

WRAPPER PARA LA CONSTRUCCIÓN DE MODELOS DE  
APRENDIZAJE SUPERVISADO BASADO EN ARREGLOS DE  
COBERTURA QUE PERMITE LA ESTIMACIÓN DE LA  
IMPORTANCIA DE LAS VARIABLES DE ENTRADA Y LA  
SELECCIÓN DE ATRIBUTOS



HUGO ANDRES DORADO BETANCOURT

Tesis de Maestría en Computación

Director: PhD. Carlos Alberto Cobos Lozada  
Co-Director: PhD. José Torres Jiménez  
(CINVESTAV, Tamaulipas, México)

Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Sistemas  
Grupo de I+D en Tecnologías de la Información (GTI)  
Línea de Investigación: Minería de datos  
Popayán, Enero de 2019

HUGO ANDRES DORADO BETANCOURT

WRAPPER PARA LA CONSTRUCCIÓN DE MODELOS DE  
APRENDIZAJE SUPERVISADO BASADO EN ARREGLOS DE  
COBERTURA QUE PERMITE LA ESTIMACIÓN DE LA  
IMPORTANCIA DE LAS VARIABLES DE ENTRADA Y LA  
SELECCIÓN DE ATRIBUTOS

Tesis presentada a la Facultad de Ingeniería  
Electrónica y Telecomunicaciones de la  
Universidad del Cauca para la obtención del  
Título de

Magíster en  
Computación

Director  
Director: PhD. Carlos Alberto Cobos Lozada  
Co-Director: PhD. José Torres Jiménez  
(CINVESTAV, Tamaulipas, México)

Popayán  
2019

# Agradecimientos

Al Doctor Carlos Alberto Cobos Lozada, por su dirección, supervisión, disponibilidad y acompañamiento en este trabajo de grado, su calidad de enseñanza hizo que el desarrollo de esta investigación fuera un proceso muy enriquecedor.

Al Doctor José Torres Jiménez y al Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV – Tamaulipas, México) por su asesoramiento y apoyo en el desarrollo de este proyecto.

Al Doctor Daniel Jiménez y al Centro Internacional de Agricultura Tropical (CIAT), por permitirme la disponibilidad de tiempo en cursar la maestría, facilitarme los recursos computacionales para hacer las pruebas de esta investigación y apoyar mi pasantía de investigación en la oficina de Nairobi, África.

Al Doctor Enrique Herrera del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada, España, por su asesoría y apoyo en la estancia de investigación.

A mis colegas del CIAT, familiares y amigos, por la motivación y paciencia en este proceso.

# Resumen Estructurado

Los wrappers son un tipo de método para la selección de atributos que pretenden encontrar un subconjunto de variables que mejore el desempeño de un Modelo de Aprendizaje Supervisado (MAS) al remover variables redundantes o irrelevantes. Por otro lado, la importancia de variables es una medida que permite cuantificar la relevancia de cada variable de entrada en la explicación de la variable de salida, ambos métodos son utilizados en la minería de datos cuando se realiza el preprocesamiento de los datos y para facilitar la interpretación de los resultados.

Los arreglos de cobertura (CAs), son objetos matemáticos que han sido ampliamente utilizados en la detección de fallas en software y hardware, siendo esta una alternativa viable con mayor cobertura y menor costo, frente a las pruebas exhaustivas. No obstante, hasta el momento en la literatura no se ha evaluado el potencial de utilizar este concepto en el proceso de selección de atributos o importancia de variables, por lo tanto en el presente trabajo se presenta una propuesta para implementar un nuevo wrapper basado en una estrategia de búsqueda que utiliza CAs o arreglos de cobertura incrementales binarios (ICA), y además se plantea un nuevo algoritmo para calcular la importancia de variables, el cual puede ser adaptado a cualquier wrapper. La evaluación de ambas propuestas se realizó utilizando seis MAS, sobre siete conjuntos de datos obtenidos del UCI Machine Learning Repository.

Los resultados sugieren que el uso de CAs e ICAs de fuerza seis en la selección de atributos, mejoran significativamente el desempeño de los MAS y reducen el número de variables requeridas, además en el orden de importancia de variables retienen en las primeras variables una mayor exactitud y consistencia con los resultados del proceso de selección de atributos. La comparación realizada con otros métodos del estado del arte mostró que el wrapper propuesto escoge una cantidad de variables similar a un wrapper basado en algoritmos Genéticos (GA) y a otro basado en Optimización por enjambre de partículas (PSO) pero no los supera en exactitud, y que la propuesta de importancia de variables domina los resultados de los filtros basados en ganancia de información (IG) y radio de ganancia (GR) y supera en la mayoría de los casos los obtenidos con la medida de importancia de variables de Random Forest (RFIV).

**Palabras claves:** Wrappers, Importancia de Variables, Algoritmos Genéticos, Optimización por enjambre de partículas, Modelos de Aprendizaje Supervisado, Filtros

# Structured Abstract

Wrappers is a feature selection method that aims to find a subset of variables that improves the performance of a Supervised Learning Model (MAS) by removing redundant or irrelevant variables. On the other hand, the importance of variables is a measure that allows to quantify the relevance of each input variable in the explanation of the output variable, both methods are used in data mining when the data is preprocessed and also facilitates the interpretation of the results.

Covering Arrays (CAs), are mathematical objects that have been widely used in the detection of software and hardware failures, this being a viable alternative with greater coverage and lower cost, compared to the exhaustive tests. However, until now in the literature the potential of using this concept in the process of selecting attributes or importance of variables has not been evaluated. Therefore, in the present work a proposal is presented to implement a new wrapper based on a search strategy that uses CAs or binary incremental coverage arrangements (ICA) for feature selection. In addition, a new algorithm is also proposed in order to calculate a variable importance measure, which can be adapted to any wrapper. The evaluation of both proposals was carried out using six MAS on seven data sets obtained from the UCI Machine Learning Repository.

The results suggest that the use of CAs and ICAs, of force six in the feature selection, significantly improve the performance of the MAS and reduce the number of required variables. In addition, in the order of importance of variables they retain in the first variables a greater accuracy and consistency with the results of the attribute selection process. The comparison made with other state-of-the-art methods showed that the proposed wrapper chooses a number of variables similar to a wrapper based on Genetic Algorithms (GA) and another based on Particle Swarm Optimization (PSO), but does not exceed them in accuracy. Additionally, the proposal of importance of variables dominates the results of the filters based on Information Gain (IG) and Gain Ratio (GR), and in most cases, exceeds those obtained with the Random Forest Variable Importance Measure (RFVI).

**Keywords:** Wrappers, Variable Importance Measure, Genetic algorithms, Particle Swarm Optimization, Supervised learning models, filters.



# Tabla de Contenido

1	Introducción.....	13
1.1	Definición del Problema .....	14
1.2	Justificación.....	15
1.3	Objetivos .....	15
1.3.1	Objetivo General.....	15
1.3.2	Objetivos específicos.....	16
1.4	Resultados Obtenidos .....	16
1.5	Organización del documento.....	17
2	Marco Teórico .....	18
2.1	Selección de atributos .....	18
2.1.1	Filtros.....	18
2.1.2	Wrappers .....	20
2.1.3	Embebidos.....	21
2.2	Importancia de variables .....	22
2.3	Arreglos de cobertura.....	23
2.3.1	Arreglos de cobertura incrementales .....	25
2.4	Test no paramétricos para comparación de algoritmos.....	26
2.4.1	Prueba de Friedman .....	27
2.4.2	Prueba de Rangos Alineados de Friedman .....	27
3	Estado del Arte.....	29
3.1	Métodos de selección de atributos.....	29
3.1.1	Filtros basados en entropía .....	29
3.1.2	Wrappers basados en metaheurísticas .....	30
3.2	Medidas de importancia de variables .....	40
4	Selección de atributos utilizando arreglos de cobertura .....	46
4.1	Algoritmo wrapper utilizando CAs o ICAs.....	46
4.2	Evaluación de wrapper utilizando CA e ICA.....	48
4.2.1	Conjuntos de datos para validación.....	48

4.2.2	Modelos de aprendizaje supervisado (MAS) y optimización de parámetros .....	49
4.2.3	CAs e ICAs usados.....	50
4.2.4	Comparación de los resultados según la fuerza .....	51
4.2.5	Efecto del wrapper propuesto sobre los diferentes MAS .....	55
4.2.6	Comparación contra métodos del estado de arte .....	55
5	Importancia de variables utilizando arreglos de cobertura .....	63
5.1	Algoritmo para importancia de variables .....	63
5.2	Comparación de los resultados según la fuerza .....	65
5.3	Comparación con métodos del estado de arte .....	68
5.4	Aplicaciones .....	78
6	Conclusiones y trabajo futuro.....	80
6.1	Conclusiones.....	80
6.2	Trabajo Futuro.....	82
7	Referencias .....	83
8	Anexos .....	90

## Lista de Figuras

Figura 1. El enfoque de selección de atributos por filtros. ....	19
Figura 2. El enfoque de wrappers para la selección de atributos. ....	21
Figura 3. Ejemplo de un CA binario de fuerza dos y alfabeto binario .....	24
Figura 4. Extracción de un subconjunto de variables utilizando CA .....	25
Figura 5. ICA(2,7,13,24;4,10,2) con t máximo de 4, k=10 y v=2.....	26
Figura 6. Número de pruebas requeridas por nivel de fuerza en cada dataset .....	51
Figura 7. Porcentaje de dataset-MAS que superan a BL según la fuerza del CA.....	52
Figura 8. Incremento en exactitud y fuerza para los wrappers basados en CAs. ....	54
Figura 9. Incremento en la exactitud respecto a la BL por fuerza y MAS. ....	55
Figura 10. Diferencia entre la exactitud de PSO y los wrappers basados en CAs.....	60
Figura 11. Comparación entre la exactitud y el número de EFOs, por dataset y MAS. ....	62
Figura 12. Importancia de variables según CAs de fuerza 6, para Zoo. ....	66
Figura 13. Importancia de variables en GA y PSO, para Zoo.....	69
Figura 14. Incremento de la exactitud según orden de importancia de variables, en los 6 MAS para Zoo.....	70
Figura 15. Incremento en la exactitud según el orden de importancia de las variables, para Zoo. ....	71

# Lista de Tablas

Tabla 1. Calculo de medidas de ranking para selección de atributos .....	19
Tabla 2. Resumen del estado del arte de los wrappers.....	36
Tabla 3. Resumen del estado del arte de la importancia de variables.....	43
Tabla 4. Descripción de los datasets de prueba.....	48
Tabla 5. MAS y sus parámetros correspondientes.....	49
Tabla 6. Friedman para la comparación de CA-Fuerza de los wrappers. ....	53
Tabla 7. Comparación de la exactitud y número de variables de los wrappers. ....	57
Tabla 8. Friedman para la comparación de CA-Fuerza según AMIV y COBESV. ....	67
Tabla 9. Comparación de las importancias de variables según el AMIV. ....	72
Tabla 10. Comparación de las importancias de variables según el COBESV. ....	74
Tabla 11. Prueba de Friedman para la comparación de AMIV y su post hoc sobre la totalidad de los resultados.....	76
Tabla 12. Prueba de Friedman para la comparación de COBESV y su post hoc sobre la totalidad de los resultados. ....	76

## Lista de Ecuaciones

Ecuación 1. Estadístico de prueba para test de Friedman [51] .....	27
Ecuación 2. Estadístico de prueba para post hoc test de Friedman [51] .....	27
Ecuación 3. Estadístico de prueba para rangos alineados de Friedman [50] .....	28
Ecuación 4. Estadístico de prueba para post hoc de FART [50] .....	28
Ecuación 5. Wrapper para selección de atributos utilizando filas de un CA. ....	48

# Lista de Algoritmos

Algoritmo 1. Wrapper basado en CAs e ICAs .....	47
Algoritmo 2. Pseudocódigo del algoritmo para definir la importancia de variables .....	64

# 1 Introducción

En un momento en el que la tecnología permite recopilar gran cantidad de información a través de dispositivos, sensores, imágenes o sonido, se hace común la construcción de bases de datos a gran escala, lo que ha generado una nueva tendencia masiva para tomar decisiones basadas en datos, que cada vez se extiende a más campos de la ciencia, los negocios, el mercadeo, entre otros [1], [2]. Consecuentemente se ha popularizado el uso de modelos de predicción o Modelos de Aprendizaje Supervisado (MAS), los cuales se construyen a partir de funciones que maximizan las relaciones entre un conjunto de variables de entrada con respecto a una variable de salida, esto con el propósito de realizar predicciones, algunos ejemplos son: la clasificación de correos electrónicos como spam y no spam de acuerdo a la frecuencia de ciertas palabras o aparición de algunos caracteres, la identificación de clientes que pagaran o no un crédito de acuerdo a sus condiciones económicas e historia crediticia, el estado del tiempo el día de mañana de acuerdo al tiempo del día de hoy [3], [4]. La construcción de MAS tiene varios retos, uno de ellos es el de establecer una medida de *importancia de variables*, que permita identificar el orden de las variables que más afectan en la predicción, y otro reto es el de la *selección de atributos*, el cual se hace necesario en casos en los que se disponga de información con grandes cantidades de variables o alta dimensionalidad, y se requiera remover variables redundantes e irrelevantes, dado que mantenerlas puede terminar por afectar el desempeño de la calidad de la predicción [5].

La *importancia de variables*, pretende establecer un ranking sobre el conjunto de variables de entrada, por medio de la estimación de una medida cuantitativa para cada una, indicando la relevancia o contribución sobre la variación de la variable de salida [6]. Dicha medida permite a los investigadores reconocer y controlar factores influyentes en el fenómeno que están modelando [2], a su vez fortalecer la comprensión de las relaciones entre las variables que participan en un MAS y permitiendo además ir más allá de una simple predicción basada en la ejecución de un algoritmo que se comporta como una caja negra [6].

Por otro lado, el proceso de *selección de atributos* busca identificar un subconjunto óptimo de variables que se deben utilizar para la construcción de un MAS y con ello mejorar el desempeño, interpretabilidad o ambas al tiempo en comparación a cuando se tienen en cuenta todas las variables de entrada [8]. Algunos métodos de

selección de atributos utilizan medidas de distancia, información o independencia de las variables, sin utilizar el modelo (MAS) de interés, a estos métodos se les conoce como **Filtros**. Otros métodos utilizan el algoritmo de aprendizaje como parte de la función que evalúa la calidad del subconjunto de variables que se está buscando, a estos métodos se les conoce como **Wrappers**. También existe otro enfoque que utilizan algunos MAS internamente en el proceso de aprendizaje para estabilizar la influencia de cada atributo y son conocidos como **Embebidos** [9].

## 1.1 Definición del Problema

Los wrappers en general son más precisos que los filtros pero tienen la desventaja de ser costosos a nivel computacional (el costo computacional crece exponencialmente con el número de variables de entrada) dado que requiere ejecutar varias veces el MAS que se está construyendo, y pese a que existen metaheurísticas tales como los algoritmos genéticos [10], la optimización por enjambre de partículas [11], el recocido simulado y sus variaciones [12], las soluciones tienen el riesgo de quedar atrapadas en óptimos locales y demandan un número elevado de experimentos [8].

Por otro lado, los resultados de las soluciones visitadas o evaluadas por un wrapper usualmente no son aprovechados para definir una medida de importancia de variables, dicha estimación podría aportar un valor agregado al esfuerzo computacional realizado. Adicionalmente se detectó que hay un concepto que tiene el potencial de ser implementado para un wrapper debido a ciertas propiedades que le permiten competir con los métodos reportados en el estado del arte, dicho concepto es el denominado Arreglos de Cobertura (Covering Arrays, CA), que son objetos matemáticos que pueden ser expresados en matrices numéricas con  $N$  filas y  $k$  columnas o factores, asociados a un parámetro  $t$  conocido como la fuerza de la cobertura, el cual representa una medida del cubrimiento de las interacciones entre los diferentes factores o columnas [13].

Sí bien las cualidades de los CA han sido explotadas en varios campos, hasta ahora no se han usado como herramienta para la selección de atributos o la estimación de la importancia de las variables en un MAS. Por lo tanto, la presente investigación buscó dar respuesta a la siguiente pregunta: ¿Cómo mejorar el desempeño y la eficiencia en la estimación de la importancia de variables y la selección de atributos en un MAS a partir de un wrapper que se basa en CA?

## 1.2 Justificación

Los CAs han sido utilizados en el diseño de experimentos, test de software y hardware, entre otros campos, dado que en estas aplicaciones se requiere probar combinaciones de distintos factores donde muchas veces no es práctico realizar test exhaustivos debido a alto costo en tiempo y esfuerzo [14]. El enfoque de los CAs, reduce en gran parte el número de pruebas que se necesitan ejecutar, mientras mantienen la efectividad de la tarea que se está desarrollando [15], [16], y dada su capacidad de dar la mayor cobertura con el menor esfuerzo posible, en esta tesis se les consideró una herramienta viable y prometedora para soportar la propuesta de un nuevo wrapper que logre integrar la selección de atributos y estimar una medida de importancia de variables de entrada.

Desde el punto de vista de investigación, con la definición e implementación de un algoritmo wrapper que incluya una medida de importancia de variables basado en CAs, se buscó aportar nuevo conocimiento en el área del aprendizaje de máquina y minería de datos, además de ser un nuevo enfoque que se puede seguir explotando en el futuro.

Se espera que la implementación del wrapper y la medida de importancia de variables propuesta impacte los campos que utilizan MAS, minería de datos y big data tales como: Medicina, riesgo crediticio, agricultura, entre otros, especialmente cuando se analizan conjuntos de información de alta dimensionalidad, ya que la propuesta tiene el potencial de competir con otros métodos del estado del arte teniendo el chance de encontrar mejores soluciones y/o reducir el tiempo de ejecución.

## 1.3 Objetivos

A continuación, se presentan los objetivos (general y específicos) tal y como fueron aprobados por el Consejo de Facultad de la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca en el anteproyecto de la tesis.

### 1.3.1 Objetivo General

Proponer un wrapper para la construcción de modelos de aprendizaje supervisado basado en arreglos de cobertura que permite la estimación de la importancia de las

variables de entrada y la selección de atributos con resultados comparables a los presentados en el estado del arte en tiempo de ejecución o exactitud.

### 1.3.2 Objetivos específicos

- Definir un wrapper para realizar la selección directa de atributos utilizando CAs sobre las siguientes técnicas de aprendizaje supervisado: Random Forest, Multi-layer Perceptron, C4.5, Naïve Bayes y Support Vector Machines.
- Definir una medida que permita establecer la importancia de las variables de entrada en modelos de aprendizaje supervisado utilizando el wrapper previamente propuesto.
- Evaluar la calidad de los resultados y la eficiencia del wrapper planteado y compararlo con resultados de otros métodos en el estado del arte, usando el error de clasificación, la precisión, el recuerdo, la medida  $f$  o el tiempo de ejecución sobre los conjuntos de datos de Glass, Ionosfera, Sonar, Vehicle, Wdbc, Wine y Zoo, reconocidos por el estado del arte y disponibles en el repositorio de la UCI (University of California Irvine).

## 1.4 Resultados Obtenidos

- Monografía del trabajo de grado. Corresponde al presente documento, que recopila los logros más importantes del trabajo de investigación.
- CD con el código fuente del wrapper y la medida de importancia de variables documentados en R (**Anexo 1**).
- Artículo que será sometido a una revista indexada JCR (Journal Citation Reports) cuartil 1 o 2 con la siguiente referencia preliminar: Dorado H, Cobos C., Torres-Jimenez J., Burra D., Herrera-Viedma E., "*Wrapper for building classification models using Covering Arrays*" (**Anexo 2**).
- Artículo publicado en la revista Lecture Notes in Computer Science, indexada en la categoría A2 del PUBLINDEX de Colciencias, que corresponde a las memorias del evento MIKE 2018 - Mining Intelligence and Knowledge Exploration (MIKE) realizado en Cluj, Rumania, Diciembre del 2018. El artículo tiene la siguiente referencia: Dorado H., Cobos C., Torres-Jiménez J., Jimenez D., Mendoza M. "*A proposal to estimate the variable importance measures in predictive models using results from a Wrapper*", LNCS, volume 11308, DOI: 10.1007/978-3-030-05918-7\_33. (**Anexo 3**).

- Artículo publicado en la revista Lecture Notes in Computer Science, indexada en la categoría A2 del PUBLINDEX de Colciencias, que corresponde a las memorias del evento MICAI 2018 - Mexican International Conference on Artificial Intelligence que se realizó del 22 al 27 de octubre en Guadalajara México. El artículo tiene la siguiente referencia: Dorado H, Delerece S., Jimenez D., Cobos C. "*Finding optimal farming practices to increase crop yield through Global-best Harmony Search and predictive models, a data-driven approach*", LNAI Volume 11289, DOI: 10.1007/978-3-030-04497-8\_2. (**Anexo 4**)

## 1.5 Organización del documento

El resto de la monografía está estructurada de la siguiente manera:

Capítulo 2. Presenta el marco teórico, que incluye los conceptos básicos necesarios para entrar en el contexto de la temática que se trata.

Capítulo 3. Presenta el estado del arte, donde se analizan los métodos y los algoritmos más significativos reportados en la literatura para selección de atributos e importancia de variables. Se presenta en mayor detalle los wrappers basados en GA (genéticos) y PSO (enjambre de partículas).

Capítulo 4. Presenta la propuesta de selección de atributos utilizando CAs, se evalúan los resultados con diferentes niveles de fuerza, y se hace la respectiva comparación con los métodos del estado del arte.

Capítulo 5. Presenta la propuesta para medir la importancia de variables utilizando CAs, al igual que en capítulo anterior, se evalúan los resultados con diferentes niveles de fuerza, y se hace la respectiva comparación con los métodos del estado del arte.

Capítulo 6. Presenta las conclusiones del trabajo realizado y el trabajo futuro que el grupo de investigación espera adelantar en el corto plazo.

Capítulo 7. Presenta las referencias que soportan el trabajo realizado.

## 2 Marco Teórico

### 2.1 Selección de atributos

El proceso de selección de atributos es fundamental en la construcción de MAS y se desarrolla en la etapa de preprocesamiento. Esta tarea se realiza con el fin de escoger un subconjunto de variables de entrada, cuya selección reduzca el ruido de la información, descarte variables redundantes, mejore significativamente la efectividad y robustez del MAS, y reduzca el tiempo de procesamiento que puede demandar la ejecución del modelo, sin que se afecte negativamente la calidad de la predicción [17].

Existe un gran número de aplicaciones donde ha sido relevante realizar la selección de atributos, por ejemplo en el análisis de micro arreglos genéticos, dado que cada expresión estandarizada de un gen puede contener cientos de variables donde algunas de ellas están altamente correlacionadas y por lo tanto hay presencia de variables redundantes que no proporcionan información adicional y terminan haciendo ruido o sesgo [9]. Otra aplicación es la de clasificación de textos donde la tarea puede variar desde clasificar automáticamente una página web en un directorio web, hasta la detección de correos electrónicos no solicitados (Spam). En estos casos una observación está representada por un vector donde cada componente representa la frecuencia de aparición de una palabra específica (representación basada en el modelo de espacio vectorial, también conocido como bolsa de palabras), y dado que son muchas las palabras del vocabulario, se puede comenzar a hacer un filtro descartando las menos frecuentes. Luego, dependiendo de la tarea que se quiera realizar, hay unas palabras mucho más influyentes y otras con mucho menos relevancia que se pueden eliminar, de esta manera, el proceso de selección de atributos es de gran utilidad en estas aplicaciones [18].

La selección de atributos se puede implementar bajo tres enfoques principales que son: filtros, wrappers y embebidos.

#### 2.1.1 Filtros

La selección de atributos realizada con filtros se hace previa al entrenamiento del MAS (**Figura 1**), normalmente se establece un ranking por cada una de las variables

de entrada, el cual pretende definir la relevancia de cada variable según su contribución para discriminar las clases de la variable de salida, luego, las variables se descartan cuando quedan por debajo de un umbral que es definido previamente [9], [19]. Los filtros son conocidos por su sencillez y han sido de utilidad en conjuntos de datos con muchas variables, no requieren un proceso de entrenamiento lo cual les da la ventaja de generar resultados rápidos.

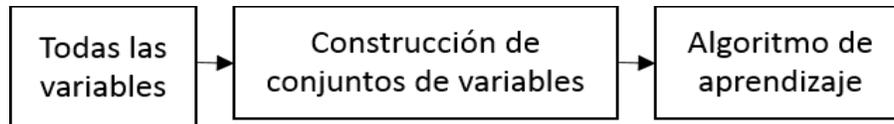


Figura 1. El enfoque de selección de atributos por filtros.

En la literatura se han propuesto una amplia variedad de medidas para aplicar selección de atributos utilizando filtros, algunos de ellos estadísticos y otros basados en una medida de incertidumbre sobre la clasificación llamada entropía [20].

Entre los más comunes están: **La ganancia de información** (Information Gain, IG) o **información mutua** (Mutual Information, MI), el cual refleja la disminución en la entropía sobre la clasificación de la variable de respuesta después de que uno de los predictores haya hecho su aporte. Esta medida es sesgada ya que tiende a favorecer variables con muchos valores [9], [21], [22]; **Radio de ganancia** (Gain ratio, GR), que pretende reducir el sesgo que tiene el IG al dividirlo por la entropía de la variable de entrada, su valor se mueve en un intervalo entre [0,1], siendo 1 un escenario en el cual la información de la variable de entrada predice completamente a la variable de salida, desafortunadamente favorece variables con pocos valores [21], [23]. **Incetidumbre simétrica** (Symmetric Uncertainty, SU), que se calcula a partir de IG y la suma de la entropía de la variable de respuesta con la entropía de la variable de entrada, luego se multiplica por un factor de corrección para que su valor quede estandarizado en un intervalo de [0,1], SU también favorece variables con pocos valores [24], [25]. **Chi cuadrado**, es un test estadístico que mide la asociación entre dos variables, la hipótesis nula sugiere independencia, entre más altos sean los valores del estadístico, es mayor la evidencia en contra de la hipótesis nula ( $H_0$ ) [26], [27]. La descripción de cada filtro puede ser visto en la **Tabla 1**.

