

Desarrollo de un marco de trabajo para la selección de parámetros de un modelo de máquina de soporte vectorial en un problema de pronósticos de series de tiempo

Sibelis Estella Barceló Muzza, 2130381, Mayra Alejandra Durán Amaya, 2101890

Trabajo de Grado para Optar el título de Ingeniero Industrial

Director: Daniel Orlando Martínez Quezada

Msc. Ingeniería Industrial

Codirector: Henry Lamos Díaz

Ph.d. Física-Matemática

Universidad Industrial de Santander

Facultad de Ingeniería Fisicomecánicas

Escuela de Estudios Industriales y Empresariales

Bucaramanga

2019

Dedicatoria

Primeramente doy gracias a Dios por haberme dado la fuerza y las ganas seguir adelante, a mis padres mi mayor gratitud por ser los pilares de mi vida y mi mayor motivación, a mi hermano mayor a quien admiro y respeto, a Alejandra quien me acompañó en esta gran aventura de principio a fin y a todas esas personas que estuvieron siempre brindándome su apoyo incondicional, gracias infinitas.

Síbelis Estella Barceló Muzza

Dedicatoria

A Dios por ser mi guía y haberme dado el regalo de una familia hermosa, mis padres Lidia y Ferney, que incondicionalmente han creído en mí, por su apoyo constante, por ser mi ejemplo de esfuerzo y superación, a mi hermana Yuly quien se convirtió en mi compañera fiel, por sus enormes detalles llenos de amor, a Sibelis por ser parte de este proyecto con su enorme carisma siempre, a mis amigos y familiares quienes siempre estuvieron allí para darme una mano en el momento en que lo necesité. Gracias a su invaluable contribución pude culminar este paso en mi carrera con Éxito.

Mayra Alejandra Durán Amaya

Agradecimientos

Agradecemos a los profesores Daniel Orlando Martínez Quezada y Henry Lamos Díaz, por el apoyo y la confianza que nos brindaron a lo largo de este proyecto, por siempre tener la mejor disposición y dedicarnos gran parte de su tiempo y conocimiento, sin ustedes esto no habría sido posible.

Tabla de contenido

Introducción	19
1. Justificación	23
2. Planteamiento del problema.....	24
3. Objetivos	27
3.1. Objetivo general	27
3.2. Objetivos específicos	27
4. Marco teórico	28
4.1. Modelos de regresión.....	28
4.1.1. Regresión lineal	28
4.1.2. Regresión lineal simple	29
4.1.3. Regresión múltiple	31
4.2. Algoritmos de optimización	32
4.2.1. Heurística	32
4.2.2. Metaheurística	32

4.3.	Aprendizaje automático	33
4.3.1.	Aprendizaje supervisado	33
4.3.2.	Aprendizaje no supervisado	34
4.4.	Series de tiempo.....	34
4.4.1.	Arma	35
4.4.2.	Arima	36
4.5.	Máquinas de soporte vectorial.....	37
4.5.1.	Regresión vectorial de soporte.....	38
4.5.2.	Función kernel	40
4.6.	Métodos de validación.....	41
4.7.	Metodologías para la selección de parámetros	43
4.7.1.	Búsqueda de cuadrículas o método grid	44
4.7.2.	Algoritmos genéticos	44
4.8.	Particle swarm optimization	45
5.	Estado del arte.....	50
5.1.	Análisis de la literatura	50

5.2.	Marco de trabajo	54
6.	Resultados computacionales	57
6.1.	Resultados computacionales	57
6.2.	Resultados modelo svr- grid	58
6.3.	Resultados modelo svr-pso	61
6.4.	Comparación de resultados	63
7.	Conclusiones	65
8.	Recomendaciones	67
	Referencias bibliográficas	69

Lista de Figuras

<i>Figura 1.</i> Regresión lineal del Precio de un automóvil	29
Figura 2. Líneas de regresión en la regresión lineal simple	30
Figura 3. Hiperplano de margen máximo canónico	37
Figura 4. Regresión lineal unidimensional con banda intensiva ϵ . Adaptado de Support Vector Machines and other kernel-based learning methods.....	39
Figura 5. Regresión lineal en un espacio de características	41
Figura 6. Modelo de Cross Validation.....	43
Figura 7. Movimiento de las partículas dentro del enjambre.	48
Figura 8. Rango de parámetros seleccionados para cada función Kernel en el modelo SVR-GRID.	55
Figura 9. Diagrama de Flujo del Modelo SVR_PSO.	56
Figura 10. Parámetros modelo SVR-GRID.....	59
Figura 11. Resultados gráficos del Modelo SVR_GRID con cada tipo de Kernel.	60
Figura 12. Resultados gráficos del Modelo SVR_PSO con cada tipo de Kernel.....	63

Figura 13. Resultados gráficos del error RMSE calculado en cada tipo de Kernel de los modelos SVR_GRID y SVR_PSO..... 65

Lista de Tablas

Tabla 1. Cumplimiento de Objetivos.....	18
Tabla 2. Tipos de Funciones Kernels.....	41
Tabla 3. Resultados del modelo SVR_GRID.	59
Tabla 4. Tiempo computacional y RMSE del modelo SVR_GRID.	61
Tabla 5. Resultados del modelo SVR_PSO.....	62
Tabla 6. Tiempo Computacional y RMSE del modelo SVR_PSO.....	62
Tabla 7. Comparación de Resultados.....	64

Lista de Apéndices**(Ver apéndices adjuntos en el CD)**

Apéndice A. Base de datos Índice COLCAP sin transformación.

Apéndice B. Base de datos Índice COLCAP transformada con dos periodos anteriores.

Apéndice C. Código Maquinas de Soporte Vectorial con Grid para Kernel Lineal.

Apéndice D. Código Maquinas de Soporte Vectorial con Grid para Kernel Radial.

Apéndice E. Código Maquinas de Soporte Vectorial con Grid para Kernel Sigmoid.

Apéndice F. Código Maquinas de Soporte Vectorial con Grid para Kernel Polynomial.

Apéndice G. Código Maquinas de Soporte Vectorial con PSO para Kernel Lineal.

Apéndice H. Código Maquinas de Soporte Vectorial con PSO para Kernel Radial.

Apéndice I. Código Maquinas de Soporte Vectorial con PSO para Kernel Sigmoid.

Apéndice J. Código Maquinas de Soporte Vectorial con PSO para Kernel Polynomial.

Apéndice K. Artículo: Desarrollo de un marco de trabajo para la selección de parámetros de un modelo de Máquina de Soporte Vectorial en un problema de pronósticos de series de tiempo

RESUMEN

TÍTULO: DESARROLLO DE UN MARCO DE TRABAJO PARA LA SELECCIÓN DE PARÁMETROS DE UN MODELO DE MÁQUINA DE SOPORTE VECTORIAL EN UN PROBLEMA DE PRONÓSTICOS DE SERIES DE TIEMPO.*

AUTOR: BARCELÓ MUZA, Sibelis Estella.**

DURÁN AMAYA, Mayra Alejandra.**

PALABRAS CLAVE: Series de Tiempo, Máquinas de Soporte Vectorial, Regresión, Optimización por enjambre de partículas.

DESCRIPCIÓN:

El propósito principal de la siguiente investigación es crear un marco de trabajo para la selección de parámetros de un modelo de Máquina de Soporte Vectorial en un problema de Series de Tiempo, debido a que la predicción de Series de Tiempo se ha convertido en un tema importante de investigación en los últimos años por las amplias aplicaciones que tiene en diferentes campos de la ciencia. En este documento se propone crear un marco de trabajo, combinando la optimización de enjambre de partículas (PSO) y regresión de vectores de soporte (SVR), para realizar pronósticos en series temporales. Este marco de trabajo es evaluado utilizando el índice COLCAP logrando optimizar la selección de los parámetros para encontrar soluciones más rápidas y con mejor ajuste.

Este documento condensa, un marco teórico referente a modelos de regresión a partir de máquinas de soporte vectorial para regresión y metodologías para la selección de parámetros, un análisis preliminar de la literatura para saber el estado actual de las investigaciones en torno a este problema, el desarrollo de una Metaheurística la cual fue comparada con la metodología grid y la documentación de la base de datos utilizada, junto a los resultados computacionales tras la aplicación de dichos modelos.

* Trabajo de grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Estudios Industriales y Empresariales. Director: Daniel Orlando Martínez Quesada, Msc. Ingeniería Industrial. Codirector: Henry Lamos Díaz, PhD. Física-Matemática.

ABSTRACT

TITLE: DEVELOPMENT OF A FRAMEWORK FOR THE SELECTION OF PARAMETERS OF A MODEL OF SUPPORT VECTOR MACHINE IN A PROBLEM OF PROGNOSTICS OF TIME SERIES.*

AUTHOR: BARCELÓ MUZA, Sibelis Estella.**

DURÁN AMAYA, Mayra Alejandra.**

KEY WORDS: Time Series, Support Vector Machine, Regression, Particle Swarm Optimization.

DESCRIPTION:

The main purpose of the following research is to create a framework for the parameter selection of a Support Vector Machine model in a Time Series problem, because the Time Series prediction has become an important topic of research in recent years due to the broad applications it has in different fields of science. In this document, it is proposed to create a framework, combining Particle Swarm Optimization (PSO) and Support Vector Regression (SVR), to make forecasts in time series. This framework is evaluated using the COLCAP index, optimizing the selection of parameters to find faster solutions with better adjustment.

This document condenses, a theoretical framework concerning the regression models based on Support Vector Machine for regression and methodologies for the selection of parameters, a preliminary analysis of the literature to know the current state of the investigations around this problem, the development of a metaheuristic which was compared with the grid methodology and the documentation of the database used, together with the computational results after the application of these models.

* Bachelor Thesis

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Estudios Industriales y Empresariales. Director: Daniel Orlando Martínez Quesada, Msc. Ingeniería Industrial. Codirector: Henry Lamos Díaz, PhD. Física-Matemática.

Tabla 1.

Cumplimiento de Objetivos.

CUMPLIMIENTO DE OBJETIVOS	
Objetivo	Cumplimiento
Realizar una revisión de la literatura sobre los modelos de máquina de soporte vectorial aplicados para el pronóstico de series de tiempo.	Este objetivo se da por cumplido en el capítulo 5 que consiste en el análisis de la literatura.
Consolidar bases de datos de prueba para la evaluación de modelos de pronóstico para series de tiempo.	En el capítulo 6 se consolidó la base de datos utilizada en el estudio.
Construir una metaheurística basada en la optimización de enjambre de partículas (PSO), para integrar el marco de trabajo que permita la selección de parámetros para modelos de máquinas de soporte vectorial SVM en problemas de pronóstico de series de tiempo.	En el capítulo 5 se hace un marco de trabajo que explica el proceso de diseño del modelo de Maquinas de Soporte Vectorial (SVM) y el algoritmo de Optimización por Enjambre de Partículas (PSO) por el cual se realiza la selección de parámetros en un problema de pronóstico de series de tiempo.
Comparar el desempeño de la metaheurística con metodologías de selección de parámetros existentes.	Este objetivo se da por cumplido en el capítulo 6 en donde se hace una comparación de los resultados de los modelos SVR-GRID y el modelo construido SVR-PSO después de ser evaluados.
Elaborar un artículo de carácter publicable sobre los resultados obtenidos en la investigación.	El artículo se entrega en un documento anexo a la escuela de estudios industriales y empresariales.

Nota: Contiene los objetivos planteados para el proyecto y el capítulo donde se les da cumplimiento.

Introducción

Cuando hablamos de Big Data nos referimos a la gran cantidad de datos (estructurados, no estructurados y semi estructurados) que crecen de forma exponencial y de manera acelerada. Se refiere a todos los datos que son usados para obtener información y generar valor a través de ellos. El concepto de Big Data abarca toda la información que no puede ser utilizada, procesada o analizada de forma tradicional. Esta cantidad de datos es utilizada en múltiples disciplinas de la ciencia, una de ellas es la inteligencia artificial pues se puede decir que el combustible de la inteligencia artificial son los datos.

Por otra parte, el aprendizaje automático es una rama de la inteligencia artificial y se refiere al hecho de que los ordenadores a partir de datos disponibles son capaces de aprender a generar modelos que permiten predecir datos y comportamientos futuros, esto quiere decir que su objetivo es lograr que las máquinas aprendan de la experiencia (datos que reciben). El aprendizaje automático es útil en la toma de decisiones. Esta metodología permite resolver problemas en un ámbito multidisciplinar como la medicina, el medio ambiente, la ingeniería y más.

