

# Optimización de Support Vector Machine mediante metaheurísticas para clasificación de retinopatía diabética

Matias G. Rojas<sup>1</sup>, Jessica A. Carballido<sup>2,4</sup>  
Ana C. Olivera<sup>1,3,4</sup> y Pablo J. Vidal<sup>1,3,4</sup>

<sup>1</sup> Instituto para las Tecnologías de la Información y las Comunicaciones, *Universidad Nacional de Cuyo*, Mendoza, Argentina

<sup>2</sup> Instituto de Ciencias e Ingeniería de la Computación, *Universidad Nacional del Sur*, Bahía Blanca, Argentina

<sup>3</sup> Universidad Nacional de Cuyo, *Facultad de Ingeniería*, Mendoza, Argentina

<sup>4</sup> Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina

**Abstract.** La máquina de vectores de soporte (SVM, *Support Vector Machine*) es una técnica de clasificación de aprendizaje supervisado, ampliamente utilizada en diferentes campos de aplicación. SVM busca identificar el mejor hiperplano que permita lograr una separación clara entre un conjunto de datos. Desafortunadamente, el uso práctico de SVM está limitado por la calidad de la configuración de sus hiperparámetros, que tienen un impacto directo sobre su rendimiento en clasificación y generalización. Por lo tanto, es necesario desarrollar un enfoque efectivo y rápido para determinar los valores de estos hiperparámetros que lleven a una clasificación eficiente y confiable. Este trabajo presenta la evaluación de cuatro metaheurísticas, en la labor de optimizar los hiperparámetros de un SVM que utiliza un kernel Wavelet. Las evaluaciones se realizaron sobre un conjunto de datos relacionados a la enfermedad retinopatía diabética. A partir de los resultados del experimento, se puede concluir que el uso de técnicas metaheurísticas para optimizar los hiperparámetros puede ayudar a mejorar la capacidad de clasificación y generalización del SVM.

**Keywords:** Bioinformática · Support Vector Machine · Clasificación · Optimización · Metaheurísticas · Retinopatía Diabética.

## 1 Introducción

Clasificar consiste en identificar la clase o categoría a la que pertenece una instancia de datos determinada, de acuerdo a las características que presenta. Su aplicación es de vital importancia en diferentes campos científicos debido a que brinda la posibilidad de distinguir entre tipos de objetos. Es así que por ejemplo, en la medicina se la utiliza para la clasificación de tumores y cánceres [7], en el procesamiento de imágenes para el reconocimiento de diferentes patrones [22], en la economía para la clasificación del riesgo de dificultades financieras [20], entre otros.

La máquina de vectores soporte (SVM, *Support Vector Machine*) es un modelo de aprendizaje automático supervisado para clasificación y análisis de regresión. Consiste en la identificación del hiperplano que mejor separe a dos o más clases de instancias pertenecientes a un conjunto de datos [25]. En la literatura, SVM ha demostrado una alta precisión en predicción y capacidad de modelado en una amplia gama de problemas de clasificación real, reconocimiento de patrones y regresión.

El rendimiento de un SVM, depende fuertemente del ajuste de sus hiperparámetros, los cuales inciden en la eficiencia de la clasificación. Por un lado está la constante de suavización de margen  $C$  que afecta el equilibrio entre la complejidad del modelo y la proporción de muestras no separables. Por otro lado, se tienen los pertenecientes a la función kernel, que varían de acuerdo a los datos a clasificar [3]. El ajuste de estos hiperparámetros es un problema de optimización complejo, que requiere muchos cálculos para encontrar la configuración que maximice la capacidad de generalización al mismo tiempo que se logre el máximo poder de predicción.

Existen en la literatura varios enfoques para resolver este problema. El más común es utilizar métodos deterministas, los cuales no garantizan encontrar una buena solución cuando el espacio de búsqueda es muy grande y cuenta con un gran número de óptimos locales. Esto motivó el uso de metaheurísticas, que mostraron una alta eficiencia en la generación de soluciones aceptables en este tipo de escenarios.

En este trabajo, se propone la utilización de metaheurísticas para encontrar la mejor configuración entre el hiperparámetro  $C$  y los hiperparámetros del kernel Wavelet [29]. Se busca evaluar el rendimiento de las metaheurísticas en la tarea de optimizar los hiperparámetros de SVM, a la vez que se observa el comportamiento de clasificación utilizando el kernel Wavelet. Se evalúa el comportamiento de las configuraciones obtenidas por las metaheurísticas con las entregadas por otras funciones kernel, incluido el Wavelet sin optimizar, con el propósito de evaluar la capacidad de mejora obtenida.

Este trabajo se organiza de la siguiente manera. En la sección 2, se introduce el modelo SVM y la función de sus hiperparámetros y el kernel a la hora de clasificar datos, definiendo formalmente el problema y presentando trabajos relacionados. En la sección 3 se presenta la propuesta de este trabajo y el kernel utilizado. En la sección 4, se detallan los experimentos realizados y los resultados experimentales para el caso de estudio seleccionado. Finalmente, en la sección 5, se describen las conclusiones y posibles trabajos futuros.

