



POSGRADOS

Maestría en

MÉTODOS

MATEMÁTICOS Y SIMULACIÓN

NUMÉRICA EN INGENIERÍA

RPC-SO-42-NO.769-2019

Opción de Titulación:

Artículos profesionales de alto nivel

Tema:

Optimización Matemática del
Rendimiento de un Modelo de
Forecasting con Suavizamiento
Exponencial Simple

Autor

Diego Fernando Vallejo Huanga

Director:

Julio Ricardo Proaño Orellana

QUITO - Ecuador

2023



Autor:



Diego Fernando Vallejo Huanga

Ingeniero en Electrónica y Telecomunicaciones

Candidato a Magíster en Métodos Matemáticos y Simulación

Numérica en Ingeniería por la Universidad Politécnica Salesiana –

Sede Quito.

dvallejoh@ups.edu.ec

Dirigido por:



Julio Ricardo Proaño Orellana

Ingeniero en Electrónica

Doctor en Tecnologías Informáticas Avanzadas

jproanoo@ups.edu.ec

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

DERECHOS RESERVADOS

2023 © Universidad Politécnica Salesiana.

QUITO– ECUADOR – SUDAMÉRICA

DIEGO FERNANDO VALLEJO HUANGA

**OPTIMIZACIÓN MATEMÁTICA DEL RENDIMIENTO DE UN MODELO DE FORECASTING
CON SUAVIZAMIENTO EXPONENCIAL SIMPLE**

DEDICATORIA

“Cada cosa que amas es muy probable que la pierdas, pero al final, el amor volverá de una forma diferente”.

Franz Kafka

Para mi amado hijo (†)


DIEGO FERNANDO VALLEJO HUANGA


AGRADECIMIENTO

A Pauly, a mi familia y a mis amigos.

DIEGO FERNANDO VALLEJO HUANGA

Optimización Matemática del Rendimiento de un Modelo de *Forecasting* con Suavizamiento Exponencial Simple

1^{er} Diego Vallejo 
dvallejoh@ups.edu.ec

2^{do} Julio Proaño 
jproanoo@ups.edu.ec

Resumen—En *forecasting* para series de tiempo, la técnica de suavizamiento exponencial simple, permite pronosticar un estado siguiente en función de los datos pasados. Este modelo asigna automáticamente una ponderación a cada una de las instancias con una función exponencial, mediante un coeficiente de suavizamiento α , que se encuentra en el dominio de $\mathbb{R} \in [0, 1]$. La asignación de un valor adecuado de α , no es una tarea trivial al estar definido por un transfinito y, de forma genérica, se selecciona empíricamente un coeficiente que permita encontrar un balance entre complejidad computacional y error del modelo. Esta investigación científica desarrolla un proceso de optimización matemática para mejorar el rendimiento de un modelo de *forecasting* en series de tiempo con suavizamiento exponencial simple, utilizando técnicas de optimización tradicionales y modelos derivados del campo de la Inteligencia Artificial. Se utilizaron cinco técnicas de optimización y dos algoritmos de *machine learning* entrenados para poder encontrar un α optimizado. Los resultados fueron analizados con veinte conjuntos de datos, de diferente naturaleza y tamaño, donde se analizó el error del modelo mediante cuatro métricas que fueron minimizadas. Para cada técnica de optimización se analizó el tiempo de convergencia del algoritmo, con el objetivo de analizar la viabilidad de implementación en un entorno de producción. El coste computacional, en términos de tiempo de ejecución, es de 0.01 segundos para los algoritmos de aprendizaje automático en *datasets* de cualquier tamaño, mientras que un algoritmo de optimización tradicional convergió en menos de un segundo, incluso para conjuntos de datos de más de 100 000 instancias.

Palabras Clave—Pronóstico, Evaluación de Error, Rendimiento Computacional, Aprendizaje de Máquina.

Abstract—In time series forecasting, the simple exponential smoothing technique allows one to forecast the next state based on past data. This model automatically assigns a weight to each instance with an exponential function, using a smoothing coefficient α , in the domain of $\mathbb{R} \in [0, 1]$. Assigning an appropriate value of α is not a trivial task as a transfinite defines it, and generically, a coefficient is empirically selected to find a balance between computational complexity and model error. This scientific research develops a mathematical optimization process to improve the performance of a forecasting model in time series with simple exponential smoothing using traditional optimization techniques and models derived from Artificial Intelligence. Five optimization techniques and two trained machine learning algorithms were used to find an optimized α . The results were analyzed with twenty datasets of different natures and sizes, where the model error was analyzed using four metrics that were minimized. For each optimization technique, the convergence time of the algorithm was analyzed, to analyze the feasibility of implementation in a production

environment. Regarding running time, the computational cost is 0.01 seconds for machine learning algorithms on datasets of any size. In contrast, a traditional optimization algorithm converged in less than a second, even for datasets from more than 100,000 instances.

Keywords—Forecasting, Error Evaluation, Computational Performance, Machine Learning

I. INTRODUCCIÓN

El incremento de transacciones comerciales, automatización de procesos, el uso masificado de Internet y la aparición de nuevas tecnologías ha hecho que muchas organizaciones cambien el paradigma de desarrollo y despliegue de su infraestructura [1]. Muchas organizaciones han desarrollado parte de su infraestructura y comunicaciones con medios ubicuos donde la ubicación física de la organización no es relevante para el funcionamiento y operación de la industria [2]. La ingente cantidad de datos generados por los diferentes procesos industriales ha hecho que muchas empresas empiecen a utilizar técnicas de manejo y explotación de datos como una ventaja competitiva para realizar modelos predictivos [3]. Por otro lado, la Inteligencia Artificial (IA) es un campo de las Ciencias de la Computación que tiene por objetivo simular procesos, mediante algoritmos computacionales, que normalmente son desarrollados por el ser humano [4]. Uno de los grandes paradigmas enmarcados dentro de la IA, es el *Machine Learning* (ML), que es una sub-disciplina encargada de desarrollar técnicas que le permitan a un ordenador aprender, o meta-aprender, de forma autónoma y cuyo insumo principal son los datos. Estas técnicas buscan encontrar patrones característicos dentro de los datos, que a-posteriori, puedan ser extrapolados a otros sistemas y a conjuntos de datos con los que el algoritmo no fue entrenado [5]. El ML constituye una herramienta, que ayuda a reducir y eliminar el sesgo en la interpretación de la información, ya que no es manipulada directamente por las personas [6].

La información temporal de las organizaciones es almacenada en *data warehouses* y bases de datos para su posterior explotación. Las empresas, de forma genérica, poseen información temporal de los acontecimientos y una de las tareas de predicción más frecuentemente utilizadas es la de pronóstico en series de tiempo [7]. Generalmente, dentro de los métodos de pronóstico en series temporales se

suele diferenciar a los modelos tradicionales estadísticos de los modelos basados en ML. Los modelos de *forecasting* en series de tiempo, suelen ser caracterizados en el campo de la IA y son estudiados en conjunto con técnicas de ML para mejorar sus rendimientos [8].

Las series de tiempo y modelos de pronóstico en procesos estocásticos son utilizados en diversas áreas del campo empresarial como planificación estratégica de servicios de salud [9], evaluación de rendimiento en servicios hospitalarios de emergencia [10], pronósticos de demanda en productos farmacéuticos [11] [12], planificación de servicios de hotelería [13] [14] y turismo [15] [16], optimización de precios de agroproductos alimenticios [17] [18], predicción de precios en tiendas minoristas [19], pronósticos de demanda energética [20] [21], optimización de diferentes cadenas de suministro [22] [23] [24] [25], etc.

Uno de los métodos más utilizados en pronóstico en series de tiempo es el de Suavizamiento Exponencial Simple (SES), propuesto a finales de los años 1950 [26] [27]. Su uso es frecuente, en diversas disciplinas y con diversos objetivos, por la facilidad de parametrización, la rapidez en la generación de pronósticos y por su confiabilidad [28]. Los modelos de SES pueden tener aplicaciones en muchos campos como finanzas, pronósticos del tiempo, control de inventario, etc. Uno de los inconvenientes más frecuentes en este método es la selección adecuada del coeficiente de suavizamiento, α , que permite ponderar de forma exponencial a las observaciones de la serie temporal y que finalmente determina el rendimiento del pronóstico [29]. Entonces, es importante encontrar un método que permita optimizar matemáticamente el rendimiento de un modelo de *forecasting* en series de tiempo con suavizamiento exponencial simple, para mejorar los resultados de pronóstico y, que posteriormente el modelo, pueda ser utilizado en procesos industriales o empresariales.

Por otro lado, el teorema de optimización *No Free Lunch* (NFL) [30] establece que, en promedio, ningún algoritmo de búsqueda o aprendizaje universal es mejor que cualquier otro algoritmo para resolver todos los problemas de manera general, i.e. que es imposible teóricamente caracterizar una estrategia de optimización universal, de propósito general, para cualquier conjunto de datos o problema [31]. Por lo tanto, la elección de un algoritmo de optimización depende en gran medida del problema específico que se está tratando de resolver y de sus características, como el tamaño del conjunto de datos, la complejidad del modelo, la dimensionalidad de los datos, la naturaleza del *dataset*, entre otros factores [32]. Este teorema tiene implicaciones importantes en el campo de la optimización y el ML, y ha influido en la comprensión de las limitaciones de los algoritmos y enfoques de búsqueda. Entonces, es una tarea esencial y crítica explorar, en diferentes áreas científicas, un espacio de búsqueda que caracterice una solución óptima para una familia de problemas utilizando procesos algorítmicos y de simulación, dado que no existe un modelo de *forecasting* que tenga un mejor rendimiento sobre todos los *datasets* en series de tiempo [33].

A. Trabajos Relacionados

Dado que el rendimiento para el método de pronóstico SES depende de la adecuada selección del coeficiente α , entonces, algunos trabajos previos han intentado caracterizar este parámetro. En [29] se proporciona una guía para determinar el tipo de modelo de suavizamiento exponencial y α óptimo. Para elegir el modelo adecuado se deben conocer las características de tendencia y estacionalidad de la serie temporal. El autor determina un intervalo de confianza que contiene la tendencia estocástica y determinística. La elección entre los dos tipos depende de cada serie de tiempo de forma individual. El artículo no acota la familia de problemas para los cuales es posible tener una solución óptima.

Para Stevenson [34] seleccionar la constante de suavizamiento es una cuestión de prueba y error, donde este parámetro del modelo debe ser ajustado manualmente hasta que la medida de bondad de ajuste sea maximizada. Además el autor expresa que el principal objetivo es equilibrar las variaciones aleatorias y responder a los cambios reales de un conjunto de datos y escenario específico. Los valores de α comúnmente utilizados oscilan entre 0.05 y 0.50, siendo utilizados los valores bajos cuando los promedios subyacentes son estables y valores altos cuando son susceptibles a cambiar. Aunque varios autores recomiendan seleccionar valores entre $0 \leq \alpha \leq 0.5$ Gardner [26] afirma que los rangos de los parámetros y valores iniciales no deberían ser arbitrarios porque esto puede restar precisión al modelo de pronóstico.

Berry et al. [35], propusieron un algoritmo denominado de búsqueda de patrones para evaluar dos parámetros, el primero relacionado con el error del modelo y el segundo con el tiempo computacional necesario para llegar a la solución, cuando se intenta caracterizar un valor del coeficiente de suavizamiento exponencial simple que minimice los errores, realizando aproximaciones hacia el valor de convergencia.

Otra posible solución para encontrar el valor óptimo, del coeficiente de suavizamiento α , es explorar de forma exhaustiva, i.e., probando valores dentro del rango de definición $\alpha \in [0, 1]$ [36]. Así, en [37] se realizan saltos de 0.1 y se exploran todas las posibilidades para determinar el valor de α más apropiado. Esta forma trivial de encontrar la solución óptima es ineficiente computacionalmente si se desea explorar una solución con una alta tolerancia porque el espacio de soluciones crece. Esta búsqueda exhaustiva, también denominada búsqueda en cuadrícula, fue probada por Dielman [38] con saltos de 0.01, dentro del rango de operación de α para minimizar el *Mean Absolute Percentage Error* (MAPE), *Root Mean Square Error* (RMSE) y *Mean Absolute Deviation* (MAD).

Existen algunos trabajos relacionados que han intentado previamente realizar la optimización de coeficiente de suavizamiento α . En [39] se propone la optimización del coeficiente de suavizamiento exponencial mediante la optimización de los parámetros de rendimiento *Mean Square*

Error (MSE) y MAD, mediante un método de optimización no lineal. Las simulaciones fueron hechas con el software Microsoft Solver, usando el registro de datos de seis años de una empresa de bebidas y alimentos en Bangladesh. Los resultados mostraron que la técnica propuesta alcanzaba un mejor rendimiento que un método empírico de selección al azar de α .

En el artículo de Ravinder et al. también se evalúa la factibilidad y validez de implementación de modelos no lineales de optimización para la selección adecuada de α [40]. Entonces, si los datos no tienen tendencia se minimiza el MSE utilizando un valor pequeño para α o incluso 0, mientras que si se desean alinear los pronósticos con datos reales, se necesitarán valores más grandes de α . Los autores concluyen que Microsoft Solver, en su modo predeterminado usando el motor no lineal, podría proporcionar un suavizado óptimo incorrecto cuando hay un óptimo local cerca de la solución inicial, por lo tanto, este método tiene ciertas limitaciones en su despliegue. El reciente artículo de Junthopas et al. [41] también utiliza Microsoft Solver para poder determinar un valor óptimo de α , mediante la configuración de un valor inicial por ponderación. La nueva propuesta es comparada con dos enfoques tradicionales y el error es medido mediante el MSE. El estudio concluye que para valores de α inicializados en el rango de 0.1 a 0.4 los resultados no son los más adecuados, mientras que cuando el valor es mayor que 0.5 el método propuesto es más efectivo.

En el trabajo de [42], se propone la optimización de los coeficientes de suavizamiento exponencial triple en un caso de estudio específico para mejorar el rendimiento del MAPE. Los resultados de la investigación muestran que la selección de forma estratificada mejora el proceso de pronóstico y por lo tanto, la empresa de Indonesia, donde se despliega y valida el modelo, obtiene una ventaja competitiva. Además se pone de manifiesto que una simple técnica de inicialización correcta, permite una rápida convergencia de los algoritmos de optimización y permite disminuir el coste computacional y la latencia.

Algunos algoritmos derivados del campo de IA también han sido utilizados como técnicas de optimización para procesos en series temporales. En problemas de optimización combinatoria los algoritmos metaheurísticos permiten obtener soluciones a problemas de búsqueda mediante la definición de una función genérica y abstracta del problema. Estos algoritmos pueden ayudar a encontrar el valor óptimo del coeficiente de suavizamiento exponencial que minimiza el error de pronóstico o maximiza la precisión del modelo. Así, Swari et al. [43] utilizaron la metaheurística de Optimización por Enjambre de Partículas (PSO) [44] para optimizar el pronóstico de ventas diarias de productos en una serie temporal. El trabajo utiliza un solo conjunto de datos para la experimentación y las métricas MAPE y RMSE para medir el rendimiento de los resultados. Además, se comparó el método propuesto con SES sin optimización y con el método de *forecasting* Promedio Móvil Simple (SMA). En un alcance al trabajo anterior en [45] se utiliza una mod-

ificación del algoritmo PSO, denominado MPSO, para la optimización del coeficiente de suavizamiento exponencial, α , en el mismo modelo de pronóstico.