Recientemente en la predicción de series de tiempo, el aprendizaje automático ha tenido un papel importante, debido a que es una tarea difícil; los datos de este tipo pueden tener problemas de estacionalidad, ruido, entre otras. Es así como se han dedicado diversos esfuerzos en la investigación del aprendizaje automático para lograr predicciones eficientes en situaciones de la vida real que presentan comportamientos no estacionarios y condiciones operacionales en evolución dado el entorno dinámico actual, por esta razón estos modelos deben tener la capacidad

de realizar cambios en sus pronósticos de manera oportuna. Para la Ingeniería Industrial es importante el estudio de estas series de tiempo gracias a su capacidad de contribuir al mejoramiento continuo de los diferentes procesos dentro de las compañías en múltiples campos laborales esto debido a que el correcto pronóstico de estas series de tiempo crea ventajas únicas. La cantidad de datos que día a día aumenta significativamente genera grandes desafíos para las organizaciones que quieren aprovechar los beneficios de utilizarlos para entre otras cosas proporcionar información que sirva para el análisis de patrones de compra de los clientes, tendencias en el mercado, demanda de productos, lo que conlleve a la reducción de gastos y la toma de decisiones más efectivas así como la obtención de ventajas competitivas; como ejemplo se encuentra la utilización de modelos para pronósticos de series de tiempo que generen ventajas en la cadena de suministro, el mejoramiento de su flexibilidad y la integración entre las cadenas de suministro y los procesos logísticos todo esto para satisfacer la demanda la cual, como ya se mencionó antes, debido al mundo dinámico en el que vivimos, tiene una alta fluctuación lo que se convierte en un gran desafío, por lo tanto el pronóstico de la demanda se vuelve un punto crucial para la planificación de las operaciones en la cadena de suministro. Otro ejemplo importante se da en los pronósticos del precio de las acciones pues es una de las tareas más desafiantes que existen en el ámbito financiero debido a su alta fluctuación generada por el mercado. Esto genera múltiples desafíos en investigadores pues los métodos tradicionales presentan grandes problemas como son pronósticos con poca precisión, así como la necesidad de tener grandes cantidades de datos para poder crear modelos eficientes. Estos ejemplos señalan la gran necesidad del uso de nuevas tecnologías y la implementación del aprendizaje automático para la solución de los desafíos actuales. Una técnica importante del aprendizaje automático utilizada para dar solución a estos problemas en los últimos años son las máquinas de soporte vectorial SVM.

Las SVM fueron desarrolladas por Vapnik (1982) y han recibido mucha atención en su aplicación en los campos relacionados con inteligencia artificial debido a que estas se basan en la teoría del aprendizaje estadístico. Su especificación se centra en el principio de minimización del riesgo estructural lo que equivale a minimizar el límite superior del error de generalización del modelo. Así, la estimación de una SVM busca encontrar tanto la estructura óptima del modelo como los valores óptimos de sus parámetros, logrando que el modelo sea eficaz a lo largo del tiempo, esto quiere decir, mejorar su capacidad de generalización. Las SVM se encargan de resolver problemas de clasificación, así como de regresión. Inicialmente las máquinas de soporte vectorial basaron sus estudios en la solución de problemas de predicción lineal. En la actualidad las SVM también se usan para resolver problemas de regresión no lineal y son conocidas como regresión de soporte vectorial (SVR), las cuales utilizan la teoría de kernels para aproximar funciones no lineales.

En las SVR, los parámetros son los encargados de ajustar adecuadamente el modelo para el pronóstico de observaciones futuras, por esta razón es de vital importancia su correcta selección dado que se puede generar un sobreajuste o un subajuste, por lo que se hace necesario el uso de métodos que optimicen los parámetros, los cuales pueden ser escogidos de dos formas generalmente aceptadas. La primera es una manual que se basa en la experiencia del autor y tiende a tardar mucho más tiempo.

La segunda es la utilización de herramientas de búsqueda, algunos ejemplos son los algoritmos basados en la naturaleza, como los algoritmos genéticos (GA), el algoritmo de optimización de colonias de hormigas (ACO) y el algoritmo de optimización de enjambres de partículas. Este último propuesto por Kennedy y Eberhart (1995), fue inspirado en el comportamiento de las aves

y los peces en conjunto, y está basado en la población y en la inteligencia de enjambres, teniendo en cuenta principios psicológicos sociales y proporciona información sobre el comportamiento social, además de contribuir a las aplicaciones de ingeniería. Un PSO tiene facilidad en su implementación y es computacionalmente eficiente, es utilizado para la optimización de los parámetros ya que tiene un menor número de estos, aparte de que tiene la capacidad de escapar a mínimos locales, hecho que supone una oportunidad para su utilización en la predicción de series de tiempo.

La organización de este documento es la siguiente. La sección 4 y 5 proporcionan una revisión literaria con el objetivo de identificar palabras claves y términos relacionados con las máquinas de soporte vectorial, así como el análisis bibliométrico a partir de una ecuación de búsqueda definida y la consolidación de las bases de datos que servirán para la construcción de la metaheurística mediante PSO. En la sección 6 se encuentran los resultados de cada uno de los algoritmos desarrollados, el análisis de su desempeño y la evaluación de los modelos de pronóstico de series de tiempo. Finalmente, en las secciones 7 y 8 muestran las conclusiones y recomendaciones a las que se llegó a lo largo de este libro.

1. Justificación

En los últimos años se han dedicado esfuerzos importantes en la investigación de la predicción de series de tiempo ya que estas son utilizadas en múltiples campos de la ciencia. Una técnica que ha estado tomando fuerza en los últimos años es el aprendizaje automático debido a que los datos de las series de tiempo pueden presentar diversos problemas como pueden ser comportamientos no estacionarios y condiciones operacionales en evolución, aparte del constante dinamismo del entorno actual entre otras. Es por esto que las predicciones deben ser capaces de realizar cambios de manera oportuna y eficaz. Una técnica importante que se ha venido desarrollando en los últimos años para dar solución a estos problemas son las máquinas de soporte vectorial SVM.

Las SVM se basan en la teoría de aprendizaje estadístico, de esta manera lo que busca es que el modelo se pueda generalizar para lograr que sea eficaz a lo largo del tiempo. Dentro de las SVM se encuentran los parámetros que son los encargados de ajustar adecuadamente el modelo. Una mala selección de estos parámetros puede generar un subajuste o un sobreajuste del modelo, generando errores en la predicción, por otra parte, una selección “pobre” de parámetros puede ocasionar un alto costo computacional, generando así una mayor dificultad por parte del programador para su correcta selección. Es por esto que se hace necesaria la utilización de herramientas de búsqueda, las cuales se encargan de solucionar el problema de identificación de parámetros. El algoritmo de optimización de enjambre de partículas (PSO) el cual se basa en la simulación del comportamiento de los individuos en un grupo para maximizar la supervivencia de las especies, entre las características deseables es su fácil implementación debido a que utiliza

pocas líneas de codificación, y de tener beneficios como ser computacionalmente eficiente, un hecho fundamental en la predicción de las series de tiempo.

En el presente trabajo se pretende realizar un marco de trabajo que sirva como guía en posteriores investigaciones en el cual se realice la selección de parámetros de un modelo de máquinas de soporte vectorial en un problema de pronósticos de series de tiempo por medio de una metaheurística, en el caso particular para la metaheurística enjambre de partículas (PSO), con el fin de solucionar los problemas existentes con los modelos anteriormente planteados.

2. Planteamiento del problema

Debido a las amplias aplicaciones que tiene la predicción de series de tiempo en campos de la ingeniería, economía y finanzas, se hace necesario la constante investigación para el desarrollo de nuevos modelos que optimicen los resultados de las predicciones los cuales pueden ser comparados con estudios previos. Unos de los más conocidos son las redes neuronales y los modelos ARIMA, los cuales tienen la necesidad de utilizar muchos datos para su entrenamiento, esto hace que sea difícil los pronósticos de las series de tiempo, y el coste computacional sea muy alto, por esta razón se hace necesaria la introducción de nuevos modelos los cuales puedan solucionar estos problemas.

Es por esto que la utilización de las Máquinas de Soporte Vectorial para la predicción de series de tiempo ha tomado cada vez más fuerza debido a su capacidad de generalización, la cual es

función directa de su estructura y de la metodología utilizada para la estimación de sus parámetros, además de las ventajas que tiene sobre otras técnicas no lineales:

Su especificación se basa en el principio de minimización del riesgo estructural que equivale a minimizar el límite superior del error de generalización del modelo. Así, la estimación de una SVM busca encontrar tanto la estructura óptima del modelo como los valores óptimos de sus parámetros, permitiendo una mayor capacidad de generalización del modelo. La estimación de los parámetros de otros modelos tradicionales, tales como los perceptrones multicapa (MLP, por sus siglas en inglés), se basa en la aplicación del principio de minimización de riesgo empírico que depende fundamentalmente del ajuste a los datos históricos, tomando la estructura del modelo como un parámetro de entrada.

La estimación de los parámetros de una SVM es equivalente a la solución de un modelo de programación cuadrática con restricciones lineales; ello implica, que la solución óptima es global y única, lo cual representa una clara ventaja sobre modelos como los MLP que se caracterizan por poseer múltiples puntos de mínimo local. (Velásquez, Olaya, & Franco, 2010).

En la literatura, se han propuesto diversas metodologías que permiten la selección de los parámetros adecuados en las series de tiempo, sin embargo, estos dependen del tipo de problema. Por lo tanto, no existe una técnica que generalice dicha selección, anteriormente esta tarea se realizaba de forma empírica e intuitiva, lo que a su vez tomaba mucho tiempo y en algunos casos arrojaba porcentajes de error muy altos; es por eso que se ha venido trabajando en el desarrollo de técnicas que permitan optimizar la selección de los parámetros. Una técnica que ha venido tomando fuerza en los últimos años es la optimización por enjambre de partículas (PSO). Este

algoritmo se implementa de forma fácil, utilizando pocas líneas de codificación, lo que lo hace altamente eficiente computacionalmente, además de tener una alta robustez en la selección de parámetros. De acuerdo con lo anterior se evidencia la importancia de investigar métodos que permitan realizar una selección de parámetros, en este caso PSO, de manera que pueda utilizarse en diferentes problemáticas de series temporales eficientemente con tiempos competitivos, por este motivo se espera que este proyecto sirva para futuras investigaciones.

3. Objetivos

3.1. Objetivo general

Desarrollar un marco de trabajo para la selección de parámetros de un modelo de máquinas de soporte vectorial en un problema de pronósticos de series de tiempo.

3.2. Objetivos específicos

- Realizar una revisión de la literatura sobre los modelos de máquinas de soporte vectorial aplicados para el pronóstico de series de tiempo.
- Consolidar bases de datos de prueba para la evaluación de modelos de pronóstico para series de tiempo.
- Construir una metaheurística basada en la optimización de enjambre de partículas (PSO), para integrar el marco de trabajo que permita la selección de parámetros para modelos de máquinas de soporte vectorial SVM en problemas de pronóstico de series de tiempo.
- Comparar el desempeño de la metaheurística con metodologías de selección de parámetros existentes.
- Elaborar un artículo de carácter publicable sobre los resultados obtenidos en la investigación.

4. Marco teórico

4.1. Modelos de Regresión

4.1.1. Regresión Lineal. Un modelo de regresión predice valores de una variable respuesta continua o discreta en base a una serie de variables denominadas predictoras. Por ejemplo, los modelos de regresión hacen predicciones que responden a preguntas como las siguientes: a). ¿Cuál es el valor de una casa en California?, b). ¿Cuál es la probabilidad de que un usuario haga clic en un anuncio? La regresión lineal es un método que permite determinar los parámetros de un modelo matemático que aparecen de manera lineal en la ecuación matemática propuesta. La ecuación matemática bien puede ser una línea recta o el hiperplano que mejor se ajusta a un conjunto de puntos. Por ejemplo, en el caso de querer tener un sistema que pueda predecir el precio de un automóvil usado. Las entradas son los atributos del automóvil: marca, año, capacidad del motor, kilometraje y otra información que afectan el valor de un automóvil; la salida es el precio del auto.

Como se observa en la figura 1 la variable X indica el kilometraje del automóvil y la variable Y es el precio del automóvil. Nuevamente examinando las transacciones pasadas, podemos recopilar datos de entrenamiento y el programa de aprendizaje automático ajusta una función a estos datos para aprender Y en función de X; esto se puede observar en la gráfica de la Figura 1. (Ethem, 2014).

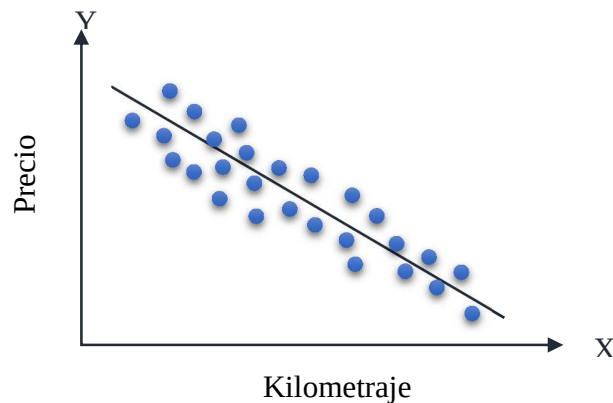


Figura 1. Regresión lineal del Precio de un automóvil. Adaptado de Alpaydin, Ethem (2014, p.10).

4.1.2. Regresión lineal simple. El tipo de análisis de regresión que se conoce como regresión lineal simple es aquel en el que interviene una variable independiente (variable que se usa para predecir el valor de la variable dependiente) y una variable dependiente (variable que se va a predecir) y en el que la relación entre estas variables es aproximada mediante una línea recta. La ecuación que describe la relación entre la variable dependiente y la variable independiente x como se denomina modelo regresión lineal, para el caso es el siguiente:

$$y = \beta_0 + \beta_1 x + \epsilon. \quad (1)$$

β_0 y β_1 se conocen como los parámetros del modelo, y ϵ (la letra griega épsilon) es una variable aleatoria que se conoce como término del error. El término del error da cuenta de la variabilidad de y que no puede ser explicada por la relación lineal entre x y y .