## 2 Support Vector Machine

La máquina de vectores soporte (SVM, *Support Vector Machine*) [25] es uno de los algoritmos de machine learning supervisado más recientes y uno de los más populares en la actualidad. Se basa en lograr la mejor separación entre instancias de datos, maximizando la distancia existente (margen) entre el hiperplano de separación y las instancias de entrenamiento más cercanas de

cada clase, conocidas como vectores soporte [28]. Una vez identificado el mejor hiperplano de separación, SVM se encuentra entrenado para clasificar nuevos ejemplos.

En lugar de enfocarse en reducir el error de entrenamiento como lo hacen otros algoritmos de clasificación, SVM se centra en minimizar el error de generalización mediante la ampliación de los márgenes entre el hiperplano de separación y las instancias, tomando como base al principio de minimización de riesgo estructural, propuesta en la teoría estadística de aprendizaje [30].

Si bien SVM en principio fue diseñado para trabajar con datos lineales, puede ser fácilmente extendido para clasificar datos no lineales mediante la utilización de funciones kernel, que mapean los datos de entrenamiento en un espacio lineal superior donde se los pueda separar mediante un hiperplano [16].

## 2.1 Influencia de los hiperparámetros en el rendimiento del SVM

Los hiperparámetros de SVM son la constante de suavización de margen  $C$  y los parámetros de la función kernel utilizada. Identificar la configuración correcta es fundamental en la clasificación realizada por el SVM, pudiendo ser la diferencia entre la obtención de buenos o malos resultados.

En la práctica es común encontrarse con instancias que pertenezcan a una clase sin tener similitudes claras con las otras instancias de esa clase. La constante de suavización de margen  $C$  contempla la existencia de estas instancias, permitiendo que el hiperplano de separación se genere teniéndolas en cuenta. En otras palabras, flexibiliza la separabilidad del hiperplano permitiendo que algunas instancias no lo respeten [11].

Por otro lado, no siempre las instancias de datos a clasificar son linealmente separables. En este caso, se utilizan las funciones kernel que realizan una transformación matemática sobre los datos, elevándolos a una dimensión más alta. Esto provoca una distribución que posibilita la identificación de un hiperplano de separación [24]. Sin embargo, la efectividad de la función kernel depende de los valores que tomen sus parámetros, por lo que resulta fundamental identificar aquellos que mejores resultados proporcionen. En [26], se demuestra que el parámetro  $\gamma$  afecta la amplitud de la distribución Gaussiana de los datos generada por el kernel Función de Base Radial (RBF, *Radial Basis Function*), indicando que dependiendo del valor que tome ese parámetro la generalización aumenta o disminuye.

## 2.2 Definición formal del problema

La configuración de SVM es un conjunto  $K$  de hiperparámetros de tipo real. El objetivo es identificar los valores de cada uno de los  $K$  hiperparámetros, que logre mejorar la capacidad de clasificación del SVM sobre un conjunto de datos determinado. La función objetivo presentada en la Ecuación 1, busca maximizar la exactitud de clasificación obtenida por el SVM utilizando la configuración de hiperparámetros aportada por un individuo  $x$  determinado.

$$aptitud(x) = \frac{Aciertos_x}{TotalInstancias_x} \quad (1)$$

Donde *Aciertos* es la cantidad de instancias bien clasificadas por el SVM con la configuración del individuo  $x$  y *TotalInstancias* es la cantidad de instancias sometidas al proceso de clasificación.

### 2.3 Trabajos Relacionados

Dado que los enfoques basados en metaheurísticas resultan prometedores para abordar el problema de la optimización de hiperparámetros de SVM, en los últimos años se ha observado el surgimiento de distintas estrategias.

En el trabajo desarrollado en [27], se presenta un Algoritmo Avanzado de Optimización de Ballenas para la identificación de valores óptimos de los hiperparámetros de SVM. Los resultados demostraron que este enfoque puede lograr valores altos de exactitud y una buena convergencia, comparado con otros algoritmos de optimización.

En [12] proponen la utilización de un algoritmo de Optimización de Enjambres Múltiples de Partículas para la selección de valores óptimos de los hiperparámetros de SVM, obteniendo resultados superiores a métodos tradicionales tales como Grid Search.

Por su parte, en [5] se implementa un Algoritmo Genético Multiobjetivo Autoajutable basado en el algoritmo NSGA-II, para la mejora de clasificación de SVM sobre conjuntos de datos de diagnóstico médico.

Los trabajos de optimización de hiperparámetros de SVM con Kernel Wavelet, se enfocan principalmente en problemas de aproximación y clasificación de señales continuas. Un ejemplo se da en [14], donde se propone la utilización de un algoritmo de Optimización de Enjambre de Partículas aplicado a un conjunto de datos de análisis de vibración de elementos.

Para nuestro conocimiento, no existe ningún método para la optimización de hiperparámetros de SVM utilizando el kernel Wavelet, aplicado al escenario de retinopatía diabética.

