



Departamento de
Ingeniería Industrial
Universidad de Concepción

Metaheurística multiobjetivo para support vector machines con feature selection

POR

Mathias Felipe Badilla Salamanca

Memoria de título presentada a la Facultad de Ingeniería de la Universidad de Concepción
para optar al título profesional de Ingeniero Civil Industrial

Profesores guía

Carlos Emilio Contreras Bolton

Rosa Daniela Medina Durán

Agosto 2024

Concepción Chile

©Mathias Felipe Badilla Salamanca

© 2024, *Mathias Felipe Badilla Salamanca*

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento.

Esta memoria está dedicada a mi familia
Miryam, Luciano (Chanito) y Luciano (papá)



Eclipse total de sol, julio 2019, Chile

Agradecimientos

Quisiera agradecer de forma muy especial a mis padres, Luciano, por ser un papá con quien comparto mi espíritu curioso y que desde chiquito me dijo que programar era divertido, a pesar de que yo sólo quería jugar al computador; a Miryam, quien no sólo me heredó su risa, sino también quien me dejó el ejemplo de decir “yo puedo” y darle adelante, a ambos por su apoyo en mi proceso educativo y de desarrollo personal y porque en cada idea loca siempre estuvieron para decirme que sí y hacer de mí la persona que soy hoy día. A mi hermano por ser esa continua fuente de desahogo y palmaditas en la espalda para seguir adelante. A mi grupo de amigos por tener la paciencia de explicarme cada vez que lo necesité y tirarme una talla cuando se extrañaba casita o durante las pausas activas de 2 horas. A mis profesores guía, Carlos y Rosa, por su buena onda y confianza depositada mí a lo largo de toda la carrera, tristemente no podré seguir siendo su ayudante de Opti. Finalmente, una mención honrosa mi perrita Luna, que se desveló todos los fines de semana que me quedé estudiando y que, sin quejas, se dejaba abrazar cuando lo necesité.

Esta memoria de título fue parcialmente apoyada por ANID mediante el FONDECYT INICIACIÓN 11241132.

Resumen

La selección de *features* es una área importante en el aprendizaje supervisado, especialmente al manejar grandes volúmenes de datos con muchas features. Support Vector Machines (SVM) ha demostrado ser un modelo sencillo y eficiente para trabajar en la selección de features gracias al uso regresores específicos para cada feature, los cuales son anulados si no son relevantes. Por ello, esta memoria de título propone una metaheurística multiobjetivo para mejorar la selección de features aplicando SVM. La metaheurística multiobjetivo se basa en el NSGA-II y busca optimizar tanto el rendimiento predictivo como la eficiencia computacional del modelo.

El enfoque propuesto es evaluado en diversos conjuntos de datos, considerando métricas de clasificación y tiempo de ejecución. Los resultados muestran que la metaheurística presentada no solo reduce la dimensionalidad del problema sino que también mantiene o mejora la calidad de las predicciones. Este trabajo contribuye al campo de la inteligencia artificial y la investigación de operaciones al demostrar que el uso de metaheurísticas pueden ser efectivas en la mejora de algoritmos de aprendizaje supervisado como SVM.

Abstract

Feature selection is an important area in supervised learning, especially when handling large volumes of data with many features. Support Vector Machines (SVM) has proven to be a simple and efficient model to work on feature selection thanks to the use of feature-specific regressors, which can be overridden if they are not relevant. Therefore, this thesis proposes a multi-objective metaheuristic to improve feature selection using SVM. The multi-objective metaheuristic is based on NSGA-II and seeks to optimize both the predictive performance and the computational efficiency of the model.

The proposed approach is evaluated on various datasets, considering classification metrics and runtime. The results show that the designed metaheuristic not only reduces the dimensionality of the problem but also maintains or improves the quality of the predictions. This work contributes to the field of artificial intelligence and operations research by demonstrating that the use of metaheuristics can be effective in improving supervised learning algorithms such as SVM.

Índice General

Índice General

Índice de Figuras

Índice de Tablas

1. Introducción	1
1.1. Motivación	1
1.1.1. Objetivos	2
1.1.2. Objetivo general	2
1.1.3. Objetivos específicos	2
1.2. Estructura del documento	2
2. Support vector machines con feature selection	3
2.1. Descripción del problema	3
2.2. Modelo de programación matemático	5
2.3. Revisión bibliográfica	7
3. NSGA-II propuesto	11
3.1. Introducción	11
3.2. Algoritmo general	11
3.3. Distancia de <i>crowding</i>	13
3.4. Representación de la solución	14
3.5. Población inicial	14
3.6. Torneo basado en la dominancia	15
3.7. Cruzamiento	15
3.8. Mutación	16

4. Experimentos computacionales	18
4.1. Descripción del conjunto de datos	18
4.2. Métricas utilizadas	20
4.3. Configuración del algoritmo	22
4.4. Configuración de los experimentos	22
5. Resultados y discusión	24
6. Conclusión	30
Referencias Bibliográficas	31
Anexos	36

Índice de Figuras

1.	Estructura SVM.	4
2.	Representación del individuo.	14
3.	Representación del cromosoma del ejemplo de la Figura 1.	14
4.	Ejemplo de aplicación del operador de PMX.	15
5.	Distribución de las clases por instancia.	20
6.	Matriz de confusión.	21
7.	Ejemplo K-fold.	23
8.	Ejemplo discordancia Cohen Kappa y FSC.	27
9.	Funciones de pérdida por instancia.	29

Índice de Tablas

1.	Resumen revisión de literatura.	10
2.	Características de los conjuntos de datos.	20
3.	Parámetros utilizados	22
4.	Comparación de resultados en los promedios de los promedios de las métricas.	25
5.	Comparación de resultados en los promedios de los cinco máximos.	25
6.	Media de <i>features</i> consideradas por los mejores individuos	27
7.	Resultados Arcene	36
8.	Resultados BioResponse	37
9.	Resultados Duke	38
10.	Resultados GermanCredit	39
11.	Resultados Gisette	40
12.	Resultados Ionosphere	41
13.	Resultados WBC	42

Índice de Algoritmos

1. NSGA-II	12
------------------	----

Capítulo 1

Introducción

Este capítulo presenta una breve motivación e introducción al estudio, el objetivo general y los objetivos específicos y finalmente, presenta la estructura del presente documento.

1.1. Motivación

La clasificación supervisada es un área que, a medida que pasa el tiempo, más se usa en cosas cotidianas del día a día de las personas. En general, considera conjuntos de datos que, al ser sometidos a ciertas suposiciones, principalmente estadísticas, permiten la creación de modelos con el objetivo de comprender y anticipar cómo se comportarán nuevas muestras en el futuro. Este enfoque permite analizar patrones y tendencias en datos existentes, facilitando así la toma de decisiones informadas y la predicción de resultados para situaciones aún no observadas.

En el ámbito del aprendizaje supervisado, existen varios algoritmos comúnmente utilizados para llevar a cabo tareas de clasificación. Entre ellos está el algoritmo de Support Vector Machine (SVM) ([Cortes & Vapnik, 1995](#)), que se enfoca en encontrar un hiperplano óptimo que divida el espacio generando una separación entre las clases. Estos algoritmos son entrenados con grandes volúmenes de datos, donde no siempre todas las variables aportan información que sea valiosa al modelo, por lo cual se genera la necesidad de tener: i) algoritmos que se entrenen de manera rápida y ii) algoritmos que puedan realizar una selección de las features sin perder la calidad de las predicciones, esta técnica es conocida como Feature Selection (FS) ([Miao & Niu, 2016](#)).

Por otro lado, en el área de investigación de operaciones, se ha estado investigando la aplicación de metaheurísticas multiobjetivo como NSGA-II ([Deb et al., 2002](#)) a estas problemáticas de clasificación o de selección de features, con el fin de buscar soluciones que consuman menos recursos computacionales en términos de tiempo. Así, recientemente, [Alcaraz et al. \(2022\)](#) y [Valero-Carreras et al. \(2023\)](#) han

utilizado un enfoque metaheurístico multiobjetivo para abordar FS-SVM con resultados predictivos y temporales que pueden mejorar cambiando la estructura de la metaheurística y la representación del individuo. Estas mejoras son especialmente relevantes dado los costos de oportunidad que generan los tiempos de espera en el procesamiento de grandes volúmenes de datos, por el retraso en la obtención de resultados o por los costos asociados al cómputo (arriendo de servidores, luz, entre otros). Además, mejorar la calidad de las predicciones permite tomar decisiones informadas y precisas en áreas clave como las finanzas, la banca, logística e incluso en sectores sensibles como la medicina. Por tanto, esta memoria de título propone una metaheurística multiobjetivo aplicado al algoritmo de SVM para realizar FS analizando su rendimiento predictivo y temporal.

1.1.1. Objetivos

A continuación, se presenta el objetivo general junto a los objetivos específicos de esta memoria de título.

1.1.2. Objetivo general

Proponer una metaheurística multiobjetivo para aproximar la curva de pareto de FS considerando los objetivos de SVM, además de medir su rendimiento con métricas sobre la precisión de su predicción.

1.1.3. Objetivos específicos

- Realizar una revisión del estado del arte de FS y SVM.
- Proponer un algoritmo multiobjetivo que mejore en rendimiento temporal y calidad de predicciones.
- Analizar los resultados en términos temporales y de la calidad de las predicciones obtenidas.

1.2. Estructura del documento

El documento está estructurado de la siguiente forma. En el [Capítulo 2](#) se presenta formalmente el problema, incluyendo modelos matemáticos relacionados y una revisión de la literatura de métodos que abordan el FS-SVM. El [Capítulo 3](#) presenta la metaheurística propuesta con sus componentes y explicación. El [Capítulo 4](#) aborda los experimentos computacionales y sus resultados son discutidos en el [Capítulo 5](#). Finalmente, se presentan las conclusiones en el [Capítulo 6](#).

Capítulo 2

Support vector machines con feature selection

Este capítulo presenta formalmente el problema y el modelo matemático asociado. Además, se revisa la literatura relevante y reciente de: FS y SVM, enfocándose en los algoritmos del estado del arte que resuelven el problema del FS-SVM y sus características.

2.1. Descripción del problema

El problema de FS se centra en identificar un subconjunto de *features* de un conjunto de datos ya sea por ser redundantes, ruidosas o irrelevantes para su eliminación. Por tanto, FS es crucial porque incluir un mayor número de *features* no siempre mejora los resultados obtenidos al hacer un análisis. Incluso podrían deteriorarlos. Además, un mayor número de *features* aumenta el costo computacional al momento de procesar los datos en términos temporales.

De manera paralela, SVM es un modelo de inteligencia artificial que busca realizar una división óptima de un espacio n-dimensional mediante dos hiperplanos que buscan la minimización del error empírico y estructural para tareas de clasificación binaria. El error empírico refiere a la discrepancia entre las etiquetas predichas por el modelo y las etiquetas reales en el conjunto de datos de entrenamiento. Mientras que el error estructural se relaciona con la distancia entre ambos hiperplanos generados, ya que a mayor distancia menor es la incertidumbre, así mejorando la capacidad del modelo para hacer inferencias precisas en datos nuevos.

Los SVM son introducidos por primera vez en [Cortes & Vapnik \(1995\)](#). Estos buscan definir un conjunto de vectores que soporten un hiperplano y que pueda dividir de manera eficiente un espacio de entrada de alta dimensión. Esta división se utiliza para realizar clasificaciones binarias, determinando

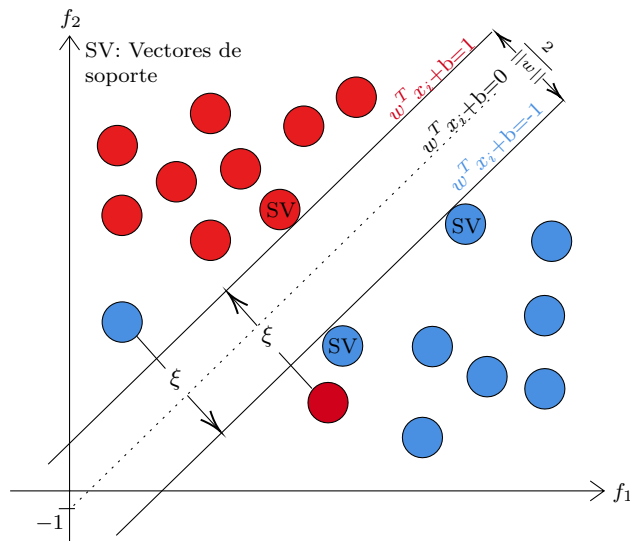


Figura 1: Estructura SVM.

si los datos se encuentran “por encima” o “por debajo” de estos vectores de soporte. En este contexto, se busca optimizar dos objetivos fundamentales: maximizar la distancia entre los hiperplanos definidos por los vectores de soporte y minimizar el error de clasificación. Así, el SVM es un recurso esencial en diversas aplicaciones, desde la medicina hasta la visión por computador, contribuyendo significativamente a la resolución de problemas del mundo real (Cervantes et al., 2020). La Figura 1 ilustra la idea de vectores de soporte, con un conjunto de datos en dos dimensiones. Por tanto, es necesario el uso de tres vectores para sostener a los planos que dividen el espacio. Además, se muestra la distancia entre los planos, considerada en la función objetivo.

