	yes	0.517	0.292	3.260	0.139	0.103	2.080	0.526	0.392	2,107	0.250	0.203	1.154
			(0.160)	(0.100)	(1.423)	(0.066)	(0.062)	(0.748)	(0.075)	(0.070)	(0,246)	(0.061)	(0.053)	(0.174)
80	5	no	0.464	0.274	3.020	0.126	0.092	1.892	0.491	0.370	2,028	0.237	0.182	1.124
			(0.136)	(0.085)	(1.123)	(0.056)	(0.051)	(0.655)	(0.063)	(0.056)	(0,239)	(0.040)	(0.036)	(0.169)
80	5	yes	0.463	0.265	3.384	0.114	0.084	2.191	0.488	0.357	2,159	0.214	0.172	1.187
			(0.142)	(0.097)	(1.581)	(0.054)	(0.049)	(0.679)	(0.066)	(0.060)	(0,246)	(0.046)	(0.037)	(0.186)
80	10	no	0.431	0.246	3.214	0.110	0.080	2.010	0.458	0.337	2,077	0.205	0.163	1.158
			(0.128)	(0.076)	(1.215)	(0.054)	(0.048)	(0.649)	(0.059)	(0.051)	(0,246)	(0.037)	(0.034)	(0.166)
80	10	yes	0.440	0.247	3.569	0.104	0.081	2.327	0.459	0.331	2,192	0.190	0.163	1.217
			(0.127)	(0.082)	(1.346)	(0.058)	(0.051)	(0.690)	(0.059)	(0.050)	(0,245)	(0.039)	(0.031)	(0.159)
90	0	no	0.496	0.289	2.925	0.131	0.085	1.849	0.519	0.389	2,034	0.258	0.189	1.122
			(0.141)	(0.092)	(0.794)	(0.048)	(0.034)	(0.470)	(0.057)	(0.051)	(0,201)	(0.036)	(0.029)	(0.138)
90	0	yes	0.493	0.283	3.293	0.120	0.085	2.287	0.515	0.379	2,199	0.233	0.185	1.184
			(0.151)	(0.099)	(0.929)	(0.049)	(0.040)	(0.688)	(0.062)	(0.053)	(0,263)	(0.044)	(0.035)	(0.153)
90	5	no	0.463	0.261	3.101	0.114	0.080	1.946	0.489	0.356	2,065	0.224	0.171	1.154
			(0.139)	(0.087)	(0.797)	(0.046)	(0.035)	(0.513)	(0.055)	(0.043)	(0,202)	(0.031)	(0.025)	(0.130)
90	5	yes	0.469	0.257	3.545	0.108	0.078	2.318	0.486	0.345	2,219	0.203	0.165	1.231
			(0.142)	(0.097)	(0.948)	(0.055)	(0.049)	(0.646)	(0.058)	(0.048)	(0,225)	(0.039)	(0.028)	(0.134)
90	10	no	0.424	0.229	3.232	0.093	0.062	2.084	0.453	0.321	2,121	0.188	0.145	1.184
			(0.132)	(0.086)	(0.832)	(0.043)	(0.031)	(0.520)	(0.053)	(0.048)	(0,203)	(0.029)	(0.025)	(0.141)
90	10	yes	0.428	0.229	3.639	0.088	0.067	2.382	0.451	0.312	2,242	0.171	0.148	1.254
			(0.138)	(0.093)	(0.917)	(0.048)	(0.037)	(0.540)	(0.059)	(0.052)	(0,203)	(0.034)	(0.026)	(0.141)
100	0	no	0.476	0.267	3.113	0.128	0.086	1.843	0.508	0.377	2,025	0.249	0.186	1.162
			(0.121)	(0.095)	(0.821)	(0.074)	(0.068)	(0.576)	(0.051)	(0.045)	(0,223)	(0.033)	(0.028)	(0.123)
100	0	yes	0.477	0.253	3.607	0.115	0.082	2.312	0.501	0.355	2,218	0.213	0.174	1.257
			(0.125)	(0.105)	(0.915)	(0.085)	(0.079)	(0.748)	(0.054)	(0.045)	(0,272)	(0.041)	(0.028)	(0.151)
100	5	no	0.432	0.236	3.165	0.107	0.073	1.979	0.471	0.340	2,081	0.209	0.158	1.182
			(0.111)	(0.097)	(0.855)	(0.071)	(0.069)	(0.517)	(0.046)	(0.043)	(0,215)	(0.027)	(0.025)	(0.135)
100	5	yes	0.430	0.226	3.670	0.097	0.072	2.354	0.465	0.323	2,228	0.186	0.155	1.275
			(0.108)	(0.086)	(1.033)	(0.064)	(0.059)	(0.682)	(0.048)	(0.039)	(0,270)	(0.031)	(0.021)	(0.151)
100	10	no	0.392	0.205	3.269	0.092	0.065	2.025	0.435	0.304	2,096	0.177	0.141	1.207
			(0.108)	(0.084)	(0.916)	(0.071)	(0.070)	(0.555)	(0.043)	(0.034)	(0,214)	(0.025)	(0.020)	(0.148)
100	10	yes	0.399	0.203	3.674	0.090	0.071	2.325	0.433	0.294	2,214	0.164	0.149	1.272
			(0.108)	(0.091)	(1.058)	(0.074)	(0.072)	(0.629)	(0.043)	(0.037)	(0,251)	(0.025)	(0.026)	(0.159)