## 3 Optimización de hiperparámetros de SVM mediante metaheurísticas

En esta sección, se describe brevemente a las metaheurísticas utilizadas, se introduce a la función kernel Wavelet y se define como se compone el individuo utilizado por cada metaheurística.

### 3.1 Metaheurísticas consideradas

Para realizar las pruebas sobre el problema de optimización de hiperparámetros de SVM, se consideraron las versiones canónicas de cuatro metaheurísticas distintas.

**Algoritmo Genético (AG)** se basa en la teoría evolutiva de las especies. Comienza con la generación de una población de soluciones iniciales. Cada solución posee un conjunto de variables (genes) conocido como cromosoma, y un valor que indica que tan bueno es el individuo como solución al problema (valor de aptitud). Este algoritmo busca alcanzar el óptimo global de manera iterativa. Las iteraciones son conocidas como generaciones y al finalizar cada una se obtiene una nueva población producto de la evolución. En una generación la población es expuesta a tres operaciones básicas, la selección de individuos a ser sometidos a la evolución -aquellos con mayor valor de aptitud tienen mas probabilidad de ser seleccionados-, el cruzamiento que consiste en combinar dos soluciones intercambiando sus genes, y la mutación que consiste en alterar uno o mas genes del cromosoma de la solución [9].

**Evolución Diferencial (ED)** Es conocido como un algoritmo evolutivo debido a que está basado en poblaciones y utiliza métodos de selección, cruzamiento y mutación para evolucionarla. Comienza con la generación de una población inicial. Por cada solución de la población, se seleccionan tres soluciones padre de manera aleatoria, para que a partir de ellos y un factor de escalado se genere un nuevo valor para cada uno de sus genes con una probabilidad de mutación determinada. La solución resultante de este proceso reemplaza a la solución original si y solo si la mejora en el valor de aptitud. Este proceso se repite hasta alcanzar la condición de parada [21].

**Algoritmo Genético Celular (AGC)** Es una variante del AG caracterizada por tener una población descentralizada en la que las soluciones tentativas evolucionan en vecindarios superpuestos. Los individuos son ubicados en una malla toroidal bidimensional, y se aplican operadores genéticos teniendo en cuenta únicamente a los individuos cercanos (vecindario). En este contexto, la exploración se da mediante la lenta difusión de las soluciones sobre la población, mientras que la explotación ocurre en los vecindarios [1].

**Recocido Simulado (RS)** Surge a partir de una analogía entre la simulación del recocido de solidos y los problemas de optimización combinatoria. El algoritmo comienza con la generación de una solución inicial. A esta solución se le aplican perturbaciones con el objetivo de identificar a sus vecinos. Si un vecino generado mediante la perturbación es mejor que la mejor solución actual, se lo acepta manteniéndolo como nueva mejor solución. Si el vecino generado es peor que la mejor solución actual, se lo acepta con una probabilidad  $p = \exp(-\theta/T)$ , que se corresponde con la probabilidad de Boltzman, donde  $\theta$  es la diferencia entre el valor de aptitud del vecino y la mejor solución actual, y  $T$  es la temperatura que es establecida en un valor inicial y decrece constantemente de acuerdo a un coeficiente de enfriamiento [13].

### 3.2 Función Kernel evaluada

Dado que el rendimiento de un SVM se reduce notablemente cuando debe clasificar instancias de datos no separables, se utilizan funciones kernel que las eleven a una dimensión en la que exista un hiperplano que los separe. En este

trabajo se usa el kernel Wavelet [29] debido a las ventajas que supone su utilización.

La idea detrás del kernel Wavelet es aproximar funciones o señales mediante la dilatación de una función Wavelet madre, para lograr una nueva representación de las instancias del conjunto de datos. Estas instancias se ubican en las filas de una matriz  $D$  de dimensiones  $M$  (filas)  $\times$   $N$  (columnas), sobre la que se aplica el kernel Wavelet. Como resultado, se obtiene una nueva matriz  $P$  de dimensiones  $M \times M$ , que será utilizada por el SVM para la clasificación, donde cada elemento  $P_{ij}$  se obtiene mediante la Ecuación 2.

$$P_{ij} = \prod_{l=1}^N h\left(\frac{D_{il} - D_{jl}}{a}\right) \quad (2)$$

Donde  $N$  es la cantidad de columnas de la matriz  $D$  (cantidad de características de las instancias a clasificar),  $a$  es una constante de dilatación y  $h$  es la función Wavelet madre conocida como Morlet, presentada en la Ecuación 3.

$$h(u) = \cos(t_1 * u) * \exp\left(-\frac{u^2}{t_2}\right) \quad (3)$$

En este caso,  $t_1$  y  $t_2$  influyen sobre la amplitud y el periodo de la función Wavelet madre, mientras que el parámetro  $a$  de la Ecuación 2 afecta su dilatación. Los valores que tomen estos parámetros deben ser optimizados para que se adecúen al conjunto de datos utilizado, con el fin de obtener una distribución que beneficie el proceso de entrenamiento y clasificación del SVM.