Al utilizar el SVM para abordar el problema de FS, surge la noción de estimar un hiperplano en el espacio dimensional que implique anular los regresores asociados a ciertas *features* del conjunto de datos. Esta acción conduce a una reducción del espacio dimensional, donde solo las *features* relevantes para la tarea de clasificación son consideradas en la toma de decisiones del modelo.

Con el desarrollo de algoritmos de aprendizaje supervisado como SVM, ha surgido el desafío de manejar conjuntos de datos cada vez más grandes de manera eficiente. A medida que aumenta la cantidad de datos, el proceso de entrenamiento y evaluación de modelos se vuelven computacionalmente costoso. Es aquí donde entra en juego el estudio de FS, ya que, el FS busca abordar el problema de la alta dimensionalidad de los conjuntos de datos, reduciendo el número de *features* de entrada, mientras se conserva la información esencial para resolver el problema en cuestión. Esto no solo mejora la eficiencia computacional, sino que también puede conducir a modelos más simples y generalizables.

Los enfoques de selección de *features* se clasifican en tres: *Filter*, *Wrapper* y *Embedded* (Miao & Niu, 2016). *Filter* evalúa la relevancia de cada *feature* de forma independiente del modelo. *Wrapper*

utiliza un modelo predictivo para evaluar la calidad de diferentes subconjuntos de *features*. *Embedded* integra la selección de *features* en el proceso de entrenamiento del modelo, seleccionando *features* relevantes durante la construcción del modelo. El algoritmo propuesto en esta investigación presenta un enfoque *embedded*, puesto que la selección de *features* es realizada de manera simultánea con el entrenamiento del modelo, ya que ambas son características de la representación de la solución del algoritmo propuesto.

2.2. Modelo de programación matemático

El SVM puede ser modelado como un problema de optimización cuadrática convexo (Cortes & Vapnik, 1995). Formalmente: sea un conjunto de datos Ω dividido en dos clases, donde cada objeto $i \in \Omega$ está representado por un par $(\mathbf{x}_i, y_i) \in \mathbb{R}^m \times \{-1, 1\}$, y n es el número de *features* analizadas para cada elemento de Ω . El SVM determina un hiperplano $f(x) = w^T X + b$ que separa de manera óptima los datos de entrenamiento (Labbé et al., 2019). La medida del error se realiza mediante la variable ξ_i ($i = \{1, \dots, m\}$) siendo la distancia de los puntos mal clasificados al hiperplano que divide su clase, esta distancia se mide en los m puntos del conjunto de datos y es considerada 0 en caso de estar bien clasificado. La formulación de Cortes & Vapnik (1995) es presentada a continuación:

$$\text{minimizar } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i \quad (1)$$

sujeto a:

$$y_i(w^T x_i + b) \geq 1 - \xi_i \quad \forall i \in \{1, \dots, m\} \quad (2)$$

$$\xi_i \in \mathbb{R}_0^+ \quad \forall i \in \{1, \dots, m\} \quad (3)$$

$$w_j \in \mathbb{R} \quad \forall j \in \{1, \dots, n\} \quad (4)$$

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

El primer término de la función objetivo (1) $(\frac{\|w\|^2}{2})$ corresponde al riesgo estructural del modelo dado que $\frac{1}{\|w\|}$ representa la distancia entre los hiperplanos. El segundo término $(\sum_{i=1}^m \xi_i)$ es referido al riesgo empírico y busca la minimización de los vectores mal clasificados que son multiplicados por un hiperparámetro C . Así, C otorga una ponderación de importancia entre ambos objetivos. Respecto a las restricciones (2), corresponden a la definición de ambos hiperplanos, así, el vector y indica si el

vector de entrenamiento es de la clase positiva o negativa, y al ser multiplicado por la evaluación de x , indica si está bien o mal clasificado. En una mal clasificación, el error es sumado a la variable ξ , penalizando la función objetivo. Finalmente, las ecuaciones (3)-(5) corresponden a las restricciones de dominio.

El FS-SVM también puede ser formulado como un modelo de programación lineal entera mixta (MILP, por sus siglas en inglés: *mixed integer linear programming*) como el propuesto por [Labbé et al. \(2019\)](#). Este modelo necesita una variable binaria: t_j ($j \in \{1, \dots, n\}$) que se encarga de restringir las *features* que participan en la solución. Así el modelo de programación lineal entero mixto consiste en la adición de:

$$\sum_{j=1}^n t_j = p \tag{6}$$

$$|w_j| \leq Mt_j \quad \forall j \in \{1, \dots, n\} \tag{7}$$

$$t_j \in \{0, 1\} \quad \forall j \in \{1, \dots, n\} \tag{8}$$

Las restricciones (6) determina la cantidad de *features* t_j que pueden tomar el valor 1 a una cantidad de *features* preestablecidas en un parámetro p . De esta manera en las restricciones (7) mediante el uso de un número muy grande (M) se induce que la cantidad de valores máximos que pueden ser distintos de 0 sean p . La ecuación (7) son las restricciones de dominio.

Recientemente, [Alcaraz et al. \(2022\)](#) proponen un modelo multiobjetivo basado en el modelo de [Labbé et al. \(2019\)](#) y redefinen la función objetivo (1) en dos objetivos (9) y (10), eliminando así el parámetro C .

$$\text{minimizar} \quad O_1 = \frac{1}{2} \|w\|^2 \tag{9}$$

$$\text{minimizar} \quad O_2 = \sum_{i=1}^m \xi_i \tag{10}$$

El modelo abordado en esta memoria contempla la modificación del objetivo O_2 (9) y el reemplazo del parámetro p por una variable. El objetivo O_2 y la variable p quedan de la siguiente manera:

$$\text{minimizar } O_2 = \sum_{i=1}^m \alpha_i \xi_i \quad (11)$$

$$p \in \{1, \dots, n\} \quad (12)$$

Ahora p es una variable y permite al modelo seleccionar entre una o el total de las *features* que incluya el conjunto de datos. Así, el modelo abordado considera las restricciones (2)-(8), (12) y la función objetivo (9) y (11). Es posible notar que aumenta la complejidad computacional de este modelo, ya que las formas de seleccionar las *features* crece de manera exponencial según $\sum_{i=1}^n \binom{n}{i}$, y de manera simultánea se realiza la asignación de los pesos para cada variable seleccionada. Esta complejidad exponencial justifica el uso de metaheurísticas para abordar el problema dado el costo computacional del uso de métodos exactos.

De igual forma, el modelo propuesto contempla la adición del parámetro α_i ($i = \{1, \dots, m\}$) en O_2 que hace una ponderación del error a considerar según la clase a la que pertenece el vector mal clasificado en función de su prevalencia (Yang et al., 2005). Por ejemplo, considerando una instancia donde el 70% de los vectores son de clase 1 y el 30% de clase -1, el ponderador del error de clasificación en 1 tendría un valor α de 0,7 y el ponderador del error de clasificación en -1 un valor α de 0,3.

2.3. Revisión bibliográfica

La revisión bibliográfica se enfoca en estudios relacionados con la selección de características en SVM (SVM-FS), particularmente en modelo y algoritmos basados en el modelo de SVM de margen suave con FS. Sin embargo, existe otra línea de investigación en SVM-FS que buscan optimizar simultáneamente los parámetros del modelo SVM y la selección de características. Por ejemplo, Bouraoui et al. (2018) emplean un algoritmo genético multiobjetivo para ajustar los parámetros del SVM y seleccionar un subconjunto óptimo de características. Faris et al. (2018) también utilizan una metaheurística para optimizar tanto las características como los parámetros del SVM, logrando una alta precisión y una significativa reducción en el número de características. Candelieri et al. (2019) aplican la optimización global paralela para ajustar los hiperparámetros del SVM en un contexto aplicado, mejorando el rendimiento computacional. Dudzik et al. (2021) proponen una técnica evolutiva que optimiza el SVM en conjuntos de datos grandes y desbalanceados. Así, se pueden encontrar varios otros enfoque más actuales como los propuesto por Wang et al. (2023), Bomze et al. (2024) y Zandvakili et al. (2024).

Por lo tanto, este trabajo y la presente revisión solo se centran en estudios basados en el modelo de SVM de margen suave con FS utilizando tanto modelos de programación lineal y metaheurísticas multiobjetivo.

[Maldonado et al. \(2014\)](#) fueron pioneros en abordar el desafío de combinar SVM con FS mediante un enfoque exacto, considerando la selección de *features* como un problema NP-hard, introduciendo dos formulaciones MILP que optimizan simultáneamente la selección de variables y la construcción del clasificador. Este enfoque demostró que la reducción del número de *features* relevantes puede mejorar el rendimiento del clasificador significativamente en comparación con técnicas heurísticas tradicionales. Una de las estrategias exactas para FS-SVM fue el método basado en la Descomposición Generalizada de Benders presentado por [Aytug \(2015\)](#). Este enfoque permitió descomponer el problema original en un problema maestro y varios subproblemas manejables, facilitando la selección óptima de *features* durante la inducción del modelo SVM. Este método mostró una sensibilidad considerable a la calibración del parámetro de penalización de *features*.

Posteriormente, [Gaudioso et al. \(2017\)](#) exploraron la relajación Lagrangiana como un enfoque alternativo para mejorar la eficiencia en la selección de *features*. La relajación Lagrangiana permite una mayor flexibilidad en la formulación del problema, facilitando la inclusión de restricciones que penalizan la selección de muchas *features*. Este método se destaca por su capacidad para equilibrar la complejidad del modelo y su rendimiento predictivo, aunque su desempeño también depende críticamente de la elección de los parámetros.

Por otro lado, [Benítez-Peña et al. \(2019\)](#) proponen un nuevo enfoque al abordar la selección de *features* en SVM desde una perspectiva sensible a los costos cuando obtener los datos de algunas *features* sea muy difícil o económicamente costoso. Este enfoque considera no solo la precisión del modelo, sino también los costos asociados con la selección de *features*, lo que es particularmente relevante en aplicaciones donde algunas *features* pueden ser más costosas o difíciles de obtener que otras. La integración de la sensibilidad a los costos añade una dimensión adicional al proceso de selección, proporcionando soluciones más equilibradas y prácticas en escenarios del mundo real.

Paralelamente, [Labbé et al. \(2019\)](#) ampliaron el campo al proponer una formulación de MILP que incluye una restricción de presupuesto, estableciendo un límite en el número de *features* utilizadas en el proceso de clasificación. Su estudio se centra en la eficacia de esta formulación, ofreciendo tanto procedimientos exactos como heurísticos para resolver el problema y validando el modelo a través de comparaciones con métodos de clasificación clásicos.

[Baldomero-Naranjo et al. \(2021\)](#) presentan un modelo robusto basado en SVM que aborda simultáneamente la detección de outliers y la selección de características. El modelo propuesto incorpora un

margen de error de pérdida rampante y una restricción presupuestaria para limitar el número de características seleccionadas. La formulación del clasificador se modela mediante una formulación entera mixta con parámetros big M . Los autores desarrollan dos enfoques para resolver el modelo: uno exacto y otro heurístico. La eficacia del enfoque heurístico se valida comparando la calidad de las soluciones obtenidas con el enfoque exacto. Además, los clasificadores obtenidos mediante el método heurístico se prueban y comparan con modelos existentes basados en SVM para demostrar su eficiencia en conjuntos de datos reales.