Tabla 1. Calculo de medidas de ranking para selección de atributos

Medida	Calculo
Entropía ( $H(Y)$ )	$-\sum_y p(y)\log(p(y))$

Medida	Calculo
Entropía condicionada ( $H(Y X)$ )	$-\sum_x \sum_y p(x,y)\log(p(y x))$
IG	$H(Y) - H(Y X)$
GR	$\frac{IG}{H(X)}$
SU	$2 \frac{IG}{H(X) + H(Y)}$
Chi cuadrado	$\sum_{i=1}^r \sum_{j=1}^c \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$ ; $O_{ij}$ : Frecuencia observada $E_{ij}$ : Frecuencia esperada bajo $H_c$

Existe otro enfoque para el diseño de filtros, que es multivariado y evalúa el desempeño no individualmente por cada variable, sino por subconjuntos de variables; esto por medio de una estrategia de búsqueda que pretende encontrar el mejor subconjunto de variables [19]. Algunos de estos filtros multivariados son: Criterio de inconsistencia, mínima redundancia máxima relevancia (mRmR) [8], correlación basada en selección de atributos (CFS) [28] y rápida correlación basada en filtros (FCBF) [29].

### 2.1.2 Wrappers

La principal característica de la selección de atributos vía wrappers, es que utiliza el MAS dentro del proceso de evaluación de cada subconjunto de variables y por esta razón el proceso es costoso computacionalmente. La búsqueda del mejor subconjunto de variables es un problema *NP* que se hace computacionalmente prohibitivo en la medida que el número de variables de entrada crece, y por esta razón se requiere de una estrategia de búsqueda que indique las soluciones más prometedoras que se deben evaluar [30]. Adicionalmente, es necesario un criterio de evaluación que defina el desempeño del MAS que se obtiene con cada subconjunto de variables. El esquema general de un wrapper se presenta en la **Figura 2**, en donde se puede observar que en la selección de atributos participa el MAS (algoritmo de aprendizaje) como una caja negra que permite identificar el mejor subconjunto de variables.

Existen varios enfoques que han sido adaptados para la implementación de wrappers y la principal característica que los diferencia es el criterio de búsqueda utilizado para definir los subconjuntos de variables que se van a experimentar [19].

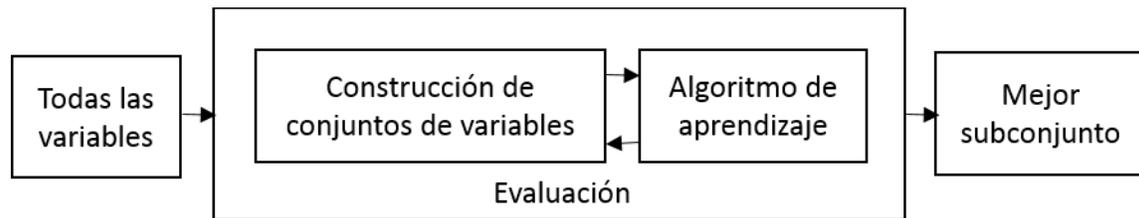


Figura 2. El enfoque de wrappers para la selección de atributos.

Los primeros son conocidos como **métodos de complejidad exponencial**, implican un número grande de pruebas a realizar, de hecho, entre ellos se encuentra la búsqueda exhaustiva, y pese a que garantiza una buena probabilidad de encontrar la mejor solución, frecuentemente es impráctica dado los altos costos computacionales. En consecuencia, los métodos de complejidad exponencial son difíciles de ejecutar incluso con conjuntos de datos con pocas variables de entrada [30], [31].

Otra estrategia de búsqueda bastante clásica es el enfoque **secuencial** [32], siendo parte de la familia de algoritmos greedy que utilizan funciones iterativas. Entre ellos se encuentra, el **forward** o selección hacia adelante, el cual parte de un subconjunto de variables a seleccionar vacío y en sus próximos pasos va adicionando cada vez una variable que enriquezca el desempeño del MAS evaluado; este proceso es realizado hasta que no se encuentran mejoras cuando se adiciona una variable más. Aunque el forward retorna resultados razonables en poco tiempo, en varios casos su rendimiento es pobre debido a la limitación que tiene de no poder excluir variables una vez han sido ingresadas en iteraciones previas. Su opuesto es el **backward**, en este caso se parte de una solución con todo el conjunto de variables, luego dentro del proceso se van removiendo algunas variables de manera que el desempeño del MAS incremente, y se detiene en la medida que excluir una variable más no produce mejores resultados. El backward se vuelve costoso computacionalmente al procesar MAS con muchas variables [33]. Existe otra línea de wrappers que se basa en metaheurísticas, más adelante en la sección 3.1.2 se hace una ampliación de estos métodos.

### 2.1.3 Embebidos

Los métodos embebidos surgen como una alternativa más rápida computacionalmente comparada con los wrapper. Son reconocidos por llevar a cabo la selección de atributos simultáneamente con el proceso de entrenamiento y

pueden estar incluidos dentro del MAS o pueden ser una función extendida del mismo [9], [19]. Es posible clasificarlos en tres tipos [8], [30]: el primero, son los **métodos de poda**, que consisten en entrenar un MAS con todas las variables y luego descartar algunas cuya eliminación mantenga o aumente el desempeño del MAS, esto puede ser realizado por medio de estrategias secuenciales, búsquedas greedy o medidas de contribución (vector de pesos, información mutua, mínima redundancia-máxima relevancia, entre otros), tal es el caso de la eliminación recursiva utilizando máquinas de soporte vectorial [34], [35]. El segundo, son los MAS con **mecanismos integrados** de selección de atributos, los cuales internamente en el aprendizaje, filtran o dan prioridad a las variables más informativas, por ejemplo ID3 [36], CART [37] y C4.5 [38]. El tercero, son los MAS que incluyen **modelos de regularización** [39], los cuales contienen funciones que llevan a los coeficientes de algunas variables a valores bajos o incluso a cero, al mismo tiempo que buscan minimizar el error de ajuste en el MAS, en este caso el valor del coeficiente indica la influencia que tiene cada variable sobre el MAS, por ejemplo Regression Lasso [40], [41] o Elastic Net [42].

## 2.2 Importancia de variables

La Importancia de variables puede ser comprendida desde distintas definiciones de acuerdo al método que se esté implementando para estimarla, esto debido a que fue un concepto que surgió desde distintos campos de la ciencia los cuales desarrollaron su propia metodología y terminología independientemente, algunas definiciones comunes son [7]:

1. Un indicador que cuantifica el cambio del valor de la respuesta del modelo con respecto al cambio en la permutación de un predictor o un conjunto de ellos.
2. Un indicador que estima la contribución de la incertidumbre de un predictor o un conjunto de ellos, a la incertidumbre de la variable de respuesta de un modelo.
3. Un indicador que cuantifica la fuerza de la dependencia entre la variable de respuesta de un modelo con respecto a una o un conjunto de predictores.

La implementación de medidas para importancia de variables surge de la necesidad para algunos investigadores y practicantes que trabajan en modelos computacionales (tales como los MAS), en detectar y priorizar bajo un orden, un grupo pequeño de variables de entrada, a partir de un conjunto mayor de variables candidatas donde algunas de ellas pueden ser no influyentes en la explicación de

una variable de salida. En la literatura se pueden encontrar términos asociados tales como análisis de sensibilidad o medidas de contribución o impacto [7].

Respecto a los tipos de importancia de variables, algunos de ellos según Wei et. al [7], son:

- Los *basados en diferencias*, los cuales son enfoques que utilizan derivadas parciales para medir la sensibilidad, en este grupo se incluyen métodos locales y métodos de Morris.
- Un *enfoque paramétrico*, donde se incluyen algunos índices basados en dependencia como el coeficiente de correlación de Pearson, coeficiente de correlación parcial de regresión lineal múltiple, métodos basados en rango y regresiones polinomiales.
- Un *enfoque no paramétrico*, del que hacen parte los MAS e involucra una lista de modelos que tienen la cualidad de definir la importancia de variables de acuerdo con los patrones que más influyen en su proceso de aprendizaje, algunos de ellos son Regresión Local Ponderada (LOESS), Modelos Aditivos Generalizados (GAM) y Random Forest (RF).
- Los métodos *basados en varianza*, los cuales son medidas de importancia relativa de una variable de entrada por la varianza parcial de la variable de salida explicada por esta variable, se clasifican en casos dependientes e independientes.
- Los métodos basados en *momentos de independencia*, los cuales pretenden obtener más información de la distribución de la variable aleatoria en la definición de su aporte a la varianza sobre la variable de respuesta, entre ellos están el índice delta y su respectiva extensión.

Algunos ejemplos de aplicación son: (i) en la agricultura para definir el orden de los factores climáticos más importantes que afectan la variación en rendimiento de un cultivo [43], [44], (ii) en medicina para detectar las características epigenómicas principales en un individuo que favorecen la aparición de cierto tipo de cáncer [45], y (3) en la industria para identificar variables responsables del cambio de la calidad en un producto [46].

## 2.3 Arreglos de cobertura

Los Arreglos de Cobertura (Covering Arrays, CA) son objetos matemáticos en los que se evalúa un número de factores o variables de interés y cada factor contiene un número determinado de posibilidades o valores. Han sido aplicados en campos

como el diseño de experimentos, biología, análisis de fallas en ingeniería, entre otros, pero actualmente son muy utilizados en las pruebas de calidad de software y hardware, buscando con estas pruebas, proveer productos de calidad sin incurrir en costos exagerados [14]. En comparación con las pruebas exhaustivas, en donde se evalúan todas las combinaciones posibles de los factores, los CAs permiten reducir el número de ejecuciones o pruebas con base en un parámetro denominado fuerza ( $t$ ), el cual define el número mínimo de las interacciones entre los factores que se desean evaluar.

Un CA puede ser expresado a partir de la notación  $CA(N: t, k, v)$ , que representa un arreglo bidimensional (matriz) de tamaño  $N \times k$ , donde  $N$  se refiere al número de pruebas (casos o filas),  $k$  el número de parámetros (factores o columnas).  $v$  el alfabeto, que indica el número posible de valores que puede tomar cada factor (por ejemplo, si se tiene un arreglo cuyos valores son 0 y 1 se dice que tiene un alfabeto de 2 o que es binario) y  $t$  que corresponde a la fuerza o grado de interacción entre los parámetros. La característica especial que tienen los CAs es que cualquier conjunto  $t$  de columnas que se extraigan del arreglo, contiene todas las combinaciones posibles de  $v^t$  tuplas en al menos una de las filas [13]. La **Figura 3** muestra un CA de fuerza dos ( $t=2$ ) y alfabeto binario en el que se puede observar que para cualquier conjunto de 2 columnas siempre aparecen la combinación de valores (0,0), (0,1), (1,0) y (1,1).

$$CA(N = 5: t = 2, k = 4, v = 2) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Figura 3. Ejemplo de un CA binario de fuerza dos y alfabeto binario

Los CAs pueden tener diferentes tamaños de filas y la misma fuerza. En caso de que alguno contenga el menor número de filas reportado en la literatura, se dice que es el CA óptimo conocido y su notación es  $CAN(N: t, k, v)$ . La construcción de los CAs es un problema complejo, para ello se han propuesto métodos exactos, algebraicos, greedy y basados en metaheurísticas [47]. En este trabajo no es de interés la creación de estos algoritmos dado que normalmente los CAs se pueden descargar de repositorios en internet, sino más bien el interés en esta tesis, es el uso de los CAs para la construcción de un nuevo wrapper y estimación de importancia de variables.

Respecto a los CAs y la selección de atributos, en esta investigación se asume que 0 representa la ausencia de una variable y 1 la presencia de esta, en una posible solución al problema de selección de atributos. En este sentido, cada fila de un CA binario se puede utilizar como una referencia que sugiere un subconjunto de variables candidatas a ser variables de entrada en un MAS, de manera que sí se construyen varios MAS dados por dichos subconjuntos y se evalúan con un criterio de calidad que implica ejecutar el MAS que se está implementando, se obtiene un nuevo algoritmo wrapper que utiliza como criterio de búsqueda un CA. El éxito de encontrar un buen subconjunto depende del número de interacciones posibles que el arreglo haya tomado en cuenta, lo cual puede ser controlado con el parámetro fuerza ( $t$ ). La **Figura 4**, muestra los subconjuntos de variables formados por un CA de fuerza dos ( $t=2$ ) y cuatro factores o variables ( $k=4$ ). Cuando un CA es óptimo las veces que un factor es seleccionado es aproximadamente igual al de los otros factores, en este ejemplo es posible observarlo, ya que todas las variables aparecen tres veces en el CA.

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \Rightarrow \text{Subconjuntos} \Rightarrow \begin{bmatrix} \text{Null} \\ x_2, x_3, x_4 \\ x_1, x_3, x_4 \\ x_1, x_2, x_4 \\ x_1, x_2, x_3 \end{bmatrix}$$

Figura 4. Extracción de un subconjunto de variables utilizando CA

### 2.3.1 Arreglos de cobertura incrementales

Una Torre de arreglos de cobertura o arreglo de cobertura incremental (Incremental Covering Arrays, ICA) de altura  $h$  es definida como una secuencia de  $h + 1$  CAs los cuales pueden ser representados por:  $C_0, C_1, \dots, C_h$ , donde  $C_0$  es un CA de fuerza  $t$  y  $k$  factores, este CA inicial es llamado base del ICA. Para el resto,  $i = 1, 2, \dots, h$ , cada  $C_i$  es un CA de fuerza  $t + i$  y  $k$  factores [48]. De manera que cada ICA de una fuerza y número de factores superiores tiene contenido en forma de sub matrices los CAs de fuerzas y factores inferiores.

Como ejemplo, se puede observar en la **Figura 5** un ICA que en sus primeras dos filas contiene el CA (2; 1, 10, 2), luego en sus primeras siete filas contiene el CA (7; 2, 10, 2), después en sus primeras trece filas contiene el CA(13; 3, 10, 2) y finalmente, en todas sus 24 filas el CA(24; 4, 10, 2).

Un ICA difiere de un CA clásico en que este es construido con base en los CA de menor fuerza, mientras que el CA clásico es construido independientemente de los

anteriores [48]. Dicha característica de los ICA es útil al realizar los experimentos para analizar la influencia de la fuerza sobre los resultados de la selección de atributos, dado que en caso de que se desee aumentar la fuerza para probar una mayor cobertura, solo implica adicionar unas cuantas pruebas, lo cual resulta en un proceso más eficiente, a diferencia en los CAs clásicos, en donde aumentar la fuerza implica evaluar un arreglo con muchas filas nuevas y por tanto, un número mayor de pruebas para el wrapper.

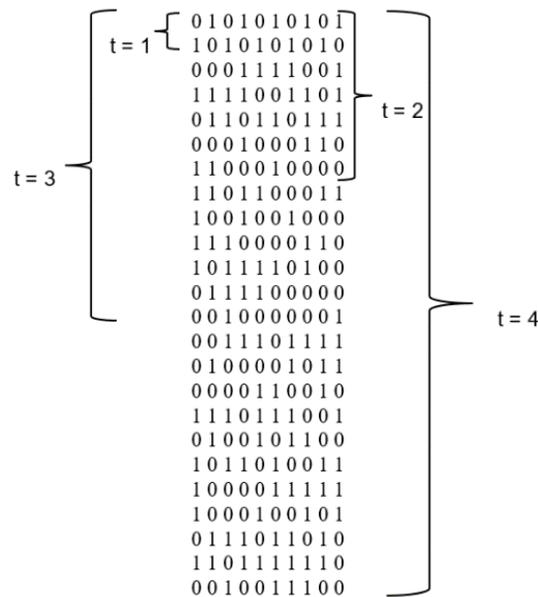


Figura 5. ICA(2,7,13,24;4,10,2) con  $t$  máximo de 4,  $k=10$  y  $v=2$

## 2.4 Test no paramétricos para comparación de algoritmos

Frecuentemente en la investigación de nuevos métodos de optimización y minería de datos se requiere comparar resultados con algoritmos ya publicados en el estado del arte, para ello se utilizan test estadísticos no paramétricos que permiten evaluar las diferencias de distintos métodos, estas pruebas son usadas para definir cuál algoritmo podría ser mejor con base en significancia estadística [49]. En este trabajo se utiliza la prueba de Friedman y la Prueba de Rangos Alienados de Friedman para realizar las comparaciones, ambos fueron seleccionados entre otras pruebas estadísticas por adaptarse a un contexto en el que hay pocas observaciones (conjuntos de datos de prueba), no requiere supuestos sobre la distribución de los datos analizados al ser no paramétricos y por ser frecuentemente utilizados para propósitos similares en el campo de la computación [50].

### 2.4.1 Prueba de Friedman

La Prueba (test) de Friedman es un test estadístico homólogo al análisis de varianza de dos vías, con esta prueba se pretende determinar si entre una muestra de resultados hay diferencias significativas entre tratamientos, el método ordena los datos por filas o bloques, remplazando el rango por su respectivo orden de la siguiente manera: el resultado del mejor desempeño debe ser remplazado con un valor de uno (1) el segundo mejor con dos (2), y así sucesivamente, dado el caso de empates ambos rangos se promedian [50].

Se define a  $r_i^j$  como el rango del  $j$ -ésimo método  $k$  sobre el  $i$ -ésimo de los  $n$  conjuntos de datos y se calcula el promedio de los rangos de cada método,  $R_j = \frac{1}{n} \sum_i r_i^j$ . Bajo la hipótesis nula de que todos los rangos son iguales, el estadístico de prueba se presenta en la **Ecuación 1**.

$$\chi_F^2 = \frac{12n}{k(k+1)} \left[ \sum_j R_j^2 - \frac{k(k+1)^2}{4} \right]$$

*Ecuación 1. Estadístico de prueba para test de Friedman [51]*

$\chi_F^2$  tiene distribución  $ji$ -cuadrado con  $k-1$  grados de libertad

La prueba post hoc para definir entre que pares de métodos existen las diferencias significativas está dado por la **Ecuación 2**.

$$z = (R_i - R_j) / \sqrt{\frac{k(k+1)}{6n}}$$

*Ecuación 2. Estadístico de prueba para post hoc test de Friedman [51]*

Donde  $R_i$  y  $R_j$  son el promedio del ranking de Friedman de dos métodos.

### 2.4.2 Prueba de Rangos Alineados de Friedman

La Prueba de Rangos Alineados de Friedman (Friedman Aligned Rank Test, FART) propone una modificación a la prueba tradicional de Friedman en el que se pretende hacer una adaptación con el propósito de construir una prueba más justa para los casos en los que se comparan pocos métodos (cuatro o cinco), esto, dado que la versión clásica puede verse afectada cuando  $k$  es pequeño [50].

En FART, se calcula el rendimiento medio de cada método en cada conjunto de datos, llamado valor de localización. Posteriormente, se calcula las diferencias entre el rendimiento obtenido de cada método con respecto al valor de localización. Este paso se repite para todos los métodos y conjuntos de datos. Las diferencias resultantes, llamadas observaciones alineadas, se ordenan desde uno hasta  $kn$  de forma relativa unas con otras. Los rankings asignados a las observaciones alineadas se denominan rankings alineados [50]. El estadístico de prueba para los rangos alienados de Friedman se calcula con base en la **Ecuación 3**.

$$T = \frac{(k - 1)[\sum_{j=1}^k \hat{R}_j^2 - (kn^2/4)(kn + 1)^2]}{\{kn(kn + 1)(2kn + 1)/6\} - (1/k) \sum_{i=1}^n \hat{R}_i^2}$$

*Ecuación 3. Estadístico de prueba para rangos alineados de Friedman [50]*

$\hat{R}_i$  es equivalente al rango total del  $i$ -ésimo conjunto de datos y  $\hat{R}_j$  es el rango total del  $j$ -ésimo algoritmo. El estadístico  $T$  es comparado con una distribución  $ji$ -cuadrado con  $k-1$  grados de libertad.

La prueba post hoc se calcula a partir de la **Ecuación 4**.

$$z = (\hat{R}_i - \hat{R}_j) / \sqrt{\frac{k(k + 1)}{6n}}$$

*Ecuación 4. Estadístico de prueba para post hoc de FART [50]*

Los valores de  $z$  en las dos pruebas post hoc, corresponden con una distribución normal estándar y se comparan cada uno bajo un nivel de significancia  $\alpha$  dado.

## 3 Estado del Arte

### 3.1 Métodos de selección de atributos

A continuación, se presentan en mayor detalle y en orden cronológico, algunos métodos y aplicaciones recientes sobre selección de atributos, se da especial énfasis en los más utilizados en la literatura que son los filtros basados en entropía y los wrappers basados en metaheurísticas.

#### 3.1.1 Filtros basados en entropía

En el 2015, Bennasar *et al.* [52] plantearon dos métodos de selección de atributos a los que denominaron Maximización Conjunta de la Información Mutua y Maximización Conjunta de la Información Mutua Normalizada, las propuestas se enfocaron en mejorar algunos aspectos de los filtros tradicionales ante las desventajas de no tomar en cuenta la correlación entre las variables de entrada, la inclusión de variables irrelevantes a medida que se tienen conjuntos de datos de alta dimensionalidad y el problema de descartar variables solo con base en redundancia sin tener en cuenta el aporte a la variable de salida. La propuesta consistió en utilizar un algoritmo hacia adelante (forward) para seleccionar conjuntos de variables de acuerdo al indicador de información mutua conjunta [53] y posteriormente ir agregando las variables de acuerdo al criterio del máximo del mínimo [54]. Los resultados mostraron que los nuevos algoritmos tienen la habilidad de seleccionar subconjuntos de variables con un alto poder discriminativo además de ser más estables.

En 2017 Wang *et al.* [55], propusieron un nuevo termino basado en ganancia de información para selección de atributos denominado Información de Clasificación Independiente (ICI), el cual incorpora dos componentes; uno de ellos es la información nueva que aporta el conjunto de predictores para mejorar la clasificación del modelo, y el otro una medida de redundancia dentro del conjunto de predictores seleccionados. En este trabajo, se demuestra que la maximización del ICI favorece la búsqueda de un subconjunto de variables con un balance entre ambos aspectos. Los experimentos demostraron una buena efectividad del nuevo filtro dado que logra incrementar la capacidad predictiva de los clasificadores, mientras reduce la redundancia en las variables de entrada.

En el 2017 Liu *et al.* [56], desarrollaron una propuesta utilizando la noción de vecindad para construir un selector de variables basado en la calidad de la información. La premisa empleada considera que muestras con valores similares en las variables de entrada deben ser clasificadas bajo la misma clase dado que provienen de la misma vecindad. Posteriormente, se propuso una medida de discriminación para las muestras, dicha medida fue basada en el vecino más cercano el cual es definido empleando el concepto de entropía. Por último, se utilizó un algoritmo hacia adelante para encontrar el conjunto de variables de entrada que optimicen la medida propuesta. Se utilizaron los clasificadores, Máquina de Soporte Vectorial (Support Vector Machine, SVM), Árboles de Clasificación y Regresión (Classification And Regression Trees, CART) y  $K$  vecinos más cercanos ( $K$  nearest neighbor, KNN), sobre seis conjuntos de datos extraídos del repositorio de la UCI [57]. Los resultados experimentales mostraron que el método propuesto es más efectivo en términos de clasificación, estabilidad y escalabilidad que otros algoritmos populares para selección de atributos.

### 3.1.2 Wrappers basados en metaheurísticas

Las metaheurísticas han sido ampliamente utilizadas en la tarea de selección de atributos, y entre ellos los métodos **basados en evolución**, cuyo enfoque es inspirado en el proceso de cambio que tienen las especies como resultado de la continua evolución en el tiempo, haciendo énfasis en el hecho de que las especies recientes se adaptaron mejor a los cambios generados en el ambiente en comparación con las más antiguas. Abd-Alsabour realizó una revisión de métodos de evolución que han sido implementados en selección de atributos [58]. Esta estrategia parte de un conjunto de soluciones que se conoce como población inicial a las cuales en posteriores iteraciones se les optimiza por medio de operadores que simulan procesos de evolución o interacción observados en la naturaleza. La evaluación de cada solución candidata se define a partir de una función de aptitud y la búsqueda de mejores soluciones se detiene cuando se alcanza un criterio de parada. En general, las metaheurísticas tienen varios parámetros los cuales impactan la calidad de los resultados finales. En esta investigación se profundiza en el estado del arte de los algoritmos genéticos (GA) y optimización por enjambre de partículas (PSO), dado que en la literatura son ampliamente utilizados para la selección de atributos, al final también se nombran algunas propuestas basadas en otras metaheurísticas.

## Algoritmos genéticos

Un algoritmo genético (Genetic Algorithm, GA) es un algoritmo que ha sido inspirado en la teoría de evolución natural de Charles Darwin [59], el método comienza con una población de soluciones candidatas que son llamadas cromosomas las cuales siguen un proceso iterativo de selección y modificación, evolucionando hacia una mejor solución la cual es continuamente evaluada por medio de una función de aptitud [60]. Su adaptación para el proceso de selección de atributos consiste en representar las soluciones en secuencias de números binarios indicando 0 para la ausencia de una variable y 1 para la presencia de la misma; cada solución candidata representa un subconjunto de variables y es llamada cromosoma; durante cada iteración a la población se le aplica un conjunto de operaciones de selección, cruce, mutación y reemplazo, con el propósito de mejorar la calidad de las soluciones en términos de la función de aptitud que normalmente es expresada en el caso de wrappers por un índice de calidad de predicción (exactitud, recuerdo, concordancia, error cuadrático medio u otras) [61]–[63].

El proceso inicia generando una población aleatoria de cromosomas los cuales son luego evaluados por la función de aptitud. Posteriormente, en cada iteración, los individuos de la población con mejor aptitud son frecuentemente seleccionados para reproducirse, dicha reproducción es realizada por una estrategia de cruce, donde se combinan partes de cada solución, originando nuevos cromosomas (hijos). De allí se realizan mutaciones a algunos hijos, realizando pequeñas modificaciones aleatorias a algunos componentes de éstos, finalmente se reemplazan los padres con menor aptitud por los hijos que tienen mejor aptitud. El proceso se realiza hasta alcanzar una solución deseada o hasta que se hayan realizado un número determinado de iteraciones [59].