El kernel Wavelet es una función ortonormal capaz de aproximar funciones continuas con gran precisión. La distribución de datos que genera provoca una mejora en la generalización y el rendimiento del SVM, destacándolo por sobre otras funciones kernel comúnmente utilizadas [6, 10, 29].

### 3.3 Definición del Individuo

El individuo se compone de cuatro variables. Tres corresponden a los parámetros del kernel Wavelet y el último al coeficiente de suavización de margen  $C$ . Todos los hiperparámetros son del tipo real, por lo que se define por cada uno un rango de valores posibles, los cuales son presentados en la Tabla 1.

**Tabla 1.** Rango de valores considerados para cada uno de los hiperparámetros.

Hiperparámetro	Descripción	Rango	Por Defecto
$a$	Coef. dilatación de wavelet	$[1E - 5; 1E5]$	1
$t_1$	Coeficiente Morlet	$[1E - 5; 2.5]$	1.75
$t_2$	Coeficiente Morlet	$[2; 100]$	2
$C$	Suavización de margen	$[1E - 5; 1E5]$	1

El rango del hiperparámetro  $C$  fue establecido a partir de los valores adoptados por otros trabajos [12, 15]. Los rangos de valores posibles de los hiperparámetros del kernel Wavelet  $a$ ,  $t_1$  y  $t_2$  se definieron mediante un análisis de sensibilidad aplicado sobre cada uno de ellos.

Los valores por defecto para los hiperparámetros  $a$ ,  $t_1$  y  $t_2$  fueron establecidos con base en los definidos en [29]. En cuanto a  $C$ , se consideró el valor por defecto sugerido por la librería Scikit-Learn [18].

## 4 Experimentos y Resultados

Esta sección informa aspectos relacionados a la experimentación, tales como detalles de del conjunto de datos utilizado, la configuración de cada metaheurística y los resultados obtenidos en cada caso.

### 4.1 Conjunto de Datos y configuración utilizada

Para la evaluación de las metaheurísticas propuestas, se utilizó el conjunto de datos de Retinopatía Diabética (DR) perteneciente a la universidad de Debrecen [2], que consiste en 1151 instancias compuestas por 19 atributos y un identificador de clase que distingue los casos positivos de DR, de aquellos negativos. Los atributos contemplan aspectos como la presencia de microaneurismas y exudaciones, la distancia entre la mácula y el disco óptico, entre otros. Este conjunto de datos resulta atractivo debido a que sirve de contribución a la aplicación de técnicas de machine learning a casos médicos. Múltiples trabajos con distintas aproximaciones fueron desarrolladas utilizándolo [17, 19, 23]. Para los fines de este trabajo, el conjunto de datos se dividió en 75% de las instancias para entrenamiento y el restante 25% a ser utilizadas para test.

Teniendo en cuenta los valores comúnmente adoptados en la literatura y en base a análisis previamente realizados, en la Tabla 2 se define la configuración de parámetros utilizada por cada metaheurística en los experimentos a realizar.

Los experimentos fueron ejecutados sobre un procesador AMD A10-5750M con cuatro núcleos, una memoria física total de 12 GB y un sistema operativo Windows 10. Las metaheurísticas fueron implementadas utilizando la librería JMetalPy desarrollada sobre Python 3.7 [4]. La condición de parada establecida por cada una, fue alcanzar las 5000 evaluaciones del valor de aptitud. Dada la cualidad no determinista de las metaheurísticas, en este trabajo se realizó por cada una 20 ejecuciones independientes, con el objetivo de observar sus resultados y analizar sus rendimientos.

El algoritmo de Máquina de Vectores Soporte, fue implementado mediante la librería Scikit-Learn [18]. En las evaluaciones se contrastan los resultados obtenidos por SVM utilizando el kernel Wavelet con la configuración obtenida por cada metaheurística contra los resultados obtenidos por SVM utilizando las funciones kernel Función de Base Radial (RBF, *Radial Basis Function*), Lineal y Sigmoidal. Los parámetros por defecto de estas funciones kernel, son los

**Tabla 2.** Configuración de las metaheurísticas.

Alg.	Parámetro	Valor
<b>AG</b>	Tamaño población	100
	Operador Selección	Torneo Binario
	Operador Cruza	Cruza SBX - Prob.: 0.9, Índice dist.: 20
	Operador de Mutación	Mutación Caótica - Prob.: 0.2
<b>ED</b>	Tamaño población	100
	Operador de Cruza	Cruza de ED - Prob.: 0.5
	Factor de Escalado	0.5
<b>AGC</b>	Tamaño población	10×10
	Vecindario	L5
	Operador Selección	Torneo Binario
	Operador Cruza	Cruza SBX - Prob.: 0.9, Índice dist.: 20
<b>RS</b>	Operador de Mutación	Mutación Caótica - Prob.: 0.2

sugeridos por la librería Scikit-Learn, con base en que, en general, presentan buenos resultados en los problemas en los que se aplica un clasificador SVM. [18].  $\Gamma$  se establece como la inversa del número de características de las instancias y el coeficiente  $C$  de suavización de margen es igual a 1.