Chusyairi et al. [46] desarrollaron una aproximación para optimizar el valor de α en el método de suavizamiento exponencial simple, doble y triple, mediante el uso de un algoritmo genético [47]. Esta propuesta es aplicada en un *dataset* de 22850 instancias, durante 16 meses y se utilizan las medidas de error MAD, MSE y MAPE, para determinar el rendimiento del pronóstico. Algunas otras aproximaciones similares, al trabajo de Chusyairi et al., se han realizado en series temporales con el uso de algoritmos genéticos y variaciones algorítmicas [48] [49] [50].

Entonces, este trabajo de investigación tiene por objetivo optimizar matemáticamente el rendimiento del modelo de *forecasting* de suavizamiento exponencial simple. Para este efecto se probarán diferentes técnicas y algoritmos de optimización y se medirá el rendimiento de cada propuesta. Las soluciones serán probadas con distintos conjuntos de datos de *benchmarking* con la finalidad de observar los resultados cuando se varía la naturaleza de los datos y la cantidad de instancias involucradas en el proceso de pronóstico. Ya que existen diversas métricas para la evaluación de rendimiento del modelo, esta investigación utilizará únicamente aquellas que se derivan del error de pronóstico, y que la literatura científica relacionada, determina como las más importantes. En todos los experimentos se medirá el tiempo computacional de ejecución de los algoritmos, para analizar la viabilidad de implementación en un prototipo de software con las soluciones óptimas. La contribución de esta propuesta de investigación se direcciona en brindar a los usuarios de esta técnica de pronóstico un formato simplificado del modelo, donde los parámetros de convergencia no sean determinados de forma randómica y que puedan ser consumidos desde una aplicación web de software libre.

II. MATERIALES Y MÉTODOS

A. Descripción de la Metodología de Desarrollo

Esta investigación es de naturaleza científico descriptiva, ya que busca identificar patrones y tendencias en los datos, con el objetivo de analizar, comparar y medir magnitudes que nos permitan generar inferencias y pronósticos optimizados. Por la naturaleza de la información es de tipo cuantitativa ya que se manejan valores numéricos tanto en las entradas de información que corresponden a los conjuntos de datos históricos de las series temporales, como en sus salidas determinados por los valores pronosticados. El diseño de la investigación es de tipo experimental porque se manipulará deliberadamente la variable coeficiente de suavizamiento α , con el objetivo de optimizar el modelo de pronóstico. El nivel de la investigación es práctica, pues se aplican diversos conocimientos enfocados a la optimización de los parámetros de pronóstico.

En el procesamiento de datos se utilizó la metodología *Knowledge Discovery in Databases* (KDD) [51], que utiliza técnicas de minería de datos integradas con modelos de

inteligencia artificial y estadística para descubrir patrones dentro de las bases de datos. Esta metodología está adaptada para un modelo de pronóstico mediante la selección del conjunto de datos, análisis, limpieza y procesamiento del *dataset*.

Para el desarrollo del proceso de optimización, se ha planificado cuatro etapas. En la primera etapa se busca realizar una recolección de conjuntos de datos, en series temporales, con ciertas características intrínsecas que permitan aplicar el método de suavizamiento exponencial simple. Para un posterior análisis, se extraen las medidas estadísticas más relevantes de cada conjunto de datos, con el objetivo de realizar ciertas pruebas de hipótesis que permitan determinar el cumplimiento de los supuestos que se establecen en el modelado de *forecasting*, para datos espaciados de forma equidistante en el tiempo.

En la segunda etapa se diseñan los algoritmos matemáticos que permitan optimizar la solución del modelo de pronóstico. Para el desarrollo de esta investigación se utilizarán cinco diferentes técnicas para la optimización del parámetro α : un algoritmo de fuerza bruta, un algoritmo basado en aproximaciones numéricas, uno de búsqueda aleatoria, un algoritmo genético y un algoritmo metaheurístico de optimización por enjambre de partículas. Para validar los resultados se utilizarán métricas de evaluación de rendimiento, propias de los modelos de *forecasting*, así como los parámetros que definen la latencia de los algoritmos y su rendimiento a nivel computacional. Esto último es importante, porque la optimización del parámetro α se encuentra dentro del rango de un transfinito donde la solución puede ser óptima pero con un tiempo de respuesta no viable, lo que imposibilitaría la implementación dentro de un prototipo funcional.

En una tercera etapa, con las características comunes extraídas de los conjuntos de datos y los resultados de los valores de α optimizados, se plantea el uso de un algoritmo de ML para generar un modelo que permita definir un valor aproximado del coeficiente de suavizamiento, únicamente, con las características propias de la serie temporal. Se realiza validación cruzada para la evaluación de este modelo de ML. Finalmente, se plantea una etapa de prototipado para el despliegue de los resultados de la investigación en un software que pueda ser consumido por personas externas a los *stakeholders* de esta investigación.

El resumen de la metodología planteada en este trabajo de investigación, que comprende la etapa de caracterización, limpieza y selección del conjunto de datos, aplicación de los algoritmos de optimización y despliegue web de los resultados, se muestra en el diagrama de bloques de la Figura 1.

B. Funcionamiento del Modelo de Forecasting SES

El modelo de *forecasting* con suavizamiento exponencial simple, permite pronosticar un siguiente estado F_{t+1} , en función de los datos pasados de una serie de datos temporales. En este tipo de modelo interviene un conjunto de n observaciones, diseminadas en un lapso de tiempo t , que

asigna automáticamente un peso a cada una de las instancias con una función exponencial en el tiempo. Para la asignación automática de las ponderaciones en las observaciones, se utiliza un coeficiente α de suavizamiento que se encuentra en el dominio de $\mathbb{R} \rightarrow [0, 1]$. Así, el valor del pronóstico F_{t+1} depende de: el valor actual para el período t de la serie de tiempo Y_t , el pronóstico F_t para el período t de la serie de tiempo y del valor de α , como se muestra en la Ecuación 1. Siendo α el único parámetro que se puede modificar para mejorar el rendimiento del pronóstico en función del error del modelo, ya que este controla la influencia con la que las observaciones pasadas disminuyen o aumentan exponencialmente [26].

$$F_{t+1} = \alpha Y_t + (1 - \alpha) F_t \quad (1)$$

El dominio del valor de α está definido por un transfinito [52]. En teoría de conjuntos, un número transfinito es el término que el matemático alemán Georg Cantor introdujo para referirse a los ordinales infinitos, que son mayores que cualquier número natural. Entonces, el problema se vuelve complejo ya que existe un infinito número de valores para seleccionar en el rango de $[0, 1]$, pero un solo valor acotado de α que permitirá optimizar la solución.

Ahora, el objetivo de un método de pronóstico en series de tiempo con suavizamiento exponencial es encontrar el mejor pronóstico para un tiempo $t+1$, con un valor de α optimizado mediante la evaluación de las métricas de rendimiento.

Desde la Ecuación 1, se puede obtener una expresión que permita tener el error ε_t del modelo, considerando la diferencia entre el valor real Y_t de la serie de tiempo y el valor pronosticado F_t por el método SES, es decir que $\varepsilon_t = Y_t - F_t$, como se muestra en la Ecuación 2.

$$\begin{aligned} F_{t+1} &= \alpha Y_t + F_t - \alpha F_t \\ &= (\alpha Y_t - \alpha F_t) + F_t \\ &= \alpha(Y_t - F_t) + F_t \\ &= \alpha(\varepsilon_t) + F_t \end{aligned} \quad (2)$$

Así, el error de la Ecuación 2 para el modelo de pronóstico es una función $\varepsilon(F_{t+1}, F_t, \alpha)$ con relación directamente proporcional para la diferencia entre el valor pronosticado siguiente y actual, y una relación inversamente proporcional con el coeficiente de suavizamiento exponencial simple (Ecuación 3).

$$\varepsilon_t = \frac{F_{t+1} - F_t}{\alpha} \quad (3)$$

C. Descripción de los Conjuntos de Datos

Este proyecto de investigación utilizará *datasets*, de diferentes disciplinas, alojados en repositorios de datos de acceso abierto y liberados bajo licencias que permitan su uso y modificación de forma amplia. Por lo tanto, la propuesta no incluye una etapa de recolección de datos e información histórica de organizaciones y procesos, para la validación de los modelos de optimización.

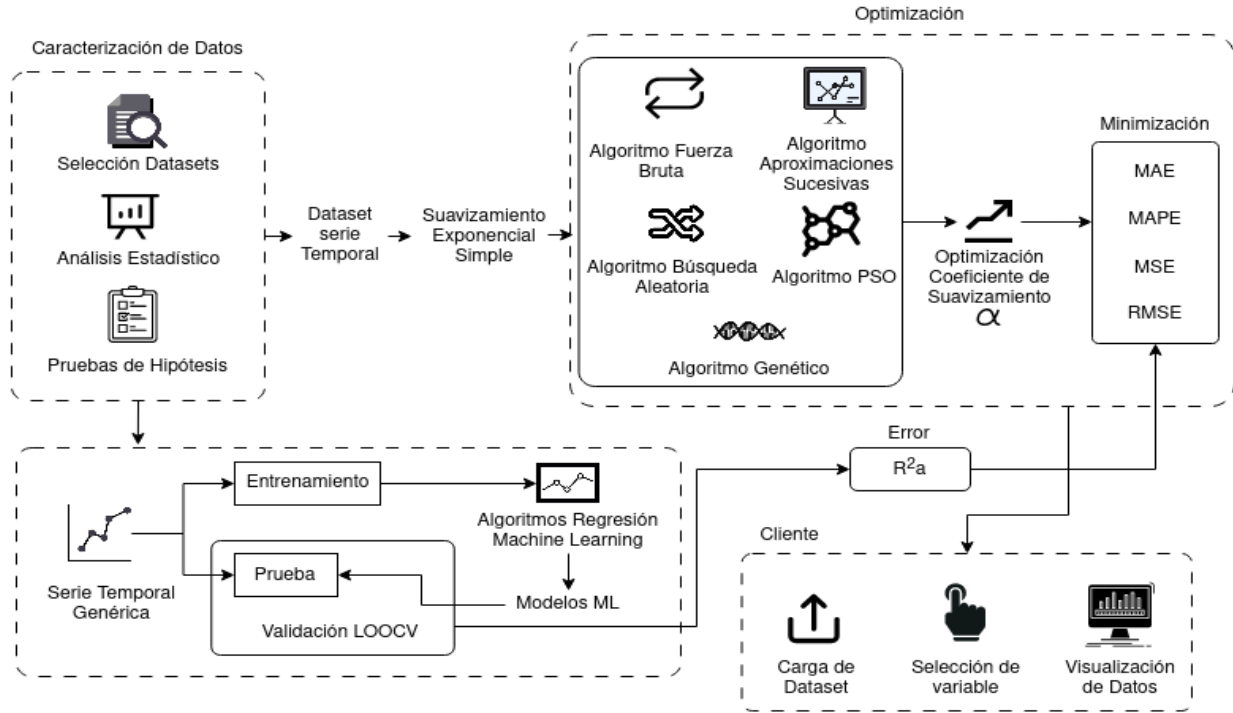


Fig. 1: Esquema metodológico utilizado para caracterizar a los *datasets* de series temporales con el método SES y optimizar el coeficiente de suavizamiento

Para formalizar la nomenclatura matemática se utilizará la notación de teoría de conjuntos. Considerando una colección de conjuntos de datos D_k para $k = \{1, \dots, m\}$, donde cada *dataset* tiene n instancias y p variables, que puede ser representado como un conjunto de pares ordenados. Cada par consiste en una tupla de dos elementos correspondientes a los valores de las variables en una instancia específica. Así, cada instancia tiene las variables representadas como $\{x_1, x_2, \dots, x_p\}$. Entonces, es posible denotar a cada conjunto de datos con la Ecuación 4

$$D = \{(x_{11}, x_{12}, \dots, x_{1p}), (x_{21}, x_{22}, \dots, x_{2p}), \dots, (x_{n1}, x_{n2}, \dots, x_{np})\} \quad (4)$$

Donde $(x_{i1}, x_{i2}, \dots, x_{ip})$ representa la i -ésima instancia del dataset con $i = \{1, \dots, n\}$ y cada x_{ij} es el valor de la j -ésima variable en la i -ésima instancia con $j = \{1, \dots, p\}$.

Aunque los *datasets* de series temporales, utilizados para modelar el proceso de *forecasting* con SES, pueden tener p variables representadas en un espacio \mathbb{R}^p , se seleccionará únicamente a aquella variable para la cual se ha decidido ejecutar el proceso de pronóstico. Esta variable x_j , específica para cada conjunto de datos, representa los valores actuales Y_t en el tiempo, que el modelo SES recibirá como entrada, i.e. que $x_j = Y_t$. Entonces, los valores de las n instancias, para la variable Y_t a pronosticar, representan el número de observaciones que determinan la longitud de la serie de tiempo $(Y_t)_{t=1}^n$ y se pueden expresar como $(Y_t : t = 1, \dots, n)$. Para el procesamiento de la información,

a-posteriori, las n instancias temporales $\{Y_1, Y_2, \dots, Y_n\}$, se almacenarán en un vector columna $Y \equiv [Y_1, Y_2, \dots, Y_n]'$ de orden $n \times 1$.

Los conjuntos de datos, para el posterior proceso de experimentación, se obtuvieron de repositorios públicos como el de la Universidad de California Irvine (UCI) para tareas de ML [53], la plataforma de concursos de tareas de aprendizaje automático Kaggle [54], catálogos de datos abiertos gubernamentales como Data Gov [55], etc.