Table 7
Comparison of execution time (mean) of FDH, DEA, SVF and SSVF.

Scenario			Time (sec.)			
Size	Frontier	Random noise	FDH	DEA	SVF	SSVF
20	0%	no	0.48	0.1	64.68	30.06
20	0%	yes	0.44	0.14	65.46	28.94
20	5%	no	0.42	0.16	65.3	26.04
20	5%	yes	0.44	0.14	65.04	26
20	10%	no	0.5	0.1	66.68	38.26
20	10%	yes	0.46	0.14	64.82	42.8
30	0%	no	0.68	0.2	226.28	48.16
30	0%	yes	0.6	0.26	246.22	50.58
30	5%	no	0.66	0.22	226.36	46.98
30	5%	yes	0.68	0.2	228.3	47.18
30	10%	no	0.68	0.18	225.96	69.86
30	10%	yes	0.64	0.22	241.14	72.3
40	0%	no	0.88	0.24	620.66	83.8
40	0%	yes	0.88	0.26	625.88	83.76
40	5%	no	0.88	0.26	617.92	80.76
40	5%	yes	0.92	0.22	629.98	81.2
40	10%	no	0.9	0.24	627.96	103.02
40	10%	yes	0.72	0.42	633.76	110.02
50	0%	no	1.04	0.36	1437.72	133.98
50	0%	yes	1.06	0.32	1403.52	133.5
50	5%	no	1.02	0.38	1383.02	132.74
50	5%	yes	1	0.38	1399.56	132.52
50	10%	no	1.02	0.34	1392.44	158.66
50	10%	yes	1	0.38	1418.16	159.5
60	0%	no	1.18	0.42	2827.02	209.74
60	0%	yes	1.14	0.48	2907.6	210.42
60	5%	no	1.24	0.58	2885.06	220.92
60	5%	yes	1.22	0.62	2738.88	219.66
60	10%	no	1.3	0.5	2793.6	232.26
60	10%	yes	1.18	0.44	2783.3	230.52
70	0%	no	1.56	0.6	5093.72	307.3
70	0%	yes	1.42	0.76	5064.26	306.24
70	5%	no	1.62	0.54	5041.26	328.92
70	5%	yes	1.54	0.62	5029.46	337.9
70	10%	no	1.46	0.7	5139.62	330.7
70	10%	yes	1.44	0.72	5149.9	328.46
80	0%	no	1.78	0.72	8744.5	435.22
80	0%	yes	1.72	0.74	8533.1	435.94
80	5%	no	1.82	0.64	8650.18	459.62
80	5%	yes	1.84	0.66	8405.92	459.94
80	10%	no	1.8	0.7	8329.84	456.28
80	10%	yes	1.76	0.72	8306.38	462.66
90	0%	no	2.22	0.68	13092.18	601.58
90	0%	yes	2.06	0.9	13382.6	597.32
90	5%	no	2.08	0.88	13862.18	642.58
90	5%	yes	2.1	0.84	13401.34	621.7
90	10%	no	2.08	0.8	13297.62	621.08
90	10%	yes	2.06	0.86	13281.56	625.6
100	0%	no	2.28	0.9	19,999	794.3
100	0%	yes	2.34	0.9	20006.06	795.76
100	5%	no	2.34	0.9	20392.2	832.18
100	5%	yes	2.36	0.84	19970.82	822.14
100	10%	no	2.36	0.88	20370.76	817.6
100	10%	yes	2.32	0.9	20731.42	821.54

(the model introduced in this paper) and the single-output approach (the model introduced in Valero-Carreras et al., 2021). Seeking simplicity, we resorted to the simplified version of Support Vector Frontiers because, as was shown above, the original and the simplified versions seem to yield similar levels of performance, with the simplified version demanding less computational effort. Table 6 shows the mean and the standard deviation in brackets of MSE and bias for all the simulated scenarios in this section. For these simulations, the figures in Table 6 appear to demonstrate the superiority of the methodology based on the multi-output approach in comparison with the single-output approach.