La variable dependiente puede verse también como una colección de subpoblaciones, una para cada uno de los valores de x . A la ecuación que describe la relación entre el valor esperado de y ,

que se denota $E(Y)$, y x se le llama ecuación de regresión. La siguiente es la ecuación de regresión para la regresión lineal simple:

$$E(y) = \beta_0 + \beta_1 x. \quad (2)$$

La gráfica de la ecuación de regresión lineal simple es una línea recta; β_0 es la intersección de la recta de regresión con el eje y , β_1 es la pendiente y $E(y)$ es la media o valor esperado de y para un valor dado de x . En la Ilustración 10, se presentan ejemplos de posibles rectas de regresión. La recta de regresión de la gráfica A indica que el valor medio de y está relacionado positivamente con x .

La recta de regresión de la gráfica B indica que el valor medio de y está relacionado negativamente con x , valores menores de $E(y)$ corresponden a valores mayores de x . La recta de regresión de la gráfica C muestra el caso en el que el valor medio de y no está relacionado con x ; es decir, el valor medio de y es el mismo para todos los valores de x . Algunos ejemplos de regresión lineal simple se muestran en la figura 2. (Anderson, 2008).

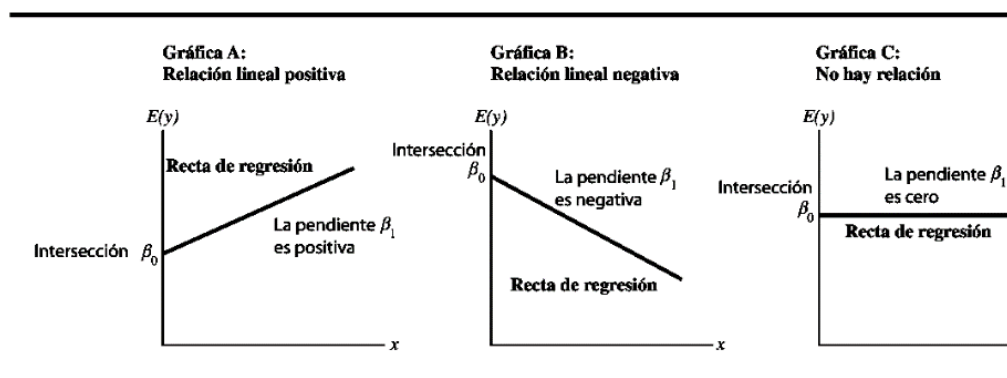


Figura 2. Líneas de regresión en la regresión lineal simple. Adaptado de Anderson, Sweeney y Williams (2008, p.546).

4.1.3. Regresión Múltiple. El análisis de regresión múltiple es el estudio de la relación de una variable dependiente con dos o más variables independientes, de tal manera que permite tomar más factores en consideración. Para denotar el número de variables independientes se suele usar p .

A la ecuación que describe cómo está relacionada la variable dependiente y con las variables independientes $x_1, x_2, \dots, x_p$ se le conoce como modelo de regresión múltiple. Se supone que el modelo de regresión múltiple toma la forma siguiente:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + \epsilon \quad (3)$$

En el modelo de regresión múltiple, $\beta_0, \beta_1, \beta_2, \dots, \beta_p$, son parámetros y el término del error ϵ (la letra griega épsilon) es una variable aleatoria. Examinando con atención este modelo se ve que y es una función lineal de los parámetros desconocidos $\beta_i, i = 0, 1, \dots, p$ más el término del error ϵ . El término del error corresponde a la variabilidad en y que no puede atribuirse o explicarse al efecto lineal de las p variables independientes.

Uno de los supuestos del modelo de regresión múltiple es que la media o valor esperado de ϵ es cero. Una consecuencia de este supuesto es que la media o valor esperado de y , que se denota $E(y)$, es igual a $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$. A la ecuación que describe cómo está relacionada la media de y con $x_1, x_2, \dots, x_p$ se le conoce como ecuación de regresión múltiple:

$$E(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p \quad (4)$$

4.2. Algoritmos de optimización

Al abordar problemas de optimización donde el espacio de posibles soluciones se centra en encontrar un máximo o un mínimo dadas unas restricciones, se hace necesario utilizar métodos que permitan obtener los resultados deseados. A continuación, se explican dos algoritmos que se caracterizan por darle solución este tipo de problemas; las heurísticas y las metaheurísticas.

4.2.1. Heurística. El término heurística se deriva de la palabra griega “heuriskein” que significa encontrar o descubrir y se usa en el ámbito de la optimización para describir una clase de algoritmos de resolución de problemas. Con un método heurístico se puede obtener una buena solución en un tiempo razonable y es mucho más flexible que un método exacto, puesto que permite, por ejemplo, la incorporación de condiciones de difícil modelización. El método heurístico se utiliza como parte de un procedimiento global que garantiza el óptimo de un problema. Existen dos posibilidades:

1. Proporciona una buena solución inicial de partida.
2. Participa en un paso intermedio del procedimiento.

4.2.2. Metaheurística. Una metaheurística se refiere a heurísticas generales que pueden adaptarse fácilmente a una clase muy variada de problemas de optimización. Más allá de este uso variado y una implementación relativamente simple, la ventaja principal de las metaheurísticas es que son capaces, en contraste con la mayoría de las heurísticas estándar, de evitar ser bloqueadas en un óptimo local de la función que se está optimizando y, por lo tanto, se acerca a un óptimo

global de este. Un algoritmo metaheurístico cuenta generalmente con dos componentes principales: La selección de las mejores soluciones y la aleatorización. Por una parte, la selección de los mejores garantiza que las soluciones convergerán a la optimalidad, mientras que la aleatorización evita que las soluciones queden atrapadas en los óptimos locales y, al mismo tiempo, aumente la diversidad de las soluciones.

4.3. Aprendizaje Automático

El aprendizaje automático se puede definir ampliamente como métodos computacionales que utilizan la experiencia (información pasada) para mejorar el rendimiento o para hacer predicciones precisas, combinando conceptos fundamentales en informática con ideas de estadísticas, probabilidad y optimización con el fin de diseñar algoritmos eficientes y robustos para producir estas predicciones, incluso para problemas a gran escala.

4.3.1. Aprendizaje Supervisado. Es aquel que toma un conjunto de datos con entrada y respuesta. Entrena un modelo para generar predicciones razonables para la respuesta a nuevos datos, su objetivo es construir un modelo que hace predicciones basadas en evidencia en la presencia de incertidumbre.

El aprendizaje supervisado usa técnicas de clasificación y regresión para desarrollar modelos predictivos.

Las técnicas de clasificación predicen respuestas discretas, por ejemplo, si un correo electrónico es genuino o no deseado, o si un tumor es canceroso o benigno. Los modelos de

clasificación clasifican los datos de entrada en categorías. Las aplicaciones típicas incluyen imágenes médicas, reconocimiento de voz, y puntuación de crédito.

Las técnicas de regresión predicen respuestas continuas, por ejemplo, cambios en la temperatura o fluctuaciones en la demanda de energía. Las aplicaciones típicas incluyen electricidad previsión de carga y negociación algorítmica.

4.3.2. Aprendizaje No supervisado. Encuentra patrones ocultos o estructuras intrínsecas en datos. Se utiliza para extraer inferencias a partir de conjuntos de datos que consisten en datos de entrada sin respuestas etiquetadas.

El clustering (agrupación) es la técnica de aprendizaje no supervisada más común, se utiliza para el análisis exploratorio de datos para encontrar patrones o agrupaciones ocultos en los datos. Las aplicaciones para clustering incluyen análisis de secuencia de genes, mercado investigación y reconocimiento de objetos.

4.4. Series de Tiempo

Una serie de tiempo es un conjunto de observaciones que se toman de manera secuencial a través del tiempo. Estas observaciones pueden tener diversos orígenes a partir de números, etiquetas, colores y muchos otros. Por otro lado, las observaciones son tomadas en tiempos los cuales pueden estar regular o irregularmente espaciados, donde este puede ser continuo o discreto.

Básicamente, lo que se pretende con el estudio univariante de las series temporales es el conocimiento de una variable a lo largo del tiempo para, a partir de este conocimiento y bajo el supuesto de que no van a producirse cambios estructurales, poder realizar predicciones de ella.

4.4.1. ARMA. Los modelos ARMA son herramientas fundamentales para analizar series temporales de memoria corta. Se puede demostrar que esta clase de modelos se aproxima arbitrariamente a cualquier proceso estacionario lineal con densidad espectral continua. Además, existe una gran cantidad de herramientas numéricas y computacionales para ajustar, diagnosticar y pronosticar modelos ARMA. Han sido muy útiles para modelar un gran número de series de tiempo que exhiben una dependencia débil. (Palma, 2016). Sea Y_t una variable aleatoria que sigue un proceso estocástico autorregresivo de primer orden, o AR. Aquí el valor de Y en el tiempo t depende de su valor en el periodo anterior y de un término aleatorio; los valores de Y están expresados como desviaciones de su valor medio. En otras palabras, el modelo dice que el valor de pronóstico de Y en el periodo t es simplemente alguna proporción (α_1) de su valor en el periodo $(t - 1)$ más un “choque” o perturbación aleatoria en el tiempo t ; de nuevo, los valores de Y están expresados alrededor del valor de su media. En todos los modelos sólo se consideran los valores actuales y anteriores de Y ; no hay otras regresoras. En este sentido, decimos que “los datos hablan por sí mismos”. Suponga el modelo para Y de la siguiente manera:

$$Y_t = \mu + \beta_0 u_t + \beta_1 u_{t-1} \quad (5)$$

Donde μ es una constante y u_t , es el término de error estocástico de ruido blanco. Aquí, Y en el periodo t es igual a una constante más un promedio móvil de los términos de error presente y pasado, es decir del término error. Así, que en este caso se dice que Y sigue un proceso de promedios móviles de primer orden, o MA(1). En forma más general:

$$Y_t = \mu + \beta_0 u_t + \beta_1 u_{t-1} + \beta_2 u_{t-2} + \dots + \beta_q u_{t-q} \quad (6)$$

Es un proceso MA(q). En resumen, un proceso de promedios móviles es tan sólo una combinación lineal de términos de error de ruido blanco.

Desde luego, es muy probable que, Y tenga características de AR y de MA a la vez, y, por consiguiente, sea ARMA. Así, Y_t sigue un proceso ARMA (1, 1) si se escribe como:

$$Y_t = \theta + \alpha_1 Y_{t-1} + \beta_0 u_t + \beta_1 u_{t-1} \quad (7)$$

Debido a que hay un término autorregresivo y uno de promedios móviles. En general, en un proceso ARMA (p, q), habrá p términos autorregresivos y q términos de promedios móviles. (Gujarati & Porter, 2013).

4.4.2. ARIMA. Los modelos de series de tiempo analizados se basan en el supuesto de que las series de tiempo consideradas son (débilmente) estacionarias. Es decir, la media y la varianza de ARIMA. Los modelos de series de tiempo analizados se basan en el supuesto de que las series de tiempo consideradas son (débilmente) estacionarias. Es decir, la media y la varianza de la serie de tiempo son constantes y su covarianza es invariante en el tiempo. Sin embargo no todas las series de tiempo económicas son no estacionarias, es decir, son integradas. Por consiguiente, si debemos diferenciar una serie de tiempo d veces para hacerla estacionaria y luego aplicarle el modelo ARMA (p,q), una serie de tiempo original es ARIMA(p, d, q), es decir, es una serie de tiempo autorregresiva integrada de promedios móviles, donde p denota el número de términos autorregresivos, del número de veces que la serie debe diferenciarse para hacerse estacionaria y q el número de términos de promedios móviles (Guajarati & Porter, 2013).

4.5. Máquinas de Soporte Vectorial

Las máquinas de soporte vectorial (SVM) son modelos de aprendizaje supervisado, utilizados para problemas de clasificación y regresión. Una máquina de soporte vectorial construye un hiperplano o un conjunto de hiperplanos en un espacio de alta dimensión, que puede usarse para tareas de clasificación, regresión u otras tareas. Intuitivamente, se logra una buena separación mediante el hiperplano que tiene la mayor distancia a los puntos de datos de entrenamiento más cercanos de cualquier clase (llamado margen funcional), ya que, en general, cuanto mayor es el margen, menor es el error de generalización del clasificador.

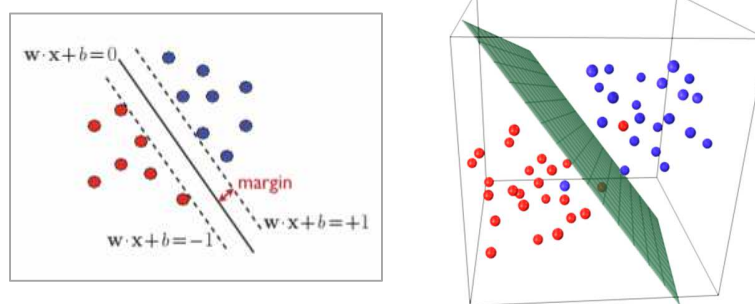


Figura 3. Hiperplano de margen máximo canónico. Adaptado de Mohri, Mehryar, et al. (2014).

Se utilizan diversas técnicas para validar los métodos de regresión, tales como; la comparación de los parámetros obtenidos mediante modelos físicos, teóricos o con simulaciones, o enumeración exhaustiva combinada con validación cruzada; esta última es una manera de predecir el ajuste de un modelo a un hipotético conjunto de datos de prueba cuando no disponemos del conjunto explícito de dichos datos (Refaeilzadeh, Tang, & Lui, 2008).