El avance de modelos de MILP para la selección de *features* se complementa con enfoques más recientes. Las investigaciones de [Alcaraz et al. \(2022\)](#) y [Valero-Carreras et al. \(2023\)](#) introducen el NSGA-II para abordar el problema de FS-SVM desde una perspectiva multiobjetivo. Este enfoque permite especificar el número deseado de *features* y determina automáticamente cuáles deben incluirse en la solución del problema, buscando optimizar tanto el rendimiento del modelo SVM como la selección de *features* de manera conjunta. Estos estudios reflejan una evolución en las técnicas de selección de *features*, con estrategias multiobjetivo que integran la optimización del rendimiento del clasificador con la selección eficiente de *features*.

La investigación reciente mencionada, por [Alcaraz et al. \(2022\)](#), compara NSGA-II con AUGMECON2, un algoritmo exacto de optimización lexicográfica propuesto por [Mavrotas & Florios \(2013\)](#). Este enfoque jerárquico busca soluciones que sean óptimas en términos de un objetivo principal, considerando otros objetivos solo si hay múltiples soluciones igualmente óptimas en el objetivo principal. Esta comparación evalúa la eficiencia computacional y la capacidad de dominar soluciones en el contexto específico de SVM-FS.

Por último, en la investigación de [Valero-Carreras et al. \(2023\)](#), reemplazan el primer objetivo de la investigación de [Alcaraz et al. \(2022\)](#) por la cantidad de vectores mal clasificados y evalúan NSGA-II en términos de la calidad predictiva en la selección de *features*. De esta manera, se amplía su aplicación más allá de la dominación de soluciones hacia la capacidad de hacer predicciones precisas, haciendo un nexo entre la perspectiva de la optimización combinatoria y la investigación de inteligencia artificial. Un resumen de los trabajos referidos es presentado en la [Tabla 1](#), donde se presenta el problema abordado, los métodos utilizados para abordarlo, la perspectiva variable o estática del número de *features* y los números de objetivos. Se puede notar como dentro de los métodos exactos existen diversas maneras de resolver el problema y como estas se combinan con perspectivas de *features* como parámetro o variable.

Tabla 1: Resumen revisión de literatura.

Trabajo	Metodo de resolución	Número de <i>features</i>	Objetivos
Maldonado et al. (2014)	MILP	Parámetro	1
Aytug (2015)	<i>Generalized Benders Decomposition</i>	Parámetro y Variable	1
Gaudioso et al. (2017)	MILP	Variable	1
Benítez-Peña et al. (2019)	MILP	Variable	1
Labbé et al. (2019)	MILP	Parámetro	1
Baldomero-Naranjo et al. (2021)	MILP y heurística	Variable	1
Alcaraz et al. (2022)	Augmecon2 y NSGA-II	Parámetro (5)	2
Valero-Carreras et al. (2023)	NSGA-II	Parámetro (5)	2
Presente memoria	NSGA-II	Variable	2

Capítulo 3

NSGA-II propuesto

Este capítulo presenta el NSGA-II propuesto. Se proporciona una definición general del algoritmo, así como una descripción específica de los operadores que lo componen y su funcionamiento.

3.1. Introducción

El NSGA-II es un algoritmo genético y que se inspira en la adaptación biológica de las especies, que operan mediante la generación, selección y combinación de soluciones candidatas para encontrar soluciones óptimas o cercanas (Eiben & Smith, 2015). Por tanto, NSGA-II es un algoritmo genético que puede abordar problemas multiobjetivo (dos o más objetivos). Así, NSGA-II se centra en encontrar soluciones no dominadas en un espacio multiobjetivo, utilizando los operadores genéticos previamente mencionados para generar y mejorar poblaciones de soluciones candidatas.

Además, NSGA-II está basado en el principio de la no dominancia para generar y mantener un conjunto de soluciones frente a múltiples objetivos. La no dominancia entre soluciones se refiere a la imposibilidad de determinar, dado dos individuos, la superioridad absoluta de uno sobre otro cuando se consideran dos objetivos como funciones objetivo. En cambio, la dominancia implica que un individuo es superior en todos los objetivos respecto a otro. Además, un conjunto de soluciones no dominadas entre sí comparten una misma curva de indiferencia (Malczewski & Rinner, 2015). Si este conjunto no es dominado por ninguna otra solución, se denomina frontera de Pareto.

3.2. Algoritmo general

El funcionamiento de NSGA-II es descrito en el [Algoritmo 1](#) y comienza con la creación de la población inicial (línea 1) compuesta por N individuos, que son evaluados respecto a los objetivos O_1 y O_2

de la **función objetivo** (línea 2). Una vez evaluados los objetivos, se deben determinar las distintas fronteras de no dominancia, las cuales son utilizadas en etapas como el cruzamiento o la selección de individuos para definir aquellos aptos para prevalecer durante las generaciones. Para esto, se aplica el operador **fast-non-dominated-sort**, que busca la primera frontera de no dominancia del conjunto de individuos que recibe como entrada. Dado que es ejecutado por primera vez este operador de ordenamiento, es posible descartar aquellos individuos que ya formen parte de la primera frontera y ejecutarlo nuevamente con los individuos restantes, repitiendo de esta forma el operador hasta que todos los individuos pertenezcan a alguna frontera (línea 3). Luego, el algoritmo entra al ciclo principal

Algoritmo 1 NSGA-II

Input: Número de individuos (N)

Output: Individuos de la frontera de pareto (P_t)

```

1  $P_0 \leftarrow$  crear-población-inicial( $N$ )
2  $P_0 \leftarrow$  evaluar-población( $P_0$ )
3  $F \leftarrow$  fast-non-dominated-sort( $P_0$ )
4  $t \leftarrow 0$ 
5 while tiempo limite ( $t$ ) no es excedido do
6   for  $j \leftarrow 1$  to  $N$  do
7      $(p_1, p_2) \leftarrow$  selección-por-torneo-de-dominancia( $P_t$ )
8      $Q_t^j \leftarrow$  entrecruzamiento( $p_1, p_2$ )
9     if random( $0, 1$ )  $\leq P_t^{m_1}$  then
10    |  $Q_t^j \leftarrow$  mutación-1( $Q_t^j$ )
11    if random( $0, 1$ )  $\leq P_t^{m_2}$  then
12    |  $Q_t^j \leftarrow$  mutación-2( $Q_t^j$ )
13   $Q_t \leftarrow$  evaluar-población( $Q_t$ )
14   $R_t \leftarrow P_t \cup Q_t$ 
15   $F \leftarrow$  fast-non-dominated-sort( $R_t$ )
16   $P_{t+1} \leftarrow \emptyset$ 
17   $i \leftarrow 1$ 
18  while  $|P_{t+1}| + |F_i| \leq N$  do
19  |  $P_{t+1} \leftarrow P_{t+1} \cup F_i$ 
20  |  $i \leftarrow i + 1$ 
21  if  $|P_{t+1}| < N$  then
22  | selección-última-frontera( $F_i$ )
23   $t \leftarrow t + 1$ 

```

que consiste en el ciclo evolutivo (línea 5). Se comienza la creación de los nuevos N individuos que son agregados a la población (líneas 6-12). Para crear cada nuevo individuo, se seleccionan al azar dos individuos de la población original y se los somete a un torneo basado en la dominancia para determinar quién será el progenitor 1 y quién será el progenitor 2.

Luego, se realiza el cruzamiento de los progenitores seleccionados, para que luego su descendencia pueda ser sometida a uno, dos o ningún operador de mutación (líneas 8-12), según las probabilidades de mutación (p^{m_1} y p^{m_2}). Al terminar la creación de los nuevos N individuos, se procede a evaluar la nueva población y se une a la población original (líneas 13-14). Esto genera un total de $2N$ individuos que son sometidos nuevamente al operador `fast-non-dominated-sort` (línea 15), para seleccionar qué individuos pasan a la siguiente generación. Para esto, se define la nueva población como un conjunto vacío con capacidad máxima N individuos (línea 16). Posteriormente, se analiza cada frontera para determinar si el número total de individuos en la frontera, sumado a los individuos ya presentes en la nueva población, es menor o igual a la capacidad máxima de N individuos. Si esta condición se cumple, todos los individuos de esa frontera se añaden a la nueva población (líneas 18-19). Si no se cumple, se calcula la distancia de *crowding* para cada individuo y se seleccionan aquellos con mayor distancia hasta completar los N individuos de la nueva población, priorizando la diversidad en el espacio de soluciones, de acuerdo al operador `selección-ultima-frontera` (línea 22). Finalmente, se repiten los operadores previamente mencionados hasta que se exceda el tiempo límite y se termina por retornar los individuos que conforman la frontera de pareto.

3.3. Distancia de *crowding*

La distancia de *crowding* es una medida de que tan cerca está un individuo a sus vecinos en la frontera. Por lo tanto, se busca la maximización de esta distancia con el fin de de lograr abarcar una mayor porción de la frontera.

La [Ecuación \(13\)](#) presenta la formulación de la distancia de crowding, donde $d_j(F_i)$ representa la distancia de crowding del individuo j perteneciente a la frontera i y $O_k(j)$ es el valor del objetivo k para el individuo j . Además $\min_k(F_i)$ y $\max_k(F_i)$ representan el valor mínimo y máximo respectivamente del objetivo k en la frontera i .

$$d_j(F_i) = \frac{O_1(j+1) - O_1(j-1)}{\max_1(F_i) - \min_1(F_i)} + \frac{O_2(j+1) - O_2(j-1)}{\max_2(F_i) - \min_2(F_i)} \quad (13)$$

La distancia de Crowding utilizada fue propuesta por [Chu & Yu \(2018\)](#), quienes sugieren un cambio en los índices a la hora de considerar el valor de los objetivos, mejorando de esta manera la velocidad de convergencia de las soluciones. Esta formulación utilizada es planteada en la [Ecuación \(14\)](#).

$$d_j(F_i) = \frac{O_1(j+1) - O_1(j)}{\max_1(F_i) - \min_1(F_i)} + \frac{O_2(j+1) - O_2(j)}{\max_2(F_i) - \min_2(F_i)} \quad (14)$$

3.4. Representación de la solución

Cada individuo de la población P_i ($i = 1, \dots, N$) está compuesto de tres cromosomas (π_1, π_2, π_3). El primer cromosoma es el conjunto de *features*, con un tamaño de *features* variable y perteneciente al intervalo $[4, n]$. El segundo cromosoma indica los coeficientes que acompañen a las *features* que han sido seleccionadas por el primer cromosoma, representado la influencia de esa variable en la clasificación final. Al ser el segundo cromosoma los coeficientes de las *features* seleccionadas por el primero, estos tienen el mismo tamaño. Finalmente, el tercer cromosoma indica el intercepto del modelo (b), siendo un único valor. La [Figura 2](#) muestra la representación de un individuo que tiene n *features* seleccionadas. El vector de *features* contiene solo números enteros, los cuales son identificadores de las variables del conjunto de datos. El vector de pesos y el intercepto son valores reales.

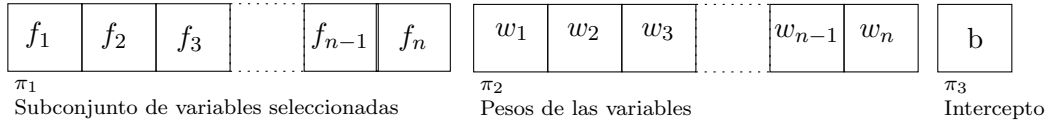


Figura 2: Representación del individuo.

La [Figura 3](#) muestra la representación del individuo representado en la [Figura 1](#). Muestra la selección de ambas *features* f_1 y f_2 , mientras sus pesos son 1 y -1, respectivamente. Además, se presenta el valor del intercepto de -1. Con estos datos entonces es posible construir la recta que divide el espacio solución como: $0 = f_1 - f_2 - 1$. Esta ecuación representa a la recta punteada de la [Figura 1](#), por lo que al cambiar el valor de 0 por un 1 o un -1, se obtienen los otros dos planos que dividen el espacio.

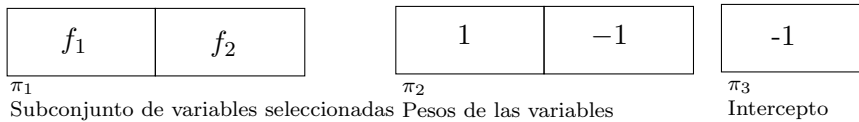


Figura 3: Representación del cromosoma del ejemplo de la [Figura 1](#).