En el 2014 Anirudha *et al.* [61], implementaron un híbrido que combina clustering, GA y un clasificador tal como Árboles de decisión, Naive bayes (NB), KNN o SVM, para crear un modelo predictivo de enfermedades. Los conjuntos de datos empleados para realizar las pruebas fueron Type-2, Diabetes, Pima e Indians. La integración del GA fue determinante para seleccionar las variables relevantes y además aumentar el poder de clasificación, el análisis clúster permitió remover puntos atípicos del conjunto de datos inicial, los resultados incluso superaron al modelo tradicional de predicción de enfermedades.

En el 2015 Soufan *et al.* [64], desarrollaron una herramienta web para selección de atributos en la que integran algunos algoritmos tipo filtro como parte de una selección previa en el preprocesamiento de MAS, y posteriormente aplican un GA en paralelo, el cual permite una ejecución basada en un flujo de trabajo de maestro/esclavo, donde el maestro almacena y aplica las operaciones del GA mientras los esclavos son los encargados de evaluar las funciones de aptitud. Los resultados mostraron superar a otros algoritmos del estado del arte, además generaron soluciones en tiempo récord.

En el 2016 Ghareb *et al.* [65], realizaron modificaciones sobre algunos operadores del GA, el cual fue implementado con el propósito de categorizar textos. Su propuesta consistió en utilizar las frecuencias de los términos y documentos de cada variable para efectuar el cruce de los cromosomas, y una mutación basada en el desempeño obtenido por los padres en cada atributo candidato a ser actualizado, la ventaja fue que desde esta perspectiva ambas operaciones se realizan con base en información útil y no en probabilidades y modificaciones al azar como lo hace un GA en su versión clásica, los clasificadores utilizados fueron NB y Clasificación Asociativa. Adicionalmente en el flujo de trabajo que ellos proponen como un paso previo al wrapper, está un preprocesamiento basado en seis filtros que permiten descartar variables que no aporten información desde el comienzo. Al comparar los resultados con filtros únicos y el GA clásico encontraron que el enfoque propuesto es capaz de producir modelos con un menor número de variables sin producir pérdidas de precisión.

En 2017 Das *et al.* [66], propusieron un ensamble basado en selección de atributos al cual se le adaptó una función bi-objetivo sobre un GA donde la primera función es construida bajo una implementación de la teoría de conjuntos aproximados (Rough Set Theory) [67], [68] y la segunda función utilizando una medida multivariada de información mutua [69]. En el algoritmo propuesto, del conjunto de datos inicial se extraen varios subconjuntos utilizando muestreo con remplazo y posteriormente a cada uno se le aplica la selección de atributos bi-objetivo basada en GA, cada muestra es realizada independientemente y en paralelo. Finalmente se encuentra la mejor solución no dominada por medio de una estrategia de agregación que pretende comparar el desempeño de cada par de soluciones obtenidas por cada subconjunto de datos. Este método mostró eficacia superando resultados del estado del arte sobre selección de atributos en pruebas realizadas sobre algunos conjuntos de datos extraídos del repositorio de la UCI y sobre siete clasificadores.

### Optimización por enjambre de partículas

Otra metaheurística destacada en wrappers es la **optimización por enjambre de partículas** (PSO), en este caso las soluciones candidatas llamadas partículas, tienen las mismas características de las soluciones en un GA respecto a la secuencia binaria que representa la presencia y ausencia de variables, luego a cada solución se le asigna una velocidad con el fin de definir en cada iteración la magnitud del cambio producido en la solución candidata. Cada solución se desplaza por el espacio de búsqueda de acuerdo a su posición actual y su velocidad pero además se influncia por: (i) la información de la mejor posición visitada en iteraciones anteriores, (ii) la posición de la mejor partícula cercana o afín o un grupo de ellas y (iii) la posición de la mejor solución en todo el enjambre [70], [71].

La aplicación de PSO sobre el problema de selección de atributos comienza desde 1997, cuando Kennedy y Eberhary [72] plantearon por primera vez una versión discreta binaria para el algoritmo, esto debido a que en su origen había sido desarrollado principalmente para problemas en espacio continuo. Su propuesta consistía en utilizar una función sigmoideal para transformar los componentes del vector de velocidad en rangos de 0 a 1, y de allí utilizarla como probabilidades para deducir la posición binaria de la partícula, dicha modificación dio paso a que PSO fuera implementado para el problema de selección de atributos. De allí en adelante muchos investigadores propusieron variantes, siendo capaces de obtener mejores resultados que con la versión clásica del mismo en este campo [73]–[76].

En 2014, Xue *et al.* [74] evaluaron la capacidad del algoritmo PSO, para selección de atributos sobre modelos de clasificación, en dichas pruebas utilizaron primero una función de aptitud para evaluar la calidad de clasificación y luego una función multiobjetivo que además como segundo propósito minimizara el número de variables de la solución candidata, la tarea de búsqueda se realiza con base el frente de Pareto para soluciones no dominadas. Los resultados mostraron que la propuesta logra resultados en un tiempo récord que superaron el estado del arte de algoritmos de evolución multiobjetivo sobre selección de atributos en una comparación realizada sobre 12 conjuntos de datos extraídos del repositorio de la UCI [57] y utilizando el clasificado KNN.

En 2015, Ghamisi y Benediktsson [77], propusieron un algoritmo para selección de atributos basados en un híbrido entre en PSO y GA, la unión de ambos algoritmos consiste en integrar las reglas estándar de velocidad y actualización de PSO con

selección, cruce y mutación del GA. El nuevo enfoque permitió automáticamente seleccionar las variables más informativas en términos de clasificación y además en un tiempo de ejecución aceptable, el algoritmo fue implementado sobre el clasificador SVM en dos aplicaciones, uno sobre el conjunto de datos hiperespectrales de pino indio y otro de reconocimiento de rutas desde imágenes.

En 2016, Moradi y Gholampor [75], plantearon un algoritmo que integra a PSO con una estrategia de búsqueda local, su propuesta emplea el índice de correlación de Pearson para guiar la búsqueda del PSO, dándole una ventaja para seleccionar soluciones no correlacionadas. La idea consistió en que en cada nueva posición de una partícula evaluada por el PSO se realizará una comparación entre un grupo de variables con baja correlación y otro grupo de variables con alta correlación, dando prioridad a las menos correlacionadas. En la experimentación se realizó una comparación con métodos del estado del arte los cuales involucran tanto filtros como wrappers, además para las pruebas utilizaron ocho conjuntos de datos tanto de baja dimensión como de alta dimensión y para realizar las respectivas comparaciones se utilizó las pruebas estadísticas *t* student (paramétrica) y Friedman (no paramétrica). Los indicadores de calidad utilizados fueron exactitud de la clasificación, tamaño del conjunto de datos y tiempo de ejecución. Las pruebas reflejaron que los resultados por el algoritmo propuesto son significativamente superiores a los encontrados en el estado del arte.

En 2017, Zhang *et al.* [78], desarrollaron una propuesta para selección de atributos basada en PSO pero con un enfoque multiobjetivo donde no solo se buscaba un subconjunto de variables con la mejor calidad de clasificación, sino que también se tomaba en cuenta el costo de obtener cada variable (tiempo, inversión o esfuerzo). Para encontrar las soluciones no dominadas se utilizó el frente de Pareto, y se creó un archivo externo para almacenar dichas soluciones las cuales una vez lleno el archivo son actualizadas con base en una distancia de apiñamiento (crowding distance) [79]. Por otro lado, en la implementación del algoritmo también se le agregó algunas modificaciones para darle fuerza a la exploración y evitar que se quedara atascado en óptimos locales, la primera consistía en permitir una re-inicialización del 10% de las velocidades de las partículas en cada generación, la segunda en permitir que algunas partículas tuvieran una mutación basada en una probabilidad asignada previamente. Las pruebas de comparación se realizaron utilizando el clasificador de los *k* vecinos más cercanos (*k* nearest neighbor) y cinco (5) conjuntos de datos de prueba. Los resultados evidenciaron que el nuevo

clasificador es altamente competitivo con otros métodos multiobjetivo usados para el mismo propósito.

En general las estrategias basadas en metaheurísticas normalmente implican un gran número de parámetros [80], como es el número de soluciones candidatas evaluadas en cada iteración, en GA las probabilidades de mutación o cruce [59], en PSO la máxima y mínima velocidad permitida y constantes de aceleración, y dichos parámetros pueden influenciar la calidad de la solución obtenida [11], por tal razón las implementaciones requieren un proceso de afinamiento de parámetros, demandando tiempo de computo previo a realizar el proceso de selección de atributos definitivo.

### Otras metaheurísticas

Existen otros métodos principalmente basados en población que han sido recientemente utilizados como criterios de búsqueda en wrappers. En el 2013, Nakamura *et al.* [81], realizaron una propuesta basada en una versión binaria del *algoritmo de murciélagos* el cual es inspirado en la capacidad que tienen estos mamíferos de detectar obstáculos y encontrar alimentos a partir de gritos cortos y pulsos pequeño de sonido. Este algoritmo es de la misma línea de los enjambres por lo tanto también utiliza los conceptos de velocidad y posición como el PSO. En el mismo año Chen *et al.* [82] utilizaron para selección de atributos en imágenes, el algoritmo de *optimización por colonias de hormigas*, el cual se basa en las habilidades de las colonias para encontrar comida utilizando principios para identificar la ruta más corta y logrando sacar ventaja de la comunicación y cooperación entre los individuos. La implementación consiste en el recorrido de hormigas sobre un gráfico compuesto por nodos que representan las variables y están conectados por bordes que definen las posibles trayectorias. Las hormigas dejan unas feromonas indicando las soluciones más promisorias, pero estas se desvanecen con el tiempo dando la posibilidad de nuevas exploraciones.

En el 2016 Das *et al.* [83], implementaron un wrapper utilizando el *algoritmo de búsqueda armónica*, el cual es adaptado del proceso en el que los músicos hacen improvisación musical cuando prueban distintos tonos de instrumentos con el propósito de encontrar una buena armonía. El caso de estudio fue el reconocimiento holístico de palabras en el idioma oficial de Bangladés (Bengalí) a partir de figuras, los resultados mostraron una mejora en la exactitud la cual justifica la utilización del método de selección de atributos a través de esta estrategia. Por otro lado Ghanem

and Jantan [84] en el mismo año propusieron para el problema de detección de intrusos [85], un wrapper cuya estrategia de búsqueda utiliza *enjambre de abejas*, la cual es una metaheurística basada en la tarea de recolección de néctar de las abejas, en dicha tarea se buscan fuentes y cantidades suficiente de alimento y en la colmena se distribuyen las tareas entre: quienes encuentran y traen el néctar, quienes evalúan la calidad y quienes se dedican solamente a explorar nuevas fuentes. La función de aptitud utilizada se construyó con base en tres objetivos: reducir el error de clasificación, el número de variables y la tasa de falsas alarmas.

Para sintetizar las propuestas de wrappers que fueron tenidas en cuenta en el estado del arte, la **Tabla 2** muestra el resumen de cada una de ellas, y pone en evidencia las ventajas y desventajas.

Tabla 2. Resumen del estado del arte de los wrappers

Enfoque	Año	Referencia	Propuesta / (Ventajas y Desventajas)
Basados en entropía	2015	Bennasar <i>et al.</i> [52]	<p><b>Propuesta:</b> Maximización Conjunta de la Información Mutua y Maximización Conjunta de la Información Mutua Normalizada.</p> <p><b>Ventajas:</b> En los test realizados esta propuesta mostró el mejor compromiso entre estabilidad y exactitud, además al igual que todas las propuestas basadas en entropía, por ser filtros son más rápidos y menos complejos que los wrappers sin alcanzar a ser más efectivos, pero útiles para problemas de alta dimensionalidad.</p> <p><b>Desventajas:</b> No tiene en cuenta la interacción entre las variables de entrada y el clasificador, ni la información conjunta entre más de dos variables de entrada.</p>
	2017	Wang <i>et al.</i> [55]	<p><b>Propuesta:</b> Información de Clasificación Independiente.</p> <p><b>Ventajas:</b> Permite encontrar subconjuntos de variables de entrada con alta capacidad predictiva al mismo tiempo que conserva variables con menor redundancia, por lo tanto, es de gran utilidad en conjuntos de datos de alta dimensionalidad.</p> <p><b>Desventajas:</b> No tiene en cuenta las interacciones entre variables de entrada.</p>
	2017	Liu <i>et al.</i> [56]	<p><b>Propuesta:</b> Una propuesta utilizando la noción de vecindad para construir un selector de variables basado en la calidad de la información.</p>

Enfoque	Año	Referencia	Propuesta / (Ventajas y Desventajas)
Algoritmos genéticos			<p><b>Ventajas:</b> Las pruebas de comparación con otros algoritmos mostraron que es competitiva en escalabilidad al ser implementada sobre datasets de distintos dominios, además de sobresalir en la estabilidad de los resultados.</p> <p><b>Desventajas:</b> No tiene en cuenta las interacciones entre variables de entrada.</p>
	2014	Anirudha <i>et al.</i> [61]	<p><b>Propuesta:</b> Híbrido, entre clustering, GA y un clasificador (NB, KNN, SVM).</p> <p><b>Ventajas:</b> El paso del clustering ayuda a remover puntos atípicos, reduciendo el ruido del conjunto de datos.</p> <p><b>Desventajas:</b> Se puede volver costoso en la medida que se tenga un dataset de alta dimensionalidad, y al igual que en todos los wrapper basados en GA se deben ajustar parámetros de optimización.</p>
	2015	Soufan <i>et al.</i> [64]	<p><b>Propuesta:</b> Híbrido entre selección de atributos tipo filtro y GA (propuesta de ejecución en paralelo).</p> <p><b>Ventajas:</b> La implementación de filtros previos al wrapper ayuda a descartar rápidamente variables irrelevantes, y el diseño propuesto en paralelo genera resultados en mejor tiempo.</p> <p><b>Desventajas:</b> Requiere un sistema con características adecuadas para lograr la eficiencia computacional.</p>
	2016	Ghareb <i>et al.</i> [65]	<p><b>Propuesta:</b> Modificaciones sobre operadores de cruce y mutación a partir de probabilidades que dependan de las frecuencias de términos en la categorización de textos.</p> <p><b>Ventajas:</b> Las propabilidades ajustadas de cruce y mutación son asignadas de acuerdo a información útil y no de manera aleatoria, guiando mejor la búsqueda.</p> <p><b>Desventajas:</b> Según el artículo, la propuesta se adapta bien cuando las variables de entrada son discretas, y se pueden llevar a una escala de probabilidad que tenga sentido, sí las variables son de otra escala, por ejemplo, numérica, desarrollar una implementación similar puede ser inviable, esto debido a que el algoritmo propuesto es desarrollado principalmente utilizando frecuencias de palabras en textos.</p>
	2017	Das <i>et al.</i> [66]	<p><b>Propuesta:</b> Ensamble de submuestras con remplazo y una una función bi-objetivo sobre GA que involucra teoría de conjuntos</p>

Enfoque	Año	Referencia	Propuesta / (Ventajas y Desventajas)
Optimización de enjambre de partículas			<p>aproximados y una medida multivariada de información mutua.</p> <p><b>Ventajas:</b> El ensamble de submuestras y la función bi-objetivo permiten encontrar resultados eficientes y robustos.</p> <p><b>Desventajas:</b> Requiere suficientes recursos computacionales para ejecutar el algoritmo en paralelo, sino, puede llevar largos tiempos de ejecución en la medida que se utilizan datasets con muchas observaciones y variables.</p>
	2014	Xue <i>et al.</i> [74]	<p><b>Propuesta:</b> Función multiobjetivo utilizando PSO para encontrar subconjuntos con el mejor desempeño y a la vez el menor número de variables seleccionadas.</p> <p><b>Ventajas:</b> El planteamiento de la función multiobjetivo logra encontrar mejores soluciones en comparación a versiones clásicas de un solo objetivo sobre PSO.</p> <p><b>Desventajas:</b> Aún se requieren más pruebas para probar efectividad en datasets de alta dimensionalidad. Además al igual que en GA, en las propuestas relacionadas con PSO, se deben ajustar parámetros de optimización.</p>
	2015	Ghamisi y Benediktsson [77]	<p><b>Propuesta:</b> Integración de GA y PSO para selección de atributos usando SVM como clasificador.</p> <p><b>Ventajas:</b> Combinar ambas estrategias de búsqueda permitieron lograr resultados en tiempo record, y la adaptación de SVM como MAS, permite lidiar con problemas de alta dimensionalidad.</p> <p><b>Desventajas:</b> Pese que se hacen pruebas para evaluar la dependencia del dataset sobre los parámetros del GA y PSO, y se concluye que los parámetros óptimos fueron siempre parecidos para los dos casos evaluados, las pruebas solo se realizaron con ejemplos de detección de patrones en imágenes. Nuevos datasets de otros dominios pueden requerir test previos para identificar los mejores parámetros convirtiéndolo en un proceso más tedioso en comparación a otros wrappers.</p>
	2016	Moradi y Gholampor [75]	<p><b>Propuesta:</b> Híbrido entre PSO y búsqueda local basada en el coeficiente de correlación de Pearson u otra medida de dependencia.</p> <p><b>Ventajas:</b> Funciona muy bien para datasets que contienen variables correlacionadas, logrando equilibrar la influencia que tienen</p>

Enfoque	Año	Referencia	Propuesta / (Ventajas y Desventajas)
Otras metaheurísticas			dichas variables de entrada sobre la solución final. <b>Desventajas:</b> En caso de que el dataset esté compuesto por variables mixtas y además, haya combinación de variables numéricas y categóricas no ordinales, se debe seleccionar muy cuidadosamente la medida de correlación a utilizar.
	2017	Zhang <i>et al.</i> [78]	<b>Propuesta:</b> PSO basado en una función multiobjetivo que involucra el desempeño del MAS y una función de costo de medición de las variables de entrada (Tiempo, inversión o esfuerzo) <b>Ventajas:</b> La propuesta permite incluir más información fuera de los objetivos comunes en selección de atributos, lo cual puede ser más pertinente cuando existen variables de entrada de alto costo de medición. <b>Desventajas:</b> La solución no necesariamente es la mejor en términos de desempeño, dado que también se le da prioridad al costo en la solución final.
	2013	Nakamura <i>et al.</i> [81]	<b>Propuesta:</b> Búsqueda basada en el algoritmo de murciélagos. <b>Ventajas:</b> Los pocos test realizados mostraron que este es un método sobresaliente para datasets de alta dimensionalidad. <b>Desventajas:</b> Al igual que todas las metaheurísticas revisadas que son parte del estado del arte, requiere optimización de parámetros, ya que estos pueden cambiar dependiendo de cada dataset.
	2013	Chen <i>et al.</i> [82]	<b>Propuesta:</b> Búsqueda basada en optimización por colonias de hormigas. <b>Ventajas:</b> Los resultados mostraron efectividad en precisión, y velocidad de procesamiento para datasets con pocas variables comparados con otros wrappers. <b>Desventajas:</b> Según el artículo, aún la propuesta no es efectiva en datasets de alta dimensionalidad.
	2016	Das <i>et al.</i> [83]	<b>Propuesta:</b> Búsqueda harmónica, basada en el proceso de componer música. <b>Ventajas:</b> Los resultados mostraron ser eficientes para el reconocimiento de palabras a partir de imágenes, esto en términos de exactitud.

Enfoque	Año	Referencia	Propuesta / (Ventajas y Desventajas)
			<p><b>Desventajas:</b> Se debe aún explorar el potencial o escalabilidad en otros campos.</p> <p><b>Propuesta:</b> Búsqueda basada en enjambre de abejas con una función de aptitud multiobjetivo para: la exactitud, número de atributos y tasa de falsas alarmas.</p>
	2016	Ghanem and Jantan [84]	<p><b>Ventajas:</b> Los resultados mostraron ser eficientes para el problema de detección de intrusos en los objetivos evaluados en la función de aptitud.</p> <p><b>Desventajas:</b> Se debe aún explorar el potencial o escalabilidad en otros campos.</p>

### 3.2 Medidas de importancia de variables

Entre los métodos más conocidos para estimar la importancia de variables, está la medida que se obtiene del modelo Random Forest (RF) introducido por Breiman en el 2001 [86], el cual es construido a partir del entrenamiento de varios árboles de clasificación o regresión utilizando muestras independientes de las observaciones. Cada muestra con la que se construye cada árbol se selecciona utilizando una estrategia de bootstrap usualmente con remplazo y debido al uso de dicha estrategia de muestreo, siempre queda un conjunto de observaciones denominadas fuera de la bolsa (out-of-bagging, OOB) que no hacen parte del entrenamiento de cada árbol. Un aspecto adicional, es que en la construcción de cada árbol no participan todas las variables de entrada candidatas a predecir en el modelo, sino que un subconjunto de ellas de tamaño ( $mtry$ ) se selecciona al azar.

Para estimar una medida de importancia de variables para el RF, se utilizan distintas métricas, la más conocida es el coeficiente GINI la cual se basa en la reducción del índice de impureza (incertidumbre) sobre todos los nodos en los que aparece cada una de las variables en los distintos árboles, el concepto sugiere que las variables que más reducen la impureza son las más importantes en la construcción del modelo. También existe otra métrica denominada importancia de variable por permutación que consiste en tomar las observaciones obtenidas del OOB dentro de cada árbol y calcular la medida de desempeño de la predicción, ya sea MSE en regresión o la tasa de error en clasificación, posteriormente para calcular la medida de importancia de cada variable se permutan sus valores aleatoriamente y se calcula nuevamente la medida de desempeño dada esta variación, esto con el fin de obtener la diferencia de ambos desempeños en el OOB (original y variable

permutada), luego las diferencias de todos los árboles son promediadas obteniendo una medida de importancia para la variable evaluada. Con esto se esperaría que la variable que presente cambios más altos después de ser alterada tenga mayor relevancia. Esta medida se puede interpretar como la contribución de cada variable al desempeño del modelo tomando en cuenta la interacción con otras variables.

En 2008, Strobl *et al.* [87], demostraron que el método clásico para medir importancia de variables por RF tiene un sesgo dado que tiende a sobrestimar la importancia de conjuntos de variables correlacionadas, lo cual es causado debido a la preferencia que le dan los árboles en los nodos superiores a esta variables, por lo tanto propusieron una variación a la forma en que se calcula la importancia de variables por permutación que consiste en emplear una medida basada en permutación condicional, la cual considera la estructura de correlación que tienen los predictores, permitiendo estimar una importancia marginal, el tipo de árboles utilizados en el entrenamiento son árboles de inferencia condicional [7], [88]. La nueva medida mostró el verdadero impacto de cada variable sin dejarse influenciar por la correlación con otras variables de entrada, no obstante, el algoritmo requiere una mayor carga computacional en la medida que se tienen un elevado número de observaciones y variables.

En 2015 Giam *et al.* [89], propusieron una métrica para medir la importancia de variables, la cual fue basada en permutación siendo muy similar a la de Random Forest pero implementada para modelos de redes neuronales artificiales. La propuesta se basa en el concepto de que una variable permutada en un modelo de predicción rompería la asociación con la variable de respuesta sí esta asociación es existente, y por lo tanto ocasionaría una reducción en el desempeño del modelo estimado con el coeficiente de determinación ( $R^2$ ). Por lo tanto, la idea consiste en permutar cada variable de entrada y calcular la diferencia de  $R^2$  obtenido de una predicción con la variable permutada versus una predicción con la variable intacta. Las pruebas se realizaron sobre regresiones simuladas donde se involucró una prueba para relaciones lineales, otra prueba para relaciones cuadráticas y otra prueba para términos de interacción. Los resultados mostraron una mayor exactitud de la técnica propuesta comparada con otras medidas tales como producto de la conexión de pesos y producto de pesos estandarizados también empleadas para este propósito.

En 2016 Yun *et al.* [90], desarrollaron una estrategia en la que propusieron agregar nueve medidas de importancia de variables en las que involucran test estadísticos,

modelos de ajuste y modelos predictivos para construir un único ranking que defina un solo orden de importancia de variables de entrada. Su propuesta la desarrollaron con el objetivo de resolver el problema de obtener diferencias entre distintas medidas, ya que cada una se basa en un criterio distinto y actúa como un experto independiente. En el proceso se transforman los valores de cada una de las nueve medidas en rangos ordenados y posteriormente se define un nuevo rango que simultáneamente esté lo más próximo posible a dichos rangos construidos. Esta tarea se lleva a partir de una función de minimización que utiliza la distancia de Spearman footrule [91] para calcular distancias entre listas ordenadas y entropía cruzada para encontrar la solución al problema de minimización [92]. Este proceso fue implementado en el problema de descubrimiento de biomarcadores en metabolómica en un modelo de análisis discriminante de mínimos cuadrados parciales, los resultados mostraron que la nueva medida basada en agregación alcanza un mejor desempeño en predicción comparada con métodos basados en penalización como LASSO.

En 2017 Akarachantachote *et al.* [93], plantearon una medida de importancia de variables basados en un modelo supervisado utilizando mínimos cuadrados parciales (Partial Least Squares, PLS) y la elección de un subconjunto óptimo tomando como referencia el punto de corte en un boxplot. El método PLS permite llevar a un nuevo espacio más pequeño el conjunto de variables de entrada original con el propósito de reducir su dimensión mientras mantiene la mayor cantidad de información posible en el nuevo espacio, en dicho proceso se crean unos vectores de peso que relacionan cada variable de entrada con cada componente del nuevo espacio. La nueva propuesta de los autores consiste en calcular la importancia de variables utilizando la capacidad predictiva de cada componente para explicar la variable de salida (medida por el coeficiente de determinación  $R^2$ ) y ponderarla por los pesos que tiene cada variable con cada componente. La selección de variables óptimas se hace con base en la estimación de la importancia de variables propuesta sobre una matriz combinada por las variables originales de entrada y una matriz del mismo tamaño, pero permutada aleatoriamente (matriz de ruido). Las variables de entrada del conjunto de datos original que superen el tercer cuartil más 1.5 el rango intercuartílico serán el subconjunto óptimo. Las pruebas se realizaron a través de un conjunto de datos simulado y la nueva propuesta mostró ser eficiente en la identificación de variables informativas.