## 4.2 Resultados Experimentales

Los experimentos realizados apuntan a obtener un elemento de análisis en cuanto a la robustez y efectividad de las metaheurísticas evaluadas. En este sentido se observa la calidad de las configuraciones de hiperparámetros obtenidas por cada una, el efecto producido sobre la clasificación efectuada por el SVM y el rendimiento relacionado al tiempo de ejecución. En cada tabla presentada, se resalta en color gris oscuro al mejor resultado y en color gris claro al segundo mejor valor.

La Tabla 3 muestra los valores promedio y desviación estándar de los atributos de calidad aptitud y tiempo de ejecución, y las mejores configuraciones de hiperparámetros obtenidas a partir de las 20 ejecuciones realizadas. En cuanto al valor de aptitud, el que mejor resultados obtuvo fue ED, que se destaca además por tener una desviación estándar nula, indicando que en todas las ejecuciones logró alcanzar el mismo resultado. El segundo mejor valor corresponde a AGC debido a que posee una menor desviación estándar comparado al AG que obtuvo el mismo promedio del valor de aptitud. Sin embargo, vale destacar que todas las metaheurísticas convergieron a valores cercanos al máximo en la mayor parte de las ejecuciones, lo que refleja robustez para abordar el problema de identificar el valor de los hiperparámetros de SVM. Los valores de aptitud obtenidos por cada metaheurística a lo largo de las 20 ejecuciones, fueron evaluados estadísticamente mediante el test de Mann-Whitney Wilcoxon [8] considerando un nivel de significancia  $\alpha = 0.05$ .



Los resultados mostraron que los enfoques *ED*, *AGC* y *AG* no tuvieron diferencias entre sí, siendo los tres estadísticamente superiores al *RS*.

**Tabla 3.** Mejor Configuración y Promedio y desviación estándar del valor de aptitud (Exactitud) y el tiempo de ejecución (Segundos) por cada metaheurística.

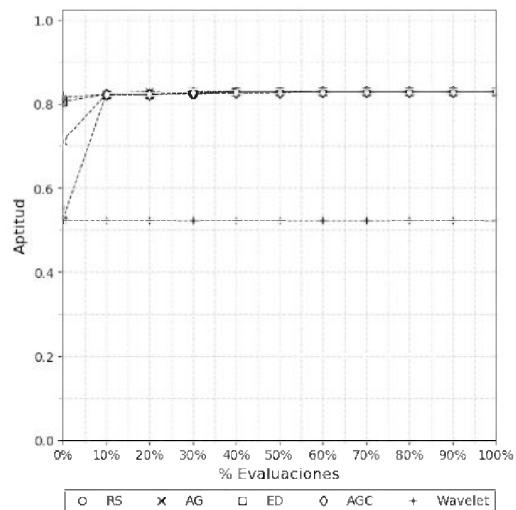
Alg.	Aptitud		Tiempo		Mejor Configuración			
	Promedio	Desv. Est.	Promedio	Desv. Est.	$a$	$t_1$	$t_2$	$C$
<b>AG</b>	0.829	0.002	21000	2900	199.7	1.9	81.8	4766.9
<b>ED</b>	0.830	0.000	21200	2030	285.6	2.5	23.2	5000.0
<b>AGC</b>	0.829	0.001	18400	1500	262.3	2.3	34.5	4988.0
<b>RS</b>	0.828	0.002	21400	2500	279.7	2.5	26.6	4833.4

La metaheurística con menor tiempo de ejecución promedio fue el AGC, que tuvo una diferencia significativa con respecto al resto. El segundo mejor tiempo de ejecución fue logrado por AG y el peor tiempo de ejecución lo obtuvo RS. Los algoritmos mas estables en tiempo de ejecución, tomando como base la desviación estándar, fueron el AGC y el ED.

La Tabla 3 también muestra la mejor configuración de hiperparámetros SVM obtenida por cada metaheurística a lo largo de las 20 ejecuciones realizadas. Se observa que todas las metaheurísticas alcanzaron valores similares en todos los hiperparámetros, excepto en  $t_2$  donde AG se diferenció con respecto a los otros algoritmos. Esto tiene relación en que, en este problema, se posee una región óptima con infinitas soluciones en lugar de una única solución.

Para un mejor análisis del comportamiento de cada metaheurística, se muestra en la Figura 1 la evolución del valor de aptitud a lo largo de las 5000 evaluaciones realizadas. En esta figura se observan las cuatro metaheurísticas y el valor obtenido por el kernel Wavelet con los valores establecidos por defecto. En la Figura 1, el eje  $x$  corresponde al porcentaje de evaluaciones completadas y el eje  $y$  es el valor de aptitud obtenido en ese punto. En ella se observa que la calidad de las soluciones obtenidas por las metaheurísticas resulta superior al SVM con kernel Wavelet sin optimizar, evidenciando la mejora producida. También se destaca que el ED y el AGC generan soluciones iniciales mejores que las del AG y RS, lo que sugiere que poseen una mayor facilidad para identificar las regiones óptimas del espacio de búsqueda desde un inicio. No obstante, todas las metaheurísticas confluyen a valores óptimos a partir del 30% de las evaluaciones realizadas.