3.5. Población inicial

La población inicial de N individuos es generada de forma aleatoria. La creación de cada individuo comienza con la definición del número de *features* a considerar, el cuál es obtenido de forma aleatoria desde un rango establecido. Así, para la construcción del primer cromosoma que corresponde al subconjunto de *features*, las n *features* son seleccionadas de manera aleatoria. Donde cada variable tiene la misma probabilidad de ser seleccionada. Respecto al segundo y tercer cromosoma, cada valor w y b es obtenido de una distribución aleatoria uniforme.

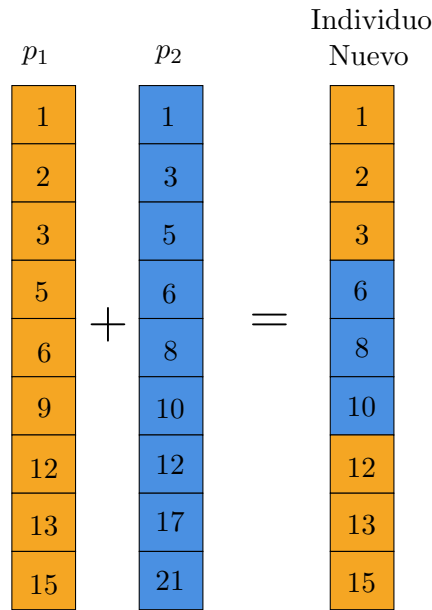


Figura 4: Ejemplo de aplicación del operador de PMX.

3.6. Torneo basado en la dominancia

Dado dos progenitores, se define un p_1 y un p_2 , para conservar las características de aquel con mejor rendimiento (p_1), se realiza un torneo basado en la dominancia. El ganador es p_1 y el perdedor es p_2 . p_1 es definido siguiendo los siguientes criterios, donde aquel que se cumpla primero, establece al ganador.

1. p_1 es aquel individuo que pertenezca a una frontera más baja.
2. p_1 es aquel individuo que domine a una mayor cantidad de soluciones.
3. p_1 es escogido de manera aleatoria.

3.7. Cruzamiento

El cruzamiento es un operador genético que combina la información de dos individuos (p_1 y p_2) para generar nuevas soluciones (hijos), promoviendo la exploración del espacio de búsqueda. Este proceso permite que las soluciones hereden características de ambos padres, facilitando la convergencia hacia un conjunto de soluciones no dominadas en problemas multiobjetivo.

El cruzamiento es una adaptación del operador de *partially mapped crossover* (PMX) (Goldberg, 1985). Este operador comienza con la cantidad de k subdivisiones en las que son divididos los cromosomas 1 (de *features*) y 2 (de pesos) del p_1 , donde al largo de cada subdivisión generada es L . Primero, se

considera el cromosoma de las *features*, ya que por factibilidad, las *features* al interior del cromosoma no pueden estar repetidas. Se seleccionan las primeras L *features* de p_1 a copiar en el cromosoma π_1 del nuevo individuo. Como el cromosoma de *features* es un vector ordenado de manera creciente, se busca en p_2 aquella primera *feature* que sea mayor que la última agregada al nuevo individuo. Luego, se copia al nuevo individuo las L nuevas *features* desde el punto mencionado. En caso que el p_2 no tenga L *features* nuevas que aportar, se copian todas las disponibles y se termina por copiar de p_1 las *features* restantes siguiendo el principio de no repetir *features*. Este proceso es repetido tantas veces como divisiones del cromosoma π_1 se hayan hecho, intercalando por cada iteración el progenitor del cuales son copiadas las *features*.

El cruzamiento del cromosoma π_1 se ilustra de manera resumida en la [Figura 4](#), con un valor $L = 3$ para dos progenitores de largo 9. Se observa cómo las *features* se seleccionan de manera intercalada entre ambos progenitores. Así, el primer tercio del nuevo cromosoma proviene de p_1 , el segundo tercio de p_2 , y el último tercio nuevamente de p_1 . Este proceso garantiza que no se repitan los elementos, resultando en un vector ordenado que integra *features* de ambos progenitores.

En el cruzamiento de π_2 , se consideran las *features* que componen el nuevo individuo. Por cada *feature* se obtiene el peso desde el respectivo progenitor que aporte esa *feature*. Finalmente, el cruzamiento del cromosoma π_3 , se obtiene del promedio de los interceptos de los progenitores.

3.8. Mutación

La mutación introduce diversidad en la población al alterar aleatoriamente los genes de los individuos, permitiendo explorar nuevas regiones del espacio de búsqueda. Este proceso ayuda a evitar la convergencia prematura al mejorar la capacidad del algoritmo para escapar de óptimos locales y contribuyendo a la diversidad genética de la población.

La mutación es realizada por dos operadores con enfoques diferentes. El primer operador trabaja solo sobre los valores que componen al individuo, mientras que el segundo, no admite cambios en el número de *features* que lo componen. A continuación, se detallan ambos.

- **Mutación-1:** Este operador puede cambiar el valor de alguno de los pesos, aumentando o disminuyendo en un valor de 5%. Mientras que el valor del intercepto puede aumentar o disminuir en un 10%, o cambiar alguna variable seleccionada por alguna que no lo esté. Además, cada variable no seleccionada tiene una misma probabilidad de ser seleccionada. Cada uno de estos cambios tiene un 33% de probabilidad de ocurrencia y no ocurren de manera excluyente, es decir, si bien el valor esperado de mutaciones es una, podrían ocurrir las tres de manera simultánea.

-
- **Mutación-2:** Este operador permite mayor variabilidad al individuo respecto al número de *features* que dispone, con la misma probabilidad de aumentar o disminuir el número de *features*. Para realizar este cambio, se obtiene, desde una distribución uniforme discreta un valor para el número de *features* a agregar o eliminar. Si se agregan o eliminan *features*, se pueden añadir o sacar desde una feature hasta un máximo del 20 % del número de *features* que actualmente posee el individuo. Para la eliminación, esta se aplica si el número de *features* consideradas es mayor que siete asegurando así la factibilidad del individuo.

Ambos operadores son aplicados de manera no excluyente, donde el primero tiene una probabilidad de ocurrencia inicial de $P_0^{m_1}$ y el segundo una probabilidad inicial de $P_0^{m_2}$. Posteriormente, ambas probabilidades disminuyen con el transcurso de las iteraciones de acuerdo a la [Ecuación \(15\)](#). B permite la disminución gradual de la probabilidad de mutación haciendo que la exploración provocada por la mutación sea cada vez más localizada a medida que ocurren las iteraciones (i), imitando la idea de la metaheurística simulated annealing ([Kirkpatrick et al., 1983](#)).

$$P_{i+1}^{m_j} = P_i^{m_j} \times B_j \quad ; \quad \forall j \in \{1, 2\} \quad (15)$$

Capítulo 4

Experimentos computacionales

Este capítulo proporciona: los experimentos computacionales realizados, los conjuntos de datos utilizados, las métricas evaluadas, la configuración del algoritmo y la forma de ejecutar los experimentos.

4.1. Descripción del conjunto de datos

Para realizar los experimentos computacionales, el modelo propuesto fue probado en diferentes conjuntos de datos de clasificación binaria con dimensiones y características diferentes utilizados con frecuencia en la literatura. A continuación, se describe brevemente cada uno.

- **Ionosphere:** Este conjunto de datos es utilizado para la clasificación de señales de radar medidas sobre la ionosfera. Cada registro representa una serie de mediciones de radar reflejadas en la ionosfera, con el objetivo de clasificar las señales como “buenas” o “malas”, ayudando a determinar la relevancia de los objetos en el radar ([Sigillito & Baker, 1989](#)).
- **Wisconsin Breast Cancer (WBC):** Enfocado en la clasificación de tumores mamarios, este conjunto de datos tiene como objetivo distinguir entre tumores benignos o malignos. Sus *features* son características extraídas de imágenes digitales de biopsias ([Wolberg, 1992](#)).
- **Arcene:** Diseñado para problemas de clasificación de alta dimensión en bioinformática, este conjunto de datos contiene perfiles de expresión génica para la clasificación de cáncer ([Guyon et al., 2008a](#)).
- **Gisette:** Conjunto de datos enfocado en la distinción de números escritos a mano, “4” y “9”. Las *features* incluidas contienen *features* distractoras, a forma de ser utilizados en problemas de feature selection ([Guyon et al., 2008b](#)).

-
- **German Credit:** Este conjunto de datos contiene información sobre solicitudes de crédito en Alemania, incluyendo diversas características de los solicitantes. El objetivo es predecir si un solicitante de crédito es apto para otorgarle un crédito basado en su historial y características personales (Hofmann, 1994).
 - **Bioresponse:** Tiene como objetivo predecir la respuesta biológica de moléculas a partir de sus propiedades químicas, estas respuestas son clasificadas como positivas o negativas (Abdunabi & Basir, 2014).
 - **Duke:** Enfocado en la detección de cáncer, este conjunto de datos incluye características de células que han sido estudiadas en biopsias (West et al., 2001).

Ionosphere, WBC, Arcene, Gisette y *German Credit* pueden ser encontrados en el repositorio UCI (Kelly et al., 2024). Bioresponse puede ser encontrado en la plataforma Kaggle (2024). Duke puede ser obtenido de LIBSVM (Chang & Lin, 2011). La Tabla 2 presenta las principales características de cada dataset en términos dimensionales, tamaño y *features*.

Estos conjuntos de datos han sido seleccionados por Valero-Carreras et al. (2023). Sin embargo, en esta investigación, solo han sido seleccionados aquellos que originalmente son de clasificación binaria, puesto que alguno de los conjuntos de datos presentes en esa investigación, son de clasificación multiclase o de regresión. Se descartaron los conjuntos de datos que no corresponden a problemas de clasificación binaria, dado que la naturaleza del SVM se enfoca en clasificación binaria. Aunque es posible adaptar el SVM para tareas de clasificación multiclase o regresión, la investigación de Valero-Carreras et al. (2023) no indica la estrategia empleada para estos casos. Además, en esta memoria se utiliza la totalidad del conjunto de datos para el análisis. No se emplean las divisiones predefinidas de entrenamiento y testeo, incluso cuando estas están presentes en los conjuntos de datos, con el fin de realizar una comparación justa con Valero-Carreras et al. (2023). Esto se debe a que dicho trabajo no detalla cómo se manejan los conjuntos de datos preprocesados, lo que genera una inconsistencia entre lo establecido en el estado del arte y el diseño de su experimento.

Además, el algoritmo propuesto estandariza los datos de cada instancia, eliminando la media (μ) y escalando según la varianza (σ) de cada feature. Así, la normalización es realizada según la Ecuación (16).

$$z_i = \frac{x_i - \mu}{\sigma} \quad \forall i = \{1, \dots, m\} \quad (16)$$

La Figura 5 muestra la distribución de las clases por cada conjunto de dato, permitiendo evidenciar

Tabla 2: Características de los conjuntos de datos.

Instancia	Vectores	<i>features</i>
Ionosphere	351	34
WBC	569	30
Arcene	900	10.000
Gisette	13.500	5.000
German Credit	1.000	20
Bioresponse	3.000	1.776
Duke	44	7.129

el desbalance presente en algunas de las instancias. El ejemplo más claro es el caso del conjunto de datos de German Credit que tiene a la clase 1 representada con el 70% de los vectores del conjunto de datos. Por tanto, se justifican la ponderación agregada en el objetivo 2.

