



**Programa de Doctorado en Estadística, Optimización y
Matemática Aplicada**

**Nueva metodología, avances
computacionales y aplicaciones a
través de Support Vector Machines**

Daniel Valero Carreras

Director de la tesis:

Dr. Juan Aparicio Baeza

Codirector de la tesis:

Dr. Javier Alcaraz Soria

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

2023

INDICIOS DE CALIDAD DE LA TESIS DOCTORAL

El trabajo realizado por D. Daniel Valero Carreras, dirigido por Juan Aparicio Baeza y codirigido por Javier Alcaraz Soria, titulado **Nueva metodología, avances computacionales y aplicaciones a través de Support Vector Machines**, dentro del Programa de Doctorado en Estadística, Optimización y Matemática Aplicada se presenta bajo la modalidad de **tesis convencional con los siguientes artículos como indicios de calidad:**

- 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.
<https://doi.org/10.1016/j.omega.2021.102490>
- Valero-Carreras, D., Aparicio, J., & Guerrero, N. M. (2022). Multi-output support vector frontiers. *Computers & Operations Research*, 143, 105765.
<https://doi.org/10.1016/j.cor.2022.105765>
- Valero-Carreras, D., Alcaraz, J., & Landete, M. (2023). Comparing two SVM models through different metrics based on the confusion matrix. *Computers & Operations Research*, 152, 106131.
<https://doi.org/10.1016/j.cor.2022.106131>

También existe un capítulo de esta tesis que es objeto de publicación pero que todavía no ha sido publicado ya que se encuentra en proceso de revisión. El capítulo es 4.2 SUPPORT VECTOR FRONTIERS Y RANKING DE VARIABLES.



INFORME DEL DIRECTOR Y CODIRECTOR DE LA TESIS

JUAN APARICIO BAEZA, Catedrático de Universidad del Departamento de Estadística, Matemáticas e Informática de la Universidad Miguel Hernández de Elche.

JAVIER ALCARAZ SORIA, Profesor Titular Doctor del Departamento de Estadística, Matemáticas e Informática de la Universidad Miguel Hernández de Elche.

INFORMAN QUE

D. Daniel Valero Carreras ha realizado bajo nuestra supervisión el trabajo titulado **Nueva metodología, avances computacionales y aplicaciones a través de Support Vector Machines**, en el Programa de Doctorado de Estadística, Optimización y Matemática Aplicada conforme a los términos y condiciones definidos en su Plan de Investigación y de acuerdo al Código de Buenas Prácticas de la Universidad Miguel Hernández de Elche, cumpliendo los objetivos previstos de forma satisfactoria para su defensa pública como tesis doctoral.

Y para que conste, en cumplimiento de la legislación vigente y a los efectos oportunos, firmamos el presente certificado.

Fdo.: Juan Aparicio Baeza

Fdo.: Javier Alcaraz Soria

INFORME DE LA COMISIÓN ACADÉMICA DEL PROGRAMA DE DOCTORADO

Dr. Domingo Morales González, Coordinador del Programa de Doctorado en Estadística, Optimización y Matemática Aplicada de la Universidad Miguel Hernández de Elche

INFORMA QUE

El trabajo realizado por D. Daniel Valero Carreras, dirigido por el Dr. Juan Aparicio Baeza y codirigido por Javier Alcaraz Soria, titulado **Nueva metodología, avances computacionales y aplicaciones a través de Support Vector Machines**, ha sido autorizado por la Comisión Académica del Programa de Doctorado en Estadística, Optimización y Matemática Aplicada para su presentación y defensa en el correspondiente tribunal en el Universidad Miguel Hernández de Elche.

Y para que conste, en cumplimiento de la legislación vigente y a los efectos oportunos, firmo el presente certificado.

Fdo.: Domingo Morales González

Coordinador del Programa de Doctorado en Estadística, Optimización y Matemática
Aplicada

FINANCIACIÓN

Esta tesis ha sido posible gracias a la ayuda ACIF/2020/155 financiada por la Generalitat Valenciana.

Esta publicación es parte del proyecto de I+D+i PID2019-105952GB-I00, financiado por el Ministerio de Ciencia e Innovación/ Agencia Estatal de Investigación/ 10.13039/501100011033/.

ÍNDICE

LISTADO DE FIGURAS Y TABLAS	5
FIGURAS	5
TABLAS	5
RESUMEN	7
CAPÍTULO 1: INTRODUCCIÓN	11
CAPÍTULO 2: OBJETIVOS	21
CAPÍTULO 3: MATERIALES Y MÉTODOS	23
3.1. FREE DISPOSAL HULL	23
3.2. DATA ENVELOPMENT ANALYSIS	26
3.3. MEDIDAS DE EFICIENCIA	27
3.4. SUPPORT VECTOR MACHINES	29
3.4.1. SVM PARA CLASIFICACIÓN	29
3.4.2. SUPPORT VECTOR REGRESSION	32
3.5. SUPPORT VECTOR MACHINES EN EL CONTEXTO DE LA ORDENACIÓN DE CARACTERÍSTICAS	35
3.5.1. SUPPORT VECTOR MACHINE-PERMUTATION FEATURE IMPORTANCE	35
3.5.2. SUPPORT VECTOR MACHINE-BOOTSTRAP RESAMPLING	36
3.5.3. SVM-RFE PSEUDOSAMPLES	38
3.5.4. SVM-RFE DUAL OBJECTIVE VARIATION	38
3.6. SUPPORT VECTOR MACHINES EN EL CONTEXTO DE SELECCIÓN DE CARACTERÍSTICAS	39
3.6.1. TÉCNICAS PARA LA SOLUCIÓN DE SVM CON SELECCIÓN DE CARACTERÍSTICAS	41
3.6.2. MÉTRICAS DE RENDIMIENTO PARA LA COMPARACIÓN DE DIFERENTES CLASIFICADORES	48
3.6.3. MÉTRICAS DE COMPARACIÓN DE FRONTERAS	51

3.7. SUPPORT VECTOR MACHINE Y EL COSTE DE LA SELECCIÓN DE CARACTERÍSTICAS COMO OBJETIVO.....	52
CAPÍTULO 4: RESULTADOS	55
4.1. SUPPORT VECTOR FRONTIERS	55
4.1.1. EL CASO DE UNA ÚNICA VARIABLE RESPUESTA.....	55
4.1.1.1. LA ESTIMACIÓN DE SUPERFICIES ENVOLVENTES.....	57
4.1.1.2. CUMPLIENDO LOS AXIOMAS DE MONOTONICIDAD Y CONCAVIDAD DE LA FUNCIÓN DE PRODUCCIÓN.....	63
4.1.1.3. FDH Y DEA COMO UN CASO PARTICULAR DE SVF	71
4.1.2. EL CASO DE MÚLTIPLES VARIABLES RESPUESTA.....	77
4.1.2.1. VERSIÓN SIMPLIFICADA DE LAS TÉCNICAS SVF Y CSVF	84
4.1.3. EXPERIENCIA COMPUTACIONAL.....	86
4.1.3.1. SIMULACIONES MONO-OUTPUT	86
4.1.3.2. SIMULACIONES MULTI-OUTPUT.....	92
4.1.3.3. ILUSTRACIÓN EMPÍRICA	101
4.2. SUPPORT VECTOR FRONTIERS Y RANKING DE VARIABLES	104
4.2.1. SVF-PERMUTATION FEATURE IMPORTANCE.....	104
4.2.2. SVF-BOOTSTRAP RESAMPLING	105
4.2.3. SVF-RFE PSEUDOSAMPLES	107
4.2.4. SVM-RFE DUAL OBJECTIVE VARIATION.....	108
4.2.5. EXPERIENCIA COMPUTACIONAL	111
4.2.6. COSTE COMPUTACIONAL.....	115
4.3. SUPPORT VECTOR MACHINE Y SELECCIÓN DE CARACTERÍSTICAS ..	116
4.3.1. SVM Y SELECCIÓN DE CARACTERÍSTICAS TENIENDO EN CUENTA MÉTRICAS BASADAS EN LA MATRIZ DE CONFUSIÓN	116
4.3.1.1. EXPERIENCIA COMPUTACIONAL.....	118
4.3.2. SVM Y SELECCIÓN DE CARACTERÍSTICAS CON COSTES DE SELECCIÓN	129
4.3.2.1. EXPERIENCIA COMPUTACIONAL.....	132
CAPÍTULO 5: DISCUSIÓN	141
CAPÍTULO 6: CONCLUSIONES	147
REFERENCIAS	151

AGRADECIMIENTOS.....	161
----------------------	-----

LISTADO DE FIGURAS Y TABLAS

FIGURAS

Figura 1: Métodos de estimación de fronteras. Fuente: Aparicio (2007).....	14
Figura 2: Ejemplo de FDH estándar	25
Figura 3: Ejemplo de DEA estándar.....	26
Figura 4: Ejemplo de SVM de margen suave.....	32
Figura 5: Ejemplo de SVR estándar	34
Figura 6: Ejemplo de codificación propuesto por Alcaraz et al. (2022).....	47
Figura 7: Matriz de confusión	48
Figura 8: Ejemplo de AUC para dos clasificadores	50
Figura 9: Ejemplo de rejilla.....	59
Figura 10: Ejemplo de estimación de SVF (a) y CSVF (b).....	68
Figura 11: Ejemplo de DMUs ε -insensibles bajo SVF.....	70
Figura 12: Ilustraciones gráficas de SVF frente a FDH y CSVF frente a DEA en un conjunto de datos simulado	92
Figura 13: Tiempo de ejecución en segundos de SVF y SSVF.....	98
Figura 14: Hold-out estratificado con repeticiones	121
Figura 15: Ejemplo de aplicación de la metaheurística de dos objetivos frente a la de tres objetivos.....	132

TABLAS

Tabla 1: Número de restricciones en el modelo (25).....	85
Tabla 2. Descripción de los tres escenarios.....	86
Tabla 3: Los mejores hiperparámetros SVF	89
Tabla 4: Rendimiento relativo de los métodos de estimación basados en el MSE	90
Tabla 5: Rendimiento relativo de los métodos de estimación según los criterios de sesgo	91
Tabla 6: Cálculo del estadístico MSE y del sesgo para cada método.....	94
Tabla 7: Mejores hiperparámetros para el escenario sin ruido aleatorio.....	96
Tabla 8: Simulación multi-output: Resultados sin ruido aleatorio	99
Tabla 9: Simulación multi-output: Resultados con ruido aleatorio.....	100
Tabla 10: Medidas de eficiencia obtenidas a partir del ejemplo empírico.....	103
Tabla 11: Lista de experimentos realizados	112
Tabla 12: Tabla de resultados.....	113
Tabla 13: Tabla de tiempos de ejecución	116
Tabla 14: Características de las instancias usadas.....	119
Tabla 15: Resultados para el Modelo 1 en datasets con $n \leq 500$	125
Tabla 16: Resultados para el Modelo 2 en datasets con $n \leq 500$	126
Tabla 17: Resultados para el Modelo 1 en datasets con $n > 500$	127

Tabla 18: Resultados para el Modelo 2 en datasets con $n > 500$	128
Tabla 19: Tabla de frecuencias de éxito de los algoritmos con respecto a su contrincante	129
Tabla 20: Descripción de las instancias utilizadas	133
Tabla 21: Comparativa de resultados entre el algoritmo repetido tres veces, 20 minutos y el algoritmo ejecutado una hora: Medidas de cardinalidad, convergencia y distribución	134
Tabla 22: Comparativa de resultados entre AUGMECON y la NSGAI: Medidas de cardinalidad, convergencia y distribución	136
Tabla 23: Comparativa de las metaheurísticas al considerar dos o tres objetivos.....	137
Tabla 24: Comparativa de métricas relacionadas con la matriz de confusión en base al número de características seleccionadas.....	139

RESUMEN

Las Máquinas de Soporte Vectorial son en la actualidad una de las técnicas más estudiadas y utilizadas por los investigadores desde su invención en 1995 por Vladimir Vapnik. A su vez, la teoría sobre estimación de fronteras de producción y evaluación de la eficiencia técnica es un tema de interés para responsables políticos, así como gestores empresariales. Del mismo modo, la selección de características es un problema del aprendizaje automático que ha sido muy tenido en cuenta por los analistas de datos a la hora de reducir la complejidad de los problemas, reducir los costes y evitar el sobreajuste.

En esta tesis se introduce por primera vez una metodología, llamada *Support Vector Frontiers* (SVF), que une los mundos de Máquinas de Soporte Vectorial (SVM) y estimación de fronteras. Para ello, se ha adaptado el algoritmo para que cumpla con los principios de la teoría de estimación de fronteras. Este nuevo método permite competir con otras técnicas de la literatura tradicional como: Data Envelopment Analysis (DEA) y Free Disposal Hull (FDH) que tienden a sobreajustar los datos al cumplir con el principio de mínima extrapolación. Además, se ha desarrollado un nuevo concepto de eficiencia técnica, llamado eficiencia técnica ε -insensible, apoyado en el concepto de margen de las Máquinas de Soporte Vectorial de Regresión (SVR). Los resultados empíricos demostraron que el nuevo método SVF mejoraba tanto en escenarios mono-output como multi-output a los métodos DEA y FDH.

Asimismo, se ha utilizado la nueva metodología SVF para el desarrollo de técnicas de ranking de variables predictoras. Para ello, se ha hecho una revisión bibliográfica de la literatura de SVM y ranking de variables, seleccionando cuatro métodos para adaptarlos

al contexto de ranking de variables con técnicas de estimación de fronteras. Este trabajo ha dado lugar a cuatro técnicas conocidas, en inglés, como: SVF-Permutation Feature Importance, SVF Bootstrap Resampling, SVF Recursive Feature Elimination Pseudosamples y SVF Recursive Feature Elimination Dual Objective Variation.

Por otro lado, los avances de esta tesis no sólo se han producido en el ámbito de la estimación de fronteras y las SVR, sino que se ha ampliado el objeto de estudio y se ha abarcado también el problema de las SVM de clasificación. En este sentido se ha abordado el problema de las Máquinas de Soporte Vectorial de margen suave con selección de características. En los problemas de Máquinas de Soporte Vectorial los dos objetivos principales son maximizar la distancia de los hiperplanos separadores y a su vez minimizar la suma de las desviaciones de las observaciones mal clasificadas a su respectivo hiperplano. Si estos dos objetivos se combinan con la selección de p características, el problema resultante da lugar a un modelo no convexo difícil de resolver que genera una frontera Pareto. Esa frontera puede ser utilizada por los decisores para determinar cuál es la mejor opción dentro de un grupo de clasificadores. En esta tesis se ha desarrollado un nuevo modelo de SVM y selección de características que tiene en cuenta dos objetivos distintos a los planteados por Vapnik originalmente. Ya que las soluciones de la frontera Pareto pueden ser evaluadas a través de métricas basadas en la matriz de confusión, se ha propuesto un nuevo modelo en el que los dos principales objetivos a minimizar sean la suma de falsos positivos y la suma de falsos negativos. El resultado ha sido un nuevo modelo de programación lineal entera que requiere un gran esfuerzo computacional, por lo que las técnicas exactas no son capaces de resolver instancias de tamaño medio. Para resolver este modelo, se ha adaptado la metaheurística existente haciendo que el modelo mejore el rendimiento de los clasificadores en comparación con los métodos de la literatura existente.

Por último, a la hora de realizar un análisis, la selección de características con mínimo coste puede ser una tarea compleja para los decisores. Por ejemplo, si existe una prueba médica muy costosa con una alta precisión de diagnóstico, esta técnica podría ser sustituida por varias técnicas más baratas que otorguen una precisión similar. En base a esto, se ha desarrollado un nuevo modelo y metaheurística basados en la literatura

existente que abordan el problema de SVM y selección de características desde una perspectiva tri-objetivo de reducción de costes. Los dos primeros objetivos del nuevo modelo son los tradicionales de las SVM (maximizar distancia de los hiperplanos separadores de las clases y minimizar la suma de las distancias de las observaciones mal clasificadas a su hiperplano separador) mientras que se ha añadido un nuevo objetivo que consiste en minimizar el coste total, dado por la suma de los costes de las características seleccionadas. De este modo, se ha desarrollado un modelo que permite obtener soluciones parecidas o mejores a las obtenidas por el método bi-objetivo, pero reduciendo el coste al mínimo.

CAPÍTULO 1: INTRODUCCIÓN

En dos de los ámbitos más en boga en la sociedad actualmente, el Aprendizaje Automático y la Inteligencia Artificial, las Máquinas de Soporte Vectorial (Support Vector Machine en inglés) han surgido como una herramienta eficaz para la resolución de problemas de clasificación y de regresión. Desde que fueron introducidos por primera vez ([Vapnik \(1995, 1998\)](#)), este enfoque ha demostrado su rendimiento en una amplia gama de aplicaciones, como el reconocimiento de imágenes, el análisis de texto o el diagnóstico médico, entre otros. Todo ello debido a su gran capacidad para abordar problemas de aprendizaje tanto lineales como no lineales, así como su potencial para manejar datos con grandes dimensiones.

Esta técnica se basa en dos objetivos que tratan de complementarse uno al otro: minimizar el riesgo estructural (es decir, el error cometido por la máquina de aprendizaje sobre los datos que están fuera del conjunto de entrenamiento) y minimizar el riesgo empírico (el error que se comete al evaluar el conjunto de datos de entrenamiento). Cuando el error de generalización es alto se tiende a sobreajustar mucho los resultados a la muestra original lo que puede dar lugar a malas predicciones sobre datos que se encuentren fuera de la muestra de aprendizaje. Mientras que cuando el error empírico es alto la capacidad de generalización es muy alta pero los resultados sobre la predicción de los datos parecidos a la muestra de aprendizaje serán bajos. Debido a esta compensación entre riesgos, se genera un hiperplano separador en el que dependiendo del tipo de variable objetivo (categórica o numérica) se dividen las observaciones tratando de minimizar al máximo ambos errores.

En los últimos años, se han producido avances significativos en la comprensión teórica de la SVM, así como en la aplicación del enfoque a muchos contextos empíricos diferentes. La técnica ha alcanzado un punto en el que se considera claramente como una de las subáreas de investigación más fructíferas dentro del aprendizaje automático, una subárea que ha alcanzado un alto grado de madurez tanto en conocimientos teóricos como en utilidad práctica. Sin embargo, actualmente no existe en la literatura ninguna adaptación de SVM para aproximar funciones de producción en microeconomía, un campo que está relacionado con la medición de la eficiencia técnica de las empresas. Del mismo modo, han surgido nuevas aplicaciones para las Máquinas de Soporte Vectorial y se han mezclado con otras técnicas ya existentes como la selección de características aumentando así su potencialidad y capacidad de apoyo a los decisores.

Con respecto a la estimación de funciones de producción, la evaluación de la eficiencia técnica se ocupa de medir el rendimiento de las empresas, que convierten inputs en productos. La evaluación de la eficiencia en la producción ha sido y es un tema relevante para gestores y responsables políticos, así como un área que merece atención desde un punto de vista práctico y metodológico tanto en ingeniería como en economía (véase, por ejemplo, [Arnaboldi et al. \(2014\)](#), [Aparicio et al. \(2017\)](#) u [O'Donnell \(2018\)](#)). Desde una perspectiva metodológica, el objetivo principal de dicha evaluación es analizar la eficiencia técnica de un conjunto de entidades, generalmente conocidas como Unidades Tomadoras de Decisiones (DMU, por sus siglas en inglés), comparando su rendimiento con respecto al denominado conjunto de posibilidades de producción o tecnología, que es desconocido y debe estimarse a partir de una muestra de datos (una muestra de aprendizaje en la terminología del aprendizaje automático). En este contexto, la piedra angular del análisis de la eficiencia de las DMUs es la noción de función de producción. Una función de producción representa el producto máximo obtenible a partir de la combinación de recursos (inputs). Su estimación permite calcular el correspondiente valor de ineficiencia técnica como la desviación de cada DMU respecto a la frontera de la tecnología, caracterizada por la función de producción. De hecho, dado un nivel de consumo de inputs, la medida más habitual y natural de la eficiencia técnica de las empresas se define como la relación entre el producto realmente producido y el producto máximo producible, determinado por la función de producción.

En la literatura especializada, los investigadores en economía de la primera mitad del siglo XX especificaban paramétricamente la correspondiente función de producción, estimando dichos parámetros a través de regresión lineal y mínimos cuadrados ordinarios (OLS), determinando así una función de producción “promedio” y que, por tanto, no se ajustaba a la definición tradicional de función de producción en microeconomía.

Farrell (1957) fue el primer autor en mostrar cómo estimar una función de producción a través de la envolvente superior de un conjunto de observaciones. La metodología propuesta por Farrell puede ser categorizada dentro del área actual de técnicas no paramétricas, dado que no se requiere de la especificación funcional, a través de parámetros, de la función de producción. Esta misma línea de investigación fue más tarde continuada por Charnes et al. (1978) y Banker et al. (1984) dando lugar al Análisis Envolvente de Datos (Data Envelopment Analysis, DEA, en inglés). El DEA genera una superficie (frontera) que envuelve superiormente a la nube de puntos observada y que geoméricamente se corresponde con una función de producción lineal a trozos que define un conjunto de posibilidades de producción poliédrico. En particular, el poliedro determinado mediante DEA satisface una serie de axiomas básicos de la teoría de la producción, como la convexidad y la monotonidad. La relajación de la propiedad de convexidad en este tipo de técnicas conduce a una metodología alternativa que genera una función de producción escalonada denominada Free Disposal Hull (FDH) (Deprins et al. (1984)).

Por otra parte, del lado paramétrico la primera contribución relevante se debe a Aigner y Chu (1968), que mostraron cómo estimar los parámetros de una función de producción tipo Cobb-Douglas envolviendo a todas las observaciones de la muestra, minimizando la suma de los residuos al cuadrado, pero restringiendo el signo de los residuos a ser no negativos en lugar de libres, como sucede en OLS. Con esto se conseguía que los datos, necesariamente, tuvieran que posicionarse a un lado de la frontera estimada a través de su metodología.

A continuación, ilustramos gráficamente los principales métodos de estimación de fronteras citados hasta el momento. Asumimos el caso de producción más sencillo, en

donde las observaciones están formadas por pares de puntos (X, Y) , donde X representa al input e Y representa al output.

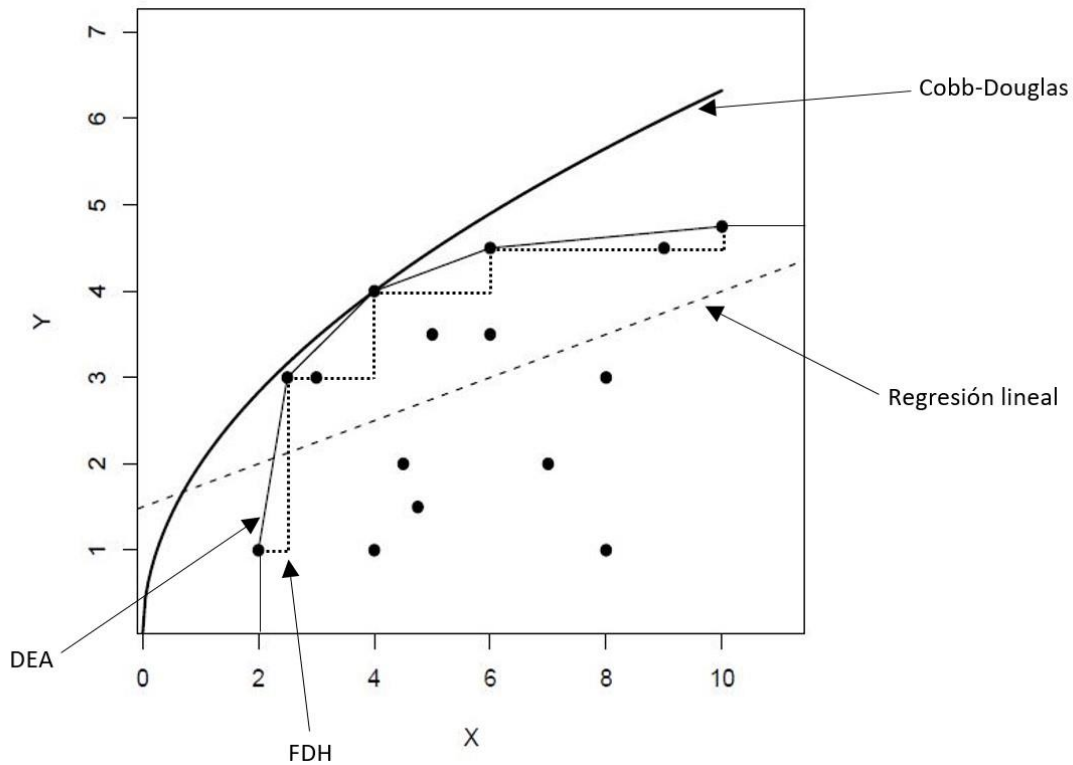


Figura 1: Métodos de estimación de fronteras. Fuente: Aparicio (2007)

Nótese que en la Figura 1 todas las observaciones se encuentran situadas por debajo de la frontera (la función de producción estimada) haciendo uso de las distintas técnicas, excepto en el caso del método de regresión lineal cuando se utiliza mínimos cuadrados para estimar los coeficientes de la recta de regresión; que pasa por el centroide de la nube de puntos. Es por ello que, en situaciones donde se tiene como objetivo la estimación de una envolvente superior de los datos, se evita el uso de este tipo de técnicas no adaptadas al contexto “frontera”. Adicionalmente, observamos que la función tipo Cobb-Douglas (curvilínea) no se ajusta tan bien a los datos como lo hacen las técnicas DEA y FDH. De hecho, estas dos últimas metodologías presentan un problema de sobreajuste (overfitting), explicando bien “a nivel frontera” el comportamiento de los datos de la muestra de aprendizaje y mal cualquier otro dato, en general, observado fuera de la muestra (out-of-sample).

Estas técnicas básicas de estimación de fronteras (comportamientos extremos de los datos) han evolucionado en los últimos años hacia metodologías algo más sofisticadas que consideran Procesos Generadores de Datos (PGD) que siguen ciertas distribuciones de probabilidad. Nos estamos refiriendo a los distintos avances que se han producido a través del Stochastic Frontier Analysis (SFA), DEA combinado con Bootstrapping (remuestreo), el Stochastic Non-parametric Envelopment of Data (StoNED) y otros [véase, por ejemplo, [Kuosmanen y Johnson \(2010\)](#) y [Olesen y Petersen \(2016\)](#)]. Asimismo, las aplicaciones de los métodos frontera no se han reducido al ámbito de la microeconomía sino que, por el contrario, estas metodologías han sido aplicadas con éxito a problemas de muy diferente naturaleza como, por ejemplo, para predecir la evolución futura de los supercomputadores ([Lim et al. \(2015\)](#)), para evaluar el rendimiento de servicios en la nube ([Jatoth et al. \(2017\)](#)), o para la determinación de rankings de universidades ([Liu et al. \(2019\)](#)), donde la determinación de la frontera de mejores prácticas resulta crucial. Sin embargo, al inicio de tesis muy pocas contribuciones abordaban el problema de la estimación de fronteras mezclándolo con el mundo del aprendizaje automático y muchos menos con Máquinas de Soporte Vectorial. En la actualidad, los trabajos de [Esteve et al. \(2020, 2022\)](#), [Guerrero et al. \(2022\)](#) y [Guillen et al. \(2023\)](#) sí que han conseguido adaptar técnicas de aprendizaje automático como son: Árboles de Regresión, Random Forest o Gradient Tree Boosting, a la temática de estimación de fronteras.

Con respecto a la aplicación de Máquinas de Soporte Vectorial con selección de características para problemas de clasificación, uno de los principales inconvenientes que presentan las SVM es el sobreajuste de los datos cuando se tiene un problema de grandes dimensiones. Una forma de evitarlo es incorporar dentro del propio modelo la selección de características, lo que confiere robustez al problema. Además, la selección de las características más representativas permite reducir el tamaño del problema haciendo que los costes computacionales y económicos sean menores. Algunos de los autores que han estudiado el tema de SVM con selección de características son: [Maldonado et al. \(2014\)](#) introducen dos modelos enteros mixtos de SVM con selección de características; [Aytug \(2015\)](#) resuelve la SVM con selección de características con la ayuda de la descomposición de Benders; [Gaudioso et al. \(2017\)](#) proponen un enfoque de relajación

lagrangiana para el problema SVM con selección de características; [Labbe et al. \(2019\)](#) mejoran el modelo de Maldonado o [Benitez-Peña et al. \(2019\)](#) proponen un problema lineal entero más un problema convexo cuadrático.

Dado que el problema SVM tiene dos objetivos diferentes (riesgo estructural o generalización y riesgo empírico), la salida del problema no es una solución óptima, sino el conjunto de soluciones no dominadas, que forman una frontera Pareto. Cada una de estas soluciones representa un clasificador diferente y, por lo tanto, disponer de la frontera Pareto proporciona al responsable de la toma de decisiones una amplia variedad de soluciones alternativas entre las que elegir el mejor clasificador.

La mayoría de los métodos desarrollados para resolver el problema combinan los dos objetivos del problema en una única función objetivo, asignando un peso diferente a cada uno de los objetivos. Sin embargo, cuando el número de características seleccionadas se fija en p la función objetivo para el modelo SVM no es convexa y la frontera Pareto no puede obtenerse simplemente variando el peso del compromiso entre objetivos. De este modo, diferentes valores de peso podrían conducir a la misma solución y debido a las características del modelo, este procedimiento podría no encontrar algunas de las soluciones Pareto. Además, la introducción de nuevos parámetros en el modelo hace que la precisión del método dependa en gran medida de ellos. Se han desarrollado varias metaheurísticas para especificar los parámetros del modelo y seleccionar el subconjunto de características. De esta forma, una vez determinados los parámetros del modelo y la selección de características por la metaheurística, se resuelve el modelo SVM, combinando ambos objetivos, y se obtiene un clasificador.

En las últimas décadas, las técnicas metaheurísticas se han convertido en una herramienta eficaz para resolver problemas de optimización muy diversos. El primer conjunto de metaheurísticas clásicas se introdujo entre principios de los 70 y los 90. Algunos de ellos son, por ejemplo: los Algoritmos Genéticos (GA), el algoritmo de Recocido Simulado (SA), la Búsqueda Tabú (TS), la Optimización por Colonia de Hormigas (ACO) o la Optimización por Enjambre de Partículas (PSO). Posteriormente, se propusieron varias mejoras de éstos y nuevas ideas para resolver problemas de optimización difíciles. En

[Ezugwu et al. \(2021\)](#) se puede encontrar una buena revisión y clasificación de las metaheurísticas.

Dentro del complejo problema de determinar los parámetros de un modelo y la selección de características, algunas de las metaheurísticas que han sido recientemente utilizadas son: [Huang and Wang \(2006\)](#) y posteriormente [Zhao et al. \(2011\)](#) que proponen algoritmos genéticos que también son comparados con otros; [García-Pedrajas et al. \(2014\)](#) proponen un algoritmo mimético y [Carrizosa et al. \(2014\)](#) una heurística anidada; [Raman et al. \(2017\)](#) presentan una técnica de detección de intrusos adaptativa y robusta; [Aladeemy et al. \(2017\)](#) proponen una variación del algoritmo Cohort Intelligence; [Bouraoui et al. \(2018\)](#) proponen un enfoque multiobjetivo para optimizar simultáneamente los parámetros de SVM y el subconjunto de características utilizando diferentes funciones de kernel; [Faris et al. \(2018\)](#) presentan un enfoque de optimizador multiverso basado en una arquitectura de sistema robusto; [Candelieri et al. \(2019\)](#) proponen una optimización global paralela para ajustar los parámetros de un sistema de previsión de la demanda de agua; [Dudzik et al. \(2021\)](#) proponen una técnica evolutiva que clasifica eficientemente conjuntos de datos difíciles, incluyendo casos muy grandes y extremadamente desequilibrados; [Xue et al. \(2021\)](#) proponen una novedosa metaheurística multi-objetivo para un problema de clasificación de 3 objetivos. Ellos obtienen la frontera Pareto cuando los objetivos son maximizar el margen entre los dos hiperplanos paralelos asociados, minimizar el número de características seleccionadas y minimizar la cantidad de datos perdidos.

La mayoría de los métodos anteriores seleccionan características óptimas y optimizan los parámetros de la SVM simultáneamente con el objetivo de reducir el número de características manteniendo la precisión de la predicción. Sin embargo, ninguno de los métodos presentados anteriormente proporciona la frontera Pareto óptima del problema SVM de margen suave con selección de características o, al menos, una aproximación al mismo. [Alcaraz et al. \(2022\)](#) diseñan una metaheurística eficiente para obtener la frontera Pareto del modelo SVM con selección de características y comparan las fronteras dadas por la metaheurística y por un método exacto. También demuestran que los métodos exactos fallan en la obtención de la frontera en problemas de tamaño medio o

grande. Hasta la fecha, este método es el único desarrollado para construir la frontera Pareto del modelo SVM de margen suave con selección de características.

Una vez que tenemos todas las soluciones alternativas al modelo SVM con selección de características, formadas por la frontera, si el objetivo es seleccionar el mejor clasificador, la comparación se puede realizar a través de métricas basadas en la matriz de confusión, como el área bajo la curva ROC (Receiver Operating Characteristic), el coeficiente Kappa de Cohen o el F-Score. Una curva ROC es una técnica para organizar, visualizar y seleccionar clasificadores en función de su rendimiento (Fawcett (2006)). Hoy en día, el análisis ROC es una técnica extendida y se emplea en una amplia variedad de campos, como la ingeniería (Ierimonti et al. (2021)), la economía y las finanzas (Yang et al. (2022)), las matemáticas (Vijayan et al. (2021)), ciencias empresariales y de la decisión (Florian et al. (2021)), informática (Ponmalar and Dhanakoti (2022)), medicina (Halder et al. (2022)), agricultura (Loh et al. (2022)) o física (Kwiatkowski and Sotor (2022)), por citar sólo algunos. El coeficiente Kappa de Cohen y el F-Score son otras dos métricas ampliamente utilizadas basadas en la matriz de confusión.

Los costes asociados a una investigación son algunos de los problemas con los que se puede encontrar el analista de datos a la hora de realizar un estudio. Por ejemplo, a la hora de realizar un diagnóstico médico, las pruebas necesarias tienen un coste económico y temporal. En este contexto la selección de características y el machine learning se convierten en un potente aliado para el tomador de decisiones. Existen muchas técnicas para la selección de características que pueden ser divididas en diferentes grupos: métodos de filtro, métodos de envoltura o métodos embebidos (véase Ventakesh et al. (2019)). Pero la mayoría de los métodos no tienen en cuenta los costes asociados a la selección de características, asignándoles el mismo peso a cada una. En la actualidad, sí que hay autores que tienen en cuenta el coste de la selección de características. Por ejemplo, por mencionar algunos autores: Min et al. (2014), Zhang et al. (2015), Zhou et al. (2015). No obstante, no existe en la actualidad un método basado en SVM y selección de características que proporcione una frontera Pareto de soluciones y a la vez tenga en cuenta el coste de la selección de características.

La presente tesis se ha enfocado en establecer nueva metodología y nuevas aplicaciones a los usos ya convencionales de las Máquinas de Soporte Vectorial. En el [Capítulo 2](#) se presentan los objetivos principales de la tesis. En el [Capítulo 3](#) se introducen las técnicas FDH, DEA y se hace una introducción a distintas medidas de eficiencia. Asimismo, se hace una descripción del problema de SVM tanto en el contexto de regresión (Support Vector Regression, SVR) como en el de clasificación. También se hace un repaso de técnicas de ranking de variables con SVM. Finalmente, en este capítulo se habla sobre el problema de SVM con selección de características, recopilando metaheurísticas que se utilizan para su resolución y analizando las diferentes métricas que se utilizan para evaluar clasificadores y fronteras. Dentro del [Capítulo 4](#), vamos a presentar los diferentes avances realizados en la nueva metodología desarrollada a partir de SVM. En la [Sección 4.1](#) se muestra la nueva metodología, bautizada con el nombre de Support Vector Frontiers, basada en la unión del mundo de la estimación de fronteras con el de SVM. Además, se añaden ejemplos ilustrativos de la nueva técnica. En la [Sección 4.2](#) se presentan diferentes métodos de ranking de variables utilizando la técnica de la [Sección 4.1](#). En la [Sección 4.3](#) se propone un nuevo método para la resolución del problema de SVM con selección de características basado en métricas de la matriz de confusión y un nuevo modelo de SVM tri-objetivo con selección de características que tiene en cuenta los objetivos tradicionales más el coste de la selección de características. Por último, en el [Capítulo 5](#) se recogen los avances obtenidos durante la investigación y en el [Capítulo 6](#) las conclusiones y futuras líneas de investigación.

CAPÍTULO 2: OBJETIVOS

Como se ha mencionado en el capítulo introductorio, las Máquinas de Soporte Vectorial se han convertido en una herramienta ampliamente utilizada para la solución de problemas de clasificación o de regresión y han alcanzado la madurez necesaria para convertirse en una de las áreas de investigación más productivas del aprendizaje automático. En esta tesis se han establecido una serie de objetivos para dotar a SVM de nueva metodología y aplicaciones:

El primer objetivo de esta tesis (Objetivo 1) es crear una nueva metodología que una el mundo de la estimación de fronteras con el mundo de las Máquinas de Soporte Vectorial. De este modo, podríamos utilizar este enfoque no paramétrico para generar fronteras que se parezcan a FDH pero no sufran el inconveniente del sobreajuste. Para ello, será necesario definir una transformación de los datos que permita convertir la predicción lineal de las SVR en una frontera escalonada como la de FDH. Una vez obtenida la estructura de la frontera, se podrá obtener la versión convexificada de la frontera escalonada de la SVR modificada, al igual que el FDH hace con el DEA.

El segundo objetivo (Objetivo 2) es dotar a la metodología resultante del Objetivo 1 de capacidad para jerarquizar la importancia de los inputs, y dado que ésta se basa en Support Vector Machines, centramos nuestra atención en los principales enfoques de la literatura sobre aprendizaje automático que combinan SVM y la determinación de la relevancia de las variables.

Para no centrarnos exclusivamente en una de las aplicaciones de las SVM, la regresión, hemos planteado una nueva meta (Objetivo 3) cuya finalidad es el desarrollo de un problema multiobjetivo basado en SVM con selección de características en el que el

primer objetivo sea minimizar el número de falsos positivos y el segundo minimizar el número de falsos negativos. Para ello, se comparará el nuevo método desarrollado con los resultados del modelo propuesto por [Alcaraz et al. \(2022\)](#).

Finalmente, en los problemas reales, los recursos son limitados, ya sean tecnológicos, temporales o económicos. En estos casos, la utilización de una variable tiene asociados diferentes costes (véase, por ejemplo, el coste económico y temporal de realizar una prueba de detección de cáncer). La selección de características puede reducir el coste de un proceso de analítica de datos analizando si esa variable es importante o si se pueden utilizar otras. El último objetivo (Objetivo 4) que se propone en esta tesis es el desarrollo de un modelo tri-objetivo basado en el modelo de [Alcaraz et al. \(2022\)](#) en el que se incorpora la minimización del coste de las características seleccionadas. Esto puede ayudar a los decisores a obtener soluciones bastante precisas reduciendo los costes del proceso. Al igual que en el modelo del Objetivo 3, se modificará la metaheurística propuesta para obtener las soluciones tri-objetivo de este problema.

CAPÍTULO 3: MATERIALES Y MÉTODOS

En este capítulo se describen las diferentes técnicas y métodos utilizados como punto de partida para la investigación: Free Disposal Hull (FDH), Data Envelopment Analysis (DEA), Support Vector Machines (SVM), Support Vector Machine en el contexto de ranking de características, Support Vector Machine y selección de características y metaheurísticas para la selección de características con SVM de dos y tres objetivos.

3.1. FREE DISPOSAL HULL

Free Disposal Hull (FDH) es un enfoque no paramétrico propuesto por [Deprins et al. \(1984\)](#) para la estimación de fronteras de producción y medición de la eficiencia.

Dado un conjunto de n unidades tomadoras de decisión (en inglés, Decision Making Units, DMUs). La DMU_i consume $\mathbf{x}_i = (x_i^{(1)}, \dots, x_i^{(m)}) \in \mathcal{R}_+^m$ cantidades de inputs para la producción de $\mathbf{y}_i = (y_i^{(1)}, \dots, y_i^{(s)}) \in \mathcal{R}_+^s$ cantidades de productos. La eficiencia relativa de cada DMU de la muestra se evalúa con referencia al denominado conjunto de posibilidades de producción o tecnología (T), que es el conjunto de combinaciones técnicamente viables de (\mathbf{x}, \mathbf{y}) . Se define en términos generales como:

$$T = \{(\mathbf{x}, \mathbf{y}) \in \mathcal{R}_+^{m+s} : \mathbf{x} \text{ puede producir } \mathbf{y}\} \quad (1)$$

Sobre este conjunto se realizan diversas suposiciones, como la libre disponibilidad de inputs y outputs, lo que significa que si $(\mathbf{x}, y) \in T$, entonces $(\mathbf{x}', y') \in T$, siempre que $\mathbf{x}' \geq \mathbf{x}$ e $y' \leq y$. A menudo también se asume la convexidad de T (véase, por ejemplo, [Färe y Primont, \(1995\)](#)).

En el caso de que $s = 1$, este contexto se limita a la noción clave de función de producción. En consecuencia, m variables de entrada se utilizan para producir una única salida. De este modo, la tecnología se define como

$$T = \{(\mathbf{x}, y) \in \mathbb{R}_+^{m+1} : y \leq f(\mathbf{x})\} \quad (2)$$

En este escenario, la propiedad de monotonidad se traduce en que se supone que la función de producción f es monótona no decreciente: si $\mathbf{x} \leq \mathbf{x}'$, por lo tanto $f(\mathbf{x}) \leq f(\mathbf{x}')$.

Por lo que respecta a la medición de la eficiencia técnica, interesa cierta parte de la frontera de T . En concreto, nos referimos a la frontera eficiente de T , definida como $\partial(T) := \{(\mathbf{x}, y) \in T : y = f(\mathbf{x})\}$. De este modo, la ineficiencia técnica se define como la distancia de un punto interior a esta frontera. Si $(\mathbf{x}', y') \in T$, entonces su eficiencia técnica viene determinada por la ratio $y'/f(\mathbf{x}')$.

FDH se basa únicamente en el supuesto de libre disponibilidad (monotonidad) y en el principio de extrapolación mínima. Esto permite a este enfoque ofrecer un estimador más flexible que otros de la literatura como Data Envelopment Analysis (DEA), ya que no exige el cumplimiento de otros postulados como el de convexidad del conjunto T (véase [Aragon et al. \(2005\)](#)).

Los modelos no paramétricos, como el FDH, son especialmente atractivos, ya que no dependen de hipótesis restrictivas sobre el Proceso Generador de Datos, una característica compartida con las técnicas habituales de aprendizaje automático, que son enfoques claramente basados en datos. En particular, [Deprins et al. \(1984\)](#) propusieron el Free Disposal Hull del conjunto de observaciones (DMUs) para estimar T , de la siguiente forma:

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

En el caso de haber un único output, la función de producción se estimaría mediante $f_{FDH}(\mathbf{x}) = \max_{i: \mathbf{x} \geq \mathbf{x}_i} \{y_i\}$. A continuación, se muestra un ejemplo gráfico (Figura 2) del estimador FDH de una función de producción, mostrando su típica forma escalonada no decreciente.

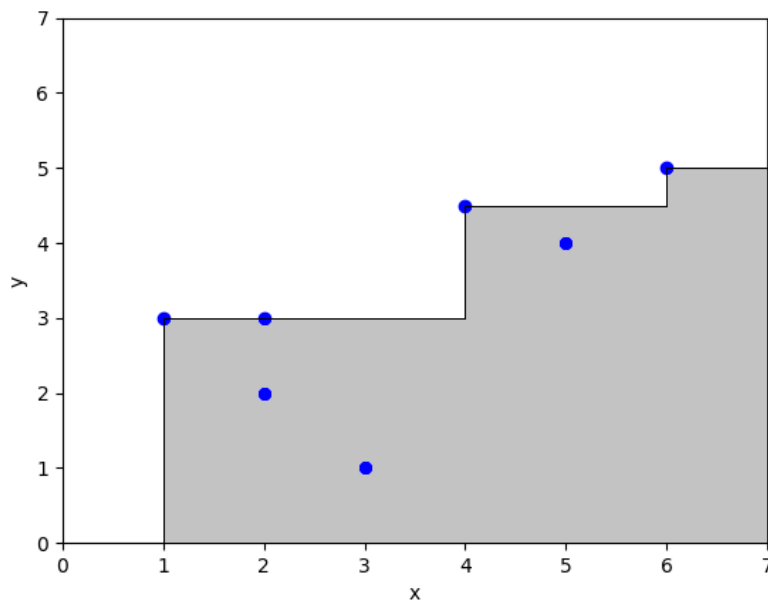


Figura 2: Ejemplo de FDH estándar

La técnica FDH es muy utilizada porque se basa en muy pocos supuestos, pero, por construcción, sufre sobreajuste debido al principio de extrapolación mínima. Este principio obliga a que la función escalonada no decreciente se aproxime lo más posible a la nube de datos, mientras que (por arriba) envuelve todas las observaciones. Obsérvese, en la Figura 2, que la frontera estimada por FDH se ajusta perfectamente a la muestra de datos, además de satisfacer la propiedad de monotonía.

3.2. DATA ENVELOPMENT ANALYSIS

Data Envelopment Analysis (DEA) también es una técnica no paramétrica para la estimación de fronteras de producción propuesta por [Banker et al. \(1984\)](#). A diferencia de FDH, el Análisis Envoltente de Datos asume, además, la convexidad. Esto significa que si (\mathbf{x}, y) y (\mathbf{x}', y') pertenecen a T , entonces $\lambda(\mathbf{x}, y) + (1 - \lambda)(\mathbf{x}', y') \in T$, para todos $\lambda \in [0, 1]$. De este modo, [Banker et al. \(1984\)](#) propusieron el estimador DEA del conjunto de posibilidades de producción T de la siguiente manera:

$$\hat{T}_{DEA} = \left\{ (\mathbf{x}, y) \in R_+^{m+s} : y^{(r)} \leq \sum_{i=1}^n \lambda_i y_i^{(r)}, \forall r, x^{(j)} \geq \sum_{i=1}^n \lambda_i x_i^{(j)}, \forall j = 1, \sum_{i=1}^n \lambda_i = 1, \lambda_i \geq 0, \forall i \right\} \quad (4)$$

A continuación, se muestra un ejemplo gráfico del estimador DEA de una función de producción (un único output), mostrando su típica forma lineal a trozos. La convexidad del conjunto de posibilidades de producción (el área sombreada) implica, en este caso, la concavidad de la función de producción. Obsérvese también que la convexificación del estimador FDH da lugar al estimador DEA ([Daraio y Simar \(2005\)](#)).