Otro aspecto de interés a la hora de analizar las configuraciones obtenidas por las metaheurísticas, es observar el impacto sobre la calidad de clasificación del SVM. En primer lugar, se utilizan las Curvas de Característica Operativa del Receptor (ROC, *Receiver Operating Characteristic*), que muestran la evolución de la Tasa de Verdaderos Positivos (proporción de ejemplos positivos bien clasificados) a medida que crece la Tasa de Falsos Positivos (proporción de ejemplos negativos mal clasificados). Esta curva proporciona una idea de la separación de clases realizada por el clasificador. La situación ideal es lograr un



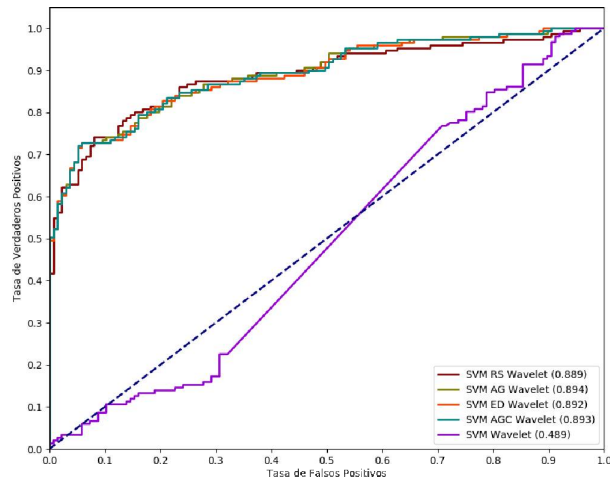
**Figura 1.** Evolución del valor del indicador de calidad Aptitud por cada metaheurística.

área bajo la curva de 1, que significa que todos los ejemplos fueron clasificados correctamente. Por otro lado, en este gráfico se establece un límite -que se corresponde con una función identidad- indicando que si la curva ROC se encuentra por debajo, la separación de clases es mala o nula.

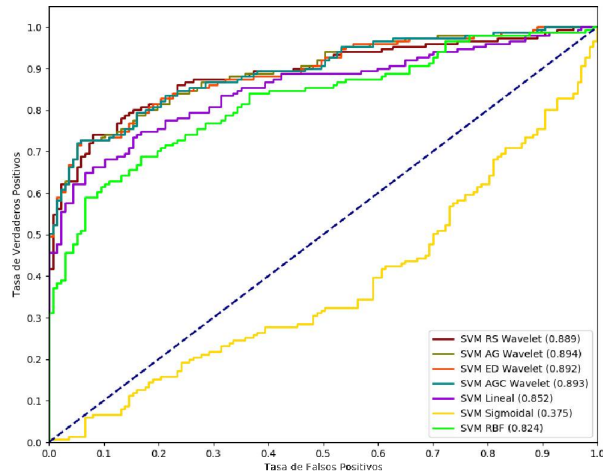
La Figura 2 muestra las curvas ROC de los SVM optimizados y la del SVM con kernel Wavelet con valores por defecto. En general el rendimiento mostrado por los SVM optimizados fue similar, destacándose por una mínima diferencia la configuración realizada por el AG. La curva del kernel Wavelet con valores por defecto se ubicó por debajo del límite, lo que sugiere una mala distinción entre casos positivos y negativos.

La Figura 3 presenta las curvas ROC de los SVM con kernel Wavelet optimizado y las de los SVM con kernel Lineal, Sigmoideal y Función de Base Radial (*RBF, Radial Basis Function*). En la Figura se observa que los SVM optimizados logran una curva ROC cuya área es superior a los SVM con otras funciones kernel, reflejando una mejor separación de clases y por ende una mejor clasificación.

Finalmente, con el fin de tener un punto de vista más amplio, se contemplan otros valores relacionados directamente con el poder de clasificación del SVM. Se compara la mejor configuración obtenida por cada metaheurística contra el kernel Wavelet, Lineal, RBF y Sigmoideal. En la Tabla 4 se analiza para cada configuración: el Área Bajo la Curva ROC (*AUC, Area Under the Curve*) que representa la calidad de la separación de clases del SVM, la Exactitud que informa la proporción de ejemplos que fueron bien



**Figura 2.** Curva ROC de las distintas variantes de SVM Wavelet



**Figura 3.** Curva ROC obtenido tanto por los SVM con kernel Wavelet optimizado como por los kernel Lineal, RBF y Sigmoidal

clasificados, la Precisión que es la proporción de los ejemplos clasificados como positivos que realmente son positivos, la Especificidad que es la proporción de ejemplos negativos clasificados como negativos y la sensibilidad que es la proporción de ejemplos positivos clasificados como positivos.