En forma genérica, los *datasets* pueden ser clasificados, de acuerdo a su tamaño, en pequeños, medianos y grandes [56]. Los conjuntos de datos pequeños generalmente contienen un número relativamente reducido de instancias. No existe un umbral estricto para definir lo que constituye un conjunto de datos pequeño, pero frecuentemente se considera que tienen entre 1000 a 10000 instancias. Los *datasets* medianos suelen contener de 10000 a 100000 objetos, mientras que los grandes conjuntos de datos poseen millones o miles de millones de instancias [57]. Los conjuntos de datos utilizados para esta investigación se distribuyen desde el orden de las decenas hasta los cientos de miles de instancias, es decir que pueden ser clasificados como *datasets* de pequeño y mediano tamaño. Esta clasificación de tamaño en los conjuntos de datos es aproximada y no existe una definición universalmente aceptada para cada categoría, ya que los límites entre los tamaños de los conjuntos de datos pueden variar según el contexto y el área de conocimiento. Además, por la gran cantidad de datos generados actualmente, el aumento en la capacidad de cómputo y la mejora en las

D_k	n	x_p	T_u	R_k	$\mu \pm \sigma$	Área de Conocimiento	Subárea de Conocimiento	Subárea Específica de Conocimiento
D_{01}	25	x_2	día	[38, 100]	62.64 ± 14.35	TIC	Información y Comunicación	Uso del ordenador
D_{02}	30	x_4	año	[825, 946]	873.57 ± 38.4	Administración	Negocios y Administración	Finanzas, banca y seguros
D_{03}	36	x_2	mes	[119.3, 682.0]	312.6 ± 148.94	Administración	Negocios y Administración	Ventas al por mayor y menor
D_{04}	60	x_{13}	semana	[129.41, 616.45]	300.87 ± 89.6	Servicios	Servicios de Transporte	Servicios de transporte
D_{05}	61	x_2	mes	[2.1, 5.4]	3.46 ± 1.05	Administración	Negocios y Administración	Finanzas, banca y seguros
D_{06}	139	x_5	mes	[373, 724]	534.99 ± 73.13	Salud y bienestar	Salud	Medicina
D_{07}	222	x_5	año	[10, 2000]	268.7 ± 187.8	Administración	Negocios y Administración	Ventas al por mayor y menor
D_{08}	248	x_{13}	día	[13196, 408583]	92675.02 ± 50541.61	Administración	Negocios y Administración	Finanzas, banca y seguros
D_{09}	476	x_2	mes	[64.8, 217.8]	136.4 ± 33.74	Ingeniería, industria y construcción	Fabricación y procesos	Procesamiento de alimentos
D_{10}	630	x_5	año	[1.01, 9931]	931.58 ± 1942.85	Administración	Negocios y Administración	Finanzas, banca y seguros
D_{11}	1000	x_{10}	día	[5743, 402201]	53100.5 ± 35329.83	Administración	Negocios y Administración	Finanzas, banca y seguros
D_{12}	2072	x_6	día	[0, 106920]	13741.03 ± 11487.03	Administración	Negocios y Administración	Finanzas, banca y seguros
D_{13}	3009	x_3	año	[1.65, 8069.85]	2602.86 ± 1913.07	Servicios	Servicios Personales	Deportes
D_{14}	3322	x_{12}	día	[2.42e+11, 8.16e+15]	$1.07e+14 \pm 2.63e+14$	Administración	Negocios y Administración	Finanzas, banca y seguros
D_{15}	2745	x_5	año	[4.7, 51.7]	20.64 ± 6.88	Salud y bienestar	Salud	Medicina
D_{16}	10468	x_7	día	[0, 7421640800]	330848870 ± 338841843	Administración	Negocios y Administración	Finanzas, banca y seguros
D_{17}	25101	x_2	día	[0.04, 22.36]	4.6 ± 3.63	Administración	Negocios y Administración	Finanzas, banca y seguros
D_{18}	142524	x_3	2 min	[20000, 1262819994]	12347275 ± 30654858	Administración	Negocios y Administración	Ventas al por mayor y menor
D_{19}	155119	x_{16}	día	[0, 5413]	77.31 ± 237.10	Servicios	Servicios de transporte	Servicios de Transporte
D_{20}	197161	x_6	30 min	[6, 2000]	94.16 ± 82.38	Administración	Negocios y Administración	Ventas al por mayor y menor

TABLA I: Resumen estadístico de los conjuntos de datos en series temporales, utilizados para la experimentación, desgregados por área de conocimiento

técnicas de minería y explotación de datos, es posible que estos rangos referenciales se modifiquen continuamente.

Para facilitar la presentación y tratamiento de los datos, en esta investigación, se realizó una taxonomía empírica interna, para formar cuatro grupos de acuerdo al número de instancias. Así, cada grupo ($G_l : l = 1, \dots, 4$) tiene una cardinalidad $|G_l| = 5$, i.e. que existen cinco D_k por cada grupo G_l , de diferentes disciplinas científicas. La Ecuación 5 muestra los rangos de definición de cada G_l .

$$G_l \begin{cases} G_1 & D_k \rightarrow [1, 100) \\ G_2 & D_k \rightarrow [100, 1000) \\ G_3 & D_k \rightarrow [1000, 10000) \\ G_4 & D_k \rightarrow [10000, +\infty) \end{cases} \quad (5)$$

Otra consideración importante en la selección de los conjunto de datos, para el proceso de pronóstico, fue que los valores de α no se acerquen a las fronteras de definición del rango, por las implicaciones expuestas en las Observaciones 1 y 2. Este par de observaciones, para efectos prácticos de experimentación, redefinen a $\alpha \rightarrow \mathbb{R} \in (0, 1)$.

Observación 1: El valor mínimo 0 de α , no es considerado porque para este caso el pronóstico F_{t+1} sería igual al pronóstico F_t para el período t , i.e., no existiría una actualización del pronóstico a-posteriori.

$$\begin{aligned} F_{t+1} &= \alpha Y_t + (1 - \alpha)F_t \\ &= (0)Y_t + (1 - 0)F_t \\ &= F_t \end{aligned} \quad (6)$$

Observación 2: El valor máximo 1 de α , es descartado porque para este caso el pronóstico F_{t+1} sería igual al valor real para el período t de la serie de tiempo Y_t , i.e., no existiría pronóstico.

$$\begin{aligned} F_{t+1} &= \alpha Y_t + (1 - \alpha)F_t \\ &= (1)Y_t + (1 - 1)F_t \\ &= Y_t \end{aligned} \quad (7)$$

Es menester acotar que los D_k tienen diferentes unidades de tiempo, T_u , e.g. minutos, días, semanas, meses o años, dependiendo de la naturaleza intrínseca del origen de los datos y del proceso de muestreo llevado a cabo en cada área de conocimiento. Los *datasets* seleccionados corresponden a algunas áreas de conocimiento catalogadas dentro de la clasificación internacional para los campos de ciencia y tecnología establecida por la UNESCO [58]. La Tabla 1 resume la nomenclatura D_k definida para los veinte *datasets* seleccionados, la n cantidad de instancias contenidas en cada conjunto de datos, la variable x_p que corresponde a la columna de la variable Y_t en el *dataset* original. Además del área de conocimiento de cada D_k , se detalla la subárea de conocimiento y la subárea específica de conocimiento. El valor R_k corresponde al rango de valores que tiene cada Y_t junto con su media μ y desviación estándar σ .

La Figura 2 muestra los diagramas de caja y bigotes de todos los conjuntos de datos que se utilizarán en esta investigación. Dado que los R_k de cada *dataset* difieren entre si y existe una gran variabilidad para los veinte conjuntos de datos, entonces, se realizó un proceso de normalización min-max [59] [60] para escalar a cada una de las instancias temporales dentro del dominio de $\mathbb{R} \in [0, 1]$, con el objetivo de comparar la variabilidad y medidas de tendencia central entre cada D_k en una misma escala.

D. Métricas de Evaluación de Rendimiento del Método SES

Las métricas para evaluar el rendimiento de un modelo de *forecasting* son medidas de bondad de ajuste calculadas en función del error acumulado en el pronóstico. Para medir el rendimiento del modelo de serie de tiempo se utilizan diferentes métricas que permiten evaluar el comportamiento del modelo, y por ende, del valor de α seleccionado. Estas métricas, de forma general, usan el error como medida de rendimiento. Una selección correcta de α , permite minimizar el error que determina la precisión del modelo de pronóstico en series temporales. En esta investigación se han considerado cuatro métricas para la evaluación del rendimiento. Para todas las medidas de error son deseables valores cercanos a cero ya que esto implicaría mejores

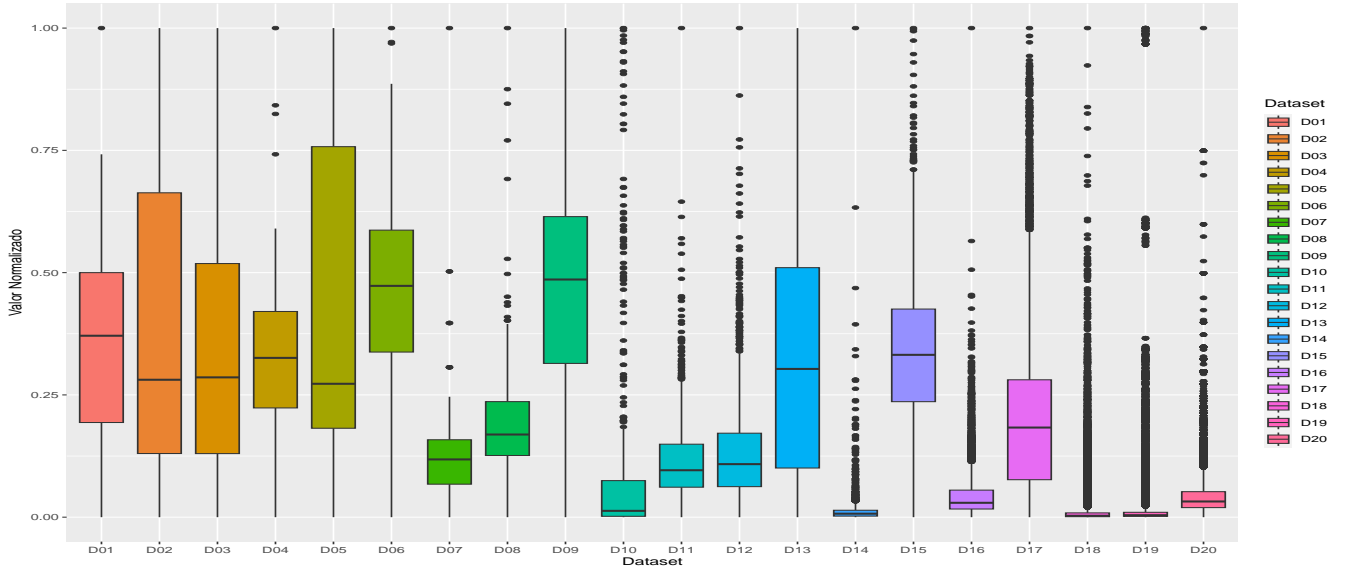


Fig. 2: Diagramas de caja y bigotes con normalización min-max de todos los conjuntos de datos

resultados en el pronóstico.

1) *Mean Square Error (MSE)*: El error cuadrático medio representa la media aritmética de los errores calculados en cada instancia de la serie de tiempo, elevados al cuadrado. El cuadrado de los errores tiende a incrementar el peso de los valores atípicos, afectando la precisión de los resultados [61]. Su representación matemática está descrita en la Ecuación 8:

$$MSE = \frac{1}{n} \sum_{t=1}^n (\varepsilon_t)^2 \quad (8)$$

Esta métrica se expresa en términos de unidades cuadráticas de la variable Y_t a predecir.

2) *Root Mean Square Error (RMSE)*: Esta métrica está determinada por la raíz del *MSE* y suele ser calculada para tener una mejor interpretabilidad de los resultados del error, dado que se expresa en las mismas unidades de la variable de pronóstico Y_t [62], como se muestra en la Ecuación 9:

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (\varepsilon_t)^2} \quad (9)$$

3) *Mean Absolute Error (MAE)*: El error absoluto medio mide qué tan cercanas están las observaciones pronosticadas a los resultados obtenidos en unidades absolutas [61], como se define en la Ecuación 10:

$$MAE = \frac{1}{n} \sum_{t=1}^n |\varepsilon_t| \quad (10)$$

4) *Mean Absolute Percentage Error (MAPE)*: El error porcentual absoluto medio se obtiene calculando el error

absoluto en cada período dividido por los valores observados y luego se promedian los porcentajes fijos [63].

$$MAPE = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{\varepsilon_t}{Y_t} \right| \quad (11)$$

E. Pruebas de Convexidad de los conjuntos de Datos

Dado que los algoritmos planteados en este proyecto de investigación buscan un valor $\alpha \rightarrow \mathbb{R} \in (0, 1)$ que optimice el error de pronóstico, es necesario analizar si los conjuntos de datos D_k , en su espacio de soluciones, cumplen con la propiedad de convexidad [64].

Es importante que el espacio de soluciones en un problema de optimización sea convexo por varias razones: i) unicidad del mínimo global, ya que en un espacio convexo existe un único mínimo global, i.e., que hay solo una solución globalmente óptima sin mínimos locales que puedan atrapar al algoritmo de optimización en soluciones subóptimas; ii) convergencia garantizada para los algoritmos de optimización hacia el mínimo global de la función objetivo, para obtener soluciones óptimas confiables y eficientes; iii) soluciones estables que permiten que pequeñas perturbaciones en los datos, o en los parámetros del problema, no afecten drásticamente la solución óptima; iv) facilidad en la formulación del problemas, dado que la convexidad del espacio de soluciones permite una formulación matemática más simple del problema de optimización [65].

Para la demostración de convexidad del espacio de soluciones, se utilizará la definición de convexidad [66], considerando que un espacio convexo se puede representar como la intersección de hiperplanos. Entonces, al variar los valores α el error ε_t del modelo SES cambia y se busca:

$$\min_{\substack{\forall \varepsilon_t \in \mathbb{R} \\ \alpha \in (0,1)}} \varepsilon_t(\alpha) \quad (12)$$

Entonces, la función objetivo $\varepsilon_t(\alpha) = f$ está definida en el espacio de soluciones, donde α se puede representar como un vector de datos $(\alpha_q : q = 1, \dots, N - 1)$ donde $\alpha_{q+1} - \alpha_q = \delta$, siendo δ un valor constante que permite tener variaciones equidistantes de α y N representa el número de subintervalos de ancho δ contenidos en el rango de α . Si se considera un punto del espacio con coordenadas $(\alpha_q, \varepsilon_t(\alpha_q))$ y se utiliza la definición de convexidad, entonces, la combinación lineal de α_q y $\varepsilon_t(\alpha_q)$ se puede expresar como $\beta\alpha_q + (1-\beta)\varepsilon_t(\alpha_q)$, $\forall \beta \in [0, 1]$. Ahora, aplicando la definición de convexidad a la función f , si f es convexa, entonces se cumple la desigualdad convexa:

$$f(\beta\alpha_q + (1-\beta)\varepsilon_t(\alpha_q)) \leq \beta f(\alpha_q) + (1-\beta)f(\varepsilon_t(\alpha_q)) \quad (13)$$

Luego, para que el espacio de solución sea convexo para todos los puntos $(\alpha_q, \varepsilon_t(\alpha_q))$, se debe cumplir que $\beta\alpha_q + (1-\beta)\varepsilon_t(\alpha_q)$ también debe pertenecer al espacio de soluciones.

Así, para los pares de datos formados entre V_α y ε_t , se considera que el espacio es convexo si, para cualquier par de puntos en el conjunto, la línea recta que los une se encuentra completamente dentro del conjunto. Para realizar este proceso de verificación de convexidad se utilizó el *convex hull* [67], o envolvente convexa, que en geometría computacional se refiere al menor conjunto convexo que contiene un conjunto dado de puntos en un espacio euclidiano y que permite observar la convexidad del espacio de definición para cada uno de los veinte conjuntos de datos D_k , en series temporales, seleccionados para la experimentación de esta investigación.