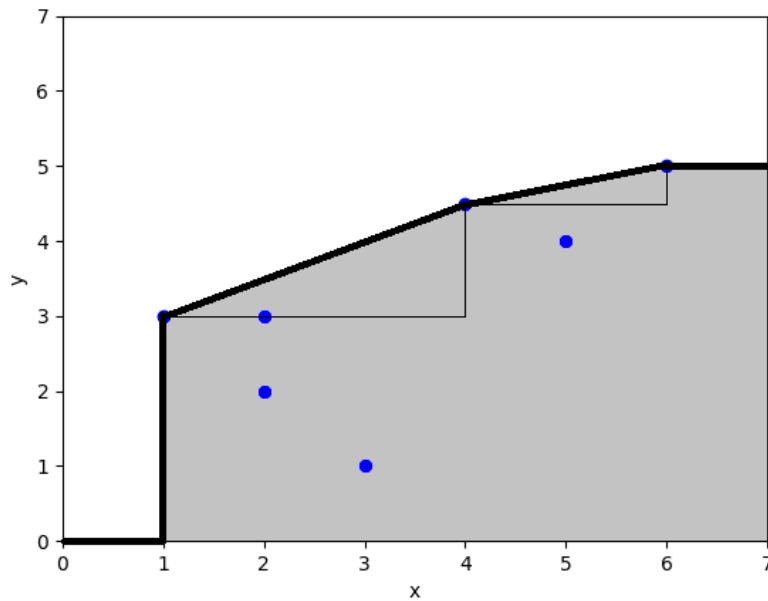


Figura 3: Ejemplo de DEA estándar

Como puede observarse en la figura, la frontera DEA sufre el mismo problema de sobreajuste que FDH. Una vez más, el principio de extrapolación mínima obliga a determinar una descripción “perfecta” de la muestra observada a nivel de frontera. Esto se consigue calculando el conjunto convexo más pequeño que contiene la nube de datos y satisface la libre disponibilidad (monotonicidad).

3.3. MEDIDAS DE EFICIENCIA

Como se ha mencionado anteriormente, a través de las técnicas anteriores es posible medir la eficiencia técnica de las DMUs, calculando el desplazamiento que hay que realizar para situar esas unidades sobre la frontera de buenas prácticas. En la literatura, se han desarrollado diferentes medidas que permiten cuantificar la eficiencia. Algunas de esas medidas aparecen detalladas a continuación bajo el enfoque de la técnica FDH.

El modelo radial orientado a los outputs (Banker et al. (1984)): es un modelo de programación que, fijando los inputs, determina la proporción en la que deben incrementarse todos los outputs a la vez para que la unidad evaluada se sitúe sobre la frontera eficiente. El modelo de optimización que permite la implementación de esta medida es la siguiente:

$$\begin{aligned}
 & \max \quad \phi && (.0) \\
 & s.t. && \\
 & \sum_{k=1}^n \lambda_k x_k^{(j)} \leq x_i^{(j)}, \quad j=1, \dots, m && (.1) \\
 & \sum_{k=1}^n \lambda_k y_k^{(r)} \geq \phi y_i^{(r)}, \quad r=1, \dots, s && (.2) \\
 & \sum_{k=1}^n \lambda_k = 1, && (.3) \\
 & \lambda_k \in \{0,1\}, \quad k=1, \dots, n && (.4)
 \end{aligned}
 \tag{5}$$

El modelo radial orientado a los inputs (Banker et al. 1984): determina la proporción en la que deben reducirse todos los inputs, manteniendo constante los outputs para que la unidad evaluada sea eficiente.

$$\begin{aligned}
\min \quad & \theta & (.0) \\
s.t. \quad & & (.1) \\
& \sum_{k=1}^n \lambda_k x_k^{(j)} \leq \theta x_i^{(j)}, \quad j=1, \dots, m & (.2) \\
& \sum_{k=1}^n \lambda_k y_k^{(r)} \geq y_i^{(r)}, \quad r=1, \dots, s & (.3) \\
& \sum_{k=1}^n \lambda_k = 1, & (.4) \\
& \lambda_k \in \{0,1\}, \quad k=1, \dots, n & (.5)
\end{aligned} \tag{6}$$

La función de distancia direccional (DDF) (Chambers et al. (1998)) proyecta la DMU (x_i, y_i) sobre la frontera de la tecnología siguiendo la dirección $g = (-g^-, g^+)$, con $g^- \in R_+^m$ y $g^+ \in R_+^s$. Su implementación es como sigue:

$$\begin{aligned}
\max \quad & \beta & (.0) \\
s.t. \quad & & (.1) \\
& \sum_{k=1}^n \lambda_k x_k^{(j)} \leq x_i^{(j)} - \beta g^{-j}, \quad j=1, \dots, m & (.2) \\
& \sum_{k=1}^n \lambda_k y_k^{(r)} \geq y_i^{(r)} + \beta g^{+r}, \quad r=1, \dots, s & (.3) \\
& \sum_{k=1}^n \lambda_k = 1 & (.4) \\
& \lambda_k \in \{0,1\}, \quad k=1, \dots, n & (.5)
\end{aligned} \tag{7}$$

El modelo aditivo ponderado (WA) (Lovell and Pastor (1995)) maximiza una agregación ponderada de las ineficiencias específicas de los inputs y de los outputs (slacks):

$$\begin{aligned}
\max \quad & \sum_{j=1}^m \rho^{-(j)} s^{-(j)} + \sum_{r=1}^s \rho^{+(r)} s^{+(r)} & (.0) \\
s.t. \quad & \sum_{k=1}^n \lambda_k x_k^{(j)} \leq x_i^{(j)} - s^{-(j)}, \quad j=1, \dots, m & (.1) \\
& \sum_{k=1}^n \lambda_k y_k^{(r)} \geq y_i^{(r)} + s^{+(r)}, \quad r=1, \dots, s & (.2) \\
& \sum_{k=1}^n \lambda_k = 1, & (.3) \\
& \lambda_k \in \{0,1\}, \quad k=1, \dots, n & (.4) \\
& s^{-(j)}, s^{+(r)} \geq 0 & (.5) \\
& & j=1, \dots, m, \\
& & r=1, \dots, s
\end{aligned} \tag{8}$$

Si, además, se asume la convexidad, entonces las restricciones $\lambda_k \in \{0,1\}$, $\forall k = 1, \dots, n$, deben sustituirse por $\lambda_k \geq 0$, $\forall k = 1, \dots, n$, en los programas de optimización anteriores, obteniéndose así la determinación de las medidas de eficiencia bajo el enfoque del DEA.

3.4. SUPPORT VECTOR MACHINES

Las Máquinas de Soporte Vectorial (Support Vector Machines, SVM, en inglés) son una técnica de aprendizaje automático desarrollada por Corinna Cortes y Vladimir Vapnik (Cortes y Vapnik (1995), Vapnik (1995, 1998)).

Desde un punto de vista teórico, SVM es un procedimiento de aprendizaje automático constructivista basado en la teoría de aprendizaje estadístico y en el principio de minimización del riesgo estructural. Su objetivo es minimizar el error de generalización en lugar de reducir exclusivamente el error empírico, como el error cuadrático medio sobre el conjunto de datos (Vapnik 1995,1998). Esto otorga una buena capacidad de generalización ya que la SVM es capaz de obtener buenos resultados sobre datos que son diferentes a los que se encuentran en el conjunto de entrenamiento. Debido a esto, SVM es adecuado para tratar con un número limitado de muestras, independientemente del número de dimensiones del problema y ha sido utilizado en diferentes tareas del aprendizaje automático como: la minería de datos, el reconocimiento de patrones, aproximaciones de funciones, regresión, etc.

Dependiendo de la naturaleza de las variables objetivo, SVM actúa como un modelo no paramétrico de clasificación cuando la variable respuesta es categórica y como un modelo de regresión cuando la variable respuesta es numérica. De este modo, da lugar a dos tipos de algoritmos: Máquinas de Soporte Vectorial de Regresión (Support Vector Regression, SVR) y Máquinas de Soporte Vectorial de Clasificación también conocidas simplemente como Máquinas de Soporte Vectorial.

3.4.1. SVM PARA CLASIFICACIÓN

Support Vector Machines para clasificación es un algoritmo de aprendizaje supervisado que se utiliza en problemas en los que la variable respuesta es cualitativa. Si sólo se consideran dos clases de elementos, busca el hiperplano de separación óptimo, que es el hiperplano intermedio de dos hiperplanos paralelos, uno situado por encima de los

elementos de la primera clase y el otro por debajo de los elementos de la segunda clase. Los hiperplanos de separación buenos son aquellos cuyo margen entre los dos hiperplanos paralelos asociados es grande. Si los datos son linealmente separables, se dice que este margen es duro. Si un límite lineal no es factible o se permiten clasificaciones erróneas con la esperanza de lograr una mejor generalidad, este margen se dice que es suave (soft margin). Para el problema SVM de margen suave, los hiperplanos de separación buenos son aquellos que tienen un gran margen entre los dos hiperplanos paralelos y una pequeña distancia entre los vectores mal clasificados y el hiperplano correspondiente. Estos dos objetivos se conocen respectivamente como riesgo estructural y riesgo empírico.

Dado un conjunto de entrenamiento Ω de clasificación, los elementos (normalmente denominados vectores) se dividen en dos clases y se representan mediante un par $(x_i, y_i) \in \mathbb{R}^m \times \{-1, 1\}$, donde m es el número de características observadas para cada vector, x_i contiene los valores de las características del vector i e y_i indica a cuál de las dos clases de Ω pertenece el vector i . Si observamos la Figura 4, el objetivo de SVM consiste en determinar el hiperplano clasificador $f(x) = w^T \cdot x + b$ (línea continua) que separe los puntos de las clases a estudiar. Por ejemplo: los puntos amarillos pueden ser de la clase 1 y los puntos lilas de la clase -1. Este hiperplano tiene dos hiperplanos paralelos (líneas discontinuas) que soportan algún vector de las dos clases (vectores soporte). La distancia entre los dos hiperplanos es lo que se conoce como margen o riesgo estructural. Por otro lado, los puntos que se encuentren en el lado correcto de su hiperplano no contribuirán al error del modelo mientras que los que sí que estén en el lado incorrecto sí que lo harán. Esto es lo que se conoce como riesgo empírico.

El modelo (9) desarrollado por [Cortes y Vapnik \(1995\)](#), minimiza la función objetivo mediante un compromiso entre los dos objetivos mencionados (riesgo estructural o de generalización y riesgo empírico).

$$\begin{aligned}
\min \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i & (0) \\
\text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \xi_i \quad i = 1, \dots, n, & (1) \\
& \xi_i \geq 0 \quad i = 1, \dots, n, & (2) \\
& w_j \in \mathbb{R} \quad j = 1, \dots, m, & (3) \\
& b \in \mathbb{R} & (4)
\end{aligned} \tag{9}$$

Las variables w_j y b representan los coeficientes de los dos hiperplanos paralelos $w^T x + b = 1$ y $w^T x + b = -1$. El primer término de la función objetivo $\frac{1}{2} \|w\|^2$ representa el riesgo estructural, ya que $\|w\|$ es el doble de la inversa de la distancia entre estos dos hiperplanos (este elemento procede de la cota superior del error de generalización que se pretende minimizar). El segundo término de la función objetivo $\sum_{i=1}^m \xi_i$ representa el empírico, es decir, la suma de la desviación de los objetos mal clasificados. Estos dos errores se equilibran mediante un hiperparámetro C que debe ser establecido por los decisores (generalmente este hiperparámetro es determinado mediante un procedimiento de validación). Las restricciones del modelo garantizan que, o bien los vectores i de la clase representada por $y_i = 1$ satisfacen $(w^T x_i + b) \geq 1$ y los vectores de la clase $y_i = -1$ satisfacen $(w^T x_i + b) \leq -1$, o bien la variable ξ_i toma un valor positivo.

El trabajo de [Burges \(1998\)](#) constituye un tutorial práctico sobre SVM y el libro de [Vapnik \(1995\)](#) es una completa introducción a la relevancia de la teoría del aprendizaje para el diseño de la clasificación supervisada.

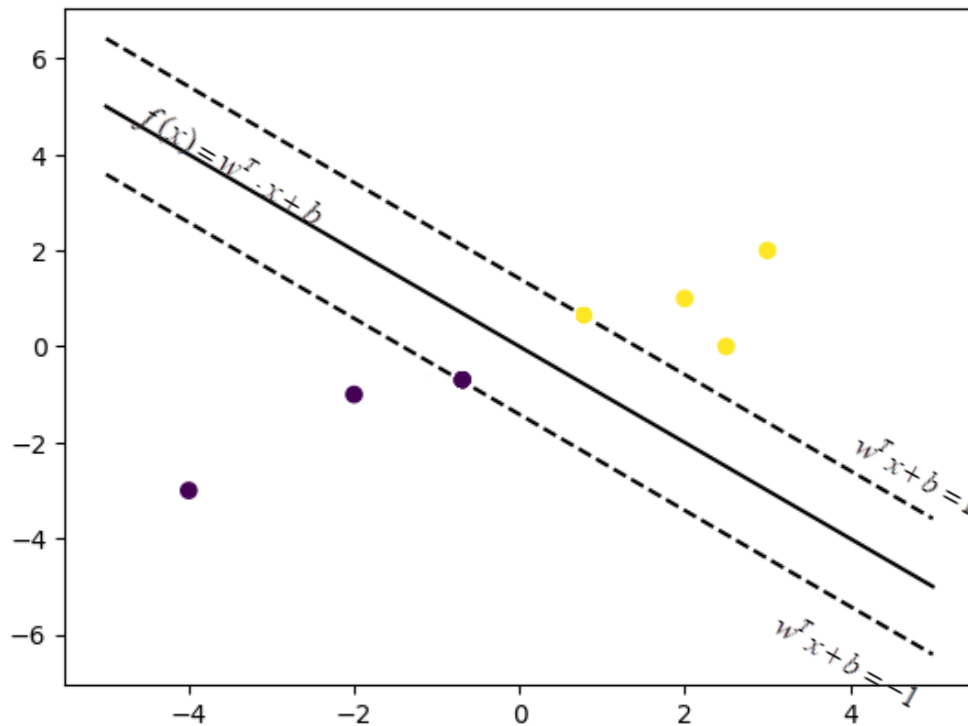


Figura 4: Ejemplo de SVM de margen suave

3.4.2. SUPPORT VECTOR REGRESSION

Support Vector Regression es un modelo particular de la familia de SVM. Como el resto de técnicas de regresión, SVR trata de construir una función que prediga el comportamiento de la variable respuesta que interviene en un estudio, utilizando para ello los principales beneficios del aprendizaje automático. El modelo estándar de SVR está construido a partir de técnicas cuyo objetivo es predecir el valor de la variable objetivo $y \in \mathbb{R}$ dado un vector de covariables $\mathbf{x} \in \mathbb{R}_+^m$. De esta manera, SVR establece una función $\hat{f}: \mathbb{R}_+^m \rightarrow \mathbb{R}$ tal que, $\hat{f}(\mathbf{x}) = \hat{y}$, donde \hat{y} es la predicción de la variable respuesta. La predicción de \hat{f} es definida como $\hat{f}(\mathbf{x}) = \mathbf{w}^* \phi(\mathbf{x}) + b^*$, donde $\mathbf{w}^* \in \mathbb{R}^q$ y $b^* \in \mathbb{R}$ son óptimos del modelo (10), $\phi(\cdot)$ es una función de transformación del espacio de las covariables y los valores $C \in \mathbb{R}_+$ y $\varepsilon \in \mathbb{R}_+$ son obtenidos por un proceso de validación.

$$\underset{\mathbf{w}, b, \xi'_i, \xi_i}{Min} \quad \|\mathbf{w}\| + C \sum_{i=1}^n (\xi'_i + \xi_i) \quad (.0)$$

$$y_i - (\mathbf{w}\phi(\mathbf{x}_i) + b) \leq \varepsilon + \xi'_i, \quad i = 1, \dots, n \quad (.1) \quad (10)$$

$$(\mathbf{w}\phi(\mathbf{x}_i) + b) - y_i \leq \varepsilon + \xi_i, \quad i = 1, \dots, n \quad (.2)$$

$$\xi'_i, \xi_i \geq 0, \quad i = 1, \dots, n \quad (.3)$$

Nótese que $\hat{f}(\mathbf{x}) = \mathbf{w}^* \phi(\mathbf{x}) + b^*$ tiene la estructura de un hiperplano en el espacio de transformación $(\phi(\mathbf{x}), y)$. El algoritmo genera un predictor $\hat{f}(\mathbf{x})$ de la variable respuesta para el vector \mathbf{x} y las superficies de “corrección” inferior y superior, definidas como $\hat{f}(\mathbf{x}) - \varepsilon$ y $\hat{f}(\mathbf{x}) + \varepsilon$, donde ε es un margen que otorga robustez al estimador asociado a SVR (ver *Figura 5*). Una vez definido este margen, las observaciones que se encuentren por debajo de la superficie $\hat{f}(\mathbf{x}) - \varepsilon$ tienen un error (empírico) de $\xi'_i > 0$ (con $\xi_i = 0$), mientras que las observación por encima de la superficie $\hat{f}(\mathbf{x}) + \varepsilon$ presentan un error (empírico) $\xi_i > 0$ (con $\xi'_i = 0$). Aquellas observaciones que se encuentren entre $\hat{f}(\mathbf{x}) - \varepsilon$ y $\hat{f}(\mathbf{x}) + \varepsilon$ tienen un error de cero (con $\xi_i = \xi'_i = 0$). La función objetivo de (10), representa la combinación de regresión y regularización que intervienen en SVR, en la que los objetivos de error empírico $\sum_{i=1}^n (\xi'_i + \xi_i)$ y error de generalización $\|\mathbf{w}\|$ se equilibran el uno al otro a través de un hiperparámetro de peso C (Vazquez and Walter (2003)). Además, aunque los hiperplanos tienen formas lineales, SVR puede generar funciones de predicción que no tienen que ser necesariamente lineales en el espacio original (véase la *Figura 5*). Esto se debe a la función ϕ , que es una transformación del espacio de covariables, $\phi: \mathbb{R}_+^m \rightarrow Z$. Es importante remarcar que $Z = \phi(\mathbb{R}_+^m)$, conocido como espacio de características, puede tener multitud de dimensiones, incluso infinitas. Esta transformación es la causa de los diferentes enfoques no lineales en \mathbb{R}_+^m , mientras que la función de predicción genera un hiperplano lineal en Z .

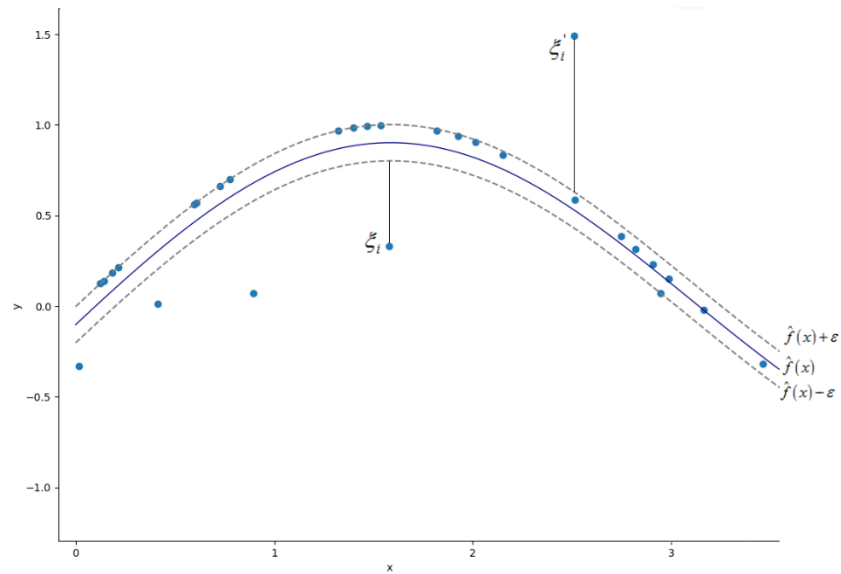


Figura 5. Ejemplo de SVR estándar

A veces, en la literatura, se aconseja trabajar con el modelo dual del problema (10) y con funciones kernel, definidas como $K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})\phi(\mathbf{x}')$. Esta técnica, también conocida como el truco del kernel (kernel trick), consiste en crear nuevas dimensiones a partir del conjunto de los inputs originales (espacio de atributos) para crear un espacio de mayor dimensionalidad y más fácilmente separable, conocido como espacio de características.

Tanto la función de transformación como la función kernel dependen de varios hiperparámetros (τ), que deben ser determinados con C y ϵ por validación cruzada. La validación cruzada es una técnica estándar de aprendizaje automático y estadística utilizada para ajustar los hiperparámetros de un modelo. En un ejemplo de validación cruzada de V -folds, la muestra de entrenamiento Ω es dividida aleatoriamente en $\Omega_1, \dots, \Omega_V$ submuestras separadas con el mismo tamaño de muestra o lo más parecido posible. Los valores típicos de V son 5 ó 10 (Friedman et al. (2001)). Sea la ν -ésima submuestra de aprendizaje $\Omega^{(\nu)} = \Omega - \Omega_\nu$ y dada $\Omega_i = \Omega - \Omega_{\nu(i)}$, donde $\Omega_{\nu(i)}$ es la submuestra tal que $i \in \Omega_{\nu(i)}$. Para cada $\nu = 1, \dots, V$, se construye un modelo individual aplicando el algoritmo a los datos de entrenamiento $\Omega^{(\nu)}$. A continuación, este modelo se evalúa mediante una función de costes utilizando los datos de test en Ω_ν . En particular,

dato un conjunto de hiperparámetros (C, ε, τ) , el error de predicción se determina como

$$\frac{1}{n} \sum_{i=1}^n \left(y_i - \hat{f}_{\Omega_i}(\mathbf{x}_i; C, \varepsilon, \tau) \right)^2, \text{ donde } \hat{f}_{\Omega_i}(\mathbf{x}_i; C, \varepsilon, \tau) \text{ es el predictor SVR evaluado en } \mathbf{x}_i$$

cuando se han considerado los datos de aprendizaje Ω_i y los hiperparámetros (C, ε, τ) .

Para elegir la mejor combinación de hiperparámetros utilizando la validación cruzada, se calcula el error de predicción para diferentes valores de hiperparámetros (C, ε, τ) .

Finalmente, se selecciona la combinación $(C^*, \varepsilon^*, \tau^*)$ con el menor error de predicción y se utiliza para entrenar un modelo SVR sobre el conjunto completo de datos Ω .

3.5. SUPPORT VECTOR MACHINES EN EL CONTEXTO DE LA ORDENACIÓN DE CARACTERÍSTICAS

Uno de los principales problemas del aprendizaje automático es identificar aquellas variables predictoras que son realmente importantes para mejorar la precisión del modelo a la hora de predecir los valores de una variable de respuesta. Esto se debe a que la inclusión de determinadas características en el problema puede provocar un sobreajuste del modelo o la aparición de ruido innecesario. Además, el uso de miles de variables puede ser inabordable computacionalmente a pesar de los avances en computación y almacenamiento de memoria. Debido a esto, se han implementado diferentes técnicas de ordenación de características utilizando métodos de aprendizaje automático, como SVM. El objetivo de estas técnicas es establecer un ranking de aquellas variables que suponen una mayor influencia en la consecución de los outputs. A continuación, se exponen algunas de las técnicas que se han desarrollado para su uso en el contexto de modelos basados en SVM.

3.5.1. SUPPORT VECTOR MACHINE-PERMUTATION FEATURE IMPORTANCE

Support Vector Machine-Permutation Feature Importance (SVM-PFI) es un método para establecer la importancia de las variables en un conjunto de datos, que fue introducido por [Breiman \(2001\)](#). En este artículo, el autor explica los fundamentos de la técnica de Árboles de Búsqueda Aleatoria (Random Forest) y argumenta que en numerosos análisis

de datos es necesario conocer la interacción que se produce entre las variables para mejorar la predicción de la precisión del modelo. Para ello, establece un método de evaluación de variables basado en la disminución de la precisión de la predicción.

A partir de un conjunto de datos $\mathbf{V} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$, donde \mathbf{x} representa un vector de variables predictoras e y es una variable de respuesta del problema, el primer paso consiste en seleccionar aleatoriamente un conjunto de entrenamiento \mathbf{V}_{train} y un conjunto de prueba \mathbf{V}_{test} del conjunto de datos original. A continuación, el modelo se entrena con el conjunto de datos \mathbf{V}_{train} mediante la técnica predictiva que se va a utilizar, como Random Forest o SVM, y se calcula el error E_0 con \mathbf{V}_{test} . Una de las métricas que se

pueden utilizar es el error cuadrático medio (MSE), $E_0 = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$ donde \hat{y}_i es la

predicción del modelo e y_i el valor observado. Después, para cada variable u -ésima del problema, los valores de los datos de esta variable u -ésima se barajan aleatoriamente creando un nuevo conjunto de datos, \mathbf{V}_{train}^u , y se entrena el modelo. Para el test, se utiliza

el conjunto de datos restante (\mathbf{V}_{test}) y el error se calcula como: $E_u = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i^{(u)} - y_i)^2$,

donde $\hat{y}_i^{(u)}$ es la predicción del modelo entrenado sobre \mathbf{V}_{train}^u .

La variación del error de predicción del modelo con respecto al error del modelo con los

datos originales, $\Delta E_u = \frac{E_u - E_0}{E_0}$, es la métrica utilizada para determinar el poder

predictivo de la variable.

Por último, se establece una clasificación con la variación de errores obtenida. Las variables asociadas con una mayor reducción del error de predicción del modelo son los más importantes.

3.5.2. SUPPORT VECTOR MACHINE-BOOTSTRAP RESAMPLING

Support Vector Machine-Bootstrap Resampling (SVM-BR) es una técnica desarrollada por [Alonso-Atienza et al. \(2012\)](#) que se utiliza para la selección de características,

aunque también se puede utilizar para clasificar tareas. Este método se basa en la combinación de SVM con técnicas de remuestreo Bootstrap (Efron (1979)).

El enfoque SVM-BR se basa en el uso de Bootstrap para medir la variación del rendimiento de SVM cuando se entrena con el conjunto de datos completo y, por otro lado, cuando se entrena con una submuestra en la que el número de características del problema se han reducido. Resolviendo el programa de optimización asociado al SVM, obtenemos unos coeficientes (\mathbf{w}^*, b_0^*) que dependen del conjunto de datos \mathbf{V} y de algunos hiperparámetros.

Una muestra Bootstrap $\mathbf{V}' = \{(\mathbf{x}_{1'}, y_{1'}), \dots, (\mathbf{x}_{n'}, y_{n'})\}$ es un nuevo conjunto de datos creado seleccionando al azar observaciones de \mathbf{V} con reemplazo. Una muestra de \mathbf{V} se puede ver en términos de remuestreo como: $\mathbf{V}' = (\mathbf{V}'_{in}, \mathbf{V}'_{out})$, donde \mathbf{V}'_{in} y \mathbf{V}'_{out} son el subconjunto de muestras incluidas y excluidas en el remuestreo. Los coeficientes (\mathbf{w}^*, b_0^*) se obtienen resolviendo el programa de optimización correspondiente utilizando únicamente la información contenida en el conjunto de datos \mathbf{V}'_{in} , mientras que el error de predicción (basado en el MSE) R^* se calcula utilizando los datos \mathbf{V}'_{out} .

Dado un conjunto de muestras de arranque independientes B , $\{\mathbf{V}'(1), \dots, \mathbf{V}'(B)\}$, podemos calcular la función de densidad de riesgo a partir de la histograma generado por los errores de todas las muestras $R^*(b)$, donde $b = 1, \dots, B$.

Alonso-Atienza et al. (2012) utilizan un subconjunto reducido \mathbf{V}_u de observaciones en las que se elimina la variable u y crean una muestra Bootstrap con sus respectivos subconjuntos: $\mathbf{V}'_{u,in}$ y $\mathbf{V}'_{u,out}$. A partir de ahí, el error se calcula como: $R_u^*(b)$. Finalmente, utilizan la diferencia entre el error del modelo completo con el modelo al quitar la variable u dando lugar a la medición: $\Delta R_u^*(b) = R_u^*(b) - R^*(b)$, donde un mayor valor de $\Delta R_u^*(b)$, indica una mayor importancia de la variable u .

3.5.3. SVM-RFE PSEUDOSAMPLES

SVM-RFE Pseudosamples es un método propuesto por [Sanz et al. \(2018\)](#) que tiene como objetivo evaluar la importancia de las variables de un conjunto de datos en función de la variación de la desviación absoluta media (MAD) cuando se crea una pseudomuestra de los datos. Esta técnica consiste en una extensión de las ideas propuestas por [Postema et al., \(2011\)](#) y [Krooshof et al., \(2010\)](#), que evalúan la importancia de las variables a través de las transformaciones de los datos que se producen al utilizar SVR.

Para obtener una clasificación de la importancia de las variables predictoras del modelo, primero se entrena un modelo SVM con todo el conjunto inicial de datos, \mathbf{V} . A partir de ahí se obtienen los coeficientes (\mathbf{w}^*, b_0^*) e hiperparámetros óptimos. A continuación, se crean m submuestras de datos (\mathbf{V}_m) , donde m es igual al número de variables predictoras en el conjunto de datos. Estas matrices de tamaño $q \times p$ estarán formadas por unas pseudomuestras en las que los valores de la columna de la variable a evaluar estarán formados por los cuantiles q de esa columna. El resto de valores de las demás columnas estarán formados por la media o mediana de estas variables, aunque dado que los datos se consideran normalizados se sustituyen por 0.

Una vez creadas las pseudomuestras, se utilizan las soluciones del modelo inicialmente entrenado (\mathbf{w}^*, b_0^*) para obtener la predicción de la variable respuesta asociada a cada una de ellas (\hat{y}_{q_p}) . Luego, la variabilidad de las predicciones de la pseudomuestra se calcula a través de la desviación absoluta de la mediana: $MAD_p = \text{median}\left(\left|\hat{y}_{q_p} - \text{median}(y_p)\right|\right)$.

Finalmente, se eliminarán las variables que tengan un menor valor de MAD por considerarse que tienen una menor contribución al modelo. El proceso se repite hasta que se cumple un determinado criterio de parada.

3.5.4. SVM-RFE DUAL OBJECTIVE VARIATION

Otro enfoque en la literatura, presentado por [Guyon et al., \(2002\)](#), analiza el cambio en la función objetivo del modelo SVM dual cuando una variable se incluye o excluye en el proceso de producción.

El problema dual tiene un valor objetivo J , y este método calcula el cambio en la función de coste estimada del problema dual causado por la eliminación de una característica, es decir, al forzar que cada término relacionado con la variable en cuestión sea 0, manteniendo la constante de las variables duales.

Primero se resuelve el modelo original con validación cruzada para obtener los mejores hiperparámetros para el modelo, antes de centrar la atención en el problema dual.

Para hacer que este método sea manejable computacionalmente, en lugar de resolver el modelo dual con cada característica eliminada, el problema, en su forma dual, se resuelve solo una vez por paso con todas las características, obteniendo valores para las variables duales y el valor objetivo J del problema dual. Luego, la solución para las variables duales obtenidas se mantiene constante y estos valores se utilizan para calcular el efecto en J de eliminar una sola característica u , denotada por $J(u)$.

La característica u que da como resultado el cambio más pequeño en el valor objetivo $DJ(u)$ se considera la menos importante en la clasificación, y el proceso se itera sin esta variable, hasta que se clasifican todas las variables.

3.6. SUPPORT VECTOR MACHINES EN EL CONTEXTO DE SELECCIÓN DE CARACTERÍSTICAS

Las Máquinas de Soporte Vectorial, a pesar de poseer una gran capacidad generalizadora y actuar sobre conjuntos de datos con múltiple número de inputs, pueden sufrir problemas de sobreajuste. Por ello, la reducción del número de características puede ayudar a mejorar la predicción del modelo o simplemente contribuir a reducir los costes tanto computacionales como económicos del análisis.

Partiendo del modelo (9), cuando se incorpora la selección de características, el problema se vuelve más difícil y no convexo. Para todo $j \in \{1, \dots, m\}$, sea t_j una variable binaria que toma el valor de 1 si y sólo si se selecciona la característica j . Para restringirse a las soluciones factibles del modelo anterior que utilizan p características, es necesario añadir las siguientes restricciones:

$$\sum_j t_j = p \quad (1)$$

$$|w_j| \leq M t_j \quad j = 1, \dots, m, \quad (2) \quad (11)$$

$$t_j \in \{0, 1\} \quad j = 1, \dots, m, \quad (3)$$

donde M es una constante grande. Si $t_j = 1$, la restricción $|w_j| < M$ es redundante porque M es grande. Si $t_j = 0$, es decir, la característica j no se selecciona y la restricción 11.2 establece que la característica j no pertenece al modelo. Por lo tanto, obtener la frontera Pareto combinando los dos objetivos en uno solo y asignando valores diferentes al parámetro C , es para algunos casos imposible. Para asegurar dicho frente, es necesario considerar el problema desde un enfoque puramente multiobjetivo, manteniendo la independencia de cada uno de los objetivos y optimizándolos de forma simultánea. El modelo para considerar ambos objetivos por separado es el siguiente:

$$\min \quad O_{11} = \frac{1}{2} \|w\|^2 \quad (0)$$

$$\min \quad O_{12} = \sum_{i=1}^n \xi_i \quad (1)$$

$$\text{s.t.} \quad y_i(w^T x_i + b) \geq 1 - \xi_i \quad i = 1, \dots, n, \quad (2)$$

$$\xi_i \geq 0 \quad i = 1, \dots, n, \quad (3)$$

$$w_j \in \mathbb{R} \quad j = 1, \dots, m, \quad (4) \quad (12)$$

$$b \in \mathbb{R} \quad (5)$$

$$\sum_j t_j = p \quad j = 1, \dots, m, \quad (6)$$

$$|w_j| \leq M t_j \quad j = 1, \dots, m, \quad (7)$$

$$t_j \in \{0, 1\} \quad j = 1, \dots, m, \quad (8)$$

Las soluciones del modelo anterior forman una frontera Pareto óptima y nos ofrecen una variedad de alternativas entre las que el decisor puede elegir. Seguidamente, se pueden utilizar varias métricas de rendimiento para seleccionar el mejor clasificador.

3.6.1. TÉCNICAS PARA LA SOLUCIÓN DE SVM CON SELECCIÓN DE CARACTERÍSTICAS

En la literatura existen varias técnicas para la solución del problema planteado para SVM con selección de características. La frontera Pareto exacta se puede calcular utilizando el método AUGMECON desarrollado por [Mavrotas \(2009\)](#) o su versión mejorada, AUGMECON2, que básicamente mejora la eficiencia del método cuando crece el número de objetivos ([Mavrotas y Florios, 2013](#)).

En este método, asumimos que $f_1(x)$ y $f_2(x)$ son las dos funciones objetivo a maximizar, S es la región factible para el programa multi-objetivo entero y $r_2 = UB_2 - LB_2$ es el rango de $f_2(x)$.

El rango r_2 se divide en q intervalos iguales usando $q-1$ puntos intermedios de rejilla equidistantes. De este modo, los $q+1$ puntos de la cuadrícula se utilizan para variar paramétricamente el RHS (e_2) de $f_2(x)$: $e_2 = LB_2 + \ell r_2 / q$ para $\ell = 0, \dots, q$. En los métodos AUGMECON el problema resuelto iterativamente es el siguiente:

$$\begin{aligned}
 \text{Max} \quad & f_1(x) + \varepsilon s_2 / r_2 & (.0) \\
 \text{s.a.} \quad & f_2(x) - s_2 = e_2 & (.1) \\
 & x \in S, s_2 \in R^+ & (.2)
 \end{aligned} \tag{13}$$

donde ε es un número adecuadamente pequeño (normalmente entre 10^{-3} y 10^{-6}) y s_2 es la variable de excedente de la restricción.

En cada iteración la variable s_2 se utiliza para calcular el coeficiente de derivación $b = \text{int}(q s_2 / r_2)$. $\text{int}()$ es la función que devuelve la parte entera de un número real. Si

$b > r_2/q$, luego en la siguiente iteración se obtendrá la misma solución con la única diferencia dada por la variable excedente. El coeficiente de derivación b en realidad indica cuántas iteraciones consecutivas podemos omitir.

AUGMECON2 utiliza la optimización lexicográfica para calcular el rango r_2 . En términos generales, la optimización lexicográfica de una serie de funciones objetivo consiste en optimizar la primera función objetivo y luego entre los posibles óptimos alternativos optimizar la segunda función objetivo y así sucesivamente.

Cuando se tiene en cuenta la selección de características es necesario incorporar las restricciones mostradas en (11). De este modo, el problema iterativo a solucionar es el siguiente:

$$\begin{aligned}
 \text{Min} \quad & O_1(x) - \varepsilon s_2 / r_2 & (.0) \\
 \text{s.a.} \quad & O_2(x) + s_2 = e_2 & (.1) \\
 & x \in S, s_2 \in R^+ & (.2)
 \end{aligned} \tag{14}$$

En AUGMECON2, para el SVM con selección de características, el límite superior lexicográfico UB_2 en O_2 se define como $UB_2 = \min \{ (w, b, \varepsilon, t) \in S, O_1(w, b, \varepsilon, t) = O_1^* \}$ donde $O_1^* = \min \{ O_1(w', b', \varepsilon', t') : (w', b', \varepsilon', t') \in S \}$. Claramente, UB_2 es un límite superior en el valor de O_2 para cualquier punto eficiente SVM. Finalmente, el límite inferior lexicográfico del O_2 es $LB_2 = \min \{ O_2 : (w, b, \varepsilon, t) \in S \}$.

No obstante, la técnica exacta no consigue dar soluciones en problemas de tamaño medio o grande en tiempos de computación aceptables. Por ello, es necesario recurrir a otro tipo de métodos que, si bien no aseguran la frontera exacta, proporciona una aproximación muy cercana a ella: las técnicas metaheurísticas. [Alcaraz et al. \(2022\)](#) propusieron una metaheurística multiobjetivo demostrando su capacidad para construir una buena aproximación de la frontera Pareto en una amplia variedad de instancias de tamaños muy variados. Aunque en la literatura existen otros trabajos que proponen metaheurísticas para resolver el problema de SVM con selección de características, todos ellos implementan

el modelo con una única función en la que se combinan ambos objetivos, el riesgo estructural y el riesgo empírico. En estos casos, la metaheurística se utiliza para determinar los parámetros de este modelo y no para resolver el modelo en sí. Seguidamente, una vez establecidos estos parámetros, se resuelve el modelo clásico SVM con un único objetivo mediante técnicas exactas. La principal ventaja de la optimización multiobjetivo es que no proporciona una solución única, sino que proporciona un conjunto de soluciones no dominadas en la que todas son óptimas. Además, todas las soluciones de los enfoques con un único objetivo, que es una combinación lineal de los múltiples objetivos, están contenidas en la frontera eficiente del problema multiobjetivo, por lo que cualquier decisión que se adopte optimizando un objetivo es una opción cuando se trata de optimizar varios. Y lo que es más importante, entre todas las soluciones eficientes del problema multiobjetivo, el decisor puede encontrar soluciones novedosas para su problema que de otra forma no habría considerado.

El coste computacional de la optimización multiobjetivo frente a la optimización mono-objetivo es elevado. Mientras que el valor óptimo de un problema de un solo objetivo es único, el número de soluciones no dominadas en un problema multiobjetivo es grande, incluso puede ser infinito. Dado que el coste computacional de encontrar la solución óptima del problema con un objetivo es similar al coste computacional de encontrar una solución eficiente del problema con varios objetivos, se puede suponer que el coste computacional del problema con varios objetivos es similar al producto del coste computacional de una solución eficiente por el número de soluciones eficientes. Este hecho implica que los problemas con varios objetivos se tratan a menudo mediante una heurística en lugar de un enfoque exacto.

Los algoritmos evolutivos multiobjetivo (MOEA) se han convertido en una alternativa a los enfoques exactos para resolver problemas de optimización difíciles cuando se consideran varios objetivos ya que estas técnicas requieren un tiempo de computación excesivo. Uno de los más utilizados es el NSGA-II, (Non-Dominated Sorting Genetic Algorithm II, en inglés, [Deb \(2002\)](#)) que se ha aplicado con éxito para encontrar un conjunto diverso de soluciones en muy diferentes tipos de problemas de optimización con

varios objetivos. NSGA-II mejora el rendimiento de su predecesor NSGA desarrollado por [Srinivas y Deb \(1995\)](#), que fue uno de los primeros MOEAs.

NSGA-II combina ideas de los algoritmos genéticos con conceptos de dominación de la optimización multiobjetivo. Algunos otros MOEAs también se han utilizado con éxito para diferentes problemas multiobjetivo. Por ejemplo, el Strength Pareto Evolutionary Algorithm (SPEA-2) definido por [Kim et al. \(2004\)](#) o el Pareto Archive Evolutionary Strategy (PAES), [Knowles y Corne, \(2000\)](#).

[Alcaraz et al. \(2022\)](#) proponen una nueva metaheurística para obtener las fronteras Pareto (o sus aproximaciones) del modelo clásico SVM de margen suave con selección de características. La metaheurística está basada en el NSGA-II y se presenta en pseudocódigo en el Algoritmo 1. Aunque el funcionamiento de este algoritmo se explica con detalle en el artículo citado, puede resumirse como sigue. En primer lugar, se crea aleatoriamente una población de un tamaño fijo (N) de individuos. Se evalúa cada uno de estos individuos, es decir, se calculan los dos objetivos y luego se agrupan las soluciones en función de criterios de dominación. Los individuos que no están dominados por ningún otro en la población forman el primer frente (F_1), los individuos dominados por una o más soluciones de F_1 forman F_2 , y así sucesivamente. A continuación, comienza el proceso de evolución, que se lleva a cabo hasta que se cumple el criterio de parada, y suele estar relacionado con el tiempo de cálculo o el número de soluciones evaluadas. En primer lugar, se emplean el cruce y la mutación para generar una nueva población de tamaño N . Se eligen dos soluciones, una madre y un padre, utilizando de nuevo criterios de dominación, que se recombinan para generar una nueva solución que también puede verse afectada por la mutación. También se evalúan los individuos de la nueva población. A continuación, la población original y la nueva se fusionan en una población de tamaño doble y los individuos se ordenan de nuevo. El último paso consiste en reducir el tamaño de la población actual de $2N$ a su tamaño original, N . Para ello, seleccionamos los individuos, por orden, de los frentes consecutivos, F_1, F_2, \dots . Cuando todos los individuos de un determinado frente no pueden ser colocados en la nueva

población, éstos se ordenan con un criterio de distancia, y sólo los mejores son seleccionados para formar parte de la nueva población.

```

1  $P_0 = \text{create\_initial\_population}(N)$ ;
2  $t = 0$ ;
3  $P_0 = \text{evaluate\_population}(P_0)$ ;
4  $F = \text{fast\_non\_dominated\_sort}(P_0)$ ;
5 while not stopping\_criterion do
6   for  $j = 1$  to  $N$  do
7      $(\text{mother}, \text{father}) = \text{tournament\_dominance\_selection}(P_t)$ ;
8      $Q_t[j] = \text{crossover}(\text{mother}, \text{father})$ ;
9     if  $\text{random}() \leq PMUT$  then
10       $Q_t[j] = \text{mutation}(Q_t[j])$ ;
11    $Q_t = \text{evaluate\_population}(Q_t)$ ;
12    $R_t = P_t \cup Q_t$ ;
13    $F = \text{fast\_non\_dominated\_sort}(R_t)$ ;
14    $P_{t+1} = \emptyset$ ;
15    $i = 1$ ;
16   while  $|P_{t+1}| + |F_i| \leq N$  do
17      $P_{t+1} = P_{t+1} \cup F_i$ ;
18      $i = i + 1$ ;
19   if  $|P_{t+1}| < N$  then
20      $\text{i\_distance\_assignment}(F_i)$ ;
21      $\text{sort}(F_i, \text{i\_distance})$ ;
22      $P_{t+1} = P_{t+1} \cup F_i[1 : (N - |P_{t+1}|)]$ ;
23    $t = t + 1$ ;
24 return  $F_1$ 

```

Algoritmo 1: Pseudocódigo del algoritmo propuesto por Alcaraz et al. (2022)

Los individuos de la población en el Algoritmo 1, representan soluciones del problema y se codifican con una estructura. La codificación de las soluciones es uno de los aspectos fundamentales en el diseño de la metaheurística y contribuye en gran medida a la eficiencia de la técnica. La estructura propuesta en el Algoritmo 1 consta de tres componentes:

- Modo: dos posibles valores. Modo=A indica que tenemos p vectores del conjunto de datos inicial (Ω) de la clase -1 y uno de la clase 1; modo=B indica que se construyen hiperplanos con un vector de la clase -1 y p vectores de la clase 1.

- Vectores: un array con los índices de los $p+1$ vectores utilizados para construir los dos hiperplanos. La/s primera/s posición/es de la matriz se ocupará/n con el/los vector/es de la clase -1 y la/s siguiente/s posición/es con el/los vector/es de la clase 1.
- Características: una matriz con los índices de las p características seleccionadas en la solución. Las características seleccionadas consisten en una lista no ordenada de un subconjunto del conjunto completo de características.

En la Figura 6 se muestra un ejemplo de codificación para un ejemplo sencillo. En el modo A, tenemos dos vectores de la clase + y un vector de la clase \times . Las características seleccionadas son: 1 y 2. En el modo B, tenemos el caso contrario, dos vectores de la clase \times y un vector de la clase + y las mismas características seleccionadas. Para cada solución, se pueden construir los hiperplanos paralelos y el hiperplano separador. Esto crea un clasificador y a partir de él se pueden calcular los diferentes objetivos.