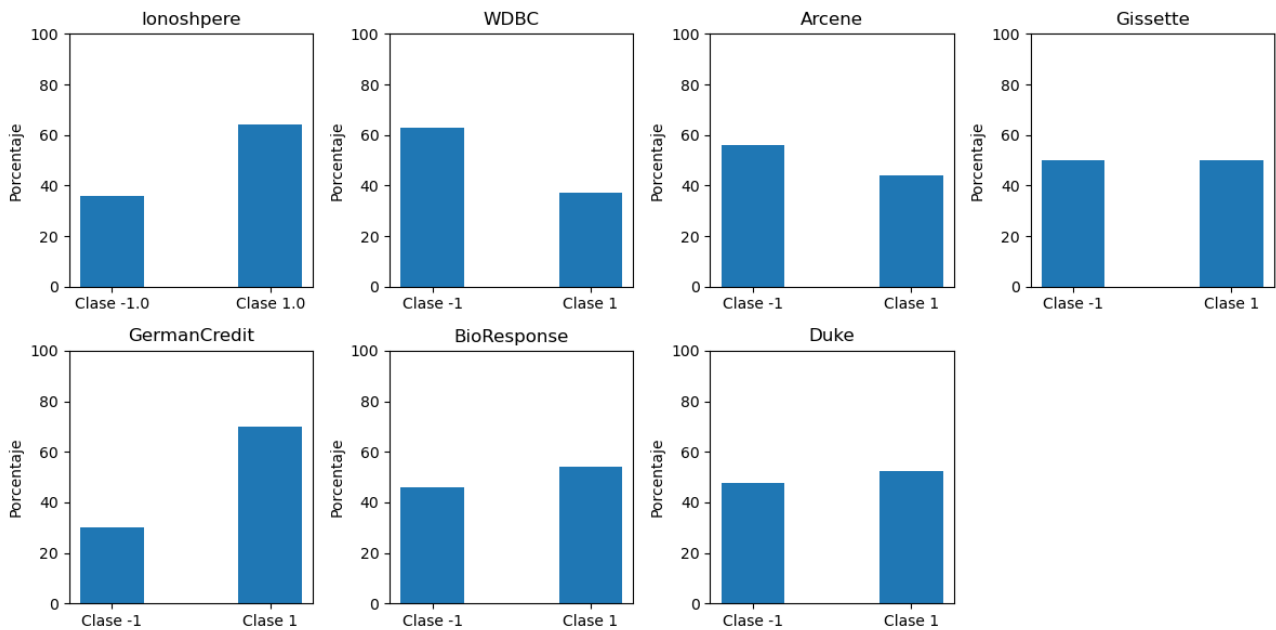


Figura 5: Distribución de las clases por instancia.

4.2. Métricas utilizadas

El algoritmo propuesto es evaluado con tres métricas usadas en Valero-Carreras et al. (2023). Con el fin de realizar una comparación del rendimiento de los algoritmos. Estas métricas provienen del análisis de la matriz de confusión, en la Figura 6 se presenta la matriz de confusión y cada métrica, con su forma de calcularla y su interpretación.

La matriz de confusión es una herramienta utilizada en la evaluación de modelos de clasificación, proporcionando una visión detallada del desempeño del modelo (Sandeep et al., 2023). Consiste en

	Valor real Clase 1	Valor real Clase -1
Predicción Clase 1	Verdaderos Positivos TP	Falsos Positivos FP
Predicción Clase -1	Falsos Negativos FN	Verdaderos Negativos TN

Figura 6: Matriz de confusión.

una tabla que compara las predicciones del modelo con los valores reales, mostrando el número de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. Esta matriz permite calcular las métricas utilizadas como el área bajo la curva ROC, la F1-Score y el coeficiente Kappa de Cohen.

- **Area bajo la curva ROC (AUC-ROC):** La curva ROC muestra la relación entre la tasa de verdaderos positivos (TPR) (Ecuación (17)) y la tasa de falsos positivos (FPR) (Ecuación (18)) del modelo en diferentes umbrales de decisión. La AUC-ROC representa la probabilidad que el modelo clasifique una instancia positiva con una puntuación de probabilidad más alta que una instancia negativa. Esta métrica se hace en función de las probabilidades de ser de una clase u otra, se ha utilizado una función sigmoide para efectos del cálculo de tal probabilidad. La AUC-ROC varía entre 0 y 1, donde valores más cercanos a 1 indican un modelo con mejor capacidad de discriminación entre clases.

$$TPR = \frac{TP}{TP + FN} \quad (17)$$

$$FPR = \frac{FP}{FP + TN} \quad (18)$$

- **F1-Score (FSC):** Se calcula como la media armónica entre **Precision** ($\frac{TP}{TP+FP}$) y **Recall** ($\frac{TP}{TP+FN}$), proporcionando un equilibrio entre la exactitud y la capacidad del modelo para encontrar todas las instancias positivas. La **F1-score** oscila entre 0 y 1, donde a mayor cercanía a 1 es considerado un mejor modelo. La fórmula de F1-Score es presentada en la Ecuación (19).

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (19)$$

-
- **Coefficiente Kappa de Cohen (CK):** Es una medida estadística que evalúa la concordancia entre dos clasificaciones categóricas, teniendo en cuenta el acuerdo que se esperaría por azar. Su valor varía de -1 (discordancia total) a 1 (concordancia perfecta), con 0 indicando acuerdo equivalente al azar. El cálculo de este coeficiente es presentado en la [Ecuación \(22\)](#).

$$p_o = \frac{TP + TN}{TP + TN + FP + FN} \quad (20)$$

$$p_c = \frac{(TP + FP) \cdot (TP + FN) + (TN + FN) \cdot (TN + FP)}{(TP + TN + FP + FN)^2} \quad (21)$$

$$\kappa = \frac{p_o - p_c}{1 - p_c} \quad (22)$$

4.3. Configuración del algoritmo

El algoritmo contiene nueve parámetros que son presentados en la [Tabla 3](#). Los valores de estos parámetros son seleccionados desde la revisión de literatura ([Alcaraz et al., 2022](#)) y también, mediante la técnica de búsqueda de parámetros como “grid search” ([Myung et al., 2013](#)) en experimentos preliminares.

Tabla 3: Parámetros utilizados

Parámetro	Descripción	Valor
N	Número de individuos de la población inicial	50
t	Tiempo de ejecución	{1200,3600}
s	Semillas ejecutadas	{3,5,7}
f	Rango de número de <i>features</i> iniciales	U(0,8;1,0)
w	Rango para inicialización de pesos	U(-3;3)
b	Rango para inicialización del intercepto	U(-5;5)
k	Divisiones del cromosoma en el cruzamiento	3
$P_0^{m_1}$	Probabilidad inicial mutador 1	0,6
$P_0^{m_2}$	Probabilidad inicial mutador 2	0,3

4.4. Configuración de los experimentos

Los experimentos han sido ejecutados en un clúster perteneciente a la Facultad de Ingeniería llamado Luthier. Luthier está compuesto por 30 nodos de cómputo, donde todos los nodos tienen la misma configuración de hardware y software. Todos los nodos tienen CPUs Intel Xeon E3-1270 v6 a 3.8GHz

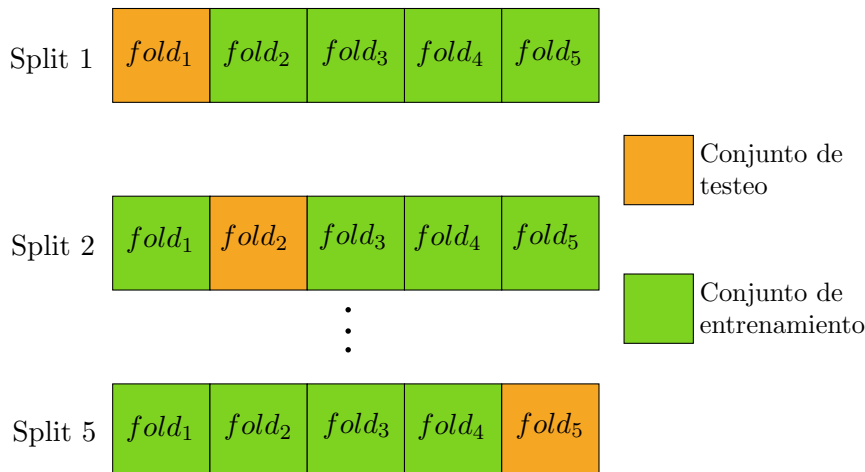


Figura 7: Ejemplo K-fold.

con 64GB de RAM. El lenguaje utilizado fue Python en contraste con Valero-Carreras et al. (2023) quienes han utilizado C++.

El diseño de los experimentos es replicado desde el trabajo de Valero-Carreras et al. (2023), donde cada una de las siete instancias es ejecutada con dos diferentes tiempos de ejecución, 1200 y 3600 segundos, con tres semillas diferentes. Además, se consideran cinco divisiones diferentes de conjuntos de entrenamiento y testeo utilizando la técnica de validación cruzada K-fold (Raschka, 2018). K-fold consiste en dividir el conjunto total de datos en subconjuntos de igual dimensión, siguiendo a Valero-Carreras et al. (2023), se han utilizando cinco subconjuntos, de manera que se selecciona un subconjunto para ser utilizado en la fase de testeo y el resto en la fase de entrenamiento. Este proceso se ha repetido cinco veces, haciendo que cada subconjunto sea destinado una vez para realizar testeo. Este proceso representado en la Figura 7, donde por cada iteración (split), un subconjunto diferente se usa como conjunto de testeo (en naranja), y los demás como conjunto de entrenamiento (en verde). De esta manera, la ejecución de un experimento, genera un total de 240 combinaciones.

Para resumir los resultados de todos los experimentos, se considera que para cada combinación de tiempo e instancia se realizan 15 ejecuciones, dadas las cinco subdivisiones generadas con K-fold y las tres semillas utilizadas. En cada una de estas ejecuciones es seleccionado el mejor individuo según cada métrica a reportar. Por tanto, para cada una de las cinco subdivisiones se obtienen tres valores, para los cuales se calculan el máximo y el promedio. Con estos cinco máximos y cinco promedios, se pueden definir dos perspectivas para cada métrica. La primera perspectiva, es el promedio de los promedios, y se reporta con el nombre simple de la métrica evaluada (AUC, FSC, CK). Además, se calcula el promedio de los cinco máximos, que se reportan con un asterisco (AUC*, FSC*, CK*).

Capítulo 5

Resultados y discusión

Este capítulo presenta los resultados obtenidos y una discusión sobre estos. La [Tabla 4](#) y la [Tabla 5](#) presentan la comparación de los resultados obtenidos en esta memoria con los resultados de la investigación de [Valero-Carreras et al. \(2023\)](#). En ambas se presentan todas las instancias y sus respectivos tiempos. Sin embargo, la [Tabla 4](#) presenta la perspectiva simple de presentación de métricas (sin asterisco) y la [Tabla 5](#) presenta la perspectiva de los promedios máximos (con asterisco). El detalle de los resultados obtenidos en esta memoria son presentados en el [Apéndice A](#).

Tanto el AUC-ROC y la FSC presentan un mejor rendimiento global que los resultados presentados por [Valero-Carreras et al. \(2023\)](#). Se observa en los promedios finales que el AUC-ROC y AUC-ROC* obtienen valores cercanos a 0,8, mostrando una gran capacidad predictiva general. A su vez, son respaldados por los valores de FSC y FSC*, donde estos muestran una oportunidad de mejora en términos de precisión. Esta mejora respecto al trabajo de [Valero-Carreras et al. \(2023\)](#) en las métricas mencionadas puede ser debido a tres factores:

1. **Representación de la solución propuesta:** La representación del individuo propuesta en esta memoria (mediante la generación de los pesos) implica una representación directa de los hiperplanos que dividen el espacio solución. En contraste a [Valero-Carreras et al. \(2023\)](#) que selecciona vectores del conjunto de datos y luego calcula el hiperplano que lo intersecta siguiendo la idea presentada en [Alcaraz et al. \(2022\)](#). Esta diferencia implica para el algoritmo propuesto un menor tiempo de computo por individuo, que se traduce en una mayor cantidad de iteraciones y en consecuencia, una búsqueda más amplia en el espacio solución. Incluso considerando la ventaja de [Valero-Carreras et al. \(2023\)](#) al utilizar C++ respecto a Python, lenguaje que puede ser hasta 100 veces más lento que C++ dependiendo de la tarea ([Zehra et al., 2020](#)).
2. **Número de *features* dinámico:** El algoritmo propuesto en esta memoria de título es capaz de

Tabla 4: Comparación de resultados en los promedios de los promedios de las métricas.