La interfaz que utiliza R, para resolver el problema de optimización de las SVM es el paquete e1071, el cual fue diseñado para ser lo más intuitivo posible y donde se encuentra la función SVM. Su tarea es ajustar los modelos y a partir de nuevos datos hacer un pronóstico. Todos los problemas

se formulan como problemas de optimización cuadrática los cuales se resuelven mediante técnicas conocidas.

Una de las extensiones que actualmente están incluidas en la librería libsvm y es utilizada para resolver el problema de pronósticos de series de tiempo es el ε – *Regression*. En este caso los puntos de datos se encuentran entre los dos bordes del margen que se maximizan en condiciones adecuadas para evitar una inclusión atípica. En la ecuación 7 se ve la formulación del modelo ε – *Regression* que utiliza el paquete e1071 en R.

$$\min_{\alpha, \alpha^*} \frac{1}{2}(\alpha - \alpha^*)^T Q(\alpha - \alpha^*) + \epsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l y_i(\alpha_i - \alpha_i^*) \quad (8)$$

s.a.

$$0 \leq \alpha_i, \alpha_i^* \leq C, \quad i = 1, \dots, l, \quad \sum_{i=1}^l (\alpha_i + \alpha_i^*) = 0 \quad (9)$$

4.5.1. Regresión Vectorial de Soporte. Dentro de las SVM, se encuentra un modelo de máquinas de soporte vectorial para regresión; propuesto en 1997 por Vapnik, Steven Golowich y Alex Smola, conocido como Support Vector Regression (SVR), este modelo es la forma de aplicación más común de las SVM e intenta minimizar el límite de error de generalización para lograr un rendimiento generalizado. La idea de SVR se basa en el cálculo de una función de regresión lineal en un espacio de características de alta dimensión, donde los datos de entrada se asignan a través de una función no lineal. (Basak, D., Pal, S. & Patranabis, D. C, 2007).

Los resultados efectivos de las máquinas de soporte vectorial para problemas de clasificación se trasladaron para abordar problemas de regresión, donde existen dos clases: ν -regresión y ε -regresión. En un modelo SVR, se selecciona el hiperplano regresor que mejor se ajuste al conjunto

de datos de entrenamiento (Martín, 2016, p. 15). y se explica de la siguiente manera: Considere un conjunto de datos de entrenamiento $\{ (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \}$, donde $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, $i = 1 \dots n$. Los datos de entrenamiento x_i del espacio de entrada X se asignan a un espacio de características Q tal como $\theta: x_i \rightarrow (\cdot)$ usando una función no lineal predefinida $\phi(x)$. Sea f la función lineal que tiene la forma como en la ecuación 10:

$$f(x) = w^T x + b = \langle w, x \rangle + b \quad (10)$$

Donde $w \in X$, $b \in \mathbb{R}$, $\langle \cdot, \cdot \rangle$ denota producto punto. (Thomas, Pillai, y Pal, 2017).

SVR - Regresión ϵ -insensitive. Se define como una función de perdida la cual ignora los errores que se encuentran dentro de una cierta distancia del valor verdadero. Esta función se conoce como función de perdida ϵ -insensiva, donde ϵ es el error de generalizar, es decir, la probabilidad de clasificar erróneamente un ejemplo de prueba dibujado al azar. Las variables ξ miden el costo de los errores en los puntos de entrenamiento. Estos toman un valor cero para todos los puntos dentro de la banda. En la figura 4 se muestra un ejemplo de una regresión lineal unidimensional con banda insensiva ϵ .

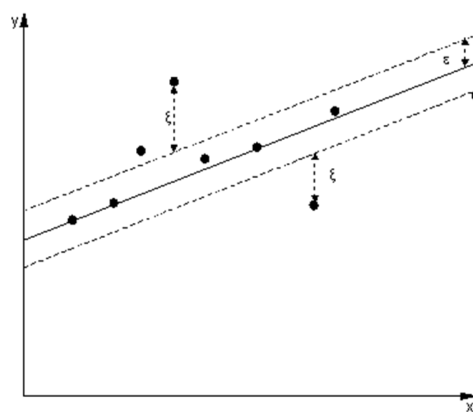


Figura 4. Regresión lineal unidimensional con banda insensiva ϵ . Adaptado de Support Vector Machines and other kernel-based learning methods.

4.5.2. Función Kernel. En aprendizaje automático y estadísticas, el término kernel se usa para definir un algoritmo de reconocimiento de patrones. Cualquier solución de métodos kernel consta de dos partes: Un módulo que realiza la asignación en el espacio inmerso o de características y un algoritmo de aprendizaje diseñado para descubrir patrones lineales en ese espacio. Hay dos razones principales por las que este enfoque debería funcionar. En primer lugar, la detección de relaciones lineales ha sido el centro de muchas investigaciones en estadística y aprendizaje automático durante décadas, y los algoritmos resultantes son bien entendidos y eficientes. En segundo lugar, existe un atajo computacional que permite representar patrones lineales de manera eficiente en espacios de alta dimensión para asegurar un poder de representación adecuado. El atajo es lo que llamamos una función kernel. Esta puede detectar patrones estables de manera robusta y eficiente a partir de una muestra de datos finitos. La estrategia adoptada es integrar los datos en un espacio donde los patrones se pueden descubrir como relaciones lineales. Esto se hará de forma modular. Dos componentes distintos realizarán los dos pasos. El componente de mapeo inicial se define implícitamente mediante la llamada función kernel. Este componente dependerá del tipo de datos específico y el conocimiento del dominio con respecto a los patrones que se esperan de la fuente de datos en particular.

En la figura 5, la función ϕ mapea los datos en un espacio de características donde el patrón no lineal ahora parece lineal. El kernel calcula productos internos en el espacio de características directamente desde las entradas.

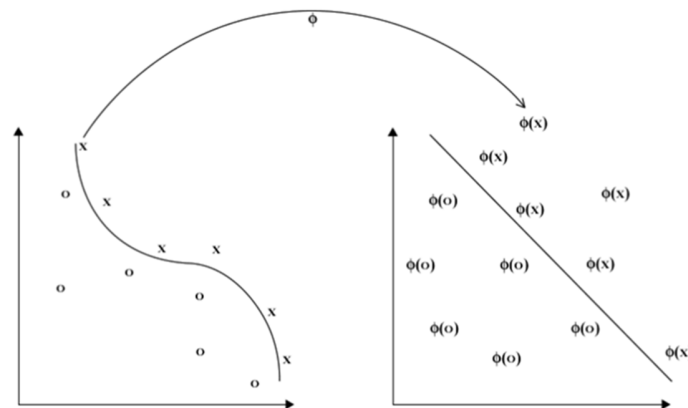


Figura 5. Regresión lineal en un espacio de características. Adaptado de Shawe-Taylor, John, and Nello Cristianini (2004).

Tabla 2.

Tipos de Funciones Kernels.

FUNCIÓN KERNEL	FÓRMULA	PARÁMETROS
Lineal	$k(x, x') = x * x'$	Ninguno
Polinomial	$k(x, x') = \gamma(x * x' + C_0)^d$	γ, d, C_0
Radial	$k(x, x') = e^{-\gamma x-x' ^2}$	γ
Sigmoidal	$k(x, x') = \tan \gamma(x * x' + C_0)$	γ, C_0

Nota: Muestra cada uno de los parámetros a utilizar según el tipo de función Kernel.

4.6. Métodos de Validación

Cross Validation. La validación cruzada es una técnica para validar métodos de regresión (Kozak and Kozak, 2003). La idea es dividir los datos para estimar el riesgo de cada algoritmo. Se utiliza una muestra de entrenamiento que entrena el algoritmo, y la muestra restante llamada de validación se usa para estimar el riesgo del algoritmo, finalmente selecciona el algoritmo con el menor riesgo estimado. (Arlot, S., Celisse, A. 2010).

Hold-Out. El método hold-out separa en dos subconjuntos los datos disponibles, uno se encarga de entrenar el modelo y el otro realiza el test de validación (Arlot y Celisse, 2010). Este método crea un modelo con los datos de entrenamiento el cual genera unos datos de salida que se comparan con el subconjunto de datos reservados para el test de validación. Los estadísticos que se obtienen del subconjunto de validación son los que dan la validez del modelo en términos de error. (Pérez-Planells et al, 2015).

K-Folds. El método K-fold, está basado en el método Hold-Out, con una utilidad mayor en cuanto el conjunto de datos es pequeño (Yang y Huang, 2014). Idealmente si se cuentan con suficientes datos, se subdividen estos datos y se obtiene un subconjunto de validación el cual evalúa el rendimiento del modelo de predicción. Dado que los datos frecuentemente son escasos, esto no es posible. Para solucionar este problema el modelo k-fold usa parte de los datos disponibles para adaptarse al modelo, y una parte diferente para probarlo (Hastie, Tibshirani, & Friedman, 2001). El conjunto total de datos se divide en k subconjuntos, de tal manera que se aplica el método hold-out k veces, utilizando cada vez un subconjunto distinto para validar el modelo entrenado con los otros k-1 subconjuntos (Jung y Hu, 2015). El error medio obtenido de los k análisis realizados nos proporciona el error cometido por el método, permitiendo así evaluar su validez. (Pérez-Planells et al, 2015).



Figura 6. Modelo de Cross Validation. Adaptado de Python Model Tuning Methods Using Cross Validation and Grid Search. (2018).

4.7. Metodologías para la selección de Parámetros

Dado que las máquinas de soporte vectorial han demostrado una gran capacidad para la clasificación y reconocimiento de patrones en general, se han venido utilizando con el fin de obtener un algoritmo óptimo que permita encontrar un modelo final de regresión, asociado a problemas de optimización de una serie de parámetros; uno de ellos, denotado C , que controla la compensación entre la maximización de margen y la penalización del error. Otros parámetros aparecen en el mapeo no lineal en el espacio de características, llamados parámetros kernel, los cuales se establecen para entrenar el modelo; dicha selección de parámetros resulta ser un trabajo laborioso de realizar y para ello existen diversas metodologías, entre las que podemos encontrar:

4.7.1. Búsqueda de cuadrículas o método grid. La búsqueda grid o búsqueda de cuadrícula se basa en una búsqueda exhaustiva de un subconjunto especificado manualmente de parámetros dentro del espacio de un algoritmo de aprendizaje. (Hsu, Chang, & Lin, 2016). Se utiliza cuando el máximo requerido se encuentra dentro de un área finita definida por la parte superior y límites inferiores de cada una de las variables independientes. Este método buscaría sistemáticamente todos los estados posibles. Para este propósito, se debe establecer una rejilla sobre el área de interés y evaluar la función objetivo en cada nodo de la rejilla. Luego de calcular los valores de la función objetivo en todos los nodos de la rejilla, es posible interpolar y encontrar un máximo entre las líneas de la cuadrícula. Este método es muy simple de programar, pero poco eficiente porque requiere un tiempo de cálculo excesivo. Sin embargo, se puede usar para encontrar un buen espacio de inicio para uno de los métodos más eficientes. (Ataei & Osanloo, 2012).

4.7.2. Algoritmos genéticos. Los algoritmos genéticos (GA) se introdujeron formalmente en los Estados Unidos en la década de 1970 por John Holland en la Universidad de Michigan. Estos son estocásticos, y se basan en la teoría de la selección natural y el proceso de evolución. Un algoritmo genético comienza creando la primera población de individuos (representada por cromosomas) que están dispersos aleatoriamente sobre el espacio de búsqueda. Luego los pasos sucesivos evolutivos, conocidos como generaciones, se producen de la siguiente manera: (i) decodificar cada cromosoma para obtener la función objetivo y evaluar de acuerdo con algunos criterios de calidad predefinidos, denominados aptitud o función objetivo, (ii) clasificar los cromosomas según su evaluación de la función objetivo, (iii) utilizar alguna estrategia de mejoramiento para combinar buenos cromosomas y descartar los malos, permitiendo que los mejores cromosomas mejoren con una mayor probabilidad. Esto genera un nuevo grupo de

individuos, (iv) aplicar una pequeña cantidad de mutación a los cromosomas. Esto está motivado por la esperanza de que nueva población será mejor que la anterior. Esto se repite hasta que algunas condiciones tales como el número de poblaciones o la mejora de la mejor solución sean satisfechas.(Ataei & Osanloo, 2012).

4.8. Particle Swarm Optimization

La metaheurística de Optimización por Enjambre de Partículas fue propuesta por Kennedy y Eberhart (1995), como una de las técnicas basada en poblaciones e inspirada en la naturaleza, que simula el comportamiento de los individuos en un grupo para maximizar la supervivencia de las especies. En el PSO, cada solución se denomina partícula, el grupo de partículas es la población o enjambre y el espacio de búsqueda es el área a explorar.

La optimización empieza con una población inicial de soluciones, que se actualiza en cada iteración debido al movimiento de las partículas de su posición original a una nueva posición, cada partícula "vuela" en una dirección que se basa en su experiencia particular y en la del grupo completo. Las partículas individuales se mueven estocásticamente hacia la posición afectada por la velocidad actual, el mejor rendimiento anterior y el mejor rendimiento previo del grupo. (Chen, 2003) (Kennedy & Eberhart, 2001).

La velocidad de cada partícula es influenciada por la información social y cognitiva de las partículas. La información social se refiere a la mejor posición encontrada por el enjambre, y se denomina gbest; por su parte, la información cognitiva es la mejor posición que la partícula ha visitado y se denomina pbest.

Cada partícula debe cambiar la velocidad (aceleración) de sus ubicaciones pbest y gbest. La aceleración es ponderada por un término aleatorio, con números aleatorios separados que se generan para la aceleración hacia las ubicaciones pbest y gbest.