Additionally, in Table 7, we illustrate the differences in terms of execution time that we can find when FDH, DEA, SVF and SSVF are applied on the simulated databases of this computational section. The results confirm expectations. The SVF and SSVF methods, as well as their convexified versions, inherit the complex validation procedure from the standard Support Vector Regression, which is the machine learning technique that was adapted in this paper to estimate technical efficiency. The cross-validation process carried out by standard SVR was described at the beginning of Section 3 above. Thus, the process followed by the SVF and SSVF approaches to determine values for C and ε is based upon the evaluation of a grid of possible values for both parameters. Once the values for C and ε are fixed, we must solve model for the different folds considered (subsamples). Once the best combination (C, ε) is determined, i.e., that associated with the minimum mean squared error, technical efficiency must be calculated for each of the n DMUs belonging to the original data sample resorting to model or any of the models described next in Section 5. Notice that these models are based on linear programming for the convex case and mixed-integer linear programming for the non-convex case. Thus, from a computational viewpoint, the SVF and SSVF techniques seem to be more complex than standard DEA and FDH. It implies that the corresponding execution time is considerably higher when SVF, and even SSVF, is applied in comparison with DEA and FDH.

Finally, it is worth mentioning the main fundamentals that somehow explain why the new approach obtains better results than standard FDH and DEA. Overall, the success of SVF, as an adaptation of SVR for estimating technical efficiency, is based on two principles. First, the minimization of the 'complex' notion of structural risk (see Vapnik, 1995, 1998) and its implementation through an optimization model (model (8)), which depends on two parameters (C and ε). Second, the determination of these two parameters through a computer-intensive procedure: the cross-validation method. In this regard, the essence of cross-validating is to fit the model on a subset of the data and check its validity by evaluating another different subset of the original database. And this step must be repeated for each subsample when resorting to V-fold cross-validation. In this sense, if we compare the SVF-based approach regarding FDH and DEA, which directly minimize the empirical error associated with the observations (see Kuosmanen and Johnson, 2010) and do not apply any cross-validation process, it is to be expected that interesting findings could be obtained with respect to reduction in bias and mean squared error. Nevertheless, a computational experience was necessary to really confirm this previous intuition.

5. Other efficiency measures

In the previous section, we paid attention to the output-oriented radial model. However, in the last decades, many different technical efficiency measures have been introduced in the DEA literature. In this new section, we describe how to implement well-known efficiency measures under Support Vector Frontiers. Nevertheless, seeking brevity, we will go on to exclusively show the standard definition of efficiency for the evaluation of an observation $(\mathbf{x}_i, \mathbf{y}_i)$, without resorting to the concept of ε-insensitive technically efficiency, and only assuming the SSVF technique, instead of the non-simplified version (SVF).

The input-oriented radial model (Banker et al., 1984):.

$$
\begin{aligned}
&\min && \theta && && (17.0)\\
&s.t. && && && (17.1)\\
& && \sum_{\substack{l_1=1,\ldots,k_1\\ \vdots\\ l_m=1,\ldots,k_m}} \lambda_{l_1\ldots l_m} a^{(j)}_{l_1\ldots l_m} \leqslant \theta x_i^{(j)}, && j=1,\ldots,m && (17.2)\\
& && \sum_{\substack{l_1=1,\ldots,k_1\\ \vdots\\ l_m=1,\ldots,k_m}} \lambda_{l_1\ldots l_m} \hat{y}^{(r)}_{SSVF}(\boldsymbol{a}_{l_1\ldots l_m}) \geqslant y_i^{(r)}, && r=1,\ldots,s && (17.3)\\
& && \sum_{\substack{l_1=1,\ldots,k_1\\ \vdots\\ l_m=1,\ldots,k_m}} \lambda_{k_1\ldots k_m} = 1, && && (17.4)\\
& && \lambda_{l_1\ldots l_m} \in \{0,1\}, && \forall l_j=1,\ldots,k_j, \forall j=1,\ldots,m && (17.5)
\end{aligned}
\tag{17}
$$

The directional distance function (DDF) (Chambers et al., 1998) projects $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ onto the frontier of the technology following the direction $\boldsymbol{g} = (-\boldsymbol{g}^-, \boldsymbol{g}^+)$, with $\boldsymbol{g}^- \in R^m_+$ and $\boldsymbol{g}^+ \in R^s_+$..