F. Pruebas de Hipótesis

El objetivo de esta investigación no es verificar que el método de pronóstico SES sea la técnica de *forecasting* más adecuada para un conjunto de datos específico, es decir que, es posible que exista alguna técnica de pronóstico que obtenga mejores resultados, considerando el error acumulado en el proceso estocástico. Existen varias técnicas de *forecasting* que funcionarán mejor, considerando aspectos propios de la serie temporal como la aditividad o multiplicatividad [68].

Entonces, para conocer las características más comunes que poseen los D_k seleccionados en este trabajo se verificarán ciertas propiedades intrínsecas de la serie temporal, mediante pruebas de hipótesis. Para nuestro objetivo, estos resultados permitirán definir un conjunto de atributos comunes en las series temporales para acotar la familia de posibles soluciones, sobre los cuales los algoritmos de optimización pueden funcionar de manera estable y también para intentar predecir con el algoritmo de ML un valor óptimo de α . Las pruebas de hipótesis buscan comprobar, en los D_k conjuntos de datos: normalidad, estacionariedad, independencia, heterocedasticidad y tendencia. Para aceptar o rechazar las hipótesis planteadas, en cada test, se ha determinado un nivel de confianza del 95% que implica un nivel de significancia $\alpha_s = 0.05$.

Se realizaron dos tipos de pruebas de hipótesis para la comprobación de la normalidad, Shapiro-Wilk [69] y Kolmogorov-Smirnov [70] con corrección de Lilliefors [71]. Esta última prueba no necesita especificar la media y varianza de la distribución de los datos temporales. La primera se utilizó para los D_k donde $n < 50$ instancias, mientras que la segunda se empleó cuando el D_k tenía $n \geq 50$ instancias. En ambos casos las hipótesis se enuncian como:

H_0 : La serie tiene una distribución normal

H_a : La serie no tiene una distribución normal

Entonces, si el $p - value \leq \alpha_s$ se rechaza H_0 , ergo, no hay normalidad; mientras que si el $p - value > \alpha_s$ no se rechaza H_0 y existe normalidad.

Para comprobar la estacionariedad de la variable Y_t , en las series de tiempo, se utilizó la prueba de Kwiatkowski-Phillips-Schmidt-Shin (KPSS) [72], la misma que propugna el siguiente contraste de hipótesis:

H_0 : La serie es estacionaria en tendencia

H_a : La serie no es estacionaria, i.e., tiene una raíz unitaria

Para rechazar la hipótesis nula H_0 y aceptar la hipótesis alternativa H_a , se necesita que el $p - value \leq \alpha_s$.

La prueba de Ljung Box [73] es utilizada para validar si la serie de tiempo $(Y_t)_{t=1}^n$ contiene autocorrelación, i.e. que permite determinar la independencia entre los residuales de los datos. Las dos hipótesis que se contrastarán se enuncian como:

H_0 : Los errores están independientemente distribuidos

H_a : Los errores no están independientemente distribuidos

Cuando el $p - value \leq \alpha_s$ se rechaza H_0 , por lo tanto, no hay independencia i.e. que los datos exhiben autocorrelación serial. Si el $p - value > \alpha_s$ no se rechaza H_0 y existe independencia.

Otra propiedad ha considerar en un D_k en serie de tiempo es la homocedasticidad. La homocedasticidad implica que en un modelo de regresión lineal los errores se distribuyen con una varianza constante. Para una serie temporal la variable independiente está representada por Y_t y la variable dependiente es la componente temporal t . Cuando esta propiedad no se cumple se dice que los residuales no tienen varianza constante y, por lo tanto, existe heterocedasticidad. Se utilizó la prueba de hipótesis de Breusch-Pagan [74], donde:

H_0 : La serie tiene homocedasticidad

H_a : La serie no tiene homocedasticidad

Un $p - value \leq \alpha_s$ implica que la H_0 se rechaza y por lo tanto existe heterocedasticidad en la serie temporal,

mientras que si no se rechaza la H_0 se comprueba la homocedasticidad.

Finalmente, se analizó la tendencia en cada serie temporal, mediante la prueba de hipótesis de Mann-Kendall que al ser de tipo no paramétrica no realiza ninguna suposición implícita acerca de la normalidad de los datos. Las dos hipótesis que se plantean son:

H_0 : La serie no tiene tendencia

H_a : La serie tiene tendencia

Cuando el $p - value > \alpha_s$ se acepta la H_0 y se rechaza la H_a , esto implica que los datos de la serie temporal no tienen tendencia.

G. Algoritmo Trivial de Optimización con Método de Fuerza Bruta (FB)

Este algoritmo es tomado como referencia en el proceso de optimización del coeficiente de suavizamiento exponencial simple, dado que es un tipo de algoritmo de fuerza bruta [75], que explora todos los posibles valores de α dentro del dominio de $\mathbb{R} \in (0, 1)$. Considerando el vector α con un ancho de intervalo constante δ , los elementos α_q representan una posible solución óptima en los que se evalúa el error $\varepsilon_t(\alpha)$.

Este espacio transfinito de definición no permite, teóricamente, probar todos los valores posibles, por lo tanto, se limitará la cantidad de decimales de cada coeficiente α_q de acuerdo a una tolerancia dada T . Entonces, el algoritmo mediante un proceso iterativo particionará el dominio de $\mathbb{R} \in (0, 1)$ en N secciones. Así por ejemplo, si se desea tener una tolerancia de 10^{-3} , cada paso δ será de 0.001, sin considerar los valores extremos del rango.

Dado que este algoritmo trivial realiza una búsqueda exhaustiva en el dominio de definición del valor de α , se considera que ha llegado al óptimo cuando las cuatro métricas de error MSE, RMSE, MAE y MAPE sean las mínimas posibles dentro de todos los valores explorados.

El Pseudocódigo [1] muestra el funcionamiento operativo del algoritmo para su posterior implementación en un lenguaje de programación.

H. Algoritmo de Aproximaciones Sucesivas (AS)

Esta aproximación es una variación del algoritmo de optimización de fuerza bruta, que intenta reducir el tiempo de ejecución del proceso, realizando divisiones sucesivas en el espacio de soluciones de α . El usuario especifica la tolerancia T para que el algoritmo, mediante un proceso iterativo, realice las aproximaciones hacia el valor de convergencia del mínimo coeficiente de suavizamiento exponencial. Entonces, el algoritmo se inicializa con una tolerancia de 10^{-1} , es decir que, en la primera iteración, la separación de los elementos de α^1 es $\delta^1 = 0.1$. Luego, se ejecuta el método de *forecasting* SES, se calculan las cuatro métricas de error y se selecciona el α_q^1 que obtiene el menor error $\alpha_{q(min)}^1$. Si se necesita una tolerancia más baja, se realiza una segunda

Algoritmo 1 Algoritmo Fuerza Bruta (FB)

Entrada: Y_t, T

Salida: α_{opt}

```

1: Paso 1: Inicialización
2:  $\alpha \leftarrow T$ 
3:  $i \leftarrow T$ 
4: Paso 2: Método SES
5: while  $\alpha < 1$  do
6:    $sua \leftarrow SES(Y_t, \alpha)$ 
7:    $val \leftarrow sua[Y_t]$ 
8:    $MSE \leftarrow mse(val)$ 
9:    $RMSE \leftarrow rmse(val)$ 
10:   $MAE \leftarrow mae(val)$ 
11:   $MAPE \leftarrow mape(val)$ 
12:   $vector \leftarrow row(\alpha, MSE, RMSE, MAE, MAPE)$ 
13:  if  $vector == min(vector)$  then
14:     $result \leftarrow vector$ 
15:  end if
16:   $\alpha \leftarrow \alpha + i$ 
17: end while
18: Paso 3: Respuesta óptima
19:  $\alpha_{opt} \leftarrow result[\alpha]$ 
20: return  $\alpha_{opt}$ 

```

iteración considerando los intervalos formados por $\alpha_q^1(min)$ y sus vecinos superior $\alpha_{q+1}^1(min)$ e inferior $\alpha_{q-1}^1(min)$. Para seleccionar el nuevo intervalo se calcula la diferencia de los errores de los límites $\Delta_{inf}^1 = \varepsilon_t(\alpha_{q(min)}^1) - \varepsilon_t(\alpha_{q-1}^1(min))$ y $\Delta_{sup}^1 = \varepsilon_t(\alpha_{q+1}^1(min)) - \varepsilon_t(\alpha_{q(min)}^1)$ y se escoge aquel intervalo donde se cumpla $min(\Delta_{inf}^1, \Delta_{sup}^1)$. Ahora, para el nuevo espacio de soluciones, se repite el proceso de la primera iteración considerando $\delta^2 = \delta^1/10$. El algoritmo repite el proceso de aproximaciones sucesivas hasta cumplir con la tolerancia T , de modo que el intervalo explorado en la k -ésima iteración cumple con la condición $min(\Delta_{inf}^{k-1}, \Delta_{sup}^{k-1})$ y $\delta^k = \delta^{k-1}/10$.

Este algoritmo no es robusto cuando el espacio de soluciones es de tipo no convexo dado que en la primera iteración, donde la tolerancia es baja, solo explora una pequeña cantidad de posibles valores de α y por lo tanto podría quedar atrapado en un mínimo local sin explorar el mínimo global.

I. Algoritmo de Búsqueda Aleatoria (BA)

El algoritmo de búsqueda aleatoria [76], para encontrar un valor α_{opt} , realiza una exploración aleatoria dentro del rango de posibles valores definidos para el coeficiente de suavizamiento exponencial simple $\mathbb{R} \in (0, 1)$. En este enfoque, se generan n_{BA} valores aleatorios de α , con una media μ_{BA} y σ_{BA} dentro del rango definido y se evalúa el criterio o función objetivo de error $\varepsilon_t(\alpha)$. El mejor valor de α encontrado se actualiza si el valor actual supera al anterior. Este proceso se repite hasta que se cumpla la convergencia hacia un valor óptimo α_{opt} . El algoritmo de búsqueda aleatoria no garantiza la obtención del valor α_{opt} , especialmente en espacios de búsqueda complejos, porque depende de los puntos aleatorios de inicialización y sus parámetros de media y desviación estándar. Sin embargo, tiene la ventaja de no requerir información específica o heurística para guiar la búsqueda.

Algoritmo 2 Algoritmo Aproximaciones Sucesivas (AS)

Entrada: Y_t, T **Salida:** α_{opt}

```
1: Paso 1: Inicialización
2:  $k \leftarrow 0$ 
3:  $\alpha \leftarrow 0$ 
4: if  $T == 1$  then
5:   while  $\alpha \leq 0.9$  do
6:      $k \leftarrow k + 1$ 
7:      $sua \leftarrow SES(Y_t, \alpha)$ 
8:      $val \leftarrow sua[Y_t]$ 
9:      $MSE \leftarrow mse(val)$ 
10:     $RMSE \leftarrow rmse(val)$ 
11:     $MAE \leftarrow mae(val)$ 
12:     $MAPE \leftarrow mape(val)$ 
13:  end while
14: else if  $T > 1$  then
15:   while  $\alpha \leq 0.9$  do
16:      $k \leftarrow k + 1$ 
17:      $sua \leftarrow SES(Y_t, \alpha)$ 
18:      $val \leftarrow sua[Y_t]$ 
19:      $MSE \leftarrow mse(val)$ 
20:      $RMSE \leftarrow rmse(val)$ 
21:      $MAE \leftarrow mae(val)$ 
22:      $MAPE \leftarrow mape(val)$ 
23:   end while
24:   for  $i$  in  $2 : T$  do
25:      $k \leftarrow 10^{-i}$ 
26:      $min\_error \leftarrow \min(MSE, RMSE, MAE, MAPE)$ 
27:      $indice \leftarrow \text{index.min}(MSE, RMSE, MAE, MAPE)$ 
28:     if  $indice == 1$  then
29:        $downValue \leftarrow min\_error$ 
30:     else
31:        $downValue \leftarrow MSE[indice - 1, 2]$ 
32:     end if
33:     if  $indice == 10$  then
34:        $upValue \leftarrow min\_error$ 
35:     else
36:        $upValue \leftarrow MSE[indice + 1, 2]$ 
37:     end if
38:   end for
39: end if
```

En el Pseudocódigo [3](#) se observa el procedimiento algorítmico de la búsqueda aleatoria particularizada para hallar el mínimo global de α .

Algoritmo 3 Algoritmo de Búsqueda Aleatoria (BA)

Entrada: Y_t, n_{BA} **Salida:** α_{opt}

```
1:  $x \leftarrow rnorm(n_{BA} = 10, \mu_{BA} = 0.5, \sigma_{BA} = 0.7)$ 
2: for  $i$  in  $x$  do
3:    $sua \leftarrow SES(Y_t, \alpha)$ 
4:    $val \leftarrow sua[Y_t]$ 
5:    $MSE \leftarrow mse(val)$ 
6:    $RMSE \leftarrow rmse(val)$ 
7:    $MAE \leftarrow mae(val)$ 
8:    $MAPE \leftarrow mape(val)$ 
9:    $vector \leftarrow \text{row}(\alpha, MSE, RMSE, MAE, MAPE)$ 
10:  if  $vector == \min(vector)$  then
11:     $result \leftarrow vector$ 
12:  end if
13:   $\alpha \leftarrow \alpha + i$ 
14: end for
15:  $\alpha_{opt} \leftarrow result[\alpha]$ 
16: return  $\alpha_{opt}$ 
```

J. Algoritmo de Enjambre de Partículas (PSO)

El algoritmo de optimización por enjambre de partículas es una metaheurística utilizada para la optimización global de problemas. El algoritmo PSO inicializa con la creación de una población de partículas n_{PSO} , donde cada partícula representa una solución potencial en el espacio de búsqueda $\mathbb{R} \in (0, 1)$ definido por el dominio del coeficiente de suavizamiento exponencial simple α . Las partículas se inicializan con una posición X_{PSO} aleatoria y una velocidad V_{PSO} inicial.

Luego, se evalúa la calidad de cada partícula utilizando la función objetivo del problema que implica la minimización del error $\varepsilon_t(\alpha)$ con el método de pronóstico SES. La función objetivo asigna un valor numérico a cada solución en función de su calidad. Cada partícula actualiza su V_{PSO} y X_{PSO} en función de su mejor posición anterior, que representa a la mejor solución encontrada por la propia partícula y la mejor posición global encontrada por todas las partículas del enjambre. En cada iteración, las partículas ajustan su V_{PSO} y X_{PSO} para explorar el espacio en busca de soluciones óptimas. Esta información guía a las partículas hacia el mínimo global α_{opt} .