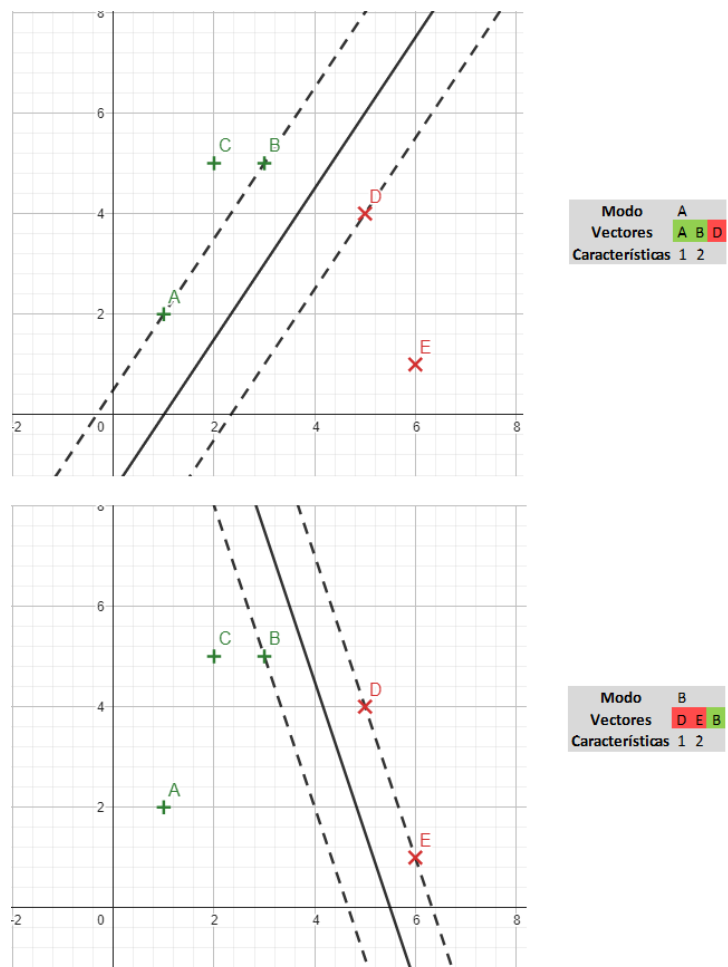


Figura 6: Ejemplo de codificación propuesto por Alcaraz et al. (2022)

La mayoría de metaheurísticas que tratan la selección de características emplean un cromosoma de máscara binaria con n genes, uno por cada característica observada (Huang and Wang (2006), García-Pedrajas et al. (2014), Aladeemy (2017), Faris (2018)) que indica si la característica correspondiente está seleccionada o no. Sin embargo, en el problema considerado en este trabajo, el número de características seleccionadas ($p \leq m$) es una entrada del problema. En ese sentido, el uso de la máscara binaria requeriría m genes mientras que esta codificación sólo utiliza p y, por tanto, la codificación empleada en este caso es más adecuada.

3.6.2. MÉTRICAS DE RENDIMIENTO PARA LA COMPARACIÓN DE DIFERENTES CLASIFICADORES

Una vez obtenida una frontera Pareto óptima a través del modelo (12), cada punto de la frontera es una solución que se corresponde con un clasificador del problema. En la literatura existen diferentes métricas para evaluar el rendimiento de una técnica de aprendizaje automático. Estas técnicas son muy utilizadas en campos tan variados como la medicina [Ismael y Sengür \(2021\)](#) o la ingeniería [Roshani et al. \(2021\)](#) por citar algunos. En problemas de clasificación binaria, como las Máquinas de Vectores Soporte, es habitual utilizar una matriz de confusión que representa la predicción del algoritmo frente a los valores reales. En la Figura 7, + indica que la clase es positiva mientras que – indica que la clase es negativa. Esto nos da cuatro posibilidades respecto a la concordancia entre valores predichos y valores reales.

- Verdadero positivo (VP): la variable observada y la proporcionada por el modelo son ambas positivas.
- Verdadero negativo (VN): la variable observada y la proporcionada por el modelo son ambas negativas.
- Falso positivo (FP): la variable observada es negativa y la proporcionada por el modelo es positiva.
- Falso negativo (FN): la variable observada es positiva y la proporcionada por el modelo es negativa.

		REAL	
		+	-
PRED.	+	VP	FP
	-	FN	VN

Figura 7: Matriz de confusión

A partir de estos valores es posible establecer diferentes medidas que permiten evaluar la calidad de un modelo. Algunas de estas medidas son las siguientes:

- La sensibilidad es la capacidad del modelo para predecir casos positivos. Se calcula como la proporción entre las predicciones realmente positivas y el conjunto de todas las predicciones positivas.

$$\text{Sensibilidad} = \frac{VP}{VP + FN}$$

- La especificidad es la capacidad del modelo para predecir casos negativos. Es la proporción entre las predicciones verdaderamente negativas y el conjunto de todas las predicciones negativas.

$$\text{Especificidad} = \frac{VN}{VN + FP}$$

- La precisión es una medida utilizada para evaluar las instancias relevantes entre las instancias recuperadas. Se calcula como la proporción entre las predicciones realmente positivas y el conjunto de todos los valores positivos reales.

$$\text{Precisión} = \frac{VP}{VP + FP}$$

- La exactitud o accuracy, en inglés, es la proporción de predicciones correctas. Mide lo bien que una clasificación predice una condición.

$$\text{Accuracy (ACC)} = \frac{VP + VN}{N}$$

- F-Score (FSC) es una métrica que tiene en cuenta la precisión y la sensibilidad de la misma manera. Es la media armónica entre precisión y sensibilidad.

$$F - \text{Score (FSC)} = \frac{\text{Precisión} * \text{Sensibilidad}}{\text{Precisión} + \text{Sensibilidad}}$$

- El área bajo la curva ROC (Area under curve (AUC), en inglés) es una medida utilizada para comparar diferentes clasificadores. Esta estadística mide el área generada por la curva ROC (Receiver Operating Characteristic, en inglés). La curva ROC representa la sensibilidad frente a la especificidad según un umbral de discriminación. En

el espacio ROC, se representa 1-especificidad en el eje X y la sensibilidad en el eje Y. Los modelos que se aproximan al punto (0,1) son mejores clasificadores, ya que presentan valores más altos de AUC.

$$AUC = \frac{\text{Sensibilidad} * \text{Especificidad}}{2}$$

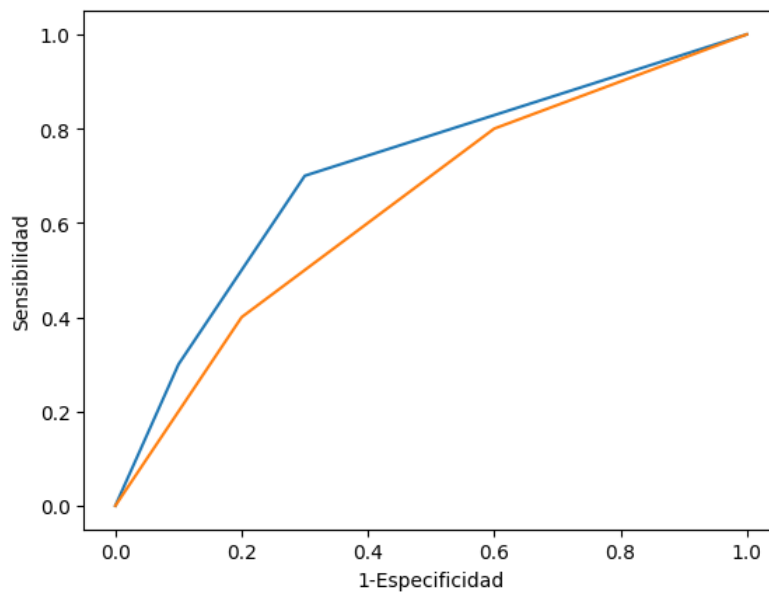


Figura 8: Ejemplo de AUC para dos clasificadores

- El índice Kappa (KAP) es un estadístico definido por [Cohen \(1960\)](#), que mide el acuerdo o desacuerdo entre dos clasificadores. Para ello, se calcula la probabilidad de acuerdos que no se dan por azar ($ACC - P_c$) y la probabilidad de desacuerdos por azar ($1 - P_c$). Las probabilidades que se tienen en cuenta para esta medida son el acuerdo observado (ACC) y la probabilidad de que dos clasificadores elijan el mismo por azar (P_c).

$$P_c = \frac{(TP + FN) * (TP + FP) + (TN + FN) * (TN + FP)}{N^2}$$

$$KAP = \frac{ACC - P_c}{1 - P_c}$$

En todas las métricas anteriores un mayor valor indica una mejor calidad del modelo a evaluar. La elección de cada medida depende del evaluador y de la naturaleza del problema por lo que se suelen elegir diferentes métricas a la hora de elaborar un estudio. Por ejemplo, en casos médicos un falso positivo o negativo pueden costar la vida de un paciente por lo que se tenderá a prestar mayor atención a minimizar esos valores.

3.6.3. MÉTRICAS DE COMPARACIÓN DE FRONTERAS

La evaluación de dos fronteras exactas o aproximadas puede realizarse a través de distintas métricas presentes en la literatura para la comparación de métodos de optimización multiobjetivo. La comparación se basa en la cuantificación de diferentes características deseables en las fronteras que obtienen los diferentes métodos como resultado. Estas métricas pueden clasificarse en varios grupos dependiendo de la característica que evalúan ([Audet et al. \(2021\)](#)):

- Cardinalidad: enumeran el número de soluciones no dominadas de un algoritmo.
- Convergencia: cuantifican lo cerca que está la frontera proporcionada por el algoritmo de la frontera óptima.
- Distribución y extensión: cuantifican la distribución de una frontera Pareto y la dispersión de las soluciones de ésta ([Custodio et al. \(2011\)](#)).

Teniendo en cuenta que el número de métricas diferentes existentes de cada tipo es extenso (en el trabajo de [Audet et al. \(2021\)](#) se hace una recopilación de ellas). Basándonos en la selección hecha por [Alcaraz et al. \(2022\)](#), hemos seleccionado las siguientes para ilustrar algunas de ellas: en el caso de las métricas de cardinalidad, tenemos la métrica OVNG (Overall non-dominated Vector Generation ([Van Veldhuizen \(1999\)](#)) que mide el número de soluciones no dominadas que tiene una frontera. Otra métrica de este tipo es C , una medida propuesta por [Zitzler et al. \(1998\)](#) que determina la fracción de soluciones dominadas de una frontera F_1 frente a otra frontera F_2 , $C(F_1, F_2)$. Como métricas de convergencia encontramos las métricas M_3 ([Zitzler et al. \(2000\)](#)) y Γ/M_3 ([Alcaraz \(2022\)](#)). M_3 es un indicador que evalúa la extensión de la frontera. Para ello, mide la distancia entre los puntos más alejados de la frontera, por lo

que valores elevados de esta métrica son deseados. Γ (Custodio et al. (2011)) calcula la separación máxima entre dos soluciones consecutivas de la frontera. En este caso, los valores deseados deben ser pequeños. La métrica Γ/M_3 relaciona las dos métricas anteriores calculando la fracción entre Γ y M_3 , por ello, valores bajos indican un mejor rendimiento de la frontera evaluada. Finalmente, como métrica de distribución podemos obtener la S-Metric o Hipervolumen (Zitzler (1999)) que determina el volumen cubierto por la frontera Pareto obtenida.

3.7. SUPPORT VECTOR MACHINE Y EL COSTE DE LA SELECCIÓN DE CARACTERÍSTICAS COMO OBJETIVO

El gran número de dimensiones y los costes asociados de una investigación son algunos de los problemas con los que se puede encontrar el analista de datos a la hora de realizar un estudio. Por ejemplo, a la hora de ejecutar un diagnóstico médico, las pruebas necesarias tienen un coste económico y temporal. La selección de características es un método ampliamente utilizado en la literatura para reducir esos costes, ya sean económicos, temporales o técnicos, además sirve para mejorar el rendimiento de los modelos evitando posibles sobreajustes o ruido innecesario.

Existen muchas técnicas para la selección de características que pueden ser divididas en diferentes grupos: métodos de filtro, métodos de envoltura o métodos embebidos (véase Ventakesh et al. (2019)). Pero la mayoría de los métodos no tienen en cuenta los costes asociados a la selección de características dentro de la selección, asignándoles el mismo peso a cada una de ellas (en el contexto de SVM véase, por ejemplo, los métodos analizados para la ordenación de características en la Sección 3.5). Ignorar el coste de las características puede llevar a buenos subconjuntos teóricos pero que en la práctica no pueden implementarse por falta de recursos. En la actualidad, sí que hay autores que tienen en cuenta el coste de la selección de características. Por ejemplo, por mencionar algunos autores: Min et al. (2014) proponen un modelo de selección de características con restricciones de coste, Zhang et al. (2015) utilizan el algoritmo genético de optimización de enjambre de partículas (PSO, en inglés) para solucionar el problema, Zhou et al. (2015) establecen un algoritmo de selección de características basado en

bosques aleatorios que incorpora el costo de las características en el proceso de construcción del árbol de decisión base, para producir subconjuntos de características de bajo costo. En cuanto a SVM con selección de características sí que existen métodos que permiten obtener una frontera Pareto de clasificadores (véase la [Sección 3.6](#)). Además, otros autores han estudiado el tema de SVM con selección de características. Por ejemplo: [Maldonado et al. \(2014\)](#) introducen dos modelos enteros mixtos de SVM con selección de características; [Aytug \(2015\)](#) resuelve la SVM con selección de características con la ayuda de la descomposición de Benders; [Gaudioso et al. \(2017\)](#) proponen un enfoque de relajación lagrangiana para el problema SVM con selección de características; [Labbe et al. \(2019\)](#) mejoran el modelo de Maldonado o [Benitez-Peña et al. \(2019\)](#) proponen un problema lineal entero más un problema convexo cuadrático. No obstante, no se conoce actualmente un método basado en SVM y selección de características que proporcione una frontera Pareto de soluciones y a la vez tenga en cuenta, además como objetivo, el coste de las características seleccionadas.

CAPÍTULO 4: RESULTADOS

En este capítulo se expone la nueva metodología y los resultados obtenidos a lo largo de la investigación. Primero se ilustra la nueva metodología obtenida para adaptar SVR al contexto de estimación de fronteras de producción en microeconomía e ingeniería. Esta nueva metodología ha sido bautizada como Support Vector Frontiers y es una técnica innovadora que mezcla por primera vez la estimación de fronteras con Máquinas de Soporte Vectorial. En un primer momento, se aborda el problema desde una perspectiva con una única variable respuesta. Posteriormente, se enfoca la cuestión desde una perspectiva con múltiples variables de respuesta. Después, se muestra las nuevas técnicas desarrolladas a partir de SVF para el ranking de inputs del proceso productivo. Seguidamente, se analiza una nueva metodología implementada para la resolución del problema de SVM con selección de características a través de un nuevo enfoque basado en métricas definidas para evaluar el rendimiento de diferentes clasificadores. Finalmente, se desarrolla un nuevo modelo de SVM con selección de características en el que se tiene en cuenta un nuevo objetivo: la suma de los costes asociados a la selección de cada característica.

4.1. SUPPORT VECTOR FRONTIERS

4.1.1. EL CASO DE UNA ÚNICA VARIABLE RESPUESTA

En esta sección, introducimos una nueva técnica basada en una adaptación metodológica del SVR estándar para la estimación de funciones de producción, que denominamos Support Vector Frontiers (SVF), centrándonos principalmente en el enfoque con una sola variable de respuesta (output). Este nuevo punto de vista permitirá estimar las funciones

de producción, satisfaciendo los axiomas habituales de la microeconomía, mediante un enfoque basado en datos que no asume ninguna distribución aleatoria particular sobre los mismos y genera, en una primera etapa, una función escalonada como estimador. En una segunda etapa, la convexificación de esta función escalonada nos permitirá proporcionar un estimador lineal a trozos de la función de producción. Estas formas son compartidas con los estimadores derivados de las técnicas FDH y DEA, respectivamente. Sin embargo, mientras que estas últimas sufren el problema del sobreajuste, el nuevo método intentará superar este inconveniente minimizando el riesgo estructural como hace SVR. Para empezar, mostraremos cómo se ha modificado el modelo SVR estándar para estimar superficies envolventes, satisfaciendo ciertos axiomas clásicos de la teoría de la producción (véase, por ejemplo, [Färe y Primont, \(1995\)](#)):

- (A1) si $\mathbf{x} = \mathbf{0}_m$, entonces $f(\mathbf{x}) \leq \mathbf{0}$;
- (A2) si $\mathbf{x} \leq \mathbf{x}'$, entonces $f(\mathbf{x}) \leq f(\mathbf{x}')$;
- (A3) f es una función cóncava.

El axioma A1 significa que si una empresa no consume ningún input, no puede producir nada. El axioma A2 significa que si una empresa consume más recursos, siempre es posible producir, al menos lo mismo; mientras que A3 está asociado a la convexidad del conjunto de posibilidades de producción T .

En esta sección se explica como el modelo SVR se ha adaptado paso a paso cumpliendo cada uno de los axiomas mencionados anteriormente. Además, se muestra la relación existente entre FDH y DEA y el nuevo marco de aprendizaje automático. De hecho, uno de nuestros objetivos era mostrar que bajo ciertas hipótesis estas dos técnicas estándar en la medición de la eficiencia pueden reinterpretarse como modelos SVM. Por último, introducimos una nueva definición robusta de eficiencia técnica, denominada eficiencia técnica ε -insensible, como aplicación natural de la noción de margen en SVM al campo de la medición de la eficiencia.

4.1.1.1. LA ESTIMACIÓN DE SUPERFICIES ENVOLVENTES

En esta subsección se explica cómo se ha adaptado el algoritmo SVR para la estimación de fronteras envolventes y como hemos ajustado trivialmente el modelo para cumplir A1.

A través del proceso de generación de datos, sabemos que nuestra función objetivo a estimar (f) y la variable respuesta tienen la siguiente relación con respecto a cada ejemplo de aprendizaje i , $i = 1, \dots, n$: $y_i \leq f(\mathbf{x}_i)$ (modelo determinista). Por lo tanto, parece natural obligar al estimador a cumplir la misma condición y añadirla a las restricciones del modelo SVR (10): $y_i \leq \hat{f}(\mathbf{x}_i) = \mathbf{w}\phi(\mathbf{x}_i) + b$, $i = 1, \dots, n$. Sin embargo, este nuevo tipo de restricción implica directamente la satisfacción de la restricción (10.1), $y_i - (\mathbf{w}\phi(\mathbf{x}_i) + b) \leq \varepsilon + \xi'_i$, ya que $y_i \leq \mathbf{w}\phi(\mathbf{x}_i) + b \Leftrightarrow y_i - (\mathbf{w}\phi(\mathbf{x}_i) + b) \leq 0$ y, además, tenemos que $\varepsilon + \xi'_i \geq 0$, para todo $i = 1, \dots, n$. En consecuencia, la restricción (10.1) puede eliminarse del nuevo modelo de optimización si se incorporan las condiciones, $y_i \leq \mathbf{w}\phi(\mathbf{x}_i) + b$ $i = 1, \dots, n$. Y, por la misma razón, las variables de decisión, ξ'_i $i = 1, \dots, n$, también pueden eliminarse. Además, una forma sencilla de satisfacer el axioma A1 consiste en forzar que el término b sea directamente cero siempre que $\phi(\mathbf{0}_m) = \mathbf{0}_q$. Esto garantiza que $\hat{f}(\mathbf{0}_m) = \mathbf{w}\phi(\mathbf{0}_m) + b = 0$. De este modo, en este punto, el modelo SVR adaptado sería el siguiente:

$$\begin{aligned} \underset{\mathbf{w}, \xi_i}{Min} \quad & \|\mathbf{w}\| + C \sum_{i=1}^n \xi_i & (.0) \\ & y_i - \mathbf{w}\phi(\mathbf{x}_i) \leq 0, \quad i = 1, \dots, n & (.1) \\ & \mathbf{w}\phi(\mathbf{x}_i) - y_i \leq \varepsilon + \xi_i, \quad i = 1, \dots, n & (.2) \\ & \xi_i \geq 0, \quad i = 1, \dots, n & (.3) \end{aligned} \tag{15}$$

En cuanto a la forma del estimador, cabe mencionar que los enfoques estándar de medición de la eficiencia como FDH y DEA recurren a ciertos tipos particulares de superficies. FDH utiliza funciones escalonadas y DEA recurre a la convexificación de la función escalonada de FDH para obtener un estimador cóncavo de la función de producción. Aunque, en este sentido, el SVM estándar es flexible, permitiendo utilizar varias familias de funciones de transformación complejas (kernels polinómicos, radiales, laplacianos, entre otros), nos centramos en una función escalonada como función de transformación $\phi(\cdot)$, buscando establecer un puente entre los enfoques estándar de

medición de la eficiencia técnica y el método SVR. De esta forma, y por analogía con FDH y DEA, se aplicó una función de transformación de tal forma que nuestro estimador fuera una función escalonada, que posteriormente podría transformarse en una función cóncava por convexificación del conjunto de posibilidades de producción. Para ello, nos basamos en una aproximación basada en *splines* del estimador construidos a partir de un número finito de nudos definida por Vapnik (1998, p. 464).

Las definiciones y notaciones necesarias para la función de transformación $\phi_{SVF}(\cdot)$ que se definieron son las siguientes. Para cada dimensión de entrada j , $j=1, \dots, m$, el conjunto de nudos se define como $T_j = \{t_{l_j}^{(j)} : l_j = 1, \dots, k_j\}$ satisfaciendo que $0 < t_1^{(j)} < t_2^{(j)} < \dots < t_{k_j}^{(j)}$, $t_{l_j}^{(j)} \in \mathbb{R}$ $l_j = 1, \dots, k_j$. Los subconjuntos disjuntos $C_{1\dots 1}, \dots, C_{k_1 \dots k_m}$ tales que, $C_{l_1 \dots l_m} = \{x \in \mathbb{R}_+^m : t_{l_j}^{(j)} \leq x^{(j)} < t_{l_j+1}^{(j)}, j = 1, \dots, m\}$ $l_1 \in \{1, \dots, k_1\}, \dots, l_m \in \{1, \dots, k_m\}$, con $t_{k_j+1}^{(j)} := \infty, \forall j = 1, \dots, m$, definen una rejilla (grid) G en \mathbb{R}_+^m . Cada subconjunto $C_{l_1 \dots l_m}$ se denomina celda de la rejilla G . Véase la Figura 9, donde se muestra un ejemplo de rejilla y sus celdas. Para cada celda de una rejilla G , $a_{l_1 \dots l_m} = (t_{l_1}^{(1)}, \dots, t_{l_m}^{(m)})$ y $b_{l_1 \dots l_m} = (t_{l_1+1}^{(1)}, \dots, t_{l_m+1}^{(m)})$ son el punto de nudo extremo inferior y el punto de nudo extremo superior, respectivamente. Así se tiene que $C_{l_1 \dots l_m} = \{x \in \mathbb{R}_+^m : a_{l_1 \dots l_m}^{(j)} \leq x^{(j)} < b_{l_1 \dots l_m}^{(j)}, j = 1, \dots, m\}$. Ahora, dado un vector de entrada $x \in \mathbb{R}_+^m$, tenemos que identificar la celda en la que se encuentra. La siguiente función de “activación” binaria definida para cada celda indicará con un valor de uno si $x \in C_{l_1 \dots l_m}$:

$$x \rightarrow L_{l_1 \dots l_m}(x) = \prod_{j=1}^m B(x^{(j)} - t_{l_j}^{(j)}), \text{ donde}$$

$$B(x^{(j)} - t_{l_j}^{(j)}) = \begin{cases} 1, & \text{if } x^{(j)} - t_{l_j}^{(j)} \geq 0 \\ 0, & \text{if } x^{(j)} - t_{l_j}^{(j)} < 0 \end{cases}, \forall l_j = 1, \dots, k_j, \forall j = 1, \dots, m \quad (16)$$

Obsérvese también que cuando se activa una celda $C_{l_1 \dots l_m}$, entonces también se activan todas las celdas $C_{s_1 \dots s_m}$ con, $s_j \leq l_j, j = 1, \dots, m$, es decir, si $L_{l_1 \dots l_m}(x) = 1$, entonces $L_{s_1 \dots s_m}(x) = 1, \forall s_j \leq l_j, j = 1, \dots, m$. Diremos que todas estas celdas son celdas dominadas

por $C_{l_1 \dots l_m}$. La Figura 9 muestra cómo funciona la activación de celdas en una rejilla de \mathbb{R}_+^m para dos dimensiones de entrada.

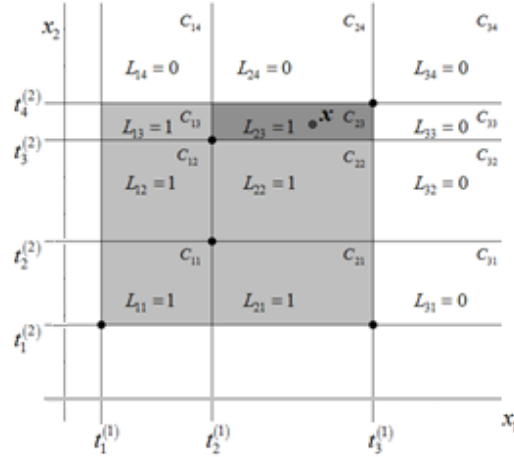


Figura 9: Ejemplo de rejilla

En este contexto, dada una rejilla G , definimos la función de transformación ϕ_{SVF}^G , que se utilizará en nuestro enfoque, como el mapeo del espacio \mathbb{R}_+^m dado por

$$\mathbf{x} \rightarrow \phi_{SVF}^G(\mathbf{x}) = (L_{11 \dots 11}(\mathbf{x}), L_{11 \dots 12}(\mathbf{x}), \dots, L_{11 \dots 1k_m}(\mathbf{x}), L_{11 \dots 21}(\mathbf{x}), L_{11 \dots 22}(\mathbf{x}), \dots, L_{11 \dots 2k_m}(\mathbf{x}), \dots, L_{\tilde{k}_1 \tilde{k}_2 \dots \tilde{k}_m}(\mathbf{x})) \quad (17)$$

Para cada vector de entrada $\mathbf{x} \in \mathbb{R}_+^m$, $\phi_{SVF}^G(\mathbf{x})$ corresponde a un vector de unos y ceros generado a partir de las funciones de activación $k_1 \cdot k_2 \cdot \dots \cdot k_m$ en la rejilla G . La transformación $\phi_{SVF}^G(\mathbf{x})$ asigna el valor de uno para la celda donde se encuentra \mathbf{x} y todas sus correspondientes celdas dominadas. De este modo, en este punto, el modelo SVR adaptado para estimar las funciones de producción sería el siguiente:

$$\begin{aligned} \underset{w, \xi_i}{Min} \quad & \|w\| + C \sum_{i=1}^n \xi_i & (0) \\ & y_i - w \phi_{SVF}^G(\mathbf{x}_i) \leq 0, \quad i=1, \dots, n & (1) \\ & w \phi_{SVF}^G(\mathbf{x}_i) - y_i \leq \varepsilon + \xi_i, \quad i=1, \dots, n & (2) \\ & \xi_i \geq 0, \quad i=1, \dots, n & (3) \end{aligned} \quad (18)$$

El estimador vinculado a (18) sería $\hat{f}(\mathbf{x}) = \mathbf{w} \phi_{SVF}^G(\mathbf{x})$. Obsérvese que $\hat{f}(\mathbf{0}_m) = 0$ ya que $\phi_{SVF}^G(\mathbf{0}_m) = \mathbf{0}_q$, con $q = k_1 \cdot k_2 \cdot \dots \cdot k_m$. De este modo se garantiza la satisfacción del axioma A1. Además, por (17), si $\mathbf{x} \in C_{l_1 \dots l_m}$, entonces tenemos que

$$\hat{f}(\mathbf{x}) = \mathbf{w} \phi_{SVF}^G(\mathbf{x}) = \sum_{\substack{s_1=1, \dots, l_1 \\ \vdots \\ s_m=1, \dots, l_m}} w_{s_1 \dots s_m}.$$

Además, cabe señalar que el número de componentes del vector de parámetros (\mathbf{w}) coincide con el número total de celdas de la rejilla correspondiente G . Explícitamente, el vector de parámetros \mathbf{w} puede escribirse de la siguiente manera.

$$\mathbf{w} = \left(w_{11 \dots 11}, w_{11 \dots 12}, \dots, w_{11 \dots 1k_m}, w_{11 \dots 21}, w_{11 \dots 22}, \dots, w_{11 \dots 2k_m}, \dots, w_{k_1 k_2 \dots k_m} \right) \quad (19)$$

Cada componente de \mathbf{w} está vinculado a una celda de la rejilla y el orden de los componentes es el mismo que el seguido por el vector $\phi_{SVF}^G(\mathbf{x})$ en (17). Esta relación permite introducir la noción de rejilla parametrizada: una rejilla G a la que se asigna un parámetro $w_{l_1 \dots l_m} \in \mathbb{R}$ para cada celda, $C_{l_1 \dots l_m}$, $l_j = 1, \dots, k_j$, $j = 1, \dots, m$, es una rejilla parametrizada con parámetros \mathbf{w} .

A continuación, vamos a demostrar que el estimador derivado del modelo (18) es una función escalonada y, además, cualquier función escalonada $f(\mathbf{x})$ definida sobre una rejilla G puede reescribirse equivalentemente del modo $f(\mathbf{x}) = \mathbf{w} \phi_{SVF}^G(\mathbf{x})$ para un determinado vector paramétrico \mathbf{w} . No obstante, primero debemos introducir la definición de función escalonada en una rejilla y la noción de función recuperable a través de una rejilla parametrizada: una función $f: \mathbb{R}_+^m \rightarrow \mathbb{R}$ es una función escalonada en una rejilla G si $f(\mathbf{x})$ es constante, $\forall \mathbf{x} \in C_{l_1 \dots l_m}$, $\forall l_1 = 1, \dots, k_1$, ..., $\forall l_m = 1, \dots, k_m$. El valor de la función f para cualquier $\mathbf{x} \in C_{l_1 \dots l_m}$ se denotará como $f(C_{l_1 \dots l_m})$. Además, una función $f: \mathbb{R}_+^m \rightarrow \mathbb{R}$ es una función recuperable a través de una rejilla parametrizada G con parámetros \mathbf{w} si y sólo si $\forall \mathbf{x} \in \mathbb{R}_+^m$, $f(\mathbf{x}) = \mathbf{w} \phi_{SVF}^G(\mathbf{x}) = \sum_{\substack{s_1=1, \dots, l_1 \\ \vdots \\ s_m=1, \dots, l_m}} w_{s_1 \dots s_m}$, con $\mathbf{x} \in C_{l_1 \dots l_m}$.

Dada una rejilla G , es evidente que $\hat{f}(\mathbf{x}) = \mathbf{w}\phi_{SVF}^G(\mathbf{x}) = \sum_{\substack{s_1=1, \dots, l_1 \\ \vdots \\ s_m=1, \dots, l_m}} w_{s_1 \dots s_m}$ es constante para

cada $\mathbf{x} \in C_{l_1 \dots l_m}$. En consecuencia, el estimador derivado de (18) es una función escalonada definida en G . Además, puede demostrarse (Proposición 1) que, dada una función escalonada $f(\mathbf{x})$ definida en una rejilla G , siempre habrá parámetros \mathbf{w} tales que $f(\mathbf{x})$ pueda expresarse de forma equivalente como $f(\mathbf{x}) = \mathbf{w}\phi_{SVF}^G(\mathbf{x}) = \sum_{\substack{s_1=1, \dots, l_1 \\ \vdots \\ s_m=1, \dots, l_m}} w_{s_1 \dots s_m}$.

Proposición 1. Cualquier función escalonada sobre una rejilla G es una función recuperable a través de la rejilla parametrizada G para un cierto vector paramétrico único \mathbf{w} .

Demostración. Sea $f: \mathbb{R}_+^m \rightarrow \mathbb{R}$ una función escalonada sobre una rejilla G . Tenemos que $f(x) = f(C_{l_1 \dots l_m})$, $\forall x \in C_{l_1 \dots l_m}$, $\forall l_1 = 1, \dots, k_1, \dots, \forall l_m = 1, \dots, k_m$. Ahora, tenemos que averiguar si el siguiente sistema tiene una solución o no.

$$\sum_{\substack{s_1=1, \dots, l_1 \\ \vdots \\ s_m=1, \dots, l_m}} w_{s_1 \dots s_m} = f(C_{l_1 \dots l_m}), l_1 = 1, \dots, k_1, \dots, l_m = 1, \dots, k_m \quad (20)$$

La expresión en (20) representa $k_1 \cdot \dots \cdot k_m$ ecuaciones. Las ecuaciones correspondientes a $l_1 = l_2 = \dots = l_{m-1} = 1$ y $l_m = 1, \dots, k_m$ son, en formato matricial, las siguientes.

$$\underbrace{\begin{pmatrix} 1 & 0 & & 0 \\ 1 & 1 & & 0 \\ \vdots & \vdots & \ddots & \\ 1 & 1 & \dots & 1 \end{pmatrix}}_{A_{1 \dots l_m}} \underbrace{\begin{pmatrix} w_{1 \dots 11} \\ w_{1 \dots 12} \\ \vdots \\ w_{1 \dots 1k_m} \end{pmatrix}}_{\mathbf{w}_{1 \dots l_m}} = \underbrace{\begin{pmatrix} f(C_{1 \dots 11}) \\ f(C_{1 \dots 12}) \\ \vdots \\ f(C_{1 \dots 1k_m}) \end{pmatrix}}_{F(C_{1 \dots l_m})}, \quad (21)$$

con $|A_{1 \dots l_m}| = 1$ porque $A_{1 \dots l_m}$ es una matriz triangular inferior (el determinante de una matriz triangular es el producto de los elementos de su diagonal). En cuanto al sistema de ecuaciones para $l_1 = l_2 = \dots = l_{m-2} = 1$, $l_{m-1} = 1, \dots, k_{m-1}$ y $l_m = 1, \dots, k_m$, tenemos el siguiente formato:

$$\begin{pmatrix}
\underbrace{\begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \dots & 1 \end{pmatrix}}_{A_{1\dots 1l_m}} & O & \dots & O \\
\underbrace{\begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \dots & 1 \end{pmatrix}}_{A_{1\dots 1l_m}} & \underbrace{\begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \dots & 1 \end{pmatrix}}_{A_{1\dots 2l_m}} & \dots & O \\
\vdots & \vdots & \ddots & \vdots \\
\underbrace{\begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \dots & 1 \end{pmatrix}}_{A_{1\dots 1l_m}} & \underbrace{\begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \dots & 1 \end{pmatrix}}_{A_{1\dots 2l_m}} & \dots & \underbrace{\begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \dots & 1 \end{pmatrix}}_{A_{1\dots k_{m-1}l_m}} \\
\hline
& & & \underbrace{\begin{pmatrix} w_{1\dots 11} \\ w_{1\dots 12} \\ \vdots \\ w_{1\dots 1k_m} \\ w_{1\dots 21} \\ w_{1\dots 22} \\ \vdots \\ w_{1\dots 2k_m} \\ \vdots \\ w_{1\dots k_{m-1}1} \\ w_{1\dots k_{m-1}2} \\ \vdots \\ w_{1\dots k_{m-1}k_m} \end{pmatrix}}_{\tilde{W}_{1\dots l_{m-1}l_m}}
\end{pmatrix} = \begin{pmatrix} f(C_{1\dots 11}) \\ f(C_{1\dots 12}) \\ \vdots \\ f(C_{1\dots 1k_m}) \\ f(C_{1\dots 21}) \\ f(C_{1\dots 22}) \\ \vdots \\ f(C_{1\dots 2k_m}) \\ \vdots \\ f(C_{1\dots k_{m-1}1}) \\ f(C_{1\dots k_{m-1}2}) \\ \vdots \\ f(C_{1\dots k_{m-1}k_m}) \end{pmatrix}, \\
\hline
& & & \underbrace{F(C_{1\dots l_{m-1}l_m})}_{F(C_{1\dots l_{m-1}l_m})}
\end{pmatrix},$$

con $|A_{1\dots l_{m-1}l_m}| = \prod_{s_{m-1}=1, \dots, k_{m-1}} |A_{1\dots 1s_{m-1}l_m}| = 1$ porque $A_{1\dots l_{m-1}l_m}$ es una matriz triangular inferior por

bloques, y el determinante de ese tipo de matrices es igual al producto de los determinantes de los bloques de su diagonal. Secuencialmente, podemos escribir todas las ecuaciones de $k_1 \dots k_{m-1} \cdot k_m$ del sistema dado en (20) en formato matricial como

$A_{l_1 \dots l_{m-1}l_m} \cdot \tilde{W}_{l_1 \dots l_{m-1}l_m} = F(C_{l_1 \dots l_{m-1}l_m})$ con $|A_{l_1 \dots l_{m-1}l_m}| = \prod_{s_1=1, \dots, k_1} |A_{s_1 l_2 \dots l_{m-1}l_m}| = 1$ porque $A_{l_1 \dots l_{m-1}l_m}$ es una

matriz triangular inferior por bloques. Dado que $|A_{l_1 \dots l_{m-1}l_m}| = 1 \neq 0$, tenemos que $A_{l_1 \dots l_{m-1}l_m}$ es invertible. Por lo tanto, el sistema dado en (20) es compatible y determinado (tiene solución única). ■

Este resultado muestra la flexibilidad de la transformación sugerida en este trabajo, al menos, con respecto a cualquier función escalonada que pueda definirse en una rejilla del espacio de entradas. Más adelante, esta proposición será la clave para demostrar que la función escalonada asociada a la técnica FDH es siempre una función escalonada factible vinculada al nuevo enfoque, siempre que recurramos a una rejilla “empírica” específica.

En este punto, el estimador derivado del modelo (18) es capaz de generar funciones escalonadas que envuelven la nube de datos desde arriba y satisfacen el axioma A1. Sin embargo, el modelo puede producir estimaciones no monotónicas, que no son válidas cuando el objetivo es determinar una función (de producción) que cumpla el axioma A2. Además, la concavidad del estimador no está asegurada, lo que contradice el axioma A3. Para ello, se propusieron las soluciones que se presentan en la siguiente subsección.

4.1.1.2. CUMPLIENDO LOS AXIOMAS DE MONOTONICIDAD Y CONCAVIDAD DE LA FUNCIÓN DE PRODUCCIÓN

La monotonicidad y la concavidad son dos axiomas básicos de las funciones de producción en microeconomía. En consecuencia, en esta sección tratamos de dotar al estimador derivado del nuevo enfoque basado en SVR de la satisfacción de estas dos propiedades. La primera parte de esta subsección se dedica a definir las condiciones necesarias para asegurar una función escalonada monótona no decreciente como estimador a partir del enfoque SVF. En la parte final de esta subsección, mostraremos cómo determinar trivialmente funciones cóncavas como estimadores de las funciones de producción aplicando directamente la convexificación del conjunto de posibilidades de producción obtenido a partir de la función escalonada previamente definida.

Comenzamos estableciendo las condiciones que aseguran que el estimador final derivado de SVF cumple la monotonicidad para cualquier función recuperable definida sobre una rejilla paramétrica G . Para ello, primero necesitamos un lema técnico.

Lema 1. Sea G una rejilla y sean $\mathbf{x} \in C_{h_1 \dots h_m}$ y $\mathbf{z} \in C_{l_1 \dots l_m}$ con $\mathbf{x} \leq \mathbf{z}$. Entonces, $h_j \leq l_j, \forall j = 1, \dots, m$.

Demostración. Supongamos que $\mathbf{x} \in C_{h_1 \dots h_m}$ y $\mathbf{z} \in C_{l_1 \dots l_m}$ con $\mathbf{x} \leq \mathbf{z}$. Supongamos que $\exists j' \in \{1, \dots, m\}$ con $h_{j'} > l_{j'}$. Por la definición de celda en una rejilla G , tenemos que $t_{h_{j'}}^{(j')} \leq x^{(j')} < t_{h_{j'}+1}^{(j')}$ y $t_{l_{j'}}^{(j')} \leq z^{(j')} < t_{l_{j'}+1}^{(j')}$. Pero obsérvese que si $h_{j'} > l_{j'}$, por la definición de nudo en una rejilla G , entonces $t_{h_{j'}}^{(j')} \geq t_{l_{j'}+1}^{(j')}$. Por lo tanto, $x^{(j')} \geq t_{l_{j'}+1}^{(j')}$ y $x^{(j')} \geq t_{l_{j'}+1}^{(j')}$. De este modo, $x^{(j')} > z^{(j')}$ ya que $z^{(j')} < t_{l_{j'}+1}^{(j')}$, lo cual es una contradicción con la hipótesis $\mathbf{x} \leq \mathbf{z}$.

■

Antes de mostrar el resultado sobre la monotonicidad, introduzcamos la siguiente

$$\text{notación: } W_{l_1 \dots l_m} := \sum_{\substack{s_1=1, \dots, l_1 \\ \vdots \\ s_m=1, \dots, l_m}} w_{s_1 \dots s_m}.$$

Teorema 1. Sea $f : \mathbb{R}_+^m \rightarrow \mathbb{R}_+$ una función recuperable a través de la rejilla parametrizada G . Entonces, f es una función monótona no decreciente si y sólo si $W_{l_1 \dots l_m} \geq W_{h_1 \dots h_m}, \forall h_j \leq l_j, \forall j = 1, \dots, m$, con $h_j, l_j = 1, \dots, k_j$.

Demostración. (i) Para $f : \mathbb{R}_+^m \rightarrow \mathbb{R}_+$, una función recuperable a través de la rejilla parametrizada G , comenzamos probando que $W_{l_1 \dots l_m} \geq W_{h_1 \dots h_m}, \forall l_1 = 1, \dots, k_1, \dots, \forall l_m = 1, \dots, k_m, \forall h_1 \leq l_1, \dots, \forall h_m \leq l_m$ implica que f es una función monótona no decreciente. Sean $\mathbf{x} \in C_{h_1 \dots h_m}$ y $\mathbf{z} \in C_{l_1 \dots l_m}$ con $\mathbf{x} \leq \mathbf{z}$. Necesitamos probar que $f(\mathbf{x}) \leq f(\mathbf{z})$. Por el Lema 1, tenemos que $h_j \leq l_j, \forall j = 1, \dots, m$.

Además, por hipótesis, f es una función recuperable a través de la rejilla parametrizada G con parámetros \mathbf{w} . De este modo, $f(\mathbf{x}) = \sum_{\substack{s_1=1, \dots, h_1 \\ \vdots \\ s_m=1, \dots, h_m}} w_{s_1 \dots s_m} = W_{h_1 \dots h_m}$ para $\mathbf{x} \in C_{h_1 \dots h_m}$ y

$$f(\mathbf{z}) = \sum_{\substack{s_1=1, \dots, l_1 \\ \vdots \\ s_m=1, \dots, l_m}} w_{s_1 \dots s_m} = W_{l_1 \dots l_m} \quad \text{para} \quad \mathbf{z} \in C_{l_1 \dots l_m}. \quad \text{Finalmente,}$$

$$W_{l_1 \dots l_m} \geq W_{h_1 \dots h_m}, \quad \forall l_1 = 1, \dots, k_1, \dots, \forall l_m = 1, \dots, k_m, \forall h_1 \leq l_1, \dots, \forall h_m \leq l_m \quad \text{implica} \quad \text{que} \\ f(\mathbf{x}) \leq f(\mathbf{z}).$$

(ii) Sea f una función monótona no decreciente. Tenemos que demostrar que $W_{l_1 \dots l_m} \geq W_{h_1 \dots h_m}, \forall l_1 = 1, \dots, k_1, \dots, \forall l_m = 1, \dots, k_m, \forall h_1 \leq l_1, \dots, \forall h_m \leq l_m$. Consideremos, sin pérdida de generalidad, $l_1 \dots l_m$ y $h_1 \dots h_m$ con $h_j \leq l_j, \forall j = 1, \dots, m$. Probemos que $a_{h_1 \dots h_m} \leq a_{l_1 \dots l_m}$. Supongamos que $\exists j' \in \{1, \dots, m\}$ tal que $a_{h_1 \dots h_m}^{(j')} > a_{l_1 \dots l_m}^{(j')}$. Por la definición de celda en una rejilla, tenemos que $t_{h_{j'}}^{(j')} = a_{h_1 \dots h_m}^{(j')} \leq x^{(j')} \forall \mathbf{x} \in C_{h_1 \dots h_m}$ y $t_{l_{j'}}^{(j')} = a_{l_1 \dots l_m}^{(j')} \leq z^{(j')} \forall \mathbf{z} \in C_{l_1 \dots l_m}$. Por lo tanto, $t_{h_{j'}}^{(j')} > t_{l_{j'}}^{(j')}$. Pero la última desigualdad sólo es posible si $h_{j'} > l_{j'}$, lo cual es una contradicción con el hecho de que $h_j \leq l_j, \forall j = 1, \dots, m$. Entonces, $a_{h_1 \dots h_m} \leq a_{l_1 \dots l_m}$ con $h_j \leq l_j, \forall j = 1, \dots, m$. Además, $a_{h_1 \dots h_m} \in C_{h_1 \dots h_m}$, $a_{l_1 \dots l_m} \in C_{l_1 \dots l_m}$ y f es una función recuperable a través de la rejilla parametrizada G con parámetros \mathbf{w} , lo que significa que $f(a_{h_1 \dots h_m}) = W_{h_1 \dots h_m}$ y $f(a_{l_1 \dots l_m}) = W_{l_1 \dots l_m}$. Por último, dado que f es una

función monótona no decreciente y $a_{h_1 \dots h_m} \leq a_{l_1 \dots l_m}$, podemos concluir que $f(a_{h_1 \dots h_m}) \leq f(a_{l_1 \dots l_m})$, es decir, $W_{h_1 \dots h_m} \leq W_{l_1 \dots l_m}$. ■

Es posible enunciar un resultado similar, pero utilizando un número de restricciones menor que en el Teorema 1. Esta afirmación se establece formalmente en la Proposición 2.

Proposición 2. El sistema de desigualdades $W_{h_1 \dots h_m} \leq W_{l_1 \dots l_m}$, $\forall h_j \leq l_j, \forall j = 1, \dots, m$, con $h_j, l_j = 1, \dots, k_j$ es equivalente al sistema de desigualdades, $W_{l_1 l_2 \dots s_j \dots l_m} \leq W_{l_1 l_2 \dots l_m}$, $\forall s_j = l_j - 1, \forall l_j = 1, \dots, k_j, \forall j = 1, \dots, m$.

Demostración. (i) $W_{h_1 \dots h_m} \leq W_{l_1 \dots l_m}, \forall h_j \leq l_j, \forall j = 1, \dots, m$, con $h_j, l_j = 1, \dots, k_j$, implica trivialmente que $W_{l_1 l_2 \dots s_j \dots l_m} \leq W_{l_1 l_2 \dots l_m} \forall s_j = l_j - 1, \forall l_j = 1, \dots, k_j, \forall j = 1, \dots, m$.