$$
\begin{aligned}
&\max && \beta && && (18.0)\\
&s.t. && && && (18.1)\\
& && \sum_{\substack{l_1=1,\ldots,k_1\\ \vdots\\ l_m=1,\ldots,k_m}} \lambda_{l_1\ldots l_m} a^{(j)}_{l_1\ldots l_m} \leqslant x_i^{(j)} - \beta g^{-(j)}, && j=1,\ldots,m && (18.2)\\
& && \sum_{\substack{l_1=1,\ldots,k_1\\ \vdots\\ l_m=1,\ldots,k_m}} \lambda_{l_1\ldots l_m} \hat{y}^{(r)}_{SSVF}(\boldsymbol{a}_{l_1\ldots l_m}) \geqslant y_i^{(r)} + \beta g^{+(r)}, && r=1,\ldots,s && (18.3)\\
& && \sum_{\substack{l_1=1,\ldots,k_1\\ \vdots\\ l_m=1,\ldots,k_m}} \lambda_{l_1\ldots l_m} = 1, && && (18.4)\\
& && \lambda_{l_1\ldots l_m} \in \{0,1\}, && \forall l_j=1,\ldots,k_j, \forall j=1,\ldots,m && (18.5)
\end{aligned}
\tag{18}
$$

The weighted additive (WA) model (Lovell and Pastor, 1995) maximizes a weighted aggregation of input-specific and output-specific inefficiencies (slacks).

$$
\begin{aligned}
&\max && \sum_{j=1}^{m} \rho^{-(j)} s^{-(j)} + \sum_{r=1}^{s} \rho^{+(r)} s^{+(r)} && && (19.0)\\
&s.t. && \sum_{\substack{l_1=1,\ldots,k_1\\ \vdots\\ l_m=1,\ldots,k_m}} \lambda_{l_1\ldots l_m} a^{(j)}_{l_1\ldots l_m} \leqslant x_i^{(j)} - s^{-(j)}, && j=1,\ldots,m && (19.1)\\
& && \sum_{\substack{l_1=1,\ldots,k_1\\ \vdots\\ l_m=1,\ldots,k_m}} \lambda_{l_1\ldots l_m} \hat{y}^{(r)}_{SSVF}(\boldsymbol{a}_{l_1\ldots l_m}) \geqslant y_i^{(r)} + s^{+(r)}, && r=1,\ldots,s && (19.2)\\
& && \sum_{\substack{l_1=1,\ldots,k_1\\ \vdots\\ l_m=1,\ldots,k_m}} \lambda_{l_1\ldots l_m} = 1, && && (19.3)\\
& && \lambda_{l_1\ldots l_m} \in \{0,1\}, && \forall l_j=1,\ldots,k_j, \forall j=1,\ldots,m && (19.4)\\
& && s^{-(j)}, s^{+(r)} \geqslant 0 && j=1,\ldots,m, r=1,\ldots,s && (19.5)
\end{aligned}
\tag{19}
$$

Table 8
Efficiency measures obtained from the empirical example.