Algoritmo 4 Optimización por Enjambre de Partículas (PSO)

Entrada: Y_t **Salida:** α_{opt}

```
1: Paso 1: Inicialización
2:  $n_{PSO} \leftarrow 100$ 
3:  $n\_variables \leftarrow 1$ 
4:  $optimizacion \leftarrow \text{"Minimizar"}$ 
5: Paso 2: Método PSO
6: function  $funcion\_objetivo$ 
7:    $sua \leftarrow SES(Y_t, \alpha)$ 
8:    $val \leftarrow sua[Y_t]$ 
9: end function
10:  $particula \leftarrow \text{vector}(n = 5, mode = \text{"list"})$ 
11: for  $i$  in  $1 : n\_variables$  do
12:    $X_{PSO}[i] \leftarrow \text{random}(n = 1, min = 0, max = 1)$ 
13: end for
14:  $X_{PSO} \leftarrow \text{replicar}(NA, veces = n\_variables)$ 
15:  $particula[\text{"X}_{PSO}"] \leftarrow X_{PSO}$ 
16:  $V_{PSO} \leftarrow \text{replicar}(0, veces = n\_variables)$ 
17:  $particula[\text{"V}_{PSO}"] \leftarrow V_{PSO}$ 
18: Paso 3: Enjambre de partículas
19: for  $i$  in  $vector(enjambre[\text{"particulas"}])$  do
20:   if  $optimizacion == \text{"Minimizar"}$  then
21:      $mejor\_particula \leftarrow enjambre[\text{"particulas"}][i]$ 
22:   end if
23: end for
24:  $valor \leftarrow \text{call}(funcion\_objetivo, list(particula))$ 
25:  $MSE \leftarrow valor[1]$ 
26:  $RMSE \leftarrow valor[2]$ 
27:  $MAE \leftarrow valor[3]$ 
28:  $MAPE \leftarrow valor[4]$ 
29:  $\alpha_{opt} \leftarrow enjambre[\text{"mejor\_particula"}]$ 
30: return  $\alpha_{opt}$ 
```

K. Algoritmo Genético (AG)

El algoritmo genético es un método de optimización basado en teoría de la evolución y procesos genéticos. Este algoritmo es inicializado con la creación de una población inicial, n_{gen} , de soluciones potenciales representadas como

individuos I_{gen} en el dominio $\mathbb{R} \in (0, 1)$ de α . Cada I_{gen} tiene genes que representan a la variable Y_t del problema. Entonces, se evalúa la aptitud de cada I_{gen} utilizando la función objetivo $\varepsilon_t(\alpha)$. Los I_{gen} con mayor aptitud tienen más probabilidades de ser seleccionados para la reproducción, por lo tanto, son usados para formar una nueva generación g_{gen} mediante la técnica de torneo.

El algoritmo también realiza operaciones de cruzamiento o recombinación genética entre los I_{gen} seleccionados. Una cruce implica combinar partes de los cromosomas de los padres para crear descendientes con características mixtas. Además, se introducen cambios aleatorios en los cromosomas de algunos I_{gen} seleccionados, mediante mutaciones. La mutación ayuda a mantener la diversidad genética en la población y explorar nuevas soluciones. Los nuevos I_{gen} creados mediante cruzamiento y mutación reemplazan a algunos individuos menos aptos en la población actual. Este proceso se repite de forma iterativa hasta alcanzar un α_{opt} que minimiza las métricas de error MSE, RMSE, MAE y MAPE. El Pseudocódigo [3] muestra los parámetros de configuración utilizados para el proceso de optimización.

Algoritmo 5 Algoritmo Genético (AG)

Entrada: Y_t
Salida: α_{opt}

- 1: Paso 1: Inicialización
- 2: $n_{gen} \leftarrow 30$
- 3: $g_{gen} \leftarrow 500$
- 4: $n_{variables} \leftarrow 1$
- 5: $optimizacion \leftarrow$ “Minimizar”
- 6: $metodo_seleccion \leftarrow$ “torneo”
- 7: $metodo_cruce \leftarrow$ “uniforme”
- 8: **function** $funcion_genetica$
- 9: $sample[n_{gen}]$
- 10: $metodo_seleccion[I_{gen}]$
- 11: $g_{gen} = 0$
- 12: **while** $\alpha_{opt} \neq min$ **do**
- 13: $g_{gen} = g_{gen} + 1$
- 14: $reproduccion[I_{gen}]$
- 15: $metodo_cruce[I_{gen}]$
- 16: $mutacion[I_{gen}]$
- 17: $metodo_seleccion[I_{gen}]$
- 18: $n_{gen} \leftarrow reemplazos[I_{gen}]$
- 19: $vector \leftarrow row(MSE, RMSE, MAE, MAPE)$
- 20: $\alpha \leftarrow vector[I_{gen}]$
- 21: **end while**
- 22: $I_{gen} = \alpha_{opt}$
- 23: **end function**

L. Algoritmos de Machine Learning para Regresión

En este proyecto de investigación se propone el uso de dos algoritmos de aprendizaje automático, una Máquina de Soporte Vectorial (SVM) [77] y *Random Forest* (RF) [78], configurados para poder generar modelos de regresión. Estos dos algoritmos fueron seleccionados dado que, la literatura científica relacionada con pronóstico en series de tiempo, muestran su alta eficiencia y porque ambos algoritmos permiten realizar modelos de regresión no lineales.

Los dos algoritmos toman como entrada los resultados del α_{opt} para los veinte D_k . Este valor de α optimizado es la variable a predecir por ambos algoritmos de ML

y las variables predictoras independientes representan las características estadísticas de cada *dataset* y su cumplimiento de los supuestos más importantes de la serie temporal.

En la SVM, existen varios parámetros importantes que se deben considerar para obtener un buen rendimiento del modelo. Estos parámetros influyen en la capacidad de la SVM para ajustarse a los datos y generalizar correctamente. En este sentido la SVM fue configurada con un kernel radial, dado que la naturaleza de los datos es no lineal. El parámetro de regularización para la penalización de los errores en la función de pérdida fue de 0.25. El modelo final tenía 18 vectores de soporte.

Por otro lado, el algoritmo RF combina múltiples árboles de decisión individuales para formar un algoritmo *ensemble* que genera una respuesta cuantitativa continua basada en el promedio de las predicciones de los árboles individuales. Cada árbol se construye a partir del conjunto de datos D_k de entrenamiento y realiza divisiones en las características seleccionadas para predecir el valor numérico. Para el proceso de regresión se configuró 500 árboles y el número de variables en cada división es dos.

Para ambos algoritmos se utilizó el proceso de validación con el método *Leave One Out Cross Validation* [79]. Este método de validación divide al conjunto de datos en dos particiones, una de entrenamiento con la que se genera el modelo de aprendizaje y otra de prueba con la que se evalúa la eficiencia del modelo en términos del error. Este proceso se hace para tener un modelo que posea menos sesgo, dado que si se utiliza todo el conjunto de datos para generar el modelo, no existe una forma de comprobar que el modelo tenga validez ante una nueva instancia no contenida en el conjunto de datos inicial. La técnica consiste en realizar tantos modelos como instancias tenga el conjunto de datos, donde se entrena el modelo con $n - 1$ instancias y se prueba los resultados del modelo en la observación restante.

III. RESULTADOS Y DISCUSIÓN

Dado que el pronóstico en series de tiempo es un proceso de tipo estocástico, i.e., no determinístico, se realizará un banco de pruebas que obedece a las necesidades específicas de solución al problema.

Para el procesamiento computacional de los datos y modelado de los algoritmos de optimización, se utilizó el lenguaje de programación R V4.2.0 y el interfaz de desarrollo RStudio V2021.09.1, debido a que es software libre, *open access* y con un amplio conjunto de librerías para modelado en series de tiempo y visualización de datos.

Los experimentos se ejecutaron en un computador personal con procesador Intel Core i7-4510U, con sistema operativo Linux con distribución Ubuntu 22.04.2. Los *scripts* con los códigos para la simulación e implementación del *benchmark* de algoritmos de optimización se encuentran disponibles en el repositorio de código Github: https://github.com/dievalhu/serial_time_optimization. Además, la herramienta web con el modelo optimizado puede ser con-

D_k	Normalidad		Estacionariedad		Independencia		Heterocedasticidad		Tendencia		Convexidad
	$p - value$	Cumple	$p - value$	Cumple	$p - value$	Cumple	$p - value$	Cumple	$p - value$	Cumple	Cumple
D_{01}	0.4649	SI	0.1000	SI	0.7487	SI	0.9895	NO	0.1215	NO	SI
D_{02}	0.0179	NO	0.1000	SI	1.1483e-05	NO	0.8288	NO	5.9605e-07	SI	SI
D_{03}	0.0104	NO	0.0100	NO	3.4397e-05	NO	0.3693	NO	0.0000	SI	SI
D_{04}	0.0120	NO	0.0152	NO	0.5683	SI	0.4758	NO	0.5031	NO	SI
D_{05}	9.4646e-10	NO	0.0959	SI	2.9199e-14	NO	0.1798	NO	0.0000	SI	SI
D_{06}	0.1393	SI	0.0100	NO	0.0000	NO	0.4286	NO	0.0000	SI	SI
D_{07}	4.5552e-31	NO	0.1000	SI	8.6280e-09	NO	0.3065	NO	0.1067	NO	SI
D_{08}	4.3687e-16	NO	0.0709	SI	2.4858e-13	NO	0.6326	NO	0.0006	SI	SI
D_{09}	0.0021	NO	0.0100	NO	0.0000	NO	3.8235e-06	SI	0.0000	SI	NO
D_{10}	2.3905e-215	NO	0.1000	SI	0.0000	NO	0.0431	SI	0.0355	SI	SI
D_{11}	2.7254e-55	NO	0.0687	SI	0.0000	NO	0.0002	SI	0.0000	SI	SI
D_{12}	7.5721e-76	NO	0.0100	NO	0.0000	NO	6.8112e-17	SI	0.0006	SI	NO
D_{13}	3.4588e-61	NO	0.1000	SI	0.0000	NO	0.0312	SI	0.2740	NO	SI
D_{14}	0.0000	NO	0.0100	NO	0.0000	NO	4.1969e-05	SI	0.0000	SI	SI
D_{15}	1.7947e-08	NO	0.1000	SI	0.0000	NO	0.0369	SI	0.0004	SI	SI
D_{16}	0.0000	NO	0.0100	NO	0.0000	NO	1.2454e-09	SI	0.0000	SI	SI
D_{17}	0.0000	NO	0.0100	NO	0.0000	NO	7.7969e-196	SI	0.0000	SI	SI
D_{18}	0.0000	NO	0.0100	NO	0.0000	NO	1.2521e-15	SI	0.0000	SI	SI
D_{19}	0.0000	NO	0.1000	SI	0.0000	NO	5.8934e-05	SI	0.0000	SI	SI
D_{20}	0.0000	NO	0.0100	NO	0.0000	NO	5.6216e-72	SI	0.0000	SI	NO

TABLA II: Características de convexidad y comprobación de los supuestos para el modelado en los conjuntos de datos en series temporales

sumido y consultado en la siguiente URL: <https://dievalhu.shinyapps.io/AlfaOpt/>

Es menester describir los recursos computacionales, bajo los cuales las simulaciones fueron realizadas, dado que los tiempos de ejecución de los algoritmos dependen de los ciclos de reloj de la computadora, i.e., no se ha calculado la complejidad computacional de los algoritmos propuestos para la optimización del coeficiente de suavizamiento exponencial simple, sino únicamente el coste computacional que tiene dependencias del hardware, software y la forma en la que se hayan codificado los algoritmos en el lenguaje de programación. Para el banco de pruebas todos los algoritmos fueron codificados siguiendo la misma lógica y paradigma de programación, evitando utilizar la menor cantidad de anidaciones y priorizando el uso de las librerías y funciones directas que R posee, ya que estas se encuentran optimizadas para cada proceso.

Antes de realizar el proceso de *benchmark* experimental, para cada uno de los algoritmos de optimización, se ha realizado las comprobaciones de supuestos en los D_k conjuntos de datos de: normalidad, estacionariedad, independencia, heterocedasticidad y tendencia, mediante las pruebas de hipótesis que permiten determinar si se cumplen o no las condiciones supuestas, con el método del $p - value$. La Tabla II resume los resultados de cumplimiento de cada característica del conjunto de datos especificando su respectivo $p - value$ de decisión.

Con el objetivo de ejemplificar los resultados obtenidos en el proceso experimental, se ha tomado como caso de estudio el conjunto de datos D_{15} . Este *dataset* de 2745 instancias, derivadas del área de salud y cuya variable a pronosticar es el porcentaje de probabilidad de muerte por enfermedades cardiovasculares o cáncer, es una serie temporal donde los lapsos de tiempo son medidos en años. Para D_{15} , agrupado en G_3 , se cumplen con los supuestos de estacionariedad

y tendencia, mientras que la normalidad, independencia y homocedasticidad no se cumplen cuando se consideran las pruebas de hipótesis con un nivel de significancia de 0.05.

Con un 95% de confianza se observa que únicamente los *datasets* D_{01} y D_{06} cumplen con la propiedad de normalidad en la distribución de los datos en la serie temporal. La estacionariedad es una propiedad que indica si las características estadísticas de los datos se mantienen constantes a lo largo del tiempo y se cumple para diez de los veinte D_k . La independencia se cumple para los conjuntos de datos D_{01} y D_{04} , mientras que la propiedad de la homocedasticidad se cumple para los primeros ocho D_k ($k = \{1, \dots, 8\}$). Finalmente, la ausencia de tendencia en los datos temporales, está presente en D_{01} , D_{04} , D_{07} y D_{13} .

En la primera fase de experimentación se utilizó el algoritmo de fuerza bruta para determinar el valor del coeficiente de suavizamiento exponencial simple óptimo α_{opt} . Se considera que este algoritmo es el que proporciona la solución óptima de α al explorar todo el dominio de definición con una determinada tolerancia T . Para este algoritmo el tiempo de latencia de ejecución crece cuando se especifica un valor alto de T .

La Figura 3 muestra como el tiempo de ejecución t_{ex} del algoritmo se incrementa de forma no lineal, considerando los D_k con mayor número de instancias en cada grupo G_l . Es importante notar que los valores de t_{ex} también se incrementan cuando el número de instancias crece, dado que el modelo de *forecasting* explora más datos. Por ejemplo, para D_{15} con 2745 instancias, cuando $T = 10^{-1}$ el t_{ex} para determinar el α_{opt} fue de 0.011 [seg], para una $T = 10^{-3}$ el algoritmo converge en 0.629 [seg] y con una $T = 10^{-5}$ el tiempo se incrementa hasta 183.102 [seg]. Entonces, cuando el número n de instancias aumenta, junto con el valor de tolerancia T , el algoritmo FB tiene tiempos de latencia altos para la convergencia hacia el óptimo global. Esto último

se hace evidente, para el D_{20} con 197161 instancias con $T = 10^{-5}$, donde el algoritmo tarda más de 30 [min] para entregar un α_{opt} .