(ii) Supongamos que $W_{l_1 l_2 \dots s_j \dots l_m} \leq W_{l_1 l_2 \dots l_m}, \forall s_j = l_j - 1, \forall l_j = 1, \dots, k_j, \forall j = 1, \dots, m$. Supongamos entonces que $h_j, l_j \in \{1, \dots, k_j\}$ con $h_j \leq l_j, \forall j = 1, \dots, m$. Entonces, tenemos que $W_{l_1 l_2 \dots l_m} \geq W_{l_1 - 1 l_2 \dots l_m} \geq W_{l_1 - 2 l_2 \dots l_m} \geq \dots \geq W_{h_1 l_2 \dots l_m} \geq W_{h_1 l_2 - 1 \dots l_m} \geq W_{h_1 l_2 - 2 \dots l_m} \geq \dots \geq W_{h_1 h_2 \dots l_m} \geq W_{h_1 h_2 \dots l_{m-1}} \geq W_{h_1 h_2 \dots l_{m-2}} \geq \dots \geq W_{h_1 \dots h_m}$. ■

Una vez identificadas las condiciones que caracterizan la monotonidad para las funciones escalonadas definidas sobre una rejilla parametrizada, estamos en condiciones de introducir el modelo de optimización final correspondiente a la nueva técnica de estimación de las funciones de producción, las llamadas Fronteras de Soporte Vectorial o Support Vector Frontiers (SVF):

$$\text{Min}_{\mathbf{w}, \xi_i} \|\mathbf{w}\|_1 + C \sum_{i=1}^n \xi_i \quad (22.0)$$

$$\text{s.t. } y_i - \mathbf{w} \phi_{SVF}^G(\mathbf{x}_i) \leq 0, \quad i = 1, \dots, n \quad (22.1)$$

$$\mathbf{w} \phi_{SVF}^G(\mathbf{x}_i) - y_i \leq \varepsilon + \xi_i, \quad i = 1, \dots, n \quad (22.2)$$

$$W_{h_1 l_2 \dots s_j \dots l_m} \leq W_{h_1 l_2 \dots l_m}, \quad \forall l_1, \dots, l_m, \forall s_j = l_j - 1, \forall j = 1, \dots, m \quad (22.3)$$

$$\xi_i \geq 0, \quad i = 1, \dots, n \quad (22.4)$$

Las restricciones (22.1)-(22.4) corresponden a las restricciones del modelo (18) con el nuevo conjunto de restricciones (22.3) para garantizar la monotonidad. En cuanto a la función objetivo, (22.0) coincide con (10.0) pero, en el modelo final (22) se elige una

norma específica. Aunque la norma euclídea es la métrica más habitual cuando se utiliza la SVM estándar, existen otras posibilidades en la literatura (véase, por ejemplo, [Blanco et al. \(2020\)](#)). En particular, en este trabajo consideramos la norma L1 porque uno de nuestros objetivos es comparar el nuevo enfoque con FDH y DEA estándar, que se basan en programación lineal. Obsérvese que, al asumir la norma L1, y dado que

$$\mathbf{w}\phi_{SVF}^G(\mathbf{x}_i) = \sum_{\substack{s_1=1,\dots,l_1 \\ \vdots \\ s_m=1,\dots,l_m}} w_{s_1\dots s_m} \text{ con } \mathbf{x}_i \in C_{l_1\dots l_m}, \text{ el modelo (22) es un programa lineal. Dada una}$$

solución óptima de (22), $(\mathbf{w}^*, \xi_1^*, \dots, \xi_n^*)$, el estimador de la frontera de producción

$$\text{vinculado a SVF se define como } f_{SVF}(\mathbf{x}) = \mathbf{w}^* \phi_{SVF}^G(\mathbf{x}), \text{ que es igual a } \sum_{\substack{s_1=1,\dots,l_1 \\ \vdots \\ s_m=1,\dots,l_m}} w_{s_1\dots s_m}^* \text{ si}$$

$\mathbf{x} \in C_{l_1\dots l_m}$. Por la discusión anterior, $f_{SVF}(\mathbf{x})$ es una función monótona escalonada no

decreciente que envuelve los datos de aprendizaje. Para conseguir además una función cóncava, que en teoría de la producción se traduce en tecnologías convexas, basta con

convexificar el estimador $f_{SVF}(\mathbf{x})$. Una estrategia similar se sigue para obtener un

conjunto de posibilidades de producción convexo en DEA a partir de una frontera de

producción escalonada estimada por FDH. Para ello, en el caso de SVF, debemos

identificar un determinado conjunto de puntos extremos. En concreto, cada celda $C_{l_1\dots l_m}$

de la rejilla genera uno de estos puntos: uno con el vector de entrada $\mathbf{a}_{l_1\dots l_m}$, el nudo-punto

extremo inferior de la celda $C_{l_1\dots l_m}$, y la salida equivale a la estimación de la variable de

respuesta para $\mathbf{a}_{l_1\dots l_m}$, es decir, $f_{SVF}(\mathbf{a}_{l_1\dots l_m})$. Dado el conjunto de puntos “virtuales” de

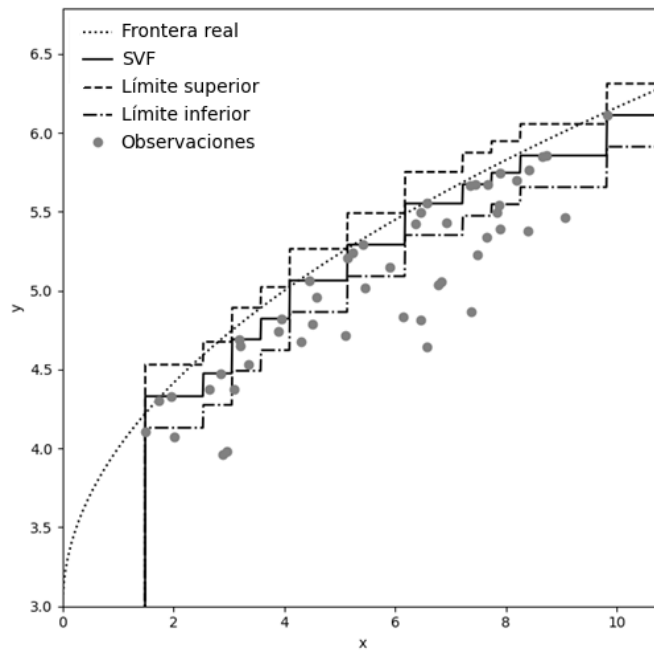
entrada-salida $\left\{ \left(\mathbf{a}_{l_1\dots l_m}, f_{SVF}(\mathbf{a}_{l_1\dots l_m}) \right) \right\}_{\substack{l_1=1,\dots,k_1 \\ \vdots \\ l_m=1,\dots,k_m}}$, puede determinarse un conjunto convexo de

posibilidades de producción aplicando una estimación de tipo DEA:

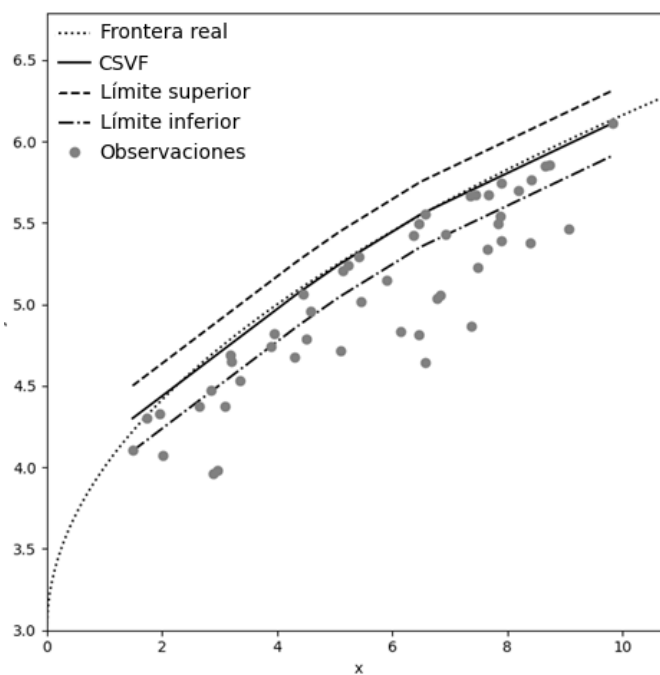
$$\hat{T}_{CSVF} = \left\{ \begin{array}{l} (\mathbf{x}, \mathbf{y}) \in \mathbb{R}_+^{m+1} : \mathbf{y} \leq \sum_{\substack{l_1=1,\dots,k_1 \\ \vdots \\ l_m=1,\dots,k_m}} \lambda_{l_1\dots l_m} f_{SVF}(\mathbf{a}_{l_1\dots l_m}), \mathbf{x}^{(j)} \geq \\ \sum_{\substack{l_1=1,\dots,k_1 \\ \vdots \\ l_m=1,\dots,k_m}} \lambda_{l_1\dots l_m} \mathbf{a}_{l_1\dots l_m}^{(j)}, \forall j, \sum_{\substack{l_1=1,\dots,k_1 \\ \vdots \\ l_m=1,\dots,k_m}} \lambda_{l_1\dots l_m} = 1, \lambda_{l_1\dots l_m} \geq 0, \forall l_1, \dots, \forall l_m \end{array} \right\}, \quad (23)$$

donde las siglas CSVF significan “Convex Support Vector Frontiers”.

La Figura 10a muestra un ejemplo gráfico de estimación de una función de producción a partir de una muestra de aprendizaje con una entrada y una salida. La línea continua muestra la función escalonada vinculada al estimador derivado directamente de la solución del modelo (22). Se trata de una función monótona no decreciente que envuelve los datos. A diferencia de FDH, SVF no considera el principio de mínima extrapolación y, en consecuencia, el estimador SVF podría situarse estrictamente por encima del estimador FDH. Por un lado, el hiperparámetro C de la función objetivo del modelo de programación (22), que se ajustará mediante un proceso de validación cruzada como en la SVM estándar, es el valor que ayuda a situar la frontera más o menos por encima de los datos. Valores mayores de C dan más importancia a las observaciones y se asocian a una frontera estimada situada más cerca de los datos. Por otra parte, el hiperparámetro ε del modelo, que también se ajustará mediante validación cruzada, permite considerar cierto grado de robustez en el análisis. Las superficies de “corrección” superior e inferior, definidas como $f_{SVF}(\mathbf{x}) + \varepsilon$ y $f_{SVF}(\mathbf{x}) - \varepsilon$, respectivamente, también se ilustran en la figura mediante líneas discontinuas. Las observaciones entre las superficies de corrección superior e inferior tienen un error empírico de cero. Además, en la Figura 10b, mostramos la convexificación del estimador SVF. La Figura 10b muestra una función de producción estimada cóncava, lo que implica que el área por debajo de esta frontera define un conjunto convexo. Asimismo, se muestra las superficies de corrección superior e inferior para el ajuste convexo, que se determina de forma natural por convexificación a partir de $f_{SVF}(\mathbf{x}) + \varepsilon$ y $f_{SVF}(\mathbf{x}) - \varepsilon$ en la Figura 10a, respectivamente.



(a)



(b)

Figura 10: Ejemplo de estimación de SVF (a) y CSVF (b)

La Figura 10 y la noción de margen ε en SVM estándar inspiran una noción nueva y más robusta de eficiencia técnica en la teoría de la producción. Nos referimos a la Definición

1, donde, en particular, se introduce el concepto de eficiencia técnica insensible al margen ε .

Definición 1. La DMU (\mathbf{x}_i, y_i) es técnicamente eficiente e insensible a ε si y sólo si $f_{SVF}(\mathbf{x}_i) - \varepsilon \leq y_i \leq f_{SVF}(\mathbf{x}_i) + \varepsilon$.

Obsérvese que, por supuesto, el conjunto de unidades técnicamente eficientes insensibles a ε incluye las DMU técnicamente eficientes tradicionales, es decir, las situadas en la frontera de producción (estimada) f_{SVF} (véase Figura 11). A grandes rasgos, el modelo no discrimina, en cuanto al grado de eficiencia técnica, entre las unidades situadas entre el estimador y las superficies de corrección superior e inferior. Para el enfoque SVF, todas son igual de eficientes desde el punto de vista técnico.¹ Además, para cualquier DMU (\mathbf{x}_i, y_i) que pertenezca a la muestra de aprendizaje, tenemos que es ε -insensiblemente eficiente técnicamente si y sólo si $f_{SVF}(\mathbf{x}_i) - \varepsilon \leq y_i \leq f_{SVF}(\mathbf{x}_i)$, ya que $y_i \leq f_{SVF}(\mathbf{x}_i)$. Este no es el caso de las unidades que no pertenecen a la muestra de aprendizaje. Por ejemplo, en el caso del análisis intertemporal, los cambios tecnológicos a lo largo del tiempo pueden dar lugar a que unidades pertenecientes a un determinado periodo se sitúen fuera de la tecnología correspondiente a un periodo de tiempo anterior o posterior.

¹ La eficiencia técnica insensible a ε está relacionada de algún modo con la noción de eficiencia de algunos modelos DEA que tratan los conceptos de imprecisión e incertidumbre, como el DEA difuso (Hatami-Marbini et al. (2011)) o el DEA estocástico (Olesen y Petersen (2016)), por citar solo algunos.

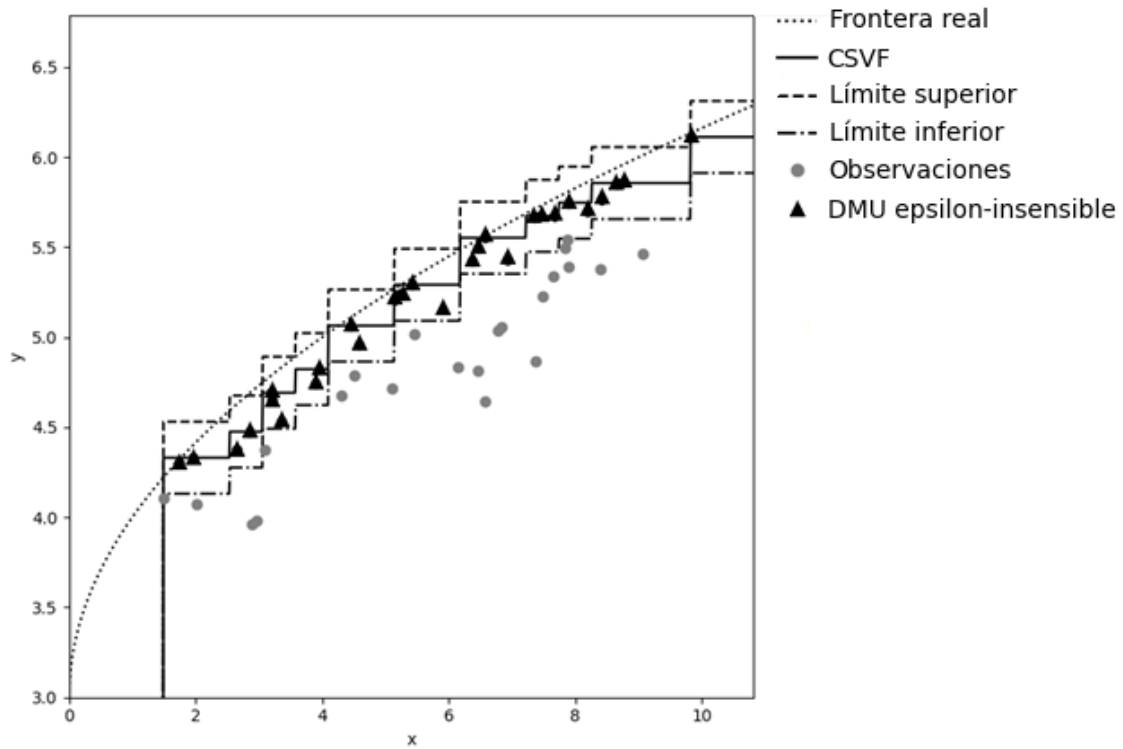


Figura 11: Ejemplo de DMUs ε -insensibles bajo SVF

Por último, para terminar esta subsección, cabe mencionar que el modelo (22) necesita la determinación previa de ciertos parámetros. Tanto C como ε se mencionaron anteriormente en el texto. Sin embargo, no son los únicos parámetros a determinar. Al igual que en la SVM estándar, la función de transformación (o, en su lugar, el kernel asociado) depende de otro conjunto de parámetros. En nuestro contexto, la función de transformación SVF ϕ_{SVF}^G está vinculada a la determinación de un conjunto de nudos para cada dimensión de entrada considerada en el problema. En nuestro estudio, aplicamos un método de búsqueda simplificado para determinar estos puntos, ya que su identificación podría resultar una tarea difícil desde el punto de vista computacional si se deja libre su número y posición. En su lugar, el parámetro clave que se determinará mediante validación cruzada será d , el número de celdas que se definirán para cada dimensión de entrada. En un contexto empírico, observamos un valor máximo y mínimo en cada variable de entrada, lo que permite calcular el rango y dividirlo en d intervalos con la misma amplitud.

4.1.1.3. FDH Y DEA COMO UN CASO PARTICULAR DE SVF

El Free Disposal Hull (FDH) y el Análisis Envoltente de Datos (DEA) son enfoques no paramétricos bien conocidos para el análisis de la eficiencia técnica. En esta subsección, mostramos que FDH y DEA pueden interpretarse alternativamente como una SVR sujeto a restricciones de forma (monotonicidad y concavidad) que genera superficies envolventes de la muestra de aprendizaje. Esta reinterpretación revela la naturaleza de FDH y DEA como parte de las técnicas de aprendizaje automático. En particular, demostramos que estas dos técnicas estándar son siempre soluciones factibles del modelo de optimización Support Vector Frontiers, siendo óptimas en el caso más simple de trabajar con una única entrada y una única salida.

El modelo no paramétrico como FDH es especialmente atractivo, ya que no se basa en hipótesis restrictivas sobre el proceso generador de datos, una característica compartida con las técnicas habituales de aprendizaje automático, que son enfoques claramente basados en datos. En particular, [Deprins et al. \(1984\)](#) propusieron el FDH del conjunto de observaciones (DMUs) para estimar T , definido como sigue: $\hat{T}_{FDH} = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}_+^{m+s} : \mathbf{y} \leq \mathbf{y}_i, \mathbf{x} \geq \mathbf{x}_i, i = 1, \dots, n\}$. En el caso de producción univariante, la función de producción se estimaría mediante $f_{FDH}(\mathbf{x}) = \max_{i: \mathbf{x} \geq \mathbf{x}_i} \{y_i\}$, que es una función escalonada. La técnica FDH es atractiva porque se basa en pocos supuestos, pero, por construcción, sufre sobreajuste debido al principio de extrapolación mínima. Este principio obliga a que la función escalonada no decreciente se aproxime lo máximo posible a la nube de datos, mientras que envuelve todas las observaciones desde arriba.

A diferencia del FDH, el DEA asume además la convexidad. Esto significa que si (\mathbf{x}, \mathbf{y}) y $(\mathbf{x}', \mathbf{y}')$ pertenecen a T , entonces $\lambda(\mathbf{x}, \mathbf{y}) + (1 - \lambda)(\mathbf{x}', \mathbf{y}') \in T$, para todos $\lambda \in [0, 1]$. Banker et al. (1984) propusieron el estimador DEA del conjunto de posibilidades de producción T como sigue:

$$\hat{T}_{DEA} = \left\{ (\mathbf{x}, \mathbf{y}) \in \mathbb{R}_+^{m+s} : \mathbf{y}^{(r)} \leq \sum_{i=1}^n \lambda_i \mathbf{y}_i^{(r)}, \forall r, \mathbf{x}^{(j)} \geq \sum_{i=1}^n \lambda_i \mathbf{x}_i^{(j)}, \forall j, \sum_{i=1}^n \lambda_i = 1, \lambda_i \geq 0, \forall i \right\}.$$

En un contexto de salida única, el estimador DEA de una función de producción es una función lineal a trozos que envuelve los datos desde arriba. La convexidad del conjunto de posibilidades de producción implica la concavidad de la función de producción. Asimismo, la convexificación del estimador FDH da como resultado el estimador DEA (Daraio y Simar (2005)). DEA sufre el mismo problema de sobreajuste que FDH. De nuevo, el principio de extrapolación mínima obliga a determinar una descripción “perfecta” de la muestra observada a nivel de frontera

A lo largo de esta sección, centraremos nuestra atención en FDH, alcanzando resultados similares con respecto a DEA mediante convexificación directa. Comenzamos mostrando que la función escalonada obtenida aplicando la técnica FDH es siempre factible en el modelo SVF (22) cuando se considera una rejilla específica basada en las observaciones. Nos referimos a la rejilla “empírica”, que utiliza las observaciones como nodos para cada dimensión de entrada y que se introducirá formalmente en esta subsección. Antes, introduzcamos una nueva notación. Sea $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ una muestra de aprendizaje con $\mathbf{x}_i \in \mathbb{R}_+^m$ y $y_i \in \mathbb{R}_+$ $\forall i = 1, \dots, n$. Entonces, el conjunto de elementos $\{\tilde{x}_l^{(j)}\}_{l=1}^{n^{(j)}}$, para cada $j = 1, \dots, m$, es el conjunto de valores ordenados de las observaciones una vez eliminadas las repeticiones en cada dimensión. En consecuencia, $n^{(j)} \leq n$ y $\tilde{x}_1^{(j)} < \dots < \tilde{x}_{n^{(j)}}^{(j)}$. De este modo, la rejilla empírica G^E es la rejilla definida a partir de los siguientes conjuntos de nodos, $T_j^E = \{\tilde{x}_l^{(j)}, l_j = 1, \dots, n^{(j)} + 1\}$ $j = 1, \dots, m$, con $\tilde{x}_{n^{(j)}+1}^{(j)} := \infty$ $\forall j = 1, \dots, m$. Las celdas de la rejilla empírica se denotarán por $C_{l_1 \dots l_m}^E$ y los nudos-puntos extremos de cada una como $a_{l_1 \dots l_m}^E$ y $b_{l_1 \dots l_m}^E$. Otra definición necesaria es la que introduce la noción de conjunto de observaciones dominadas. En particular, dado un vector de entrada $\mathbf{x} \in \mathbb{R}_+^m$, el conjunto de observaciones dominadas se define como $X = \{i \in \{1, \dots, n\} : \mathbf{x}_i \leq \mathbf{x}\}$.

A continuación, vamos a demostrar que el estimador generado por FDH es una función escalonada definida en la rejilla empírica G^E . Este resultado es clave para probar que el estimador FDH es siempre factible del modelo SVF. Para demostrarlo, primero necesitamos enunciar algunos resultados técnicos previos.

Lema 2. Sean $i \in \{1, \dots, n\}$ y $\mathbf{x}_i \leq \mathbf{a}_{l_1 \dots l_m}^E$. Entonces, $\mathbf{x}_i \leq \mathbf{x}$, $\forall \mathbf{x} \in C_{l_1 \dots l_m}^E$.

Demostración. Sea $\mathbf{x} \in C_{l_1 \dots l_m}^E$. Entonces $\mathbf{a}_{l_1 \dots l_m}^E \leq \mathbf{x}$ porque $C_{l_1 \dots l_m}^E = \{\mathbf{x} \in \mathbb{R}_+^m : \mathbf{a}_{l_1 \dots l_m}^E \leq \mathbf{x} < \mathbf{b}_{l_1 \dots l_m}^E\}$. Por hipótesis, tenemos que $\mathbf{x}_i \leq \mathbf{a}_{l_1 \dots l_m}^E$. Por lo tanto, $\mathbf{x}_i \leq \mathbf{a}_{l_1 \dots l_m}^E \leq \mathbf{x}$. ■

Corolario 1. Sea $\mathbf{x} \in C_{l_1 \dots l_m}^E$. Entonces $A_{l_1 \dots l_m}^E := \{i \in \{1, \dots, n\} : \mathbf{x}_i \leq \mathbf{a}_{l_1 \dots l_m}^E\} \subset X$.

Demostración. El resultado es consecuencia directa del Lema 2. ■

Lema 3. Sean $i \in \{1, \dots, n\}$ y $\mathbf{x} \in C_{l_1 \dots l_m}^E$. Si $\mathbf{x}_i \leq \mathbf{x}$, entonces $\mathbf{x}_i \leq \mathbf{a}_{l_1 \dots l_m}^E$.

Demostración. Sean $\mathbf{x} \in C_{l_1 \dots l_m}^E$ y $\mathbf{x}_i \leq \mathbf{x}$ para un $i \in \{1, \dots, n\}$. Supongamos que no es cierto que $\mathbf{x}_i \leq \mathbf{a}_{l_1 \dots l_m}^E$. Entonces $\exists j' \in \{1, \dots, m\}$ tal que $x_i^{(j')} > a_{l_1 \dots l_m}^{E(j')}$. Por la definición de $C_{l_1 \dots l_m}^E = \{\mathbf{x} \in \mathbb{R}_+^m : \mathbf{a}_{l_1 \dots l_m}^E \leq \mathbf{x} < \mathbf{b}_{l_1 \dots l_m}^E\}$, tenemos que $a_{l_1 \dots l_m}^{E(j')} < x_i^{(j')} \leq x^{(j')} < b_{l_1 \dots l_m}^{E(j')}$. Por lo tanto, existen $\tilde{x}_{l_{j'}}^{(j')}$ y $\tilde{x}_{l_{j'+1}}^{(j')}$, elementos sucesivos en el conjunto ordenado de valores $\{\tilde{\mathbf{x}}_{l_j}^{(j')}\}_{l_j=1}^{n^{(j)}}$ tales que $\tilde{x}_{l_{j'}}^{(j')} < x_i^{(j')} < \tilde{x}_{l_{j'+1}}^{(j')}$, con $\tilde{x}_{l_{j'}}^{(j')} = a_{l_1 \dots l_m}^{E(j')}$ y $\tilde{x}_{l_{j'+1}}^{(j')} = b_{l_1 \dots l_m}^{E(j')}$. Sin embargo, esto es una contradicción con el hecho de que $x_i^{(j')} \in \{\tilde{\mathbf{x}}_{l_j}^{(j')}\}_{l_j=1}^{n^{(j)}}$. En consecuencia, $\mathbf{x}_i \leq \mathbf{a}_{l_1 \dots l_m}^E$. ■

Corolario 2. Sea $\mathbf{x} \in C_{l_1 \dots l_m}^E$. Entonces $X \subset A_{l_1 \dots l_m}^E$.

Demostración. El resultado es consecuencia directa del Lema 3. ■

Proposición 3. Sea $\mathbf{x} \in C_{l_1 \dots l_m}^E$. Entonces $X = A_{l_1 \dots l_m}^E$.

Demostración. Este resultado es consecuencia de los Corolarios 1 y 2. ■

El siguiente teorema demuestra que el estimador asociado a la técnica FDH es una función escalonada definida en la rejilla empírica. Nótese que es bien conocido en la literatura

que la metodología FDH siempre genera funciones escalonadas. Sin embargo, también necesitamos demostrar que se trata de una función escalonada definida en una rejilla. En este caso, identificamos dicha rejilla con la empírica.

Teorema 2. El estimador obtenido mediante la técnica FDH es una función escalonada definida en la rejilla G^E . En particular, $f_{FDH}(\mathbf{x}) = f_{FDH}(\mathbf{a}_{l_1 \dots l_m}^E)$, $\forall \mathbf{x} \in C_{l_1 \dots l_m}^E$.

Demostración. Si $\mathbf{x} \in C_{l_1 \dots l_m}^E$, entonces

$$f_{FDH}(\mathbf{x}) = \max_{i: \mathbf{x} \geq \mathbf{x}_i} \{y_i\} = \max_{i \in X} \{y_i\} = \max_{i \in A_{l_1 \dots l_m}^E} \{y_i\} = \max_{i: \mathbf{x}_i \leq \mathbf{a}_{l_1 \dots l_m}^E} \{y_i\} = f_{FDH}(\mathbf{a}_{l_1 \dots l_m}^E),$$

donde la tercera igualdad es cierta por la Proposición 3, la última igualdad por la monotonicidad de la función FDH y las otras igualdades por definición. ■

Por el Teorema 2 y la Proposición 1, sabemos que existe un vector \mathbf{w}^{FDH} tal que el estimador asociado a la técnica FDH puede expresarse equivalentemente como $f_{FDH}(\mathbf{x}) = \mathbf{w}^{FDH} \phi_{SVF}^{G^E}(\mathbf{x})$. Además, f_{FDH} es una función recuperable a través de la rejilla parametrizada G^E con parámetros \mathbf{w}^{FDH} . Además, si definimos los errores empíricos como

$$\xi_i^{FDH} := \begin{cases} (f_{FDH}(\mathbf{x}) - \varepsilon) - y_i, & \text{if } (f_{FDH}(\mathbf{x}) - \varepsilon) - y_i \geq 0 \\ 0, & \text{if } (f_{FDH}(\mathbf{x}) - \varepsilon) - y_i < 0 \end{cases}, \forall i = 1, \dots, n, \quad (24)$$

entonces podemos afirmar que FDH siempre proporciona una solución factible del modelo (22). La expresión (24) puede escribirse de forma compacta como

$$\xi_i^{FDH} = \max\{(f_{FDH}(\mathbf{x}) - \varepsilon) - y_i, 0\}.$$

Teorema 3. $(\mathbf{w}^{FDH}, \xi_1^{FDH}, \dots, \xi_n^{FDH})$ es una solución factible del modelo (22).

Demostración. Las condiciones (22.1) se cumplen trivialmente porque $\mathbf{w}^{FDH} \phi_{SVF}^{G^E}(\mathbf{x}) = f_{FDH}(\mathbf{x})$ es una función envolvente de los datos. Las restricciones (22.2) y (22.4) se cumplen gracias a la definición de los errores empíricos ξ_i^{FDH} , $\forall i = 1, \dots, n$.

Además, f_{FDH} es una función recuperable a través de la rejilla parametrizada G^E con parámetros w^{FDH} . Entonces, dado que es bien sabido que f_{FDH} es una función monótona no decreciente, invocando el Teorema 1 y la Proposición 2, las condiciones (22.3) son satisfechas por el vector de parámetros w^{FDH} . ■

En consecuencia, la función escalonada generada por la técnica FDH estándar siempre puede recuperarse a partir del nuevo enfoque, la denominada técnica de Support Vector Frontiers. La función de producción estimada mediante la técnica FDH clásica siempre se considera mediante el modelo (22) dado cualquier parámetro C y ε , como un posible estimador SVF. Otra cosa es que la función escalonada FDH fuera óptima. Esto no siempre es cierto. No obstante, podemos demostrar que, para cualesquiera que sean los parámetros C y ε , en el marco más sencillo, es decir, cuando el número de variables predictoras es igual a uno, el estimador FDH siempre es óptimo del modelo (22) cuando se considera la rejilla empírica. Será el Teorema 4. Antes de demostrarlo, estableceremos un lema útil.

Lema 4. Sea $m = 1$ y sea $(w', \xi'_1, \dots, \xi'_n)$ una solución factible del modelo (22) cuando se considera la rejilla empírica. Entonces, $w' \phi_{SVF}^{G^E} \left(\max_{1 \leq i \leq n} \{x_i\} \right) \geq \max \{y_1, \dots, y_n\}$.

Demostración. Sea $i' = 1, \dots, n$ tal que $y_{i'} = \max \{y_1, \dots, y_n\}$. Entonces, por (22.1),

$w' \phi_{SVF}^{G^E}(x_{i'}) \geq y_{i'}$. Por (22.3), $h(x) := w' \phi_{SVF}^{G^E}(x)$ es una función monótona no decreciente.

Por lo tanto, $\max_{1 \leq i \leq n} \{x_i\} \geq x_{i'}$ implica $w' \phi_{SVF}^{G^E} \left(\max_{1 \leq i \leq n} \{x_i\} \right) \geq w' \phi_{SVF}^{G^E}(x_{i'}) \geq y_{i'} = \max \{y_1, \dots, y_n\}$.

■

Teorema 4. Sea $m = 1$. Entonces, $(w^{FDH}, \xi_1^{FDH}, \dots, \xi_n^{FDH}) \geq 0$ es una solución óptima del modelo (22).

Demostración. Sea $(w', \xi'_1, \dots, \xi'_n)$ cualquier solución factible del modelo (22). Por (22.3),

$w'_1 \leq w'_1 + w'_2$, $w'_1 + w'_2 \leq w'_1 + w'_2 + w'_3$, ..., $w'_1 + w'_2 + w'_3 + \dots + w'_{n^{(0)}-1} \leq w'_1 + w'_2 + w'_3 + \dots + w'_{n^{(0)}}$.

Por lo tanto, $w'_l \geq 0$, $l = 2, \dots, n^{(1)}$. Sea $i', i'' = 1, \dots, n$ tal que, $x_{i'} = \min_{1 \leq i \leq n} \{x_i\}$ y $x_{i''} = \max_{1 \leq i \leq n} \{x_i\}$. Por (22.1), $\mathbf{w}' \phi_{SVF}^{GE}(x_{i'}) \geq y_{i'}$, que es equivalente a $w'_1 \geq y_{i'} \geq 0$. En cuanto a la función objetivo en (22)

$$\|\mathbf{w}'\|_1 + C \sum_{i=1}^n \xi'_i = w'_1 + \dots + w'_{n^{(1)}} + C \sum_{i=1}^n \xi'_i = \mathbf{w}' \phi_{SVF}^{GE}(x_{i'}) + C \sum_{i=1}^n \xi'_i.$$

Además, es bien sabido que $f_{FDH}(x_{i'}) = \max\{y_1, \dots, y_n\}$. Entonces, $\mathbf{w}^{FDH} \phi_{SVF}^{GE}(x_{i'}) = f_{FDH}(x_{i'}) = \max\{y_1, \dots, y_n\}$, que es, por el Lema 4, el valor mínimo que puede tomar $\mathbf{w}' \phi_{SVF}^{GE}(x_{i'})$ para cualquier solución factible $(\mathbf{w}', \xi'_1, \dots, \xi'_n)$ de (22). Además, dada cualquier solución factible, $(\mathbf{w}', \xi'_1, \dots, \xi'_n)$ $\xi'_i = \max\left\{\left(\mathbf{w}' \phi_{SVF}^{GE}(x_i) - \varepsilon\right) - y_i, 0\right\}$ porque estamos minimizando (22.0), que incluye el término $C \sum_{i=1}^n \xi'_i$, y también gracias a las restricciones (22.2). Por último, nótese que, por el principio de extrapolación mínima, $f_{FDH}(x_i) = \mathbf{w}^{FDH} \phi_{SVF}^{GE}(x_i) \leq \mathbf{w}' \phi_{SVF}^{GE}(x_i)$, $\forall i = 1, \dots, n$, que es equivalente a $\left(\mathbf{w}^{FDH} \phi_{SVF}^{GE}(x_i) - \varepsilon\right) - y_i \leq \left(\mathbf{w}' \phi_{SVF}^{GE}(x_i) - \varepsilon\right) - y_i$, $\forall i = 1, \dots, n$. Estas últimas desigualdades implican que $\xi_i^{FDH} \leq \xi'_i$, $i = 1, \dots, n$, para cualquier solución factible $(\mathbf{w}', \xi'_1, \dots, \xi'_n)$. En consecuencia, $\mathbf{w}^{FDH} \phi_{SVF}^{GE}(x_{i'}) + C \sum_{i=1}^n \xi_i^{FDH} \leq \mathbf{w}' \phi_{SVF}^{GE}(x_{i'}) + C \sum_{i=1}^n \xi'_i$ para cualquier solución factible $(\mathbf{w}', \xi'_1, \dots, \xi'_n)$, lo que significa que $(\mathbf{w}^{FDH}, \xi_1^{FDH}, \dots, \xi_n^{FDH})$ es una solución óptima del modelo (22). ■

Sin embargo, el Teorema 4 no puede extenderse al marco general, es decir, cuando el número de entradas es arbitrario. Considerando el marco empírico, un posible contraejemplo estaría relacionado con la siguiente muestra de aprendizaje: $(x_1^A, x_2^A, y^A) = (1, 4, 2)$, $(x_1^B, x_2^B, y^B) = (2, 2, 1)$ y $(x_1^C, x_2^C, y^C) = (3, 1, 3)$.

Mediante la convexificación, y teniendo en cuenta que la frontera DEA estándar puede derivarse de la función escalonada FDH aplicando esta misma técnica, todos los resultados anteriores pueden extenderse trivialmente a la estimación de tecnologías de producción convexas. De este modo, la estimación DEA tradicional de la función de

producción es siempre una de las posibles soluciones asociadas a la técnica de CSVF. Además, en el contexto de un input y un output, la estimación DEA se obtiene a partir de la solución óptima de las CSVF tras aplicar la convexificación de la función escalonada SVF, independientemente de los valores de los hiperparámetros C y ε .

4.1.2. EL CASO DE MÚLTIPLES VARIABLES RESPUESTA

En esta parte introducimos la adaptación al contexto de múltiples variables respuesta (outputs) del método explicado en la anterior sección, SVF. Esta adaptación se basa en [Vázquez y Walter \(2003\)](#), que ampliaron SVR considerando el método denominado Cokriging, que es una versión multi-output de Kriging que explota las correlaciones debidas a la proximidad en el espacio de factores y outputs. En el modelo (25) mostramos los cambios introducidos en el modelo (22) para ampliar el marco de una única variable respuesta al contexto de producción de múltiples respuestas.

$$\text{Min}_{\mathbf{w}^r, \xi_i^r} \sum_{r=1}^s \|\mathbf{w}^{(r)}\|_1 + C \sum_{r=1}^s \sum_{i=1}^n \xi_i^{(r)} \quad (0)$$

$$\text{s.t.} \quad \mathbf{w}^{(r)} \phi_{SVF}^G(\mathbf{x}_i) - y_i^{(r)} \geq 0, \quad i=1, \dots, n, r=1, \dots, s \quad (1)$$

$$\mathbf{w}^{(r)} \phi_{SVF}^G(\mathbf{x}_i) - y_i^{(r)} \leq \varepsilon + \xi_i^{(r)}, \quad i=1, \dots, n, r=1, \dots, s \quad (2) \quad (25)$$

$$\mathbb{W}_{h_1 \dots s_j \dots l_m}^{(r)} \leq \mathbb{W}_{h_1 \dots l_m}^{(r)}, \quad \forall l_1, \dots, l_m / l_j \in \{1, \dots, k_j\} \forall j \in \{1, \dots, m\} \quad (3)$$

$$\forall s_j = l_j - 1, \forall j = 1, \dots, m, \forall r = 1, \dots, s$$

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

En el modelo (25) existe un vector $\mathbf{w}^{(r)}$ asociado a cada dimensión de salida r , $r=1, \dots, s$. Cada vector $\mathbf{w}^{(r)}$ tiene una componente para cada celda de la rejilla G , $w_{l_1 \dots l_m}^{(r)}$, definida en el espacio de entradas. También se definieron nuevas variables de desviación al margen $\xi_i^{(r)}$, para cada observación i en cada dimensión de salida r . En cuanto a las restricciones, son las mismas que para el modelo SVF de salida única, pero replicadas para cada dimensión de salida, donde $\mathbb{W}_{l_1 \dots l_m}^{(r)} := \sum_{s_1=1, \dots, l_1} w_{s_1 \dots s_m}^{(r)}, \forall r=1, \dots, s$. El mejor modelo

se obtendrá por validación cruzada como en el caso univariante. Una vez obtenido el mejor modelo, la predicción se define como, $\hat{\mathbf{y}}_{SVF}^{(r)}(\mathbf{x}) = \mathbf{w}^{*(r)} \phi_{SVF}^G(\mathbf{x})$, $r=1, \dots, s$. De esta

forma, definimos el estimador de la tecnología para el contexto de producción multi-salida multi-entrada como sigue:

$$\hat{T}_{SVF} := \{(\mathbf{x}, \mathbf{y}) \in \mathcal{R}_+^{m+s} : \mathbf{y} \leq \hat{\mathbf{y}}_{SVF}(\mathbf{x})\} \quad (26)$$

El estimador satisface los axiomas de envoltura de los datos y libre disposición, como demostramos a continuación. Antes es necesario enunciar un resultado técnico.

Lema 5. Sea $\mathbf{x}' \geq \mathbf{x}$, entonces $\hat{\mathbf{y}}_{SVF}(\mathbf{x}') \geq \hat{\mathbf{y}}_{SVF}(\mathbf{x})$.

Demostración. Por la forma en que definimos la estimación de cada componente del vector de salida, tenemos que si $\mathbf{x}' \geq \mathbf{x}$, entonces $\hat{\mathbf{y}}_{SVF}(\mathbf{x}') \geq \hat{\mathbf{y}}_{SVF}(\mathbf{x})$, debido a que $\hat{\mathbf{y}}_{SVF}(\mathbf{x}')$ es una función no decreciente (ver Teorema 1, de la [Sección 4.1.1.2](#)). ■

Proposición 4. $(\mathbf{x}_i, \mathbf{y}_i) \in \hat{T}_{SVF}$ para todo $i = 1, \dots, n$.

Demostración. Por la restricción (23.1), tenemos $\hat{\mathbf{y}}_{SVF}^{(r)}(\mathbf{x}_i) = \mathbf{w}^{*(r)} \phi_{SVF}^G(\mathbf{x}_i) \geq \mathbf{y}_i^{(r)}$.

Finalmente, por (26), tenemos que $(\mathbf{x}_i, \mathbf{y}_i) \in \hat{T}_{SVF}$ para todo $i = 1, \dots, n$. ■

Proposición 5. El conjunto \hat{T}_{SVF} cumple la libre disponibilidad.

Demostración. Sea $(\mathbf{x}, \mathbf{y}) \in \hat{T}_{SVF}$ y sea $(\mathbf{x}', \mathbf{y}') \in \mathcal{R}_+^{m+s}$ tales que $\mathbf{x}' \geq \mathbf{x}$ e $\mathbf{y}' \leq \mathbf{y}$. En primer lugar, $\mathbf{y} \leq \hat{\mathbf{y}}_{SVF}(\mathbf{x})$ se cumple ya que $(\mathbf{x}, \mathbf{y}) \in \hat{T}_{SVF}$. Segundo, por el Lema 5, si $\mathbf{x}' \geq \mathbf{x}$, entonces $\hat{\mathbf{y}}_{SVF}(\mathbf{x}') \geq \hat{\mathbf{y}}_{SVF}(\mathbf{x})$. Por lo tanto, $\mathbf{y}' \leq \mathbf{y} \leq \hat{\mathbf{y}}_{SVF}(\mathbf{x}) \leq \hat{\mathbf{y}}_{SVF}(\mathbf{x}')$, lo que implica que $(\mathbf{x}', \mathbf{y}') \in \hat{T}_{SVF}$ por (26). ■

Además, demostramos que el modelo SVF de salida mono-output es un caso particular del modelo SVF de salida múltiple.

Proposición 6. Si $s = 1$, entonces el modelo (25) es equivalente al modelo (22).

Demostración. Es evidente a partir de las formulaciones de los modelos (22) y (25). ■

Como ocurre con el modelo SVF de salida única, cada observación de la muestra pertenece a una celda $C_{l_1 \dots l_m}$ en una rejilla definida en el espacio de las entradas. Como hemos mencionado anteriormente, cada celda está asociada a un nudo-punto extremo inferior $(\mathbf{a}_{l_1 \dots l_m})$. Entonces, podemos demostrar que la tecnología estimada a partir del modelo SVF multi-output, es decir, \hat{T}_{SVF} puede reescribirse como una tecnología de tipo FDH definida a partir de los puntos “virtuales” (en el sentido de “no necesariamente observados”) de entrada-salida $\left\{ \left(\mathbf{a}_{l_1 \dots l_m}, \hat{\mathbf{y}}_{SVF}(\mathbf{a}_{l_1 \dots l_m}) \right) \right\}_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}}$. Este resultado se establece formalmente en la siguiente proposición.

Proposición 7. $\hat{T}_{SVF} = \{(\mathbf{x}, \mathbf{y}) \in R_+^{m+s} : \exists l_1, \dots, l_m, \text{ con } l_1 \in \{1, \dots, k_1\}, \dots, l_m \in \{1, \dots, k_m\} \text{ tal que}$

$$\mathbf{y} \leq \hat{\mathbf{y}}_{SVF}(\mathbf{a}_{l_1 \dots l_m}), \mathbf{x} \geq \mathbf{a}_{l_1 \dots l_m}\}. \quad (27)$$

Demostración. Usamos \hat{T}_{SVF}^{FDH} para denotar el lado derecho de (27). Sea $(\mathbf{x}', \mathbf{y}') \in \hat{T}_{SVF}$. Entonces, por definición, $\mathbf{y}' \leq \hat{\mathbf{y}}_{SVF}(\mathbf{x}')$. Si consideramos $\mathbf{x}' \in C_{l'_1 \dots l'_m}$, entonces $\hat{\mathbf{y}}_{SVF}(\mathbf{x}') = \hat{\mathbf{y}}_{SVF}(\mathbf{x})$ para todo $\mathbf{x} \in C_{l'_1 \dots l'_m}$. Además, tenemos que $\mathbf{a}_{l'_1 \dots l'_m} \leq \mathbf{x}'$. De este modo, tenemos que $\hat{\mathbf{y}}_{SVF}(\mathbf{a}_{l'_1 \dots l'_m}) = \hat{\mathbf{y}}_{SVF}(\mathbf{x}')$ y $\mathbf{x}' \geq \mathbf{a}_{l'_1 \dots l'_m}$. En consecuencia, $\exists l'_1, \dots, l'_m$, con $l'_1 \in \{1, \dots, k_1\}, \dots, l'_m \in \{1, \dots, k_m\}$ tal que $\mathbf{y}' \leq \hat{\mathbf{y}}_{SVF}(\mathbf{a}_{l'_1 \dots l'_m})$ y $\mathbf{x}' \geq \mathbf{a}_{l'_1 \dots l'_m}$. Entonces, $(\mathbf{x}', \mathbf{y}') \in \hat{T}_{SVF}^{FDH}$. Sea ahora $(\mathbf{x}', \mathbf{y}') \in \hat{T}_{SVF}^{FDH}$. Entonces $\exists l_1, \dots, l_m$, con $l_1 \in \{1, \dots, k_1\}, \dots, l_m \in \{1, \dots, k_m\}$ tal que $\mathbf{y}' \leq \hat{\mathbf{y}}_{SVF}(\mathbf{a}_{l_1 \dots l_m})$ y $\mathbf{x}' \geq \mathbf{a}_{l_1 \dots l_m}$. Por el Lema 5, $\hat{\mathbf{y}}_{SVF}(\mathbf{x}') \geq \hat{\mathbf{y}}_{SVF}(\mathbf{a}_{l_1 \dots l_m})$. Entonces tenemos $\mathbf{y}' \leq \hat{\mathbf{y}}_{SVF}(\mathbf{a}_{l_1 \dots l_m}) \leq \hat{\mathbf{y}}_{SVF}(\mathbf{x}')$, lo que implica que $(\mathbf{x}', \mathbf{y}') \in \hat{T}_{SVF}$ por definición de \hat{T}_{SVF} . ■

Obsérvese que el lado derecho de (27) coincide con la definición de una tecnología de tipo FDH, como (3) cuando el conjunto de observaciones $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ se sustituye por el

conjunto de puntos virtuales $\left\{ \left(\mathbf{a}_{l_1 \dots l_m}, \hat{\mathbf{y}}_{SVF} \left(\mathbf{a}_{l_1 \dots l_m} \right) \right) \right\}_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}}$. La última proposición tiene

una aplicación importante. En primer lugar, Free Disposal Hull (FDH) puede considerarse un caso particular del modelo SVF de múltiples salidas, cuando los nudos-puntos extremos inferiores y sus correspondientes estimaciones de salida vinculadas a SVF coinciden con los datos. No obstante, en general, el conjunto de posibilidades de producción estimado por FDH es siempre un subconjunto de la tecnología derivada de la técnica SVF. Se debe al hecho de que ambas tecnologías satisfacen la propiedad de envoltura de los datos y la libre disponibilidad, pero sólo la tecnología relacionada con FDH cumple la extrapolación mínima. En segundo lugar, el resultado también puede utilizarse para mostrar cómo determinar cualquier medida de eficiencia técnica utilizando \hat{T}_{SVF} . Por ejemplo, utilizando el paralelismo establecido entre la tecnología FDH y \hat{T}_{SVF} , la puntuación radial orientada a la producción puede determinarse mediante un problema de optimización como (28) donde, en las restricciones, los datos originales se sustituyen por puntos virtuales de entrada-salida:

$$\phi_{SVF}(\mathbf{x}_i, \mathbf{y}_i) = \max \phi \quad (.0)$$

s.t.

$$\sum_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}} \lambda_{l_1 \dots l_m} \mathbf{a}_{l_1 \dots l_m}^{(j)} \leq \mathbf{x}_i^{(j)}, \quad j = 1, \dots, m \quad (.1)$$

$$\sum_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}} \lambda_{l_1 \dots l_m} \hat{\mathbf{y}}_{SVF}^{(r)} \left(\mathbf{a}_{l_1 \dots l_m} \right) \geq \phi \mathbf{y}_i^{(r)} \quad r = 1, \dots, s \quad (.2) \quad (28)$$

$$\sum_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}} \lambda_{l_1 \dots l_m} = 1, \quad (.3)$$

$$\lambda_{l_1 \dots l_m} \in \{0, 1\}, \quad \begin{matrix} \forall l_j = 1, \dots, k_j, \\ \forall j = 1, \dots, m \end{matrix} \quad (.4)$$

Buscando robustez en nuestros resultados, también podemos adaptar la definición de ε -insensibilidad del contexto SVF de salida única al marco de salida múltiple incorporando el margen ε a las restricciones del modelo (28). En cuanto a la aplicación de la noción de eficiencia técnica insensible a ε , se trata de una definición de eficiencia

más robusta que la tradicional. En términos geométricos, el método construye una banda de puntos en el espacio de inputs-outputs que se utiliza para determinar si una unidad puede identificarse como técnicamente eficiente, es decir, situada dentro de la banda o no. En este marco, la ineficacia técnica puede definirse como la “distancia” del punto evaluado a esta banda.