Para sintetizar las propuestas de importancia de variables que fueron tenidas en cuenta en el estado del arte, la **Tabla 3** muestra el resumen de cada una de ellas, y pone en evidencia las ventajas y desventajas.

Tabla 3. Resumen del estado del arte de la importancia de variables

Enfoque	Año	Referencia	Propuesta / (Ventajas y Desventajas)
Importancia de variables	2001	Breiman [86]	<p><b>Propuesta:</b> Importancia de variables basada en el algoritmo RF a partir del índice de GINI y Permutación.</p> <p><b>Ventajas:</b> Bajo costo computacional en comparación con otras métricas dado que está integrada en RF y se puede paralelizar.</p> <p><b>Desventajas:</b> Es dependiente de RF entonces no aplica para otros MAS, y tiende a favorecer predictores correlacionados.</p>
	2008	Strobl <i>et al.</i> [87]	<p><b>Propuesta:</b> Importancia de variables calculada para el algoritmo de RF pero basada en árboles condicionales y permutación condicional.</p> <p><b>Ventajas:</b> Es insegada a variables correlacionadas en comparación a las propuestas por Breiman.</p> <p><b>Desventajas:</b> Alto costo computacional debido a la construcción de árboles condicionales y es dependiente de RF condicional por tanto no aplica para otros MAS.</p>
	2015	Giam <i>et al.</i> [89]	<p><b>Propuesta:</b> Importancia de variables para redes neuronales basada en permutación.</p> <p><b>Ventajas:</b> En comparación a otras métricas empleadas para redes neuronales, esta propuesta mostró ser robusta en datasets en los que hay: relaciones no lineales, interacciones y pocas observaciones.</p> <p><b>Desventajas:</b> El costo computacional depende de la complejidad de la red neuronal, y la propuesta se realiza específicamente sobre redes neuronales, sin embargo, su concepto se puede extender a otros MAS.</p>
	2016	Yun <i>et al.</i> [90]	<p><b>Propuesta:</b> Agregación de medidas de importancia de variables que involucran distintas métricas en un solo indicador.</p> <p><b>Ventajas:</b> Permite integrar en un solo indicador de importancia de variables distintas métricas usadas para el mismo propósito, actuando como un consenso de expertos lo cual puede resultar en una importancia de variables más confiable.</p>

Enfoque	Año	Referencia	Propuesta / (Ventajas y Desventajas)
			<p><b>Desventajas:</b> Dependiendo de las medidas de importancia de variables empleadas en el consenso puede ser costoso computacionalmente y el éxito dependerá de las métricas que se escojan.</p>
			<p><b>Propuesta:</b> Importancia de variables basada en PLS y el punto del corte del boxplot.</p>
	2017	Akarachantachote et al. [93]	<p><b>Ventajas:</b> Al utilizar PLS, esta medida es especialmente útil cuando se evalúan datasets con alta dimensión, y comparado con las anteriores, este método compuesto permite hallar un punto de corte para identificar el subconjunto de variables más relevantes.</p> <p><b>Desventajas:</b> Es dependiente del PLS, por lo tanto no aplica para otros MAS, además el éxito depende estrechamente de los parámetros del entrenamiento del PLS lo cual puede requerir test adicionales para optimizarlo.</p>



## 4 Selección de atributos utilizando arreglos de cobertura

### 4.1 Algoritmo wrapper utilizando CAs o ICAs

Para el desarrollo del wrapper, los CAs que se utilizaron fueron de alfabeto binario ( $v = 2$ ), por esto cada componente del arreglo solo posee valores  $\{0, 1\}$ . El proceso de selección de atributos con el Wrapper consiste principalmente en construir subconjuntos de variables a partir de las filas de dichos CAs, las cuales son la base que define la estrategia de búsqueda, el número de parámetros  $k$  corresponde con el número total de variables de entrada en el conjunto de datos, y tal como fue nombrado en la sección 2.3 (Arreglos de cobertura), el valor cero (0) indica que la variable no será tomada en cuenta en la solución y uno (1) que la variable estará presente o se incluirá en la solución (ver **Figura 4**).

Formalizando el algoritmo y tomando como referencia cualquier conjunto de datos con una sola variable de salida, sea  $X$  de dimensiones  $n \times p$  la matriz que representa el conjunto de datos evaluado cuyas filas son las observaciones y las columnas variables de entrada. Cada fila de dicha matriz está relacionada con un valor de la variable de salida  $Y$  que es un vector de dimensiones  $n \times 1$  y contiene la clase o categoría a la que pertenece cada observación.  $f$  representa una función de un MAS que predice a  $Y$  con base en los valores de  $X$ , de tal forma, que la diferencia entre valores reales y estimados sea mínima ( $f(X) \approx Y$ ) y la calidad de dicha estimación es evaluada a partir de un indicador de exactitud representado por un valor  $\alpha$ . Luego sea  $C_i$  de dimensiones  $1 \times p$ , la  $i$ -ésima fila de un CA o ICA de fuerza  $t$  con parámetros ( $N: t=t, k=p, v=2$ ).

Al expresar a  $X$  como un conjunto de vectores  $[X_1 X_2 \dots X_j \dots X_p]$ , donde  $X_j$  ( $j=1, 2, \dots, p$ ) representa la  $j$ -ésima variable de entrada y a  $C_i$  una fila de un CA o ICA que puede ser representado mediante un conjunto de escalares  $[C_{i1}, C_{i2}, \dots, C_{ip}]$  donde  $C_{ij} \in \{0, 1\}$ , se utilizan ambas expresiones como argumentos para definir una función llamada *SelectInputSubset*, tal que  $Z_i = \text{SelectInputSubset}(X, C_i)$ , de manera que recibe la matriz de datos  $X$  y el vector binario  $C_i$ , para generar una nueva sub-matriz  $Z$  siguiendo el procedimiento descrito a continuación:

Por cada columna  $X_j$  que representa la  $j$ -ésima variable de la matriz  $X$  y  $C_{ij}$  el  $j$ -ésimo escalar del vector  $C_i$ . (Ambos en la misma posición de columna), se construye la matriz  $Z$  de acuerdo con la siguiente regla:

$$\begin{aligned} \text{Sí } C_{ij} = 0 \text{ entonces } X_j \notin Z_i \\ \text{De otra manera } X_j \in Z_i \end{aligned}$$

De manera general, el wrapper consiste en ejecutar el anterior proceso en el que se evalúa a  $f$  ( $\text{SelectInputSubset}(X, C_i)$ ) =  $f(Z)$  (donde  $i=1,2,\dots,N$ ) mediante  $N$  experimentos, es decir, por todas las filas del CA o ICA, y al final obtener un conjunto de valores de exactitud u otra medida de desempeño dado por  $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_s, \dots, \alpha_N\}$  producto de evaluar cada uno de los MAS. Sí  $\alpha_s$  es el máximo valor encontrado en el vector  $\alpha$  y se ubica en la posición  $s$  entonces las columnas de la matriz  $Z_s$  corresponden con el mejor subconjunto de variables encontrado por el wrapper. A continuación, en el **Algoritmo 1** se presenta paso por paso el procedimiento.

<b>Inputs:</b>	Matriz de variables de entrada $X$ , Variable de respuesta $Y$ , CA o ICA $C$ con $N$ filas y un modelo de aprendizaje supervisado $f$
<b>Condiciones previas:</b>	Remover variables con baja varianza
<b>Salida:</b>	Una matriz $Z$ con un número menor o igual de columnas de $X$ las cuales representan un subconjunto de variables
	<ol style="list-style-type: none"> <li>1. <b>begin</b></li> <li>2.     Evaluar <math>Y = f(X)</math> // Optimizar los parámetros del MAS y calcular el desempeño <math>\alpha</math></li> <li>3.     Mejor = <math>X</math></li> <li>4.     for each row <math>C_i</math> in <math>C</math> do // Es decir <math>N</math> veces</li> <li>5.         <math>Z = \text{selectInputSubset}(X, C_i)</math> // Definir el subconjunto de variables basado en la función <math>\text{selectInputSubset}</math></li> <li>6.         Evaluar <math>Y = f(Z)</math> Optimizar parámetros del MAS y calcular el desempeño <math>\alpha_i</math> para <math>Y</math></li> <li>7.         if <math>\alpha_i &gt; \alpha</math> then Mejor = <math>Z</math>, <math>\alpha = \alpha_i</math> // Siendo '<math>&gt;</math>' (mayor que) un caso de maximización</li> <li>8.     end for</li> <li>9.     return Mejor</li> <li>10. <b>end</b></li> </ol>

Algoritmo 1. Wrapper basado en CAs e ICAs

En una versión corta, el nuevo wrapper puede ser expresado mediante la **Ecuación 5**, la cual ha sido planteada y construida como parte de esta investigación.

$$\begin{aligned} WrapperCA(X, f, C) \\ = \operatorname{argmax}_{z_i} \left( \alpha_i = \operatorname{accuracy}(f(z_i = \operatorname{selectInputSubset}(X, C_i))) \right) \end{aligned}$$

$$WrapperCA(X, f, C) = \operatorname{argmax}_{z_i} \left( \alpha_i = \operatorname{accuracy}(f(z_i)) \right)$$

*Ecuación 5. Wrapper para selección de atributos utilizando filas de un CA.*

La implementación del wrapper fue realizada usando el lenguaje de programación R versión 3.2.1 [94]. Las librerías principales utilizadas en el análisis fueron: snowfall [95] para implementar procesos en paralelo, ggplot2 [96] como herramienta gráfica y caret [97] para entrenamiento y validación de los MAS.

## 4.2 Evaluación de wrapper utilizando CA e ICA

### 4.2.1 Conjuntos de datos para validación

La evaluación del wrapper con CAs e ICAs, se realizó utilizando los conjuntos de validación (datasets) mostrados en la **Tabla 4**. Estos fueron seleccionados por ser los más utilizados en recientes publicaciones realizadas sobre selección de atributos, además por ser distintos en el número de variables y número de clases en la variable de salida, el acceso a esto datos es libre y están disponibles en el repositorio de datasets de la Universidad de California en Irvine ([UCI](https://archive.ics.uci.edu/)) [57].

*Tabla 4. Descripción de los datasets de prueba*

<b>Dataset</b>	<b>No. de variables de entrada</b>	<b>No. de clases</b>	<b>No. de observaciones</b>
Glass	9	6	214
Wine	13	3	178
Zoo	16	7	101
Vehicle	18	4	846
WDBC	30	2	569
Ionosphere	34	2	351
Sonar	60	2	208

### 4.2.2 Modelos de aprendizaje supervisado (MAS) y optimización de parámetros

En la **Tabla 5** se presentan los seis MAS con los que se evaluó el desempeño del wrapper propuesto, estos fueron seleccionados por ser ampliamente utilizados para tareas de predicción en distintos campos de la ciencia. Cada modelo tiene sus parámetros específicos los cuales se optimizaron usando validación cruzada en una configuración de 10 folders con 5 repeticiones. Los parámetros óptimos pueden cambiar dependiendo de la cantidad de variables y la información que dichas variables contienen, por tal razón por cada subconjunto de variables candidatas a ser la solución, se realizó una optimización de parámetros previa a la obtención del MAS definitivo con el que se evalúa finalmente la exactitud. Dicha optimización se hizo por medio del algoritmo greedy presentado en [98], el cual fue implementado sobre SVM, dicho algoritmo consiste en dividir en seis regiones de igual distancia el espacio de los parámetros y luego evaluar los valores correspondientes a los límites de cada región para determinar la configuración más favorable para el MAS. En caso de que el MAS requiera más de un parámetro, tal como el Perceptrón Multicapa (Multilayer Perceptron, MLP) que tiene como parámetros el número de neuronas en la capa oculta y la tasa de decaimiento o SVM con los parámetros Costo y Gamma, se selecciona uno para la optimización y el otro se mantiene con el valor de la configuración por defecto en el software. La librería caret de R facilita este proceso.

*Tabla 5. MAS y sus parámetros correspondientes*

<b>MAS</b>	<b>Descripción</b>	<b>Parámetros</b>
C4.5	Árbol de decisión C4.5	Factor de confianza
KNN	$K$ vecinos más cercanos	Número de vecinos
MLP	Perceptrón Multicapa	Número de neuronas en la capa oculta
NB	Naïve Bayes	Factor para la corrección de Laplace
RF	Random Forest	Número de variables muestreadas aleatoriamente como candidatas en cada división
SVM	Máquinas de vectores soporte	Costo de violación de restricciones y gamma

Para la evaluación del wrapper propuesto se utilizaron clasificadores (C4.5, KNN, MLP, NB, RF y SVM) como MAS, siendo las variable de salida variables categóricas, por tal razón, la medida de desempeño utilizada en las pruebas fue la exactitud

(accuracy) la cual se calcula a partir de una proporción dada entre el número de veces que la predicción del modelo acierta, sobre el número total de predicciones, este valor varía entre 0 siendo el caso en que el MAS no tiene ningún acierto y 1 cuando acertó siempre.

### 4.2.3 CAs e ICAs usados

Los CAs e ICAs que se usaron para la experimentación y análisis fueron suministrados por el laboratorio de Tecnologías de la Información del CINVESTAV (Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional) de Tamaulipas (México) [99]. Se evaluaron CAs e ICAs desde fuerza dos (2) hasta fuerza seis (6), todas con alfabeto binario. Dichas fuerzas fueron analizadas con el propósito de identificar cuál de ellas alcanza mejores resultados en términos de desempeño para el wrapper. Para cada dataset con el que se experimentó requirió un CA o ICA específico, ya que este se define con base en el número de variables del dataset (valor  $k$  del CA o del ICA).

Como se mencionó anteriormente, los ICA tienen la característica de contener las filas de ICAs de fuerza menor, lo cual garantiza que a medida que la fuerza se hace más grande, puede suceder o que se mantiene la mejor solución encontrada hasta el momento o que se encuentra una mejor solución. Con los CA clásicos no sucede lo mismo, debido a que fuerzas mayores, no necesariamente tienen las mismas filas (casos de prueba) de fuerzas inferiores, por lo tanto, el desempeño no necesariamente crece conforme aumenta la fuerza. Para obtener este esquema creciente en exactitud usando CA clásicos, se construyeron y usaron en los experimentos los que se denominaron CAs acumulados (Cumulative Covering Arrays, CCA), para los cuales cada vez que se aumenta la fuerza, se compara con la mejor solución encontrada en los CA de inferior fuerza, y sí el desempeño encontrado con el nuevo CA no es superado, entonces la mejor solución se sigue manteniendo. Esto implica que para evaluar un CCA de fuerza cuatro, se debe ejecutar previamente los CAs de fuerza dos y tres.

La cantidad de pruebas requeridas para evaluar el wrapper propuesto, depende especialmente del número de variables que tenga el dataset y de la fuerza del CA o ICA que se usará, nunca del número de registros en el dataset. La **Figura 6** muestra en el eje horizontal la fuerza del CCA o del ICA (eje  $x$ ) y en el eje vertical el número de casos de prueba (filas) del CCA o del ICA. Se puede observar que el número de casos de prueba crece exponencialmente en la medida en que crece la

fuerza. Además, se puede observar que, en fuerzas menores o iguales a cuatro, para seis de los siete datasets, la cantidad de pruebas que se hacen con CCA están por debajo del ICA, no obstante, en la fuerza seis sucede lo contrario.

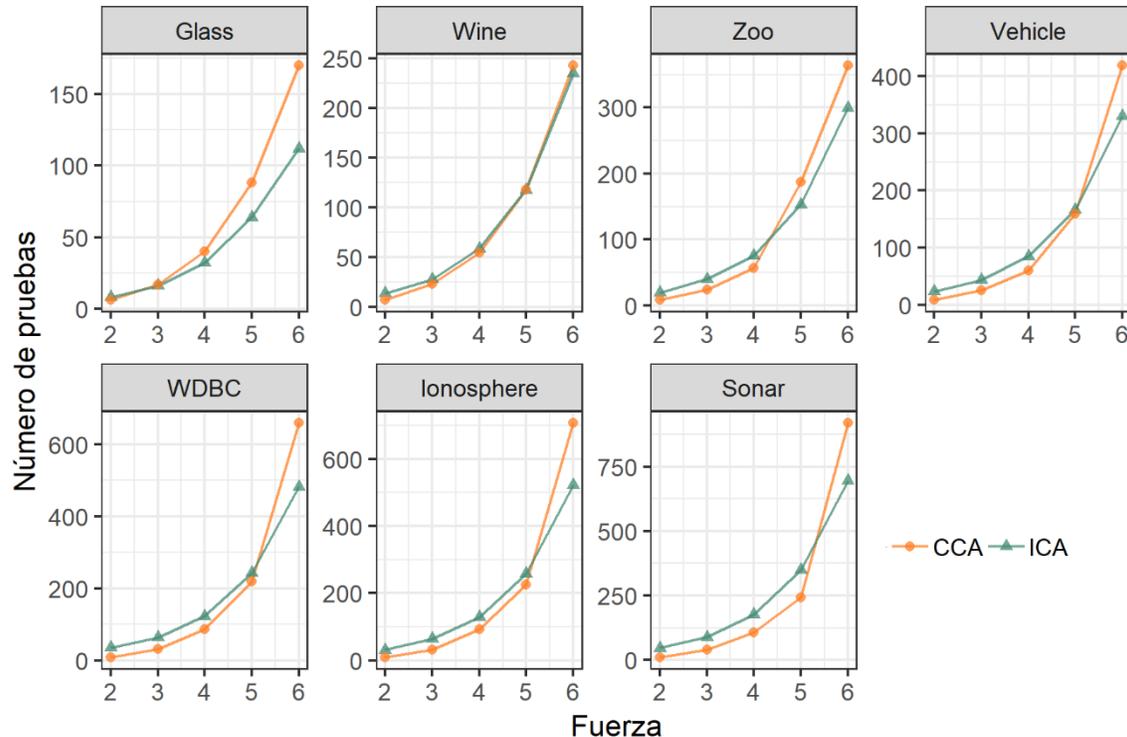


Figura 6. Número de pruebas requeridas por nivel de fuerza en cada dataset

#### 4.2.4 Comparación de los resultados según la fuerza

Para analizar los resultados obtenidos por el wrapper con los dos tipos de CAs, se comparó para cada caso (Dataset-MAS) la exactitud obtenida por el wrapper en cada una de las fuerzas (2-6) contra la exactitud de una línea base (BaseLine, BL) obtenida de un MAS (sin el wrapper) en el que se incluyen todas las variables. Posteriormente se calculó la cantidad de casos que superan dicha línea base en cada una de las fuerzas. Los resultados obtenidos se pueden observar en la **Figura 7**, donde se aprecia que conforme se incrementa la fuerza tanto para CA como CCA se van incrementando el número de casos en los que se encuentran soluciones que superen la línea base, también se observa que para la mayoría de las fuerzas excepto cuatro ( $t=4$ ), los ICA llevan la ventaja y alcanzan en fuerza seis un total de 97.6% de casos que superan la línea base mientras que CCA alcanza un 95.23% en la misma fuerza.

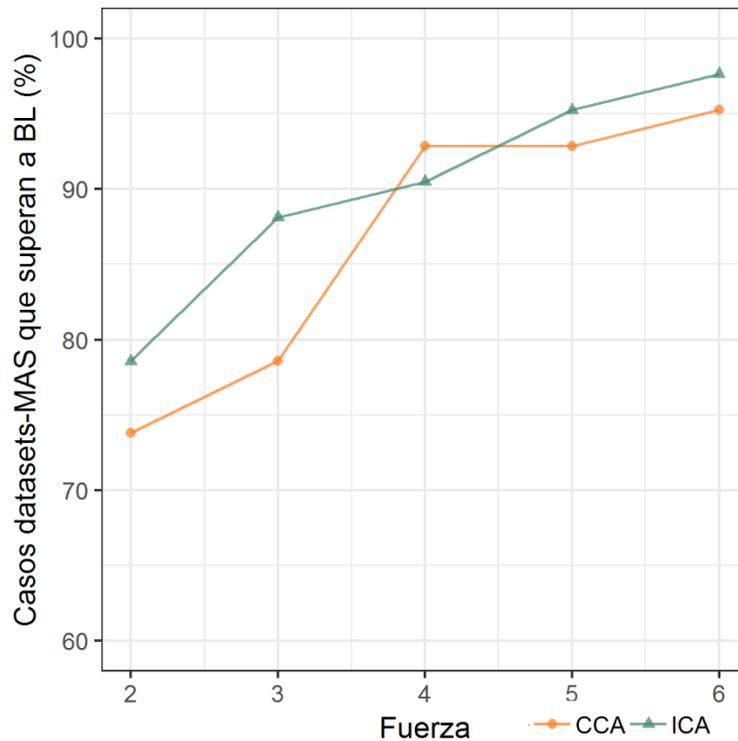


Figura 7. Porcentaje de dataset-MAS que superan a BL según la fuerza del CA.

Adicionalmente, para ambos tipos de CA, los resultados del wrapper sugieren que la mejor fuerza es seis ( $t=6$ ), donde se alcanza una mayor cantidad de casos que exceden la línea base, sin embargo, desde fuerza dos, ya ambos superan más del 70% de los casos. Un hallazgo similar en la efectividad de las fuerzas de CAs fue encontrado en el campo de la ingeniería de software, en [16], donde fue demostrado que las fallas de un sistema software pueden ser detectadas con un buen nivel de precisión a través del uso de CA incluso con fuerzas pequeñas, no obstante en la medida que se incrementa este parámetro los resultados mejoran pero exigen un mayor número de pruebas. En [100], a partir de experimentos realizados con software para dispositivos médicos, servidores, navegadores entre otros, se concluyó que con fuerza seis se alcanza virtualmente el 100% de la detección de los fallas sin tener que recurrir a pruebas exhaustivas, lo cual es consistente con los resultados presentados en la **Figura 7**.

Por otro lado, la **Figura 8** muestra el incremento en exactitud respecto a la línea base para cada caso, pareja dataset-MAS, sobre los diferentes wrappers evaluados. Allí se observa que la exactitud después de la selección de atributos incrementó hasta un 24.2% siendo el caso de Wine con KNN. No obstante, para algunos

datasets como Vehicle los MAS distintos a KNN alcanzan a lo máximo una mejora del 2% de incremento y por lo tanto el wrapper no produce una mejora significativa. Con lo anterior, se observa que el dataset y el MAS influyen tanto en el desempeño de la solución encontrada, como en la magnitud del incremento producido conforme se aumenta la fuerza del CA. A pesar de que en algunos casos se obtienen rendimientos inferiores a la línea base, la mayoría ocurren en niveles de fuerza entre dos ( $t=2$ ) y cinco ( $t=5$ ) pero usualmente se resuelven con fuerza seis ( $t=6$ ).

Por último, se desarrolló una evaluación formal utilizando la prueba de comparación de Friedman explicado en la sección 2.4.1 con el propósito de determinar el mejor CA y la mejor fuerza a través de todos los datasets. En esta prueba se evaluaron dos indicadores, por un lado, la exactitud y por otro lado el número de variables escogidas por el wrapper. En la **Tabla 6** altos valores en exactitud corresponden con los primeros puestos del ranking de Friedman, mientras que selecciones con pocas variables corresponden con los primeros ranking de los resultados de número de variables. Las columnas post hoc muestran los resultados de la prueba posterior a Friedman el cual permite detectar entre que pares de grupos existen diferencias. En la interpretación dos “CA-Fuerza” que no tengan ninguna letra en común sugieren que presentan diferencias significativas. En la prueba se utilizó un nivel de significancia de 0.05. Finalmente, en la **Tabla 6** se observa que CCA-6 es el wrapper que obtiene un accuracy mayor y a la vez un menor número de variables, no obstante los resultados no son significativamente distintos al ICA-6 que se encuentra en el segundo lugar, de manera general queda en evidencia que el incremento de la fuerza de los CAs ocasiona mayor exactitud y menor número de variables sobre los MAS.

Tabla 6. Friedman para la comparación de CA-Fuerza de los wrappers.

CA-Fuerza	Exactitud		Número de variables	
	Ranking	Post Hoc	Ranking	Post Hoc
CCA-6	<b>2.25</b>	a	<b>4.31</b>	a
ICA-6	<b>2.36</b>	a	<b>4.73</b>	ab
ICA-5	3.94	b	5.14	abc
CCA-5	4.01	b	4.99	abc
ICA-4	5.43	c	5.60	bc
CCA-4	5.57	c	5.69	bc
ICA-3	6.82	d	5.70	bc
CCA-3	7.45	d	6.04	cd
ICA-2	8.37	e	5.62	bc
CCA-2	8.80	e	7.19	d

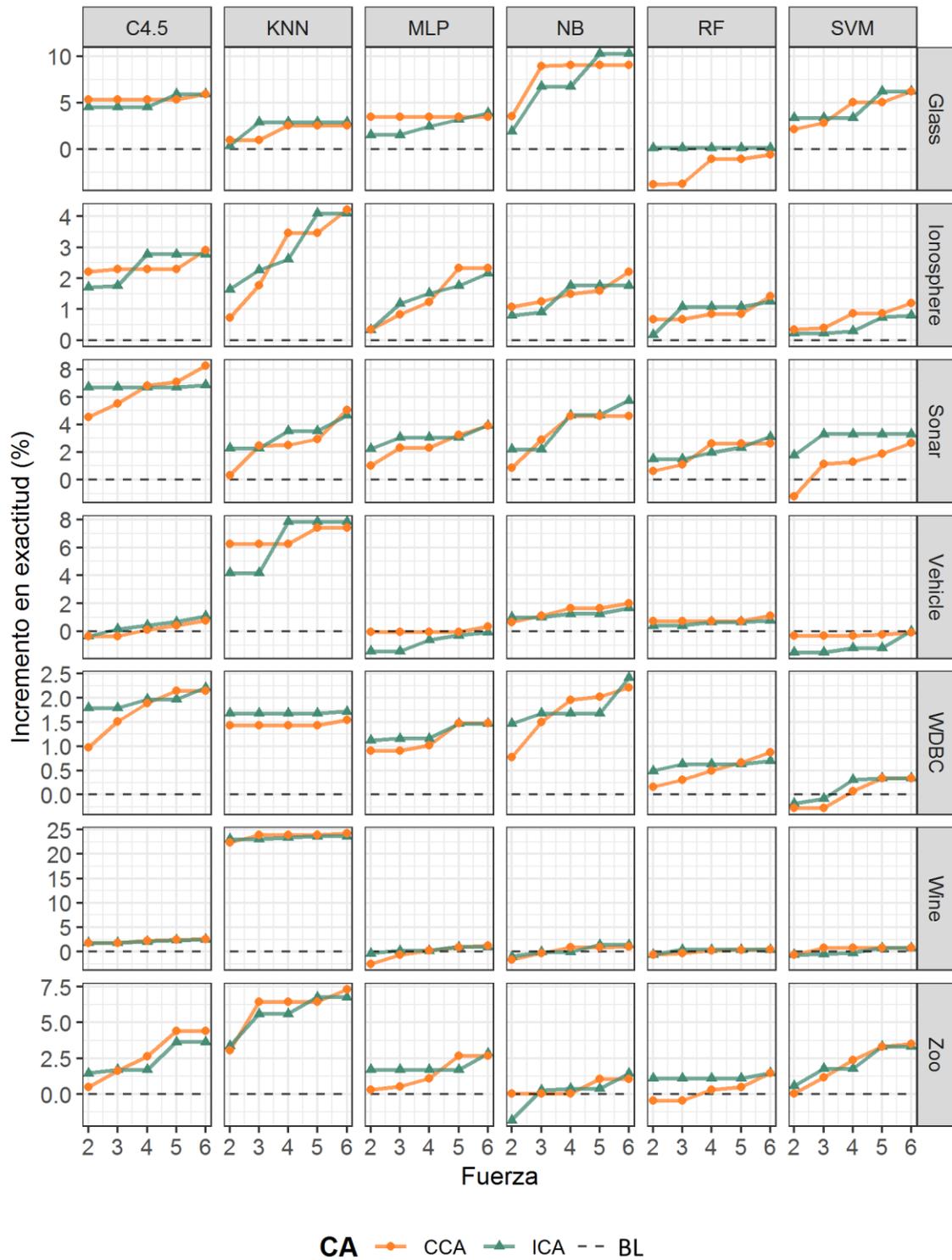


Figura 8. Incremento en exactitud y fuerza para los wrappers basados en CAs.