	OUTPUT ORIENTED RADIAL MODEL						INPUT ORIENTED RADIAL MODEL						DIRECTIONAL DISTANCE FUNCTION						WEIGHTED ADDITIVE					
DMU	FDH	DEA	SSVF	CSSVF	SSVF EPS. EFF.	CSSVF EPS. EFF.	FDH	DEA	SSVF	CSSVF	SSVF EPS. EFF.	CSSVF EPS. EFF.	FDH	DEA	SSVF	CSSVF	SSVF EPS. EFF.	CSSVF EPS. EFF.	FDH	DEA	SSVF	CSSVF	SSVF EPS. EFF.	CSSVF EPS. EFF.
Export-Import Bank	1,000	1,000	1,084	1,084	Yes	Yes	1,000	1,000	1,000	1,000	Yes	Yes	0.000	0.000	0.000	0.000	Yes	Yes	0.000	0.000	0.025	0.025	Yes	Yes
Bank of Taiwan	1,000	1,000	1,000	1,000	Yes	Yes	1,000	1,000	1,000	1,000	Yes	Yes	0.000	0.000	0.000	0.000	Yes	Yes	0.000	0.000	0.000	0.000	Yes	Yes
Taipei Fubon Bank	1,000	1,000	1,000	1,196	Yes	Yes	1,000	1,000	0.960	0.739	Yes	Yes	0.000	0.000	0.000	0.123	Yes	Yes	0.000	0.000	0.000	0.085	Yes	Yes
Bank of Kaohsiung	1,000	1,362	1,000	1,482	Yes	Yes	1,000	0.647	0.798	0.570	Yes	Yes	0.000	0.179	0.000	0.226	Yes	Yes	0.000	0.026	0.036	0.044	Yes	Yes
Land Bank	1,000	1,000	1,000	1,018	Yes	Yes	1,000	1,000	0.968	0.964	Yes	Yes	0.000	0.000	0.000	0.013	Yes	Yes	0.000	0.000	0.000	0.048	Yes	Yes
Cooperative Bank	1,000	1,000	1,000	1,051	Yes	No	1,000	1,000	0.805	0.805	Yes	No	0.000	0.000	0.000	0.040	Yes	No	0.000	0.000	0.000	0.127	Yes	No
First Bank	1,000	1,000	1,000	1,209	Yes	No	1,000	1,000	0.982	0.784	Yes	No	0.000	0.000	0.000	0.107	Yes	No	0.000	0.000	0.018	0.084	Yes	No
Hua Nan Bank	1,000	1,054	1,069	1,285	No	No	1,000	0.944	0.963	0.766	No	No	0.000	0.028	0.028	0.129	No	No	0.000	0.065	0.066	0.122	No	No
Chang Hwa Bank	1,000	1,062	1,000	1,227	Yes	No	1,000	0.935	0.961	0.802	Yes	No	0.000	0.032	0.000	0.106	Yes	No	0.000	0.093	0.000	0.137	Yes	No
Mega Bank	1,000	1,000	1,000	1,039	Yes	Yes	1,000	1,000	0.966	0.953	Yes	Yes	0.000	0.000	0.000	0.023	Yes	Yes	0.000	0.000	0.000	0.019	Yes	Yes
Cathay United Bank	1,000	1,384	1,177	1,548	No	No	1,000	0.710	0.765	0.623	No	No	0.000	0.165	0.142	0.223	No	No	0.000	0.139	0.138	0.178	No	No
The Shanghai Bank	1,000	1,110	1,175	1,465	Yes	No	1,000	0.887	0.916	0.610	Yes	No	0.000	0.056	0.084	0.212	Yes	No	0.000	0.023	0.037	0.047	Yes	No
Union Bank	1,000	1,691	1,392	1,906	Yes	No	1,000	0.499	0.937	0.419	Yes	No	0.000	0.290	0.063	0.352	Yes	No	0.000	0.091	0.107	0.108	Yes	No
Far Eastern Bank	1,000	1,000	1,000	1,120	Yes	Yes	1,000	1,000	0.870	0.806	Yes	Yes	0.000	0.000	0.000	0.084	Yes	Yes	0.000	0.000	0.018	0.018	Yes	Yes
E. Sun Bank	1,000	1,123	1,000	1,412	Yes	No	1,000	0.842	0.967	0.630	Yes	No	0.000	0.076	0.000	0.212	Yes	No	0.000	0.040	0.000	0.079	Yes	No
Cosmos Bank	1,266	2,244	1,266	2,638	Yes	No	0.228	0.228	0.228	0.228	Yes	No	0.168	0.520	0.266	0.601	Yes	No	0.064	0.064	0.079	0.079	Yes	No
Taishin Bank	1,000	1,194	1,000	1,475	Yes	No	1,000	0.824	0.712	0.641	Yes	No	0.000	0.092	0.000	0.205	Yes	No	0.000	0.124	0.000	0.151	Yes	No
Ta Chong Bank	1,000	1,137	1,000	1,359	Yes	No	1,000	0.856	0.992	0.681	Yes	No	0.000	0.070	0.000	0.169	Yes	No	0.000	0.053	0.000	0.073	Yes	No
Jih Sun Bank	1,000	1,628	1,255	1,902	Yes	No	1,000	0.482	0.828	0.354	Yes	No	0.000	0.284	0.172	0.374	Yes	No	0.000	0.047	0.055	0.063	Yes	No
Entie Bank	1,000	1,194	1,252	1,571	Yes	No	1,000	0.811	0.928	0.556	Yes	No	0.000	0.096	0.072	0.250	Yes	No	0.000	0.044	0.034	0.066	Yes	No
China Trust Bank	1,000	1,000	1,000	1,432	Yes	No	1,000	1,000	0.954	0.678	Yes	No	0.000	0.000	0.000	0.185	Yes	No	0.000	0.000	0.082	0.187	Yes	No
Sunny Bank	1,000	1,436	1,044	1,656	Yes	No	1,000	0.601	0.723	0.483	Yes	No	0.000	0.208	0.044	0.292	Yes	No	0.000	0.066	0.053	0.084	Yes	No
Bank of Panhsin	1,000	1,746	1,000	2,039	Yes	No	1,000	0.373	0.889	0.280	Yes	No	0.000	0.340	0.000	0.422	Yes	No	0.000	0.056	0.041	0.071	Yes	No
Taiwan Business Bank	1,000	1,033	1,000	1,149	Yes	No	1,000	0.967	0.996	0.861	Yes	No	0.000	0.016	0.000	0.072	Yes	No	0.000	0.050	0.000	0.094	Yes	No
Taichung Bank	1,000	1,295	1,000	1,420	Yes	No	1,000	0.725	0.927	0.640	Yes	No	0.000	0.142	0.000	0.192	Yes	No	0.000	0.046	0.035	0.066	Yes	No
China Development	1,000	1,000	1,000	1,352	Yes	Yes	1,000	1,000	0.414	0.414	Yes	Yes	0.000	0.000	0.000	0.226	Yes	Yes	0.000	0.000	0.015	0.015	Yes	Yes
Hwatai Bank	1,000	1,731	1,000	1,887	Yes	Yes	1,000	0.346	0.303	0.303	Yes	Yes	0.000	0.355	0.000	0.409	Yes	Yes	0.000	0.032	0.047	0.047	Yes	Yes
Cota Bank	1,000	1,638	1,039	1,879	Yes	Yes	1,000	0.488	0.454	0.454	Yes	Yes	0.000	0.284	0.039	0.403	Yes	Yes	0.000	0.032	0.051	0.051	Yes	Yes
Industrial Bank of Taiwan	1,000	1,000	1,328	1,767	Yes	Yes	1,000	1,000	0.716	0.716	Yes	Yes	0.000	0.000	0.284	0.284	Yes	Yes	0.000	0.000	0.031	0.031	Yes	Yes
Bank SinoPac	1,000	1,111	1,000	1,276	Yes	No	1,000	0.893	0.968	0.763	Yes	No	0.000	0.054	0.000	0.128	Yes	No	0.000	0.043	0.000	0.068	Yes	No
Shin Kong Bank	1,000	1,347	1,000	1,480	Yes	No	1,000	0.702	0.938	0.624	Yes	No	0.000	0.160	0.000	0.210	Yes	No	0.000	0.081	0.071	0.102	Yes	No
Mean	1,009	1,243	1,067	1,462			0.975	0.799	0.833	0.650			0.005	0.111	0.039	0.202			0.002	0.039	0.034	0.076		