En particular, cuando el objetivo es evaluar el nivel de eficiencia de una observación perteneciente a la muestra de datos, basta con incorporar la información asociada a la cota inferior de la estimación de los outputs. Esto se debe al hecho de que, para cualquier observación $(\mathbf{x}_i, \mathbf{y}_i)$, tenemos que $\hat{y}_{SVF}^{(r)}(\mathbf{a}_{l_1 \dots l_m}) \geq \hat{y}_{SVF}^{(r)}(\mathbf{x}_i) \geq y_i^{(r)}$, con $C_{l_1 \dots l_m}$ tal que $\mathbf{x}_i \in C_{l_1 \dots l_m}$. Y, en consecuencia, $\hat{y}_{SVF}^{(r)}(\mathbf{a}_{l_1 \dots l_m}) + \varepsilon \geq y_i^{(r)}$, con $\varepsilon \geq 0$, se cumple trivialmente. En este escenario, el modelo a resolver sería el siguiente:

$$\phi_{SVF-}(\mathbf{x}_i, \mathbf{y}_i) = \max_{s.t.} \phi^- \quad (0)$$

$$\sum_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}} \lambda_{l_1 \dots l_m} \mathbf{a}_{l_1 \dots l_m}^{(j)} \leq \mathbf{x}_i^{(j)}, \quad j=1, \dots, m \quad (1)$$

$$\sum_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}} \lambda_{l_1 \dots l_m} \left(\hat{y}_{SVF}^{(r)}(\mathbf{a}_{l_1 \dots l_m}) - \varepsilon^* \right) \geq \phi^- y_i^{(r)} \quad r=1, \dots, s \quad (2) \quad (29)$$

$$\sum_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}} \lambda_{l_1 \dots l_m} = 1, \quad (3)$$

$$\lambda_{l_1 \dots l_m} \in \{0, 1\}, \quad \begin{matrix} \forall l_j = 1, \dots, k_j, \\ \forall j = 1, \dots, m \end{matrix} \quad (4)$$

Para las observaciones, y dado el valor óptimo del modelo (29), denotado como $\phi_{SVF-}(\mathbf{x}_i, \mathbf{y}_i)$, tenemos dos posibilidades. Si $\phi_{SVF-}(\mathbf{x}_i, \mathbf{y}_i) > 1$, entonces la observación se sitúa por debajo del límite inferior y es ε -insensible técnicamente ineficiente. En caso contrario, es decir, si $\phi_{SVF-}(\mathbf{x}_i, \mathbf{y}_i) \leq 1$, la observación se sitúa entre los márgenes y, en consecuencia, el modelo la identifica como ε -insensiblemente ineficiente desde el punto de vista técnico.

En cambio, en el caso de evaluar una nueva observación (\mathbf{x}, \mathbf{y}) , tenemos que considerar tanto el límite superior como el inferior. De este modo, el modelo (29) debe utilizarse para evaluar (\mathbf{x}, \mathbf{y}) , obteniendo el valor $\phi_{SVF-}(\mathbf{x}, \mathbf{y})$. Además, el modelo (30) debe determinarse también con el fin de tener en cuenta la información vinculada al límite superior.

$$\phi_{SVF+}(\mathbf{x}_i, \mathbf{y}_i) = \max \phi^+ \quad (.0)$$

s.t.

$$\sum_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}} \lambda_{l_1, \dots, l_m} \mathbf{a}_{l_1, \dots, l_m}^{(j)} \leq \mathbf{x}_i^{(j)}, \quad j=1, \dots, m \quad (.1)$$

$$\sum_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}} \lambda_{l_1, \dots, l_m} \left(\hat{\mathbf{y}}_{SVF}^{(r)}(\mathbf{a}_{l_1, \dots, l_m}) + \varepsilon^* \right) \geq \phi^+ \mathbf{y}_i^{(r)} \quad r=1, \dots, s \quad (.2) \quad (30)$$

$$\sum_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}} \lambda_{l_1, \dots, l_m} = 1, \quad (.3)$$

$$\lambda_{l_1, \dots, l_m} \in \{0, 1\}, \quad \begin{array}{l} \forall l_j = 1, \dots, k_j, \\ \forall j = 1, \dots, m \end{array} \quad (.4)$$

En el caso de evaluar una nueva observación, si $\phi_{SVF-}(\mathbf{x}, \mathbf{y}) \leq 1$ y $\phi_{SVF+}(\mathbf{x}, \mathbf{y}) \geq 1$, entonces el modelo señala esta unidad como ε -insensible técnicamente eficiente ya que se encuentra entre los límites superior e inferior. Este tipo de situación puede darse cuando la unidad a evaluar no se encuentra en la muestra de aprendizaje. Por ejemplo, las mejoras tecnológicas a lo largo del tiempo pueden dar lugar a observaciones en el periodo de tiempo t que pueden estar situadas fuera de la tecnología correspondiente a un periodo de tiempo anterior t' .

Además, para proporcionar una estimación de la tecnología que satisfaga el supuesto de convexidad, podemos convexificar directamente \hat{T}_{SVF} , dando lugar a la extensión CSVF para tratar el contexto de producción multi-salida. De este modo, el conjunto de posibilidades de producción estimado por CSVF podría definirse como $\hat{T}_{CSVF} := \text{conv}(\hat{T}_{SVF})$. Como mostramos a continuación, invocando la Proposición 7, es

posible reescribir \hat{T}_{CSVF} en términos del conjunto virtual de puntos $\left\{ \left(\mathbf{a}_{l_1 \dots l_m}, \hat{\mathbf{y}}_{SVF} \left(\mathbf{a}_{l_1 \dots l_m} \right) \right) \right\}_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}}$, del mismo modo que la tecnología DEA puede obtenerse a partir de la convexificación de la tecnología FDH.

$$\hat{T}_{CSVF} = \left\{ \begin{array}{l} (\mathbf{x}, \mathbf{y}) \in \mathbb{R}_+^{m+s} : \mathbf{y}^{(r)} \leq \sum_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}} \lambda_{l_1 \dots l_m} \hat{\mathbf{y}}_{SVF}^{(r)} \left(\mathbf{a}_{l_1 \dots l_m} \right), \forall r = 1, \dots, s, \\ \mathbf{x}^{(j)} \geq \sum_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}} \lambda_{l_1 \dots l_m} \mathbf{a}_{l_1 \dots l_m}^{(j)}, \forall j = 1, \dots, m, \sum_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}} \lambda_{l_1 \dots l_m} = 1, \lambda_{l_1 \dots l_m} \geq 0, \forall l_1, \dots, l_m \end{array} \right\} \quad (31)$$

En cuanto al cumplimiento de los supuestos microeconómicos habituales, la convexidad se satisface trivialmente, mientras que la envoltura de los datos y la libre disponibilidad deben comprobarse.

Proposición 8. $(\mathbf{x}_i, \mathbf{y}_i) \in \hat{T}_{CSVF}$

Demostración. Dado que $\hat{T}_{SVF} \subset \hat{T}_{CSVF}$, por la Proposición 4, tenemos que $(\mathbf{x}_i, \mathbf{y}_i) \in \hat{T}_{CSVF}$. ■

Proposición 9. \hat{T}_{CSVF} cumple con la libre disponibilidad.

Demostración. Sea $(\mathbf{x}, \mathbf{y}) \in \hat{T}_{CSVF}$. Por (31) existen $\lambda_{l_1 \dots l_m} \geq 0$, $l_1 = 1, \dots, k_1, \dots, l_m = 1, \dots, k_m$,

, con $\sum_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}} \lambda_{l_1 \dots l_m} = 1$, tales que $\mathbf{y}^{(r)} \leq \sum_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}} \lambda_{l_1 \dots l_m} \hat{\mathbf{y}}_{SVF}^{(r)} \left(\mathbf{a}_{l_1 \dots l_m} \right)$, $\forall r = 1, \dots, s$,

$\mathbf{x}^{(j)} \geq \sum_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}} \lambda_{l_1 \dots l_m} \mathbf{a}_{l_1 \dots l_m}^{(j)}$, $\forall j = 1, \dots, m$. Sea ahora $(\mathbf{x}', \mathbf{y}') \in \mathbb{R}_+^{m+s}$, $\mathbf{x}' \geq \mathbf{x}$ con $\mathbf{y}' \leq \mathbf{y}$.

Entonces, $\mathbf{y}'^{(r)} \leq \mathbf{y}^{(r)} \leq \sum_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}} \lambda_{l_1 \dots l_m} \hat{\mathbf{y}}_{SVF}^{(r)} \left(\mathbf{a}_{l_1 \dots l_m} \right)$, $\forall r = 1, \dots, s$,

$x'^{(j)} \geq x^{(j)} \geq \sum_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}} \lambda_{l_1, \dots, l_m} a_{l_1, \dots, l_m}^{(j)}, \quad \forall j=1, \dots, m.$ Esto implica que $(\mathbf{x}', \mathbf{y}') \in \hat{T}_{CSVF}$, como

queríamos demostrar. ■

Como en el caso del SVF, dado un punto, el CSVF puede aplicarse para determinar la medida radial de eficiencia técnica orientada al output, así como sus versiones robustas, basadas en la noción de eficiencia insensible a ε , simplemente añadiendo la restricción

$$\sum_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}} \lambda_{l_1, \dots, l_m} = 1 \text{ a los programas (28), (29) y (30).}$$

Por último, cabe mencionar que el Análisis Envolvente de Datos puede considerarse un caso particular de la técnica CSVF cuando el conjunto de puntos virtuales

$$\left\{ \left(\mathbf{a}_{l_1, \dots, l_m}, \hat{\mathbf{y}}_{SVF} \left(\mathbf{a}_{l_1, \dots, l_m} \right) \right) \right\}_{\substack{l_1=1, \dots, k_1 \\ \vdots \\ l_m=1, \dots, k_m}} \text{ coincide con el conjunto de observaciones } \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n.$$

No obstante, en general, el conjunto de posibilidades de producción estimado por DEA es siempre un subconjunto de la tecnología derivada de CSVF. La razón de este resultado es que ambas tecnologías satisfacen la propiedad de envoltura superior de los datos y la libre disponibilidad, pero sólo la tecnología relacionada con DEA cumple el principio de extrapolación mínima, que, al mismo tiempo, relaciona DEA con el sobreajuste.

4.1.2.1. VERSIÓN SIMPLIFICADA DE LAS TÉCNICAS SVF Y CSVF

El modelo SVF multi-output (25) es un programa de optimización difícil de resolver debido a que consume mucho tiempo de ejecución y recursos computacionales. Esto se debe al enorme número de parámetros y restricciones que presenta cuando aumentan la dimensionalidad y el tamaño muestral del problema. En cuanto al número de restricciones asociadas al modelo (25) la siguiente tabla muestra cuántas restricciones están vinculadas a cada tipo de restricción en el planteamiento.

Restricción	Número de restricciones
25.1	sn

25.2	sn
25.3	$\approx md^m$

Tabla 1: Número de restricciones en el modelo (25)

La Tabla 1 muestra que la cantidad de restricciones en el modelo SVF aumenta exponencialmente a medida que aumenta el número de entradas, especialmente debido a la restricción 23.3, donde d representa el hiperparámetro asociado al número de particiones en que se divide cada dimensión de entrada a través de los nodos. En este sentido, proponemos un modelo alternativo, denominado enfoque de Simplified Support Vector Frontiers (SSVF), que intenta aliviar las dificultades computacionales asociadas al modelo SVF original de múltiples salidas. Para ello, sustituimos la restricción compleja 23.3 por la restricción más simple 30.3, de la siguiente manera:

$$\begin{aligned}
 \text{Min}_{\mathbf{w}, \xi_i} \quad & \sum_{r=1}^s \|\mathbf{w}^{(r)}\|_1 + C \sum_{r=1}^s \sum_{i=1}^n \xi_i^{(r)} \quad (.0) \\
 \text{s.t.} \quad & \mathbf{w}^{(r)} \phi_{SVF}^G(\mathbf{x}_i) - y_i^{(r)} \geq 0, \quad i = 1, \dots, n, r = 1, \dots, s \quad (.1) \\
 & \mathbf{w}^{(r)} \phi_{SVF}^G(\mathbf{x}_i) - y_i^{(r)} \leq \varepsilon + \xi_i^{(r)}, \quad i = 1, \dots, n, r = 1, \dots, s \quad (.2) \\
 & \mathbf{w}^{(r)} \geq 0, \quad r = 1, \dots, s \quad (.3) \\
 & \xi_i^{(r)} \geq 0, \quad i = 1, \dots, n, r = 1, \dots, s \quad (.4)
 \end{aligned} \tag{32}$$

A continuación, demostramos que $\mathbf{w}^{(r)} \geq 0$, para todo $r = 1, \dots, s$, es una condición suficiente para garantizar la satisfacción de la restricción 23.3.

Proposición 10. Si $\mathbf{w}^{(r)} \geq 0 \quad \forall r = 1, \dots, s$, entonces, $W_{l_1 l_2 \dots s_j \dots l_m}^{(r)} \leq W_{l_1 l_2 \dots l_m}^{(r)}, \quad \forall s_j = l_j - 1$
 $\forall j = 1, \dots, m, \quad \forall r = 1, \dots, s$.

Demostración. $\mathbf{w}^{(r)} \geq 0, \quad \forall r = 1, \dots, s$, implica que para cada $r = 1, \dots, s, \quad w_{h_1 \dots h_m}^{(r)} \geq 0$

$\forall h_j = 1, \dots, k_j \quad \text{y} \quad \forall j = 1, \dots, m$. De este modo,

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

$\forall l_j = 1, \dots, k_j, \quad \forall j = 1, \dots, m$; y, por tanto, por definición de $W_{l_1 \dots l_j - 1 \dots l_m}^{(r)}$, tenemos $W_{l_1 \dots l_j - 1 \dots l_m}^{(r)} \leq W_{l_1 \dots l_m}^{(r)}, \quad \forall l_j = 1, \dots, k_j, \quad \forall j = 1, \dots, m$. ■

Una vez determinado el mejor modelo mediante validación cruzada, la predicción del enfoque simplificado se define como $\hat{y}_{SSVF}^{(r)}(\mathbf{x}) = \mathbf{w}^{*(r)} \phi_{SVF}^G(\mathbf{x}), \quad r = 1, \dots, s$. Además, por analogía con las técnicas SVF y CSVF descritas en la subsección anterior, es posible definir versiones simplificadas de la estimación de la tecnología sin asumir convexidad y asumiendo convexidad, denotadas como $\hat{T}_{SSVF} := \{(\mathbf{x}, \mathbf{y}) \in R_+^{m+s} : \mathbf{y} \leq \hat{y}_{SSVF}(\mathbf{x})\}$ y $\hat{T}_{CSSVF} := \text{conv}(\hat{T}_{SSVF})$, respectivamente. Además, se puede derivar una estimación de la medida de eficiencia radial orientada a la producción, así como sus correspondientes versiones insensibles a ε . En la próxima sección, comprobaremos la precisión del enfoque simplificado comparándolo con el original mediante una experiencia computacional.

4.1.3. EXPERIENCIA COMPUTACIONAL

En esta sección se describen los resultados de la simulación que sirven para comparar los métodos: FDH vs SVF y DEA vs CSVF. En primer lugar, se analizan el rendimiento obtenido para el problema de una sola variable de tipo output. Seguidamente se expondrán los resultados conseguidos en escenarios simulados con varias variables outputs.

4.1.3.1. SIMULACIONES MONO-OUTPUT

En primer lugar, presentamos una comparación de los métodos en tres entornos simulados alternativos mono-output. Sus descripciones aparecen en la Tabla 2.

Escenario	Inputs	Forma funcional
I	x_1	$y = 3 + x_1^{0.5} - u$
II	x_1, x_2	$y = 3 + x_1^{0.2} + x_2^{0.3}$
III	x_1, x_2, x_3	$y = 3 + x_1^{0.05} + x_2^{0.15} + x_3^{0.3}$

Tabla 2. Descripción de los tres escenarios

El escenario I representa un caso de un solo input y los escenarios II y III representan casos de múltiples variables predictoras. Para todos ellos, probamos con conjuntos de datos de tamaños 20, 30, 40, 50, 60, 70, 80, 90 y 100. Los datos de entrada se obtuvieron aleatoriamente de una $Uni[1,10]$, independientemente para cada entrada y observación. A continuación, se calculó el valor del output eficiente y se restó un término de ineficiencia aleatorio $u \sim |N(0,0.4)|$ para obtener los datos utilizados para el análisis.

Realizamos 50 pruebas ($t = 1, \dots, 50$) para cada combinación de escenario y tamaño del conjunto de datos para investigar el rendimiento relativo de los métodos. Además, y debido a la naturaleza de SVF, al igual que ocurre con SVM y SVR, el mejor modelo debe seleccionarse mediante la determinación de la mejor combinación de hiperparámetros. Esto se llevó a cabo mediante un proceso de validación cruzada basado en cinco pliegues (folds, en inglés). En nuestro marco, los hiperparámetros son C , ε y d (el número de intervalos con la misma anchura que debe definirse para cada dimensión de entrada). En la práctica, es necesario establecer un número finito de combinaciones posibles de estos hiperparámetros, ya que el coste computacional de probar las infinitas combinaciones de hiperparámetros sería inmanejable. Arbitrariamente, fijamos los siguientes valores para cada hiperparámetro: $C \in \{0.1, 0.5, 1, 2, 5\}$,

$\varepsilon \in \{0, 0.001, 0.01, 0.1, 0.2\}$ y $d = 0.1 \cdot h \cdot n$ (redondeado al número entero más próximo), donde $h = 1, \dots, 10$. Estos valores generan una cantidad total de 250 combinaciones diferentes de hiperparámetros para el problema estudiado. Además, el rendimiento de cada método se evalúa mediante dos criterios estándar: el error cuadrático medio (MSE) y el sesgo. El estadístico MSE se define como

$$\sum_{t=1}^{50} \sum_{i=1}^n \left(f(\mathbf{x}_i^t) - \hat{f}(\mathbf{x}_i^t) \right)^2 / 50n, \quad \text{mientras que el sesgo se calcula como}$$

$$\sum_{t=1}^{50} \sum_{i=1}^n \left| f(\mathbf{x}_i^t) - \hat{f}(\mathbf{x}_i^t) \right| / 50n.$$

La Tabla 3 muestra la media y la desviación estándar (entre paréntesis) de los mejores hiperparámetros en nuestras simulaciones. Todos ellos se estimaron mediante validación cruzada, excepto en el escenario III, en el que, por cuestiones computacionales, fijamos

el hiperparámetro en $d = 20$. En cuanto a los resultados, el hiperparámetro C fue indiferente para el escenario con una entrada (escenario I), porque el error de generalización obtenido en la validación cruzada fue siempre el mismo. Sin embargo, el valor del hiperparámetro C fue relevante para los escenarios con más entradas, como el II y el III. Algo similar ocurrió con el valor del hiperparámetro ε en el escenario I con tamaños de muestra pequeños (20 y 30 observaciones). En cuanto al hiperparámetro d , determina el tamaño de la rejilla G (el número de celdas) y, en consecuencia, el número de variables de decisión y restricciones del modelo (22). La tendencia observada muestra que cuanto mayor es el número de observaciones, mayor es el valor del hiperparámetro óptimo correspondiente a d .

Escenario	Número observaciones	Hiperparámetros						
		C	ϵ		d			
		MEAN (STD)	MEAN (STD)	MIN	MAX	MEAN (STD)	MIN	MAX
I	20		indifferent	0.001	0.001	14.2(4.005)	6	20
	30			0.001	0.001	23.16(5.751)	9	30
	40		0.037(0.077)	0	0.2	31.68(6.944)	12	40
	50		0.079(0.095)	0	0.2	41.8(8.003)	25	50
	60	indiferente	0.088(0.089)	0	0.2	49.08(9.207)	24	60
	70		0.094(0.121)	0	0.3	60.9(9.083)	35	70
	80		0.061(0.087)	0	0.2	72.32(8.698)	48	80
	90		0.033(0.068)	0	0.2	76.86(12.226)	45	90
	100		0.016(0.047)	0	0.2	89.8(12.204)	60	100
	II	20	1.804(1.924)	0.047(0.073)	0	0.2	11.52(5.339)	4
30		3.064(1.941)	0.025(0.045)	0	0.2	16.92(7.756)	6	30
40		3.332(1.811)	0.009(0.024)	0	0.1	25.28(9.23)	8	40
50		3.66(1.695)	0.022(0.049)	0	0.2	24.6(12.529)	5	50
60		3.15(1.785)	0.012(0.033)	0	0.2	38.04(16.603)	12	60
70		2.98(1.635)	0.006(0.02)	0	0.1	48.72(17.26)	14	70
80		3.96(1.551)	0.002(0.004)	0	0.01	49.76(23.214)	16	80
90		3.38(1.589)	0.011(0.027)	0	0.1	62.82(22.082)	18	90
100		3.46(1.717)	0.011(0.034)	0	0.2	63.2(25.67)	10	100
III		20	1.9(1.897)	0.054(0.08)	0	0.2		
	30	1.964(1.755)	0.019(0.041)	0	0.2			
	40	1.978(1.799)	0.023(0.044)	0	0.2			
	50	2.364(1.841)	0.028(0.058)	0	0.2			
	60	2.216(1.836)	0.011(0.027)	0	0.1		20(0)	
	70	1.996(1.697)	0.014(0.032)	0	0.1			
	80	2.104(1.664)	0.012(0.027)	0	0.1			
	90	2.572(1.773)	0.023(0.053)	0	0.2			
	100	3.25(1.882)	0.025(0.045)	0	0.2			

Tabla 3: Los mejores hiperparámetros SVF

La Tabla 4 describe el rendimiento de los distintos métodos estudiados: FDH, SVF, DEA y CSVF en función del MSE. Las dos primeras columnas indican el escenario y el tamaño de la muestra. Las cuatro columnas siguientes muestran la media (MEAN) y la desviación típica (entre paréntesis) de los métodos considerados. Las dos columnas siguientes informan de la fracción de experimentos en los que el SVF mejora o iguala el MSE del FDH y del porcentaje de mejora de este método respecto al otro. Las dos últimas columnas son como las anteriores, pero comparan CSVF frente a DEA. En cuanto a los

resultados, todos los métodos se vieron afectados por el aumento de la dimensionalidad, de una a varias entradas, ya que el MSE aumenta a medida que lo hace el número de inputs. Asimismo, la mejora del SVF respecto al FDH fue bastante sustancial, dando mejores resultados en todas las simulaciones. Las mejoras oscilaron entre el 14,3% y el 34,4% de media. En cuanto a la comparación entre CSVF y DEA, los porcentajes de mejora fueron superiores a los de la comparación anterior entre SVF y FDH, con mejoras que oscilaron entre el 38,7% y el 78,4% de media. Un resultado a destacar es que la superioridad de SVF y CSVF frente a los enfoques tradicionales es elevada incluso en contextos con un número reducido de observaciones.

Escenario	Número de observaciones	Error cuadrático medio							
						Fracción de simulaciones		Fracción de simulaciones	
		FDH	SVF	DEA	CSVF	SVF<=FDH	Mejora (%)	CSVF<=DEA	Mejora (%)
I	20	0.06(0.026)	0.046(0.024)	0.021(0.017)	0.012(0.011)	1	23.126(18.476)	0.96	39.28(29.892)
	30	0.048(0.016)	0.038(0.014)	0.017(0.011)	0.01(0.008)	1	20.47(13.607)	0.98	39.604(27.292)
	40	0.036(0.01)	0.029(0.009)	0.01(0.007)	0.006(0.006)	1	18(9.163)	0.98	41.54(22.823)
	50	0.033(0.011)	0.026(0.01)	0.008(0.006)	0.004(0.005)	1	19.686(13.358)	1	49.347(24.152)
	60	0.027(0.008)	0.022(0.008)	0.007(0.006)	0.004(0.004)	1	18.776(10.464)	1	47.294(20.497)
	70	0.026(0.008)	0.021(0.007)	0.007(0.005)	0.003(0.004)	1	20.539(10.891)	1	52.575(21.751)
	80	0.022(0.007)	0.018(0.006)	0.005(0.004)	0.003(0.002)	1	17.649(10.122)	1	48.986(23.684)
	90	0.02(0.005)	0.017(0.005)	0.004(0.003)	0.002(0.002)	1	14.712(8.33)	1	40.99(17.318)
	100	0.018(0.005)	0.015(0.004)	0.004(0.003)	0.002(0.002)	1	15.18(9.632)	1	38.709(22.785)
	II	20	0.093(0.031)	0.062(0.029)	0.048(0.023)	0.019(0.012)	1	32.906(21.515)	0.96
30		0.088(0.03)	0.065(0.027)	0.039(0.02)	0.014(0.01)	1	25.792(15.893)	0.98	62.702(21.254)
40		0.074(0.021)	0.058(0.021)	0.031(0.013)	0.011(0.008)	1	21.86(16.536)	1	64.682(19.001)
50		0.067(0.015)	0.052(0.016)	0.022(0.009)	0.009(0.009)	1	23.709(13.996)	0.96	55.665(45.693)
60		0.064(0.015)	0.05(0.015)	0.021(0.01)	0.008(0.007)	1	21.534(13.903)	0.98	61.858(24.149)
70		0.06(0.012)	0.05(0.013)	0.02(0.007)	0.008(0.007)	1	17.301(13.545)	0.98	58.134(21.477)
80		0.055(0.01)	0.046(0.01)	0.017(0.007)	0.006(0.003)	1	15.941(11.554)	0.98	63.6(16.983)
90		0.051(0.011)	0.044(0.011)	0.016(0.007)	0.007(0.005)	1	14.314(11.453)	0.98	60.297(19.467)
100		0.047(0.008)	0.04(0.01)	0.014(0.005)	0.005(0.003)	1	15.497(12.1)	0.98	62.138(18.285)
III		20	0.125(0.046)	0.085(0.034)	0.078(0.036)	0.029(0.019)	1	30.401(20.041)	1
	30	0.11(0.03)	0.075(0.028)	0.062(0.03)	0.019(0.013)	1	31.182(19.658)	0.98	66.747(18.605)
	40	0.106(0.032)	0.073(0.03)	0.054(0.02)	0.015(0.01)	1	31.681(18.429)	0.98	70.078(20.382)
	50	0.093(0.021)	0.063(0.018)	0.042(0.015)	0.011(0.006)	1	31.281(17.332)	1	73.877(15.085)
	60	0.094(0.018)	0.063(0.017)	0.042(0.014)	0.011(0.006)	1	32.396(14.453)	1	73.107(12.174)
	70	0.092(0.019)	0.061(0.018)	0.04(0.015)	0.01(0.007)	1	33.688(15.497)	1	74.851(14.008)
	80	0.082(0.017)	0.054(0.018)	0.034(0.011)	0.007(0.003)	1	34.423(15.374)	1	78.375(12.496)
	90	0.08(0.015)	0.053(0.015)	0.03(0.009)	0.009(0.013)	1	33.81(15.881)	0.96	66.782(53.812)
	100	0.079(0.013)	0.056(0.013)	0.03(0.009)	0.007(0.007)	1	29.867(11.365)	0.96	74.618(22.103)

Tabla 4: Rendimiento relativo de los métodos de estimación basados en el MSE

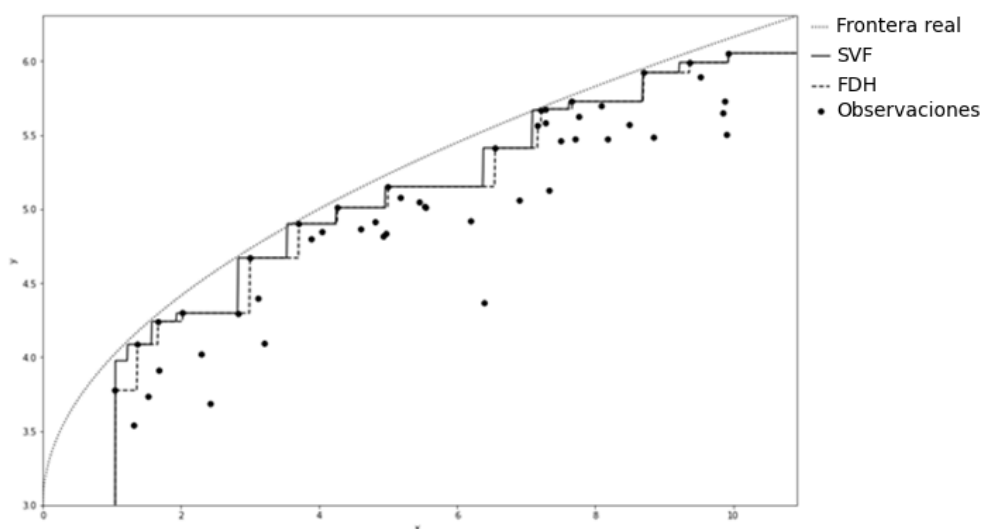
La Tabla 5 muestra los resultados basados en el sesgo en lugar del MSE. La estructura es similar a la de la Tabla 4. En cuanto al sesgo, SVF supera a FDH en todas las experiencias computacionales realizadas, con una reducción que oscila entre el 8% y el 19,9%. En cuanto a las tecnologías convexas, CSVF funciona mejor que el DEA tradicional, con mejoras que van del 27,3% al 56,1%. En el caso de CSVF, el porcentaje de mejora

aumenta a medida que aumenta el número de inputs. Como en el caso del MSE, el dominio de los nuevos enfoques frente a los tradicionales es claro, incluso en escenarios basados en pocas observaciones.

Por último, la Figura 12 muestra un ejemplo gráfico del resultado de una de nuestras simulaciones.

Escenario	Número de observaciones	Sesgo									
		Fracción de simulaciones				Mejora (%)		Fracción de simulaciones		Mejora (%)	
		FDH_ABS	SVF_ABS	DEA_ABS	CSVF_ABS	SVF_ABS <= FDH_ABS	SVF_ABS vs FDH_ABS	CSVF_ABS <= DEA_ABS	CSVF_ABS vs DEA_ABS		
I	20	0.198(0.05)	0.17(0.05)	0.104(0.042)	0.074(0.034)	1	14.211(10.371)	0.9	27.252(24.578)		
	30	0.18(0.034)	0.156(0.034)	0.089(0.029)	0.06(0.023)	1	13.383(7.095)	0.96	30.993(18.712)		
	40	0.155(0.024)	0.137(0.026)	0.066(0.022)	0.045(0.021)	1	12.288(6.549)	1	33.668(15.439)		
	50	0.15(0.023)	0.132(0.024)	0.06(0.016)	0.04(0.016)	1	11.767(7.046)	0.98	35.347(16.288)		
	60	0.134(0.018)	0.119(0.02)	0.053(0.016)	0.035(0.014)	1	11.152(5.373)	1	36.312(11.539)		
	70	0.131(0.019)	0.116(0.02)	0.049(0.015)	0.032(0.015)	1	11.157(4.782)	1	37.062(13.023)		
	80	0.123(0.02)	0.111(0.019)	0.046(0.015)	0.031(0.013)	1	9.347(3.65)	1	33.281(11.379)		
	90	0.116(0.013)	0.106(0.014)	0.039(0.01)	0.027(0.009)	1	9.009(3.961)	1	31.853(10.586)		
	100	0.11(0.015)	0.101(0.015)	0.038(0.011)	0.027(0.01)	1	8.722(3.565)	0.98	30.588(11.75)		
	II	20	0.249(0.045)	0.2(0.05)	0.17(0.042)	0.101(0.032)	1	19.867(13.551)	0.96	39.658(18.271)	
30		0.246(0.044)	0.212(0.046)	0.151(0.038)	0.082(0.031)	1	14.097(9.055)	0.98	44.811(18.545)		
40		0.223(0.035)	0.197(0.04)	0.132(0.029)	0.07(0.026)	1	12.283(8.81)	1	46.997(15.129)		
50		0.218(0.026)	0.189(0.032)	0.114(0.021)	0.063(0.027)	1	13.594(8.496)	0.96	43.884(22.736)		
60		0.207(0.025)	0.183(0.028)	0.104(0.02)	0.057(0.022)	1	11.829(7.956)	0.98	45.423(17.326)		
70		0.205(0.022)	0.187(0.023)	0.1(0.013)	0.061(0.021)	1	8.593(6.036)	0.98	39.713(18.1)		
80		0.193(0.017)	0.177(0.018)	0.091(0.013)	0.05(0.014)	1	8.19(5.649)	0.96	44.273(14.527)		
90		0.187(0.02)	0.173(0.022)	0.087(0.017)	0.05(0.017)	1	7.24(5.748)	0.98	42.471(15.682)		
100		0.182(0.017)	0.168(0.021)	0.081(0.013)	0.046(0.013)	1	8.035(6.362)	0.96	43.309(16.463)		
III		20	0.283(0.049)	0.234(0.048)	0.214(0.045)	0.117(0.036)	1	16.94(12.771)	0.98	44.533(15.679)	
	30	0.271(0.038)	0.224(0.045)	0.19(0.042)	0.098(0.032)	1	17.024(12.169)	0.98	47.769(15.344)		
	40	0.264(0.039)	0.22(0.046)	0.173(0.034)	0.085(0.026)	1	17.113(10.631)	0.96	50.106(16.306)		
	50	0.25(0.03)	0.206(0.03)	0.153(0.027)	0.072(0.023)	1	17.063(9.746)	1	52.36(15.557)		
	60	0.25(0.027)	0.207(0.031)	0.15(0.024)	0.07(0.02)	1	17.391(8.809)	1	52.876(11.499)		
	70	0.248(0.023)	0.204(0.031)	0.144(0.023)	0.069(0.022)	1	17.825(9.214)	0.96	52.017(16.303)		
	80	0.235(0.026)	0.192(0.032)	0.134(0.021)	0.058(0.015)	1	18.561(8.696)	1	56.066(12.685)		
	90	0.231(0.021)	0.189(0.027)	0.124(0.017)	0.064(0.035)	1	18.151(9.2)	0.94	47.215(31.566)		
	100	0.232(0.019)	0.195(0.022)	0.124(0.016)	0.058(0.024)	1	15.712(6.472)	0.96	53.283(18.828)		

Tabla 5: Rendimiento relativo de los métodos de estimación según los criterios de sesgo



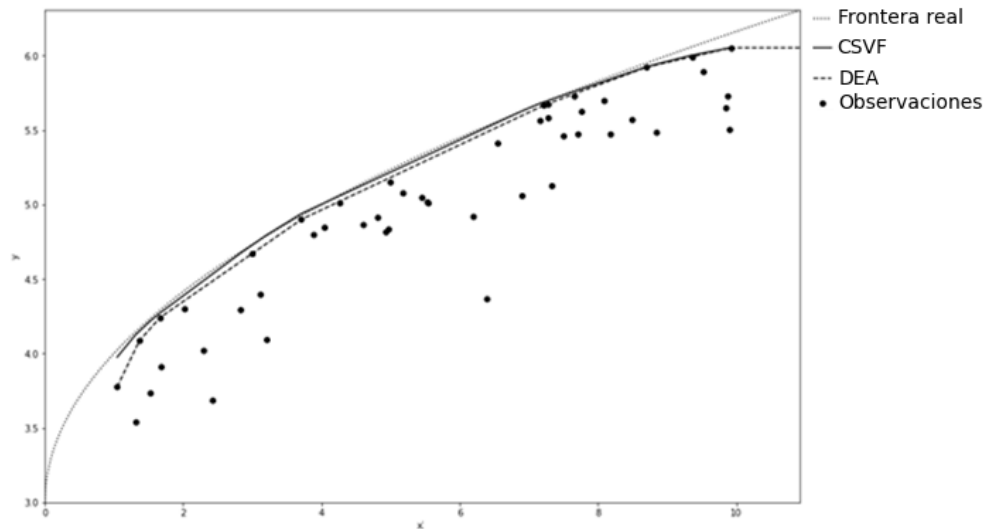


Figura 12: Ilustraciones gráficas de SVF frente a FDH y CSVF frente a DEA en un conjunto de datos simulado

El nuevo enfoque se basa en SVR, que combina la minimización del error empírico con la minimización del error estructural. La minimización del error estructural procede de una teoría compleja, en la que la atención se centra en la determinación de un límite del llamado aprendizaje PAC (Probably Approximately Correct) (véase, por ejemplo, [Cristianini y Shawe-Taylor, \(2000\)](#)). El objetivo es que la función estimada tenga muy probablemente un error de generalización bajo (es decir, una medida de la precisión con la que un modelo es capaz de estimar valores de resultados para datos no vistos previamente). En consecuencia, las SVF, basadas en SVR, aprenden cómo es el proceso generador de datos oculto evitando hacer una simple “descripción” de la muestra de datos a nivel de frontera, algo que sí ocurre con FDH y DEA, que se basan exclusivamente en la minimización del error empírico debido al principio de extrapolación mínima. Además, los experimentos mostraron cómo la frontera de producción estimada por SVF se aproxima más a la frontera teórica que FDH y DEA, que de alguna manera representan modelos sobreajustados. La frontera estimada por SVF se desvía de las fronteras de producción de FDH y DEA, evitando así en algunos aspectos el problema del sobreajuste.

4.1.3.2. SIMULACIONES MULTI-OUTPUT

En esta sección, compararemos los estimadores derivados de FDH, DEA, SVF y CSVF como se ha hecho anteriormente pero bajo escenarios de producción de tipo multi-output. Además, recurriremos a la aplicación del modelo SVF y su versión simplificada. En

concreto, simulamos los datos de un proceso de producción con dos variables input y dos variables output. Para la simulación de los escenarios, seguimos el método propuesto por [Perelman y Santin \(2009\)](#), cumpliendo las condiciones habituales de regularidad microeconómicas. El proceso realizado para la generación de los conjuntos de datos fue el siguiente:

En primer lugar, los datos de entrada se crean aleatoriamente utilizando una distribución uniforme sobre el intervalo [5,50] independientemente para cada entrada y observación. Asimismo, se genera aleatoriamente un término z utilizando una distribución uniforme sobre el intervalo [-1,5,1,5] para cada observación, donde $z = \ln \frac{y_2^*}{y_1^*}$, y_1^* e y_2^* son los valores de las variables tipo output situados en la frontera de producción para el output 1 y 2, respectivamente. En segundo lugar, calculamos el valor de $-\ln(y_1^*)$ utilizando:

$$\begin{aligned}
 -\ln(y_1^*) = & -1 + 0.5 \left(\ln \frac{y_2^*}{y_1^*} \right) + 0.25 \left(\ln \frac{y_2^*}{y_1^*} \right)^2 - 1.5 (\ln x_1) \\
 & - 0.6 (\ln x_2) + 0.2 (\ln x_1)^2 + 0.05 (\ln x_2)^2 \\
 & - 0.1 (\ln x_1) (\ln x_2) + 0.05 (\ln x_1) \left(\ln \frac{y_2^*}{y_1^*} \right) \\
 & - 0.05 (\ln x_2) \left(\ln \frac{y_2^*}{y_1^*} \right)
 \end{aligned} \tag{33}$$

A partir de la última expresión, obtenemos el valor del primer output en la frontera de producción: $y_1^* = \exp(\ln(y_1^*))$. En tercer lugar, determinamos el valor del segundo producto en la frontera de producción: $y_2^* = \exp(\ln(y_2^*))$. Como sugirieron Perelman y Santin, permitimos que el 0%, el 5% y el 10% de las unidades simuladas estuvieran en la frontera teórica. Para las DMUs que no se encontraban en la frontera, consideramos un escenario en el que calculamos una distribución seminormal $\ln D = u \sim |N(0, 0.3)|$ para generar el término de ineficiencia para cada producto. Los resultados simulados que

reflejan la ineficiencia técnica se calcularon como sigue: $\hat{y}_1 = y_1^* \frac{1}{\exp(\ln D)}$ e

$$\hat{y}_2 = y_2^* \frac{1}{\exp(\ln D)}.$$

Finalmente, otro escenario incluyó ruido aleatorio para cada salida de aquellas DMUs que no estaban situadas en la frontera por definición. En ese escenario, estimamos los valores

de salida como: $\hat{y}_1 = \hat{y}_1 \frac{1}{\exp(v_1)}$ e $\hat{y}_2 = \hat{y}_2 \frac{1}{\exp(v_2)}$, donde $v_1 \sim N(0, 0.01)$,

$v_2 \sim N(0, 0.01)$, para cada observación.

Una vez obtenido el conjunto de datos para las simulaciones, realizamos 50 pruebas, $e=1, \dots, 50$, de cada combinación de tamaño de muestra, porcentaje de unidades en la frontera verdadera y consideración o no de ruido aleatorio para evaluar el rendimiento de cada enfoque. Se utilizó la validación cruzada, basada en cinco folds, para seleccionar la mejor combinación de hiperparámetros para cada ensayo. En nuestro modelo, los hiperparámetros son C , ε y d . Fijamos $C \in \{0.1, 0.5, 1, 2, 5\}$ y $\varepsilon \in \{0, 0.001, 0.01, 0.1, 0.2\}$. Además, $d = 0.1 \cdot h \cdot n$ (redondeado al entero más próximo), donde $h = 1, \dots, 10$. Estos valores generaron 250 combinaciones diferentes de hiperparámetros para el modelo SVF. El error cuadrático medio (MSE) y el sesgo se utilizaron para medir el rendimiento de cada enfoque (véase la Tabla 6). Se aplicaron las mismas fórmulas en el enfoque SSVF.

Método	MSE	Sesgo
SVF	$\sum_{e=1}^{50} \sum_{i=1}^n (\phi(\mathbf{x}_i, \mathbf{y}_i) - \phi_{SVF}(\mathbf{x}_i, \mathbf{y}_i))^2 / 50n$	$\sum_{e=1}^{50} \sum_{i=1}^n \phi(\mathbf{x}_i, \mathbf{y}_i) - \phi_{SVF}(\mathbf{x}_i, \mathbf{y}_i) / 50n$
FDH	$\sum_{e=1}^{50} \sum_{i=1}^n (\phi(\mathbf{x}_i, \mathbf{y}_i) - \phi_{FDH}(\mathbf{x}_i, \mathbf{y}_i))^2 / 50n$	$\sum_{e=1}^{50} \sum_{i=1}^n \phi(\mathbf{x}_i, \mathbf{y}_i) - \phi_{FDH}(\mathbf{x}_i, \mathbf{y}_i) / 50n$
DEA	$\sum_{e=1}^{50} \sum_{i=1}^n (\phi(\mathbf{x}_i, \mathbf{y}_i) - \phi_{DEA}(\mathbf{x}_i, \mathbf{y}_i))^2 / 50n$	$\sum_{e=1}^{50} \sum_{i=1}^n \phi(\mathbf{x}_i, \mathbf{y}_i) - \phi_{DEA}(\mathbf{x}_i, \mathbf{y}_i) / 50n$
CSVF	$\sum_{e=1}^{50} \sum_{i=1}^n (\phi(\mathbf{x}_i, \mathbf{y}_i) - \phi_{CSVF}(\mathbf{x}_i, \mathbf{y}_i))^2 / 50n$	$\sum_{e=1}^{50} \sum_{i=1}^n \phi(\mathbf{x}_i, \mathbf{y}_i) - \phi_{CSVF}(\mathbf{x}_i, \mathbf{y}_i) / 50n$

Tabla 6: Cálculo del estadístico MSE y del sesgo para cada método

La Tabla 7 muestra los mejores hiperparámetros en nuestras simulaciones sin ruido aleatorio (media y desviación típica entre paréntesis). Los resultados con ruido fueron muy similares. La primera columna indica el tipo de algoritmo utilizado (original o su versión simplificada), mientras que la segunda columna muestra los porcentajes de DMU simuladas en la frontera. En cuanto a los hiperparámetros, todos los valores suelen ser mayores en el método SVF que en el SSVF, en particular el hiperparámetro C . Además, obsérvese que cuanto mayor es el tamaño de la muestra, mayor es el valor del hiperparámetro d .

Método	Número observaciones	Frontera	Hiperparámetros		
			C	ϵ	d
			MEAN (STD)	MEAN (STD)	MEAN (STD)
Original	20	0%	2.350(1.995)	0.100(0.101)	13.200(4.982)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	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)
Simplificado	100	10%	0.810(0.885)	0.084(0.100)	77.000(17.871)

Tabla 7: Mejores hiperparámetros para el escenario sin ruido aleatorio

La Tabla 8 incluye los escenarios simulados sin ruido aleatorio en las salidas, mientras que la Tabla 9 recoge los resultados correspondientes a escenarios con ruido aleatorio. La estructura de ambas tablas es la misma: las tres primeras columnas indican el método utilizado, el tamaño de la muestra y los porcentajes de DMUs en la frontera, por este orden. Las tablas también muestran la media y la desviación típica entre paréntesis, la fracción de ensayos en los que el SVF mejora o iguala el MSE del FDH y el porcentaje de mejora de este método respecto al otro. Las dos columnas siguientes son similares a las anteriores, pero comparando CSVF frente a DEA. Además, incluimos dos columnas que comparan la ratio de ensayos en los que los resultados de SSVF/CSSVF son mejores o iguales que los de SVF/CSVF. Por último, mostramos las mismas métricas para los resultados de sesgo.

