UNIVERSIDAD SANTO TOMAS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA

Comunicaciones en Estadística
Marzo 2025, Vol. 1, No. 1, pp. 1–23

PREDICCIÓN DEL PRECIO DE LA POSICIÓN DE LAS EXPORTACIONES DE FLORES EN COLOMBIA USANDO MODELOS DE APRENDIZAJE DE MÁQUINA

Prediction of Flowers Export Position Prices in Colombia Using Machine Leraning Models

Sebastián Hernández[a] Javier Sierra[b]

Resumen

Las exportaciones en los países en surgimiento económico, como Colombia, son importantes para los distintos procesos que dinamizan la economía. Por ejemplo, se registró que el año 2020 fue un año de recesión, donde la economía mundial cayó un 4.4 %. Observamos que las economías más avanzadas cayeron un 5.8 %, mientras que las economías en surgimiento de América Latina experimentaron una caída del 8.1 %. En Colombia, a raíz de la pandemia, el producto interno bruto cayó un 7 %, lo cual representa una caída del 3.8 % con relación al PIB del año 2019, las restricciones impuestas por la pandemia genero un impacto sobre el mercado de exportación de las flores el cual según el Ministerio de Agricultura y Desarrollo Rural de Colombia (2024) aporta a la económica nacional 130000 empleos rurales. Teniendo en cuenta estas fluctuaciones en los mercados internacionales, es esencial desarrollar modelos estadísticos predictivos que permitan comprender cómo se comportarán los mercados que son referentes para Colombia. Es por esta razón que esta investigación tiene como objetivo desarrollar, mediante el uso de aprendizaje automático, modelos estadísticos como Red neuronal LSTM , Bosques Aleatorios y XGBoost, además de modelos estadísticos tradicionales como las series de tiempo, con el fin de predecir el precio por posición en la exportación de flores en Colombia.

***Palabras clave*:** Aprendizaje, Máquina, Modelos Estadísticos, Exportaciones, Colombia, Flores.

Abstract

Exports in emerging economies, such as Colombia, play a vital role in various processes that drive economic dynamics. For instance, 2020 was recorded as a year of recession, where the global economy contracted by 4.4 %. Advanced economies experienced a decline of 5.8 %, while emerging economies in Latin America faced a sharper contraction of 8.1 %. In Colombia, the GDP fell by 7 % due to the pandemic, representing a 3.8 % decrease compared to the GDP of 2019. The restrictions imposed during the pandemic had a significant impact on the flower export market, which, according to Ministerio de Agricultura y Desarrollo Rural de Colombia (2024), contributes to the national economy by generating 130,000 rural jobs. Considering these fluctuations in international markets, it is essential to develop predictive statistical models to understand how key markets for Colombia will behave. Therefore, this research aims to develop predictive models using machine learning techniques, such as Neural Networks LSTM, Random Forests, and XGBoost, as well as traditional statistical models like time series analysis, to forecast export prices by tariff position in Colombia's flower market.

***Keywords*:** Machine Learning, Statistical Models, Exports, Colombia, Flowers..

[a]Estudiante de Maestría en Estadísticas Aplicadas
[b]Docente

1. Introducción

Los avances en las ciencias de la computación y el aprendizaje de máquina se han vuelto una buena opción para los entes gubernamentales. La implementación de algoritmos predictivos, como los bosques aleatorios, máquinas de soporte vectorial, árboles de decisión, redes neuronales, entre otros, permite tomar mejores decisiones con soporte estadístico y matemático. Por ejemplo, la Universidad de Ingeniería y Administrativa de Envigado, Antioquia, realizó un algoritmo de aprendizaje profundo usando redes neuronales con el fin de estimar el potencial de algunos productos de consumo Administrativos (2020a).

Las exportaciones, según el boletín técnico de exportaciones del Departamento Administrativo Nacional de Estadística (DANE) (2019) , son parte fundamental del PIB nacional. Solo en el año 2019 se exportaron $3,357$ productos a 187 países desde Colombia, aportando $39,489$ millones de dólares. Los principales productos exportados desde nuestro país fueron aceites crudos de petróleo, hullas bituminosas, café sin tostar y sin descafeinar, y oro. A continuación, se presentan los principales países a los que se exporta.