If, additionally, convexity is assumed, then the constraints $\lambda_{l_1 \ldots l_m} \in \{0,1\}, \forall l_j = 1,\ldots,k_j, \forall j = 1,\ldots,m$, should be substituted by $\lambda_{l_1 \ldots l_m} \geqslant 0, \forall l_j = 1,\ldots,k_j, \forall j = 1,\ldots,m$, in the above optimization programs.

6. Empirical illustration

In this section, we resort to a real dataset to illustrate the performance of different technical efficiency measures when the SSVF technique is applied. To do this, we use the data corresponding to 31 Taiwanese banks for the year 2010, previously studied by Juo et. al (2015).[1] In this empirical context, inputs were FINANCIAL FUNDS (x_1), LABOR (x_2), and physical CAPITAL (x_3), while outputs were financial INVESTMENTS (y_1) and LOANS (y_2). All monetary variables are measured in million TWD, while labor is the number of employees. A complete discussion of the statistical sources and variable specifications can be found in Juo et al. (2015).

Regarding the computational task, to determine the best model, we implemented a 5-fold cross-validation process with $C \in \{0.0001, 0.0005, 0.001, 0.01, 0.1, 1, 10\}$, $\varepsilon \in \{0, 0.001, 0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.2\}$ and $d \in \{3, 6, 9, 12, 16, 19, 22, 25, 28, 31\}$. The best combination was: $C = 10, \varepsilon = 0.04, d = 25$ after rescaling the outputs to deal with values between zero and one.

Table 8 shows the results for different efficiency measures. The first column indicates the name of the assessed bank. Next, we present four blocks with six columns in each. These blocks correspond to the different efficiency models that we consider in our study: the output-oriented radial model, the input-oriented radial model, the directional distance function and the weighted additive model. In the case of the DDF, the selected directional vector was $\boldsymbol{g}^- = \boldsymbol{x}_i$ and $\boldsymbol{g}^+ = \boldsymbol{y}_i$, whereas in the case of the Weighted Additive model, the weights correspond to those used in the Range-Adjusted Measure (RAM) by Cooper et al. (1999). Each block comprises six columns associated with different approaches: FDH, DEA, SSVF and CSSVF. The last two columns indicate whether the corresponding DMU is ε-insensitive technically efficient or not.

Regarding the results, we can see how SSVF and CSSVF detect more inefficiency than FDH and DEA, respectively: the means are higher. In a unit-to-unit analysis, we observe that, regardless of the considered technical efficiency measure, the SSVF and CSSVF efficiency scores are always greater or equal than the scores yielded by FDH and DEA. This is a feature that will be always observed in practice due to the satisfaction of the minimal extrapolation principle by FDH and DEA, something that does not hold in the case of the Support Vector Frontiers technique. Additionally, under any of the considered efficiency measures, the FDH technique is only able to detect one bank, Cosmos Bank, as being technically inefficient. This is a usual characteristic of FDH when the relationship between the sample size and the number of dimensions (inputs and outputs) is not good from a statistical point of view. This is the well-known curse of dimensionality problem, which is studied in depth in the literature (see, for example, Shen et al., 2016). In contrast, SSVF detects many more banks as being technically inefficient: 10 units in the case of the DDF, 11 in the case of the output-oriented radial model, 20 with the WA model and, finally, 29 in the case of the input-oriented radial model. This pattern is also evident when comparing the results yielded by DEA and CSSVF. As for the application of the notion of ε-insensitive technical efficiency, this is a more robust definition of efficiency than the traditional one, as we pointed out above. For each considered measure in the analysis, the two last columns indicate whether each unit is located inside the band associated with the margin ε under Support Vector Frontiers and its convexificated version, respectively.