### 4.2.5 Efecto del wrapper propuesto sobre los diferentes MAS

La **Figura 9** muestra por MAS el incremento de exactitud respecto a la línea base a través de todos los datasets. Se puede observar en KNN los puntos atípicos que corresponden a la mejora realizada sobre Wine ya evidenciada en la **Figura 8**, donde siempre supera el 20%. También se observa que el mayor beneficiado en la selección de atributos (sin importar si el wrapper usa CCA o ICA) es KNN, seguido de C4.5 y NB, los cuales poseen una distribución del incremento en exactitud mayor a la obtenida sobre el resto de MAS, siendo los resultados esperados debido a que tanto KNN como los primeros MAS basados en árboles (C4.5 en este caso) son bastante sensibles a variables con ruido [101]–[104]. En segundo lugar, se encuentran MLP y SVM, que, al ser comparados con los anteriores, tienen un crecimiento más lento en la exactitud conforme aumenta la fuerza, sin embargo, la selección de atributos también los beneficia, aunque en menor medida. Esto se debe a los fuertes supuestos que tienen estos MAS sobre la independencia de las variables y la presencia de variables redundantes [105]–[108]. Por último con el menor incremento se encuentra RF, el cual realiza internamente un proceso de selección de atributos controlado por el parámetro *mtry*, que define un número de variables que se seleccionan al azar para la construcción de cada árbol del bosque [7], [86], [109].

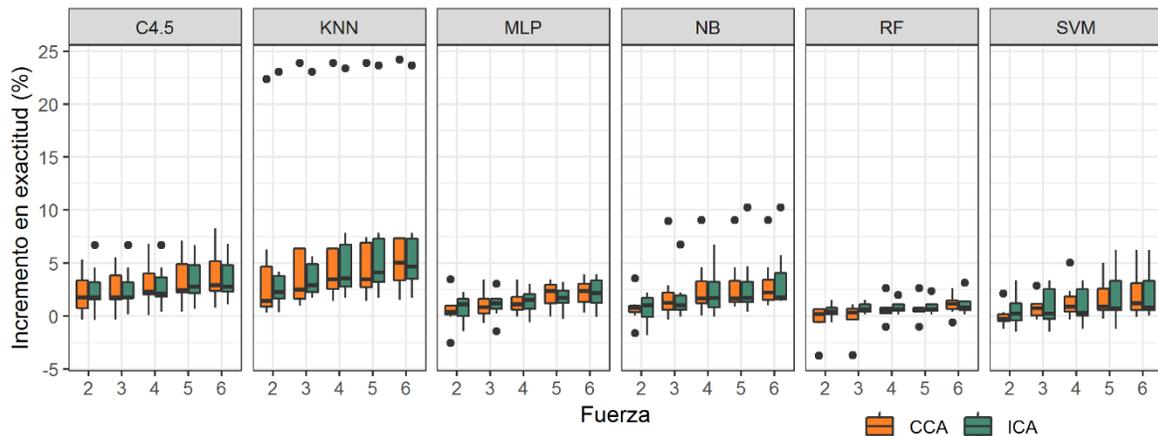


Figura 9. Incremento en la exactitud respecto a la BL por fuerza y MAS.

### 4.2.6 Comparación contra métodos del estado de arte

Los resultados obtenidos por los wrapper propuestos también fueron comparados con dos de los algoritmos más exitosos para la construcción de wrappers reportados en la literatura, estos son: algoritmos genéticos (GA) y optimización por enjambre

de partículas (PSO). Los operadores utilizados en el GA fueron: selección por ruleta, cruce de HUX, mutación de un gen y reemplazo del peor, mientras que el algoritmo implementado para PSO fue la versión binaria descrita en [71]. La configuración del GA se realizó con una población inicial de 20 individuos y una probabilidad de mutación de 0.01. En PSO se utilizó una población inicial de 20 partículas, una velocidad máxima de 6 y mínima de -6 y unas constantes de aceleración de 2, dichas configuraciones se escogieron de acuerdo a los test realizados en [110], donde hacen una comparación de algoritmos evolutivos para selección de atributos. Respecto al número de evaluaciones de la función objetivo (EFOs) controladas por el número de generaciones, se ajustan para que coincidan con la misma cantidad de casos evaluados por CCA en fuerza seis ( $t=6$ ), de esta manera se garantiza una comparación justa, siendo CCA el que requiere un mayor número de pruebas.

Para llevar a cabo la ejecución, se programaron el GA y el PSO en el software estadístico R y se evaluó la función de aptitud con base en la exactitud obtenida por el MAS y un subconjunto de variables candidatas elegidas a partir de un arreglo binario. Para GA y PSO, se realizaron 30 experimentos con distintas semillas en la inicialización para definir el comportamiento promedio de los resultados que entregan estos algoritmos. Pero para garantizar que los resultados sean comparables y eliminar el efecto producido por las particiones obtenidas en la validación cruzada, dentro del entrenamiento de los MAS en todos los wrappers se les ajustó la misma semilla (número aleatorio inicial), de manera que una misma solución candidata debe lograr el mismo resultado en PSO, GA, ICA o CCA.

La exactitud y el número de variables alcanzados por los wrappers evaluados a través de los seis MAS y los siete datasets se pueden observar en la **Tabla 7**, la comparación se realiza de manera similar a la sugerida en [50], [111]. En donde se utiliza la prueba de FART (presentada en la sección **2.4.2**), para analizar la significancia de las diferencias entre los resultados de los cuatro wrapper más la línea base, esto en forma independiente por cada MAS. El nivel de significancia utilizado fue de 0.05 y al encontrar en los resultados diferencias significativas se procedió a realizar un post hoc basado en FART. En la **Tabla 7** se utilizó mayúsculas para los resultados obtenidos de exactitud y minúsculas para la comparación del número de variables. También, para facilitar la comparación del desempeño de los enfoques evaluados, se utiliza el promedio de los rangos de la prueba de Friedman, cuya interpretación es similar a la de la **Tabla 6**. Al obtener los resultados de las pruebas realizadas tanto para exactitud (Acc), como para el número de variables

(NV), el FART sugirió diferencias significativas, por lo tanto, se procedió siempre a hacer la comparación con las pruebas post hoc en todas las pruebas, por cada MAS.

En las comparaciones de exactitud, la **Tabla 7** muestra que la línea base difiere significativamente con respecto a la mayoría de los wrappers exceptuando los resultados encontrados para RF, donde no hay evidencia de diferencias entre la línea base con CCA e ICA. En cuanto al valor del rango medio, PSO siempre obtuvo las exactitudes más altas a través de todos los modelos sobre los otros wrappers, además, en la mayoría de los MAS este enfoque está entre el primer y segundo puesto por escoger menos variables, convirtiéndolo en el wrapper mejor ubicado en el ranking. En NB, RF, MLP y SVM los wrapper CCA e ICA tienen desempeños más bajos y son estadísticamente distintos a PSO, y con base en los contrastes que involucran a GA versus CCA e ICA, existen diferencias significativas que sugieren que ambos son inferiores tanto en MLP como en RF, y en NB solamente con CCA. En el resto de los modelos los resultados de estos wrappers fueron comparables a nivel de la prueba post hoc. En KNN solo se presentaron diferencias en exactitud con respecto a la línea base, mientras en el resto de los wrappers del mismo KNN, no las hay.

Respecto al número de variables elegidas, se observó en la **Tabla 7**, que para todos los MAS las únicas diferencias fueron entre la línea base y los wrappers, por consiguiente, contrastando los cuatro wrappers, ninguno mostró ser distinto. Sin embargo se sabe que por propiedades de los CAs óptimos en los casos binarios, tales como los CAs e ICAS que se están utilizando en este caso, el número de veces que cada fila tiene valores 1 es equivalente al número de veces que la misma fila tiene valores 0, por lo tanto los CAs solo exploran subconjuntos con una cantidad de variables de entrada equivalente a la mitad del total de variables, siendo esta una desventaja cuando el número de variables relevantes real se distancie de la mitad de las variables totales, en este caso una estrategia de explotación como Hill Climbing podría complementar al wrapper.

*Tabla 7. Comparación de la exactitud y número de variables de los wrappers.*

MAS	Dataset	BL		CCA		ICA		PSO		GA	
		Acc	NV	Acc	NV	Acc	NV	Acc	NV	Acc	NV
C4.5	Glass	68.32	9	<b>74.24</b>	6	<b>74.24</b>	6	74.07	6	74	<b>5</b>
	Ionos.	90.05	33	92.96	12	92.84	15	<b>93.78</b>	<b>11</b>	93.46	13
	Sonar	73.14	60	81.4	27	79.99	28	<b>86.73</b>	<b>17</b>	84.08	22
	Vehicle	72.29	18	73.05	14	73.38	<b>10</b>	<b>74.06</b>	11	74	11

MAS	Dataset	BL		CCA		ICA		PSO		GA	
		Acc	NV	Acc	NV	Acc	NV	Acc	NV	Acc	NV
MAS	WDBC	93.89	30	96.04	13	96.1	15	<b>96.69</b>	<b>11</b>	96.3	12
	Wine	93.4	13	95.94	7	95.84	8	<b>96</b>	<b>6</b>	95.92	<b>6</b>
	Zoo	93.42	16	<b>97.83</b>	9	97.07	<b>8</b>	97.79	<b>8</b>	97.4	<b>8</b>
	FART	Acc: T = 20.19, df = 4, p-value < 0.01					NV: T = 19.742, df = 4, p-value < 0.01				
	Rank M.	5	5	2.64	3.14	3.36	3.14	<b>1.43</b>	<b>1.71</b>	2.57	2
	Frid. PH	C	b	AB	a	B	a	A	a	AB	a
	KNN	Glass	67.37	9	69.91	<b>5</b>	<b>70.27</b>	<b>5</b>	70.08	6	70.13
Ionos.		84.47	33	88.68	10	88.56	12	<b>91.38</b>	<b>8</b>	90.47	10
Sonar		81.31	60	86.35	32	85.97	30	<b>89.45</b>	<b>28</b>	88.46	30
Vehicle		64.52	18	71.95	<b>9</b>	72.37	10	<b>73.03</b>	11	72.88	11
WDBC		93.39	30	94.94	<b>14</b>	<b>95.11</b>	15	95.09	<b>14</b>	94.7	16
Wine		71.13	13	95.33	<b>5</b>	94.77	7	<b>95.52</b>	8	95.31	8
Zoo		85.81	16	93.11	<b>10</b>	92.58	12	<b>93.5</b>	<b>10</b>	93.27	11
FART		Acc: T = 19.96, df = 4, p-value < 0.01					NV: T = 19.49, df = 4, p-value < 0.01				
Rank M.		5	5	3.14	<b>1.86</b>	3	2.71	<b>1.43</b>	2.21	2.43	3.21
Frid. PH		B	b	A	a	A	a	A	a	A	a
MLP	Glass	66.08	9	69.54	6	69.99	<b>5</b>	<b>71.27</b>	6	70.17	<b>5</b>
	Ionos.	90.67	33	92.99	17	92.83	16	<b>93.66</b>	<b>15</b>	93.59	<b>15</b>
	Sonar	80.77	60	84.69	24	84.71	<b>22</b>	<b>87.34</b>	29	87.3	29
	Vehicle	81.44	18	81.77	<b>11</b>	81.36	13	<b>82.52</b>	14	82.21	14
	WDBC	95.92	30	97.4	16	97.4	15	<b>97.83</b>	<b>11</b>	97.76	12
	Wine	97.52	13	98.74	<b>9</b>	98.51	<b>9</b>	<b>98.8</b>	<b>9</b>	98.75	<b>9</b>
	Zoo	94.98	16	97.65	9	97.81	8	<b>98.17</b>	<b>7</b>	97.95	<b>7</b>
	FART	Acc: T = 23.36, df = 4, p-value < 0.01					NV: T = 17.59, df = 4, p-value < 0.01				
	Rank M.	4.86	5	3.43	3	3.71	<b>2.29</b>	<b>1</b>	2.43	2	<b>2.29</b>
	Frid. PH	C	b	B	a	B	a	A	a	A	a
NB	Glass	58.9	9	67.95	<b>5</b>	69.18	6	<b>69.8</b>	<b>5</b>	69.65	<b>5</b>
	Ionos.	91.34	33	93.56	17	93.11	<b>13</b>	<b>94.7</b>	14	94.4	14
	Sonar	75.58	60	80.19	32	81.32	32	<b>85.75</b>	<b>26</b>	84.97	27
	Vehicle	63.29	18	65.29	11	64.95	<b>6</b>	<b>66.02</b>	9	65.75	10
	WDBC	93.92	30	96.13	14	96.34	<b>9</b>	<b>97.46</b>	12	97.32	12
	Wine	97.78	13	98.79	<b>8</b>	99.22	<b>8</b>	<b>99.38</b>	<b>8</b>	99.35	<b>8</b>
	Zoo	88.72	16	89.79	10	90.16	11	<b>91.49</b>	<b>9</b>	91.28	10
	FART	Acc: T = 22.98, df = 4, p-value < 0.01					NV: T = 18.41, df = 4, p-value < 0.01				
	Rank M.	5	5	3.71	3.21	3.29	2.43	<b>1</b>	<b>1.93</b>	2	2.43
	Frid. PH	D	b	CD	a	BC	a	A	a	AB	a
RF	Glass	81.66	9	81.07	<b>6</b>	<b>81.8</b>	7	81.63	7	81.57	8
	Ionos.	93.58	33	94.99	16	94.82	17	<b>95.53</b>	<b>14</b>	95.4	15
	Sonar	84.4	60	87.03	32	87.52	<b>26</b>	<b>91.97</b>	27	91.07	29

MAS	Dataset	BL		CCA		ICA		PSO		GA	
		Acc	NV	Acc	NV	Acc	NV	Acc	NV	Acc	NV
	Vehicle	74.73	18	75.86	<b>10</b>	75.51	11	<b>77.11</b>	<b>10</b>	76.85	11
	WDBC	96.42	30	97.29	13	97.12	15	<b>97.77</b>	13	97.65	13
	Wine	98.45	13	98.86	<b>8</b>	98.86	12	<b>99.06</b>	9	99.01	9
	Zoo	97.52	16	99	<b>7</b>	99	<b>7</b>	<b>99.21</b>	8	98.76	9
	FART	Acc: T = 17.16, df = 4, p-value < 0.01					NV: T = 18.74, df = 4, p-value < 0.01				
	Rank M.	4.57	5	3.36	<b>2</b>	3.21	2.93	<b>1.29</b>	2.07	2.57	3
	Frid. PH	B	a	B	b	B	b	A	b	A	b
SVM	Glass	70.81	9	77.02	<b>3</b>	77.02	<b>3</b>	<b>77.37</b>	<b>3</b>	76.67	4
	Ionos.	95.27	33	96.47	<b>13</b>	96.07	14	<b>97.33</b>	16	97.22	17
	Sonar	88.49	60	91.15	30	91.81	31	<b>95.72</b>	<b>26</b>	94.7	30
	Vehicle	83.41	18	83.31	<b>13</b>	83.43	<b>13</b>	<b>84.87</b>	<b>13</b>	84.55	14
	WDBC	97.82	30	98.17	<b>15</b>	98.17	<b>15</b>	<b>98.4</b>	16	98.32	18
	Wine	98.57	13	99.33	<b>7</b>	99.32	8	<b>99.47</b>	8	99.43	9
	Zoo	94.69	16	98.21	11	98.01	11	<b>98.32</b>	<b>9</b>	98.11	10
	FART	Acc: T = 18.74, df = 4, p-value < 0.01					NV: T = 19.69, df = 4, p-value < 0.01				
	Rank M.	4.86	5	3.36	<b>1.93</b>	3.36	2.5	<b>1</b>	2.07	2.43	3.5
Frid. PH	C	b	BC	a	BC	a	A	a	AB	a	

En la **Figura 10** se evaluó la magnitud de la diferencia en la exactitud entre los wrappers propuestos en esta tesis y el algoritmo mejor ubicado en el ranking (PSO). Se puede observar con excepción de algunos máximos y valores atípicos, que gran parte de estas diferencias se encuentran por debajo de 2%.

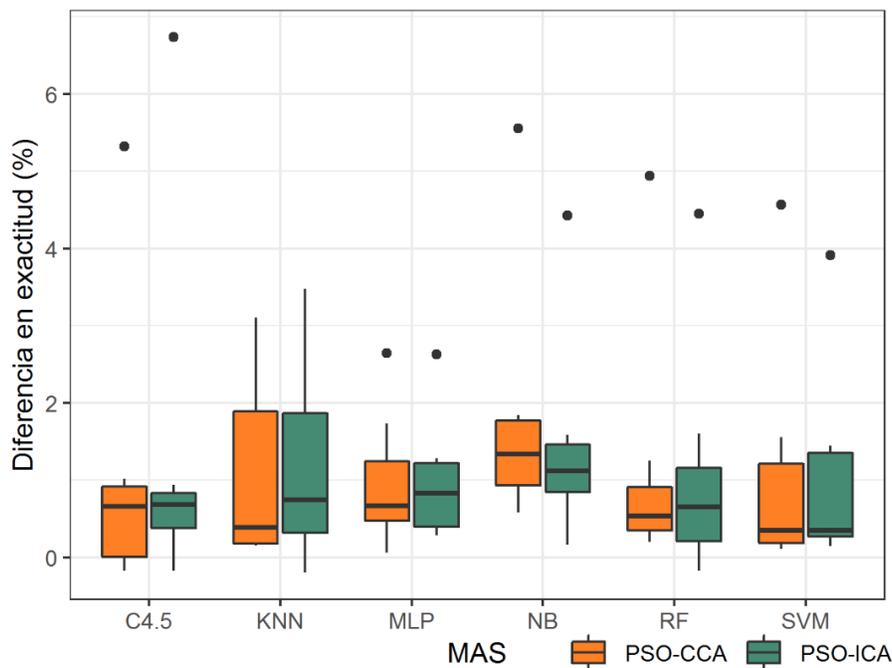


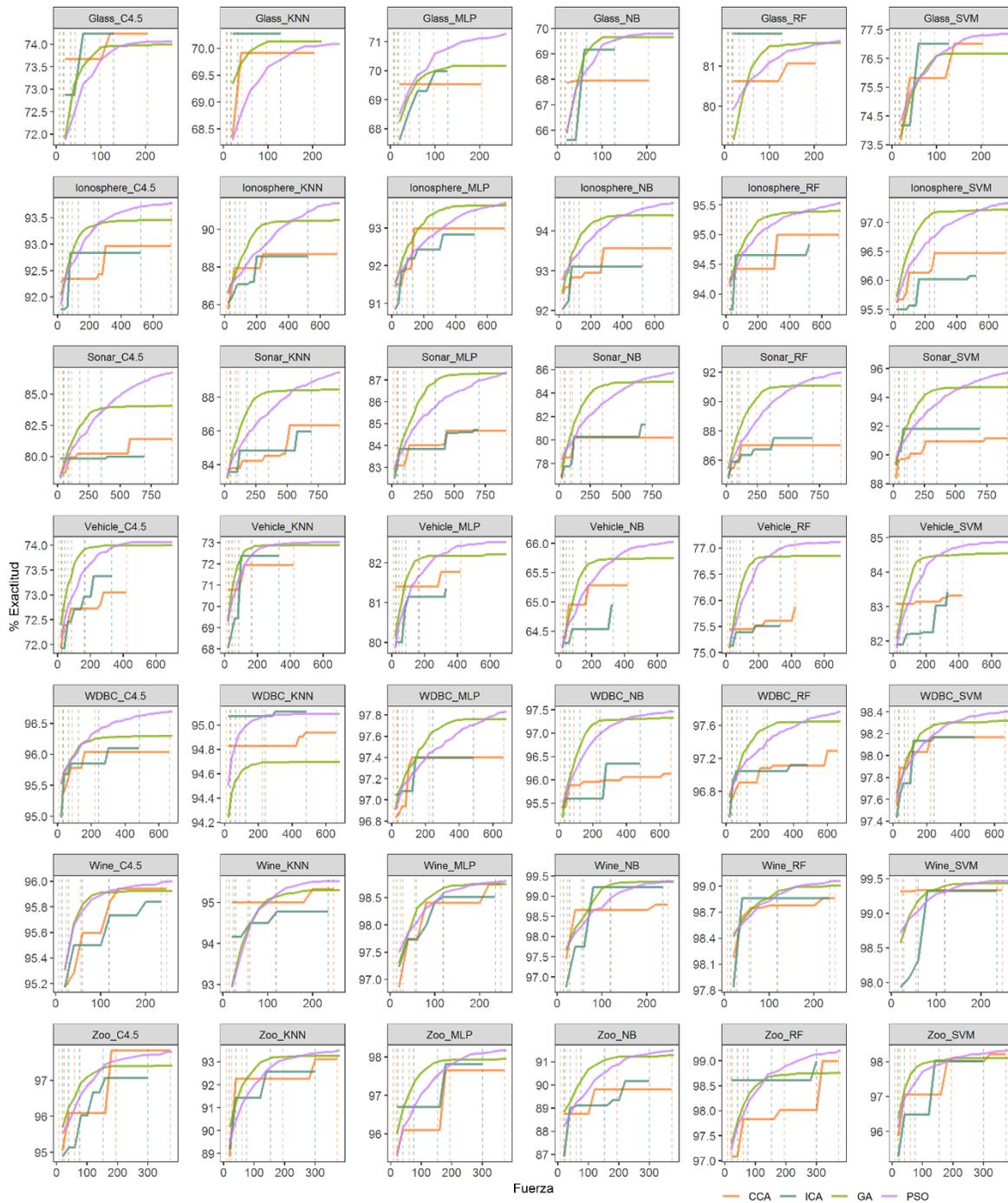
Figura 10. Diferencia entre la exactitud de PSO y los wrappers basados en CAs.

Otro aspecto analizado en la comparación de los wrappers fue la evolución del número de evaluaciones (EFOs) versus la exactitud alcanzada. Dichos resultados se pueden observar en la **Figura 11**, donde claramente se ve que CCA e ICA poseen un crecimiento escalonado que en la mayor parte de los casos no alcanza a PSO o GA, quienes tienen un crecimiento logarítmico. Este comportamiento se explica dado que los wrappers basados en CAs, no utilizan un enfoque de explotación por tanto al encontrar una mejor solución siguen explorando nuevas soluciones sin darle prioridad a los hallazgos, caso contrario a los wrappers inspirados en la naturaleza, que se centran en soluciones con alto potencial.

Adicionalmente, en la **Figura 11** también se puede observar el efecto de la exactitud con base en los niveles de fuerza del CA y el ICA. Una característica que resulta interesante es que el número de casos de prueba que se realizan en fuerza cuatro sobre un CCA, se ubica normalmente en el punto donde la curva del GA se estabiliza y por este motivo, se puede pensar que la cantidad de filas que posee el arreglo, pueden dar una guía del número de EFOs necesarias para alcanzar una solución aceptable para un wrapper con GA sin importar el MAS. Además, que para fuerza seis en los CCA, concuerda con el número de pruebas donde el PSO también comienza a estabilizarse. En el futuro se podría evaluar la posibilidad de que el

número de filas de CCA ayude a definir el número de generaciones necesarias para que GA y PSO alcancen la estabilidad.

Finalmente, y de manera general, en este capítulo se planteó la propuesta de un nuevo wrapper que está basado en CAs binarios a partir de la evaluación de subconjuntos de variables guiados por cada fila del CA. Si bien los resultados no superaron a PSO, los wrapper propuestos tienen la ventaja de que pueden ser más fácilmente ejecutados en paralelo y además al fijar las pruebas en CCAs e ICAs con fuerza seis, no es necesario involucrar parámetros adicionales, caso contrario a los GA y PSO donde se involucran otros parámetros como el tamaño de la población o enjambre, probabilidades, velocidades y algunos valores límites que deben ser afinados. Por consiguiente, de acuerdo con el contexto en el que se esté evaluando el MAS, un investigador puede tomar la opción de sacrificar exactitud (la mayoría de veces un valor aproximado de 2%) de acuerdo a lo observado en **Figura 10**, por velocidad y simplicidad, sobre todo en datasets grandes que día a día son más comunes (big data). Con estos resultados, en este capítulo se responde la primera parte de la pregunta de investigación, dado que se ha propuesto un wrapper basado en CAs que ofrece algunas ventajas respecto a los wrappers del estado del arte.