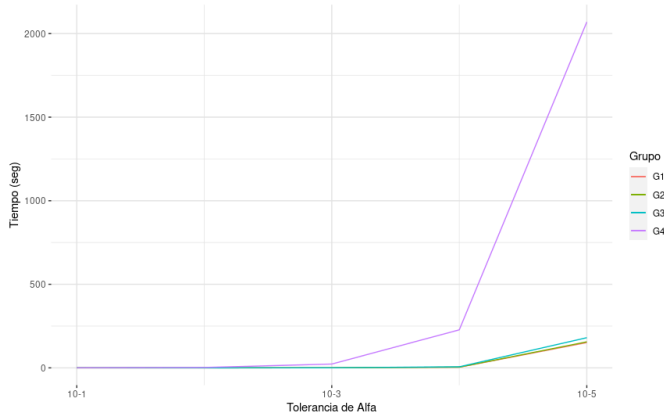


Fig. 3: Tiempo de ejecución del algoritmo de FB con variaciones en la tolerancia de α

Considerando una tolerancia $T = 10^{-3}$ se han determinado cuáles son los valores de α_{opt} en cada conjunto de datos. En cualquier serie de tiempo, para cada valor de α existe un correspondiente error asociado al modelo de pronóstico, entonces, este par de coordenadas $(\alpha_q, \varepsilon_t(\alpha_q))$ puede ser representado en un espacio \mathbb{R}^2 , con el objetivo de observar la convexidad de los conjuntos de datos. Se consideró al RMSE con normalización min-max como métrica de error para la determinación de la convexidad.

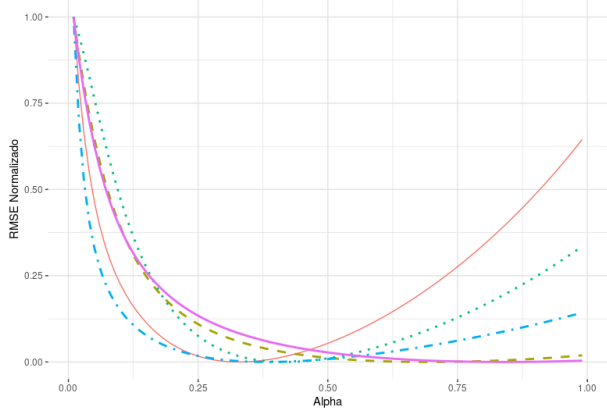


Fig. 4: Variación del RMSE normalizado del G_1 de datasets con respecto a α

Dado que este algoritmo explora todas las posibilidades de α , con la tolerancia definida, se puede observar en la Figura 4 como el RMSE va cambiando hasta encontrar el óptimo global para todos los datasets del G_1 . En todo este grupo de conjuntos de datos el espacio es convexo, ergo, se tiene un solo mínimo global, sin mínimos locales.

Es importante conocer si el espacio de soluciones es no convexo, dado que un algoritmo de búsqueda que utilice un punto inicial para encontrar el α_{opt} basado en una técnica

heurística o meta-heurística, podría encontrar el óptimo únicamente en el espacio de soluciones cercano al mínimo local. Para el G_2 de datasets, las curvas muestran que se encontró un mínimo local y un mínimo global en el D_{09} , como se observa en la Figura 5.

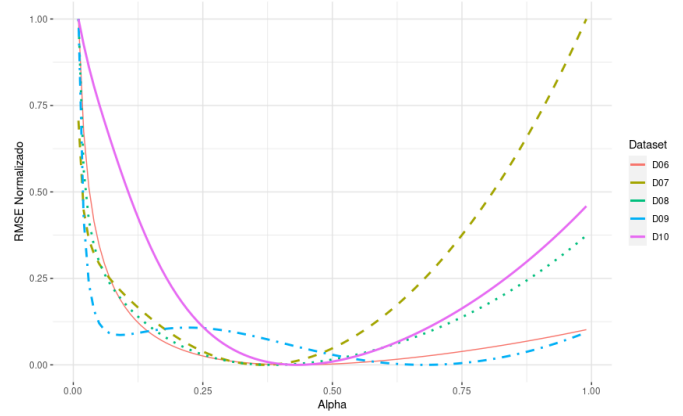


Fig. 5: Variación del RMSE normalizado del G_2 de datasets con respecto a α

En el conjunto de curvas de variación del RMSE normalizado del G_3 , se puede observar que el D_{12} no cumple con las características de convexidad. La Figura 6 muestra como este dataset tiene un mínimo local cuando $\alpha = 0.06$.

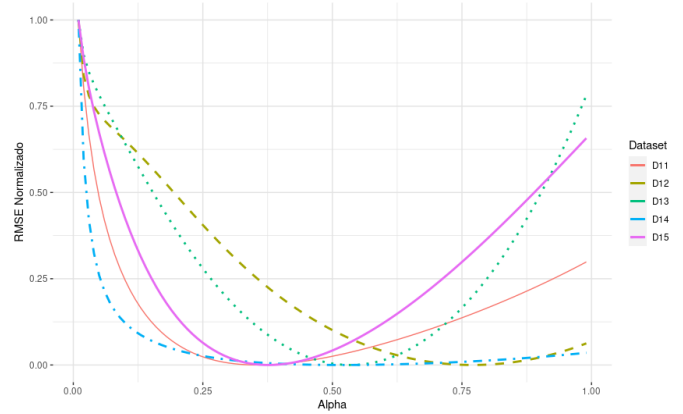


Fig. 6: Variación del RMSE normalizado del G_3 de datasets con respecto a α

En la Figura 7, que representa el conjunto de datos del grupo G_4 , el D_{20} tiene un máximo local al inicio de la curva de variación del RMSE, cuando el valor de $\alpha = 0.042$.

Para los veinte conjuntos de datos se utilizó la definición de convexidad para verificar esta propiedad, altamente deseable, mediante el *convex hull* que permite observar si la envolvente encuentra mínimos o máximos locales. Únicamente tres datasets no cumplieron con esta propiedad como se muestra en la Tabla 11. En la Figura 8 se observa el *convex hull* del D_{09} que no cumple con la propiedad de convexidad.

El *convex hull* de la Figura 9 muestra como la envolvente confina todos los puntos de coordenadas $(\alpha_q, \varepsilon_t(\alpha_q))$ en el

	D_{01}	D_{02}	D_{03}	D_{04}	D_{05}	D_{06}	D_{07}	D_{08}	D_{09}	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}	
FB	α_{opt}	0.329	0.694	0.417	0.383	0.839	0.429	0.366	0.385	0.678	0.432	0.770	0.535	0.519	0.377	0.418	0.665	0.474	0.497	0.201	
	t_{ex}	0.287	0.826	0.36	0.348	0.332	0.376	0.371	0.371	0.369	0.413	0.455	0.642	0.621	0.558	3.893	2.75	15.217	52.525	21.842	
	MSE	260.339	524.33	707.5e+01	105.9e+02	0.094	1638.22	347.3e+02	223.7e+07	357.521	237.1e+04	636.1e+06	624.6e+05	314.4e+04	3.01e+28	27.609	4.1e+016	0.094	475.6e+02	417.6e+02	6178.017
	RMSE	16.135	22.898	84.115	102.904	0.307	40.475	186.351	472.9e+01	18.908	153.9e+01	252.2e+02	790.3e+01	177.3e+01	173.4e+012	5.254	204.4e+06	0.307	218.1e+05	204.354	78.6
	MAE	13.171	18.152	67.481	76.376	0.204	30.696	92.831	287.7e+02	14.965	53.374	1.6e+04	4.9e+03	1.4e+03	4.83e+13	3.918	1.04e+08	0.120	348.5e+04	49.159	50.025
MAPE	0.225	0.021	0.235	0.276	0.06	0.058	0.622	0.329	0.109	85.038	0.333	Inf	9.757	0.628	0.218	Inf	0.039	0.3	Inf	0.717	
AS	α_{opt}	0.329	0.694	0.417	0.383	0.839	0.429	0.366	0.385	0.678	0.432	0.770	0.535	0.519	0.377	0.418	0.665	0.474	0.497	0.201	
	t_{ex}	0.065	0.063	0.05	0.047	0.049	0.05	0.048	0.05	0.051	0.05	0.011	0.056	0.056	0.073	0.041	0.209	0.112	0.300	0.375	0.524
	MSE	2.6e+02	5.2e+02	7.1e+03	1.1e+04	0.094	1.6e+03	3.5e+04	2.2e+09	3.6e+02	2.4e+06	6.4e+08	6.2e+07	3.1e+06	3.01e+28	27.610	4.18e+16	0.094	4.8e+14	4.2e+04	6.2e+03
	RMSE	16.135	22.898	84.116	1.02e+02	0.307	40.475	1.9e+02	4.7e+04	18.908	1.5e+03	2.5e+04	7.9e+03	1.8e+03	1.73e+14	5.255	2.04e+08	0.307	2.2e+07	204.355	78.618
	MAE	13.171	18.152	67.481	76.376	0.204	30.696	92.515	288e+04	14.961	53.374	1.6e+04	4.9e+03	1.4e+03	4.83e+13	3.918	1.04e+08	0.120	3.5e+06	49.099	49.902
MAPE	0.225	0.021	0.235	0.276	0.06	0.058	0.621	0.329	0.109	85.1	0.333	Inf	9.733	0.628	0.218	Inf	0.039	0.297	Inf	0.708	
AG	α_{opt}	0.607	0.470	0.641	0.557	0.573	0.434	0.520	0.499	0.403	0.465	0.551	0.527	0.382	0.382	0.468	0.383	0.397	0.152	0.471	0.474
	t_{ex}	2.029	1.934	2.014	1.991	1.978	2.074	2.053	2.108	2.202	2.459	2.975	3.209	3.801	4.423	3.762	8.852	17.291	117.859	129.759	192.100
	MSE	260.339	524.330	7075.290	10589.27	0.094	1638.22	34726.76	2.2e+09	357.521	2.4e+06	6.4e+08	6.2e+07	3.1e+06	3.01e+28	27.610	4.18e+16	0.094	4.76e+14	41760.67	6178.017
	RMSE	16.528	22.944	84.144	102.942	0.877	44.116	186.675	49435.17	19.523	1781.242	25667.56	9104.2	1822.753	1.73e+14	5.620	2.1e+08	0.320	2.4e+07	206.602	85.267
	MAE	13.564	18.029	67.343	76.112	0.619	33.437	98.451	30251.81	15.752	953.518	16224.94	6131.031	1497.334	4.81e+13	4.216	1.1e+08	0.137	6.1e+06	46.752	51.630
MAPE	0.228	0.0204	0.234	0.277	0.150	0.062	0.647	0.359	0.116	72.554	0.326	Inf	11.410	0.630	0.233	Inf	0.049	0.951	Inf	0.641	
BA	α_{opt}	0.328	0.604	0.431	0.392	0.766	0.755	0.356	0.411	0.573	0.380	0.276	0.871	0.489	0.515	0.326	0.414	0.399	0.533	0.423	0.219
	t_{ex}	0.017	0.014	0.013	0.017	0.015	0.012	0.018	0.012	0.014	0.021	0.006	0.038	0.034	0.029	0.027	0.062	0.071	0.265	0.47	0.238
	MSE	260.339	527.024	7079.319	1.1e+04	0.095	1824.795	34729.017	2.2e+09	360.407	2.4e+06	6.4e+08	6.3e+07	3.1e+06	3.01e+28	27.767	0.414	0.095	4.8e+14	41828.745	6180.488
	RMSE	16.135	22.957	84.139	102.910	0.308	42.718	186.357	47302.504	18.984	1542.412	25312.269	7935.043	1774.286	1.7e+14	5.269	4.18e+016	0.308	2.2e+07	204.520	78.616
	MAE	13.166	18.015	67.579	76.393	0.203	33.115	93.616	28669.555	15.217	561.556	16230.561	4972.991	1377.772	4.8e+13	3.935	1.04e+08	0.116	3.38e+06	204.521	49.908
MAPE	0.225	0.020	0.235	0.276	0.060	0.062	0.626	0.328	0.111	84.361	0.340	Inf	9.982	0.629	0.220	Inf	0.037	0.273	Inf	0.709	
PSO	α_{opt}	0.219	0.368	0.352	0.359	0.794	0.444	0.983	0.456	0.894	0.431	0.846	0.735	0.385	0.381	0.445	0.589	0.988	0.929	0.331	
	t_{ex}	0.949	0.956	0.989	0.888	0.945	0.966	0.986	0.839	0.923	0.864	0.926	1.056	1.023	1.142	1.014	1.647	3.168	15.227	16.924	22.642
	MSE	267.018	576.602	7177.349	10597.8	0.095	1638.735	42811.31	2.2e+09	370.075	2.9e+06	6.4e+08	6.3e+07	3.2e+06	3.04e+28	27.611	4.18e+16	0.100	5.65e+14	44312.98	6282.106
	RMSE	16.341	24.013	84.719	102.946	0.308	40.481	206.909	47369.88	19.237	1713.59	25277.9	7921.483	1795.139	1.74e+14	5.255	2.04e+08	0.316	2.4e+07	210.750	79.260
	MAE	12.976	17.522	67.179	76.117	0.202	30.693	60.318	28524.41	14.719	417.018	16109.49	4970.831	1340.957	4.79e+13	3.918	1.04e+08	0.112	3.1e+06	46.380	49.579
MAPE	0.819	0.920	0.769	0.947	0.838	0.997	0.893	0.988	0.997	0.964	0.986	0.968	0.932	0.818	0.949	1.001	0.999	0.995	0.892		

TABLA III: Resumen estadístico de los conjuntos de datos en series temporales, utilizados para la experimentación, desgregados por área de conocimiento

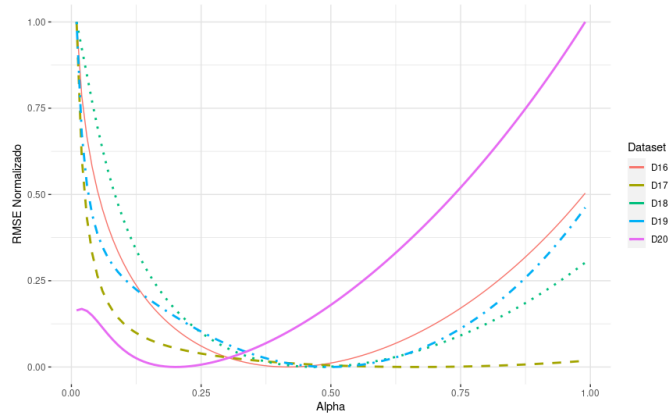


Fig. 7: Variación del RMSE normalizado del G_4 de datasets con respecto a α

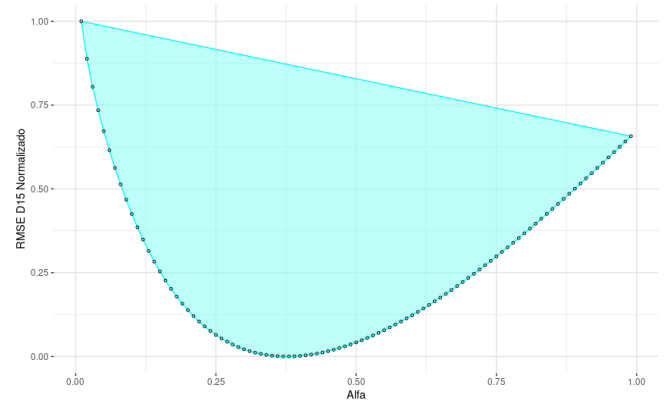


Fig. 9: Convex hull del conjunto de datos convexo D_{15}

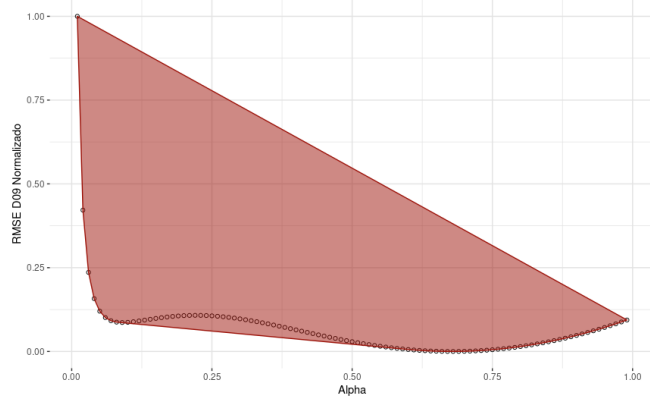


Fig. 8: Convex hull del conjunto de datos no convexo D_{09}

interior de su área, por lo tanto se concluye que el D_{15} , tomado como caso de estudio, cumple con la propiedad de convexidad.