		Algoritmo propuesto			Valero-Carreras et al. (2023)		
Conjunto de datos	Tiempos	AUC	FSC	CK	AUC	FSC	CK
Arcene	1200	0,728	0,440	0,215	0,717	0,676	0,435
	3600	0,728	0,440	0,215	0,724	0,683	0,455
BioResponse	1200	0,625	0,661	0,073	0,586	0,720	0,175
	3600	0,625	0,661	0,073	0,571	0,719	0,144
Duke	1200	0,814	0,704	0,341	0,617	0,623	0,236
	3600	0,814	0,704	0,341	0,555	0,558	0,112
GermanCredit	1200	0,514	0,823	0,000	0,706	0,593	0,351
	3600	0,514	0,823	0,000	0,705	0,591	0,360
Gisette	1200	0,788	0,673	0,358	0,684	0,758	0,368
	3600	0,797	0,684	0,368	0,711	0,766	0,423
Ionoshpere	1200	0,767	0,843	0,415	0,858	0,915	0,742
	3600	0,767	0,843	0,415	0,863	0,919	0,753
WDBC	1200	0,980	0,715	0,619	0,971	0,964	0,943
	3600	0,980	0,715	0,619	0,975	0,969	0,950
Promedio		0,746	0,695	0,290	0,732	0,747	0,461

Tabla 5: Comparación de resultados en los promedios de los cinco máximos.

		Algoritmo propuesto			Valero-Carreras et al. (2023)		
Conjunto de datos	Tiempos	AUC*	FSC*	CK*	AUC*	FSC*	CK*
Arcene	1200	0,805	0,573	0,306	0,742	0,720	0,478
	3600	0,805	0,573	0,306	0,751	0,723	0,505
BioResponse	1200	0,706	0,709	0,140	0,609	0,725	0,221
	3600	0,706	0,709	0,140	0,589	0,724	0,179
Duke	1200	1,000	0,872	0,648	0,627	0,627	0,255
	3600	1,000	0,872	0,648	0,629	0,623	0,257
GermanCredit	1200	0,570	0,844	0,000	0,711	0,597	0,373
	3600	0,570	0,844	0,000	0,712	0,599	0,375
Gisette	1200	0,852	0,796	0,521	0,699	0,769	0,399
	3600	0,864	0,790	0,523	0,773	0,797	0,545
Ionoshpere	1200	0,868	0,902	0,630	0,858	0,916	0,744
	3600	0,868	0,902	0,630	0,866	0,923	0,762
WDBC	1200	0,985	0,834	0,747	0,973	0,967	0,947
	3600	0,985	0,834	0,747	0,975	0,969	0,950
Promedio		0,828	0,790	0,428	0,751	0,763	0,499

ir cambiando el número de *features* consideradas de manera no supervisada, buscando converger al número óptimo de estas, en contraste a Valero-Carreras et al. (2023) quienes consideran un número estático de cinco *features*. Esto es representado en el cambio de la Ecuación (11) al considerar p como una variable y no un parámetro. Esta implementación permite al algoritmo recorrer mejor el espacio de búsqueda, ayudándole así a encontrar aquellas *features* que mayor información le entreguen para realizar predicciones.

3. **Penalización de desbalance en el conjunto de datos** La función objetivo al incluir un valor α , se encarga permite ponderar la importancia de cada vector mal clasificado según la presencia de su clase respecto al total de vectores del conjunto de datos. Esto permite que los individuos que componen la población tiendan a intentar predecir de mejor manera la clase predominante del conjunto de datos, esto es más notorio a mayor desbalance de este. Como consecuencia, la proporción de valores bien clasificados es mayor, mejorando las métricas presentadas como el AUC-ROC y FSC.

Respecto al coeficiente Kappa de Cohen, se observa una menor concentración de valores superiores en comparación con Valero-Carreras et al. (2023). Esta métrica ha sido incluida solo para facilitar la comparación de ambos algoritmos. Esta métrica ha recibido críticas, como la hecha por Feinstein & Cicchetti (1990), sobre su susceptibilidad frente a desbalances en los totales marginales de la matriz de confusión, generando valores bajos de Kappa a pesar de una alta concordancia real. La Ecuación (22) presenta el cálculo del coeficiente en función de los valores de la matriz de confusión. Se observa la susceptibilidad al desbalance al notar que ambas probabilidades en el numerador de p_c son el producto entre las probabilidades marginales de aciertos y errores. Esto trae como consecuencia que, frente a una buena capacidad predictiva (validada por otras métricas como la AUC-ROC o la FSC), esta métrica siga siendo muy estricta. Esto se ejemplifica en la Figura 8, donde en un dataset desbalanceado con una proporción de 80 % de valores positivos podemos notar la discrepancia de CK con la métrica FSC. Además, cabe mencionar que el hecho de incluir una perspectiva balanceada del error (ponderación α de la función objetivo) puede traer penalizaciones más fuertes por parte de este coeficiente, una situación también aplicada en el ejemplo mencionado mediante la clasificación más precisa de la clase dominante.

Respecto a los resultados obtenidos por parte del algoritmo propuesto para el FS-SVM, se puede notar en la Tabla 6 el promedio del número de *features* seleccionadas por los individuos que han sido considerados para las métricas previamente presentadas. Estos tienen valores muy similares entre las distintas instancias, además de ser pocas *features* considerando las dimensiones originales de algunas

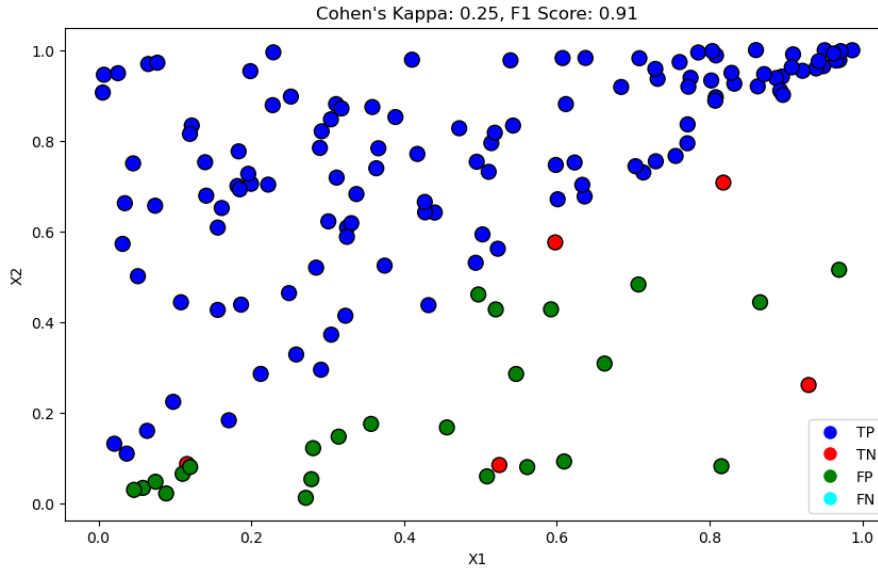


Figura 8: Ejemplo discordancia Cohen Kappa y FSC.

Tabla 6: Media de *features* consideradas por los mejores individuos

Conjunto de datos	<i>features</i> reales	Promedio de <i>features</i> consideradas
Ionosphere	34	5,0
WBC	30	4,7
Arcene	10.000	4,5
Gisette	5.000	4,0
GermanCredit	20	3,3
BioResponse	1.776	3,7
Duke	7.129	4,2

instancias. Esto puede ser por la formulación de SVM, donde al definir la Ecuación (2), el término $w^T x_i$ es equivalente a $\sum_{j=1}^m w_j x_{ij}$. Es fácil notar que la suma se hace sobre la cantidad de *features* (m). Por tanto, al realizar una clasificación errónea, el valor de ξ es mayor mientras más *features* hayan sido seleccionadas, esto siendo un incentivo al modelo para disminuir *features*. Esto puede ser analizado en futuras investigaciones añadiendo un factor regularizador a la restricción mencionada, de manera que los valores en el error de clasificación sean comparables entre individuos con distinta cantidad de *features*.

Finalmente, para analizar la convergencia del algoritmo a lo largo de las iteraciones, se considera el uso de la función de pérdida de entropía cruzada binaria (Bishop, 2007). Este permite evidenciar el rendimiento predictivo del modelo a medida que el algoritmo itera, mostrando de una forma sencilla su aprendizaje, en consideración de que es un problema multiobjetivo. La Ecuación (23) muestra la fórmula para la entropía cruzada binaria, donde para un total de N mediciones (para este caso N son

los individuos en la frontera de pareto), y_i es el valor verdadero de la etiqueta y $p(y_i)$ es la probabilidad que el modelo proporciona para esa observación un valor de 1. De manera que la entropía cruzada binaria penaliza las probabilidades altas cuando son erróneas, y así evidencia el progreso de la certeza del algoritmo y su capacidad predictiva.

$$L_{BCE} = -\frac{1}{N} \sum_{i=1}^N y_i \log(p(y_i)) + (1 - y_i) \log(1 - p(y_i)) \quad (23)$$

La [Figura 9](#) muestra las gráficas de las funciones de pérdida para cada instancia evaluada. En cada gráfica, se ha considerado como límite de iteraciones a el punto donde el algoritmo converge. En todas las instancias se observa que la tendencia de la función de pérdida es continuamente decreciente, mostrando así la mejora del algoritmo y validando las métricas presentadas. Para todas las instancias está convergencia fue alcanzada en un tiempo menor a 20 minutos como se observa en el segundo eje que muestra el tiempo.

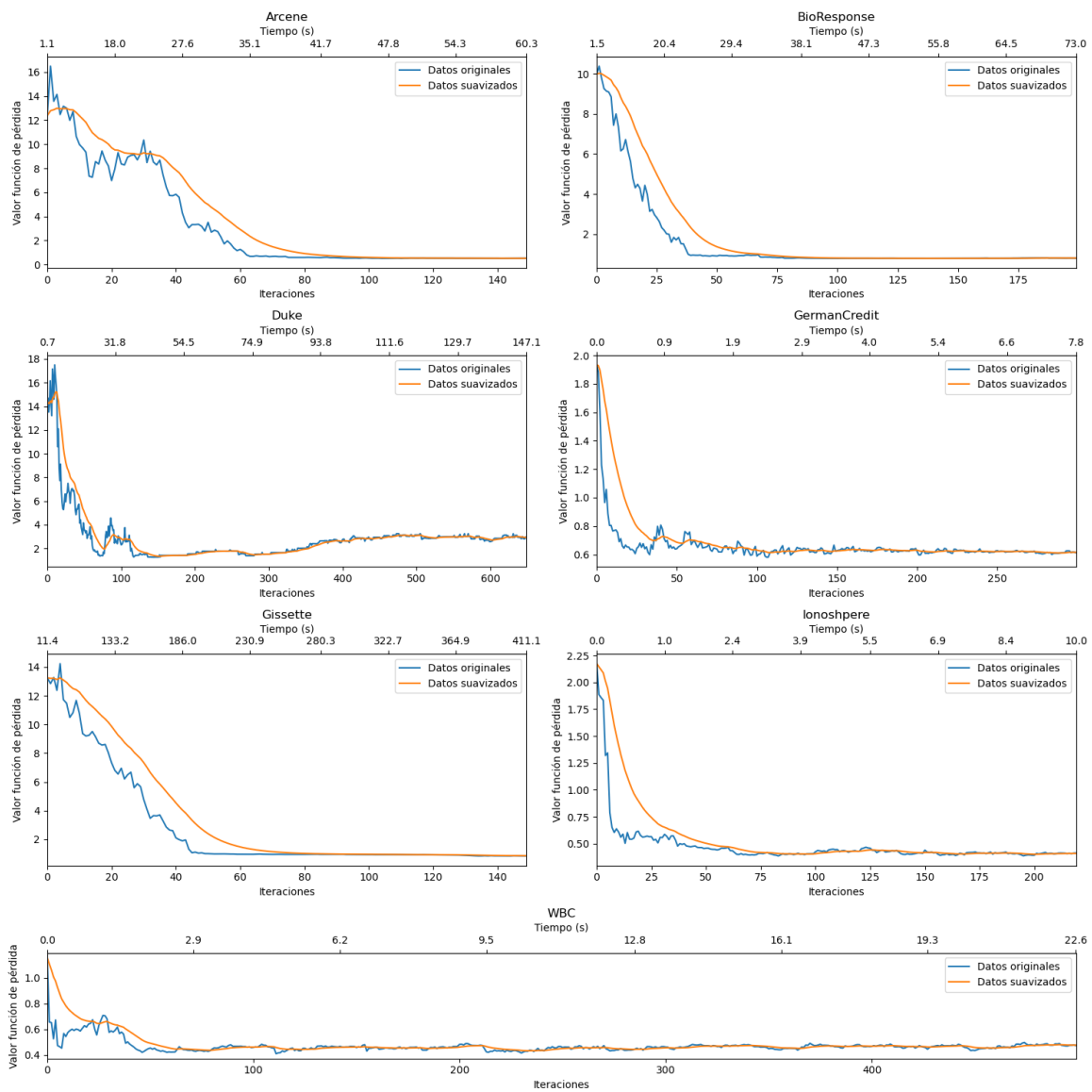


Figura 9: Funciones de pérdida por instancia.