Ref.: Las líneas verticales representan el cambio en la fuerza de la matriz de cobertura.

Figura 11. Comparación entre la exactitud y el número de EFOs, por dataset y MAS.

## 5 Importancia de variables utilizando arreglos de cobertura

### 5.1 Algoritmo para importancia de variables

La propuesta para medir la importancia de variables consiste en determinar el efecto que produce involucrar una variable de entrada en un MAS sobre la calidad de la predicción, dicho efecto es medido al ser cada variable incluida en compañía de diferentes subconjuntos de variables los cuales pueden ser definidos a partir de las soluciones que evalúa un wrapper en su proceso de búsqueda, esta nueva medida pretende dar un valor agregado en la interpretación de la selección de atributos, ya que utiliza los resultados del proceso de selección de atributos para adicionar un producto nuevo, el cual es definir el peso o importancia que tiene cada variable de entrada sobre la variable de salida. A continuación, se realiza una propuesta para cualquier wrapper, sin embargo, en este trabajo se evalúan los resultados con selección de atributos por CA e ICA los cuales fueron definidos en el capítulo anterior.

El algoritmo parte de una matriz  $W$  binaria donde se han almacenado en cada una de las filas todas las  $n$  soluciones candidatas producto de la estrategia de búsqueda implementada por un wrapper, y un vector  $\alpha$  con el desempeño que ha tenido cada una de las soluciones o filas de  $W$ . Al inicio se crean tres vectores vacíos de longitud  $p$  (igual al número de variables) y se realiza un doble ciclo que recorre cada componente de la matriz  $W$  por cada variable evaluada. Dentro de este ciclo se calculan: (i) los acumulados de desempeño de las soluciones en las cuales cada variable estuvo presente y se almacenan en  $V1$ , (ii) el acumulado del desempeño donde cada variable correspondiente estuvo ausente y se almacena en  $V2$ , (iii) el conteo de presencia de cada variable y es almacenado en  $C1$ . Finalmente, cada componente de los tres vectores se utiliza para calcular por cada variable la diferencia del promedio del desempeño en las soluciones de presencia versus las soluciones de ausencia, esta diferencia es finalmente almacenada en un vector  $V$ . Con esto se obtiene una medida de relevancia que indica el balance en el desempeño que produjo en el MAS la participación de cada variable.

La formulación propuesta permite que las variables de entrada poco importantes o irrelevantes tengan una importancia cercana a cero cuando su presencia en el MAS no produce ningún efecto sobre el desempeño o una importancia negativa si su presencia reduce el desempeño de los resultados obtenidos. Mientras que, para las variables de entrada que sean importantes se obtienen valores positivos y entre más grande sea el valor más importante es la variable. Con esto se obtiene un ranking fácil de interpretar por parte del usuario. El **Algoritmo 2**, muestra el procedimiento para definir la importancia de las variables partiendo de las soluciones evaluadas por un wrapper, el cual tiene un orden de complejidad  $O(n \times p)$  de operaciones básicas que no involucran la ejecución de un MAS.

**Inputs:** Matriz  $W$  con  $p$  columnas y  $n$  filas // Donde  $p$  es el número de variables de entrada del dataset y  $n$  es la cantidad de EFOs ejecutada por el wrapper.  
Vector  $\alpha$  con  $n$  filas // Por cada fila en  $W$  este vector ha almacenado el desempeño/Métrica de calidad (Exactitud, medida  $f$ , precisión, error cuadrático medio u otro).

**Output:** vector  $V$  con  $p$  columnas // Por cada columna se define un valor de importancia

```

1.  begin
2.     $VI =$  Vector vacío de longitud  $p$ 
3.     $V2 =$  Vector vacío de longitud  $p$ 
4.     $CI =$  Vector vacío de longitud  $p$ 
5.    for  $r = 1$  to  $n$  do
6.      for  $i = 1$  to  $p$  do
7.        if  $W[r, i] == 1$  then
8.           $CI[i] = CI[i] + 1$ 
9.           $VI[i] = VI[i] + \alpha_i$ 
10.       Else
11.          $V2[i] = V2[i] + \alpha_i$ 
12.       end if
13.     end for
14.   end for
15.   for  $i = 1$  to  $p$  do
16.      $V[i] = (V2[i] / CI[i]) - (VI[i] / (n - CI[i]))$ 
17.   end for
18.   return  $V$ 
19. end

```

*Algoritmo 2. Pseudocódigo del algoritmo para definir la importancia de variables*

## 5.2 Comparación de los resultados según la fuerza

Para la implementación de los algoritmos de importancia de variables se utilizaron los resultados generados en la selección de atributos del capítulo anterior (relacionado con la propuesta de wrapper), y los mismos datasets y MAS. Pero para ilustrar en el cuerpo del documento se mostró especialmente los resultados obtenidos con el dataset Zoo, el resto de los gráficos para los otros datasets se pueden ver en el anexo 5.

La **Figura 12**, muestra los resultados obtenidos de implementar el algoritmo de importancia de variables sobre el dataset Zoo y CCA e ICA utilizando fuerza seis, se puede observar que X13, X4 y X8 aparecieron casi siempre en los primeros puestos, lo que sugiere que son variables relevantes para la mayoría de los MAS y la variable X9 parece tener un peso importante en los algoritmos C4.5, MLP, RF y SVM mientras que no es tan relevante en NB y KNN, adicionalmente se aprecia que X4 pierde importancia en MLP y SVM, mientras que en los otros MAS siempre se mantiene en los tres primeros puestos de importancia. También se puede observar que la variable X13, en KNN fue el único caso donde quedó de última en relevancia. Por último, se puede apreciar que varias variables de entrada no llegan a ocupar los primeros puestos, tales como X7, X11, X15 y X16, lo cual implica que estas variables de entrada son poco relevantes, redundantes o irrelevantes.

Las diferencias de importancia de variables en los MAS de la **Figura 12** eran esperadas debido a que algunos MAS pueden detectar mejor las interacciones, asociaciones o efectos de escala entre las variables de entrada y por tal razón la importancia de las variables puede variar de un MAS a otro. Una de las grandes ventajas del algoritmo propuesto para medir la importancia de variables basado en wrappers, es que permite adaptarse a cada MAS, por lo que deja en evidencia los efectos de las variables de entrada sobre el mismo. Otras medidas de importancia de variables como los propuestos para RF [6], [86], solo aplican específicamente para el mismo RF, ya que hace parte de su entrenamiento.

La fuerza del CCA e ICA también puede influir sobre la calidad del resultado obtenido para la importancia de variables, por tal razón al igual que en el capítulo anterior, se realizó la comparación respectiva a los resultados obtenidos por cada fuerza a través de todos los datasets y MAS. No obstante, en la literatura, no es fácil encontrar criterios estándar para evaluar la calidad de la importancia de variables a partir de resultados por distintas metodologías, de echo Pangfei *et al.* [7] en su

revisión de literatura, comparan los métodos utilizando simulación y describiendo las cualidades de cada uno, y dado que en este proyecto de investigación se tiene como uno de los propósitos desarrollar una medida de importancia de variables a partir de los resultados de wrappers, se proponen dos nuevos indicadores que evalúan el desempeño y la concordancia con los wrappers, ambos utilizan como referencia el mejor subconjunto (Best subset, BS) de variables de entrada encontrado en el capítulo anterior. A continuación, se ilustra con un ejemplo cómo se calculan ambos indicadores.

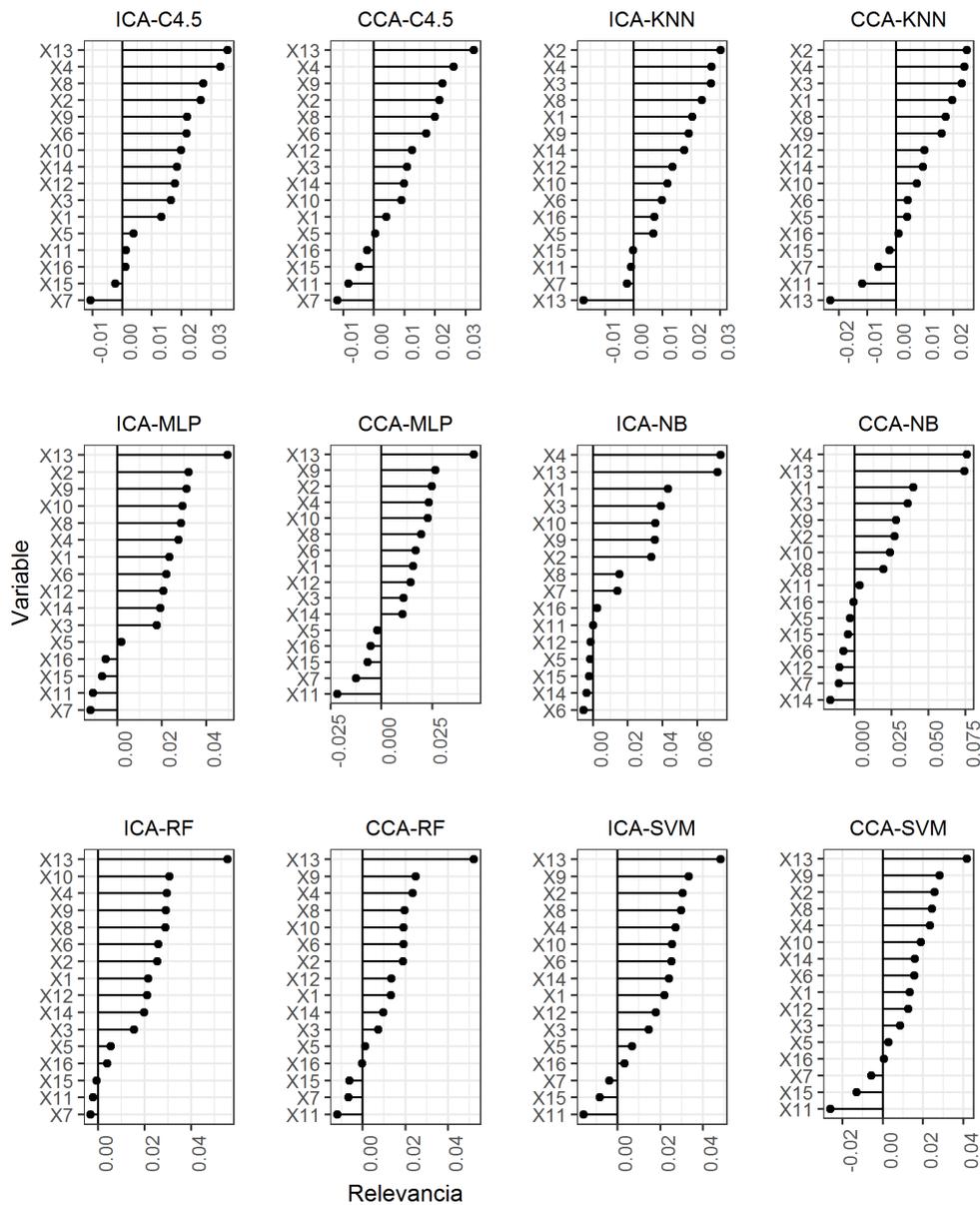


Figura 12. Importancia de variables según CAs de fuerza 6, para Zoo.

En cada caso el BS está compuesto por un total de  $k$  variables conocido por los autores. Por ejemplo, tomando como referencia Zoo para NB, el mejor subconjunto de variables encontrado fue  $BS = \{X3, X4, X7, X9, X12, X13\}$ , con  $k = 6$ , obtenido en PSO. Posteriormente se toman de los resultados de la importancia de variables de un wrapper las primeras  $k$  variables con mayor relevancia, en el ejemplo teniendo como referencia a la **Figura 12**, para CCA sería  $\{X4, X13, X1, X3, X9, X2\}$  y para ICA  $\{X4, X13, X1, X3, X10, X9\}$  siendo  $X10$  y  $X2$  las únicas variables diferentes en ambos subconjuntos. El primer indicador consiste en calcular la exactitud obtenida de las  $k$  variables más importantes siendo para CCA  $\alpha = 90.72$  y para ICA  $\alpha = 82.50$ , el segundo mide el porcentaje de coincidencia entre las  $k$  variables de importancia de variables versus las variables del BS, siendo en este caso para ambos wrappers el 66.6% (4 de 6). Finalmente, el CCA retiene en las primeras variables más relevantes, una mayor cantidad de exactitud, por tal razón se considera superior a los resultados alcanzados por el ICA en el dataset Zoo con NB. Para efectos de notación, al primer indicador se le nombra exactitud de las variables más importantes (Accuracy in the Most Important Variables, AMIV) y el segundo, coincidencia con el mejor subconjunto de variables (Coincidence with the Best Subset of Variables, COBESV).

Ambos indicadores fueron evaluados para los resultados de importancia de variables obtenidos con ACA e ICA en todas las fuerzas ( $t=2$  a  $t=6$ ), y comparados a través de la prueba de Friedman. Los resultados se presentan en la **Tabla 8**, donde nuevamente queda en evidencia que las fuerzas altas corresponden con mejores resultados tanto en AMIV como en COBESV. El mejor wrapper de acuerdo con los criterios de evaluación es el CCA-6, dado que siempre quedó en el primer puesto. Adicionalmente, ICA-6 queda en tercer puesto en AMIV y segundo puesto en COBESV sin ser significativamente distinto a CCA-6 ya que siempre están dentro del mismo grupo post hoc (a). Finalmente, CCA-6 e ICA-6 se seleccionan para hacer la comparación con el estado del arte.

Tabla 8. Friedman para la comparación de CA-Fuerza según AMIV y COBESV.

CA-Fuerza	AMIV		COBESV	
	Ranking	Post Hoc	Ranking	Post Hoc
CCA-6	<b>4.06</b>	a	<b>4.32</b>	a
CCA-5	4.11	ab	4.87	ab
ICA-6	4.75	abc	4.80	a
ICA-5	4.89	abc	5.35	abc
ICA-4	5.26	bcd	4.79	a

CA-Fuerza	AMIV		COBESV	
	Ranking	Post Hoc	Ranking	Post Hoc
CCA-4	5.40	cd	5.23	abc
ICA-3	5.90	cde	5.87	bc
CCA-3	6.44	def	6.11	c
ICA-2	6.81	ef	6.17	c
CCA-2	7.37	f	7.51	d

### 5.3 Comparación con métodos del estado de arte

Teniendo en cuenta que la propuesta realizada para medir la importancia de variables planteada en la **Algoritmo 2.**, no solo aplica para wrappers que sean CCA o ICA, sino para cualquiera, en la comparación se incluyeron los resultados de selección de atributos obtenidos de GA y PSO, y debido a que en ambos wrapper se realizaron 30 experimentos con distintas semillas en la inicialización, se obtiene por cada variable en cada MAS una distribución de 30 valores de importancia, los cuales pueden ser observados en la **Figura 13.**

Respecto al orden de relevancia en la **Figura 13**, las variables detectadas como más importantes por los MAS a través de GA y PSO son muy similares a las encontradas en la **Figura 12** por CA e ICA, tales casos son: el de X13 y X4, como las variables más relevantes en la mayoría de MAS, y el efecto de X13 en KNN. También se observó algunas diferencias entre wrappers sobre los mismos MAS como el ascenso de importancia de X6 en MLP, RF y SVM, dichas diferencias sugieren que las actividades de exploración y explotación que realiza la estrategia de búsqueda de los wrappers pueden afectar las medidas de importancia.

Posteriormente para realizar una comparación sobre el efecto que tiene el orden de las variables de entrada en la importancia de variables frente a la medida de desempeño que se evalúa en el modelo (en este caso la exactitud). Se mide la exactitud de los MAS por medio de subconjuntos acumulados de variables de entrada los cuales son conformados de acuerdo con el orden de importancia, es decir, se empieza con la variable más relevante de entrada, luego las dos más relevantes y así sucesivamente hasta completar todas las variables de entrada. Finalmente, a cada subconjunto se le mide la exactitud sobre la clasificación. Estos resultados se muestran tanto en la **Figura 14** como en la **Figura 15.**

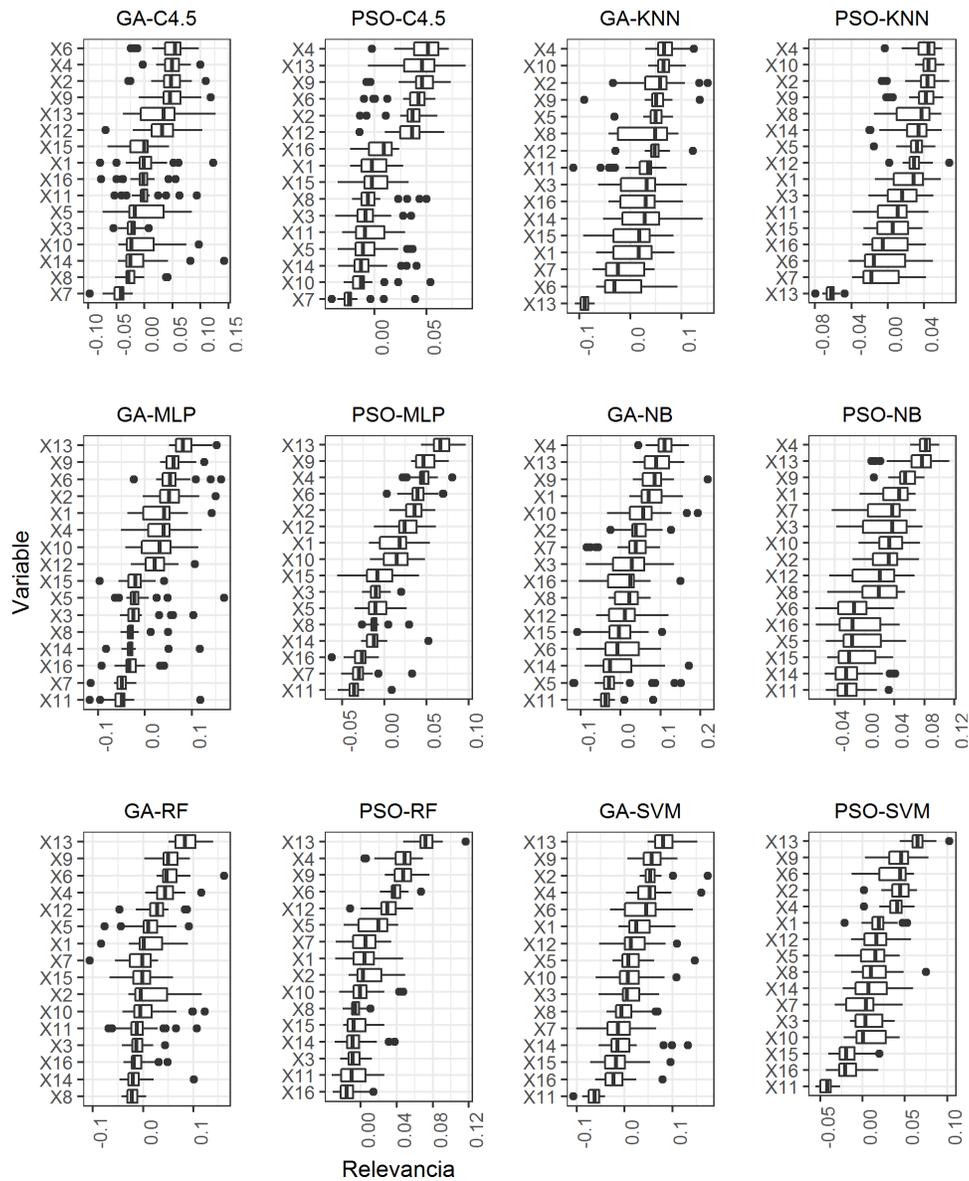


Figura 13. Importancia de variables en GA y PSO, para Zoo.

La **Figura 14** muestra para GA y PSO una nube de líneas en color verde claro debido a que se presentan los resultados de los 30 experimentos sobre cada MAS. Además, se muestra en línea sostenida de color rojo el experimento que mejor representa la exactitud que se logra con el wrapper, este experimento es identificado a partir de la curva que más se acerque a la mediana de todos los experimentos, la proximidad es definida con base en la distancia euclidiana.

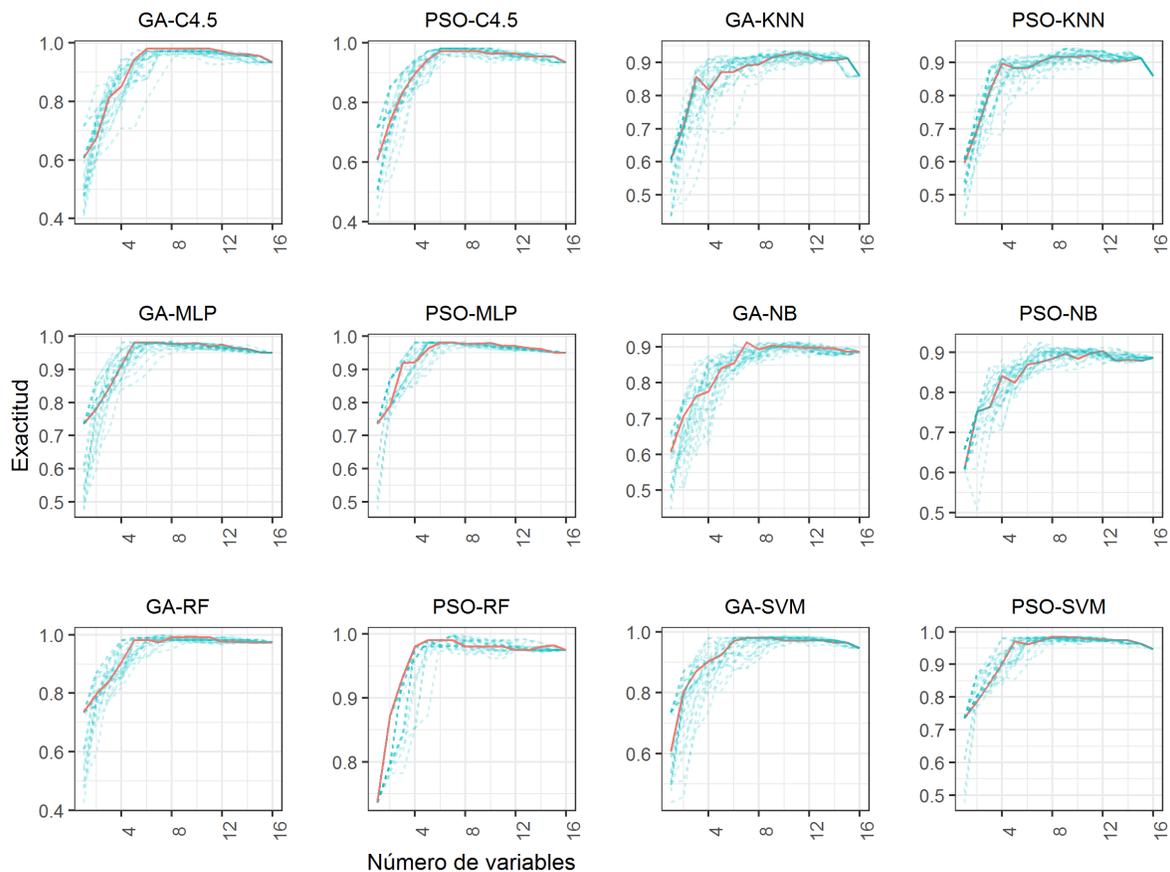
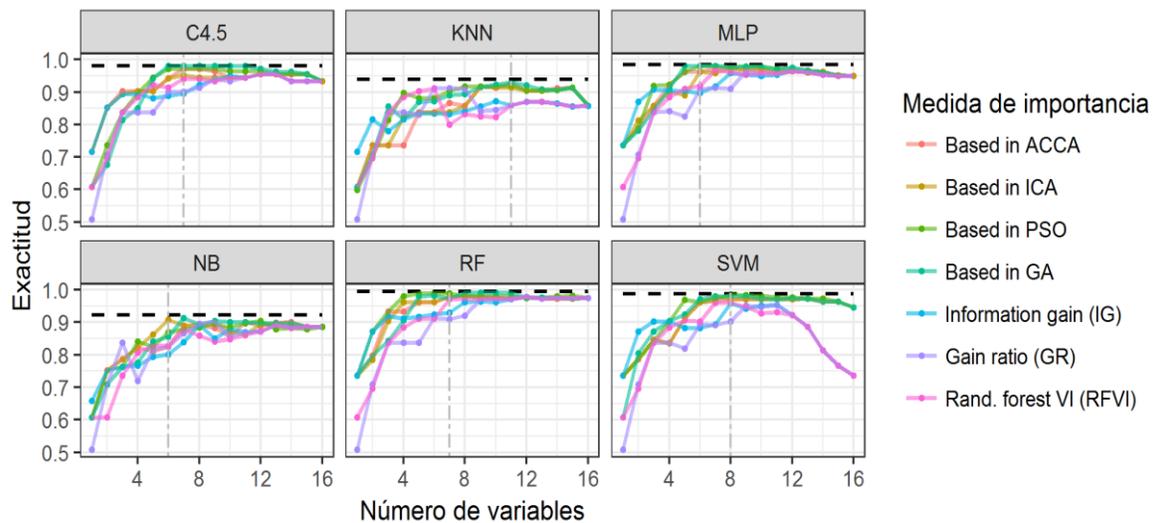


Figura 14. Incremento de la exactitud según orden de importancia de variables, en los 6 MAS para Zoo.

Adicionalmente, la **Figura 14** permite observar la tendencia y el grado de dispersión de los resultados que se obtienen, destacándose un valor alto de variabilidad de las líneas en GA-KNN y GA-NB. En dicha figura se puede observar además que todas las curvas son crecientes con tendencia logarítmica, es decir, después de cierto número de variables de entrada no hay un incremento significativo en la exactitud del MAS, lo que permite pensar que no es necesario adicionar todas las variables de entrada para obtener una buena calidad de predicción.