Por otro lado, los otros cuatro algoritmos de optimización utilizados para determinar el coeficiente de suavizamiento exponencial simple con el error más bajo α_{opt} , fueron eje-

cutados de forma individual para todos los D_k con el objetivo de calcular las cuatro métricas de error MSE, RMSE, MAE y MAPE y comparar a modo de *benchmark* los resultados. En la Tabla III se resumen los resultados de las simulaciones con los cinco algoritmos de optimización propuestos, para todos los conjuntos de datos. El valor del tiempo, t_{ex} , de la simulación se expresa en segundos y las unidades del error varían dependiendo de la métrica de evaluación.

Los resultados muestran que, para todos los conjuntos de datos, el algoritmo de aproximaciones sucesivas AS obtuvo los mismos valores de α_{opt} que el algoritmo de fuerza bruta considerado como el método de optimización referencial. El algoritmo AS convergió hacia el α_{opt} , incluso para los datasets no convexos D_{09} , D_{12} , D_{20} y además los tiempos de ejecución son menores para todos los D_k en comparación con los otros algoritmos de optimización. El rango de t_{ex} para el algoritmo AS fue de $[0.011, 0.524]$ segundos en contraste con el rango de tiempos de FB comprendidos entre $[0.287, 21.842]$ segundos. Por otro lado, el algoritmo genético fue el que mayor latencia presentó en la experimentación con 192.1 segundos para su convergencia hacia el mínimo global, en el conjunto de datos de 197161 instancias.

Para todos los algoritmos de optimización se observa una relación directamente proporcional entre el t_{ex} y el número n de instancias del conjunto de datos.

Analizando los errores, la métrica MAPE no pudo ser calculada por el software en varios *datasets* para los algoritmos FB, AS, AG y BA, debido a que el valor resultante es demasiado grande en términos porcentuales y tiende hacia el infinito. Un MAPE infinito se obtiene cuando los valores pronosticados son significativamente diferentes de los valores reales o cuando el valor de Y_t es cercano a cero, lo que puede conducir a una alta variabilidad de esta métrica.

La mayor divergencia encontrada, entre el valor α_{opt} referencial del algoritmo FB y el α de los otros cuatro algoritmos de optimización propuestos, ocurre en el conjunto de datos D_{07} con el algoritmo PSO. Por otro lado, entre los cinco α obtenidos por los algoritmos de optimización, para el D_{16} se obtiene la menor varianza $\sigma_\alpha^2 = 0.0004853$. Esto último implica que D_{16} es el *dataset* donde la mayoría de algoritmos encuentra el óptimo global para el coeficiente de suavizamiento exponencial simple.

Al analizar el rango, la media y varianza entre los α de un mismo grupo G_i , se puede observar que el número de instancias n de un conjunto de datos no determina un patrón de α fijo para ese grupo de *datasets*. Así, si se considera que $\alpha \in (0, 1)$, las diferencias entre los valores mínimos y máximos de α para los grupos G_1 , G_2 , G_3 y G_4 son del 51%, 31.2%, 41.4% y 46.4%, respectivamente.

Es importante acotar que para los algoritmos de fuerza bruta y aproximaciones sucesivas, se necesita especificar la tolerancia T para alcanzar un grado de precisión deseado. La tolerancia no es un problema para los algoritmos de búsqueda aleatoria, genético y de enjambre de partículas ya que la respuesta es un valor cuantitativo continuo donde se puede definir el número de cifras significativas de manera libre.

En la Figura 10 se muestra la serie temporal junto a la curva de pronóstico del método de suavizamiento exponencial simple con $\alpha_{opt} = 0.377$, para el conjunto de datos D_{15} de 2745 instancias. Se puede observar como la curva de datos real de la serie de tiempo se aproxima a la curva de pronósticos generada por el modelo SES.

Los resultados del proceso de optimización del coeficiente de suavizamiento exponencial simple α , ejecutados sobre los veinte D_k de experimentación, han generado un nuevo conjunto de datos que será la entrada de los modelos predictivos basados en aprendizaje automático. Este nuevo *dataset* es de tipo estructurado y sus dimensiones son 20×12 , donde cada columna es una variable dependiente o independiente, y cada fila es una observación derivada de los resultados de las simulaciones ejecutadas para cada D_k .

La variable dependiente a predecir es el valor de α_{opt} , en función de once variables independientes predictoras, i.e., que el algoritmo de ML está configurado para realizar una tarea de regresión, ya que α_{opt} es una variable cuantitativa continua. El valor de α_{opt} es el resultado del proceso de minimización ejecutado con el algoritmo referencial de FB.

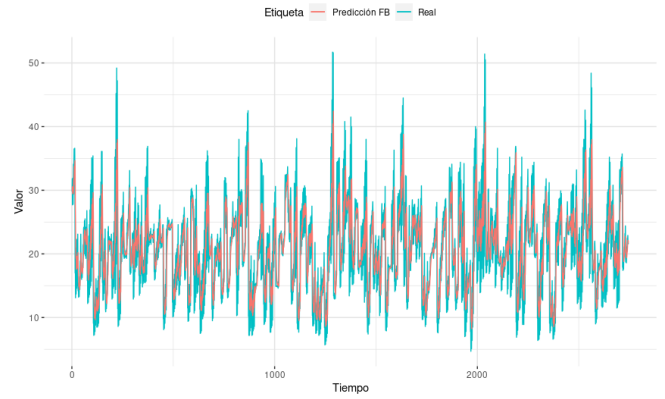


Fig. 10: Serie de tiempo vs. pronóstico SES con $\alpha_{opt} = 0.377$ para D_{15}

La Figura 11 muestra la distribución de los valores de α_{opt} , para cada D_k , con el algoritmo basado en fuerza bruta.

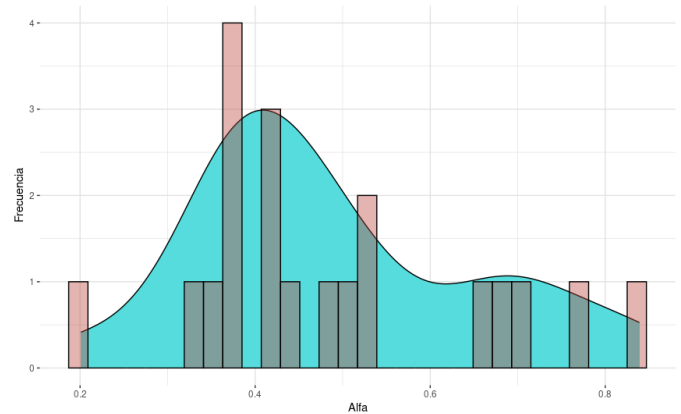


Fig. 11: Histograma y curva de densidad de los valores de α_{opt} para todos los *datasets* D_k con el algoritmo de FB

Las cinco primeras variables independientes, consideradas para el modelado, se desprenden de las características estadísticas de la serie temporal y son el número n de instancias, junto con su mínimo, máximo, media μ y desviación estándar σ . Luego, el modelo también toma como entrada, el cumplimiento o no, de las seis propiedades examinadas en cada D_k , i.e., normalidad, estacionariedad, independencia, heterocedasticidad, tendencia y convexidad. Esta condición de cumplimiento es una variable de tipo dicotómica, ergo, puede ser codificada como un booleano con estado 0 para el no cumplimiento y 1 para el cumplimiento.

Para este nuevo conjunto de datos se comprobó las suposiciones de normalidad, homocedasticidad, linealidad e independencia con las mismas pruebas de hipótesis que se usaron para las series de tiempo. Este nuevo *dataset* cumple con todos los supuestos y, por lo tanto, puede ser utilizado en los dos algoritmos de ML propuestos: *random forest* y máquina de soporte vectorial.

Las once variables independientes ingresaron en los dos

algoritmos de ML y se generó un modelo de regresión para cada uno. Para ambos modelos se realizó un proceso de validación cruzada con el método de LOOCV. Se utilizó este tipo de validación, que suele ser costosa computacionalmente, porque el conjunto de datos es pequeño y el tiempo de respuesta para ambos algoritmos estuvo en el orden de las décimas de segundo. Entonces, para la primera iteración del método LOOCV, la primera instancia representa a la partición de prueba y las 19 restantes son utilizadas para el entrenamiento del modelo. Luego, la segunda instancia se usó para la prueba y las otras 19 observaciones generaron el modelo de la segunda iteración. El proceso se repitió hasta que todas las 20 instancias, en algún momento, fueron tomadas como datos de prueba. Este proceso iterativo generó 20 modelos diferentes y se seleccionó el modelo promedio, para cada algoritmo, con el objetivo evitar sesgar los resultados en un ambiente de producción.

D_k	RF			SVM	
	α_{optFB}	α_{RF}	e_r	α_{SVM}	e_r
D_1	0.329	0.345	0.0496	0.345	0.0495
D_2	0.694	0.638	0.0813	0.625	0.0999
D_3	0.417	0.433	0.0392	0.433	0.0391
D_4	0.383	0.399	0.0426	0.399	0.0429
D_5	0.839	0.637	0.2404	0.625	0.2555
D_6	0.429	0.445	0.0382	0.445	0.0381
D_7	0.366	0.382	0.0446	0.382	0.0447
D_8	0.385	0.635	0.6492	0.625	0.6223
D_9	0.678	0.662	0.0241	0.662	0.0240
D_{10}	0.432	0.416	0.0379	0.395	0.0857
D_{11}	0.356	0.406	0.1410	0.395	0.1083
D_{12}	0.770	0.687	0.1076	0.663	0.1394
D_{13}	0.535	0.519	0.0305	0.519	0.0306
D_{14}	0.519	0.502	0.0314	0.503	0.0315
D_{15}	0.377	0.393	0.0431	0.393	0.0434
D_{16}	0.418	0.434	0.0392	0.434	0.0389
D_{17}	0.665	0.637	0.0427	0.563	0.1529
D_{18}	0.474	0.473	0.0014	0.461	0.0267
D_{19}	0.497	0.481	0.0328	0.481	0.0329
D_{20}	0.201	0.310	0.5427	0.298	0.4829

TABLA IV: Comparación entre el valor de α_{opt} referencial y los algoritmos de aprendizaje automático RF y SVM

En la Tabla IV se muestra la comparación entre el α_{optFB} real optimizado con el algoritmo FB, considerado como referencia, y los α_{RF} , α_{SVM} predichos por los dos algoritmos de ML. Para las predicciones de ambos algoritmos se calculó el valor del error relativo e_r a nivel de instancia. El conjunto de datos D_{20} , es la instancia con mayor error relativo de 54.27% y 48.29 para RF y SVM, respectivamente.

Para evaluar la bondad de ajuste de los dos algoritmos se utilizó el coeficiente de determinación r^2 ; entonces, en el algoritmo de *random forest* el $r^2 = 0.7663$ y para la máquina de soporte vectorial el $r^2 = 0.7494$. En forma genérica, la literatura científica suele considerar que una buena bondad de ajuste se alcanza cuando el valor del coeficiente de determinación es superior a 0.7. La bondad de ajuste para los algoritmos genético AG, de búsqueda aleatoria BA y de enjambre de partículas PSO fueron $r^2 = 0.00889$, $r^2 = 0.7169$ y $r^2 = 0.2179$, respectivamente. Los resultados muestran que los dos algoritmos de ML tienen mejores

resultados que AG, BA, PSO y además su ejecución t_{ex} es mucho más rápida.

IV. CONCLUSIONES

Esta investigación desarrolló un proceso de optimización matemática para mejorar el rendimiento de un modelo de *forecasting* de suavizamiento exponencial simple. Los resultados de los cinco algoritmos de optimización y los dos derivados de *machine learning* propuestos mostraron la posibilidad de determinar un valor del coeficiente de suavizamiento exponencial α , que minimizó el error del pronóstico en la serie de tiempo, con un tiempo de convergencia menor a un segundo para el algoritmo de aproximaciones sucesivas y de 0.01 segundos para los algoritmos de ML. Se puede concluir que para conjuntos de datos que no superen el orden de las 1000 instancias y cuya $T = 10^{-3}$ el algoritmo de fuerza bruta, funciona de manera apropiada, dado que el tiempo de convergencia hacia el óptimo global es menor a un segundo.

Se analizaron distintos conjunto de datos de *benchmarking* con la finalidad de observar los resultados cuando se varía la naturaleza de los datos y la cantidad de instancias involucradas en el proceso de pronóstico y se determinó que los algoritmos de ML son especialmente útiles cuando la serie de tiempo tiene una gran cantidad de instancias. Los *datasets* analizados fueron agrupados en función de la cantidad de instancias, pero como prospectivas futuras se sugiere el agrupamiento en función de una ANOVA. Los resultados experimentales mostraron que es posible explorar el transfinito original de dominio de definición de α , sin la necesidad de explorar de forma exhaustiva todos los valores del rango, alcanzando un rendimiento de convergencia óptimo independientemente del tamaño del *dataset* y de su convexidad.

Analizando las métricas de error se determinó que MAPE puede tender a infinito, cuando la magnitud de la variable Y_t de análisis en la serie de tiempo es grande. Para estos casos, se recomienda utilizar variantes del MAPE, como el *Symmetric Mean Absolute Percentage Error* (SMAPE), que soluciona esta limitación al tomar el valor absoluto de la diferencia porcentual relativa en lugar de la diferencia absoluta en el denominador.