[1] We are grateful to Juo et al. for sharing this data. We emphasize that we are using them simply for illustrating the multi-output Support Vector Frontiers technique together with different technical efficiency measures.

7. Conclusions and future work

Some researchers have tried to adapt Free Disposal Hull and Data Envelopment Analysis so that they work as inferential methods rather than as mere descriptive tools (see, for example, Banker and Maindiratta, 1992, Banker, 1993, Simar and Wilson, 1998, Kuosmanen and Johnson, 2010). However, despite the importance of the field of machine learning, there have only been a few attempts to adapt FDH and DEA to that area (see, for example, Esteve et al., 2020). Following this research line, Valero-Carreras et al. (2021) have very recently introduced a new technique called Support Vector Frontiers (SVF), based upon Support Vector Machines, which provides estimations for production functions, fulfilling well-known microeconomic regularity conditions. Nonetheless, in the contribution by Valero-Carreras et al. (2021), the Support Vector Frontiers technique was exclusively defined for working in the single-output production context, which represents an enormous restriction from a practical perspective. The reason for doing so is that, in supervised learning, moving from the single response framework to the multi-response context is not trivial and, indeed, there are several alternatives in the literature to address this problem.

In this paper, we resorted to the approach by Vazquez and Walter (2003) for adapting the SVF technique by Valero-Carreras et al. (2021) to the multi-output multi-input production context. In particular, we proved that the technology estimated by the multi-output SVF technique satisfies deterministicness, free disposability in inputs and outputs and, additionally, if desired, convexity. Moreover, we showed that FDH and DEA can be understood as particular cases of the more general multi-output Support Vector Frontiers technique. Also, a more robust notion of technical efficiency is defined, based upon the concept of margin in SVR. Moreover, due computational reasons, we also introduced a simplified version of the initial approach, whose validity was checked through simulation. Overall, our computational results showed that the new approach outperforms both FDH and DEA regarding Mean Squared Error and Bias. Finally, we showed how to implement some usual efficiency measures under the new approach and illustrate their performance through an empirical example.

As a future research line, we point out the possibility of considering a different transformation function, allowing estimator surfaces with more complex shapes than the piece-wise linear functions used. Another new avenue for further research could be the consideration of alternative approaches for moving from the single-output SVF to the multi-output framework, abandoning the model by Vazquez and Walter (2003). Increasing robustness of the model could be another research line for the future; resorting, for example, to Relaxed Support Vector Regression by Panagopoulos et al. (2019). Another avenue for further research could also be to check the accuracy of the new approach in comparison to FDH and DEA dealing with large-scale data samples. Finally, illustrating the new technique by using real data bases from very different fields could be useful to show the validity of the approach for estimating technical efficiency in practice.

CRediT authorship contribution statement

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

Acknowledgements

The authors thank the grant PID2019-105952GB-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.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.cor.2022.105765.

References

Aigner, D., Lovell, C.K., Schmidt, P., 1977. Formulation and estimation of stochastic frontier production function models. J. Econ. 6 (1), 21–37.

Aparicio, J., Borras, F., Pastor, J.T., Vidal, F., 2015. Measuring and decomposing firm′ s revenue and cost efficiency: the Russell measures revisited. Int. J. Prod. Econ. 165, 19–28.

Aparicio, J., Pastor, J.T., Vidal, F., 2016. The weighted additive distance function. Eur. J. Oper. Res. 254 (1), 338–346.

Aparicio, J., Cordero, J.M., Gonzalez, M., Lopez-Espin, J.J., 2018. Using non-radial DEA to assess school efficiency in a cross-country perspective: an empirical analysis of OECD countries. Omega 79, 9–20.

Aragon, Y., Daouia, A., Thomas-Agnan, C., 2005. Nonparametric frontier estimation: a conditional quantile-based approach. Econ. Theory 21 (2), 358–389.

Badiezadeh, T., Saen, R.F., Samavati, T., 2018. Assessing sustainability of supply chains by double frontier network DEA: a big data approach. Comput. Oper. Res. 98, 284–290.