Capítulo 6

Conclusión

Esta memoria de título ha abordado la problemática del FS-SVM mediante una metaheurística multi-objetivo, el NSGA-II. Se ha adaptado el modelo de optimización planteado en el estado del arte para FS-SVM, agregando libertad para escoger la cantidad de *features* consideradas. Además, se realizó una optimización ponderada por desbalance en el conjunto de datos. El algoritmo propuesto utiliza una nueva forma de representar a los individuos, y operadores de cruzamiento y mutación especializados para esta representación, manteniendo los conceptos de dominancia.

El algoritmo propuesto ha sido probado con conjuntos de datos conocidos en el campo del aprendizaje de máquina. Los resultados comparados con el estado del arte, demuestran que los cambios propuestos han entregado una ventaja tanto en términos temporales de ejecución como en términos predictivos. En futuras investigaciones se podría considerar agregar la normalización del número de *features* en las restricciones del cálculo del error para evitar el sesgo en menos *features*. Además, considerar mecanismos de exploración local al llegar a un punto de convergencia. Adicionalmente, se podrían agregar métricas que puedan aportar más información sobre el rendimiento del modelo como el *accuracy*, o la entropía cruzada a modo de poder facilitar las comparaciones entre las diversas investigaciones (Hosin & M.N, 2015). Finalmente, analizar el comportamiento del algoritmo al agregar alguna función de pérdida o métrica ya mencionada como un tercer objetivo a modo de intentar mejorar su rendimiento mediante la optimización directa de los resultados a reportar.

Referencias Bibliográficas

- Abdunabi, T., & Basir, O. (2014). 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). doi: [10.1109/BigData.2014.7004386](https://doi.org/10.1109/BigData.2014.7004386)
- Alcaraz, J., Labbé, M., & Landete, M. (2022). Support vector machine with feature selection: A multi-objective approach. *Expert Systems with Applications*, *204*, 117485. doi: [10.1016/j.eswa.2022.117485](https://doi.org/10.1016/j.eswa.2022.117485)
- Aytug, H. (2015). Feature selection for support vector machines using generalized benders decomposition. *European Journal of Operational Research*, *244*(1), 210–218. doi: [10.1016/j.ejor.2015.01.006](https://doi.org/10.1016/j.ejor.2015.01.006)
- Baldomero-Naranjo, M., Martínez-Merino, L. I., & Rodríguez-Chía, A. M. (2021). A robust svm-based approach with feature selection and outliers detection for classification problems. *Expert Systems with Applications*, *178*, 115017. doi: [10.1016/j.eswa.2021.115017](https://doi.org/10.1016/j.eswa.2021.115017)
- 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. doi: <https://doi.org/10.1016/j.cor.2018.03.005>
- Bishop, C. M. (2007). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1 ed.
- Bomze, I., D’Onofrio, F., Palagi, L., & Peng, B. (2024). Feature selection in linear svms via hard cardinality constraint: a scalable sdp decomposition approach. doi: [10.48550/arXiv.2404.10099](https://doi.org/10.48550/arXiv.2404.10099)
- 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*(2), 261–281. doi: [10.1007/s10462-017-9543-9](https://doi.org/10.1007/s10462-017-9543-9)

-
- Candelieri, A., Giordani, I., Archetti, F., Barkalov, K., Meyerov, I., Polovinkin, A., Sysoyev, A., & Zolotykh, N. (2019). Tuning hyperparameters of a svm-based water demand forecasting system through parallel global optimization. *Computers Operations Research*, *106*, 202–209. doi: <https://doi.org/10.1016/j.cor.2018.01.013>
- Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, *408*, 189–215. doi: [10.1016/j.neucom.2019.10.118](https://doi.org/10.1016/j.neucom.2019.10.118)
- Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*, 27:1–27:27. URL: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Chu, X., & Yu, X. (2018). Improved crowding distance for NSGA-II. *CoRR*, *abs/1811.12667*. doi: [10.48550/arXiv.1811.12667](https://arxiv.org/abs/1811.12667)
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. doi: [10.1023/A:1022627411411](https://doi.org/10.1023/A:1022627411411)
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197. doi: [10.1109/4235.996017](https://doi.org/10.1109/4235.996017)
- Dudzik, W., Nalepa, J., & Kawulok, M. (2021). Evolving data-adaptive support vector machines for binary classification. *Knowledge-Based Systems*, *227*, 107221. doi: <https://doi.org/10.1016/j.knosys.2021.107221>
- Eiben, A., & Smith, J. (2015). *Introduction to Evolutionary Computing*. Natural Computing Series. Springer. Geburtenis: 2nd edition. doi: [10.1007/978-3-662-44874-8](https://doi.org/10.1007/978-3-662-44874-8)
- 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*(8), 2355–2369. doi: [10.1007/s00521-016-2818-2](https://doi.org/10.1007/s00521-016-2818-2)
- Feinstein, A. R., & Cicchetti, D. V. (1990). High agreement but low kappa: I. the problems of two paradoxes. *Journal of Clinical Epidemiology*, *43*(6), 543–549. doi: [10.1016/0895-4356\(90\)90158-L](https://doi.org/10.1016/0895-4356(90)90158-L)
- Gaudioso, M., Gorgone, E., Labbé, M., & Rodríguez-Chía, A. (2017). Lagrangian relaxation for svm feature selection. *Computers Operations Research*, *87*, 137–145. doi: [10.1016/j.cor.2017.06.001](https://doi.org/10.1016/j.cor.2017.06.001)
-

-
- Goldberg, D. E. (1985). Alleles, loci and the traveling salesman problem. In *Proceedings of the 1st International Conference on Genetic Algorithms and Their Applications*.
- Guyon, I., Gunn, S., Ben-Hur, A., & Dror, G. (2008a). Arcene. UCI Machine Learning Repository. doi: [10.24432/C58P55](https://doi.org/10.24432/C58P55)
- Guyon, I., Gunn, S., Ben-Hur, A., & Dror, G. (2008b). Gisette. UCI Machine Learning Repository. doi: [10.24432/C5HP5B](https://doi.org/10.24432/C5HP5B)
- Hofmann, H. (1994). Statlog (German Credit Data). UCI Machine Learning Repository. doi: [10.24432/C5NC77](https://doi.org/10.24432/C5NC77)
- Hossin, M., & M.N, S. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining Knowledge Management Process*, 5, 01–11. doi: [10.5121/ijdkp.2015.5201](https://doi.org/10.5121/ijdkp.2015.5201)
- Kelly, M., Longjohn, R., & Nottingham, K. (2024). The UCI Machine Learning Repository. URL: <https://archive.ics.uci.edu>. Accedido: 2024-06.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671–680. doi: [10.1126/science.220.4598.671](https://doi.org/10.1126/science.220.4598.671)
- 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. doi: [10.1016/j.dam.2018.10.025](https://doi.org/10.1016/j.dam.2018.10.025)
- Malczewski, J., & Rinner, C. (2015). *Multiobjective Optimization Methods*, (pp. 123–143). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: [10.1007/978-3-540-74757-4_5](https://doi.org/10.1007/978-3-540-74757-4_5)
- 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. doi: [10.1016/j.ins.2014.03.110](https://doi.org/10.1016/j.ins.2014.03.110)
- Mavrotas, G., & Florios, K. (2013). An improved version of the augmented -constraint method (augmecon2) for finding the exact pareto set in multi-objective integer programming problems. *Applied Mathematics and Computation*, 219(18), 9652–9669. doi: [10.1016/j.amc.2013.03.002](https://doi.org/10.1016/j.amc.2013.03.002)
- Miao, J., & Niu, L. (2016). A survey on feature selection. *Procedia Computer Science*, 91, 919–926. Promoting Business Analytics and Quantitative Management of Technology: 4th Interna-
-

-
- tional Conference on Information Technology and Quantitative Management (ITQM 2016). doi: [10.1016/j.procs.2016.07.111](https://doi.org/10.1016/j.procs.2016.07.111)
- Myung, J. I., Cavagnaro, D. R., & Pitt, M. A. (2013). A tutorial on adaptive design optimization. *Journal of Mathematical Psychology*, *57*(3), 53–67. doi: [10.1016/j.jmp.2013.05.005](https://doi.org/10.1016/j.jmp.2013.05.005)
- Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. *CoRR*, *abs/1811.12808*. doi: [10.48550/arXiv.1811.12808](https://doi.org/10.48550/arXiv.1811.12808)
- Sandeep, M., Tiprak, K., Kaewunruen, S., Pheinsusom, P., & Pansuk, W. (2023). Shear strength prediction of reinforced concrete beams using machine learning. *Structures*, *47*, 1196–1211. doi: [10.1016/j.istruc.2022.11.140](https://doi.org/10.1016/j.istruc.2022.11.140)
- Sigillito, W.-S. H. L., V., & Baker, K. (1989). Ionosphere. UCI Machine Learning Repository. doi: [10.24432/C5W01B](https://doi.org/10.24432/C5W01B)
- Valero-Carreras, D., Alcaraz, J., & Landete, M. (2023). Comparing two svm models through different metrics based on the confusion matrix. *Computers and Operations Research*, *152*, 106131. doi: [10.1016/j.cor.2022.106131](https://doi.org/10.1016/j.cor.2022.106131)
- Wang, J., Wang, X., Li, X., & Yi, J. (2023). A hybrid particle swarm optimization algorithm with dynamic adjustment of inertia weight based on a new feature selection method to optimize svm parameters. *Entropy*, *25*(3). doi: [10.3390/e25030531](https://doi.org/10.3390/e25030531)
- West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Olson, J. A., Marks, J. R., & Nevins, J. R. (2001). Predicting the clinical status of human breast cancer by using gene expression profiles. *Proceedings of the National Academy of Sciences*, *98*(20), 11462–11467. doi: [10.1073/pnas.201162998](https://doi.org/10.1073/pnas.201162998)
- Wolberg, W. (1992). Breast Cancer Wisconsin (Original). UCI Machine Learning Repository. doi: [10.24432/C5HP4Z](https://doi.org/10.24432/C5HP4Z)
- Yang, X., Song, Q., & Cao, A. (2005). Weighted support vector machine for data classification. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2, (pp. 859–864 vol. 2). doi: [10.1109/IJCNN.2005.1555965](https://doi.org/10.1109/IJCNN.2005.1555965)
- Zandvakili, A., Javidi, M. M., & Mansouri, N. (2024). Simultaneous feature selection and svm optimization based on fuzzy signature and chaos goa. *Evolving Systems*, *15*(5), 1907–1937. doi: [10.1007/s12530-024-09595-4](https://doi.org/10.1007/s12530-024-09595-4)
-

Zehra, F., Javed, M., Khan, D., & Pasha, M. (2020). Comparative analysis of c++ and python in terms of memory and time. *Preprints*. doi: [10.20944/preprints202012.0516.v1](https://doi.org/10.20944/preprints202012.0516.v1)

Anexo A. Tablas de resultados de instancias

Tabla 7: Resultados Arcene

Tiempo	Fold	Semilla	AUC	FSC	CK	<i>Features</i>
1200	0	3	0,724	0,629	0,340	6
1200	1	3	0,895	0,100	0,055	6
1200	2	3	0,597	0,486	0,073	6
1200	3	3	0,755	0,611	0,320	6
1200	4	3	0,824	0,667	0,430	4
1200	0	5	0,571	0,160	0,032	4
1200	1	5	0,523	0,207	-0,237	3
1200	2	5	0,762	0,300	0,167	3
1200	3	5	0,811	0,560	0,371	4
1200	4	5	0,622	0,000	0,000	3
1200	0	7	0,637	0,000	0,000	4
1200	1	7	0,760	0,773	0,515	6
1200	2	7	0,716	0,739	0,438	4
1200	3	7	0,717	0,683	0,353	4
1200	4	7	0,762	0,000	0,000	5
3600	0	3	0,724	0,629	0,340	6
3600	1	3	0,895	0,100	0,055	6
3600	2	3	0,597	0,486	0,073	6
3600	3	3	0,755	0,611	0,320	6
3600	4	3	0,824	0,667	0,430	4
3600	0	5	0,571	0,160	0,032	4
3600	1	5	0,523	0,207	-0,237	3
3600	2	5	0,762	0,300	0,167	3
3600	3	5	0,811	0,560	0,371	4
3600	4	5	0,622	0,000	0,000	3
3600	0	7	0,637	0,000	0,000	4
3600	1	7	0,760	0,773	0,515	6
3600	2	7	0,716	0,739	0,438	4
3600	3	7	0,717	0,683	0,353	4
3600	4	7	0,762	0,000	0,000	5