REFERENCIAS

- [1] L. D. Xu, E. L. Xu, and L. Li, "Industry 4.0: state of the art and future trends," *International Journal of Production Research*, vol. 56, no. 8, pp. 2941–2962, 2018.
- [2] G. C. Bowker, K. Baker, F. Millerand, and D. Ribes, "Toward information infrastructure studies: Ways of knowing in a networked environment," in *International handbook of internet research*. Springer, 2009, pp. 97–117.
- [3] K. Vassakis, E. Petrakis, and I. Kopanakis, "Big data analytics: applications, prospects and challenges," in *Mobile big data*. Springer, 2018, pp. 3–20.
- [4] N. J. Nilsson, *The quest for artificial intelligence*. Cambridge University Press, 2009.
- [5] I. El Naqa and M. J. Murphy, "What is machine learning?" in *machine learning in radiation oncology*. Springer, 2015, pp. 3–11.

- [6] D. Nagar, S. Raghav, A. Bhardwaj, R. Kumar, P. Lata Singh, and R. Sindhwani, "Machine learning: Best way to sustain the supply chain in the era of industry 4.0," *Materials Today: Proceedings*, vol. 47, pp. 3676–3682, 2021, 3rd International Conference on Computational and Experimental Methods in Mechanical Engineering.
- [7] P. H. Franses, P. H. B. Franses *et al.*, *Time series models for business and economic forecasting*. Cambridge university press, 1998.
- [8] R. Prudêncio and T. Ludermir, "Using machine learning techniques to combine forecasting methods," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2004, pp. 1122–1127.
- [9] T. Latruwe, M. Van der Wee, P. Vanleenhove, J. Devriese, S. Verbrugge, and D. Colle, "A long-term forecasting and simulation model for strategic planning of hospital bed capacity," *Operations Research for Health Care*, vol. 36, p. 100375, 2023.
- [10] V. K. Sudarshan, M. Brabrand, T. M. Range, and U. K. Wiil, "Performance evaluation of emergency department patient arrivals forecasting models by including meteorological and calendar information: A comparative study," *Computers in Biology and Medicine*, vol. 135, p. 104541, 2021.
- [11] M. S. Amalnick, N. Habibifar, M. Hamid, and M. Bastan, "An intelligent algorithm for final product demand forecasting in pharmaceutical units," *International Journal of System Assurance Engineering and Management*, vol. 11, pp. 481–493, 2020.
- [12] X. Zhu, A. Ninh, H. Zhao, and Z. Liu, "Demand forecasting with supply-chain information and machine learning: Evidence in the pharmaceutical industry," *Production and Operations Management*, vol. 30, no. 9, pp. 3231–3252, 2021.
- [13] L. N. Pereira and V. Cerqueira, "Forecasting hotel demand for revenue management using machine learning regression methods," *Current Issues in Tourism*, vol. 25, no. 17, pp. 2733–2750, 2022.
- [14] T. Zheng, S. Liu, Z. Chen, Y. Qiao, and R. Law, "Forecasting daily room rates on the basis of an lstm model in difficult times of hong kong: evidence from online distribution channels on the hotel industry," *Sustainability*, vol. 12, no. 18, p. 7334, 2020.
- [15] Y. Zhang, G. Li, B. Muskat, and R. Law, "Tourism demand forecasting: A decomposed deep learning approach," *Journal of Travel Research*, vol. 60, no. 5, pp. 981–997, 2021.
- [16] A. Kulshrestha, V. Krishnaswamy, and M. Sharma, "Bayesian bilstm approach for tourism demand forecasting," *Annals of tourism research*, vol. 83, p. 102925, 2020.
- [17] N. Parida, D. Mishra, K. Das, and N. K. Rout, "Development and performance evaluation of hybrid kelm models for forecasting of agro-commodity price," *Evolutionary Intelligence*, vol. 14, pp. 529–544, 2021.
- [18] K. D. Chaudhuri and B. Alkan, "A hybrid extreme learning machine model with harris hawks optimisation algorithm: an optimised model for product demand forecasting applications," *Applied Intelligence*, vol. 52, no. 10, pp. 11489–11505, 2022.
- [19] C. Auppakorn and N. Phumchusri, "Daily sales forecasting for variable-priced items in retail business," in *Proceedings of the 4th International Conference on Management Science and Industrial Engineering*, 2022, pp. 80–86.
- [20] M. Çetinkaya and T. Acarman, "Next-day electricity demand forecasting using regression," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*. IEEE, 2021, pp. 1549–1554.
- [21] B. Banitalebi, S. S. Appadoo, A. Thavaneswaran, and M. E. Hoque, "Modeling of short-term electricity demand and comparison of machine learning approaches for load forecasting," in *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, 2020, pp. 1302–1307.
- [22] J. Van Belle, T. Guns, and W. Verbeke, "Using shared sell-through data to forecast wholesaler demand in multi-echelon supply chains," *European Journal of Operational Research*, vol. 288, no. 2, pp. 466–479, 2021.
- [23] C.-C. Wang, C.-H. Chien, and A. J. Trappey, "On the application of arima and lstm to predict order demand based on short lead time and on-time delivery requirements," *Processes*, vol. 9, no. 7, p. 1157, 2021.
- [24] S. Islam, S. H. Amin, and L. J. Wardley, "Machine learning and optimization models for supplier selection and order allocation planning," *International Journal of Production Economics*, vol. 242, p. 108315, 2021.
- [25] E. Spiliotis, S. Makridakis, A. Kaltsounis, and V. Assimakopoulos, "Product sales probabilistic forecasting: An empirical evaluation using the m5 competition data," *International Journal of Production Economics*, vol. 240, p. 108237, 2021.
- [26] E. S. Gardner Jr, "Exponential smoothing: The state of the art," *Journal of forecasting*, vol. 4, no. 1, pp. 1–28, 1985.
- [27] —, "Exponential smoothing: The state of the art—part ii," *International journal of forecasting*, vol. 22, no. 4, pp. 637–666, 2006.
- [28] R. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder, *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media, 2008.
- [29] T. B. Fomby, "Exponential smoothing models," *Manual sas/ets software: time series forecasting system*, vol. 6, pp. 225–235, 2008.
- [30] D. H. Wolpert, W. G. Macready *et al.*, "No free lunch theorems for search," Citeseer, Tech. Rep., 1995.
- [31] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.
- [32] S. P. Adam, S.-A. N. Alexandropoulos, P. M. Pardalos, and M. N. Vrahatis, "No free lunch theorem: A review," *Approximation and Optimization: Algorithms, Complexity and Applications*, pp. 57–82, 2019.
- [33] A. Bauer, M. Züfle, J. Grohmann, N. Schmitt, N. Herbst, and S. Kounev, "An automated forecasting framework based on method recommendation for seasonal time series," in *Proceedings of the ACM/SPEC International Conference on Performance Engineering*, 2020, pp. 48–55.
- [34] W. J. Stevenson, *Operations management*. McGraw-Hill Education, 2015.
- [35] W. L. Berry and F. W. Bliemel, "Selecting exponential smoothing constants: an application of pattern search," *International journal of production research*, vol. 12, no. 4, pp. 483–499, 1974.
- [36] S. K. Paul, "Determination of exponential smoothing constant to minimize mean square error and mean absolute deviation," *Global Journals of Research in Engineering*, vol. 11, no. 3, pp. 31–34, 2011.
- [37] N. L. Marpaung, K. R. Salim, R. Amri, E. Ervianto *et al.*, "Application of single exponential smoothing in forecasting number of new students acceptance," *Int. J. Technol. Eng. Stud*, vol. 5, no. 6, pp. 169–182, 2019.
- [38] T. E. Dielman, "Choosing smoothing parameters for exponential smoothing: Minimizing sums of squared versus sums of absolute errors," *Journal of modern applied statistical methods*, vol. 5, no. 1, p. 11, 2006.
- [39] C. Karmaker, "Determination of optimum smoothing constant of single exponential smoothing method: A case study," *International Journal of Research in Industrial Engineering*, vol. 6, no. 3, pp. 184–192, 2017.
- [40] H. V. Ravinder *et al.*, "Determining the optimal values of exponential smoothing constants—does solver really work?" *American Journal of Business Education (AJBE)*, vol. 9, no. 1, pp. 1–14, 2016.
- [41] W. Junthopas and C. Wongoutong, "Setting the initial value for single exponential smoothing and the value of the smoothing constant for forecasting using solver in microsoft excel," *Applied Sciences*, vol. 13, no. 7, p. 4328, 2023.
- [42] W. Setiawan, E. Juniati, and I. Farida, "The use of triple exponential smoothing method (winter) in forecasting passenger of pt kereta api indonesia with optimization alpha, beta, and gamma parameters," in *2016 2nd International Conference on Science in Information Technology (ICSITech)*. IEEE, 2016, pp. 198–202.
- [43] M. H. P. Swari, I. P. S. Handika, and I. K. S. Satwika, "Comparison of simple moving average, single and modified single exponential smoothing," in *2021 IEEE 7th Information Technology International Seminar (ITIS)*. IEEE, 2021, pp. 1–5.
- [44] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4. IEEE, 1995, pp. 1942–1948.
- [45] M. H. P. Swari, I. P. S. Handika, I. K. S. Satwika, and H. E. Wahani, "Optimization of single exponential smoothing using particle swarm optimization and modified particle swarm optimization in sales forecast," in *2022 IEEE 8th Information Technology International Seminar (ITIS)*. IEEE, 2022, pp. 292–296.
- [46] A. Chusyairi, R. N. Pelsri, and E. Handayani, "Optimization of exponential smoothing method using genetic algorithm to predict e-report service," in *2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE)*. IEEE, 2018, pp. 292–297.

- [47] J. Holland, "Adaptation in natural and artificial systems, univ. of mich. press," *Ann Arbor*, vol. 7, pp. 390–401, 1975.
- [48] J. Malaysia, "Genetic algorithm for parameter estimation in double exponential smoothing," *Australian Journal of Basic and Applied Sciences*, vol. 5, no. 7, pp. 1174–1180, 2011.
- [49] A. D. Rahajoe, "Forecasting feature selection based on single exponential smoothing using wrapper method," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 6, 2019.
- [50] A. K. Aditama *et al.*, "Optimizing single exponential smoothing method by using genetics algorithm for object selling prediction," in *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*. IEEE, 2020, pp. 77–82.
- [51] H. M. Safhi, B. Frikh, and B. Ouhbi, "Assessing reliability of big data knowledge discovery process," *Procedia Computer Science*, vol. 148, pp. 30–36, 2019, the Second International Conference On Intelligent Computing In Data Sciences, ICDS2018.
- [52] S. Feferman, "Transfinite recursive progressions of axiomatic theories," *The Journal of symbolic logic*, vol. 27, no. 3, pp. 259–316, 1962.
- [53] D. Dua and C. Graff, "Uci machine learning repository," 2017. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [54] Kaggle, "Your machine learning and data science community." [Online]. Available: <https://www.kaggle.com/>
- [55] Data.gov, "The home of the u.s. government's open data," *Data.gov*. [Online]. Available: <https://data.gov/>
- [56] D. Hand, H. Mannila, and P. Smyth, "Principles of data mining". the mit press," in *A comprehensive, highly technical look at the math and science behind extracting useful information from large databases*, 2001, vol. 546.
- [57] C. C. Aggarwal *et al.*, *Data mining: the textbook*. Springer, 2015, vol. 1.
- [58] UNESCO, "Áreas de conocimiento unesco." [Online]. Available: <https://bpb-us-w2.wpmucdn.com/blogs.udla.edu.ec/dist/b/383/files/2022/03/Áreas-de-Conocimiento-UNESCO.pdf>
- [59] J. W. Tukey, "Exploratory data analysis addison-wesley," *Reading, Ma*, vol. 688, pp. 581–582, 1977.
- [60] S. Kappal *et al.*, "Data normalization using median median absolute deviation mmad based z-score for robust predictions vs. min–max normalization," *London Journal of Research in Science: Natural and Formal*, 2019.
- [61] Z. Çamurdan and M. C. Ganiz, "Machine learning based electricity demand forecasting," pp. 412–417, 2017.
- [62] S.-I. Kuan, J. Kim, O.-H. Kwon, and H.-J. Song, "Canopy k-means combined collaborative filtering using rmse-minimization," in *2022 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2022, pp. 31–34.
- [63] U. Khair, H. Fahmi, S. Al Hakim, and R. Rahim, "Forecasting error calculation with mean absolute deviation and mean absolute percentage error," in *journal of physics: conference series*, vol. 930, no. 1. IOP Publishing, 2017, p. 012002.
- [64] J. Stoer and C. Witzgall, *Convexity and optimization in finite dimensions I*. Springer Science & Business Media, 2012, vol. 163.
- [65] A. Cambini and L. Martein, *Generalized convexity and optimization: Theory and applications*. Springer Science & Business Media, 2008, vol. 616.
- [66] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [67] W. F. Eddy, "A new convex hull algorithm for planar sets," *ACM Transactions on Mathematical Software (TOMS)*, vol. 3, no. 4, pp. 398–403, 1977.
- [68] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International journal of forecasting*, vol. 22, no. 3, pp. 443–473, 2006.
- [69] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965. [Online]. Available: <http://www.jstor.org/stable/2333709>
- [70] A. Kolmogorov, "Sulla determinazione empirica di una legge didistribuzione," *Giorn Dell'inst Ital Degli Att*, vol. 4, pp. 89–91, 1933. [Online]. Available: <https://cir.nii.ac.jp/crid/1572824501049496320>
- [71] H. W. Lilliefors, "On the kolmogorov-smirnov test for normality with mean and variance unknown," *Journal of the American statistical Association*, vol. 62, no. 318, pp. 399–402, 1967.
- [72] D. Kwiatkowski, P. C. Phillips, P. Schmidt, and Y. Shin, "Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?" *Journal of Econometrics*, vol. 54, no. 1, pp. 159–178, 1992. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/030440769290104Y>
- [73] G. M. Ljung and G. E. Box, "On a measure of lack of fit in time series models," *Biometrika*, vol. 65, no. 2, pp. 297–303, 1978.
- [74] T. S. Breusch and A. R. Pagan, "A simple test for heteroscedasticity and random coefficient variation," *Econometrica: Journal of the econometric society*, pp. 1287–1294, 1979.
- [75] M. J. Heule and O. Kullmann, "The science of brute force," *Communications of the ACM*, vol. 60, no. 8, pp. 70–79, 2017.
- [76] D. C. Karnopp, "Random search techniques for optimization problems," *Automatica*, vol. 1, no. 2-3, pp. 111–121, 1963.
- [77] C. Corte and V. Vapnik, "Support vector machines," *Machine Learning*, vol. 20, pp. 273–297, 1995.
- [78] L. Breiman, "Random forests machine learning, vol. 45," *Oct*, 2001.
- [79] A. Y. Ng *et al.*, "Preventing" overfitting" of cross-validation data," in *ICML*, vol. 97. Citeseer, 1997, pp. 245–253.