En cuanto a los resultados, el nuevo enfoque supera a FDH, reportando resultados superiores en todos los escenarios considerados. La mejora del MSE en SVF osciló entre el 20% y el 28% de media en las simulaciones sin ruido y entre el 19% y el 28% de media en las simulaciones con ruido. Además, las mejoras en MSE fueron mayores para SSVF, donde oscilaron entre el 24% y el 48% de media en simulaciones sin ruido y entre el 24% y el 50% en simulaciones con ruido aleatorio. Del mismo modo, CSVF superó a DEA, con mejoras para SVF que oscilaron entre el 17% y el 24% de media en simulaciones sin ruido y entre el 16% y el 23% de media en simulaciones con ruido. En el método SSVF, el porcentaje de mejora fue superior al correspondiente al método SVF, con mejoras que oscilaron entre el 21% y el 34% de media en las simulaciones sin ruido y entre el 22% y el 30% de media en las simulaciones con ruido. Cabe destacar que los resultados asociados a la nueva metodología son buenos, incluso para tamaños de muestra pequeños. Además, la fracción de ensayos muestra cómo el modelo SSVF obtuvo mejores resultados que el modelo SVF en la mayoría de los escenarios, especialmente cuando aumenta el tamaño de la muestra. Estas mismas tendencias de mejora se repiten con respecto al sesgo.

La Figura 13 muestra la media del tiempo de ejecución de los escenarios, agregada por tamaño de muestra. Esta métrica se calculó como: $\left[\sum_{t=1}^{50} (CV_t + BM_t + SBM_t) \right] / 50$, donde

CV es el tiempo asociado al proceso de validación cruzada, BM es el tiempo relacionado

con la construcción del mejor modelo con los hiperparámetros seleccionados por la validación cruzada y SBM es el tiempo empleado para resolver el mejor modelo. Los experimentos se llevaron a cabo en un clúster con una CPU Intel(R) Xeon(R) E5-2650 v3 a 2,3 GHz con 40 núcleos, 62 Gigabytes de RAM y un sistema operativo Ubuntu 18.04.5 LTS. Se utilizó CPLEX v12.8 para resolver los programas de optimización. Como podemos observar en la figura, el tiempo de ejecución del método SVF fue superior al correspondiente al método SSVF. Además, la figura ilustra cómo el incremento de tiempo es exponencial en el método SVF cuando el número de DMUs aumenta mientras que, en el método SSVF, el incremento es menos pronunciado.

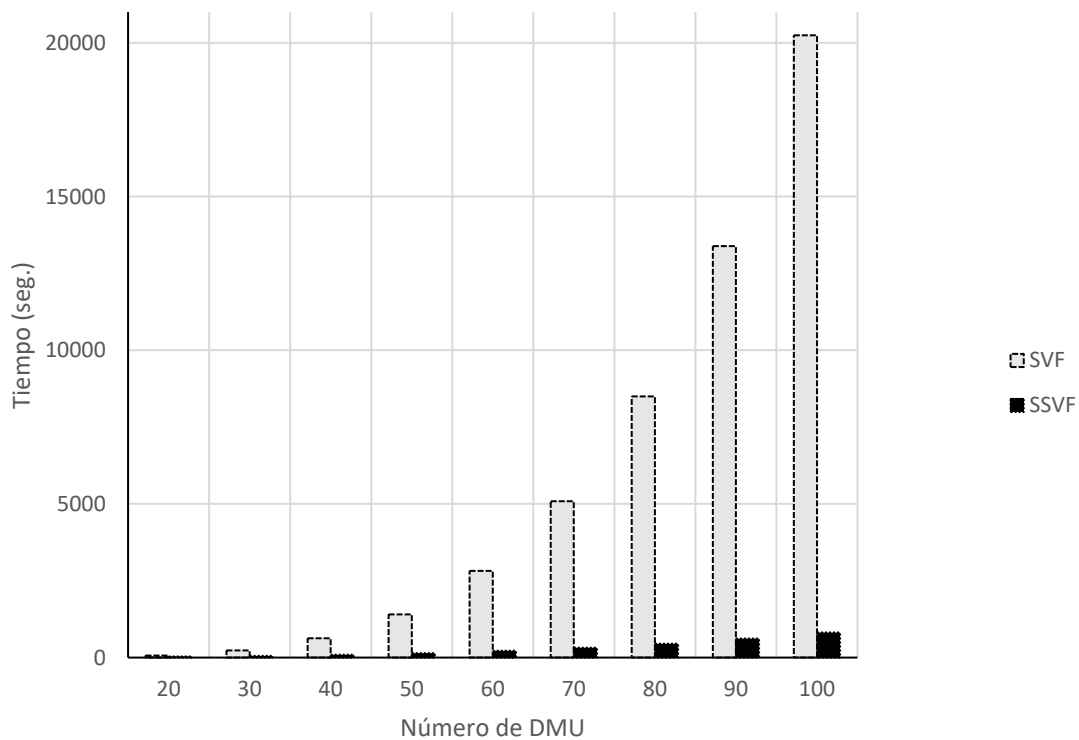


Figura 13: Tiempo de ejecución en segundos de SVF y SSVF

Método	TAM	Frontera	MSE										Sesgo											
			FDH	SVF	DEA	CSVF	Frac. de sim.		Mejora (%)		Frac. de sim.		Mejora (%)		FDH	SVF	DEA	CSVF	Frac. de sim.		Mejora (%)		Frac. de sim.	
							SVF<= FDH	SVF vs FDH	CSVF<= DEA	CSVF vs DEA	SSVF<= SVF	CSSVF<= CSVF	SVF<= FDH	SVF vs FDH					CSVF<= DEA	CSVF vs DEA	SSVF<= SVF	CSSVF<= CSVF		
Original	20	0%	0.764(0.379)	0.600(0.337)	0.391(0.237)	0.316(0.197)	1	21%	1	17%	0.64	0.82	0.620(0.150)	0.552(0.145)	0.457(0.128)	0.408(0.119)	1	11%	1	10%	0.68	0.82		
Simplificado	20	0%	0.764(0.379)	0.583(0.359)	0.391(0.237)	0.313(0.225)	0.98	24%	1	21%			0.620(0.150)	0.544(0.154)	0.457(0.128)	0.399(0.135)	1	13%	1	13%				
Original	20	5%	0.704(0.352)	0.563(0.335)	0.343(0.216)	0.282(0.195)	1	21%	1	18%	0.66	0.72	0.582(0.138)	0.518(0.145)	0.414(0.119)	0.367(0.123)	1	11%	1	12%	0.72	0.76		
Simplificado	20	5%	0.704(0.352)	0.532(0.339)	0.343(0.216)	0.266(0.195)	0.98	25%	1	24%			0.582(0.138)	0.503(0.147)	0.414(0.119)	0.352(0.124)	1	14%	1	16%				
Original	20	10%	0.671(0.356)	0.539(0.347)	0.313(0.211)	0.263(0.206)	1	21%	1	17%	0.66	0.72	0.553(0.140)	0.492(0.144)	0.379(0.115)	0.337(0.118)	1	11%	1	11%	0.68	0.72		
Simplificado	20	10%	0.671(0.356)	0.502(0.342)	0.313(0.211)	0.243(0.191)	0.98	27%	1	24%			0.553(0.140)	0.476(0.149)	0.379(0.115)	0.324(0.122)	1	15%	1	16%				
Original	30	0%	0.681(0.255)	0.551(0.235)	0.311(0.136)	0.257(0.130)	1	20%	1	17%	0.82	0.78	0.592(0.097)	0.533(0.095)	0.406(0.086)	0.359(0.083)	1	10%	1	12%	0.84	0.82		
Simplificado	30	0%	0.681(0.255)	0.478(0.215)	0.311(0.136)	0.233(0.116)	1	30%	1	25%			0.592(0.097)	0.499(0.101)	0.406(0.086)	0.343(0.088)	1	16%	1	16%				
Original	30	5%	0.631(0.260)	0.497(0.234)	0.263(0.134)	0.211(0.125)	1	22%	1	20%	0.84	0.84	0.550(0.100)	0.486(0.095)	0.349(0.079)	0.302(0.072)	1	12%	1	14%	0.9	0.88		
Simplificado	30	5%	0.631(0.260)	0.438(0.219)	0.263(0.134)	0.193(0.113)	1	32%	1	27%			0.550(0.100)	0.456(0.098)	0.349(0.079)	0.286(0.075)	1	17%	1	18%				
Original	30	10%	0.601(0.233)	0.469(0.206)	0.241(0.132)	0.200(0.129)	1	22%	1	19%	0.86	0.74	0.526(0.092)	0.463(0.086)	0.326(0.076)	0.282(0.073)	1	12%	1	14%	0.9	0.78		
Simplificado	30	10%	0.601(0.233)	0.402(0.182)	0.241(0.132)	0.170(0.105)	1	33%	1	28%			0.526(0.092)	0.428(0.087)	0.326(0.076)	0.262(0.072)	1	19%	1	20%				
Original	40	0%	0.621(0.174)	0.500(0.166)	0.252(0.128)	0.212(0.126)	1	20%	1	17%	0.86	0.86	0.579(0.080)	0.521(0.083)	0.369(0.067)	0.324(0.073)	1	10%	1	13%	0.86	0.88		
Simplificado	40	0%	0.621(0.174)	0.450(0.170)	0.252(0.128)	0.196(0.125)	1	28%	1	24%			0.579(0.080)	0.492(0.088)	0.369(0.067)	0.309(0.073)	1	15%	1	17%				
Original	40	5%	0.557(0.163)	0.440(0.155)	0.218(0.126)	0.179(0.123)	1	21%	1	20%	0.88	0.88	0.536(0.076)	0.474(0.076)	0.324(0.062)	0.278(0.068)	1	12%	1	15%	0.88	0.86		
Simplificado	40	5%	0.557(0.163)	0.385(0.154)	0.218(0.126)	0.165(0.121)	1	31%	1	27%			0.536(0.076)	0.439(0.077)	0.324(0.062)	0.263(0.065)	1	18%	1	20%				
Original	40	10%	0.519(0.167)	0.417(0.159)	0.191(0.124)	0.159(0.121)	1	20%	1	19%	0.92	0.88	0.502(0.077)	0.445(0.078)	0.289(0.056)	0.246(0.058)	1	11%	1	15%	0.94	0.88		
Simplificado	40	10%	0.519(0.167)	0.352(0.155)	0.191(0.124)	0.145(0.121)	1	32%	1	27%			0.502(0.077)	0.405(0.078)	0.289(0.056)	0.232(0.059)	1	19%	1	20%				
Original	50	0%	0.572(0.196)	0.454(0.175)	0.209(0.090)	0.172(0.087)	1	21%	1	19%	0.98	0.94	0.550(0.082)	0.487(0.079)	0.331(0.054)	0.281(0.056)	1	12%	1	15%	0.96	0.94		
Simplificado	50	0%	0.572(0.196)	0.389(0.158)	0.209(0.090)	0.156(0.084)	1	32%	1	27%			0.550(0.082)	0.447(0.071)	0.331(0.054)	0.263(0.056)	1	19%	1	21%				
Original	50	5%	0.531(0.196)	0.420(0.169)	0.181(0.076)	0.151(0.074)	1	21%	1	18%	0.96	0.88	0.520(0.085)	0.459(0.079)	0.297(0.048)	0.252(0.049)	1	12%	1	15%	0.94	0.86		
Simplificado	50	5%	0.531(0.196)	0.353(0.148)	0.181(0.076)	0.135(0.068)	1	34%	1	26%			0.520(0.085)	0.417(0.068)	0.297(0.048)	0.238(0.044)	1	20%	1	20%				
Original	50	10%	0.483(0.180)	0.374(0.159)	0.152(0.068)	0.126(0.069)	1	23%	1	19%	0.94	0.82	0.479(0.079)	0.417(0.075)	0.256(0.041)	0.217(0.047)	1	13%	1	16%	0.94	0.86		
Simplificado	50	10%	0.483(0.180)	0.305(0.133)	0.152(0.068)	0.115(0.064)	1	37%	1	26%			0.479(0.079)	0.371(0.062)	0.256(0.041)	0.204(0.042)	1	22%	1	21%				
Original	60	0%	0.557(0.193)	0.441(0.160)	0.183(0.090)	0.146(0.072)	1	21%	1	20%	1	0.9	0.538(0.075)	0.478(0.065)	0.308(0.054)	0.264(0.051)	1	11%	1	14%	1	0.9		
Simplificado	60	0%	0.557(0.193)	0.356(0.140)	0.183(0.090)	0.132(0.070)	1	36%	1	29%			0.538(0.075)	0.429(0.065)	0.308(0.054)	0.247(0.052)	1	20%	1	20%				
Original	60	5%	0.521(0.188)	0.397(0.143)	0.161(0.086)	0.126(0.069)	1	23%	1	22%	0.9	0.82	0.507(0.075)	0.439(0.062)	0.274(0.049)	0.231(0.044)	1	13%	1	16%	0.92	0.84		
Simplificado	60	5%	0.521(0.188)	0.327(0.141)	0.161(0.086)	0.115(0.066)	1	38%	0.98	29%			0.507(0.075)	0.394(0.065)	0.274(0.049)	0.218(0.046)	1	22%	1	21%				
Original	60	10%	0.480(0.191)	0.366(0.156)	0.138(0.080)	0.104(0.059)	1	24%	1	24%	1	0.82	0.471(0.074)	0.405(0.067)	0.238(0.044)	0.197(0.038)	1	14%	1	17%	0.98	0.82		
Simplificado	60	10%	0.480(0.191)	0.294(0.133)	0.138(0.080)	0.098(0.059)	1	39%	1	29%			0.471(0.074)	0.360(0.062)	0.238(0.044)	0.189(0.040)	1	24%	1	21%				
Original	70	0%	0.583(0.272)	0.437(0.177)	0.174(0.108)	0.136(0.092)	1	24%	1	22%	0.96	0.88	0.546(0.077)	0.476(0.069)	0.296(0.057)	0.245(0.052)	1	13%	1	17%	0.96	0.9		
Simplificado	70	0%	0.583(0.272)	0.346(0.134)	0.174(0.108)	0.122(0.088)	1	39%	1	31%			0.546(0.077)	0.425(0.062)	0.296(0.057)	0.228(0.052)	1	22%	1	23%				
Original	70	5%	0.539(0.266)	0.409(0.193)	0.147(0.104)	0.116(0.094)	1	24%	1	22%	0.96	0.76	0.506(0.077)	0.440(0.072)	0.250(0.051)	0.206(0.045)	1	13%	1	18%	0.98	0.8		
Simplificado	70	5%	0.539(0.266)	0.305(0.127)	0.147(0.104)	0.106(0.090)	1	42%	1	29%			0.506(0.077)	0.382(0.063)	0.250(0.051)	0.193(0.043)	1	24%	1	23%				
Original	70	10%	0.505(0.257)	0.372(0.172)	0.125(0.091)	0.098(0.078)	1	25%	1	21%	0.98	0.76	0.477(0.072)	0.409(0.063)	0.219(0.044)	0.181(0.034)	1	14%	1	17%	1	0.72		
Simplificado	70	10%	0.505(0.257)	0.281(0.118)	0.125(0.091)	0.095(0.082)	1	43%	1	25%			0.477(0.072)	0.358(0.059)	0.219(0.044)	0.174(0.037)	1	25%	1	20%				
Original	80	0%	0.511(0.155)	0.401(0.124)	0.151(0.061)	0.124(0.057)	1	21%	1	19%	1	0.92	0.528(0.068)	0.465(0.066)	0.279(0.047)	0.233(0.045)	1	12%	1	16%	1	0.94		
Simplificado	80	0%	0.511(0.155)	0.308(0.098)	0.151(0.061)	0.112(0.057)	1	40%	1	27%			0.528(0.068)	0.406(0.061)	0.279(0.047)	0.216(0.045)	1	23%	1	23%				
Original	80	5%	0.464(0.136)	0.357(0.112)	0.126(0.056)	0.103(0.055)	1	23%	1	20%	1	0.88	0.491(0.063)	0.425(0.062)	0.237(0.040)	0.197(0.038)	1	13%	1	17%	1	0.94		
Simplificado	80	5%	0.464(0.136)	0.274(0.085)	0.126(0.056)	0.092(0.051)	1	41%	1	28%			0.491(0.063)	0.370(0.056)	0.237(0.040)	0.182(0.036)	1	25%	1	23%				
Original	80	10%	0.431(0.128)	0.325(0.109)	0.110(0.054)	0.088(0.053)	1	25%	1	21%	0.98	0.9	0.458(0.059)	0.390(0.056)	0.205(0.037)	0.172(0.036)	1	15%	1	16%	0.98	0.88		
Simplificado	80	10%	0.431(0.128)	0.246(0.076)	0.110(0.054)	0.080(0.048)	1	43%	1	28%			0.458(0.059)	0.337(0.051)	0.205(0.037)	0.163(0.034)	1	26%	1	20%				
Original	90	0%	0.496(0.141)	0.377(0.111)	0.131(0.048)	0.104(0.044)	1	24%	1	22%	0.98	0.92	0.519(0.057)	0.449(0.051)	0.258(0.036)	0.208(0.034)	1	13%	1	19%	1	0.94		
Simplificado	90	0%	0.496(0.141)	0.289(0.092)	0.131(0.048)	0.085(0.034)	1	42%	1	34%			0.519(0.057)	0.389(0.051)	0.258(0.036)	0.189(0.029)	1	25%	1	26%				
Original	90	5%	0.463(0.139)	0.348(0.110)	0.114(0.046)	0.092(0.042)	1	25%	1	20%	1	0.88	0.489(0.055)	0.417(0.051)	0.224(0.031)	0.183(0.029)	1	15%	1	18%	1	0.88		
Simplificado	90	5%	0.463(0.139)	0.261(0.087)	0.114(0.046)	0.080(0.035)	1	43%	1	29%			0.489(0.055)	0.356(0.043)	0.224(0.031)	0.171(0.025)	1	27%	1	23%				
Original	90	10%	0.424(0.132)	0.315(0.111)	0.093(0.043)	0.076(0.041)	1	26%</																

En resumen, las simulaciones mostraron que la nueva metodología consigue mejores resultados que los métodos tradicionales FDH y DEA, tanto en lo que respecta al MSE como el sesgo. Aunque las tablas parecen indicar que el modelo SSVF logró mejores resultados en MSE y sesgo que en el modelo SVF, no podemos garantizar la superioridad de este método sobre el otro, debido al valor de la desviación estándar.

4.1.3.3. ILUSTRACIÓN EMPÍRICA

Para mostrar la aplicación práctica de nuestra técnica, recurrimos a un conjunto de datos reales para ilustrar el comportamiento de diferentes medidas de eficiencia técnica cuando se aplica la técnica SSVF. Para ello, utilizamos los datos correspondientes a 31 bancos taiwaneses para el año 2010, previamente estudiados por [Juo et. al \(2015\)](#). En este contexto empírico, los inputs fueron: fondos financieros (x_1), mano de obra (x_2), y capital físico (x_3), mientras que los outputs fueron inversiones financieras (y_1) y préstamos (y_2). Todas las variables monetarias se miden en millones de TWD, mientras que la mano de obra es el número de empleados.

En cuanto a la tarea computacional, para determinar el mejor modelo, aplicamos un proceso de validación cruzada de 5 folds con $C \in \{0.0001, 0.0005, 0.001, 0.01, 0.1, 1, 10\}$, $\varepsilon \in \{0, 0.001, 0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.2\}$ y $d \in \{3, 6, 9, 12, 16, 19, 22, 25, 28, 31\}$. La mejor combinación fue: $C = 10$, $\varepsilon = 0.04$, $d = 25$ después de reescalar los resultados para tratar valores entre cero y uno.

La Tabla 10 muestra los resultados para diferentes medidas de eficiencia. La primera columna indica el nombre del banco evaluado. A continuación, presentamos cuatro bloques con seis columnas en cada uno. Estos bloques corresponden a los diferentes modelos de eficiencia que consideramos en nuestro estudio: el modelo radial orientado al output, el modelo radial orientado al input, la función de distancia direccional y el modelo aditivo ponderado. En el caso de la DDF, el vector direccional seleccionado fue $\mathbf{g}^- = \mathbf{x}_i$ y $\mathbf{g}^+ = \mathbf{y}_i$, mientras que, en el caso del modelo aditivo ponderado, los pesos corresponden

a los utilizados por la medida RAM de [Cooper et al. \(1999\)](#). Cada bloque consta de seis columnas asociadas a distintos enfoques: FDH, DEA, SSVF y CSSVF. Las dos últimas columnas indican si la DMU correspondiente es ε -insensible a la eficiencia técnica o no.

En cuanto a los resultados, podemos observar cómo SSVF y CSSVF detectan más ineficiencia que FDH y DEA, respectivamente: las medias son más altas. En un análisis unidad a unidad, observamos que, independientemente de la medida de eficiencia técnica considerada, las puntuaciones de eficiencia de SSVF y CSSVF son siempre mayores o iguales que las puntuaciones arrojadas por FDH y DEA. Esta es una característica que siempre se observará en la práctica debido a la satisfacción del principio de extrapolación mínima por parte de FDH y DEA, algo que no ocurre en el caso utilizar la nueva metodología. Además, bajo cualquiera de las medidas de eficiencia consideradas, la técnica FDH sólo es capaz de detectar un banco, Cosmos Bank, como técnicamente ineficiente. Esta es una característica habitual de FDH cuando la relación entre el tamaño de la muestra y el número de dimensiones (entradas y salidas) no es buena desde un punto de vista estadístico. Se trata del conocido problema de la maldición de la dimensionalidad, estudiado en profundidad en la literatura (véase, por ejemplo, [Shen et al. \(2016\)](#)). En cambio, SSVF detecta muchos más bancos como técnicamente ineficientes: 10 unidades en el caso del DDF, 11 en el caso del modelo radial orientado a la salida, 20 con el modelo WA y, por último, 29 en el caso del modelo radial orientado a la entrada. Este patrón también es evidente cuando se comparan los resultados obtenidos por DEA y CSSVF. En cuanto a la aplicación de la noción de eficiencia técnica insensible a ε , se trata de una definición de eficiencia más robusta que la tradicional, como hemos señalado anteriormente. Para cada medida considerada en el análisis, las dos últimas columnas indican si cada unidad se sitúa dentro de la banda asociada al margen ε bajo SVF y su versión convexificada, respectivamente.

DMU	OUTPUT ORIENTED RADIAL MODEL						INPUT ORIENTED RADIAL MODEL						DIRECTIONAL DISTANCE FUNCTION						WEIGHTED ADDITIVE					
	FDH	DEA	SSVF		CSSVF		FDH	DEA	SSVF		CSSVF		FDH	DEA	SSVF		CSSVF		FDH	DEA	SSVF		CSSVF	
			EFF.	EFF.	EFF.	EFF.			EFF.	EFF.	EFF.	EFF.			EFF.	EFF.	EFF.	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
Media	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		

Tabla 10: Medidas de eficiencia obtenidas a partir del ejemplo empírico.

4.2. SUPPORT VECTOR FRONTIERS Y RANKING DE VARIABLES

Esta sección muestra una serie de métodos desarrollados para determinar el ranking de importancia de los inputs en un proceso productivo. Para ello, se han adaptado las técnicas mostradas en el apartado 3.5 al contexto de la estimación de fronteras mediante la técnica SVF.

4.2.1. SVF-PERMUTATION FEATURE IMPORTANCE

Support Vector Frontiers-Permutation Feature Importance (SVF-PFI) es una adaptación mediante SVF del método establecido por Breiman (2001) y utilizado por otros autores como Esteve et al. (2023). En este algoritmo, se combina el poder predictivo de los SVF en contextos predictivos con el método de clasificación de variables establecido por Breiman. Esta técnica se basa en el estudio del cambio en la predicción de un modelo cuando se modifica el valor de las variables del conjunto de datos original \mathbf{V} .

Como se muestra en el Algoritmo 2, primero se selecciona un conjunto de entrenamiento \mathbf{V}_{train} y un conjunto de prueba \mathbf{V}_{test} a partir del conjunto de datos original. A continuación, se entrena un modelo, (25) o (32), y se obtiene una solución óptima $(\mathbf{w}_0^*, b_0^*)^2$. A

continuación, se calcula el error del modelo inicial como $E_0 = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$, donde

$$\hat{y}_i = \mathbf{w}_0^* \phi_{SVF}^G(\mathbf{x}_i) + b_0^*.$$

Seguidamente, se realiza una copia del conjunto de entrenamiento previamente seleccionado y para cada variable se barajan los datos creando un nuevo conjunto de datos: \mathbf{V}_{train}^u . Se entrena un modelo SVF, se realiza la predicción $(y^{(u)})$ utilizando el

conjunto de prueba (\mathbf{V}_{test}), y se calcula su error como $E_u = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i^{(u)} - y_i)^2, \forall u = 1, \dots, m$.

² En el caso SVF, como se indica en la Sección 4.1.1.1, $b_0^* = 0$ porque el estimador de la frontera de producción pasa por el origen de coordenadas

Por último, establecemos una puntuación que indica el porcentaje de variación del error que se produce al permutar la variable: $S_u = 100 \frac{E_u - E_0}{E_0}$. Aquellas variables que tienen una mayor puntuación indican que son más importantes, ya que variaciones en sus valores provocan grandes perturbaciones en la predicción del modelo.

```

procedure GET_RANKING_PFI( $\mathbf{V}, C, \epsilon, d$ )
  ranking,  $S \leftarrow []$ 
   $\mathbf{V}_{train}, \mathbf{V}_{test} \leftarrow \text{create\_train\_test}(\mathbf{V})$ 
   $\mathbf{w}_0^*, C^*, \epsilon^*, d^* \leftarrow \text{SVF}(\mathbf{V}_{train}, C, \epsilon, d)$ 
   $E_0 \leftarrow \text{calculate\_error}(\mathbf{V}_{test}, \mathbf{w}_0^*)$ 
  for  $u \leftarrow 1$  to  $n_{inp}$  do
     $\mathbf{V}_{train}^u \leftarrow \text{permute\_var}(\mathbf{V}_{train}, u)$ 
     $\mathbf{w}_u^* \leftarrow \text{SVF}(\mathbf{V}_{train}^u)$ 
     $E_u \leftarrow \text{calculate\_error}(\mathbf{V}_{test}, \mathbf{w}_u^*)$ 
     $S_u \leftarrow 100 \cdot \frac{E_u - E_0}{E_0}$ 
  end for
  ranking = order_score_asc( $S$ )
  return ranking
end procedure

```

Algoritmo 2: Implementación del algoritmo SVF-PFI

4.2.2. SVF-BOOTSTRAP RESAMPLING

Basándonos en las ideas de [Alonso et al. \(2012\)](#), hemos desarrollado un método que utiliza el remuestreo bootstrap para la selección de características adaptado para SVF. A partir de un conjunto de datos inicial \mathbf{V} , se aplica el remuestreo bootstrap para crear B submuestras $\mathbf{V}'(b)$. Cada una de las submuestras $\mathbf{V}'(b)$ se compone de un conjunto de datos de entrenamiento, aquellos datos que se han incluido en el remuestreo \mathbf{V}'_{in} y un conjunto de datos de prueba, aquellos que se han excluido del remuestreo (\mathbf{V}'_{out}). Una vez generada la submuestra, el conjunto de entrenamiento se utiliza para entrenar un modelo SVF. Este modelo generará hiperparámetros óptimos: (C^*, ϵ^*, d^*) y una solución óptima: (\mathbf{w}^*, b_0^*) . El conjunto de prueba se utiliza para obtener el error de submuestra: $R^*(\mathbf{w}^*, \mathbf{V}'_{out})$.

A continuación, se itera el método eliminando una a una cada una de las entradas del problema del conjunto de datos \mathbf{V}' , generando el subconjunto \mathbf{V}'_u . Para el conjunto de prueba resultante de eliminar la variable u , y se calcula el error de predicción. Seguidamente, se calcula el error de la variable restando el error de la solución del conjunto original, $R^*(b)$, al valor del error de la solución del conjunto sin la variable u , $R_u^*(b)$.

Una vez obtenidos todos los errores de todas las variables de todas las pseudomuestras, se calculan las veces en las que el error de la variable es mayor o igual que cero o negativo. Si el número de veces en que el valor es positivo es grande, indica que la variable es muy importante en el conjunto de datos. Si, por el contrario, el valor del error de esa variable es siempre negativo, indicará que la variable examinada no es útil o tiene poca importancia.

Una vez determinadas las pseudomuestras B , se calcula la frecuencia relativa en la que cada variable es positiva y se considera que la que tiene el valor más alto es la más importante.

```

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

```

Algoritmo 3: Implementación del algoritmo SVF-BR

4.2.3. SVF-RFE PSEUDOSAMPLES

SVF-RFE Pseudosamples es la metodología que proponemos para el ranking de entradas que desarrollamos basándonos en las ideas de [Sanz et al. \(2018\)](#). Esta técnica de ranking de características establece un orden de importancia de las mismas basado en el valor de la Desviación Absoluta Mediana (MAD). Los pasos a seguir se muestran en el Algoritmo 4. En primer lugar, se establece un criterio de parada. En nuestro caso, dado que nuestro objetivo es obtener un ranking completo, debemos definir como criterio de parada que el número de inputs del problema sea mayor que 1. De esta forma, se evalúan todos los inputs y el método obtiene un ranking completo de los mismos. Mientras no se cumpla el criterio de parada, se ejecuta un modelo SVF con el conjunto de variables disponibles en ese momento. Además, se obtiene una solución óptima del modelo (\mathbf{w}^*, b_0^*). Se crea una matriz con las pseudomuestras de la variable a evaluar (\mathbf{V}_u). Para ello, seguimos el

método establecido por [Sanz et al. \(2018\)](#) que utiliza q cuantiles de la variable a evaluar y deja el resto de columnas con ceros o la media de la columna, en caso de que los datos no estén normalizados. Calculamos su predicción (\hat{y}_{uq}) y evaluamos el valor de MAD como $MAD_u = \text{median}(|\hat{y}_{uq} - \text{median}(y_u)|)$, donde y_u es el valor de las salidas en el conjunto de datos original. La variable con el MAD más bajo se elimina del conjunto de datos y se repite el mismo proceso hasta que se cumpla el criterio de parada. Por último, obtenemos una clasificación de todas las entradas que indica la importancia de cada una de ellas en el modelo.

```

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

```

Algoritmo 4: Implementación del algoritmo SVF-Pseudosamples

4.2.4. SVM-RFE DUAL OBJECTIVE VARIATION

En este caso, comenzamos configurando y resolviendo el modelo (32), y utilizamos un procedimiento de validación cruzada para obtener los hiperparámetros óptimos (C^*, ϵ^*, d^*) , que fijamos. El dual del modelo (32) es un problema de maximización cuya función objetivo $-J$, es equivalente al siguiente modelo:

$$\begin{aligned}
 \min_{\alpha, \gamma, \delta} \quad & J = \frac{1}{2} \sum_{r=1}^s \left(\|\mathbf{w}^{(r)}\|_2^2 + \epsilon^* \sum_{i=1}^n \delta_i^{(r)} - \sum_{i=1}^n (\alpha_i^{(r)} - \delta_i^{(r)}) y_i^{(r)} \right) \quad (.0) \\
 \text{s.t.} \quad & \alpha_i^{(r)} \geq 0 \quad i = 1, \dots, n, r = 1, \dots, s \quad (.1) \\
 & \gamma^{(r)} \geq 0, \quad r = 1, \dots, s \quad (.2) \\
 & 0 \leq \delta_i^{(r)} \leq C^* \quad i = 1, \dots, n, r = 1, \dots, s \quad (.3)
 \end{aligned} \quad (34)$$

En el modelo (34), tenemos $\mathbf{w}^{(r)} = \sum_{i=1}^n (\alpha_i^{(r)} - \delta_i^{(r)}) \phi_{SVF}^G(\mathbf{x}_i) + \gamma^{(r)}$, $\forall r \in \{1, \dots, s\}$. Las variables α, δ son las variables duales correspondientes a las restricciones 30.1 y 30.2, mientras que γ corresponde a la restricción 30.3. La variable β correspondiente a la restricción 30.4 da el límite superior de δ , y satisface $\beta_i^{(r)} + \delta_i^{(r)} = C$ para cada componente. Por lo tanto, se puede sustituir en la formulación dual y no aparece en el modelo.

Sustituyendo la expresión para \mathbf{w} en el objetivo J del modelo (34), obtenemos:

$$\begin{aligned}
 J = & \frac{1}{2} \sum_{r=1}^s \left[\sum_{i=1}^n \sum_{j=1}^n (\alpha_i^{(r)} - \delta_i^{(r)}) (\alpha_j^{(r)} - \delta_j^{(r)}) \langle \phi_{SVF}^G(\mathbf{x}_i) \cdot \phi_{SVF}^G(\mathbf{x}_j) \rangle \right. \\
 & + 2 \left(\sum_{i=1}^n (\alpha_i^{(r)} - \delta_i^{(r)}) \langle \phi_{SVF}^G(\mathbf{x}_i) \cdot \gamma \rangle \right) \\
 & \left. + \sum_{k=1}^{(d^*)^m} \gamma_k^{(r)2} + \epsilon^* \sum_{i=1}^n \delta_i^{(r)} - \sum_{i=1}^n (\alpha_i^{(r)} - \delta_i^{(r)}) y_i^{(r)} \right]. \quad (35)
 \end{aligned}$$

Resolvemos el modelo (34) con los hiperparámetros óptimos C^*, ϵ^*, d^* del primal y obtenemos los valores óptimos $\alpha^*, \delta^*, \gamma^*$ que luego mantendremos fijos.

Al considerar el efecto de eliminar una variable de entrada u manteniendo $\alpha^*, \delta^*, \gamma^*$ constantes, los únicos términos que cambiarán son los que implican la función de transformación $\phi_{SVF}^G(\mathbf{x}_i)$. Denotamos la transformación sin variable u por $\phi_{SVF}^G(\mathbf{x}_i(-u))$, y el valor correspondiente de J por $J(u)$.

La expresión para $DJ(u)$ viene dada por:

$$\begin{aligned}
DJ(u) = J - J(u) = \\
\frac{1}{2} \sum_{r=1}^s \left[\left(\sum_{i=1}^n \sum_{j=1}^n (\alpha_i^{*(r)} - \delta_i^{*(r)}) (\alpha_j^{*(r)} - \delta_j^{*(r)}) \left(\langle \phi_{SVF}^G(\mathbf{x}_i) \cdot \phi_{SVF}^G(\mathbf{x}_j) \rangle - \langle \phi_{SVF}^G(\mathbf{x}_i(-u)) \cdot \phi_{SVF}^G(\mathbf{x}_j(-u)) \rangle \right) \right) \right] \quad (36) \\
+ 2 \left(\sum_{i=1}^n (\alpha_i^{*(r)} - \delta_i^{*(r)}) \left(\langle \phi_{SVF}^G(\mathbf{x}_i) - \phi_{SVF}^G(\mathbf{x}_i(-u)) \rangle \cdot \gamma^* \right) \right)
\end{aligned}$$

Observamos que sólo los términos que implican \mathbf{x}_i que tienen $\alpha_i^{*(r)} - \delta_i^{*(r)} \neq 0$ (es decir, los vectores soporte) aparecerán en el cálculo final. Esto significa que sólo necesitamos considerar un subconjunto de los datos, que normalmente será mucho más pequeño que el conjunto de datos completo, para calcular este valor.

Para calcular $DJ(u)$, ahora describiremos cómo cambia $\phi_{SVF}^G(\mathbf{x}_i)$ cuando se elimina una variable. Recordemos que la función de transformación ϕ_{SVF}^G toma valores $\{0,1\}$ en una rejilla m -dimensional de tamaño $(d^*)^m$. Al eliminar una entrada, proyectamos los datos en un espacio $(m-1)$ -dimensional, y proyectamos los valores en cada celda de la rejilla en consecuencia, eliminando la contribución de la variable u . Sea un \mathbf{x}_i perteneciente a la celda $C_{l_1 \dots l_m}$, donde cada $l_j \in \{1, \dots, d^*\}$. El efecto de eliminar la entrada u en $\phi_{SVF}^G(\mathbf{x}_i)$ es que cada celda cuyo índice l_u es mayor que 1 toma el valor de 0, eliminando así el efecto de la variable u en la rejilla. El resto de las celdas conservan su valor original.

Calculamos $DJ(u)$ mediante la ecuación (36) a medida que u recorre todos los inputs que vamos clasificando. La variable que alcanza el valor más pequeño de $DJ(u)$ se considera la menos importante, se clasifica en último lugar entre las restantes y se elimina del modelo. El método se repite hasta que se hayan clasificado todas las variables. El Algoritmo 5 muestra los pasos principales del método.

```

procedure GET_RANKING_SVF_DOV(V,C, $\epsilon$ ,d)
  ranking  $\leftarrow$  []
  DJ  $\leftarrow$  []
  while !stop_criteria do
     $C^*$ ,  $\epsilon^*$ ,  $d^*$   $\leftarrow$  SVF(V, C,  $\epsilon$ , d)
     $\alpha^*$ ,  $\delta^*$ ,  $\sigma^*$   $\leftarrow$  SVF_Dual(V,  $C^*$ ,  $\epsilon^*$ ,  $d^*$ )
    for  $u \leftarrow 1$  to  $m$  do
       $DJ_u \leftarrow$  calculate_dj( $\alpha^*$ ,  $\delta^*$ ,  $\sigma^*$ , V,  $u$ )
    end for
    var_to_drop  $\leftarrow$  min(DJ)
    V  $\leftarrow$  drop_var(var_to_drop)
    ranking  $\leftarrow$  insert_var_to_ranking(ranking, var_to_drop)
  end while
  return ranking
end procedure

```

Algoritmo 5: Implementación del algoritmo SVF-DOV

4.2.5. EXPERIENCIA COMPUTACIONAL

En esta sección, vamos a medir el rendimiento de los métodos expuestos anteriormente a la hora de establecer un ranking de los inputs del proceso de producción. Además, evaluaremos el coste computacional. Para ello, se han desarrollado diferentes escenarios experimentales (ver Tabla 11). En concreto, se ha simulado una función de producción Cobb-Douglas con tres inputs (x_1 , x_2 , x_3) y un output (y) mediante la siguiente fórmula $y = x_1^a x_2^b e^{-I}$ y considerando la variable x_3 de forma independiente, es decir, la variable x_3 no forma parte del proceso productivo. Los inputs se han generado aleatoriamente siguiendo una distribución uniforme con un rango entre (1,10). En el cálculo de las ineficiencias (I) se han utilizado distintas distribuciones con distintas configuraciones (ver columna Ineficiencia en Tabla 11). La correlación entre las variables x_1 y x_2 (ρ_{x_1, x_2}) y la contribución a la producción de cada input, es decir, el valor de los coeficientes a y b en la función de producción Cobb-Douglas, se muestran en las demás columnas de la Tabla 11.

El primer escenario es el caso base. A partir de él, se modifican las características para analizar el comportamiento de los métodos con datos diferentes. En los experimentos 2 y 3, se modifica el tamaño de la muestra, fijándolo en 30 y 80 DMUs, respectivamente. En

los dos experimentos siguientes, se estudia el ranking modificando el valor de la correlación. Para establecer la correlación, la ecuación utilizada viene dada por: $x_2 = \rho_{x_1x_2}x_1 + v\sqrt{1-\rho_{x_1x_2}^2}$, donde v es una variable aleatoria obtenida a partir de una distribución $v \sim \text{Uni}(1,10)$, siguiendo el método de Wang y Schmidt (2002). En el escenario 6, estudiamos el rendimiento de los algoritmos cuando la contribución de las variables tipo input es similar. Finalmente, para comprobar la robustez de los métodos, se ha modificado la distribución de probabilidad de las ineficiencias, considerando una ineficiencia media mayor, así como una distribución exponencial.

Escenario	Correlación	Ineficiencia	Contribución	Detalles
1				Caso base, $n = 50$
2	Generados independientemente	$I \sim N(0, 0.4) $	$a = 0.6, b = 0.3$	$n = 30$
3				$n = 80$
4				
5	$x_2 = \rho_{x_1x_2}x_1 + v\sqrt{1-\rho_{x_1x_2}^2}$	$I \sim N(0, 0.4) $	$a = 0.6, b = 0.3$	Inputs bajamente correlados, $\rho_{x_1x_2} = 0.2$ Inputs altamente correlados, $\rho_{x_1x_2} = 0.8$
6	Generados independientemente	$I \sim N(0, 0.4) $	$a, b = 0.45$	x_1 y x_2 tienen igual importancia
7		$I \sim N(0, 0.8) $		
8	Generados independientemente	$I \sim \exp(0.4)$	$a = 0.6, b = 0.3$	Ineficiencia diferente
9				

Tabla 11: Lista de experimentos realizados

Realizamos 50 simulaciones de cada experimento para cada uno de los cuatro métodos desarrollados anteriormente y calculamos el porcentaje de veces que cada variable alcanza una posición en la clasificación. Debido a la naturaleza del modelo, fue necesario establecer un grid de hiperparámetros y hacer una selección a partir de ellos. El grid constaba de los siguientes valores: $C = [0, 0.001, 0.01, 0.1, 1, 100, 1000]$, $\epsilon = [0, 0.001, 0.01, 0.1, 0.2, 0.5, 1]$, $d = 0.1 \cdot h \cdot n$, donde $h = 1, \dots, 10$ y n es el tamaño de la muestra. Para obtener los mejores hiperparámetros, el conjunto de datos original se dividió en una muestra de entrenamiento con el 80 por ciento de las observaciones y una muestra de prueba compuesta por el 20 por ciento restantes de observaciones. Mientras que para el método SVF-Pseudomuestras, se utilizó $q = 10$ para crear las pseudomuestras.

El método SVF-BR se intentó implementar siendo imposible obtener resultados debido al coste computacional de ejecutar tantas submuestras.

		TAMANO MUESTRAL									CORRELACIÓN					
		1			2			3			4		5			
		1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
SVF-Pseudosamples	x_1	88%	12%	0%	92%	4%	4%	96%	4%	0%	92%	8%	0%	94%	6%	0%
	x_2	10%	78%	12%	8%	80%	12%	4%	94%	2%	6%	84%	10%	6%	72%	22%
	x_3	2%	10%	88%	0%	16%	84%	0%	2%	98%	2%	8%	90%	0%	22%	78%
SVF-PFI	x_1	98%	2%	0%	76%	22%	2%	74%	26%	0%	98%	2%	0%	90%	10%	0%
	x_2	2%	82%	16%	20%	50%	30%	24%	66%	10%	2%	80%	18%	10%	68%	22%
	x_3	0%	16%	84%	4%	28%	68%	2%	8%	90%	0%	18%	82%	0%	22%	78%
SVF-DOV	x_1	84%	16%	0%	82%	18%	0%	86%	14%	0%	76%	24%	0%	72%	28%	0%
	x_2	12%	72%	16%	18%	64%	18%	10%	86%	4%	16%	70%	14%	24%	70%	6%
	x_3	4%	12%	84%	0%	18%	82%	4%	0%	96%	8%	6%	86%	4%	2%	94%
		IGUAL PESO						INEFICIENCIAS								
		6			7			8			9					
		1	2	3	1	2	3	1	2	3	1	2	3			
SVF-Pseudosamples	x_1	36%	64%	0%	90%	10%	0%	86%	14%	0%	84%	12%	4%			
	x_2	64%	36%	0%	10%	76%	14%	12%	76%	12%	16%	70%	14%			
	x_3	0%	0%	100%	0%	14%	86%	2%	10%	88%	0%	18%	82%			
SVF-PFI	x_1	66%	34%	0%	90%	6%	4%	90%	8%	2%	88%	8%	4%			
	x_2	34%	64%	2%	10%	78%	12%	10%	74%	16%	12%	58%	30%			
	x_3	0%	2%	98%	0%	16%	84%	0%	18%	82%	0%	34%	66%			
SVF-DOV	x_1	46%	54%	0%	76%	18%	6%	84%	14%	2%	80%	14%	6%			
	x_2	50%	44%	6%	14%	62%	24%	14%	78%	8%	18%	72%	10%			
	x_3	4%	2%	94%	10%	20%	70%	2%	8%	90%	2%	14%	84%			

Tabla 12: Tabla de resultados

En la Tabla 12 se muestran los resultados (en porcentaje) obtenidos por cada uno de los algoritmos evaluados. La forma de obtener el porcentaje ha sido calculando el número de repeticiones en las que una variable ocupaba una determinada posición en el ranking de las 50 simulaciones. Los valores 1, 2, 3 indican la posición del ranking en la que quedó cada una de las tres variables (x_1, x_2, x_3) incluidas en el problema. El algoritmo SVM-BR no aparece debido a su elevado coste computacional.

En primer lugar, evaluamos los resultados asociados a los escenarios en los que existe variación en el número de DMUs. Se puede observar que tanto el algoritmo SVF-Pseudosamples como el SVF-DOV mejoran su rendimiento a medida que aumenta el número de DMUs. Sin embargo, el efecto en el método SVF-PFI es menos pronunciado.

En segundo lugar, evaluamos los escenarios en los que los inputs relevantes en el proceso de producción están correlacionados. En el escenario 4, las variables x_1 y x_2 están poco correlacionadas entre sí ($\rho_{x_1, x_2} = 0.2$), mientras que en el escenario 5 las variables están

más correlacionadas ($\rho_{x_1, x_2} = 0.8$). En estos escenarios, el método SVF-DOV obtiene buenos resultados, ya que es capaz de detectar con gran precisión que la variable x_3 es poco relevante. Respecto a los otros métodos, aunque aciertan al elegir la variable x_1 como la más relevante en un alto porcentaje de las simulaciones, no son capaces de establecer un ranking correcto al situar la variable x_3 como la segunda más relevante en el 20% de las simulaciones del escenario 5.

Además, analizamos en el escenario 6 cómo funcionan los métodos cuando las variables relevantes tienen la misma importancia ($a = b = 0.45$). En este sentido, el método SVF-DOV es el que mejor funciona en estos casos, ya que prácticamente siempre detecta que la variable x_3 es la menos relevante y, además, elige las variables x_1 y x_2 como las más relevantes en un porcentaje aproximado de 50%. Los demás métodos tienen un rendimiento similar, pero tienden a clasificar las variables relevantes en un orden determinado.

Por último, evaluamos la robustez de los métodos al cambiar las ineficiencias mediante diferentes distribuciones y varianzas. En el escenario 7, se utiliza una distribución seminormal con una ineficacia media alta. Los resultados muestran que los porcentajes de selección de la variable x_3 como variable menos importante son altos en SVF-Pseudomuestras y SVF-PFI, pero más bajos en el método SVF-DOV. En los escenarios 8 y 9, en los que modificamos la distribución seminormal por una exponencial, se puede observar como los métodos SVF-Pseudomuestras y SVF-DOV tienen un buen comportamiento mientras que el SVF-PFI tiene un comportamiento menos preciso.

En resumen, al evaluar todos los experimentos, se puede observar que el método SVF-Pseudomuestras se comporta bien en escenarios en los que el tamaño de la muestra y la distribución de las ineficiencias están alterados, mientras que en escenarios en los que las variables están más correlacionadas o cuando las variables tienen la misma contribución a la salida, este método no funciona tan bien. El método SVF-PFI se comporta de forma similar al SVF-Pseudomuestras, aunque no funciona tan bien en general. Muestra menos mejoras al aumentar el tamaño de la muestra, y se ve más afectado por los cambios en la

distribución de las ineficiencias. Por último, el método SVF-DOV obtiene peores resultados en escenarios en los que se modifican los tamaños de muestra y la distribución de las ineficiencias, pero presenta buenos resultados cuando las variables están muy correlacionadas o tienen el mismo peso, en parte debido a su capacidad para distinguir la variable x_3 como la menos importante incluso cuando la correlación entre las variables relevantes es alta (escenario 5).