Existe también una versión de PSO en la que se involucra un componente adicional de información social en la velocidad de las partículas; este componente se denomina lbest y se ha contemplado en muy pocas investigaciones. En el lbest del PSO, cada partícula tiene una vecindad más pequeña con solo un número limitado de miembros del enjambre, con los cuales la partícula tiene comunicación. La comunicación es usualmente relacionada con los índices de las partículas; así, si el tamaño de la vecindad es igual a tres, las vecinas seleccionadas para la partícula i son las partículas $i-1$, $i+1$ del enjambre. Muchas veces se prefiere el uso de los índices de las partículas para la creación de la vecindad, porque evita la complejidad y el gasto computacional de calcular las distancias entre todas las partículas para identificar la vecindad de cada una de ellas. De esta manera, la velocidad de cada partícula se actualiza según la información obtenida de la mejor partícula de la vecindad. Mientras el gbest del PSO converge rápidamente, el lbest proporciona mayor diversidad a las soluciones y les hace más difícil quedar atrapadas en óptimos locales. (Marinakis, Iordanidou & Marinaki, 2013). La forma como se representa un enjambre de partículas es la siguiente:

D: Dimensión de la partícula

n: Número de partículas

X_i : (x_{i1} , x_{i2} , ..., x_{il} , ..., x_{iD}): Posición de la i -ésima partícula

V_i : (v_{i1} , v_{i2} , ..., v_{il} , ..., v_{iD}): Velocidad de la i -ésima partícula

P_i : (p_{i1} , p_{i2} , ..., p_{il} , ..., p_{iD}): Mejor solución de la i -ésima partícula (pbest)

Pli: (pli1, pli2, ..., plil, ..., pliD): Mejor solución local de la *i*-ésima partícula (lbest)

Pgi: (pgi1, pgi2, ..., pgil, ..., pgiD): Mejor solución global del enjambre de partículas (gbest)

La actualización de las posiciones y velocidades de las partículas en cada iteración se lleva a cabo de la siguiente manera: Sea la posición de la *i*-ésima partícula en la *l*-ésima dimensión, en el

$$x_{il}(t + 1) = x_{il}(t) + v_{il}(t + 1) \quad (11)$$

momento *t*. La actualización de la posición de la partícula en el instante *t+1*, se realiza mediante la fórmula:

Donde $V_{il}(t+1)$ es la velocidad de la *i*-ésima partícula en la *l*-ésima dimensión, en el momento *t+1*, y se actualiza a través de:

$$\begin{aligned} v_{il}(t + 1) = & w(t) \cdot v_{il}(t) + (c_p \cdot r)[p_{il}(t) - x_{il}(t)] + (c_l \cdot r)[p_{lil}(t) - x_{il}(t)] \\ & + (c_g \cdot r)[p_{gl}(t) - x_{il}(t)] \end{aligned} \quad (12)$$

cp, *cl* y *cg* son los factores de aprendizaje cognitivo y sociales, *r* es un número aleatorio con distribución uniforme en el intervalo [0, 1], *w(t)* es el factor de inercia en el momento *t*, *p_{il(t)}* es la mejor posición personal de la *i*-ésima partícula en el momento *t*, *p_{lil(t)}* es la mejor solución de la vecindad de la *i*-ésima partícula en el momento *t* y *p_{gl(t)}* se refiere a la mejor solución del enjambre en el momento *t*.

Las velocidades de las partículas en cada dimensión se sujetan a una velocidad máxima *Vmax*. Si la suma de las aceleraciones causa que la velocidad en esa dimensión exceda a *Vmax*, que es un parámetro especificado por el usuario, entonces la velocidad en esa dimensión se limita a *Vmax*. La velocidad máxima, determina la resolución o finura con la que se buscan las regiones entre la

posición actual y la posición objetivo (la mejor hasta ahora). Si V_{max} es demasiado alto, las partículas podrían pasar por buenas soluciones. Por otro lado, si V_{max} es demasiado pequeño, es posible que las partículas no exploren lo suficiente más allá de las regiones localmente buenas. De hecho, podrían quedar atrapados en los óptimos locales, incapaces de moverse lo suficiente para alcanzar una mejor posición en el espacio problemático.

Las constantes de aceleración cp , cl y cg en la ecuación 12 representan la ponderación de los términos de aceleración estocástica que tiran de cada partícula hacia las posiciones p_{best} y g_{best} . Por lo tanto, el ajuste de estas constantes cambia la cantidad de "tensión" en el sistema. Los valores bajos permiten que las partículas se desplacen lejos de las regiones objetivo antes de ser arrastradas hacia atrás, mientras que los valores altos dan como resultado un movimiento abrupto hacia o hacia las regiones objetivo.

La velocidad máxima V_{max} sirve como una restricción para controlar la capacidad de exploración global de un enjambre de partículas. Como se indicó anteriormente, un V_{max} más grande facilita la exploración global, mientras que un V_{max} más pequeño fomenta la explotación local. Por otro lado, el peso de inercia w permite un equilibrio entre la exploración y explotación global y local, y da lugar a menos iteraciones en promedio para encontrar una solución suficientemente óptima.

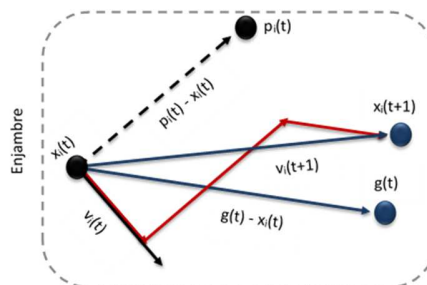


Figura 7. Movimiento de las partículas dentro del enjambre.

A continuación, se señalan los pasos que sigue un algoritmo de PSO:

Paso 1: Determinar la dimensionalidad del problema.

Paso 2: Establecer el número de partículas (n) en el enjambre.

Paso 3: Establecer número de iteraciones.

Paso 4: Realizar la primera iteración del enjambre, con posiciones y velocidades aleatorias de cada partícula.

Paso 5: Evaluar la función fitness de optimización deseada.

Paso 6: Comparar la posición obtenida por la partícula en la función fitness con la mejor posición que la partícula ha visitado ($pbest$).

Paso 7: Si el valor actual de la posición de la partícula es mejor que $pbest$, se actualiza el valor de $pbest$ por el valor actual.

Paso 8: Comparar la posición obtenida por el enjambre en la función fitness con la mejor posición encontrada por el enjambre ($gbest$).

Paso 9: Si el valor actual de la posición del enjambre es mejor que $gbest$, se actualiza el valor de $gbest$ por el valor actual.

Paso 10: Comparar la posición de la mejor partícula de la vecindad en la función fitness con la mejor posición encontrada por la vecindad ($lbest$).

Paso 11: Si el valor actual de la posición de la mejor partícula de la vecindad es mejor que

lbest, se actualiza el valor de *lbest* por el valor actual.

Paso 12: Cambiar la velocidad y la posición de la partícula.

Paso 13: Si los valores de *pbest*, *gbest* y *lbest* no mejoran en las iteraciones, no se actualiza el modelo.

Paso 14: Realizar todas las iteraciones establecidas en el modelo.

Paso 15: Obtener los resultados de todas las iteraciones y la gráfica de la función fitness.

5. Estado del arte

5.1. Análisis de la literatura

A través de los años se han dedicado diversos esfuerzos en la investigación del aprendizaje automático, la cual hace referencia al hecho de que los avances tecnológicos en hardware y software ha permitido que los ordenadores sean capaces de aprender a partir de datos disponibles, esto quiere decir, a través de la experiencia, todo esto resulta útil en la toma de decisiones, que se dan a partir de la solución a problemas en diferentes áreas de la ciencia, más recientemente en la predicción de series de tiempo. Vapnik (1982) fue la primera persona en desarrollar las Máquinas de Soporte Vectorial las cuales han recibido mucha atención en su aplicación en campos relacionados a la inteligencia artificial.

Cao & Wu, afirman que a pesar de que las redes neuronales sirven para resolver estos problemas, tienen varias desventajas, como son la necesidad de una gran cantidad de parámetros de control, así como una dificultad de encontrar una solución estable con un riesgo de sobreajuste. Gracias a Vapnik y a la introducción de la función pérdida ε -insensible, las SVR se han convertido en una herramienta poderosa para pronosticar y resolver sistemas no lineales. El rendimiento del pronóstico del modelo SVR está igualmente influenciado por el valor de sus parámetros, lo que puede conducir tanto a un ajuste excesivo como un ajuste insuficiente, lo cual fue demostrado por V. Cherkassky y Y. Ma. (2004). Debido a que no se han establecido pautas generales para establecer el valor de estos parámetros, en las últimas décadas se ha ido utilizando todo tipo de algoritmos inteligentes, muchos de los cuales se basan en el comportamiento de especie de animales, uno de ellos es el algoritmo de optimización por enjambre de partículas. Ellos concluyen en su investigación que este algoritmo se aplica para resolver tanto los valores de los parámetros, así como para mejorar el rendimiento del pronóstico (Cao & Wu, 2016).

Das, Achary, & Padhy hablan sobre la investigación que hicieron Sapankevych y Sankar en la cual publicaron una revisión exhaustiva sobre SVM para la predicción de los datos de series de tiempo. Examinaron diferentes series de tiempo para la predicción del mercado financiero, previsión de carga de servicios eléctricos, estados ambientales, predicción del clima y pronóstico de confiabilidad. A través de su revisión señalaron la importancia de la selección de los parámetros libres para SVM, así como la función kernel. Igualmente mencionan como en múltiples investigaciones la importancia de seleccionar valores óptimos de los parámetros, pues al no tener “buenos” parámetros se puede causar un ajuste excesivo o insuficiente. Introducen modelos

híbridos que utilizan las técnicas ya mencionadas de optimización para darle solución a estos problemas de ajuste (Das et al., 2016)

Wang, Jianzhou; Hou, Ru; Wang, Chen; et ál hablan en su investigación acerca de los pronósticos en el precio de las acciones, la importancia que tienen debido a que es una de las tareas más desafiantes en el ámbito financiero, proponen un modelo híbrido SVR con algoritmos de optimización de parámetros para no solo abordar el problema de la selección de parámetros si no también mejorar la precisión del pronóstico. Wang, Jianzhou; Hou, Ru; Wang, Chen; et ál concluyen en su investigación que los modelos de SVR integrados con algoritmos de optimización de parámetros obtienen errores de pronóstico mucho más bajos que utilizando parámetros por defecto. Por lo tanto, en su investigación apoyan la validez de utilizar algoritmos de inteligencia por enjambre para la selección de parámetros confirmando la necesidad de construir modelos de pronósticos híbridos (Wang, Hou, Wang, & Shen, 2016).

Alwee, Shamsuddin, & Sallehuddin, estudian el pronóstico de delitos en el campo de la criminología. Explican como en los diferentes modelos que existen actualmente sobre la predicción de estos como son los modelos ARIMA, ARMA y las Redes Neuronales son muy complejos y requieren una gran cantidad de datos para su correcta predicción; en su lugar introducen el estudio del modelado de las tasas de delincuencia mediante el uso de modelos de series de tiempo multivariante. En la investigación se trata sobre la regresión de vectores de soporte (SVR), que es más robusta para el tamaño de muestra pequeño. Construyen una hibridación entre los modelos ARIMA y SVR, en donde encuentran el problema de la configuración de sus parámetros, estos parámetros deben configurarse de manera correcta ya que pueden afectar la precisión de la regresión. Utilizan el algoritmo PSO para la selección de los

parámetros por su capacidad de escapar de los óptimos locales, de tener menores parámetros que deban ser ajustados y su facilidad de implementar. Alwee, Shamsuddin, & Sallehuddin, concluyen que el modelo propuesto es adecuado para la predicción de la tasa de robo debido a que muestra el mejor desempeño comparado con otros modelos. Igualmente hablan sobre como la aplicación del modelo no solo se limita al campo de la criminología por lo cual puede ser utilizado en la solución problemas de predicción de series de tiempo en cualquier campo (Alwee, Shamsuddin, & Sallehuddin, 2017).

Por otra parte, S. Meysam Mousavia, Behnam Vahdanib and Majid Abdollahzadec, hablan en su estudio sobre la predicción de costos del desarrollo de nuevos productos ya que esta se ha convertido en una actividad muy importante en las organizaciones modernas. Introducen las máquinas de soporte vectorial como una solución que ha recibido mucha atención con resultados notables. Utilizan específicamente la regresión de vectores de soporte SVR debido a que algunas de sus características, como lo son su menor número de parámetros ajustables y su velocidad de aprendizaje más rápida, permiten que sean más eficientes en comparación con las redes neuronales. Para ellos la importancia de la selección de los parámetros en un modelo SVR radica en el impacto negativo que puede producirse en los resultados. S. Meysam Mousavia, Behnam Vahdanib and Majid Abdollahzadec en su documento emplean un algoritmo evolutivo, PSO, en lugar de usar los algoritmos comunes, para acortar sus errores y fallas, debido a que el PSO es potente y eficiente en el campo de la computación. El modelo es validado mediante un conjunto de datos de un estudio de caso en una empresa manufacturera para predecir el costo de los proyectos NPD. Los investigadores concluyen en su investigación que el en su modelo propuesto el tiempo computacional utilizado se reduce drásticamente, así como también esta técnica es capaz

de evitar el ajuste excesivo o inadecuado en el modelo, finalmente el modelo presenta un menor error en la predicción, así como una mejor capacidad de generalización (Mousavi, S. M., Vahdani, B., & Abdollahzade, M. ,2015).