Los valores de importancia de variables para GA y PSO utilizados en las comparaciones realizadas más adelante fueron basados en los experimentos resaltados en rojo de la **Figura 14**. Además, en la comparación también se incluye la importancia de variables obtenida por la Medida de Importancia de Variables de Random Forest basada en permutaciones (Random Forest Variable Importance measure, RFVI) y los filtros de Ganancia de Información (Information Gain, IG) y Radio de Ganancia (Gain Ratio, GR).

En la **Figura 15**, se puede observar que en el punto de intersección entre la recta horizontal de la mejor calidad conocida (Mejor exactitud) con la recta vertical que representa la cantidad de variables de entrada encontradas en la mejor solución conocida para Zoo y cada MAS, es donde los algoritmos tienden a alcanzar las mejores soluciones (mejor calidad) y luego estabilizan dicho valor. La curva de GR parece caer y subir especialmente en MLP, NB y SVM debido a la repentina inclusión de variables de entrada que pueden afectar negativamente la exactitud, lo cual puede ser consecuencia de que esta medida no tiene en cuenta el efecto multivariado entre las variables de entrada. También se observa que las líneas azules, moradas y rosadas, que representan respectivamente a IG, GR y RFVI tienden a estar por debajo de las otras medidas de importancia de variables, lo cual puede explicarse por el hecho de que estas no incluyen información del MAS, con excepción del RFVI sobre el mismo RF. Más adelante se muestra una prueba formal para comprobar la dominancia de los resultados obtenidos con los wrappers.



REF: Las líneas de puntos verticales y horizontales representan la mejor solución lograda en la selección de características.

Figura 15. Incremento en la exactitud según el orden de importancia de las variables, para Zoo.

Al igual que en el capítulo anterior de selección de atributos, se procedió a comparar los resultados obtenidos de las importancias de variables propuestas y las del estado del arte, a partir de pruebas no paramétricas formales. La calidad de los resultados se evaluó con base en el AMIV y COBESV, y se realizó independiente por cada MAS. En este caso, debido a que se realizan siete comparaciones, se utilizó lo sugerido por García et al. en [50], que dice que para más de seis métodos a comparar se debe utilizar el test de Friedman explicado en la sección 2.4.1, también se realizaron test post hoc para las pruebas donde se hallaron diferencias

significativas. Valores bajos en el promedio del rango de Friedman representan AMIV o COBESV más altos, lo cual sugiere que la calidad de la medida de importancia de variables es mejor.

La **Tabla 9** muestra los resultados obtenidos de la evaluación del AMIV, donde se observa que el PSO nuevamente queda de primero en todos los MAS con excepción del KNN, donde queda de segundo. Este resultado demuestra que la capacidad de PSO para seleccionar los mejores subconjuntos de variables también se ve reflejada en la medición de la importancia de variables al mantener en el subconjunto compuesto por las variables de entrada más relevantes una mayor cantidad de exactitud que en comparación a otras medidas, incluso por encima de CCA e ICA en el ranking sin embargo en KNN y MLP no tuvo diferencias significativas.

Adicionalmente, GA fue destacado usualmente en segundo lugar, siempre siendo comparable con PSO al no evidenciar diferencias significativas con este a través de todos los MAS, sin embargo, al ser comparado con los wrappers propuestos (CCA e ICA) no evidencia diferencias significativas en C4.5, MLP y una vez en KNN, dichos MAS son altamente sensibles a variables redundantes. Por otro lado, CCA e ICA estuvieron siempre significativamente por encima de IG y GR, y la mayoría de las veces también de RFIV, dejando a estos tres en los últimos lugares de los rankings.

Tabla 9. Comparación de las importancias de variables según el AMIV.

Model	Dataset	BS	CCA	ICA	GA	PSO	IG	GR	RFIV
C4.5	Glass	74.24	70.81	69.57	<b>72.04</b>	<b>72.04</b>	68.81	68.81	70.81
	Ionosphere	94.20	92.10	92.44	93.12	<b>93.75</b>	88.94	89.64	93.46
	Sonar	89.53	82.36	80.03	81.54	<b>84.82</b>	77.98	71.55	76.00
	Vehicle	74.32	73.18	<b>73.28</b>	72.53	72.90	67.54	69.96	72.81
	WDBC	96.98	95.01	94.76	95.71	<b>96.13</b>	92.65	92.65	94.24
	Wine	96.06	95.64	95.64	96.08	<b>96.20</b>	94.63	93.85	93.85
	Zoo	98.01	<b>97.15</b>	95.10	98.07	97.18	89.56	90.20	94.15
	Rank mean		3.14	3.64	2.50	<b>1.50</b>	6.29	6.36	4.57
Friedman		$\chi = 31.2, k = 6, p\text{-value} < 0.01$							
Post hoc		b	bc	ab	a	d	d	c	
KNN	Glass	70.72	69.79	69.79	69.79	<b>70.06</b>	67.33	66.75	66.61
	Ionosphere	92.84	89.02	86.40	<b>89.47</b>	89.37	86.33	85.95	89.52
	Sonar	91.24	<b>86.63</b>	85.44	86.20	87.67	84.07	84.07	82.95
	Vehicle	73.46	<b>72.13</b>	71.82	71.94	71.65	64.37	65.06	67.09
	WDBC	95.11	92.06	92.06	<b>95.18</b>	95.15	93.32	93.32	93.40

Model	Dataset	BS	CCA	ICA	GA	PSO	IG	GR	RFIV
MLP	Wine	95.74	94.30	93.51	<b>94.85</b>	93.39	71.76	71.88	71.76
	Zoo	93.95	92.83	91.36	<b>93.02</b>	92.09	86.03	86.03	85.99
	Rank mean		2.93	4.07	<b>1.86</b>	2.57	5.71	5.64	5.21
	Friedman		$\chi = 22.9, k= 6, p\text{-value} < 0.01$						
	Post hoc		ab	bc	a	ab	c	c	c
	Glass	71.61	67.21	<b>69.27</b>	<b>69.79</b>	<b>69.79</b>	66.24	66.24	67.21
	Ionosphere	94.42	92.59	<b>93.73</b>	93.39	92.66	88.55	90.21	89.69
	Sonar	89.19	84.24	83.94	84.57	<b>84.77</b>	78.73	78.73	82.10
	Vehicle	83.00	81.65	81.65	81.23	<b>81.77</b>	78.99	77.82	81.65
	WDBC	98.52	97.33	96.98	<b>97.78</b>	96.98	95.46	95.68	95.68
NB	Wine	98.89	<b>98.34</b>	<b>98.34</b>	97.43	98.21	95.74	98.11	97.08
	Zoo	98.41	96.20	96.20	<b>98.07</b>	<b>98.07</b>	89.72	90.80	91.88
	Rank mean		3.07	2.71	2.57	<b>2.14</b>	6.71	5.86	4.93
	Friedman		$\chi = 30.4, k= 6, p\text{-value} < 0.01$						
	Post hoc		a	a	a	a	c	bc	b
	Glass	69.82	69.31	69.31	<b>70.70</b>	<b>70.70</b>	62.42	54.45	63.67
	Ionosphere	95.10	92.48	92.66	93.62	<b>94.20</b>	91.46	91.40	92.19
	Sonar	87.21	81.15	81.36	84.53	<b>85.27</b>	74.81	74.81	78.20
	Vehicle	66.42	62.72	62.99	<b>65.79</b>	<b>65.79</b>	60.66	60.03	62.04
	WDBC	97.82	95.33	95.99	<b>97.19</b>	<b>97.61</b>	94.45	94.17	96.20
RF	Wine	99.44	98.55	98.55	<b>99.33</b>	98.79	96.76	97.78	97.08
	Zoo	92.24	82.50	<b>90.76</b>	85.53	86.94	80.10	82.42	82.80
	Rank mean		4.14	3.00	1.86	<b>1.43</b>	6.36	6.50	4.71
	Friedman		$\chi = 37.3, k= 6, p\text{-value} < 0.01$						
	Post hoc		c	b	a	a	d	d	c
	Glass	82.10	79.17	79.17	<b>81.88</b>	<b>81.88</b>	79.17	79.17	79.17
	Ionosphere	95.85	94.59	95.06	95.05	<b>95.22</b>	93.29	92.56	94.03
	Sonar	92.91	87.48	87.76	90.86	<b>90.94</b>	80.69	80.69	85.56
	Vehicle	77.19	74.99	74.30	75.81	<b>76.78</b>	73.33	73.43	75.06
	WDBC	98.18	96.63	96.48	97.02	<b>98.00</b>	95.04	94.83	96.87
SVM	Wine	99.31	98.12	98.12	<b>98.55</b>	98.33	98.11	97.67	98.12
	Zoo	99.44	97.87	97.87	97.40	<b>99.03</b>	92.94	90.93	97.17
	Rank mean		4	3.86	2.21	<b>1.21</b>	6.07	6.50	4.14
	Friedman		$\chi = 34.4, k= 6, p\text{-value} < 0.01$						
	Post hoc		b	b	a	a	c	c	b
	Glass	77.38	74.09	70.04	<b>76.56</b>	<b>76.56</b>	69.60	55.65	67.50
	Ionosphere	97.83	96.30	95.84	<b>97.16</b>	97.10	89.56	90.09	90.65
	Sonar	96.45	93.42	<b>95.08</b>	90.73	92.00	66.82	67.20	69.61
	Vehicle	84.87	83.93	82.84	84.14	<b>84.28</b>	74.80	72.62	75.15
	WDBC	98.59	97.82	97.40	98.00	<b>98.21</b>	93.81	94.34	94.80

Model	Dataset	BS	CCA	ICA	GA	PSO	IG	GR	RFIV
	Wine	99.56	98.33	98.33	98.90	<b>99.44</b>	97.98	97.98	97.98
	Zoo	98.84	97.15	97.15	98.12	<b>98.61</b>	95.91	90.20	96.14
	Rank mean		3	3.43	2.07	<b>1.5</b>	6.29	6.43	5.29
	Friedman		$\chi = 36.8, k=6, p\text{-value} < 0.01$						
	Post hoc		b	b	a	a	d	d	c

En la **Tabla 10** en la comparación de los COBSEV, se puede observar que CCA en C4.5, obtiene el primer lugar, mientras que en MLP lo obtiene ICA, y que la mayoría de veces los resultados son comparables con GA y PSO, además estas cuatro importancias de variables nombradas están siempre por encima de IG, GR y RFIV a través de todos los MAS. Estos resultados sugieren que tanto CCA como ICA tienen una gran coincidencia con el número de variables de la mejor solución encontrada que les permite competir con GA y PSO.

Tabla 10. Comparación de las importancias de variables según el COBESV.

Model	Datasets	CCA	ICA	GA	PSO	IG	GR	RFIV
C4.5	Glass	<b>66.67</b>	<b>66.67</b>	<b>66.67</b>	<b>66.67</b>	<b>66.67</b>	<b>66.67</b>	<b>66.67</b>
	Ionosphere	<b>50.00</b>	<b>50.00</b>	37.50	50.00	12.50	12.50	50.00
	Sonar	40.00	46.67	40.00	40.00	33.33	26.67	33.33
	Vehicle	72.73	72.73	<b>81.82</b>	63.64	63.64	63.64	63.64
	WDBC	54.55	45.45	45.45	<b>90.91</b>	36.36	36.36	27.27
	Wine	<b>80.00</b>	<b>80.00</b>	40.00	<b>80.00</b>	40.00	60.00	60.00
	Zoo	<b>85.71</b>	71.43	<b>85.71</b>	57.14	42.86	42.86	28.57
	Rank mean	<b>2.50</b>	2.64	3.50	3.14	5.57	5.50	5.14
	FART	$\chi = 21.8, k=6, p\text{-value} < 0.01$						
Post hoc	a	a	a	a	b	b	b	
KNN	Glass	60.00	60.00	60.00	<b>100.00</b>	60.00	60.00	60.00
	Ionosphere	40.00	<b>60.00</b>	40.00	<b>60.00</b>	0.00	20.00	0.00
	Sonar	60.71	60.71	50.00	<b>64.29</b>	60.71	60.71	50.00
	Vehicle	55.56	66.67	55.56	<b>77.78</b>	44.44	44.44	66.67
	WDBC	61.54	53.85	61.54	<b>69.23</b>	61.54	53.85	53.85
	Wine	<b>66.67</b>	<b>66.67</b>	50.00	50.00	16.67	33.33	16.67
	Zoo	<b>72.73</b>	<b>72.73</b>	81.82	<b>72.73</b>	63.64	63.64	<b>72.73</b>
	Rank mean	3.43	3.29	3.79	<b>1.79</b>	5.29	5.29	5.14
	FART	$\chi = 19.2, k=6, p\text{-value} < 0.01$						
Post hoc	ab	ab	bc	a	c	c	c	
MLP	Glass	83.33	83.33	<b>100.00</b>	<b>100.00</b>	66.67	66.67	50.00
	Ionosphere	50.00	58.33	58.33	50.00	33.33	<b>75.00</b>	50.00
	Sonar	60.00	64.00	52.00	<b>72.00</b>	28.00	28.00	40.00

Model	Datasets	CCA	ICA	GA	PSO	IG	GR	RFIV
	Vehicle	<b>93.33</b>	<b>93.33</b>	86.67	<b>93.33</b>	86.67	86.67	<b>93.33</b>
	WDBC	<b>40.00</b>	<b>40.00</b>	30.00	30.00	20.00	30.00	30.00
	Wine	<b>88.89</b>	<b>88.89</b>	66.67	<b>88.89</b>	66.67	66.67	55.56
	Zoo	66.67	66.67	<b>100.00</b>	83.33	50.00	33.33	33.33
	Rank mean	3.00	<b>2.50</b>	3.50	2.64	6.00	5.00	5.36
	FART	$\chi = 20.6, k= 6, p\text{-value} < 0.01$						
	Post hoc	a	a	ab	a	c	bc	c
NB	Glass	80.00	80.00	<b>100.00</b>	<b>100.00</b>	40.00	60.00	40.00
	Ionosphere	<b>61.54</b>	<b>61.54</b>	<b>61.54</b>	<b>61.54</b>	30.77	30.77	30.77
	Sonar	65.22	56.52	56.52	<b>86.96</b>	39.13	39.13	47.83
	Vehicle	55.56	44.44	77.78	<b>77.78</b>	44.44	44.44	66.67
	WDBC	<b>66.67</b>	58.33	58.33	<b>66.67</b>	33.33	33.33	33.33
	Wine	88.89	88.89	<b>100.00</b>	88.89	55.56	55.56	44.44
	Zoo	66.67	66.67	<b>83.33</b>	50.00	50.00	50.00	50.00
Rank mean	2.71	3.50	<b>2.07</b>	2.36	6.00	5.79	5.57	
FART	$\chi = 30.6, k= 6, p\text{-value} < 0.01$							
Post hoc	ab	b	a	ab	c	c	c	
RF	Glass	87.50	87.50	<b>100.00</b>	<b>100.00</b>	87.50	87.50	87.50
	Ionosphere	64.71	64.71	<b>82.35</b>	58.82	41.18	35.29	29.41
	Sonar	71.43	75.00	<b>78.57</b>	67.86	32.14	32.14	57.14
	Vehicle	70.00	70.00	70.00	<b>80.00</b>	50.00	50.00	60.00
	WDBC	69.23	61.54	53.85	<b>84.62</b>	46.15	46.15	46.15
	Wine	62.50	<b>75.00</b>	<b>75.00</b>	<b>75.00</b>	50.00	50.00	37.50
	Zoo	57.14	57.14	<b>85.71</b>	71.43	42.86	42.86	42.86
Rank mean	3.29	3.00	<b>1.93</b>	2.21	5.79	5.93	5.86	
FART	$\chi = 32.5, k= 6, p\text{-value} < 0.01$							
Post hoc	b	ab	a	ab	c	c	c	
SVM	Glass	66.67	33.33	<b>100.00</b>	<b>100.00</b>	0.00	33.33	0.00
	Ionosphere	71.43	64.29	57.14	<b>78.57</b>	42.86	42.86	42.86
	Sonar	55.00	<b>55.00</b>	45.00	50.00	35.00	35.00	45.00
	Vehicle	84.62	76.92	92.31	<b>100.00</b>	69.23	69.23	76.92
	WDBC	<b>66.67</b>	50.00	<b>66.67</b>	<b>66.67</b>	33.33	41.67	41.67
	Wine	71.43	71.43	71.43	<b>85.71</b>	57.14	57.14	57.14
	Zoo	62.50	62.50	50.00	<b>100.00</b>	62.50	75.00	62.50
Rank mean	2.71	3.57	3.43	<b>1.50</b>	6.14	5.29	5.36	
FART	$\chi = 27.2, k= 6, p\text{-value} < 0.01$							
Post hoc	ab	b	b	a	c	C	c	

Finalmente, se realizó un análisis general de los resultados obtenidos por cada uno de los métodos de importancia de variables, teniendo en cuenta tanto los seis MAS como los siete datasets. La **Tabla 11** muestra que en la comparación realizada sobre AMIV, para niveles de significancia de 0.05, PSO y GA no se dominan entre sí, pero para un nivel de significancia de 0.10, PSO domina a GA, y para cualquier nivel de significancia ambos dominan al resto de medidas para importancia de variables, adicionalmente CA e ICA dominan a RFVI, GR e IG.

Tabla 11. Prueba de Friedman para la comparación de AMIV y su post hoc sobre la totalidad de los resultados.

		AMIV						
	Rank Mean	(1)	(2)	(3)	(4)	(5)	(6)	(7)
PSO (1)	1.73	-		●	●	●	●	●
GA (2)	2.18	○	-	●	●	●	●	●
CCA (3)	3.38	○	○	-		●	●	●
ICA (4)	3.45	○	○		-	●	●	●
RFVI (5)	4.81	○	○	○	○	-	●	●
GR (6)	6.21	○	○	○	○	○	-	
IG (7)	6.24	○	○	○	○	○		-

● = El método en la fila mejora el método de la columna., ○ = El método en la columna mejora el método de la fila. Diagonal superior, nivel de significancia 0.05 Diagonal inferior, nivel de significancia 0.10.

Por otro lado, en la **Tabla 12** sobre las comparaciones respecto a COBESV y para cualquier nivel de significancia, PSO domina a todas las medidas de importancia y nuevamente obtuvo el primer puesto, sin embargo, esta vez CCA queda en segundo lugar, siendo comparable con GA e ICA y estos tres dominando a RFVI, GR e IG.

Tabla 12. Prueba de Friedman para la comparación de COBESV y su post hoc sobre la totalidad de los resultados.

		COBESV						
	Rank Mean	(1)	(2)	(3)	(4)	(5)	(6)	(7)
PSO (1)	2.27	-	●	●	●	●	●	●
CCA (2)	2.94	○	-			●	●	●
GA (3)	3.04	○		-		●	●	●
ICA (4)	3.08	○			-	●	●	●
RFVI (5)	5.40	○	○	○	○	-		
GR (6)	5.46	○	○	○	○		-	
IG (7)	5.80	○	○	○	○			-

---

● = El método en la fila mejora el método de la columna., ○ = El método en la columna mejora el método de la fila. Diagonal superior, nivel de significancia 0.05 Diagonal inferior, nivel de significancia 0.10.

La importancia de variables estimada a partir de PSO, GA, CCA e ICA, es destacada respecto a las otras medidas de importancia de variables, lo cual se puede explicar por el hecho de que el método propuesto se basa en el efecto que tiene la presencia de cada variable de entrada sobre distintas combinaciones evaluadas dentro del wrapper lo cual puede captar un efecto multivariado. Además, debido a que los wrappers involucran en su cálculo una medida de desempeño o calidad (exactitud en este caso) sobre los diferentes subconjuntos, se logran mayores garantías de que las primeras variables de entrada sean las que más influyen y aporten en la clasificación. La efectividad del PSO y GA sobre el CCA y el ICA probablemente se debe al efecto positivo que tiene la estrategia de explotación de los dos primeros que les permite enfocarse en soluciones prometedoras, manteniendo variables informativas fijas y evaluando el efecto que tienen al variar el resto, no obstante esto como se observó en la **Tabla 9** y en la **Tabla 10**, puede ser más evidente dependiendo del MAS que se esté utilizando.

Respecto al algoritmo propuesto para definir la importancia de variables, se precisa mencionar nuevamente que se puede implementar de manera general para cualquier wrapper, lo que le permitiría trabajar bien con resultados de diversas metaheurísticas o algoritmos de selección de atributos siempre y cuando se le provean las entradas apropiadas.

Finalmente, en este capítulo se propuso una medida de importancia de variables basada en los resultados de un wrapper. Esta medida calcula la diferencia del desempeño entre las soluciones donde aparece cada variable respecto a las soluciones donde la misma variable está ausente, permitiendo tener una medida de relevancia dada por su contribución al MAS y el efecto de interacción con otras variables. Esta propuesta se caracteriza por adaptarse a cualquier wrapper y no únicamente al propuesto en esta investigación el cual está basado en CAs, también tiene la ventaja de establecer la medida de importancia de variables dependiendo del MAS utilizado, además al ser comparada con otros algoritmos del estado del arte, los resultados sugieren que las primeras variables acumulan una mayor cantidad de exactitud y coinciden con los resultados de las mejores soluciones encontradas en selección de atributos. Con lo presentado aquí y en el capítulo anterior, se resuelve la pregunta de investigación.

## 5.4 Aplicaciones

Una de las grandes ventajas de utilizar CAs en selección de atributos frente a otros métodos, es la facilidad que tienen de generar resultados rápidos, esto debido a que fácilmente se pueden ejecutar en paralelo y al sacrificar algo de exactitud es posible ganar velocidad en obtención de resultados, lo cual es vital en modelos programados para proveer decisiones en tiempo real, donde la rapidez de la respuesta es un factor clave [112]. También el enfoque con CAs es útil, al analizar modelos de alta dimensionalidad, donde se obtienen datasets con un elevado número de variables (más de 1000) y muchas de ellas están correlacionadas o no aportan información relevante, en este caso un wrapper clásico puede ser muy costoso computacionalmente, mientras los CAs pueden generar resultados en menos tiempo. La importancia de variables también tiene numerables aplicaciones, y la característica especial del algoritmo propuesto, es que abre la posibilidad de generar una importancia de variables distinta para cada MAS, permitiendo reconocer los factores y tipos de relaciones (Lineal, no lineal o interacciones) a los que cada MAS les está dando más peso, este resultado es de interés para investigadores ya sea para controlar las variables más importantes, descartar aquellas que tienen un alto costo y poca relevancia o incluso seleccionar un MAS que se ajuste a sus propósitos de análisis [7].

La aplicación de los métodos propuestos se extiende por todos los campos de la ciencia, en especial cuando el objetivo es crear modelos predictivos y analizar las variables más importantes, dado que muchas veces se mide un número elevado de variables por primera vez y no todas son de igual relevancia. Algunas áreas con potencial uso de los métodos planteados son: la minería de texto, donde lo que se busca usualmente es extraer palabras relevantes que pueden ayudar a clasificar blogs, tweets, notas de prensa, entre otros, con el objetivo de definir temáticas, evaluar opinión o incluso extraer resúmenes, por tal razón, descartar términos o palabras que no aporten al objetivo, es una tarea vital antes de crear un modelo definitivo. También en el campo de reconocimiento de patrones en imágenes, por ejemplo, en el reconocimiento de rostros, se utilizan distintas mediciones a partir de numerosos píxeles los cuales pueden identificar la coincidencia con cierta persona a partir de una fotografía, de allí filtrar los patrones de mayor interés e importancia, puede reducir el tiempo en el que el algoritmo tarda en distinguir una persona. Otro ejemplo es en la agricultura, donde se busca a partir de modelos predictivos explicar el rendimiento con base en variables relacionadas con clima, suelo y manejo agronómico, algunas mediciones pueden implicar un costo elevado, tal como el uso

de sensores, análisis de suelo o frecuentes monitoreos en el cultivo, y considerando que no todas las variables siempre tienen el mismo aporte o la misma importancia, es útil dar prioridad solo al conjunto más importante. Finalmente, otros campos de la ciencia como la predicción de clima, las tendencias de valor de la moneda de un país, la bioinformática o el reconocimiento de sonido, dependen de múltiples factores y la mayoría de veces la relevancia de dichos factores es desconocida, por lo tanto, pueden beneficiarse de esta propuesta de esta investigación.

# 6 Conclusiones y trabajo futuro

## 6.1 Conclusiones

En este trabajo se implementó un nuevo método para selección de atributos a través de un enfoque de wrapper que basa su estrategia de búsqueda en CAs binarios donde cada fila representa un subconjunto de variables seleccionadas. La experimentación se realizó utilizando dos tipos de CAs siendo estos los CAs acumulados (CCA) y los CAs incrementales (ICA). Las pruebas realizadas demuestran que los CAs aparte de ser útiles para realizar test de software como tradicionalmente se han utilizado, también pueden aportar en los procesos de selección de atributos.

Los experimentos se realizaron utilizando siete conjuntos de datos conocidos para tareas de selección de atributos y seis MAS, a saber: C4.5, KNN, NB, MLP, RF, y SVM. Los resultados mostraron que tanto el MAS como las características del conjunto de variables influyen sobre el resultado que produce el wrapper propuesto, tal fue el caso de KNN conocido por ser sensible a variables redundantes e irrelevantes y logra una mejora hasta del 25% en exactitud, mientras que RF al tener incorporado un método de selección de atributos muestra una muy pequeña mejora siendo la máxima de 2%.

De las pruebas realizadas sobre los CCA e ICA, los resultados que garantizan mayor exactitud y menor número de variables son los arreglos de cobertura con fuerza seis, y ambos wrappers pueden obtener resultados cercanos, no obstante ICA requiere menos EFOs y la diferencia respecto a los CCA se hace más evidente en la medida que el conjunto de datos tiene un mayor número de variables.

Los wrappers propuestos también fueron comparados con dos algoritmos basados en población (GA y PSO). Se realizó un análisis estadístico no paramétrico para contrastar las diferencias, en el que se evidenció que todos los wrappers son comparables en términos de número de variables escogidas, sin embargo, en exactitud, los wrappers propuestos no alcanzan los resultados obtenidos por el PSO siendo este el que alcanza mayor exactitud.