4.2.6. COSTE COMPUTACIONAL

Para evaluar el coste computacional de cada método, se ha evaluado el tiempo de ejecución de cada técnica aplicando diferentes tamaños de muestra (véanse los escenarios 1-3 de la Tabla 11).

Para el cálculo se ha utilizado un clúster de estaciones de trabajo. Este clúster utiliza un sistema operativo Rocky Linux versión 8.6 (Green Obsidian) y dispone de 80 núcleos, 755 GB de RAM y 2 CPU Intel(R) Xeon(R) Silver 4316 a 2,30GHz. Los trabajos se lanzaron en paralelo, utilizándose 1 núcleo para cada simulación de los escenarios.

La Tabla 13 muestra el tiempo medio de ejecución de cada uno de los algoritmos aplicados a 30, 50 y 80 DMUs. Como se muestra en la tabla, el método SVF-BR no ha podido ser simulado porque tiene que realizar 100 remuestreos para cada escenario. Por ejemplo, en tamaños de muestra de 30 DMUs, aunque el algoritmo presentó buenos resultados, tardó una media aproximada de 10 horas en cada una de las 50 simulaciones. Además, el método que demostró ser más rápido en la mayoría de los escenarios fue el método SVF-DOV, seguido del SVF-Pseudomuestras y del SVF-PFI. En base al porcentaje de mejora de la Tabla 12 y al tiempo de ejecución de cada algoritmo, en escenarios con baja correlación, se recomienda utilizar el modelo SVF-Pseudomuestras, ya que es el segundo método más rápido y obtiene buenos resultados a la hora de establecer un ranking de variables. Por otro lado, en el caso en el que los datos puedan

estar altamente correlacionados, se recomienda utilizar el método SVF-DOV, ya que es un método que obtiene buenos resultados en este tipo de escenarios.

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

Tabla 13: Tabla de tiempos de ejecución

4.3. SUPPORT VECTOR MACHINE Y SELECCIÓN DE CARACTERÍSTICAS

En esta sección se muestra, en primer lugar, el nuevo modelo propuesto para resolver el problema de Support Vector Machine estándar con selección de características. Este nuevo modelo tendrá un enfoque novedoso en el que en lugar de tener en cuenta los objetivos tradicionales de maximizar distancia de los hiperplanos separadores y minimizar la distancia de las observaciones mal clasificadas a su hiperplano separador, tendrá en cuenta dos objetivos distintos basados en las métricas mostradas en la [Sección 3.6.2](#). Esto dará como resultado una frontera Pareto con diferentes soluciones. Debido al elevado coste computacional requerido por las técnicas exactas para resolver el problema, es necesario aplicar una metaheurística. Por ello, se ha hecho uso de la desarrollada por [Alcaraz et al. \(2022\)](#) mostrada en la [Sección 3.6.1](#). Finalmente, se presentará un nuevo modelo en el que se propone optimizar, simultáneamente tres objetivos: los dos objetivos del problema clásico a los que añadimos el de minimizar el coste de las características seleccionadas. Para resolver este problema, hemos adaptado la metaheurística al nuevo problema tri-objetivo.

4.3.1. SVM Y SELECCIÓN DE CARACTERÍSTICAS TENIENDO EN CUENTA MÉTRICAS BASADAS EN LA MATRIZ DE CONFUSIÓN

Como se ha mostrado en la [Sección 3.6](#), el enfoque tradicional del problema de Support Vector Machines con selección de características se puede solucionar aplicando el modelo propuesto en (12). El resultado es una frontera Pareto en la que cada solución es un clasificador que puede ser evaluado por las métricas expuestas en la [Sección 3.6.2](#).

Esas métricas se basan en su mayoría en la cantidad de falsos positivos y falsos negativos asociados al clasificador. Por eso nos pareció natural proponer un nuevo modelo multi-objetivo en el que en lugar de establecer como objetivos simultáneos los utilizados habitualmente (distancia y suma de errores), se optimizaran el número de falsos positivos y negativos, ambos a minimizar.

Basándonos en eso, proponemos un nuevo modelo bi-objetivo alternativo al clásico que proporciona mejores resultados que los clasificadores anteriores cuando se utilizan las métricas de rendimiento basadas en la matriz de confusión. En este nuevo modelo, los objetivos son minimizar tanto la Predicciones Negativas Falsas (FN) como las Predicciones Positivas Falsas (FP).

$$\min \quad O_{21} = \sum_{i:y_i=1} z_i \quad (.1)$$

$$\min \quad O_{22} = \sum_{i:y_i=-1} z_i \quad (.2)$$

$$\text{s.t.} \quad y_i(w^T x_i + b) \geq 1 - \xi_i \quad (.3)$$

$$\xi_i \geq 0 \quad i = 1, \dots, n, \quad (.4)$$

$$w_j \in \mathbb{R} \quad j = 1, \dots, m, \quad (.5)$$

$$b \in \mathbb{R} \quad (.6)$$

$$\sum_j t_j = p \quad j = 1, \dots, m, \quad (.7)$$

$$|w_j| \leq Mt_j \quad j = 1, \dots, m, \quad (.8)$$

$$t_j \in \{0, 1\} \quad j = 1, \dots, m, \quad (.9)$$

$$\xi_i - 1 \leq Mz_i \quad i = 1, \dots, n, \quad (.10)$$

$$z_i \in \{0, 1\} \quad i = 1, \dots, n. \quad (.11)$$

(37)

Para solucionar este problema se ha adaptado la metaheurística propuesta por [Alcaraz et al. \(2022\)](#) expuesta en la [Sección 3.6.1](#). El desarrollo de esta metaheurística va a permitir la comparación de ambos modelos, en los que cambian sólo los objetivos a optimizar:

- Modelo 1: el primer objetivo consiste en la maximización de la distancia entre los dos hiperplanos paralelos que soportan algunos vectores de ambas clases (O'_{11}). El segundo objetivo es minimizar la suma de las distancias de los vectores mal clasificados a su hiperplano correspondiente (O'_{12}).

- Modelo 2: se trata la minimización de los vectores mal clasificados de una de las clases O_{21} y el otro la suma de vectores mal clasificados de la otra clase O_{22} .

4.3.1.1. EXPERIENCIA COMPUTACIONAL

Con el fin de comparar el rendimiento de los clasificadores proporcionados por las metaheurísticas basadas en el Modelo 1 y el Modelo 2, llevamos a cabo una serie de experimentos computacionales resolviendo diferentes conjuntos de datos. Para ello, calculamos tres métricas de rendimiento basadas en la matriz de confusión (FSC, AUC y KAP). Ambos algoritmos metaheurísticos han sido implementados en C++ y los experimentos se han llevado a cabo en el Clúster de Computación Científica de la Universidad Miguel Hernández (UMH). Este equipo informático utiliza el sistema operativo CentOS Linux release 7.5.1804 que está compuesto por cinco nodos con las siguientes características.

- Nodo 0: modelo Bull X410 con dos procesadores Xeon Planitum v5 8160-2.1 Ghz (en total 96 núcleos), 384 Gigabyte de RAM y cuatro GPUs Nvidia V100.
- Nodo 1: similar al Nodo 0 con 768 Gigabytes de RAM.
- Nodo 2 y 3: modelo Dell PowerEdge R440 con dos procesadores Xeon Gold 5120 a 2,2 GHz (en total 56 núcleos) y 128 Gigabytes de RAM.
- Nodo 4: modelo Supermicro SYS-1029GQ-TRT con dos procesadores Xeon Gold 6230N a 2,3 GHz (en total 80 núcleos), 640 Gigabytes de RAM y tres GPU (una NVidia V100 y dos NVidia A100).

Las simulaciones utilizaron el gestor de cargas de trabajo SLURM asignando un núcleo a cada conjunto de datos.

Para nuestros experimentos computacionales, consideramos varias instancias con diferentes características. Los conjuntos de datos Arcene, Arrythmia, GC, Gisette, Housing, IONO, Isolet, Madelon, MFeat y WBC se encuentran en el repositorio UCI (Dua, D., Graff, C., (2017)). La instancia Duke se puede encontrar en la biblioteca LIBSVM (Chang and Li (2011)), Gina_agnostic está disponible en el trabajo de (Guyon et al. (2007)) y Bioresponse fue utilizado por (Abdunabi y Basir (2014)). Las principales características de los conjuntos de datos se pueden ver en la Tabla 14. La primera columna

muestra el nombre de la instancia. La segunda columna indica el número de vectores de la muestra. Finalmente, la última columna muestra el número de características. Estos conjuntos de datos representan un conjunto muy heterogéneo, ya que proceden de contextos muy diferentes y tienen características muy distintas en cuanto al número de vectores y características.

Instancia	Vectores (n)	Características (m)
Housing	506	13
GC	1000	24
WBC	569	30
Iono	351	33
Arrhythmia	420	278
Madelon	2600	500
Isolet	600	617
Mfeat	2000	649
Gina_agnostic	3468	970
Bioresponse	3751	1776
Gisette	6000	4952
Duke	44	7129
Arcene	200	10000

Tabla 14: Características de las instancias usadas

En cuanto a la configuración del NSGA-II, hemos utilizado dos tamaños de población diferentes, $N = 100$ y $N = 50$; y tres niveles diferentes de tiempo de CPU, medido en segundos, como criterio de parada: $t = 1200$, 3600 y 10800 segundos. La combinación de estos dos parámetros nos da 6 versiones diferentes de cada algoritmo y cada instancia se ha resuelto con todas las versiones de ambos algoritmos. Todos los demás parámetros de la metaheurística se han ajustado a los valores utilizados por [Alcaraz et al. \(2022\)](#), dado que realizaron varios experimentos para configurar adecuadamente el algoritmo. Sin embargo, los autores explican que el rendimiento de la metaheurística no se ve muy influenciado por ligeras variaciones en los valores de estos parámetros.

Para resolver una instancia con cada una de las versiones de las metaheurísticas propuestas hemos llevado a cabo un procedimiento de hold-out estratificado con

repetición. Hemos generado 5 particiones diferentes de cada una de las instancias a resolver con $\frac{2}{3}$ de los vectores que corresponden a los 5 conjuntos de datos de entrenamiento, Train_1, ..., Train_5. Para cada conjunto de datos de entrenamiento Train_i, el resto de los vectores del conjunto de datos original no incluidos en el conjunto de entrenamiento, forman el conjunto de datos de prueba correspondiente, Test_i. En todas estas divisiones, conjuntos de entrenamiento y de prueba, el porcentaje de vectores de cada clase se mantiene como en el conjunto de datos original (esto se conoce como partición estratificada). Para cada una de las particiones, hemos ejecutado el algoritmo 3 veces independientes, obteniendo un conjunto de soluciones no dominadas por ejecución. El número de soluciones no dominadas puede variar de una ejecución a otra. Cada una de estas soluciones representa un clasificador. Los clasificadores obtenidos en una ejecución de la metaheurística, han sido probados con el conjunto de prueba correspondiente. La prueba de un clasificador consiste en calcular cada una de las medidas consideradas para realizar una comparación desde diferentes perspectivas. Estas medidas son: el Área Bajo la Curva (AUC), el F-Score (FSC) y la medida Kappa de Cohen (KAP). Por lo tanto, para cada ejecución hemos calculado tres medidas diferentes. Dada una medida, se selecciona el mejor valor para esa medida en cada ejecución. A continuación, calculamos la media y el máximo de estos tres valores, como representativos de la división. Por lo tanto, tenemos 5 máximos y 5 medias diferentes para cada una de las medidas. Finalmente, la media de cada conjunto de 5 valores nos da las dos puntuaciones que obtiene una metaheurística determinada al resolver una instancia. El procedimiento que calcula las puntuaciones de la medida para una instancia y una versión dada de la metaheurística se ha representado en la Figura 14.

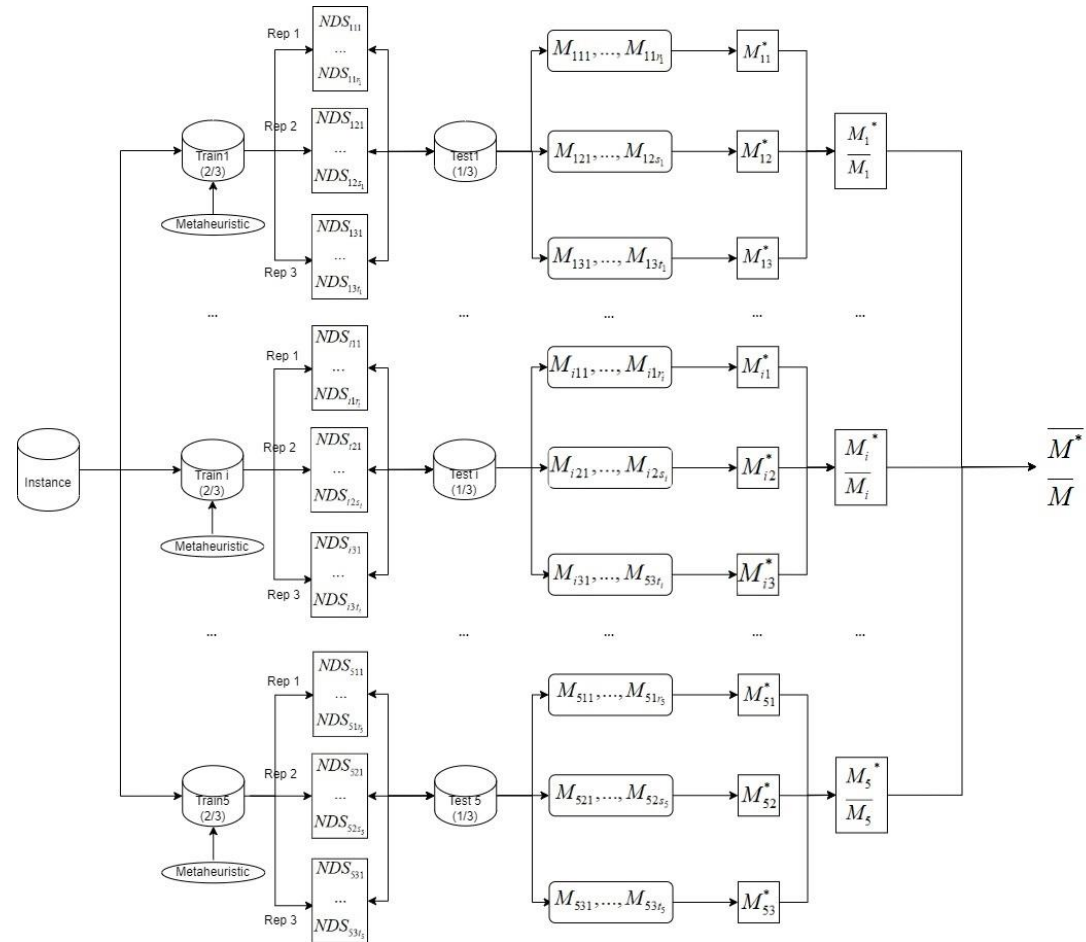


Figura 14: Hold-out estratificado con repeticiones

Las Tabla 15, 16, 17 y 18 muestran los resultados computacionales detallados de las tres medidas de interés (AUC, FSC y KAP) para los distintos conjuntos de datos. Las dos primeras tablas corresponden a los conjuntos de datos con 500 características o menos y las dos últimas tablas a los conjuntos de datos con un mayor número de características. Los resultados del Modelo 1 se presentan en las Tabla 15 y 17 y los resultados del Modelo 2 en las Tabla 16 y 18. Se registran las mismas tres medidas al ejecutar ambos algoritmos. El mismo conjunto de datos se resuelve con seis combinaciones diferentes de cada metaheurística, combinando el límite de tiempo (columna TIEMPO) y el tamaño de la población (columna TAMAÑO). Dado que para cada combinación de algoritmo, tiempo y tamaño tenemos diez valores diferentes para las tres medidas (cinco medias y cinco máximos para cada conjunto de datos de entrenamiento), hemos registrado tanto la media de las cinco medias (\bar{M}) como la media de los cinco máximos (\bar{M}^*).

Los valores en negrita en las tablas indican que la métrica para ese algoritmo es mejor que la del otro algoritmo con el que se compara. Por ejemplo, los valores de la primera fila de las Tabla 15 y Tabla 16 son los resultados para el conjunto de datos Arrhythmia cuando el límite de tiempo se establece en 1200 segundos y el tamaño de la población del algoritmo genético se establece en 50. En este caso, la medida AUC media para los cinco conjuntos de datos de entrenamiento (\bar{M}) es 0,708 para el Modelo 1 (Tabla 15) y 0,750 para el Modelo 2 (Tabla 16), la medida FSC media es 0,792 para el Modelo 1 y 0,811 para el Modelo 2; el resultado medio de KAP es 0,430 para el Modelo 1 y 0,508 para el Modelo 2; la media de las mejores AUC (\bar{M}^* , media de los cinco valores máximos de los conjuntos de datos de entrenamiento) es 0,721 para el Modelo 1 y 0,753 para el Modelo 2; la media de las mejores FSC 0,799 para el Modelo 1 y 0,814 para el Modelo 2; la media de la mejores KAP es 0,452 para el Modelo 1 y 0,514 para el Modelo 2. Las filas restantes ofrecen los mismos valores para el resto de combinaciones de conjuntos de datos, límite de tiempo y tamaño. Así, las tablas proporcionan un gran número de valores que se utilizan para comparar la bondad de los dos algoritmos. Que un algoritmo sea mejor en valores medios no implica que lo sea en valores máximos, ni que un algoritmo sea mejor para una combinación de

tiempo y tamaño implica que lo sea para otra, aunque mantengamos la instancia. La última fila muestra la media de las diferentes métricas calculadas para todos los conjuntos de datos presentados en la tabla. Si comparamos la media para las diferentes métricas mostradas en las Tabla 15 y 16, nos damos cuenta de que, para los conjuntos de datos con 500 características o menos el Modelo 2 da, en términos generales, mejores clasificadores que el Modelo 1, independientemente de la métrica considerada. Por ejemplo, la media de AUC para el Modelo 1 es de 0,77 frente al 0,79 que proporciona el Modelo 2. Los resultados proporcionados por el Modelo 2 son, para todas las métricas consideradas, mejores que los presentados por el Modelo 1 en todos los conjuntos de datos del primer grupo (los que tienen 500 características o menos).

Los resultados mostrados en las Tabla 17 y 18 Tabla 18 permiten comparar ambos algoritmos al resolver los conjuntos de datos con más de 500 características. Si nos centramos de nuevo en la última fila de estas tablas, nos damos cuenta de que el Modelo 2 proporciona mejores clasificadores que el Modelo 1 cuando resuelve los conjuntos de datos con más de 500 características, independientemente de la métrica considerada. En este caso, las diferencias entre ambos algoritmos parecen ser mayores que las de los conjuntos de datos con un menor número de características. Por lo tanto, del análisis de estas tablas podemos concluir que, en todos los conjuntos de datos considerados, el Modelo 2 obtiene, en general, mejores resultados que el Modelo 1 para todas las métricas calculadas. Sin embargo, la densidad de estas cuatro tablas hace difícil extraer conclusiones rápidas. Para facilitar la interpretación de los resultados, hemos calculado una tabla de frecuencias de éxito que se presentan en la Tabla 19, donde cada fila corresponde a una combinación de tiempo y tamaño de población, la primera fila es para 1200 segundos y un tamaño de población de 50, la segunda fila es para 1200 segundos y tamaño de población igual a 100, ..., la última es para 10800 segundos y 100 individuos. Las columnas se agrupan en dos bloques, el bloque de la izquierda compara los dos algoritmos en función de los valores medios de AUC, FSC y KAP obtenidos en las distintas ejecuciones de las heurísticas y el bloque de la derecha compara los dos algoritmos en función de la media de la mejor solución obtenida entre las distintas ejecuciones. Dentro de cada bloque de columnas, la

columna “1” indica el número de veces que el Modelo 1 supera al Modelo 2 y la columna “2” indica el número de veces que el Modelo 2 supera al Modelo 1. En todas las comparaciones se ha tenido en cuenta el redondeo a tres decimales que figura en la Tabla 19. Obviamente, para cada medida (AUC, FSC y KAP) la suma de los valores de la fila es igual a 13 (excepto en los casos en que ambos algoritmos rinden exactamente igual): hay 13 conjuntos de datos y el número de veces que un algoritmo es mejor que el otro más el número de veces que es peor debe ser el número de conjuntos de datos.

Por ejemplo, la última fila del bloque principal de la Tabla 19 muestra la comparación de ambos algoritmos cuando el tiempo es de 10800 segundos y el tamaño de la población es 100 para las diferentes medidas calculadas. Si nos centramos en la medida AUC, el Modelo 2 supera al Modelo 1 en 11 de los 13 conjuntos de datos y en los otros dos, el Modelo 1 es el que reporta mejores resultados, siendo estos conjuntos de datos WBC ($0.976 > 0.966$) que se encuentra en el grupo que tiene un número pequeño de características, y MFeat ($0.997 > 0.996$) agrupado con los conjuntos de datos que tienen más de 500 características. Los valores del resto de las filas pueden interpretarse del mismo modo. Si analizamos el rendimiento de ambos algoritmos atendiendo a la métrica AUC (media o máxima), el Modelo 2 supera al Modelo 1 en 112 de las 156 comparaciones realizadas. Si nos centramos ahora en la métrica FSC (media o máxima), el Modelo 2 supera claramente al Modelo 1 en todos los escenarios considerados, independientemente del tiempo de cálculo o del tamaño de la población. En concreto, el Modelo 2 ofrece mejores resultados para la métrica FSC en 122 de los 156 casos. Resultados similares se muestran si consideramos la métrica KAP, donde el Modelo 2 obtiene mejores resultados en 101 de los casos.

Los resultados presentados en esta sección indican claramente el mejor rendimiento de los clasificadores derivados del modelo alternativo para la SVM de margen suave, donde los objetivos clásicos han sido sustituidos por los nuevos objetivos, minimizando el número de vectores mal clasificados en cada clase. Así, cuando la bondad de la clasificación se vaya a medir en términos de AUC, FSC o KAP, recomendaríamos resolver el nuevo modelo propuesto frente al clásico.

			M			M^*		
	TIEMPO	TAMANO	AUC	FSC	KAP	AUC	FSC	KAP
Arrhythmia	1200	50	0.708	0.792	0.430	0.721	0.799	0.452
	1200	100	0.748	0.814	0.511	0.765	0.825	0.546
	3600	50	0.731	0.801	0.475	0.744	0.809	0.500
	3600	100	0.739	0.812	0.493	0.748	0.817	0.512
	10800	50	0.720	0.800	0.450	0.733	0.813	0.484
	10800	100	0.741	0.809	0.493	0.751	0.815	0.513
GC	1200	50	0.696	0.575	0.389	0.705	0.587	0.407
	1200	100	0.701	0.581	0.410	0.710	0.594	0.424
	3600	50	0.681	0.552	0.374	0.693	0.573	0.392
	3600	100	0.706	0.587	0.413	0.712	0.597	0.427
	10800	50	0.687	0.559	0.382	0.703	0.581	0.405
	10800	100	0.704	0.584	0.411	0.707	0.589	0.417
Housing	1200	50	0.844	0.846	0.687	0.852	0.854	0.703
	1200	100	0.851	0.850	0.702	0.860	0.859	0.720
	3600	50	0.836	0.839	0.672	0.844	0.845	0.687
	3600	100	0.856	0.855	0.712	0.861	0.859	0.720
	10800	50	0.841	0.845	0.682	0.853	0.857	0.706
	10800	100	0.854	0.859	0.708	0.866	0.869	0.732
IONO	1200	50	0.856	0.913	0.736	0.863	0.918	0.749
	1200	100	0.866	0.923	0.763	0.876	0.927	0.779
	3600	50	0.836	0.905	0.705	0.851	0.912	0.732
	3600	100	0.850	0.915	0.734	0.862	0.920	0.752
	10800	50	0.858	0.916	0.740	0.876	0.923	0.766
	10800	100	0.855	0.917	0.740	0.862	0.921	0.757
Madelon	1200	50	0.531	0.646	0.061	0.535	0.660	0.069
	1200	100	0.530	0.664	0.060	0.537	0.667	0.073
	3600	50	0.530	0.663	0.060	0.537	0.668	0.074
	3600	100	0.531	0.648	0.063	0.535	0.662	0.071
	10800	50	0.529	0.656	0.059	0.538	0.667	0.076
	10800	100	0.548	0.639	0.096	0.568	0.643	0.136
WBC	1200	50	0.974	0.966	0.946	0.980	0.974	0.959
	1200	100	0.975	0.967	0.947	0.976	0.968	0.948
	3600	50	0.973	0.964	0.942	0.974	0.965	0.944
	3600	100	0.970	0.961	0.937	0.974	0.964	0.941
	10800	50	0.968	0.960	0.936	0.970	0.962	0.939
	10800	100	0.976	0.967	0.947	0.976	0.968	0.948
Media			0.772	0.793	0.552	0.781	0.801	0.568

Tabla 15: Resultados para el Modelo 1 en datasets con $n \leq 500$

			\bar{M}			\bar{M}^*		
	TIEMPO	TAMANO	AUC	FSC	KAP	AUC	FSC	KAP
Arrhythmia	1200	50	0.750	0.811	0.508	0.753	0.814	0.514
	1200	100	0.755	0.824	0.528	0.760	0.827	0.537
	3600	50	0.751	0.813	0.513	0.756	0.817	0.524
	3600	100	0.754	0.824	0.527	0.764	0.831	0.544
	10800	50	0.751	0.815	0.514	0.753	0.816	0.517
	10800	100	0.752	0.825	0.524	0.752	0.825	0.524
GC	1200	50	0.706	0.593	0.351	0.711	0.597	0.373
	1200	100	0.714	0.602	0.401	0.717	0.605	0.403
	3600	50	0.705	0.591	0.360	0.712	0.599	0.375
	3600	100	0.715	0.602	0.414	0.717	0.603	0.417
	10800	50	0.704	0.590	0.346	0.706	0.592	0.367
	10800	100	0.720	0.610	0.427	0.721	0.611	0.431
Housing	1200	50	0.864	0.861	0.725	0.864	0.861	0.725
	1200	100	0.863	0.861	0.725	0.866	0.863	0.730
	3600	50	0.872	0.873	0.743	0.875	0.876	0.749
	3600	100	0.869	0.868	0.735	0.869	0.869	0.737
	10800	50	0.871	0.871	0.742	0.873	0.872	0.744
	10800	100	0.869	0.868	0.737	0.874	0.872	0.746
IONO	1200	50	0.858	0.915	0.742	0.858	0.916	0.744
	1200	100	0.867	0.918	0.750	0.873	0.921	0.760
	3600	50	0.863	0.919	0.753	0.866	0.923	0.762
	3600	100	0.862	0.917	0.749	0.866	0.921	0.759
	10800	50	0.865	0.917	0.753	0.866	0.917	0.754
	10800	100	0.866	0.918	0.752	0.867	0.919	0.754
Madelon	1200	50	0.572	0.680	0.144	0.587	0.683	0.174
	1200	100	0.617	0.696	0.235	0.620	0.696	0.241
	3600	50	0.586	0.680	0.171	0.594	0.683	0.188
	3600	100	0.611	0.690	0.222	0.625	0.693	0.249
	10800	50	0.603	0.691	0.205	0.609	0.693	0.218
	10800	100	0.583	0.687	0.165	0.598	0.690	0.195
WBC	1200	50	0.971	0.964	0.943	0.973	0.967	0.947
	1200	100	0.965	0.954	0.925	0.965	0.954	0.925
	3600	50	0.975	0.969	0.950	0.975	0.969	0.950
	3600	100	0.970	0.963	0.941	0.972	0.963	0.941
	10800	50	0.973	0.966	0.945	0.974	0.966	0.945
	10800	100	0.966	0.957	0.931	0.966	0.958	0.932
Media			0.793	0.808	0.586	0.797	0.810	0.594

Tabla 16: Resultados para el Modelo 2 en datasets con $n \leq 500$

	TIEMPO	TAMANO	M			M'		
			AUC	FSC	KAP	AUC	FSC	KAP
Arcene	1200	50	0.612	0.646	0.215	0.633	0.653	0.257
	1200	100	0.585	0.640	0.167	0.634	0.654	0.272
	3600	50	0.532	0.532	0.062	0.541	0.542	0.079
	3600	100	0.600	0.650	0.195	0.659	0.673	0.309
	10800	50	0.595	0.646	0.182	0.623	0.653	0.236
	10800	100	0.572	0.633	0.140	0.580	0.636	0.157
Bioresponse	1200	50	0.500	0.609	0.001	0.501	0.703	0.002
	1200	100	0.516	0.565	0.031	0.540	0.568	0.080
	3600	50	0.500	0.703	0.000	0.500	0.703	0.000
	3600	100	0.502	0.562	0.004	0.503	0.563	0.006
	10800	50	0.519	0.518	0.038	0.550	0.571	0.099
	10800	100	0.501	0.703	0.001	0.501	0.703	0.002
Duke	1200	50	0.724	0.739	0.445	0.725	0.739	0.446
	1200	100	0.736	0.751	0.470	0.779	0.785	0.553
	3600	50	0.730	0.734	0.452	0.745	0.744	0.479
	3600	100	0.740	0.768	0.478	0.761	0.795	0.519
	10800	50	0.691	0.704	0.376	0.714	0.736	0.421
	10800	100	0.768	0.794	0.532	0.805	0.823	0.605
Gina_agnostic	1200	50	0.636	0.630	0.274	0.657	0.641	0.315
	1200	100	0.645	0.567	0.293	0.703	0.649	0.408
	3600	50	0.617	0.656	0.235	0.668	0.676	0.337
	3600	100	0.648	0.660	0.298	0.695	0.675	0.393
	10800	50	0.626	0.562	0.254	0.660	0.642	0.323
	10800	100	0.624	0.638	0.250	0.695	0.658	0.392
Gisette	1200	50	0.720	0.747	0.441	0.737	0.778	0.475
	1200	100	0.750	0.785	0.500	0.784	0.806	0.567
	3600	50	0.767	0.799	0.534	0.857	0.860	0.714
	3600	100	0.672	0.712	0.345	0.695	0.726	0.389
	10800	50	0.762	0.797	0.524	0.842	0.849	0.685
	10800	100	0.666	0.688	0.333	0.738	0.734	0.475
Isolet	1200	50	0.641	0.746	0.282	0.646	0.751	0.293
	1200	100	0.674	0.637	0.349	0.715	0.660	0.430
	3600	50	0.593	0.722	0.186	0.593	0.722	0.186
	3600	100	0.718	0.797	0.435	0.843	0.868	0.687
	10800	50	0.630	0.731	0.260	0.648	0.743	0.297
	10800	100	0.684	0.759	0.368	0.717	0.780	0.434
MFeat	1200	50	0.997	0.996	0.996	0.997	0.997	0.997
	1200	100	0.997	0.997	0.997	0.997	0.997	0.997
	3600	50	0.997	0.997	0.997	0.997	0.997	0.997
	3600	100	0.998	0.997	0.997	0.998	0.997	0.997
	10800	50	0.997	0.995	0.995	0.998	0.995	0.995
	10800	100	0.997	0.997	0.997	0.997	0.997	0.997
Media			0.690	0.726	0.379	0.718	0.749	0.436

Tabla 17: Resultados para el Modelo 1 en datasets con $n > 500$

	TIEMPO	TAMANO	\bar{M}			\bar{M}^*		
			AUC	FSC	KAP	AUC	FSC	KAP
Arcene	1200	50	0.717	0.676	0.435	0.742	0.720	0.478
	1200	100	0.726	0.706	0.448	0.754	0.734	0.502
	3600	50	0.724	0.683	0.455	0.751	0.723	0.505
	3600	100	0.736	0.715	0.471	0.762	0.738	0.522
	10800	50	0.731	0.698	0.458	0.778	0.761	0.546
	10800	100	0.746	0.720	0.492	0.777	0.749	0.555
Bioresponse	1200	50	0.586	0.720	0.175	0.609	0.725	0.221
	1200	100	0.693	0.749	0.386	0.721	0.758	0.442
	3600	50	0.571	0.719	0.144	0.589	0.724	0.179
	3600	100	0.660	0.738	0.323	0.670	0.744	0.342
	10800	50	0.598	0.725	0.195	0.631	0.736	0.259
	10800	100	0.677	0.748	0.357	0.693	0.757	0.388
Duke	1200	50	0.617	0.623	0.236	0.627	0.627	0.255
	1200	100	0.793	0.794	0.584	0.843	0.838	0.682
	3600	50	0.555	0.558	0.112	0.629	0.623	0.257
	3600	100	0.774	0.792	0.548	0.839	0.854	0.678
	10800	50	0.681	0.712	0.362	0.705	0.731	0.410
	10800	100	0.790	0.817	0.581	0.848	0.874	0.702
Gina_agnostic	1200	50	0.611	0.678	0.221	0.619	0.679	0.238
	1200	100	0.642	0.683	0.285	0.651	0.683	0.303
	3600	50	0.650	0.683	0.300	0.688	0.694	0.378
	3600	100	0.641	0.684	0.283	0.664	0.688	0.330
	10800	50	0.597	0.687	0.195	0.613	0.687	0.226
	10800	100	0.664	0.710	0.329	0.682	0.719	0.363
Gisette	1200	50	0.684	0.758	0.368	0.699	0.769	0.399
	1200	100	0.762	0.789	0.524	0.778	0.801	0.556
	3600	50	0.711	0.766	0.423	0.773	0.797	0.545
	3600	100	0.746	0.783	0.492	0.769	0.801	0.537
	10800	50	0.741	0.788	0.483	0.778	0.815	0.556
	10800	100	0.736	0.775	0.473	0.757	0.788	0.515
Isolet	1200	50	0.980	0.980	0.960	0.982	0.982	0.964
	1200	100	0.988	0.988	0.975	0.989	0.989	0.978
	3600	50	0.983	0.983	0.966	0.983	0.983	0.966
	3600	100	0.977	0.977	0.955	0.981	0.981	0.962
	10800	50	0.984	0.983	0.967	0.986	0.986	0.972
	10800	100	0.985	0.985	0.970	0.986	0.986	0.972
MFeat	1200	50	0.996	0.988	0.986	0.996	0.990	0.988
	1200	100	0.993	0.978	0.975	0.993	0.978	0.975
	3600	50	0.995	0.983	0.981	0.997	0.985	0.983
	3600	100	0.995	0.988	0.987	0.996	0.991	0.990
	10800	50	0.996	0.988	0.986	0.998	0.995	0.995
	10800	100	0.996	0.989	0.988	0.996	0.991	0.990
Media			0.772	0.797	0.544	0.793	0.814	0.586

Tabla 18: Resultados para el Modelo 2 en datasets con $n > 500$

TIEMPO	TAMANO	\bar{M}						\bar{M}^*					
		AUC		FSC		KAP		AUC		FSC		KAP	
		1	2	1	2	1	2	1	2	1	2	1	2
1200	50	5	8	3	10	6	7	6	7	5	8	7	6
1200	100	3	10	3	10	5	8	6	7	4	9	7	6
3600	50	3	10	3	10	4	9	3	10	3	10	4	9
3600	100	2	11	1	12	2	11	3	10	2	11	4	9
10800	50	4	9	2	11	5	8	4	8	3	10	5	8
10800	100	2	11	2	11	2	11	3	10	3	10	4	9
Total		19	59	14	64	24	54	25	53	20	58	31	47

Tabla 19: Tabla de frecuencias de éxito de los algoritmos con respecto a su contrincante

4.3.2. SVM Y SELECCIÓN DE CARACTERÍSTICAS CON COSTES DE SELECCIÓN

Los costes ya sean temporales, técnicos o económicos son unos de los principales problemas a los que se enfrenta el investigador a la hora de realizar un estudio en el que tenga que tomar una decisión. Por ello, que hay veces que seleccionar las pruebas o características que se pueden ejecutar sea una tarea primordial a la hora de elaborar un proyecto. El avance de las técnicas de aprendizaje automático ha permitido establecer métodos para la selección de características en base a los costes (véase, por ejemplo: [Min et al. \(2014\)](#), [Zhang et al. \(2015\)](#) o [Zhou et al. \(2015\)](#)). Con respecto a SVM y selección de características, existe una amplia literatura al respecto que trata de unificar estas dos técnicas del aprendizaje automático. Por ejemplo, autores como: [Maldonado et al. \(2014\)](#), [Aytug \(2015\)](#), [Gaudioso et al. \(2017\)](#), [Labbe et al. \(2019\)](#) o [Benitez-Peña et al. \(2019\)](#) proponen modelos para la resolución del problema de SVM con selección de características. Sin embargo, en la actualidad, hasta donde tenemos conocimiento, no se ha establecido un modelo basado en SVM que aborde la optimización de los objetivos clásicos del SVM, lo que da lugar a buenos clasificadores, junto con la minimización del coste de las características seleccionadas. Es por ello que proponemos el modelo siguiente basado en el modelo tradicional de SVM de margen suave:

$$\begin{aligned}
\min \quad & O_{31} = \frac{1}{2} \|w\|^2 & (0) \\
\min \quad & O_{32} = \sum_{i=1}^n \xi_i & (1) \\
\min \quad & O_{33} = \sum_{j=1}^m c_j t_j & (2) \\
s.t \quad & y_i (w^T x_i + b) \geq 1 - \xi_i \quad i = 1, \dots, n, & (3) \\
& \xi_i \geq 0 \quad i = 1, \dots, n, & (4) \\
& w_j \in \mathbb{R} \quad j = 1, \dots, m, & (5) \\
& b \in \mathbb{R} & (6) \\
& \sum_{j=1}^m t_j = p \quad j = 1, \dots, m, & (7) \\
& |w_j| \leq M t_j \quad j = 1, \dots, m, & (8) \\
& t_j \in \{0, 1\} \quad j = 1, \dots, m, & (9)
\end{aligned} \tag{38}$$

El modelo (38) toma como base el modelo (12) e incluye un tercer objetivo (O_{33}) que consiste en minimizar el coste de las p características seleccionadas. Para la resolución de este problema tri-objetivo, hemos aplicado AUGMECON y, dado que está demostrado que esta técnica ya presenta dificultades al resolver el problema de forma exacta con sólo dos objetivos, lo hemos resuelto también con el algoritmo NSGA-II, presentado en la [Sección 3.6.1](#). En este caso el algoritmo se adapta para tres objetivos. Por lo tanto, a la hora de evaluar las soluciones se tiene que tener en cuenta el tercer objetivo de minimización de costes. En la metaheurística del NSGA-II bi-objetivo se tiene que maximizar la distancia de los hiperplanos separadores y minimizar la suma de las distancias de las observaciones erróneas. En este caso, se añade a estos dos, la minimización de los costes de las características seleccionadas (véase el objetivo 38.2 donde c_j es el coste asociado a la característica j cuando ésta es seleccionada ($t_j = 1$)). Del mismo modo, en el proceso evolutivo del NSGA-II, en los pasos de selección de padre y madre se aplican los nuevos criterios de dominación, así como, cuando todos los individuos de un frente no pueden ser seleccionados para formar parte de la nueva población, se ordenan los individuos por un criterio de distancia, que ahora debe tener en cuenta también este tercer objetivo..

La Figura 15 muestra un ejemplo de aplicación del algoritmo de tres objetivos frente al algoritmo de dos objetivos. Cada cruz verde es un clasificador proporcionado por el algoritmo de dos objetivos mientras que cada punto rojo es un clasificador proporcionado por el algoritmo de tres objetivos. En el caso del algoritmo bi-objetivo, los costes no forman parte del problema de optimización, por lo que una vez seleccionadas las características se procede a la suma de sus pesos. En la imagen se observa como el algoritmo de tres objetivos es capaz de encontrar soluciones más baratas frente al algoritmo de dos objetivos. Para ello, puede sacrificar la obtención de mejores rendimientos en los otros dos objetivos para obtener una solución más equilibrada en los tres objetivos, mientras que el algoritmo de dos objetivos, al sólo centrarse en la distancia y el ϵ (la suma de las distancias de las observaciones mal clasificadas) tiende a buscar mejores soluciones en esos dos objetivos. Este sencillo ejemplo demuestra la utilidad del enfoque propuesto. Esto se pone de manifiesto con mucha más rotundidad en el experimento computacional llevado a cabo que se presenta a continuación.

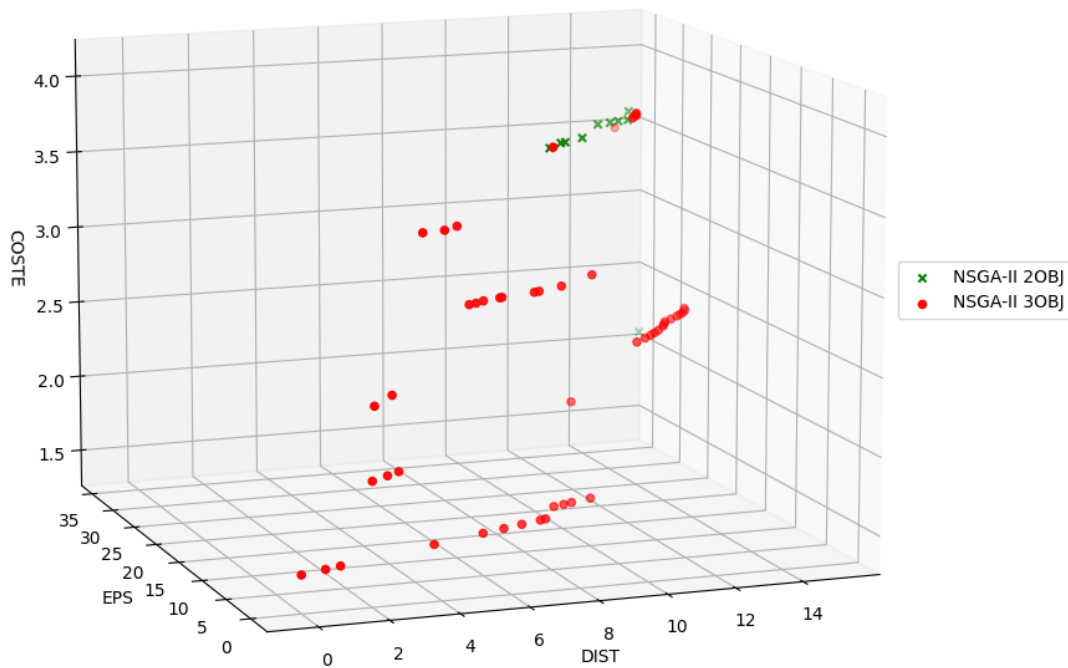


Figura 15: Ejemplo de aplicación de la metaheurística de dos objetivos frente a la de tres objetivos

4.3.2.1. EXPERIENCIA COMPUTACIONAL

En esta sección vamos a analizar los resultados que pueden ser obtenidos cuando resolvemos el nuevo modelo propuesto en el que se optimizan simultáneamente tres objetivos (38) con diferentes técnicas, exactas y metaheurísticas. En primer lugar, realizamos un experimento para analizar los tiempos de ejecución adecuados para la metaheurística. Después, estudiamos los resultados que ofrece la técnica exacta frente a la metaheurística. En este caso, comparar los resultados supone comparar fronteras de tres dimensiones, aplicando diferentes métricas presentes en la literatura para analizar el rendimiento de dos fronteras Pareto o aproximaciones a la misma. Para ello, se ha hecho uno de las métricas de comparación de fronteras expuestas en la [Sección 3.6.3](#). Seguidamente, realizamos un estudio de las ventajas que supone considerar también el objetivo del coste frente al modelo SVM clásico, en el que el coste no se considera. En este caso, se miden los diferentes valores de los objetivos obtenidos por los dos algoritmos. Finalmente, se analiza la bondad de los clasificadores obtenidos por el nuevo metaheurístico en base al número de características seleccionadas.

Las características de las instancias utilizadas en el estudio se muestran en la Tabla 20. Todas ellas, se encuentran disponibles en el repositorio UCI ([Dua et al. \(2017\)](#)). Puesto que en dichas instancias no se especifican costes para las características, éstos han sido simulados aleatoriamente siguiendo una distribución $c \sim Uni[1,10]$. Dichas instancias con costes están disponibles para su uso bajo petición a los autores. Las simulaciones se realizaron en un clúster central con sistema operativo CentOS Linux versión 7.5.1804 que constaba con cinco nodos con las siguientes características. Nodo 0: modelo Bull X410 con dos procesadores Xeon Platinum v5 8160-2.1 Ghz (en total 96 cores), 384 Gigabyte de RAM y cuatro GPU Nvidia V100. Nodo 1: similar al Nodo 0 con 768 Gigas de RAM. Nodo 2 y 3: modelo Dell PowerEdge R440 con dos procesadores Xeon Gold 5120 @ 2,2 GHz (en total 56 núcleos) y 128 Gigabyte de RAM. Nodo 4: modelo Supermicro SYS-1029GQ-TRT con dos procesadores Xeon Gold 6230N a 2,3 GHz (en total 80 núcleos), 640 Gigabytes de RAM y tres GPU (una NVidia V100 y dos NVidia A100). Las simulaciones fueron lanzadas en lotes utilizando para su gestión SLURM Workload Manager, asignando un núcleo a cada conjunto de datos.

Instancia	Vectores (n)	Características (m)
Arrhythmia	420	278
Carcinoma	36	7457
Colon	62	2000
DLBCL	77	7127
Leukemia	72	5327
Mfeat	2000	649

Tabla 20: Descripción de las instancias utilizadas

En la primera parte del análisis, hemos analizado el tiempo a establecer como criterio de parada por parte de la técnica metaheurística. En otros trabajos, como en el de [Alcaraz et al. \(2022\)](#) o el mostrado en la [Sección 4.3.1](#) utilizan una hora para la obtención de la frontera al resolver este conjunto de instancias. En este caso, queremos

comprobar si es mejor ejecutar la metaheurística durante una hora o realizar tres ejecuciones de 20 minutos y construir la frontera correspondiente con las soluciones no dominadas obtenidas en las tres ejecuciones.

La Tabla 21 muestra los resultados de los dos escenarios analizados, cuando se repite el algoritmo 3 veces con una duración de 20 minutos y cuando se ejecuta una vez durante una hora. Los valores en negrita indican el mejor valor en ambos escenarios para cada una de las métricas en cada instancia. Observamos como el número de soluciones no dominadas (OVNG) es el mismo para ambos métodos. Sin embargo, en lo que respecta al porcentaje de soluciones dominadas por una frontera frente a la otra (C) el NSGA-II de una hora obtiene mejores resultados en 5 de los escenarios. Lo mismo ocurre con la métrica M_3 , que mide la extensión de la frontera, y Γ/M_3 . En el cálculo de S-Metric o Hipervolumen el algoritmo ejecutado una hora siempre vencía a la repetición de 20 minutos. Para el cómputo de esta métrica realizamos una adaptación del algoritmo propuesto por [Beume et al. \(2009\)](#). En este caso el algoritmo trata de calcular el hipervolumen para un algoritmo tri-objetivo max-min-min en lugar de un tri-objetivo max-max-max. Viendo los resultados decidimos ejecutar el algoritmo una hora ya que ofrecía mejores resultados que la repetición del algoritmo tres veces.