5.2. Marco de trabajo

En el marco de trabajo propuesto, se plantean las siguientes fases:

Fase 1. División del conjunto de datos: En esta fase se toma una muestra inicial de 242 datos que reflejan el valor del índice COLCAP en el lapso de un año, posteriormente se utiliza el método cross validation para separar los datos, de tal forma que se tenga un conjunto para el entrenamiento (Train Data) y otro para la prueba (Test Data) del modelo; cada conjunto contiene respectivamente el 70% y el 30% de la muestra inicial. Así mismo se seleccionan los parámetros (dependen de la función Kernel utilizada) a optimizar en la siguiente fase.

Fase 2. Optimización de los parámetros del modelo por enjambre de partículas: La fase 2 muestra el proceso para optimizar los parámetros del modelo de regresión de máquina de soporte vectorial mediante la optimización de enjambre de partículas.

La figura 9, muestra los parámetros que hacen referencia a los parámetros necesarios para que el algoritmo PSO pueda optimizar los parámetros del modelo SVR, entre ellos se encuentran: Número de partículas n , número máximo de iteraciones m , peso de inercia w , componente de auto-reconocimiento c_1 , componente social c_2 , donde c_1 y c_2 son constantes positivas, definidas como coeficientes de aceleración y toman valores entre 0 y 2 ($0 < c_1 + c_2 < 4$), posición mínima $xmin$, posición máxima $xmax$ y velocidad máxima $vmax$, de cada una de las partículas del enjambre; en

esta última se utiliza un intervalo cerrado $[0,1]$ para que pueda mapear los valores reales producidos en R debido a los operadores del vector velocidad (suma y producto). (Duarte, C. Quiroga, J., 2010). A continuación, se presentan los parámetros del PSO en cada función Kernel. Luego de determinar cada uno de estos parámetros, se evalúa la función fitness y se actualizan los valores $pbest$ y $gbest$ de acuerdo con los resultados del PSO.

Fase 3. Evaluación del desempeño del modelo mediante el cálculo del error RMSE:

Finalmente se escoge la función o tipo de Kernel a utilizar y se entrena el modelo con el conjunto de datos de entrenamiento para luego calcular el error cuadrático medio RMSE con los datos de prueba (esto se hace con cada una de las funciones Kernels).

Función Kernel Lineal							
n	m	w	c1	c2	Xmin	Xmax	Vmax
100	100	0.9	0.5	1	(0.1,0.1)	1,1	(0.1,0.1)

Función Kernel Radial							
n	m	w	c1	c2	Xmin	Xmax	Vmax
100	100	0.9	0.5	1	(0.1,0.1,0.1)	1,1,1	(0.1,0.1,0.1)

Función Kernel Sigmoid							
n	m	w	c1	c2	Xmin	Xmax	Vmax
100	100	0.9	0.5	1	(0.1,0.1,0.1,0.1)	1,1,1,1	(0.1,0.1,0.1,0.1)

Función Kernel Polynomial							
n	m	w	c1	c2	Xmin	Xmax	Vmax
100	100	0.9	0.5	1	(0.1,0.1,0.1,0.1,0.1)	1,1,1,1,1	(0.1,0.1,0.1,0.1,0.1)

Figura 8. Rango de parámetros seleccionados para cada función Kernel en el modelo SVR-GRID.

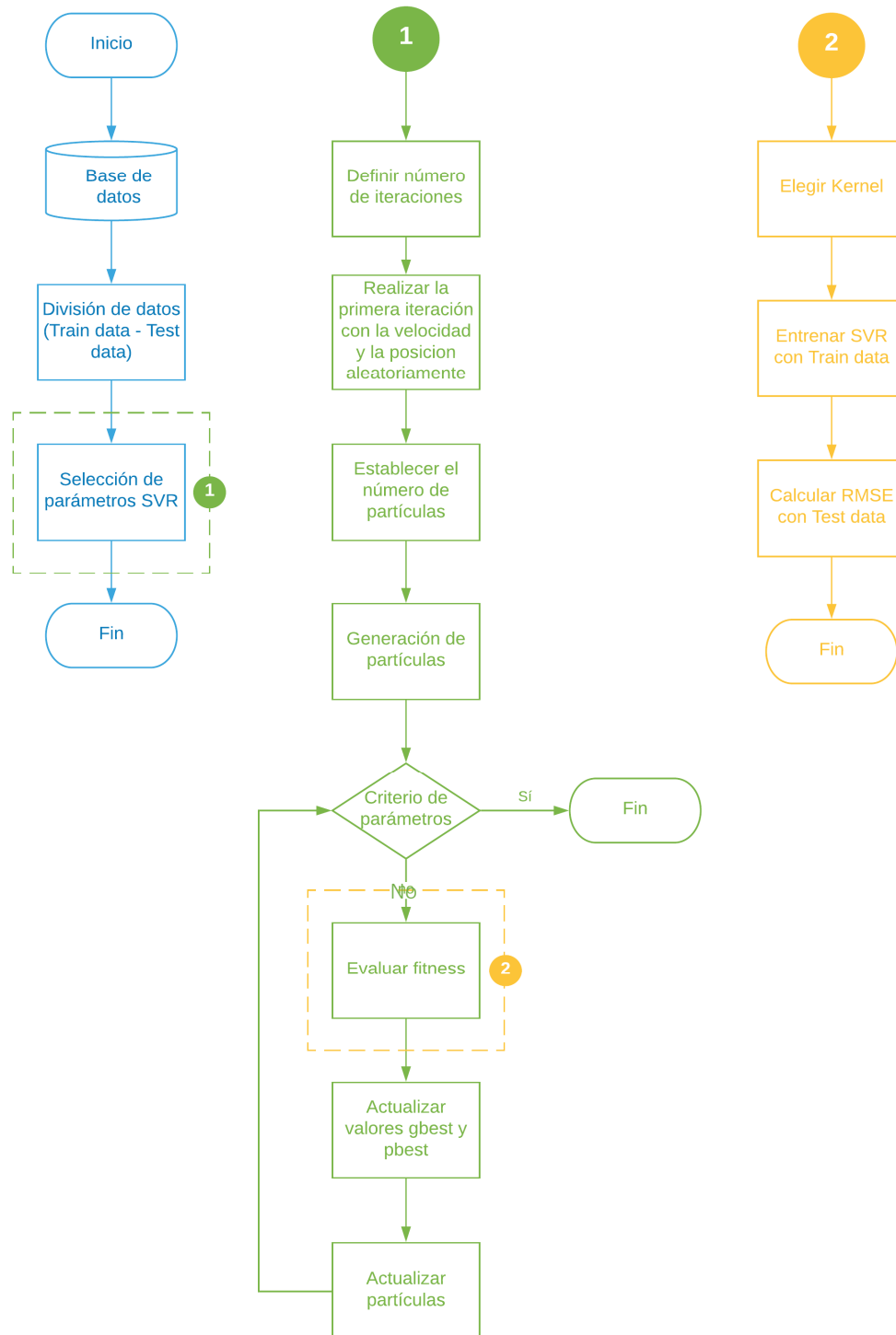


Figura 9. Diagrama de Flujo del Modelo SVR_PSO.

6. Resultados Computacionales

6.1. Resultados computacionales

En este proyecto se utiliza la base de datos de la Bolsa de Valores de Colombia (BVC), la cual contiene información relevante acerca del valor de las acciones que se manejan en Colombia, la regulación del mercado, el mercado integrado Latinoamericano (MILA), los índices bursátiles de los mercados nacionales e internacionales, noticias financieras, entre otros. En el caso particular del presente trabajo, se usa el índice COLCAP, el cual refleja las variaciones de los precios de las acciones más líquidas de la BVC, donde la participación de cada acción en el índice está determinada por el correspondiente valor de la capitalización bursátil ajustada (flotante de la compañía multiplicada por el último precio). Teniendo en cuenta que un índice bursátil es una serie de tiempo, se desarrollaron dos algoritmos de regresión de SVM que permitan optimizar los parámetros de predicción en cada pronóstico. Para ambos algoritmos, se tomó una muestra de 242 datos, que reflejan el valor que tomó el precio del índice COLCAP durante un año, a partir del 10 de octubre del año 2017, hasta el 10 de octubre de 2018. Los valores presentan una variación diaria y se tuvieron en cuenta los dos periodos anteriores para el entrenamiento del modelo (el valor del día ayer y el valor del día antes de ayer). Posteriormente se realizó una división de los datos mediante el método validación cruzada, utilizando la estrategia Hold-Out, utilizando un 70% para entrenar el modelo (Data Train) y el 30% restante para evaluarlo (Data Test).

Como se mencionó anteriormente en esta sección el problema de predicción de series de tiempo se resuelve por medio de modelos de máquinas de soporte vectorial para regresión SVR, implementando dos modelos para la solución de la selección de sus parámetros, el primer algoritmo se basa en un modelo SVR con un ajuste de parámetros por medio del método grid (tune) y el segundo es un algoritmo de Optimización por enjambre de partículas SVR_PSO.

Se presentan los resultados de cada uno de los modelos propuestos, cada una de sus características, el tiempo computacional utilizado y por último la Raíz del Error Cuadrático Medio (RMSE) para cada uno de ellos. La programación se realizó en un computador Lenovo memoria instalada (RAM) de 6,00 GB por medio del programa RStudio versión 1.1.456.

Se demuestra que el modelo SVR-PSO tiene una gran ventaja respecto al modelo SVR-GRID debido tanto a su tiempo computacional utilizado como a su RMSE que al compararlo resulta ser menor, también se demuestra que necesita un menor número de parámetros a escoger por parte del programador lo que hace que su programación resulte más sencilla y exacta.

A continuación, se detallan los resultados de cada uno de los modelos.

6.2. Resultados modelo SVR- GRID

El modelo Regresión de máquinas de soporte Grid (SVR-GRID) consta de realizar una búsqueda exhaustiva de un subconjunto especificado de parámetros dentro del espacio de un algoritmo de aprendizaje, en este se establecen rangos para los parámetros que se desean optimizar en cada una de las funciones del kernel.

Función Kernel Lineal				
Costo: 0.1-50	épsilon: 0-1			
By=1	By=4			

Función Kernel Radial		
Costo: 0.1-50	épsilon: 0-1	Gamma: 0.1-5
By=1	By=0.4	By=0.4

Función Kernel Sigmoid			
Costo: 0.1-50	épsilon: 0-1	Gamma: 0.1-5	Coef0: 0.1-5
By=1	By=0.4	By=0.4	By=0.4

Función Kernel Polynomial				
Costo:0.1-50	épsilon: 0-1	Gamma: 0.1-5	Coef0: 0.1-5	Degree: 1-5
By=1	By=0.4	By=0.4	By=0.4	By=1

Figura 10. Parámetros modelo SVR-GRID.

El modelo escoge los parámetros que mejor se ajustan y prueba el algoritmo con los datos de validación o Data Test. Para programar el código del modelo SVR-GRID se tuvo en cuenta diferentes factores como ya se ha mencionado anteriormente, el programador debe elegir por decisión propia que rangos debe tener cada parámetro dependiendo del Kernel a utilizar los cuales se muestran a continuación.

En la tabla 3 se observa cada Kernel con sus mejores parámetros los cuales se eligieron por el método grid.

Tabla 3.
Resultados del modelo SVR_GRID.

KERNEL	EPSILON	COST	GAMMA	COEF0	DEGREE
LINEAL	0	1.1	N/A	N/A	N/A
RADIAL	0.4	8.1	0.1	N/A	N/A
SIGMOID	0.4	1.1	0.1	0.1	N/A
POLYNOMIAL	0	3.1	1.3	0.1	1

Nota: Parámetros obtenidos en el modelo SVR_GRID para cada función Kernel.

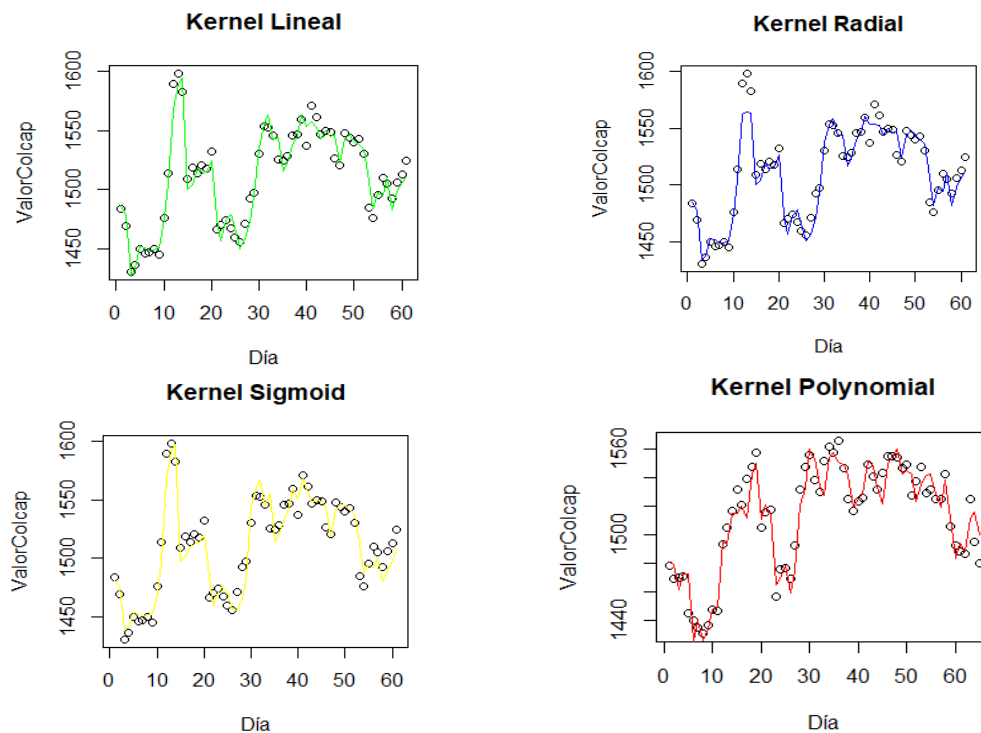


Figura 11. Resultados gráficos del Modelo SVR_GRID con cada tipo de Kernel.