En general, los SVM optimizados por las metaheurísticas lograron valores superiores a los obtenidos por los demás kernel, lo que sugiere que logran una

Tabla 4. Resultados obtenidos por cada clasificador.

Alg.	AUC	Exactitud	Precisión	Especificidad	Sensibilidad
SVM AG Wavelet	0.894	0.830	0.932	0.942	0.728
SVM ED Wavelet	0.892	0.830	0.932	0.942	0.728
SVM AGC Wavelet	0.893	0.830	0.932	0.942	0.728
SVM RS Wavelet	0.889	0.826	0.911	0.920	0.742
SVM Wavelet	0.489	0.524	0.524	0.0	1.0
SVM Lineal	0.852	0.774	0.858	0.875	0.682
SVM Kernel RBF	0.824	0.753	0.870	0.897	0.623
SVM Kernel Sigmoidal	0.375	0.385	0.420	0.307	0.457

mejor distribución de datos facilitando así la identificación de un mejor hiperplano de separación. Si bien los resultados tienen gran similitud, la metaheurística que peor resultados obtuvo fue RS.

Un aspecto llamativo son los valores de especificidad y sensibilidad obtenidos por el SVM Wavelet sin optimización -0.0 y 1.0 respectivamente-, que indican que el modelo clasificó a todos los ejemplos como positivos, por lo que el SVM no fue capaz de distinguir entre clases.

En definitiva, de estos resultados resalta la influencia de la configuración de hiperparámetros de SVM en la obtención de buenos resultados, pudiendo producir una mejora significativa en la calidad de la clasificación. Además, demuestra la factibilidad y eficacia del kernel Wavelet en la separación de datos, lo cual facilita la identificación del mejor hiperplano de separación.

## 5 Conclusiones y Trabajo Futuro

En este trabajo se propuso la utilización de metaheurísticas para la optimización de los hiperparámetros de una Máquina de Vectores Soporte que utiliza un kernel Wavelet.

Las pruebas se centraron en demostrar los aspectos relacionados al rendimiento de cada metaheurística y como su utilización tiene incidencia sobre la calidad de clasificación.

Se observó que todas las metaheurísticas lograron alcanzar valores altos de aptitud, lo que demuestra una buena exploración del espacio de búsqueda. Por otro lado, el Algoritmo Genético Celular se destacó en el tiempo de ejecución promedio que le demandó alcanzar el criterio de parada, lo que sugiere una mayor eficiencia con respecto a las otras propuestas.

Desde un punto de vista general, los resultados relacionados a la calidad de clasificación demostraron dos cosas. En primer lugar, que las metaheurísticas son capaces de obtener configuraciones que realcen notoriamente la eficiencia del clasificador, debido a su capacidad de explorar el espacio de soluciones, evitando estancarse en óptimos locales. En segundo lugar, que el kernel Wavelet es capaz de lograr una distribución adecuada de los datos, siempre y cuando posea una correcta configuración de sus parámetros.

Como trabajo futuro se propone probar la optimización mediante metaheurísticas y el kernel Wavelet sobre otros conjuntos de datos, con el fin de evaluar sus rendimientos con otras distribuciones. Además, es interesante evaluar distintas variantes del kernel Wavelet que poseen características destacables en cuanto a la distribución de datos que logran.

**Agradecimientos.** Los fondos para realizar esta investigación provienen en parte de la Universidad Nacional de Cuyo a través del Proyecto Tipo 1 Código B081 y en parte de la Universidad Nacional del Sur, Secretaría General de Ciencia y Tecnología con el proyecto PGI 24/N042.

## References

1. Alba, E., Dorronsoro, B.: Cellular Genetic Algorithms. Springer Publishing Company, Incorporated, 1st edn. (2008)
2. Antal, B., Hajdu, A.: An ensemble-based system for automatic screening of diabetic retinopathy. *Knowledge-Based Systems* **60**, 20–27 (Apr 2014). <https://doi.org/10.1016/j.knosys.2013.12.023>
3. Ben-Hur, A., Weston, J.: A user's guide to support vector machines. In: *Methods in Molecular Biology*, pp. 223–239. Humana Press (Oct 2009). [https://doi.org/10.1007/978-1-60327-241-4\\_13](https://doi.org/10.1007/978-1-60327-241-4_13)
4. Benitez-Hidalgo, A., Nebro, A.J., Garcia-Nieto, J., Oregi, I., Del Ser, J.: jmetalpy: a python framework for multi-objective optimization with metaheuristics. arXiv preprint arXiv:1903.02915 (2019)
5. Demidova, L., Egin, M., Tishkin, R.: A self-tuning multiobjective genetic algorithm with application in the SVM classification. *Procedia Computer Science* **150**, 503–510 (2019). <https://doi.org/10.1016/j.procs.2019.02.085>
6. Ding, S., Wu, F., Shi, Z.: Wavelet twin support vector machine. *Neural Computing and Applications* **25**(6), 1241–1247 (Apr 2014). <https://doi.org/10.1007/s00521-014-1596-y>
7. Farhadi, A., Chen, D., McCoy, R., Scott, C., Miller, J.A., Vachon, C.M., Ngufor, C.: Breast cancer classification using deep transfer learning on structured healthcare data. In: *2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. pp. 277–286. IEEE (2019)
8. Heumann, C., Schomaker, M., Shalabh: *Introduction to Statistics and Data Analysis*. Springer International Publishing (2016)
9. Holland, J.H.: Genetic algorithms. *Scientific American* (Jul 1992)
10. Hou, S., Zhou, Y., Liu, H., Zhu, N.: Wavelet support vector machine algorithm in power analysis attacks. *Radioengineering* **26**(3), 890–902 (Sep 2017). <https://doi.org/10.13164/re.2017.0890>
11. Izabatene, H.F., Benhabib, W., Ghardaoui, S.: Contribution of kernels on the svm performance. *Journal of Applied Sciences* **10**(10), 831–836 (2010)
12. Kalita, D.J., Singh, S.: SVM hyper-parameters optimization using quantized multi-PSO in dynamic environment. *Soft Computing* **24**(2), 1225–1241 (Apr 2019). <https://doi.org/10.1007/s00500-019-03957-w>
13. Kirkpatrick, S.: Optimization by simulated annealing: Quantitative studies. *Journal of statistical physics* **34**(5-6), 975–986 (1984)