A partir de la evaluación del número de EFOs frente al incremento en la exactitud, los resultados mostraron que el número de filas de un CCA cuatro y seis respectivamente, puede ser un indicador del número de pruebas que tarda un wrapper basado en GA y PSO en alcanzar su máximo valor de exactitud, respectivamente.

Con base en el análisis de las diferencias entre el wrapper de PSO con los wrappers de CCA e ICA se observó que más del 90% de los casos tienen una diferencia en exactitud menor a 2%, sugiriendo que la pérdida en exactitud podría ser compensada respecto al PSO, en términos del costo que implica afinar los parámetros del algoritmo y la posibilidad de hacer una mayor cantidad de ejecuciones en paralelo, a la vez que se reduce el tiempo en alcanzar una solución.

Con este trabajo se muestra las cualidades de los CAs en un nuevo campo que es el de la estimación de importancia de variables, esto a partir de un algoritmo propuesto que utiliza los datos recopilados del desempeño de las soluciones que evalúa cualquier wrapper, entre ellos uno basado en CAs, permitiendo medir la importancia a partir del rendimiento que tuvo el MAS evaluado en presencia versus ausencia de cada una de las variables de entrada.

La medida propuesta para importancia de variables proporciona un valor agregado al esfuerzo computacional de cualquier wrapper y su costo es insignificante en comparación con el esfuerzo mismo realizado por el wrapper. Además, contribuye a la interpretación del MAS, al establecer un ranking para definir un orden de las variables de entrada más influyentes, permitiendo la posibilidad de generar una importancia de variables propia por cada MAS.

La importancia de las variables calculada con fuerza seis en comparación con fuerzas menores, retienen una mayor cantidad de exactitud en las primeras variables más relevantes y coinciden con las mejores soluciones encontradas por los wrappers.

La nueva propuesta de importancia de variables basada en wrappers CCA e ICA, mostró ser superior a los filtros de IG y GR y en la mayoría de los casos a RFVI. Esto por (i) mantener un subconjunto de variables de entrada con mayor exactitud en las primeras posiciones y (ii) mantener una coincidencia entre las variables de entrada más relevantes y la mejor solución encontrada en la selección de atributos.

Sin embargo, PSO y GA en ambos indicadores en ocasiones dominan a los métodos propuestos y en ocasiones los igualan.

## 6.2 Trabajo Futuro

Como trabajo futuro se propone mezclar la capacidad de los CAs para explorar, con algoritmos exactos basados en la importancia de las variables, heurísticas (tal como Hill Climbing) o metaheurísticas que exploten las mejores soluciones encontradas con los CAs. También se espera proponer un indicador de las interacciones de las variables frente a la exactitud encontrada por cada solución, teniendo en cuenta la efectividad que logran los CAs de fuerza seis frente a lo que se puede lograr con pruebas exhaustivas.

Además, se propone evaluar el efecto que tiene la medida de importancia de variables propuesta, sobre diversos subconjuntos de variables que tengan correlación, escalas discretas y escalas nominales. También se pretende realizar pruebas sobre otros MAS e incluir en la comparación otras medidas de importancia de variables. Las pruebas se realizaron con exactitud, pero se pueden emplear otras medidas de desempeño como precisión, recuerdo, medida  $F$  o índice Kappa.

En el futuro también se propone evaluar el wrapper, pero a través de una función multiobjetivo con dos propósitos, que maximice el desempeño del modelo y que reduzca el número de variables al mismo tiempo.

## 7 Referencias

- [1] C. L. Philip Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data," *Inf. Sci. (Ny)*, vol. 275, pp. 314–347, 2014.
- [2] H. Wang, Z. Xu, H. Fujita, and S. Liu, "Towards felicitous decision making : An overview on challenges and trends of Big Data," *Inf. Sci. (Ny)*, vol. 367–368, pp. 747–765, 2016.
- [3] T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning," *Springer Ser. Stat.*, vol. 27, no. 2, pp. 83–85, 2009.
- [4] G. I. Webb, M. J. Pazzani, and D. Billsus, "Machine Learning for User Modeling," no. 1978, pp. 19–29, 2001.
- [5] B. Z. Dadaneh, H. Y. Markid, and A. Zakerolhosseini, "Unsupervised probabilistic feature selection using ant colony optimization," *Expert Syst. Appl.*, vol. 53, pp. 27–42, 2016.
- [6] L. Cui, Z. Lu, P. Wang, and W. Wang, "The ordering importance measure of random variable and its estimation," *Math. Comput. Simul.*, vol. 105, pp. 132–143, 2014.
- [7] P. Wei, Z. Lu, and J. Song, "Variable importance analysis: A comprehensive review," *Reliab. Eng. Syst. Saf.*, vol. 142, pp. 399–432, 2015.
- [8] J. Tang, S. Alelyani, and H. Liu, "Feature Selection for Classification: A Review," *Data Classif. Algorithms Appl.*, pp. 37–64, 2014.
- [9] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 16–28, 2014.
- [10] H. Shahamat and A. A. Pouyan, "Feature selection using genetic algorithm for classification of schizophrenia using fMRI data," vol. 3, no. 1, pp. 30–37, 2015.
- [11] V. Kothari, J. Anuradha, S. Shah, and P. Mittal, "A Survey on Particle Swarm Optimization," pp. 192–201, 2012.
- [12] V. Susheela, "Class Specific Feature Selection Using Simulated Annealing," *Min. Intell. Knowl. Explor.*, pp. 12–21, 2015.
- [13] G. Tzanakis, L. Moura, D. Panario, and B. Stevens, "Constructing new covering arrays from LFSR sequences over finite fields," *Discrete Math.*, vol. 339, no. 3, pp. 1158–1171, 2016.
- [14] N. Tebal, S. P. Selatan, and P. Pinang, "A Review of Covering Arrays and Their Application to Software Testing Bestoun S . Ahmed and Kamal Z . Zamli Software Engineering Group , School of Electrical and Electronic Engineering ," vol. 7, no. 9, pp. 1375–1385, 2011.
- [15] A. Hartman and L. Raskin, "Problems and algorithms for covering arrays," *Discrete Math.*, vol. 284, no. 1–3, pp. 149–156, 2004.
- [16] C. Yilmaz, M. B. Cohen, and A. A. Porter, "Covering arrays for efficient fault characterization in complex configuration spaces," *ACM SIGSOFT Softw. Eng. Notes*, vol. 32, no. 4, pp. 45–54, 2006.
- [17] W. Dubitzky, O. Wolkenhauer, K.-H. Cho, and H. Yokota, "Feature Selection, Encyclopedia of Systems Biology," no. 1, 2013, pp. 1889–2113.

- [18] I. Guyon and a Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.
- [19] A. Jovi, K. Brki, N. Bogunovi, A. Jovic, K. Brkic, and N. Bogunovic, *A review of feature selection methods with applications*, no. May. 2015.
- [20] J. Novaković, P. Strbac, and D. Bulatović, "Toward optimal feature selection using ranking methods and classification algorithms," *Yugosl. J. Oper. Res.*, vol. 21, no. 1, pp. 2334–6043, 2011.
- [21] T. A. Alhaj, M. M. Siraj, A. Zainal, H. T. Elshoush, and F. Elhaj, "Feature selection using information gain for improved structural-based alert correlation," *PLoS One*, vol. 11, no. 11, pp. 1–18, 2016.
- [22] J. Dai and Q. Xu, "Attribute selection based on information gain ratio in fuzzy rough set theory with application to tumor classification," *Appl. Soft Comput. J.*, vol. 13, no. 1, pp. 211–221, 2013.
- [23] A. Nagpal and D. Gaur, "A New Proposed Feature Subset Selection Algorithm Based on Maximization of Gain Ratio," in *International Conference on Big Data Analytics*, 2015, pp. 181–197.
- [24] B. Jiang, X. Ding, L. Ma, and Y. He, "A hybrid feature selection algorithm: Combination of symmetrical uncertainty and genetic algorithms," *Second Int. ....*, pp. 152–157, 2008.
- [25] S. S. Kannan and N. Ramaraj, "A novel hybrid feature selection via Symmetrical Uncertainty ranking based local memetic search algorithm," *Knowledge-Based Syst.*, vol. 23, no. 6, pp. 580–585, 2010.
- [26] H. Liu and H. Motoda, *Computational methods of feature selection*. CRC Press, 2007.
- [27] Y.-H. Li, "Text feature selection algorithm based on Chi-square rank correlation factorization," *J. Interdiscip. Math.*, vol. 20, no. 1, pp. 153–160, 2017.
- [28] I. H. Witten, E. Frank, and M. a. Hall, *Data Mining: Practical Machine Learning Tools and Techniques, Third Edition*, vol. 54, no. 2. 2011.
- [29] L. Yu and H. Liu, "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution," *Int. Conf. Mach. Learn.*, pp. 1–8, 2003.
- [30] S. B. Kotsiantis, "Feature selection for machine learning classification problems : a recent overview," 2011.
- [31] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, no. 1–2, pp. 273–324, 1997.
- [32] D. W. Aha and R. L. Bankert, "A comparative evaluation of sequential feature selection algorithms," in *Learning from data*, Springer, 1996, pp. 199–206.
- [33] N. El Aboudi and L. Benhlila, "Review on wrapper feature selection approaches," in *Engineering & MIS (ICEMIS), International Conference on*, 2016, pp. 1–5.
- [34] S. Maldonado, R. Weber, and F. Famili, "Feature selection for high-dimensional class-imbalanced data sets using Support Vector Machines," *Inf. Sci. (Ny)*, vol. 286, pp. 228–246, 2014.
- [35] Z. Yin and J. Zhang, "Operator functional state classification using least-square support vector machine based recursive feature elimination technique," *Comput. Methods Programs Biomed.*, vol. 113, no. 1, pp. 101–

- 115, 2014.
- [36] U. Vishwakarma and P. A. Jain, "Reduces Unwanted Attribute in Intruder File Based on Feature Selection and Feature Reduction Using ID3 Algorithm," vol. 5, no. 1, pp. 896–900, 2014.
- [37] L. Rokach and O. Maimon, *Data mining with decision trees: theory and applications*. World scientific, 2014.
- [38] V. Molano, C. Cobos, M. Mendoza, E. Herrera-Viedma, and M. Manic, "Feature selection based on sampling and C4.5 algorithm to improve the quality of text classification using naive bayes," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8856, 2014.
- [39] R. Li and H. Cui, *Variable Selection via Regularization*. Wiley Online Library, 1985.
- [40] P. Zhao and B. Yu, "On Model Selection Consistency of Lasso," *J. Mach. Learn. Res.*, vol. 7, pp. 2541–2563, 2006.
- [41] C. Cui and D. Wang, "High dimensional data regression using Lasso model and neural networks with random weights," *Inf. Sci. (Ny)*, vol. 372, pp. 505–517, 2016.
- [42] H. Zou and T. Hastie, "Regularization and variable selection via the elastic-net," *J. R. Stat. Soc.*, vol. 67, no. 2, pp. 301–320, 2005.
- [43] D. Jiménez *et al.*, "From observation to information: Data-driven understanding of on farm yield variation," *PLoS One*, vol. 11, no. 3, pp. 1–20, 2016.
- [44] S. Delerce *et al.*, "Assessing Weather-Yield Relationships in Rice at Local Scale Using Data Mining Approaches," *PLoS One*, vol. 11, no. 8, p. e0161620, 2016.
- [45] P. Polak *et al.*, "Cell-of-origin chromatin organization shapes the mutational landscape of cancer," *Nature*, vol. 518, no. 7539, pp. 360–364, 2015.
- [46] N. L. Afanador, T. N. Tran, and L. M. C. Buydens, "An assessment of the jackknife and bootstrap procedures on uncertainty estimation in the variable importance in the projection metric," *Chemom. Intell. Lab. Syst.*, vol. 137, pp. 162–172, 2014.
- [47] J. Timaná, C. A. Cobos-lozada, and J. Torres-jimenez, "Metaheuristic algorithms for building Covering Arrays : A review Algoritmos metaheurísticos para construir Covering Arrays : Revisión Algoritmos metaheurísticos para construir Covering Arrays :," *Rev. Fac. Ing.*, vol. 25, no. 43, pp. 31–45, 2016.
- [48] J. Torres-Jimenez, I. Izquierdo-Marquez, R. N. Kacker, and D. Richard Kuhn, "Tower of covering arrays," *Discret. Appl. Math.*, vol. 190–191, pp. 141–146, 2015.
- [49] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, no. 6, p. 617, 2009.
- [50] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Inf. Sci. (Ny)*,

- vol. 180, no. 10, pp. 2044–2064, 2010.
- [51] E. Theodorsson-Norheim, “Friedman and Quade tests: BASIC computer program to perform nonparametric two-way analysis of variance and multiple comparisons on ranks of several related samples,” *Comput. Biol. Med.*, vol. 17, no. 2, pp. 85–99, 1987.
- [52] M. Bennasar, Y. Hicks, and R. Setchi, “Feature selection using Joint Mutual Information Maximisation,” *Expert Syst. Appl.*, vol. 42, no. 22, pp. 8520–8532, 2015.
- [53] H. Yang and J. Moody, “Feature selection based on joint mutual information,” in *Proceedings of international ICSC symposium on advances in intelligent data analysis*, 1999, pp. 22–25.
- [54] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, 2005.
- [55] J. Wang, J. Wei, Z. Yang, and S. Wang, “Feature Selection by Maximizing Independent Classification Information,” vol. 4347, no. 6, pp. 1–14, 2017.
- [56] J. Liu, Y. Lin, M. Lin, S. Wu, and J. Zhang, “Feature selection based on quality of information,” *Neurocomputing*, vol. 225, no. June 2016, pp. 11–22, 2017.
- [57] D. Dheeru and E. Karra Taniskidou, “UCI Machine Learning Repository,” 2017. [Online]. Available: <http://archive.ics.uci.edu/ml>.
- [58] N. Abd-Alsabour, “A review on evolutionary feature selection,” in *Proceedings - UKSim-AMSS 8th European Modelling Symposium on Computer Modelling and Simulation, EMS 2014*, 2014, pp. 20–26.
- [59] K. Sastry, D. E. Goldberg, and G. Kendall, “Genetic algorithms,” in *Search methodologies*, Springer, 2014, pp. 93–117.
- [60] P. N. da Silva, A. Plastino, and A. A. Freitas, “A Novel Genetic Algorithm for Feature Selection in Hierarchical Feature Spaces,” *Proc. 2018 SIAM Int. Conf. Data Min.*, no. i, pp. 738–746, 2018.
- [61] R. C. Anirudha, R. Kannan, and N. Patil, “Genetic algorithm based wrapper feature selection on hybrid prediction model for analysis of high dimensional data,” in *Industrial and Information Systems (ICIIS), 2014 9th International Conference on*, 2014, pp. 1–6.
- [62] C.-F. Tsai, W. Eberle, and C.-Y. Chu, “Genetic algorithms in feature and instance selection,” *Knowledge-Based Syst.*, vol. 39, pp. 240–247, 2013.
- [63] S. Oreski and G. Oreski, “Expert Systems with Applications Genetic algorithm-based heuristic for feature selection in credit risk assessment,” *Expert Syst. Appl.*, vol. 41, no. 4, pp. 2052–2064, 2014.
- [64] O. Soufan, D. Klefogiannis, P. Kalnis, and V. B. Bajic, “DWFS: A wrapper feature selection tool based on a parallel Genetic Algorithm,” *PLoS One*, vol. 10, no. 2, 2015.
- [65] A. S. Ghareb, A. A. Bakar, and A. R. Hamdan, “Hybrid feature selection based on enhanced genetic algorithm for text categorization,” vol. 49, pp. 31–47, 2016.
- [66] A. K. Das, S. Das, and A. Ghosh, “Ensemble feature selection using bi-objective genetic algorithm,” *Knowledge-Based Syst.*, vol. 123, pp. 116–127, 2017.

- [67] S. Sengupta and A. K. Das, "Dimension reduction using clustering algorithm and rough set theory," in *International Conference on Swarm, Evolutionary, and Memetic Computing*, 2012, pp. 705–712.
- [68] A. K. Das, S. K. Pati, and S. Chakrabarty, "Reduct generation of microarray dataset using rough set and graph theory for unsupervised learning," in *Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology*, 2012, pp. 555–561.
- [69] J. Lee and D. W. Kim, "Feature selection for multi-label classification using multivariate mutual information," *Pattern Recognit. Lett.*, vol. 34, no. 3, pp. 349–357, 2013.
- [70] X. Liu and L. Shang, "A fast wrapper feature subset selection method based on binary particle swarm optimization," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*, 2013, pp. 3347–3353.
- [71] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1656–1671, 2013.
- [72] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," *1997 IEEE Int. Conf. Syst. Man, Cybern. Comput. Cybern. Simul.*, vol. 5, pp. 4104–4108, 1997.
- [73] Y. Lu, M. Liang, Z. Ye, and L. Cao, "Improved particle swarm optimization algorithm and its application in text feature selection," *Appl. Soft Comput.*, vol. 35, pp. 629–636, 2015.
- [74] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms," *Appl. Soft Comput.*, vol. 18, pp. 261–276, 2014.
- [75] P. Moradi and M. Gholampour, "A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy," *Appl. Soft Comput.*, vol. 43, pp. 117–130, 2016.
- [76] T. Butler-Yeoman, B. Xue, and M. Zhang, "Particle swarm optimisation for feature selection: A hybrid filter-wrapper approach," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*, 2015, pp. 2428–2435.
- [77] P. Ghamisi and J. A. Benediktsson, "Feature selection based on hybridization of genetic algorithm and particle swarm optimization," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 309–313, 2015.
- [78] Y. Zhang, D. Gong, and J. Cheng, "Multi-Objective Particle Swarm Optimization Approach for Cost-Based Feature Selection in Classification," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 14, no. 1, pp. 64–75, 2017.
- [79] T. M. Hamdani, J.-M. Won, A. M. Alimi, and F. Karray, "Multi-objective feature selection with NSGA II," in *International Conference on Adaptive and Natural Computing Algorithms*, 2007, pp. 240–247.
- [80] I. Boussaid, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Inf. Sci. (Ny)*, vol. 237, pp. 82–117, 2013.
- [81] R. Y. M. Nakamura, L. A. M. Pereira, D. Rodrigues, K. A. P. Costa, J. P. Papa, and X.-S. Yang, "Binary bat algorithm for feature selection," in *Swarm Intelligence and Bio-Inspired Computation*, Elsevier, 2013, pp. 225–237.
- [82] B. Chen, L. Chen, and Y. Chen, "Efficient ant colony optimization for image

- feature selection,” *Signal Processing*, vol. 93, no. 6, pp. 1566–1576, 2013.
- [83] S. Das, P. K. Singh, S. Bhowmik, R. Sarkar, and M. Nasipuri, “A Harmony Search Based Wrapper Feature Selection Method for Holistic Bangla word Recognition,” *Procedia - Procedia Comput. Sci.*, vol. 89, pp. 395–403, 2016.
- [84] W. A. H. Ghanem and A. Jantan, “Novel Multi-Objective Artificial Bee Colony Optimization for Wrapper Based Feature Selection in Intrusion Detection.,” *Int. J. Adv. Soft Comput. Its Appl.*, vol. 8, no. 1, 2016.
- [85] A. L. Buczak and E. Guven, “A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection,” vol. 18, no. 2, pp. 1153–1176, 2016.
- [86] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [87] C. Strobl, A. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, “Conditional variable importance for random forests,” *BMC Bioinformatics*, vol. 11, pp. 1–11, 2008.
- [88] T. Hothorn, K. Hornik, A. Zeileis, and K. Hornik, “Unbiased Recursive Partitioning: A Conditional Inference Framework,” *J. Comput. Graph. Stat.*, vol. 15, no. 3, pp. 651–674, 2006.
- [89] X. Giam and J. D. Olden, “A new  $R^2$ -based metric to shed greater insight on variable importance in artificial neural networks,” *Ecol. Modell.*, vol. 313, pp. 307–313, 2015.
- [90] Y. H. Yun, B. C. Deng, D. S. Cao, W. T. Wang, and Y. Z. Liang, “Variable importance analysis based on rank aggregation with applications in metabolomics for biomarker discovery,” *Anal. Chim. Acta*, vol. 911, no. June 2015, pp. 27–34, 2016.
- [91] V. Pihur, S. Datta, and S. Datta, “RankAggreg, an R package for weighted rank aggregation,” *BMC Bioinformatics*, vol. 10, no. 1, p. 62, 2009.
- [92] S. Lin, J. Ding, and J. Zhou, “Rank aggregation of putative microrna targets with cross-entropy monte carlo methods,” in *Preprint, presented at the IBC 2006 conference, Montral, 2006*.
- [93] N. Akarachantachote, S. Chadcham, and K. Saithanu, “Variable importance index based on the partial least squares and boxplot cutoff threshold for variable selection,” *Int. J. Data Anal. Tech. Strateg.*, vol. 9, no. 1, pp. 34–45, 2017.
- [94] R Core Team, “R: A Language and Environment for Statistical Computing.” Vienna, Austria, 2016.
- [95] J. Knaus, “snowfall: Easier cluster computing (based on snow).” 2015.
- [96] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2009.
- [97] M. K. C. from Jed Wing *et al.*, “caret: Classification and Regression Training.” 2016.
- [98] S.-W. Lin, Z.-J. Lee, S.-C. Chen, and T.-Y. Tseng, “Parameter determination of support vector machine and feature selection using simulated annealing approach,” *Appl. Soft Comput.*, vol. 8, no. 4, pp. 1505–1512, 2008.
- [99] J. Torres-Jimenez, “Laboratorio de Tecnologías de la información del Centro de Investigación y de Estudios Avanzados (CINVESTAV).” Mexico, Tamaulipas.

- [100] D. R. Kuhn, D. R. Wallace, and A. M. Gallo, "Software fault interactions and implications for software testing," *IEEE Trans. Softw. Eng.*, vol. 30, no. 6, pp. 418–421, 2004.
- [101] J. A. SáEz, J. Luengo, and F. Herrera, "Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification," *Pattern Recognit.*, vol. 46, no. 1, pp. 355–364, 2013.
- [102] C. J. Mantas and J. Abellán, "Credal decision trees in noisy domains.," in *ESANN*, 2014.
- [103] E. Tuv, "Feature Selection with Ensembles , Artificial Variables , and Redundancy Elimination," *J. Mach. Learn. Res.*, vol. 10, pp. 1341–1366, 2009.
- [104] D. D. Lewis, "Naive (Bayes) at forty: The independence assumption in information retrieval," in *Machine Learning: ECML-98: 10th European Conference on Machine Learning Chemnitz, Germany, April 21--23, 1998 Proceedings*, C. Nédellec and C. Roudier, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 4–15.
- [105] B. B. Averbeck, P. E. Latham, and A. Pouget, "Neural correlations, population coding and computation," *Nat. Rev. Neurosci.*, vol. 7, no. 5, p. 358, 2006.
- [106] S. Halkjær and O. Winther, "The effect of correlated input data on the dynamics of learning," in *Advances in neural information processing systems*, 1997, pp. 169–175.
- [107] S. Maldonado and R. Weber, "A wrapper method for feature selection using support vector machines," *Inf. Sci. (Ny)*, vol. 179, no. 13, pp. 2208–2217, 2009.
- [108] C.-W. Hsu, C.-C. Chang, C.-J. Lin, and others, "A practical guide to support vector classification," pp. 1–16, 2003.
- [109] K. J. Archer and R. V Kimes, "Empirical characterization of random forest variable importance measures," *Comput. Stat. Data Anal.*, vol. 52, no. 4, pp. 2249–2260, 2008.
- [110] Y. Wan, M. Wang, Z. Ye, and X. Lai, "A feature selection method based on modified binary coded ant colony optimization algorithm," *Appl. Soft Comput.*, vol. 49, pp. 248–258, 2016.
- [111] A. Zarshenas and K. Suzuki, "Binary coordinate ascent: An efficient optimization technique for feature subset selection for machine learning," *Knowledge-Based Syst.*, vol. 110, pp. 191–201, 2016.
- [112] Q. Shi and M. Abdel-Aty, "Big data applications in real-time traffic operation and safety monitoring and improvement on urban expressways," *Transp. Res. Part C Emerg. Technol.*, vol. 58, pp. 380–394, 2015.

## 8 Anexos

Los anexos fueron almacenados en forma digital y se encuentran en el CD incluido en la monografía. Se encuentra estructurado de la siguiente forma:

- **Anexo 1:** Carpeta Funciones\_wrapper\_impvar, Código fuente del wrapper y la medida de importancia de variables documentados en R.
- **Anexo 2:** Artículo que será sometido a una revista indexada JCR (Journal Citation Reports) cuartil 1 o 2 con la siguiente referencia preliminar: Dorado H, Cobos C., Torres-Jimenez J., Burra D., Herrera-Viedma E., "Wrapper for building classification models using Covering Arrays".
- **Anexo 3:** Artículo publicado en la revista Lecture Notes in Computer Science, indexada en la categoría A2 del PUBLINDEX de Colciencias, que corresponde a las memorias del evento MIKE 2018 - Mining Intelligence and Knowledge Exploration (MIKE) realizado en Cluj, Rumania, Diciembre del 2018. El artículo tiene la siguiente referencia: Dorado H., Cobos C., Torres-Jiménez J., Jimenez D., Mendoza M. "*A proposal to estimate the variable importance measures in predictive models using results from a Wrapper*", LNCS, volume 11308, DOI: 10.1007/978-3-030-05918-7\_33.
- **Anexo 4:** Artículo publicado en la revista Lecture Notes in Computer Science, indexada en la categoría A2 del PUBLINDEX de Colciencias, que corresponde a las memorias del evento MICA I 2018 - Mexican International Conference on Artificial Intelligence que se realizó del 22 al 27 de octubre en Guadalajara México. El artículo tiene la siguiente referencia: Dorado H, Delerece S., Jimenez D., Cobos C. "*Finding optimal farming practices to increase crop yield through Global-best Harmony Search and predictive models, a data-driven approach*", LNAI Volume 11289, DOI: 10.1007/978-3-030-04497-8\_2.
- **Anexo 5:** Carpeta Pruebas\_importancia\_de\_variables con los resultados de las pruebas de importancia de variables realizados a todos los conjuntos de datos.