La figura 11 muestra el comportamiento para cada uno de los kernels. Los círculos representan los datos reales de un conjunto de validación de la serie de tiempo, mientras que las líneas presentan la predicción para cada uno de los Kernels.

A continuación, se hace una comparación de cada uno de los Kernels con su respectivo tiempo computacional y su RMSE de modo que se pueda visualizar cuál de ellos obtuvo una mejor predicción. Como se ha mencionado anteriormente el costo computacional de elegir los parámetros en una SVR es demasiado alto en algunos casos. Como métrica de evaluación se utilizó la Raíz del Error Cuadrático Medio (RMSE), el cual consiste en la suma de las diferencias de los observados y los datos pronosticados por el modelo.

Tabla 4.
Tiempo computacional y RMSE del modelo SVR_GRID.

KERNEL	TIEMPO COMPUTACIONAL (s)	RMSE
LINEAL	19.70	9.658
RADIAL	232.28	9.949
SIGMOID	2026.20	10.389
POLYNOMIAL	348819.70	10.150

Nota: Tiempos computacionales y error cuadrático medio obtenido del modelo SVR_GRID para cada función Kernel.

A partir de la tabla 4 se puede concluir que el Kernel que tuvo una mejor predicción fue el Kernel Lineal debido a un RMSE de 9.658, con un tiempo computacional de 19.70 seg.

6.3. Resultados modelo SVR-PSO

El modelo Regresión de máquinas de soporte por Optimización por enjambre de partículas (SVR-PSO) consiste en una máquina de soporte vectorial de regresión, la cual utiliza los diferentes Kernels para hacer el pronóstico de los datos de una serie de tiempo en donde sus parámetros se escogen a través del algoritmo PSO teniendo en cuenta la velocidad y la posición de cada una de las partículas del enjambre.

Al implementar un modelo con un algoritmo PSO se deben tomar decisiones respecto a ciertos elementos que generan un correcto funcionamiento de la técnica. Una de estas decisiones es la cantidad de partículas del enjambre la cual corresponde a un número de 100, el número de iteraciones las cuales son de 100, un coeficiente inercial (w) igual a 0.9, una componente cognitiva igual a 0.5 y una componente social de 1.

Como ya se mencionó anteriormente este modelo tuvo un mejor comportamiento en su tiempo computacional utilizado donde este se vio reducido notablemente. En la siguiente tabla se muestran los resultados obtenidos por el método PSO para cada uno de los parámetros.

Tabla 5.
Resultados del modelo SVR_PSO.

KERNEL	EPSILON	COST	GAMMA	COEF0	DEGREE
LINEAL	0.525	0.441	N/A	N/A	N/A
RADIAL	0.229	0.435	0.224	N/A	N/A
SIGMOID	0.652	0.363	0.253	0.866	N/A
POLYNOMIAL	0.640	0.359	0.3148	0.806	0.988

Nota: Parámetros obtenidos en el modelo SVR_GRID para cada función Kernel.

En la tabla 6 se muestran los resultados tanto del tiempo computacional utilizado como del RMSE para cada uno de los Kernels en donde se puede observar que el mejor resultado se obtuvo con el Kernel Lineal donde se presenta un RMSE de 10.995.

Tabla 6.
Tiempo Computacional y RMSE del modelo SVR_PSO.

KERNEL	TIEMPO COMPUTACIONAL (s)	RMSE
LINEAL	2.72	10.995
RADIAL	0.67	11.024
SIGMOID	3.82	12.884
POLYNOMIAL	3.47	42.930

Nota: Tiempos computacionales y error cuadrático medio obtenido del modelo SVR_PSO para cada función Kernel.

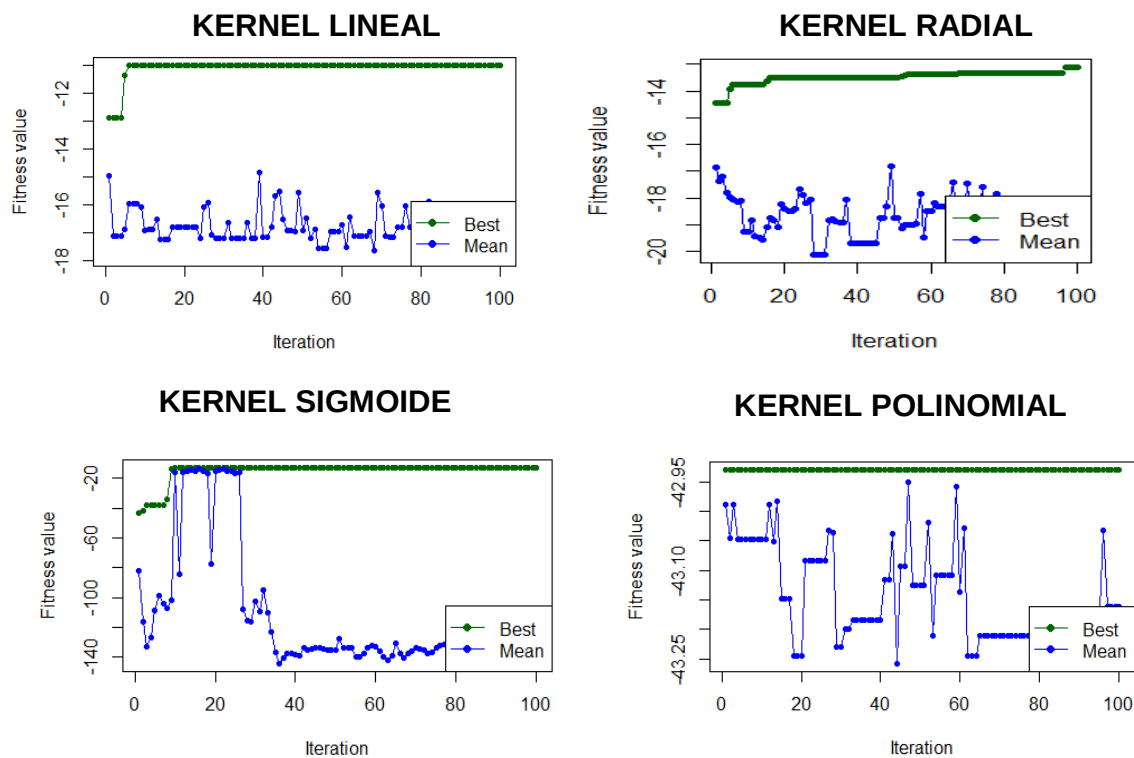


Figura 12. Resultados gráficos del Modelo SVR_PSO con cada tipo de Kernel.

La figura 12 muestra una comparación entre los datos promedio y los mejores datos de cada una de las iteraciones del valor fitness para cada uno de los Kernels en el modelo SVR-PSO.

6.4. Comparación de Resultados

En tabla 7 se hace una comparación de cada Kernel en tiempo computacional y RMSE para cada uno de los métodos utilizados. Como se puede observar la característica que más resalta es el tiempo computacional utilizado por el método SVR-PSO el cual se vio notablemente reducido, lo que hace que el modelo sea más eficaz respecto al tiempo computacional, donde los tiempos de solución son mucho más cortos en comparación con el método SVR-GRID. Por otro lado, los resultados obtenidos en el RMSE por algunos de los Kernels son muy parecidos, exceptuando por

el Kernel Polynomial cuyo RMSE se eleva repentinamente, por consiguiente, los resultados de la predicción del modelo SVR-PSO comparado con el modelo SVR-GRID son similares haciendo que los dos modelos sean igualmente eficientes.

Tabla 7.
Comparación de Resultados.

KERNEL	Tiempo computacional (s) SVR_GRID	Tiempo computacional (s) SVR_PSO	RMSE SVR_GR ID	RMSE SVR_PSO
LINEAL	19.70	2.72	9.658	10.995
RADIAL	232.28	0.67	9.949	11.024
SIGMOID	2026.20	3.82	10.389	12.884
POLYNOMIAL	348819.70	3.47	10.150	42.930

Nota: Incluye cada uno de los resultados de los modelos SVR_GRID y SVR_PSO para cada función Kernel.

Por medio de la tabla 7 se puede observar que la mejor solución al problema en cuanto a costo computacional es la solución del modelo SVR-PSO para el Kernel Lineal, en donde el resultado es de 2.72 SEG, en cuanto al error RMSE el mejor resultado se presentó para el modelo SVR-GRID en donde el resultado fue de 9.658 segundos.

En la gráfica de la figura 13 se puede observar la comparación entre los errores cuadráticos medios obtenidos de los dos modelos en cada una de las funciones kernels. En el modelo SVR-GRID las funciones Kernels que obtuvieron el menor y el mayor error fueron el Kernel Lineal y el Kernel Sigmoid respectivamente. Por el contrario, en el modelo SVR-PSO, el mayor RMSE lo registra el Kernel Polynomial, es decir, el conjunto de valores de este algoritmo posee diferencias significativas entre sí, además su resultado es mucho mayor con respecto a los valores obtenidos

en el resto de las funciones kernels utilizadas, lo que indica que estos últimos son mucho más cercanos.

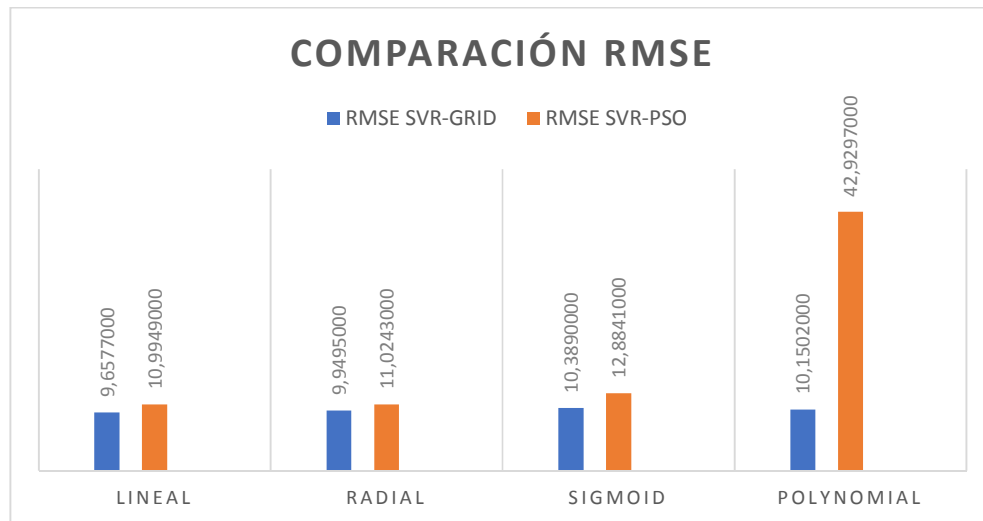


Figura 13. Resultados gráficos del error RMSE calculado en cada tipo de Kernel de los modelos SVR_GRID y SVR_PSO.

7. Conclusiones

Este estudio evalúa dos metodologías de selección de parámetros para un problema de predicción en series de tiempo- SVR-GRID, SVR-PSO- a través de un caso de estudio tomando el valor del índice COLCAP durante un año, el cual refleja las variaciones de los precios de las acciones más líquidas de la Bolsa de Valores de Colombia.

Inicialmente se realizó una revisión de la literatura acerca de los modelos de máquina de soporte vectorial existentes y las diversas aplicaciones que han tenido en los pronósticos de series de tiempo, donde se encontró que los problemas principales es el alto costo computacional que

requiere la selección de los parámetros para el control de problemas de sobreajuste o subajuste. Así mismo, se encontró que estos problemas son solucionados por medio de algoritmos evolutivos. Por esta razón también se tienen en cuenta los trabajos relacionados con los algoritmos de optimización para la selección de parámetros, en especial el algoritmo de búsqueda grid y el algoritmo de optimización por enjambre de partículas más conocido como PSO. Las investigaciones hacen énfasis en los beneficios de utilizar estos algoritmos particularmente PSO, debido a que se caracteriza por ser altamente eficiente y potente en el campo de la computación, es capaz de obtener valores bajos en el tiempo computacional utilizado, generar predicciones con un mejor desempeño comparado con modelos como ARIMA o Redes Neuronales, y presentar una mejor capacidad de generalización.