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.

Chambers, R.G., Chung, Y., Färe, R., 1998. Profit, directional distance functions, and Nerlovian efficiency. J. Optim. Theory Appl. 98 (2), 351–364.

Charnes, A., Cooper, W.W., Rhodes, E., 1978. Measuring the efficiency of decision making units. Eur. J. Operat. Res. 2 (6), 429–444.

Cooper, W.W., Park, K.S., Pastor, J.T., 1999. RAM: a range adjusted measure of inefficiency for use with additive models and relations to other models and measures in DEA. J. Prod. Anal. 11 (1), 5–42.

Daraio, C., Simar, L., 2005. Introducing environmental variables in nonparametric frontier models: a probabilistic approach. J. Prod. Anal. 24 (1), 93–121.

Daraio, C., Simar, L., 2007. Advanced Robust and Nonparametric Methods in Efficiency Analysis: Methodology and Applications. Springer Science & Business Media.

Deprins, D., Simar, L., Tulkens, H., 1984. Measuring Labor Inefficiency in Post Offices. Amsterdam: North-Holland. In: Marchand, M., Pestieau, P., Tulkens, H. (Eds.), The Performance of Public Enterprises: Concepts and Measurements, pp. 243–267.

Dyson, R.G., Allen, R., Camanho, A.S., Podinovski, V.V., Sarrico, C.S., Shale, E.A., 2001. Pitfalls and protocols in DEA. Eur. J. Oper. Res. 132 (2), 245–259.

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.

Han, Z., Liu, Y., Zhao, J., Wang, W., 2012. Real time prediction for converter gas tank levels based on multi-output least square support vector regressor. Control Eng. Pract. 20 (12), 1400–1409.

Hong, H.K., Ha, S.H., Shin, C.K., Park, S.C., Kim, S.H., 1999. Evaluating the efficiency of system integration projects using data envelopment analysis (DEA) and machine learning. Expert Syst. Appl. 16 (3), 283–296.

Juo, J.C., Fu, T.T., Yu, M.M., Lin, Y.H., 2015. Profit-oriented productivity change. Omega 57, 176–187.

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. Eur. J. Oper. Res. 262 (2), 792–801.

Lovell, C.K., Pastor, J.T., 1995. Units invariant and translation invariant DEA models. Operat. Res. Lett. 18 (3), 147–151.

Meeusen, W., van Den Broeck, J., 1977. Efficiency estimation from Cobb-Douglas production functions with composed error. Int. Econ. Rev. 18 (2), 435.

Panagopoulos, O.P., Xanthopoulos, P., Razzaghi, T., Şeref, O., 2019. Relaxed support vector regression. Ann. Oper. Res. 276 (1), 191–210.

Pastor, J.T., Lovell, C.A., Aparicio, J., 2012. Families of linear efficiency programs based on Debreu's loss function. J. Prod. Anal. 38 (2), 109–120.

Pendharkar, P.C., 2021. Hybrid radial basis function DEA and its applications to regression, segmentation and cluster analysis problems. Mach. Learning Appl. 6, 100092.

Perelman, S., Santín, D., 2009. How to generate regularly behaved production data? A Monte Carlo experimentation on DEA scale efficiency measurement. Eur. J. Oper. Res. 199 (1), 303–310.

Sanchez-Fernandez, M., de-Prado-Cumplido, M., Arenas-Garcia, J., Perez-Cruz, F., 2004. SVM multiregression for nonlinear channel estimation in multiple-input multiple-output systems. IEEE Trans. Signal Process. 52 (8), 2298–2307.

Shen, W.F., Zhang, D.Q., Liu, W.B., Yang, G.L., 2016. Increasing discrimination of DEA evaluation by utilizing distances to anti-efficient frontiers. Comput. Oper. Res. 75, 163–173.

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. Statistics 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.

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.

Vapnik, V.N. (Ed.), 1995. The Nature of Statistical Learning Theory. Springer New York, New York, NY.

Vapnik, V., 1998. Statistical Learning Theory. Wiley, New York.

Vazquez, E., Walter, E., 2003. Multi-output suppport vector regression. IFAC Proc. Volumes 36 (16), 1783–1788.

Yang, X., & Dimitrov, S., 2017. Data envelopment analysis may obfuscate corporate financial data: using support vector machine and data envelopment analysis to predict corporate failure for nonmanufacturing firms. INFOR: Information Systems and Operational Research, 55(4), 295-311.

Zhu, J., 2019. DEA under big data: data enabled analytics and network data envelopment analysis. Ann. Oper. Res. 1–23.

Zhu, Q., Wu, J., Song, M., 2018. Efficiency evaluation based on data envelopment analysis in the big data context. Comput. Oper. Res. 98, 291–300.