INSTANCIA	TIPO	OVNG	C	M_3	$\frac{\Gamma}{M_3}$	$S - Metric$
COLON	$20min \times 3$	500	15,600	119,069	0,015	9380,272
	1h	500	27,000	132,473	0,019	14328,160
LEUKEMIA	$20min \times 3$	500	18,200	139,737	0,017	8603,659
	1h	500	22,400	147,025	0,017	12535,106
DLBCL	$20min \times 3$	500	38,400	144,230	0,035	15171,098
	1h	500	11,200	189,934	0,027	20060,235
CARCINOMA	$20min \times 3$	500	18,800	72,680	0,038	6451,920
	1h	500	30,600	76,567	0,025	5986,143
ARRYTHMIA	$20min \times 3$	500	8,200	831,629	0,014	77168,700
	1h	500	41,000	824,107	0,008	80987,447
MFEAT	$20min \times 3$	500	14,000	4265,847	0,027	391177,172
	1h	500	42,800	4748,117	0,015	505497,262

Tabla 21: Comparativa de resultados entre el algoritmo repetido tres veces, 20 minutos y el algoritmo ejecutado una hora: Medidas de cardinalidad, convergencia y distribución

Seguidamente pasamos a comparar el algoritmo exacto (AUGMECON2) con los resultados obtenidos por el NSGA-II. El objetivo es comprobar la calidad de las fronteras obtenidas por ambos métodos. En este caso, AUGMECON2 se lanzó durante 24 horas mientras que la metaheurística se ejecutó durante 1 hora.

La Tabla 22 muestra las métricas de las fronteras obtenidas por AUGMECON2 y NSGA-II cuando se seleccionan $p = 5$ características. El rendimiento de la metaheurística es muy superior al de AUGMECON2 en todas las métricas teniendo en cuenta el tiempo de ejecución de ambos: una hora frente a 24. Con respecto al número de soluciones obtenidas (OVNG), NSGA-II obtiene en todos los casos fronteras con 500 soluciones, mientras que AUGMECON2, no llega a sobrepasar 11 en 24 horas de ejecución. En lo referente al porcentaje de soluciones de NSGA-II que dominan a las de la frontera de AUGMECON2, se encuentra entre el 22 y el 83 por ciento, mientras que en el caso de las que AUGMECON2 domina a NSGA-II, no supera el 3 por ciento. En el caso de Mfeat, por ejemplo, ninguna de las soluciones de AUGMECON2 domina a las del NSGA-II, mientras que éste llega a tener un 75 por ciento de soluciones que dominan a las del otro algoritmo. Las medidas de convergencia también determinan que la frontera es más dispersa en el NSGA-II y que los puntos consecutivos están más separados. Por ejemplo, en la instancia DLBCL, el valor de M_3 es aproximadamente 6 veces más grande en NSGA-II que en AUGMECON y el valor de Γ/M_3 es 10 veces más pequeño. Finalmente, el valor del Hipervolumen es muy superior para NSGA-II en todos los escenarios, lo que indica que la frontera abarca un hipervolumen superior al de la frontera proporcionada por AUGMECON2. Estos resultados evidencian claramente la superioridad de las fronteras que obtiene NSGA-II, en una hora, frente a las de la técnica exacta en un tiempo 24 veces superior.

INSTANCIA	TIPO	OVNG	C	M_3	$\frac{\Gamma}{M_3}$	S - Metric
COLON	AUGMECON	10	0,00	34,60	0,20	1349,04
	NSGA-II	500	60,00	132,47	0,02	14328,16
LEUKEMIA	AUGMECON	11	0,40	61,06	0,16	3732,71
	NSGA-II	500	72,73	147,02	0,02	12535,11
DLBCL	AUGMECON	6	0,80	35,18	0,31	1169,44
	NSGA-II	500	83,33	189,93	0,03	20060,24
CARCINOMA	AUGMECON	9	3,20	43,12	0,31	2028,35
	NSGA-II	500	22,22	76,57	0,02	5986,14
ARRYTHMIA	AUGMECON	10	1,00	365,49	0,41	25651,64
	NSGA-II	500	40,00	824,11	0,01	80987,45
MFEAT	AUGMECON	4	0,00	1439,11	0,91	74644,84
	NSGA-II	500	75,00	4748,12	0,01	505497,26

Tabla 22: Comparativa de resultados entre AUGMECON y la NSGAI: Medidas de cardinalidad, convergencia y distribución

A continuación, comparamos los resultados al resolver el problema con dos objetivos frente a los que se obtienen optimizando adicionalmente el coste. En este caso, ambos problemas se han resuelto únicamente con la técnica metaheurística. Para hacer este experimento, se lanza el algoritmo de dos objetivos sin tener en cuenta el coste y una vez seleccionadas las características se calcula la suma de los costes. En el caso del algoritmo tri-objetivo, el coste sí que es uno de los objetivos a optimizar.

La Tabla 23 muestra las diferencias entre la media (AVG), el mínimo (MIN) y el máximo valor (MAX) para cada uno de los objetivos, de los puntos pertenecientes a la frontera que genera cada uno de los algoritmos metaheurístico, con dos o tres objetivos. O_1 es el valor del objetivo que maximiza la distancia entre hiperplanos, O_2 es el objetivo que minimiza la suma de los errores de las observaciones mal clasificadas y O_3 es el objetivo que minimiza la suma de costes asociados a las características seleccionadas.

INS	TIPO	O_1			O_2			O_3		
		MIN	MAX	AVG	MIN	MAX	AVG	MIN	MAX	AVG
Colon	NSGA-II (2OBJ)	1,15	4,15	2,63	0,00	92,27	53,37	8,15	32,88	21,10
	NSGA-II (3OBJ)	0,00	4,47	2,29	0,01	125,78	51,45	5,05	46,39	13,58
Leukemia	NSGA-II (2OBJ)	0,14	3,93	2,46	0,06	101,92	40,60	18,41	32,02	22,68
	NSGA-II (3OBJ)	0,00	4,47	2,47	0,00	143,86	59,40	5,03	35,02	12,29
DLBCL	NSGA-II (2OBJ)	0,29	4,24	2,57	0,01	79,15	35,78	23,48	42,87	32,68
	NSGA-II (3OBJ)	0,00	4,47	2,57	0,00	187,44	54,66	5,01	35,38	15,19
Carcinoma	NSGA-II (2OBJ)	0,44	4,04	2,56	0,01	57,75	17,56	23,46	43,12	35,34
	NSGA-II (3OBJ)	0,00	4,47	2,64	0,00	71,60	26,64	5,02	31,79	11,49
Arrythmia	NSGA-II (2OBJ)	2,96	4,02	3,52	0,00	707,35	600,28	17,26	36,61	23,13
	NSGA-II (3OBJ)	0,00	4,47	1,98	0,00	823,04	332,71	5,56	47,29	19,82
Mfeat	NSGA-II (2OBJ)	3,72	4,32	3,92	0,00	2399,30	1131,06	14,89	28,23	20,85
	NSGA-II (3OBJ)	0,00	4,47	2,63	0,00	4748,04	1535,35	5,22	32,62	9,32

Tabla 23: Comparativa de las metaheurísticas al considerar dos o tres objetivos

En el O_1 se desean valores altos puesto que se trata de un objetivo de maximizar. En los objetivos O_2 y O_3 se esperan valores pequeños puesto que el objetivo es minimizar las distancias de los errores de clasificación y los costes. Analizando los resultados, se puede observar como la metaheurística de dos objetivos obtiene mejores resultados en los objetivos que optimiza. Por ejemplo, en el escenario Mfeat, la media de los objetivos O_1 y O_2 obtiene mejores rendimientos frente a los del tri-objetivo ($3,92 > 2,63$ y $1131,06 < 1535,35$). En los demás escenarios, si bien a veces el algoritmo de tres objetivos obtiene algunos buenos resultados, el algoritmo de dos objetivos se impone. Sin embargo, esto no sucede en el O_3 donde las medias y valores mínimos de coste se obtienen en el algoritmo tri-objetivo (véase, por ejemplo, el escenario Mfeat: $20,85 < 9,32$). Los resultados nos hacen pensar que la pérdida de rendimiento en los dos primeros objetivos se debe a que la metaheurística debe sacrificar una mejor clasificación en pos de una reducción de costes.

Finalmente, una vez demostrado que las soluciones de la frontera Pareto de la metaheurística del NSGA-II tri-objetivo proporcionan buenos resultados frente al método exacto y que al considerar el coste como objetivo se obtienen soluciones variadas, que representan buenos clasificadores desde el enfoque clásico y a la vez ofrecen alternativas de coste variadas frente a la metaheurística bi-objetivo, nos hemos propuesto examinar el efecto del número de características a seleccionar, p , que como hemos explicado antes, es un parámetro del problema.. En este contexto, hemos

ejecutado el metaheurístico tri-objetivo, seleccionando diferentes valores de características ($p = 5, 50, 500$). Para medir el rendimiento en cada caso hemos escogido las métricas basadas en la matriz de confusión y se ha realizado un proceso de validación y testeo similar al de la Figura 14.

La Tabla 24 está dividida en dos partes donde se muestra la media de las cinco repeticiones de experimentos (\bar{M}) así como la media de los cinco máximos (\bar{M}^*) para cada una de las métricas (Area Bajo la Curva, índice Kappa y F-Score). Cuando el número de características a seleccionar es pequeño (5 en este caso), se obtienen mejores resultados en tres instancias mientras que cuando seleccionamos 50 gana en dos y la selección de 500 en una ocasión. De los resultados podemos extraer la conclusión general de que, dada una instancia, no existe a priori, sin resolver antes el problema, un número de características que podamos fijar como adecuado. Depende de la instancia y no siempre un número mayor de características da mejores resultados, sino que a veces es al contrario. Por ejemplo, en la instancia Colon, donde el número de características es de 2000, el método que mejor rendimiento tiene es aquel en el que se seleccionan 500 características. Sin embargo, en otro ejemplo de gran dimensionalidad con 7127 características (DLBCL), la metaheurística que mejores métricas obtiene es la que selecciona 5 características.

En conclusión, se ha demostrado que el nuevo método tri-objetivo es una buena forma de obtener buenos clasificadores a bajo coste a la hora de resolver el problema de SVM con selección de características. La disminución de la distancia de los hiperplanos separadores y el aumento del error empírico se compensa con la reducción de los costes, cosa que el algoritmo bi-objetivo no puede conseguir. Además, ofrece la posibilidad de obtener un gran número de soluciones con tiempos de ejecución inferiores a AUGMECON2. Finalmente, el hecho de que no se pueda determinar de antemano un número fijo de características abre una vía futura de investigación que consistiría en dejar al algoritmo determinar qué número de características es óptima para la consecución de buenos clasificadores.

INSTANCIA	Nº CARAC.	P	\bar{M}			\bar{M}^*		
			AUC	FSC	KAP	AUC	FSC	KAP
COLON	2000	5	0,700	0,626	0,329	0,714	0,639	0,366
		50	0,681	0,602	0,324	0,700	0,620	0,358
		500	0,719	0,630	0,397	0,721	0,631	0,403
LEUKEMIA	5327	5	0,885	0,909	0,767	0,892	0,917	0,783
		50	0,736	0,740	0,474	0,766	0,762	0,534
		500	0,792	0,790	0,587	0,796	0,792	0,593
DLBCL	7127	5	0,658	0,822	0,265	0,684	0,859	0,298
		50	0,569	0,684	0,097	0,592	0,765	0,154
		500	0,585	0,759	0,120	0,595	0,763	0,131
CARCINOMA	7457	5	0,839	0,887	0,678	0,850	0,893	0,700
		50	0,933	0,942	0,867	0,933	0,942	0,867
		500	0,922	0,932	0,844	0,933	0,941	0,867
ARRYTHMIA	278	5	0,731	0,806	0,472	0,754	0,813	0,507
		50	0,746	0,815	0,507	0,759	0,824	0,532
		500	1,000	0,999	0,999	1,000	1,000	1,000
MFEAT	649	5	0,988	0,971	0,968	0,989	0,971	0,968
		50	0,988	0,971	0,968	0,989	0,971	0,968
		500	0,848	0,665	0,627	0,848	0,665	0,627

Tabla 24: Comparativa de métricas relacionadas con la matriz de confusión en base al número de características seleccionadas

CAPÍTULO 5: DISCUSIÓN

En esta sección se consideran los avances realizados a lo largo de la tesis relacionando los logros conseguidos con la literatura existente hasta el momento.

Actualmente, las Máquinas de Soporte Vectorial se han convertido en una de las técnicas de aprendizaje automático más utilizadas y estudiadas en la literatura especializada. Esto hace que los investigadores busquen cada vez nuevas aplicaciones y funcionalidades para ellas. Por ejemplo, una simple búsqueda de Support Vector Machines en Google Académico nos arroja más de 1,5 millones de resultados.

Siguiendo los objetivos planteados en el Capítulo 2, existe una activa corriente académica que tiene como objetivo unir el mundo de las técnicas de aprendizaje automático con el de la estimación de fronteras de producción. En la literatura existen numerosas muestras de ello. Por ejemplo, [Kuosmanen y Johnson \(2010, 2017\)](#) dieron lugar al método de Mínimos Cuadrados No Paramétricos Cóncavos Corregidos (C2NLS), probando que DEA podía ser interpretado como una regresión de mínimos cuadrados no paramétrica sujeta a restricciones de forma en la frontera y restricciones de signo de los residuos. [Parmeter y Racine \(2013\)](#) propusieron un enfoque no paramétrico y semiparamétrico con restricciones suaves. [Daouia et al. \(2016\)](#) desarrollaron un método basado en splines polinómicos para estimar funciones de producción. [Esteve et al. \(2020, 2022\)](#) adaptaron los árboles de regresión y Random Forest para la estimación de fronteras de producción. [Guerrero et al. \(2022\)](#) presentaron una nueva metodología que permite estimar tecnologías poliédricas siguiendo el principio de Minimización del Riesgo Estructural (SRM). [Tsonas \(2020,](#)

2022) desarrollaron un modelo híbrido entre Redes Neuronales y Árboles de Búsqueda Aleatoria con Análisis Envolvente de Datos (DEA). [Guillen et al. \(2023\)](#) modificaron la técnica Gradient Tree Boosting para la estimación de fronteras de producción. Todos ellos muestran el arduo esfuerzo que se está realizando para adaptar o utilizar las técnicas de aprendizaje automático con la estimación de fronteras de producción. Sin embargo, ninguno de ellos se había realizado en el ámbito de las Máquinas de Soporte Vectorial.

En esta tesis se ha mostrado paso a paso cómo se ha modificado el algoritmo de Máquinas de Soporte Vectorial de Regresión para adaptarlo a la estimación de fronteras y crear una nueva técnica, conocida como Support Vector Frontiers (Objetivo 1). El primer paso fue la creación de un modelo mono-salida que diera lugar a funciones de producción escalonadas que pudieran competir con la técnica FDH. Una vez obtenida la frontera, la convexificación de la misma daba lugar a una técnica que podía competir con DEA. Asimismo, se ha demostrado que el FDH y el DEA pueden verse como casos particulares del algoritmo SVF ya que son siempre solución factible del problema y óptima en algunos casos particulares. Del mismo modo, se ha creado un nuevo concepto de eficiencia técnica basándonos en el concepto del margen de las Máquinas de Soporte Vectorial de Regresión. Este nuevo concepto de eficiencia dota de más robustez al método. Posteriormente se desarrollaría el modelo multi-salida aplicando para ello, la adaptación de SVR a contextos de múltiples salidas de [Walter y Vázquez \(2003\)](#). Computacionalmente, se ha demostrado que el nuevo modelo SVF y su versión convexificada CSVF mejoran los rendimientos de error cuadrático medio y sesgo de los modelos FDH y DEA. Esto se debe a que la capacidad generalizadora de las SVM permite a la frontera estimada no sobreajustarse a los datos de la muestra de aprendizaje. El hecho de que el coste computacional de ejecutar la transformación de SVF hiciera inabordable la solución de algunos problemas, hizo proponer la aparición de un nuevo modelo de SVF heurístico que redujera la complejidad del modelo SVF planteado inicialmente. A pesar de ello, los resultados con respecto a las técnicas tradicionales de estimación de fronteras siguieron siendo mejores en todos los escenarios.

La aplicación de este modelo a las técnicas de jerarquización de características permitió conseguir el cumplimiento del segundo objetivo planteado en esta tesis. El ranking y la selección de características son dos tareas importantes dentro de la analítica de datos puesto que determinan la importancia relativa de las variables de entrada con respecto a las variables de salida. Esto se puede traducir en que el conocimiento objetivo de qué variables son importantes en un problema puede reducir el ruido innecesario y los costes tanto computacionales como temporales del proceso (Guyon et al. (2002) y Louppe et al. (2013)). Para ello, se hizo una revisión bibliográfica de cómo se hacía la ordenación de características en el contexto de estimación de fronteras. Algunos autores ya habían conseguido éxitos como Pastor et al. (2002), mediante pruebas de hipótesis, Ruggiero (2005) mediante regresión, (Peyrache et al. (2020) y Benitez-Peña et al. (2020)) añadiendo restricciones binarias al modelo DEA, así como Araujo et al. (2014) utilizando métodos estadísticos. Sin embargo, ninguno de ellos aplicaba las técnicas de SVM al ranking de características con estimación de fronteras. Autores como Esteve et al. (2023) sí que lo habían conseguido, pero utilizando Árboles de Regresión.

Por otro lado, se hizo también una revisión de la literatura de las diferentes técnicas de ranking de características con Support Vector Machine existentes y se seleccionaron cuatro de ellas, para adaptarlas a la aplicación del modelo SVF. La primera fue la de Breiman (2001) que establece un método de ordenación de variables basado en la disminución de la precisión de la predicción cuando se modifican los valores de una variable. La segunda técnica seleccionada fue la propuesta por Alonso-Atienza et al. (2012) que consiste en la creación de muestras bootstrap para la estimación de la importancia de las variables de entrada. La tercera técnica, propuesta por Sanz et al. (2018), evalúa la variación de la desviación media absoluta de la predicción cuando se ejecutan pseudomuestras sobre el modelo. Finalmente, se analizó el modelo de Guyon et al. (2002) que evalúa los cambios en la función objetivo del modelo SVM cuando se quita una variable del problema. Ninguno de estos métodos había sido utilizado hasta la fecha para la ordenación de características en problemas de estimación de fronteras. Se realizó una adaptación de ellos y se comparó la capacidad

de determinar correctamente el orden de las variables en experimentos simulados obteniendo buenos resultados.

En cuanto a las aplicaciones de Maquinas de Soporte Vectorial de clasificación con selección de características también existe una amplia literatura al respecto. En el problema tradicional se pretende obtener un hiperplano separador que maximice la distancia entre los hiperplanos paralelos y minimice el error de las observaciones mal clasificadas. Como se menciona en secciones anteriores, diversos autores proponen métodos para la resolución del problema: [Maldonado et al. \(2014\)](#) introducen dos modelos enteros mixtos de SVM con selección de características; [Aytug \(2015\)](#) resuelve la SVM con selección de características con la ayuda de la descomposición de Benders; [Gaudioso et al. \(2017\)](#) proponen un enfoque de relajación lagrangiana para el problema SVM con selección de características; [Labbe et al. \(2019\)](#) mejoran el modelo de Maldonado o [Benitez-Peña et al. \(2019\)](#) proponen un problema lineal entero más un problema convexo cuadrático. Sin embargo, la incorporación del problema de selección de características complicaba la obtención de una frontera Pareto de soluciones. [Alcaraz et al. \(2022\)](#) fueron los primeros en proponer un algoritmo basado en el Non-Dominated Sorting Genetic Algorithm (NSGA-II). En estos problemas solo se tenía en cuenta los objetivos de maximizar la distancia del margen y minimizar el error empírico. Cada una de las soluciones de la frontera daba lugar a un clasificador que puede ser evaluado a través de métricas, basadas en la matriz de confusión, que miden la precisión del modelo. Esto nos hizo plantearnos la posibilidad de crear un modelo en que se sustituyeran los objetivos tradicionales por unos nuevos que tuvieran en cuenta los errores de clasificación y así aumentara la precisión de los clasificadores obtenidos en base a las métricas de la matriz de confusión. Esto se materializó en un nuevo modelo que tiene como objetivo minimizar el número de Falsos Positivos y Falsos Negativos de un clasificador. El nuevo modelo se comparó con el tradicional, resolviéndose diferentes instancias mediante la aplicación de metaheurísticas obteniéndose mejores resultados para la mayoría de las instancias en evaluación.

Finalmente, se observó que en la literatura no existía ninguna técnica de SVM con selección de características que optimizara además el coste de las características a seleccionar si éstos eran conocidos de antemano. En este caso, se introdujo un tercer objetivo a la metodología tradicional de SVM y selección de características que tratara de minimizar los costes de las características seleccionadas. El resultado ha sido un nuevo modelo tri-objetivo que se puede resolver con una adaptación que hemos desarrollado de la técnica y que mejora en rendimientos computacionales a la técnica AUGMECON2, pero que reduce sustancialmente los costes de las características seleccionadas sin perder por ello calidad en los clasificadores.

CAPÍTULO 6: CONCLUSIONES

En esta tesis, se ha creado nueva metodología y aplicaciones para una de las técnicas de mayor impacto en el aprendizaje automático como son las Máquinas de Soporte Vectorial. En un primer momento, esta tesis iba a estar enfocada en el establecimiento de un puente entre las técnicas de estimación de fronteras de producción y el mundo del aprendizaje automático, aunque al final se ha ampliado a nuevas aplicaciones de las SVM en el contexto de selección de características.

Si bien ya había un intento por diversos autores de convertir las técnicas Free Disposal Hull y el Análisis Envoltente de Datos para que funcionen como métodos inferenciales y no como meras herramientas descriptivas (véase, por ejemplo, [Banker y Maindiratta \(1992\)](#), [Banker \(1993\)](#), [Simar y Wilson \(1998\)](#), [Kuosmanen y Johnson \(2010\)](#)), sólo algunos autores como [Esteve et al. \(2020\)](#), con su técnica EAT basada en Árboles de Regresión, se habían atrevido a unir el mundo del aprendizaje automático y la estimación de fronteras.

Los primeros trabajos de esta tesis han dado lugar a una nueva técnica de análisis de fronteras de producción, llamada Support Vector Frontiers (SVF), que permite la creación de fronteras que pueden competir con los métodos tradicionales FDH y DEA. Esta nueva técnica no solo mejora sus resultados en simulaciones, sino que se ha demostrado que FDH y DEA pueden verse como casos particulares de la nueva metodología. Además, se ha introducido una nueva definición más robusta de ineficiencia, basada en el concepto de margen de las Máquinas de Soporte Vectorial. Sin embargo, esta nueva técnica presenta un inconveniente y es el alto coste

computacional debido al proceso de transformación del espacio de atributos al espacio de características donde la dimensionalidad es excesivamente alta. Esto ha dado lugar a un nuevo modelo heurístico que, si bien no soluciona del todo el problema computacional, permite reducir la carga y el tiempo de ejecución del algoritmo.

Como futuras líneas de investigación en esta temática, aún quedan por explorar nuevos caminos como la aplicación de nuevas formas de transformación del espacio de atributos que permita superficies estimadoras con formas más complejas que las funciones lineales a trozos utilizadas. Otra nueva vía de investigación futura podría ser la consideración de enfoques alternativos para pasar del SVF single-output al marco multi-output, abandonando el modelo de [Vázquez y Walter \(2003\)](#).

Otro contenido de esta tesis estuvo relacionado con la aplicación de un método para jerarquizar la contribución de las variables predictoras a las variables objetivo de un problema de estimación de fronteras. En este sentido se adaptaron técnicas que utilizan SVM para el establecimiento de ranking de variables. Para ello, se utilizó el algoritmo SVF desarrollado en los dos primeros artículos. El resultado fue la generación de diferentes métodos que fueron comparados para medir su rendimiento en diferentes escenarios. Esto permitió establecer unos métodos que, si bien presentan el inconveniente del tiempo de ejecución y recursos computacionales, ofrecen buenos rendimientos a la hora de establecer un orden de importancia en las variables.

En un segundo momento, ya avanzada la investigación, se pensó en no reducir solo el ámbito de estudio de esta tesis al caso de las Máquinas de Soporte Vectorial de Regresión y se amplió el análisis a nuevas aplicaciones de Máquinas de Soporte Vectorial de Clasificación con selección de características. En la literatura, ya existían técnicas que permitían abordar el tema desde una perspectiva mono-objetivo y bi-objetivo. En ellas, los dos principales objetivos eran los planteados por [Vapnik \(1995\)](#) que consistían en alcanzar un acuerdo entre error de generalización y el error empírico. Como los resultados de esas soluciones eran unos clasificadores que podían evaluar su rendimiento en base a unas métricas basadas en la matriz de confusión, propusimos un nuevo modelo con nuevos objetivos. En ese modelo los objetivos a minimizar eran el

número de falsos positivos y falsos negativos. De este modo, pensamos que se podría obtener una frontera Pareto formada por mejores clasificadores. Como el problema es bastante complejo y las soluciones exactas difíciles de obtener, se utilizó una metaheurística basada en el algoritmo NSGA-II. El resultado fue un modelo bi-objetivo que mejoraba los resultados del método desarrollado actualmente en la literatura (Alcaraz et al. (2022)) en la mayoría de las instancias en evaluación.

Otra forma de ampliar el problema del SVM con selección de características es aumentar el número de objetivos teniendo en cuenta el coste de la selección de características. De este modo, se presentó un modelo tri-objetivo en el que se pretendía maximizar la distancia de los hiperplanos clasificadores, minimizar la distancia del número de soluciones mal clasificadas a su hiperplano y a la vez minimizar el coste de las características seleccionadas. Para resolver este modelo se adaptó la metaheurística propuesta por Alcaraz et al. (2022) para considerar de forma simultánea los tres objetivos. Los resultados demostraron que el método tri-objetivo obtenía buenos clasificadores a menor coste a la hora de solucionar el problema de SVM con selección de características. Además, el número de soluciones obtenidas, así como la calidad de las mismas, con respecto a AUGMECON2 era muy superior en menor tiempo.

Finalmente, señalar como futura línea de investigación de esta temática la posibilidad de diseñar y desarrollar nuevas metaheurísticas basadas en diferentes paradigmas como, por ejemplo, SPEA2, PAES, etc. Asimismo, se baraja la posibilidad de añadir el objetivo del coste al modelo en el que se minimizan falsos positivos y falsos negativos, así como diseñar una nueva metaheurística para la versión del problema en la que p no se fija de antemano, sino que es una variable más del problema.

REFERENCIAS

Abdunabi, T., & Basir, O. (2014, October). Predicting a biological response of molecules from their chemical properties using diverse and optimized ensembles of stochastic gradient boosting machine. In 2014 IEEE International Conference on Big Data (Big Data) (pp. 10-17). IEEE.

Adler, N., & Golany, B. (2002). Including principal component weights to improve discrimination in data envelopment analysis. *Journal of the Operational Research Society*, 53, 985-991.

Adler, N., & Yazhemsky, E. (2010). Improving discrimination in data envelopment analysis: PCA–DEA or variable reduction. *European Journal of Operational Research*, 202(1), 273-284.

Aigner, D. J., & Chu, S. F. (1968). On estimating the industry production function. *The American Economic Review*, 58(4), 826-839.

Aladeemy, M., Tutun, S., & Khasawneh, M. T. (2017). A new hybrid approach for feature selection and support vector machine model selection based on self-adaptive cohort intelligence. *Expert Systems with Applications*, 88, 118-131.

Alcaraz, J., Labbé, M., & Landete, M. (2022). Support Vector Machine with feature selection: A multiobjective approach. *Expert Systems with Applications*, 204, 117485.

Alonso-Atienza, F., Rojo-Álvarez, J. L., Rosado-Muñoz, A., Vinagre, J. J., García-Alberola, A., & Camps-Valls, G. (2012). Feature selection using support vector machines and bootstrap methods for ventricular fibrillation detection. *Expert Systems with Applications*, 39(2), 1956-1967.

Aparicio, J. (2007). Una introducción al análisis envolvente de datos. *BEIO, Boletín de Estadística e Investigación Operativa*, ISSN 1889-3805, Vol. 23, N° 1, 2007, pags. 12-17.

Aparicio, J., Pastor, J. T., Vidal, F., & Zoffio, J. L. (2017). Evaluating productive performance: A new approach based on the product-mix problem consistent with Data Envelopment Analysis. *Omega*, 67, 134-144.

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

Araújo, C., Barros, C. P., & Wanke, P. (2014). Efficiency determinants and capacity issues in Brazilian for-profit hospitals. *Health care management science*, 17, 126-138.

Arnaboldi, M., Azzone, G., & Giorgino, M. (2014). Performance measurement and management for engineers. Academic Press.

Audet, C., Bignon, J., Cartier, D., Le Digabel, S., & Salomon, L. (2021). Performance indicators in multiobjective optimization. *European journal of operational research*, 292(2), 397-422.

Aytug, H. (2015). Feature selection for support vector machines using Generalized Benders Decomposition. *European Journal of Operational Research*, 244(1), 210-218.

Banker, R. D., Charnes, A., & Cooper, W. W. (1984). Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Management science*, 30(9), 1078-1092.

Banker, R. D., & Maindiratta, A. (1992). Maximum likelihood estimation of monotone and concave production frontiers. *Journal of Productivity Analysis*, 3(4), 401-415.

Banker, R. D. (1993). Maximum likelihood, consistency and data envelopment analysis: a statistical foundation. *Management science*, 39(10), 1265-1273.

Benítez-Peña, S., Blanquero, R., Carrizosa, E., & Ramírez-Cobo, P. (2019). Cost-sensitive feature selection for support vector machines. *Computers & Operations Research*, 106, 169-178.

Benítez-Peña, S., Bogetoft, P., & Morales, D. R. (2020). Feature selection in data envelopment analysis: A mathematical optimization approach. *Omega*, 96, 102068.

Beume, N., Fonseca, C. M., Lopez-Ibanez, M., Paquete, L., and Vahrenhold, J. (2009). On the complexity of computing the hypervolume indicator. *IEEE Transactions on Evolutionary Computation*, 13(5):1075–1082.

Blanco, V., Puerto, J., & Rodriguez-Chia, A. M. (2020). On lp-support vector machines and multidimensional kernels. *The Journal of Machine Learning Research*, 21(1), 469-497.

Bouraoui, A., Jamoussi, S., & BenAyed, Y. (2018). A multi-objective genetic algorithm for simultaneous model and feature selection for support vector machines. *Artificial Intelligence Review*, 50, 261-281.

- Bradley, P. S., & Mangasarian, O. L. (1998, July). Feature selection via concave minimization and support vector machines. In *ICML* (Vol. 98, pp. 82-90).
- Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Cart. Classification and regression trees*.
- Breiman, L. (2001). Random forests. *Machine learning*, 45, 5-32.
- Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2), 121-167.
- Candelieri, A., Giordani, I., Archetti, F., Barkalov, K., Meyerov, I., Polovinkin, A., ... & Zolotykh, N. (2019). Tuning hyperparameters of a SVM-based water demand forecasting system through parallel global optimization. *Computers & Operations Research*, 106, 202-209.
- Carrizosa, E., Martin-Barragan, B., & Morales, D. R. (2014). A nested heuristic for parameter tuning in support vector machines. *Computers & operations research*, 43, 328-334.
- Chambers, R. G., Chung, Y., & Färe, R. (1998). Profit, directional distance functions, and Nerlovian efficiency. *Journal of optimization theory and applications*, 98, 351-364.
- Chang and Li (2011). LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* 2 (3), 1–27.
- Charles, V., Aparicio, J., & Zhu, J. (Eds.). (2020). *Data science and productivity analytics*. New York: Springer.
- Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. *European journal of operational research*, 2(6), 429-444.
- Chen and Lin, 2006] Chen, Y.-W. and Lin, C.-J. (2006). Combining SVMs with various feature selection strategies. In *Feature Extraction*, pages 315–324. Springer.
- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1), 37-46.
- 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. *Journal of Productivity analysis*, 11, 5-42.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273-297.
- Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

Custódio, A. L., Madeira, J. A., Vaz, A. I. F., and Vicente, L. N. (2011). Direct multisearch for multiobjective optimization. *SIAM Journal on Optimization*, 21(3):1109–1140.

Daouia, A., Noh, H., & Park, B. U. (2016). Data envelope fitting with constrained polynomial splines. *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, 3-30.

Daraio, C., & Simar, L. (2005). Introducing environmental variables in nonparametric frontier models: a probabilistic approach. *Journal of productivity analysis*, 24, 93-121.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2), 182-197.

Deprins, D., y Simar, L. (1984). Measuring labor efficiency in post offices, *The Performance of Public Enterprises: Concepts and Measurements*, M. Marchand, P. Pestieau y H. Tulkens.

Dua, D., Graff, C., 2017. UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences, URL <http://archive.ics.uci.edu/ml>.

Dudzik, W., Nalepa, J., & Kawulok, M. (2021). Evolving data-adaptive support vector machines for binary classification. *Knowledge-Based Systems*, 227, 107221.

Efron, B. (1992). Bootstrap methods: another look at the jackknife (pp. 569-593). Springer New York.

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 Systems with Applications*, 162, 113783.

Esteve, M., Aparicio, J., Rodriguez-Sala, J. J., and Zhu, J. (2023). Random Forests and the measurement of super-efficiency in the context of Free Disposal Hull. *European Journal of Operational Research*, 304(2):729–744

Ezugwu, A. E., Shukla, A. K., Nath, R., Akinyelu, A. A., Agushaka, J. O., Chiroma, H., & Muhuri, P. K. (2021). Metaheuristics: a comprehensive overview and classification along with bibliometric analysis. *Artificial Intelligence Review*, 54, 4237-4316.

Färe, R., y Primont, D. (1995). *Multi-Output Production and Duality: Theory and Applications*. Kluwer Academic Publishers.

Faris, H., Hassonah, M. A., Al-Zoubi, A. M., Mirjalili, S., & Aljarah, I. (2018). A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture. *Neural Computing and Applications*, 30, 2355-2369.

- Farrell, M. J. (1957). The measurement of productive efficiency. *Journal of the royal statistical society: series A (General)*, 120(3), 253-281.
- Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, 27(8), 861-874.
- Florian, E., Sgarbossa, F., & Zennaro, I. (2021). Machine learning-based predictive maintenance: A cost-oriented model for implementation. *International Journal of Production Economics*, 236, 108114.
- Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning*. New York: Springer series in statistics.
- García-Pedrajas, N., de Haro-García, A., & Pérez-Rodríguez, J. (2014). A scalable memetic algorithm for simultaneous instance and feature selection. *Evolutionary Computation*, 22(1), 1-45.
- Gaudioso, M., Gorgone, E., Labbé, M., & Rodríguez-Chía, A. M. (2017). Lagrangian relaxation for SVM feature selection. *Computers & Operations Research*, 87, 137-145.
- Guerrero, N. M., Aparicio, J., & Valero-Carreras, D. (2022). Combining Data Envelopment Analysis and Machine Learning. *Mathematics*, 10(6), 909.
- Guillen, M. D., Aparicio, J., & Esteve, M. (2023). Gradient tree boosting and the estimation of production frontiers. *Expert Systems with Applications*, 214, 119134.
- Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, 46, 389-422.
- Guyon, I., Saffari, A., Dror, G., & Cawley, G. (2007, August). Agnostic learning vs. prior knowledge challenge. In *2007 International Joint Conference on Neural Networks* (pp. 829-834). IEEE.
- Halder, A., Chatterjee, S., & Dey, D. (2022). Adaptive morphology aided 2-pathway convolutional neural network for lung nodule classification. *Biomedical Signal Processing and Control*, 72, 103347.
- Hatami-Marbini, A., Emrouznejad, A., & Tavana, M. (2011). A taxonomy and review of the fuzzy data envelopment analysis literature: two decades in the making. *European journal of operational research*, 214(3), 457-472.
- Huang, C. L., & Wang, C. J. (2006). A GA-based feature selection and parameters optimization for support vector machines. *Expert Systems with applications*, 31(2), 231-240.
- Ierimonti, L., Venanzi, I., & Ubertini, F. (2021). ROC analysis-based optimal design of a spatio-temporal online seismic monitoring system for precast industrial buildings. *Bulletin of Earthquake Engineering*, 19, 1441-1466.

- Ismael, A. M., & Şengür, A. (2021). Deep learning approaches for COVID-19 detection based on chest X-ray images. *Expert Systems with Applications*, 164, 114054.
- Jatoth, C., Gangadharan, G. R., & Fiore, U. (2017). Evaluating the efficiency of cloud services using modified data envelopment analysis and modified super-efficiency data envelopment analysis. *Soft Computing*, 21, 7221-7234.
- Juo, J. C., Fu, T. T., Yu, M. M., & Lin, Y. H. (2015). Profit-oriented productivity change. *Omega*, 57, 176-187.
- Kim, M., Hiroyasu, T., Miki, M., & Watanabe, S. (2004). SPEA2+: Improving the performance of the strength Pareto evolutionary algorithm 2. In *Parallel Problem Solving from Nature-PPSN VIII: 8th International Conference*, Birmingham, UK, September 18-22, 2004. *Proceedings 8* (pp. 742-751). Springer Berlin Heidelberg.
- Knowles, J. D., & Corne, D. W. (2000). Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary computation*, 8(2), 149-172.
- Krooshof, P. W., Ustun, B., Postma, G. J., & Buydens, L. M. (2010). Visualization and recovery of the (bio) chemical interesting variables in data analysis with support vector machine classification. *Analytical Chemistry*, 82(16), 7000-7007.
- Kuosmanen, T., & Johnson, A. L. (2010). Data envelopment analysis as nonparametric least-squares regression. *Operations Research*, 58(1), 149-160.
- Kuosmanen, T., & Johnson, A. (2017). Modeling joint production of multiple outputs in StoNED: Directional distance function approach. *European Journal of Operational Research*, 262(2), 792-801.
- Kwiatkowski, J., & Sotor, J. (2022). Laser wavelength shift and dual-wavelength generation in continuous-wave operation of Ho: YAG laser pumped by thulium-doped fiber laser. *Optics & Laser Technology*, 146, 107544.
- Labbé, M., Martínez-Merino, L. I., & Rodríguez-Chía, A. M. (2019). Mixed integer linear programming for feature selection in support vector machine. *Discrete Applied Mathematics*, 261, 276-304.
- Li, Y., Shi, X., Yang, M., and Liang, L. (2017). Variable selection in data envelopment analysis via Akaike's information criteria. *Annals of Operations Research*, 253(1):453-476
- Lim, D. J., Anderson, T. R., & Shott, T. (2015). Technological forecasting of supercomputer development: The March to Exascale computing. *Omega*, 51, 128-135.
- Liu, H. H., Song, Y. Y., & Yang, G. L. (2019). Cross-efficiency evaluation in data envelopment analysis based on prospect theory. *European Journal of Operational Research*, 273(1), 364-375.

- Loh, L. X., Lee, H. H., Stead, S., & Ng, D. H. (2022). Manuka honey authentication by a compact atmospheric solids analysis probe mass spectrometer. *Journal of Food Composition and Analysis*, 105, 104254.
- Loupe, G., Wehenkel, L., Sutura, A., and Geurts, P. (2013). Understanding variable importances in forests of randomized trees. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc
- Lovell, C. K., & Pastor, J. T. (1995). Units invariant and translation invariant DEA models. *Operations research letters*, 18(3), 147-151.
- Maldonado, S., Pérez, J., Weber, R., & Labbé, M. (2014). Feature selection for support vector machines via mixed integer linear programming. *Information sciences*, 279, 163-175.
- Mavrotas, G. (2009). Effective implementation of the ε -constraint method in multi-objective mathematical programming problems. *Applied mathematics and computation*, 213(2), 455-465.
- Mavrotas, G., & Florios, K. (2013). An improved version of the augmented ε -constraint method (AUGMECON2) for finding the exact pareto set in multi-objective integer programming problems. *Applied Mathematics and Computation*, 219(18), 9652-9669.
- Nataraja, N. R. and Johnson, A. L. (2011). Guidelines for using variable selection techniques in data envelopment analysis. *European Journal of Operational Research*, 215(3):662–669
- O'Donnell, C. J. (2018). *Productivity and efficiency analysis*. Springer Singapore.
- Olesen, O.B., & N.C. Petersen (2016) Stochastic Data Envelopment Analysis: A review, *European Journal of Operational Research* 251(1), 2-21.
- Parmeter, C. F., & Racine, J. S. (2013). Smooth constrained frontier analysis. In *Recent advances and future directions in causality, prediction, and specification analysis* (pp. 463-488). Springer, New York, NY.
- Pastor, J. T., Ruiz, J. L., & Sirvent, I. (2002). A statistical test for nested radial DEA models. *Operations Research*, 50(4), 728-735.
- Perelman, S., & Santín, D. (2009). How to generate regularly behaved production data? A Monte Carlo experimentation on DEA scale efficiency measurement. *European Journal of Operational Research*, 199(1), 303-310.
- Peyrache, A., Rose, C., and Sicilia, G. (2020). Variable selection in data envelopment analysis. *European Journal of Operational Research*, 282(2):644–659. Postma et al., (2011)

- Raman, M. G., Somu, N., Kirthivasan, K., Liscano, R., & Sriram, V. S. (2017). An efficient intrusion detection system based on hypergraph-Genetic algorithm for parameter optimization and feature selection in support vector machine. *Knowledge-Based Systems*, 134, 1-12.
- Roshani, M., Phan, G. T., Ali, P. J. M., Roshani, G. H., Hanus, R., Duong, T., ... & Kalmoun, E. M. (2021). Evaluation of flow pattern recognition and void fraction measurement in two phase flow independent of oil pipeline's scale layer thickness. *Alexandria Engineering Journal*, 60(1), 1955-1966.
- Ruggiero, J. (2005). Impact assessment of input omission on DEA. *International Journal of Information Technology & Decision Making*, 4(03):359–368
- Sanz, H., Valim, C., Vegas, E., Oller, J. M., and Reverter, F. (2018). SVM-RFE: selection and visualization of the most relevant features through non-linear kernels. *BMC bioinformatics*, 19(1):1–18.
- 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. *Computers & Operations Research*, 75, 163-173.
- Simar, L., & Wilson, P. W. (1998). Sensitivity analysis of efficiency scores: How to bootstrap in nonparametric frontier models. *Management science*, 44(1), 49-61.
- Simar, L., & Wilson, P. W. (2000). A general methodology for bootstrapping in non-parametric frontier models. *Journal of applied statistics*, 27(6), 779-802.
- Sirvent, I., Ruiz, J. L., Borrás, F., and Pastor, J. T. (2005). A Monte Carlo evaluation of several tests for the selection of variables in DEA models. *International Journal of Information Technology & Decision Making*, 4(03):325–343
- Srinivas, N., & Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3), 221-248.
- Tsionas, M. G. (2022). Efficiency estimation using probabilistic regression trees with an application to Chilean manufacturing industries. *International Journal of Production Economics*, page 108492.
- Ueda, T. and Hoshiai, Y. (1997). Application of principal component analysis for parsimonious summarization of DEA inputs and/or outputs. *Journal of the Operations Research Society of Japan*, 40(4):466–478.
- Van Veldhuizen, D. A. (1999). Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. Air Force Institute of Technology.
- Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer, New York.
- Vapnik, V. (1998). *Statistical Learning Theory*. Wiley, New York.

- Vapnik, V., & Chervonenkis, A. Y. (1964). A class of algorithms for pattern recognition learning. *Avtomat. i Telemekh*, 25(6), 937-945.
- Vapnik, V. N., & Chervonenkis, A. Y. (1974). The method of ordered risk minimization, I. *Avtomatika i Telemekhanika*, 8, 21-30.
- Vazquez, E., & Walter, E. (2003). Multi-output support vector regression. *IFAC Proceedings Volumes*, 36(16), 1783-1788.
- Venkatesh, B., & Anuradha, J. (2019). A review of feature selection and its methods. *Cybernetics and information technologies*, 19(1), 3-26.
- Vijayan, N. M., Johnson, M. S., & Jacob, J. (2021). Convex-optimization-based constrained control strategy for 3-DOF tandem helicopter using feedback linearization. *Journal of Optimization Theory and Applications*, 191, 736-755.
- Wang, H. J. and Schmidt, P. (2002). One-step and two-step estimation of the effects of exogenous variables on technical efficiency levels. *Journal of Productivity Analysis*, 18.
- Xue, Y., Tang, Y., Xu, X., Liang, J., & Neri, F. (2021). Multi-objective feature selection with missing data in classification. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(2), 355-364.
- Yang, Y., Hu, X., & Jiang, H. (2022). Group penalized logistic regressions predict up and down trends for stock prices. *The North American Journal of Economics and Finance*, 59, 101564.
- Zelenyuk, V. (2020). Aggregation of inputs and outputs prior to Data Envelopment Analysis under big data. *European Journal of Operational Research*, 282(1):172–187
- Zhang, Y., Gong, D. W., & Cheng, J. (2015). Multi-objective particle swarm optimization approach for cost-based feature selection in classification. *IEEE/ACM transactions on computational biology and bioinformatics*, 14(1), 64-75.
- Zhao, M., Fu, C., Ji, L., Tang, K., & Zhou, M. (2011). Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes. *Expert Systems with Applications*, 38(5), 5197-5204.
- Zhou, Q., Zhou, H., & Li, T. (2016). Cost-sensitive feature selection using random forest: Selecting low-cost subsets of informative features. *Knowledge-based systems*, 95, 1-11.
- Zitzler, E. and Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms-a comparative case study. In *Parallel Problem Solving from Nature-PPSN V: 5th International Conference Amsterdam, The Netherlands September 27–30, 1998 Proceedings* 5, pages 292–301. Springer.

Zitzler, E. (1999). Evolutionary algorithms for multiobjective optimization: Methods and applications, volume 63. Shaker Ithaca.

Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2), 173-195.

AGRADECIMIENTOS

A Juan y Javier por aguantarme en mis desvaríos.

A Mercedes, Nadia y Raúl porque sin ellos no habría artículos.

A Rafa por arreglarme el clúster cuando lo necesitaba.

A Sixto por hacer del insulto una manera de hacerme levantar de la cama cada día.

A mis compañeros/as por soportarme.

Al PAS, los/as conserjes, personal de limpieza e investigadores/as del CIO por hacer de él un sitio especial.

A Jesús por convertirse en el hermano pequeño que no tuve.

A mis maestros y profesores del colegio, instituto y universidad.

A mi familia por darme los valores y la fuerza necesaria.

A los que no creyeron y no creen en la ciencia, que se vayan a donde dijo Labordeta.

Y sobre todo a mi hermano.