En el caso de estudio se realiza una recopilación de información necesaria para el desarrollo y construcción de los modelos en base a técnicas de aprendizaje automático, para ello se consolida una base de datos, tomando una muestra inicial de 242 valores, los cuales se dividen para entrenar y validar el modelo en otros dos subconjuntos del 70% y el 30% respectivamente; seguido a esto se construye el algoritmo SVR-GRID en el que se evidencia que los tiempos computacionales varían según el número de parámetros de cada función kernel, en este caso el tiempo del modelo kernel lineal es mucho menor que los modelos radial, polinomial y sigmoide, debido a lo anterior, se puede concluir que un algoritmo grid puede ser utilizado siempre y cuando los cálculos no sean tan complejos, puesto que el tiempo que puede tardar en correr resulta excesivo en algunos casos; además el hecho de que se tengan que fijar los rangos de los parámetros hace que la posibilidad de que se estén dejando buenas soluciones fuera de este intervalo sea mayor. Por otro lado, se construye el segundo modelo el cual es una metaheurística SVR-PSO la cual se decide dividir por

Kernels debido a que la estructura del PSO solo lo permite de esa forma. En esta se observa que los tiempos computacionales al correr cada uno de los modelos (lineal, radial, polinomial y sigmoide) no presentan mucha variación entre ellos y son significativamente bajos, incluso el menor tiempo lo registra el modelo de kernel radial el cual cuenta con un parámetro más que el modelo de kernel lineal, esto indica que la cantidad de parámetros no determina la rapidez del modelo.

Al comparar el algoritmo SVR-GRID y SVR-PSO, este último resulta ser mucho más eficiente, ya que además de la rapidez con la que arroja los resultados del modelo, no es necesario fijar unos parámetros determinados, sino que el mismo algoritmo es el encargado de realizar la búsqueda y escoger los parámetros que mejor se ajustan al modelo. Posterior a los resultados computacionales de los modelos propuestos, se puede decir que los objetivos se cumplieron a cabalidad y se espera que esta investigación sirva de base para futuros estudios relacionados con el pronóstico de series de tiempo en cualquier campo.

8. Recomendaciones

En cuanto a la revisión de literatura realizada existen diversas metodologías para la optimización de parámetros de las SVM, no obstante, queda mucho por explorar en cuanto al método de optimización por enjambre de partículas (PSO) y sus aplicaciones, debido a que la literatura en este contexto es escasa y en los resultados se puede evidenciar que en comparación otros métodos tradicionales, como es el método Grid, resulta ser más efectiva.

Finalmente, teniendo en cuenta los resultados obtenidos en el modelo SVR-PSO, se recomienda seguir con la línea de trabajo de máquinas de soporte vectorial con diferentes algoritmos de optimización para la predicción de series de tiempo, esto con el fin de obtener mejores resultados en las predicciones y el tiempo computacional utilizado.

Referencias Bibliográficas

- Alpaydin, E. (2016). Introduction to machine learning. *Computers and Operation Research*, 43, 328–334.
- Alwee, R., Shamsuddin, S. M., & Sallehuddin, R. (2017). Swarm Optimized Grey SVR and ARIMA for Modeling of Larceny-Theft Rate with Economic Indicators. *International Journal of Computational Intelligence and Applications*, 16(2), 1–12.
- Ametros, D. E. H., Henao, R., & G, J. E. H. (2004). SELECCIÓN DE HIPERPARÁMETROS EN MÁQUINAS DE SOPORTE VECTORIAL.
- Anderson, D. R. (2008). *Estadística para administración y economía*. 10a Edición.
- Arlot, S., Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4 (1), 40–79, doi: 10.1214/09-SS054
- Ataei, M., & Osanloo, M. (2012). International Journal of Surface Mining, Using a Combination of Genetic Algorithm and the Grid Search Method to Determine Optimum Cutoff Grades of Multiple Metal Deposits, (July 2013), 37–41. Recuperado de: <https://doi.org/10.1076/ijsm.18.1.60.23543>
- Basak, D., Pal, S. & Patranabis, D. C. (2007). Support Vector Regression. *Neural Information Processing – Letters and Reviews*. 11(10), 203-224. Recuperado de <https://pdfs.semanticscholar.org/c5a9/67eaded74a9fc414de4ad5120b0b66acd2c3.pdf>

Bolsa de Valores de Colombia. Índice COLCAP.

[tps://www.bvc.com.co/pps/tibco/portalbvc/Home/Mercados/enlinea/indicesbursatiles?com.tibco.ps.pagesvc.renderParams.sub45d083c1_14321f5c9c5_-78350a0a600b=action%3Ddetallar%26org.springframework.web.portlet.mvc.ImplicitModel%3Dtrue%26](https://www.bvc.com.co/pps/tibco/portalbvc/Home/Mercados/enlinea/indicesbursatiles?com.tibco.ps.pagesvc.renderParams.sub45d083c1_14321f5c9c5_-78350a0a600b=action%3Ddetallar%26org.springframework.web.portlet.mvc.ImplicitModel%3Dtrue%26).

Cao, G., & Wu, L. (2016). Support vector regression with fruit fly optimization algorithm for seasonal electricity consumption forecasting. *Energy*, 115, 734–745. Recuperado de: <https://doi.org/10.1016/j.energy.2016.09.065>

Carrizosa, E., Martín-barragán, B., & Romero, D. (2014). Computers & Operations. Research A nested heuristic for parameter tuning in Support Vector Machines. *Computers and Operation Research*, 43, 328–334. Recuperado de: <https://doi.org/10.1016/j.cor.2013.10.002>

Chang, C. C., & Lin, C. J. (2011). LIBMSV: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3), 27.

Chen, P. (2003). Particle Swarm Optimization for Power Dispatch with Pumped Hydro.

Cherkassky, V. & Ma, Y. (2002). Comparison of Model Selection for Regression. *Neural Computation*, 15(7), 1691-1714, doi: 10.1162/089976603321891864

Cherkassky, V. & Ma, Y. (2002). Selection of Meta-parameters for Support Vector Regression. *Artificial Neural Networks, LNCS*, 2415, 687-693.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273-297.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273-297.

Das, S. P., Achary, N. S., & Padhy, S. (2016). Novel hybrid SVM-TLBO forecasting model incorporating dimensionality reduction techniques. *Applied Intelligence*, 45(4), 1148–1165.

Recuperado de: <https://doi.org/10.1007/s10489-016-0801-3>

Duarte, C. Quiroga, J. (2010). Algoritmo PSO para identificación de parámetros en un motor DC.

Recuperado de: <http://www.scielo.org.co/pdf/rfiua/n55/n55a12.pdf>

Eberhart, R., & Shi, Y. (2001) Particle Swarm Optimization: Developments, Applications and Resources. *Proceedings of the 2001 Congress on Evolutionary Computation*, 81-86.

Ethem, A. (2014). *Introduction to Machine Learning*. Edit by Francis Bach, MIT Press. Gujarati,

D. N., & Porter, D. C. (2013). *Econometría. Mc Graw Hill* (Vol. 53). Recuperado de: <https://doi.org/10.1017/CBO9781107415324.004>

Fazli, S., Jafari, H. (2012). Developing a hybrid multi-criteria model for investment in stock exchange. *Management Science Letters*, 2 (2012), 457–468. Recuperado de https://www.researchgate.net/publication/271068972_Developing_a_hybrid_multi-criteria_model_for_investment_in_stock_exchange.

Google developers. Curso intensivo de aprendizaje automático. (2018). Recuperado de <https://developers.google.com/machine-learning/crash-course/framing/ml-terminology>.

Gujarati, D. N. & Porter, D. C. (2013). *Econometría*. Editorial: Mc Graw Hill.

- Hastie, T., Tibshirani, R., & Friedman, J. (2001). The Elements of Statistical Learning. *The Mathematical Intelligencer*, 27(2), 83–85. Recuperado de: <https://doi.org/10.1198/jasa.2004.s339>
- Hastie, T., Tibshirani, R., & Friedman, J. (2001). The elements of statistical learning. 2001.
- Hsu, C. W., Chang, C. C. & Lin, C. J. (2012). A Practical Guide to Support Vector Classification, Technical Report, Department of Computer Science and Information Engineering, University of National Taiwan, Taipei, 1-12.
- Hsu, C., Chang, C., & Lin, C. (2016). A Practical Guide to Support Vector Classification, 1(1), 1–16.
- HUANG, K. L. & LIAO, C. J. (2018). Ant colony optimization combined with taboo search for the job shop scheduling problem. *Computers & Operations Research*, 35(4), 1030-1046, doi: 10.1016/j.cor.2006.07.003.
- Jaramillo Garzón, J. A. (2007). Metodología de optimización de los parámetros de control de un algoritmo genético, 82.
- Jarboui, B., Siarry P., Teghem J. (Eds.). (2013). *Metaheuristics for Production Scheduling*, John Wiley & Sons. Editorial: John Wiley & Sons, Incorporated.
- Kennedy, J., & Eberhart, R. C. (2001). *Swarm intelligence*. Morgan Kaufman publishers.
- Lopez Ruiz, E., & Jaen García, M. (2005). *MODELOS ECONOMETRICOS DE SERIES TEMPORALES, TEORIA Y PRACTICA*. SEPTEN EDICIONES.

- Lujan-Moreno, G. A., Howard, P. R., Rojas, O. G., & Montgomery, D. C. (2018). Desig of experiments and response surface methodology to tune machine learning hyperparameters, with a random forest case-study. *Expert Systems with Applications*, 109, 195–205. Recuperado de: <https://doi.org/10.1016/j.eswa.2018.05.024>
- Marinakis, Y., Iordanidou, G. R., & Marinaki, M. (2013). Particle Swarm Optimization for the Vehicle Routing Problem with Stochastic Demands. *Applied Soft Computing*, 13(4), 1693-1704, ddoi: 10.1016/j.asoc.2013.01.007
- Martí, R. (2003). Procedimientos Heurísticos en Optimización Combinatoria. (Tesis Pregrado) España, Universidad de Valencia, Departamento de Estadística e Investigación Operativa, p. 60.
- Martín Guareño, J. J. (2016). Support vector regression: propiedades y aplicaciones.
- Mehryar Mohri, A. R. (2014). *Foundations of Machine Learning*. MIT Press.
- Meyer, D. (2017). Support Vector Machines. *Advanced Review*, 1(3), 283–289. Recuperado de: <https://doi.org/10.1002/wics.049>
- Mohri, M. Afshin, R. & Talwalkar, A. (2006). *Foundations of Machine Learning*.
- Mohri, M., Afshin, R., & Talwalkar, A. (2006). *IA: Inteligencia Artificial*. Red Polis.
- Mousavi, S. M., Vahdani, B., & Abdollahzade, M. (2015). An intelligent model for cost prediction in new product development projects. *Journal of Intelligent & Fuzzy Systems*, 29(5), 2047–2057. Recuperado de: <https://doi.org/10.3233/IFS-151682>

- Nickson, T., Osborne, M. A., Reece, S., & Roberts, S. (2014). Stochastic Algorithm Tuning, 1–10.
- Norena, S. (Mayo, 2018). Python Model Tuning Methods Using Cross Validation and Grid Search. Medium. Recuperado de <https://medium.com/@sebastiannorena/some-model-tuning-methods-bfef3e6544f0>.
- Olsson, A. E. (2011). *Particle swarm optimization: Theory, techniques and applications*. New York: Nova Science Publishers, Inc.
- Palma, W. (2016). *Time Series Analysis*. John Wiley & Sons, Incorporated.
- Pedroza, G., Prieto, A., & Goddard, J. (2007). Aplicación de las Maquinas de Soporte Vectorial a Reconocimiento de Hablantes. Universidad Autónoma Metropolitana, México.
- Pérez Planells, L., Delegido, J., Rivera-Caicedo, J. P., & Verrelst, J. (2015). Análisis de métodos de validación cruzada para la obtención robusta de parámetros biofísicos. *Revista Española de Teledetección*, 2015, vol. 44, p. 55-65.
- Piñero, J. M. (1972). *El análisis estadístico y sociométrico de la literatura científica*. València: Facultad de Medicina.
- Ramon López de Mántaras Badia, P. M. (2017). *Inteligencia artificial*. CSIC, Consejo Superior de Investigaciones Científicas.
- Refaeilzadeh, P., Tang, L. & Lui, H. (2008). Cross-Validation. Springer Science Business Media, doi: 10.1007/978-0-387-39940-9_565.

- Refaeilzadeh, P., Tang, L., & Lui, H. (2008). *K-fold Cross Validation*. Arizona State University.
- Ruiz L., & García, J. (2005). Modelos econométricos de series temporales, teoría y práctica. Editorial: Septem.
- Shawe, T., & Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press.
- Thomas, S., Pillai, G. N., & Pal, K. (2017). Prediction of peak ground acceleration using ϵ -SVR, ν -SVR and Ls-SVR algorithm. *Geomatics, Natural Hazards and Risk*, 8(2), 177-193.
- Vapnik V. (1995). The nature of statistical learning theory. New York (NY): Springer-Verlag.
- Vapnik, V. (1998). Statistical learning theory new york. NY: Wiley.
- Velásquez, J. D., Olaya, Y., & Franco, C. J. (2010). Predicción De Series Temporales Usando Máquinas De Vectores De Soporte. *Ingeniare. Revista Chilena de Ingeniería*, 18(1), 64–75.
Recuperado de: <https://doi.org/10.4067/S0718-33052010000100008>
- Wang, J., Hou, R., Wang, C. et ál. Improved ν -Support vector regression model based on variable selection and brain storm optimization for stock price forecasting. *Applied Soft Computing*, 49(12), 164-178, doi: 10.1016/j.asoc.2016.07.024.
- Wang, J., Hou, R., Wang, C., & Shen, L. (2016). Improved ν -Support vector regression model based on variable selection and brain storm optimization for stock price forecasting. *Applied Soft Computing Journal*, 49, 164–178.

Yang, X. S. (2010). Optimización de ingeniería: una introducción con aplicaciones. metaheurísticas. Editorial: John Wiley & Sons, Incorporated.