Tabla 8: Resultados BioResponse

Tiempo	Fold	Semilla	AUC	FSC	CK	<i>Features</i>
1200	0	3	0,748	0,723	0,214	3
1200	1	3	0,571	0,699	-0,002	3
1200	2	3	0,670	0,719	0,016	6
1200	3	3	0,622	0,601	0,215	3
1200	4	3	0,585	0,694	0,000	3
1200	0	5	0,765	0,696	0,000	4
1200	1	5	0,631	0,683	0,064	3
1200	2	5	0,718	0,721	0,001	5
1200	3	5	0,620	0,361	0,163	4
1200	4	5	0,620	0,596	0,168	3
1200	0	7	0,601	0,529	0,128	4
1200	1	7	0,554	0,714	0,000	3
1200	2	7	0,535	0,695	0,019	3
1200	3	7	0,669	0,611	0,233	5
1200	4	7	0,510	0,706	0,000	3
3600	0	3	0,748	0,723	0,214	3
3600	1	3	0,571	0,699	-0,002	3
3600	2	3	0,670	0,719	0,016	6
3600	3	3	0,622	0,601	0,215	3
3600	4	3	0,585	0,694	0,000	3
3600	0	5	0,765	0,696	0,000	4
3600	1	5	0,631	0,683	0,064	3
3600	2	5	0,718	0,721	0,001	5
3600	3	5	0,620	0,361	0,163	4
3600	4	5	0,620	0,596	0,168	3
3600	0	7	0,601	0,529	0,128	4
3600	1	7	0,554	0,714	0,000	3
3600	2	7	0,535	0,695	0,019	3
3600	3	7	0,669	0,611	0,233	5
3600	4	7	0,510	0,706	0,000	3

Tabla 9: Resultados Duke

Tiempo	Fold	Semilla	AUC	FSC	CK	<i>Features</i>
1200	0	3	0,733	0,667	0,467	5
1200	1	3	0,583	0,600	0,000	3
1200	2	3	0,625	0,500	0,000	4
1200	3	3	0,867	0,857	0,750	4
1200	4	3	0,600	0,667	0,351	5
1200	0	5	1,000	0,400	0,250	4
1200	1	5	1,000	0,667	0,333	3
1200	2	5	1,000	0,800	0,500	4
1200	3	5	0,500	0,667	0,333	6
1200	4	5	0,943	0,778	0,226	3
1200	0	7	1,000	0,923	0,600	5
1200	1	7	0,667	0,667	0,333	5
1200	2	7	0,933	0,889	0,750	3
1200	3	7	0,533	0,833	0,385	6
1200	4	7	0,594	0,571	0,400	3
3600	0	3	0,733	0,667	0,467	5
3600	1	3	0,583	0,600	0,000	3
3600	2	3	0,625	0,500	0,000	4
3600	3	3	0,867	0,857	0,750	4
3600	4	3	0,600	0,667	0,351	5
3600	0	5	1,000	0,400	0,250	4
3600	1	5	1,000	0,667	0,333	3
3600	2	5	1,000	0,800	0,500	4
3600	3	5	0,500	0,667	0,333	6
3600	4	5	0,943	0,778	0,226	3
3600	0	7	1,000	0,923	0,600	5
3600	1	7	0,667	0,667	0,333	5
3600	2	7	0,933	0,889	0,750	3
3600	3	7	0,533	0,833	0,385	6
3600	4	7	0,594	0,571	0,400	3

Tabla 10: Resultados GermanCredit

Tiempo	Fold	Semilla	AUC	FSC	CK	<i>Features</i>
1200	0	3	0,558	0,820	0,000	4
1200	1	3	0,623	0,841	0,000	3
1200	2	3	0,528	0,854	0,000	3
1200	3	3	0,521	0,830	0,000	3
1200	4	3	0,511	0,769	0,000	3
1200	0	5	0,509	0,837	0,000	4
1200	1	5	0,468	0,860	0,000	3
1200	2	5	0,501	0,795	0,000	3
1200	3	5	0,494	0,813	0,000	3
1200	4	5	0,598	0,810	0,000	4
1200	0	7	0,496	0,854	0,000	3
1200	1	7	0,546	0,820	0,000	3
1200	2	7	0,556	0,817	0,000	4
1200	3	7	0,536	0,802	0,000	3
1200	4	7	0,602	0,824	0,000	3
3600	0	3	0,558	0,820	0,000	4
3600	1	3	0,623	0,841	0,000	3
3600	2	3	0,528	0,854	0,000	3
3600	3	3	0,521	0,830	0,000	3
3600	4	3	0,511	0,769	0,000	3
3600	0	5	0,509	0,837	0,000	4
3600	1	5	0,468	0,860	0,000	3
3600	2	5	0,501	0,795	0,000	3
3600	3	5	0,494	0,813	0,000	3
3600	4	5	0,598	0,810	0,000	4
3600	0	7	0,496	0,854	0,000	3
3600	1	7	0,546	0,820	0,000	3
3600	2	7	0,556	0,817	0,000	4
3600	3	7	0,536	0,802	0,000	3
3600	4	7	0,602	0,824	0,000	3

Tabla 11: Resultados Gisette

Tiempo	Fold	Semilla	AUC	FSC	CK	<i>Features</i>
1200	0	3	0,762	0,651	0,000	5
1200	1	3	0,747	0,555	0,240	4
1200	2	3	0,641	0,549	0,249	3
1200	3	3	0,849	0,705	0,489	3
1200	4	3	0,817	0,762	0,440	3
1200	0	5	0,886	0,810	0,575	6
1200	1	5	0,729	0,320	0,164	4
1200	2	5	0,827	0,780	0,537	3
1200	3	5	0,671	0,616	0,266	3
1200	4	5	0,658	0,620	0,254	3
1200	0	7	0,884	0,776	0,464	3
1200	1	7	0,839	0,781	0,439	6
1200	2	7	0,890	0,839	0,671	6
1200	3	7	0,888	0,806	0,538	4
1200	4	7	0,726	0,603	0,309	5
3600	0	3	0,873	0,651	0,000	6
3600	1	3	0,640	0,631	0,223	3
3600	2	3	0,661	0,573	0,267	3
3600	3	3	0,839	0,778	0,524	3
3600	4	3	0,817	0,762	0,440	3
3600	0	5	0,863	0,762	0,510	4
3600	1	5	0,779	0,448	0,229	6
3600	2	5	0,829	0,777	0,537	3
3600	3	5	0,671	0,616	0,266	3
3600	4	5	0,672	0,597	0,237	3
3600	0	7	0,885	0,777	0,468	3
3600	1	7	0,849	0,804	0,542	6
3600	2	7	0,687	0,718	0,328	3
3600	3	7	0,870	0,780	0,481	4
3600	4	7	0,726	0,600	0,306	5

Tabla 12: Resultados Ionosphere

Tiempo	Fold	Semilla	AUC	FSC	CK	<i>Features</i>
1200	0	3	0,889	0,913	0,676	5
1200	1	3	0,844	0,857	0,545	6
1200	2	3	0,749	0,851	0,496	6
1200	4	3	0,552	0,804	0,187	4
1200	0	5	0,948	0,951	0,818	6
1200	3	5	0,823	0,887	0,638	6
1200	0	7	0,960	0,929	0,760	6
1200	2	7	0,805	0,808	0,324	4
1200	3	7	0,796	0,667	0,000	3
3600	0	3	0,889	0,913	0,676	5
3600	1	3	0,844	0,857	0,545	6
3600	2	3	0,749	0,851	0,496	6
3600	3	3	0,805	0,893	0,601	6
3600	4	3	0,552	0,804	0,187	4
3600	0	5	0,948	0,951	0,818	6
3600	1	5	0,767	0,750	0,000	3
3600	2	5	0,808	0,835	0,420	4
3600	3	5	0,823	0,887	0,638	6
3600	4	5	0,926	0,913	0,688	6
3600	0	7	0,960	0,929	0,760	6
3600	1	7	0,675	0,852	0,248	3
3600	2	7	0,805	0,808	0,324	4
3600	3	7	0,796	0,667	0,000	3
3600	4	7	0,724	0,911	0,585	5

Tabla 13: Resultados WBC

Tiempo	Fold	Semilla	AUC	FSC	CK	Features
1200	0	3	0,974	0,838	0,673	4
1200	1	3	0,983	0,537	0,397	5
1200	2	3	0,987	0,781	0,700	5
1200	3	3	1,000	0,947	0,930	4
1200	4	3	0,992	0,870	0,838	6
1200	0	5	0,948	0,137	0,059	4
1200	1	5	0,842	0,000	0,000	3
1200	2	5	0,976	0,800	0,724	5
1200	3	5	0,988	0,750	0,688	3
1200	4	5	0,984	0,941	0,925	5
1200	0	7	0,956	0,547	0,312	4
1200	1	7	0,992	0,936	0,891	6
1200	2	7	0,981	0,946	0,920	9
1200	3	7	0,992	0,966	0,954	5
1200	4	7	0,978	0,471	0,409	3
3600	0	3	0,974	0,838	0,673	4
3600	1	3	0,983	0,537	0,397	5
3600	2	3	0,987	0,781	0,700	5
3600	3	3	1,000	0,947	0,930	4
3600	0	5	0,948	0,137	0,059	4
3600	1	5	0,842	0,000	0,000	3
3600	2	5	0,976	0,800	0,724	5
3600	3	5	0,988	0,750	0,688	3
3600	4	5	0,984	0,941	0,925	5
3600	0	7	0,956	0,547	0,312	4
3600	1	7	0,992	0,936	0,891	6
3600	2	7	0,981	0,946	0,920	9
3600	3	7	0,992	0,966	0,954	5
3600	4	7	0,978	0,471	0,409	3

Anexo B. Códigos

Los códigos desarrollados en la presente memoria se encuentran disponibles en el siguiente enlace de [GitHub](#) donde además se incluyen las instancias utilizadas.

**UNIVERSIDAD DE CONCEPCIÓN – FACULTAD DE INGENIERÍA RESUMEN DE
MEMORIA DE TÍTULO**

Departamento : Departamento de Ingeniería Industrial
Carrera : Ingeniería Civil Industrial
Nombre del memorista : Mathias Felipe Badilla Salamanca
Título de la memoria : Metaheurística multiobjetivo para support vector machines
con feature selection
Fecha de la presentación oral : 29 de agosto, 2024

Profesor(es) Guía : Carlos Emilio Contreras Bolton
Rosa Daniela Medina Durán
Profesor(es) Revisor(es) : Lorena Del Carme Pradenas Rojas
Concepto :
Calificación :

Resumen

La selección de *features* es una área importante en el aprendizaje supervisado, especialmente al manejar grandes volúmenes de datos con muchas features. Support Vector Machines (SVM) ha demostrado ser un modelo sencillo y eficiente para trabajar en la selección de features gracias al uso regresores específicos para cada feature, los cuales son anulados si no son relevantes. Por ello, esta memoria de título propone una metaheurística multiobjetivo para mejorar la selección de features aplicando SVM. La metaheurística multiobjetivo se basa en el NSGA-II y busca optimizar tanto el rendimiento predictivo como la eficiencia computacional del modelo.

El enfoque propuesto es evaluado en diversos conjuntos de datos, considerando métricas de clasificación y tiempo de ejecución. Los resultados muestran que la metaheurística presentada no solo reduce la dimensionalidad del problema sino que también mantiene o mejora la calidad de las predicciones. Este trabajo contribuye al campo de la inteligencia artificial y la investigación de operaciones al demostrar que el uso de metaheurísticas pueden ser efectivas en la mejora de algoritmos de aprendizaje supervisado como SVM.