14. Liu, Z., Cao, H., Chen, X., He, Z., Shen, Z.: Multi-fault classification based on wavelet SVM with PSO algorithm to analyze vibration signals from rolling element bearings. *Neurocomputing* **99**, 399–410 (Jan 2013). <https://doi.org/10.1016/j.neucom.2012.07.019>
15. Mantovani, R.G., Rossi, A.L., Alcobaça, E., Vanschoren, J., de Carvalho, A.C.: A meta-learning recommender system for hyperparameter tuning: Predicting when tuning improves SVM classifiers. *Information Sciences* **501**, 193–221 (Oct 2019). <https://doi.org/10.1016/j.ins.2019.06.005>
16. Noble, W.S.: What is a support vector machine? *Nature Biotechnology* **24**(12), 1565–1567 (Dec 2006). <https://doi.org/10.1038/nbt1206-1565>
17. Oladele, T.O., Ogundokun, R.O., Kayode, A.A., Adegun, A.A., Adebisi, M.O.: Application of data mining algorithms for feature selection and prediction of diabetic retinopathy. In: *International Conference on Computational Science and Its Applications*. pp. 716–730. Springer (2019)
18. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011)
19. Rojas-Domínguez, A., Padierna, L.C., Valadez, J.M.C., Puga-Soberanes, H.J., Fraire, H.J.: Optimal hyper-parameter tuning of svm classifiers with application to medical diagnosis. *IEEE Access* **6**, 7164–7176 (2017)
20. Sarkar, S.K., Oshiba, K., Giebisch, D., Singer, Y.: Robust classification of financial risk. *arXiv preprint arXiv:1811.11079* (2018)
21. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization* **11**(4), 341–359 (1997)
22. Sudharshan, P., Petitjean, C., Spanhol, F., Oliveira, L.E., Heutte, L., Honeine, P.: Multiple instance learning for histopathological breast cancer image classification. *Expert Systems with Applications* **117**, 103–111 (2019)
23. Tang, C., Ji, J., Tang, Y., Gao, S., Tang, Z., Todo, Y.: A novel machine learning technique for computer-aided diagnosis. *Engineering Applications of Artificial Intelligence* **92**, 103627 (2020)
24. Tang, Y., Guo, W., Gao, J.: Efficient model selection for support vector machine with gaussian kernel function. In: *2009 IEEE Symposium on Computational Intelligence and Data Mining*. pp. 40–45. IEEE (2009)
25. Vapnik, V.N.: *The Nature of Statistical Learning Theory*. Springer New York (2000). <https://doi.org/10.1007/978-1-4757-3264-1>
26. Wang, W., Xu, Z., Lu, W., Zhang, X.: Determination of the spread parameter in the gaussian kernel for classification and regression. *Neurocomputing* **55**(3-4), 643–663 (2003)
27. Yin, X., Hou, Y., Yin, J., Li, C.: A novel SVM parameter tuning method based on advanced whale optimization algorithm. *Journal of Physics: Conference Series* **1237**, 022140 (Jun 2019). <https://doi.org/10.1088/1742-6596/1237/2/022140>
28. Zhang, D.: Support vector machine. In: *Fundamentals of Image Data Mining*, pp. 179–205. Springer (2019)
29. Zhang, L., Zhou, W., Jiao, L.: Wavelet support vector machine. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)* **34**(1), 34–39 (Feb 2004). <https://doi.org/10.1109/tsmcb.2003.811113>
30. Zihao, W., Lan, L., Zongyi, X., Guangtao, C.: The forecasting model of wheelset size based on PSO-SVM. In: *2019 Chinese Control And Decision Conference (CCDC)*. IEEE (Jun 2019). <https://doi.org/10.1109/ccdc.2019.8832937>