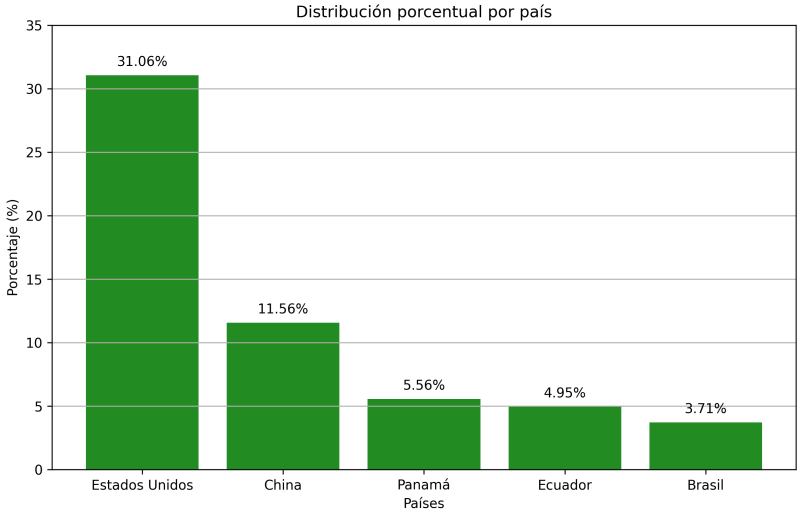

Figura 1: Principales países a los que Colombia exporta DANE.

En el caso de Colombia, una de las exportaciones más fuertes del país son las flores, las cuales, para el año 2004, ya representaban el 14 % del valor mundial de las exportaciones de este producto, ubicándose solo después de los Países Bajos. Para el año 2019, el mercado de flores en Colombia generó $1,460$ millones de dólares, mientras que, para el año 2021, entre los meses de enero y noviembre, representó $1,544$ millones de dólares y tuvo una incidencia en el **PIB** nacional de 0.49 %, según Departamento Administrativo Nacional de Estadística (DANE) (2004) y ProColombia (2021).

Con las mejoras tecnológicas en el procesamiento de información y los nuevos y mejorados modelos de machine learning , se vuelve indispensable la elaboración y aplicación de modelos estadísticos que intenten proporcionar una visión clara del panorama de los distintos sectores económicos , esto con el fin de anticiparse a las constantes fluctuaciones del mercado internacional.

Un ejemplo notable es el uso de redes neuronales y modelos como bosques aleatorios en la economía computacional. Debido a que estas herramientas han permitido el manejo de grandes cantidades de datos han proporcionado a los investigadores reducción del sobre ajuste de los resultados y eficiencia en el manejo de grandes volúmenes de datos como lo manifiesta Breiman (2001a) en su articulo. la implementación de estos

algoritmos han demostrado ser mas eficientes que los modelos estadísticos tradicionales a la hora de detectar patrones y tendencias en los datos , estos avances permiten a las distintas áreas del gobierno la elaboración de estrategias de mercado y nuevas políticas que vayan en sintonía de la generación de una estabilidad económica, Un ejemplo de esto son los algoritmos predictivos utilizados en Indonesia para predecir la cantidad de exportaciones que realizará ese país, empleando modelos híbridos de **ARIMA-LSTM** Abdullah et al. (2020). Otro caso es el cálculo del potencial de exportación de distintas empresas mediante técnicas de clasificación y predicción, como **K-Means** Silvaa et al. (2019), mientras que en los Estados Unidos, se destacó la importancia de predecir el precio del mercado del camarón para tomar decisiones gubernamentales orientadas a fomentar la mejora y desarrollo de este sector Khiem et al. (2021). Para este fin, se emplearon modelos de aprendizaje de máquina, como bosques aleatorios y Gradient Boosting Tree, Por último, en Colombia se han desarrollado algoritmos de aprendizaje de máquina y aprendizaje profundo para la estimación del potencial de exportación en el sector minero Administrativos (2020b) , mientras que otra investigación en Colombia realizo un estudio para analizar y comparar diferentes métodos de predicción para estimar los precios de venta de productos agrícolas en el departamento de Santander utilizando modelos como ARIMA y Redes Neuronales Artificiales Márquez González and Montaño Pérez (2018).

Esta investigación propone aplicar algoritmos estadísticos mediante el uso de técnicas de aprendizaje automático para predecir el valor de la posición de la exportación de las flores en el mercado colombiano. Teniendo en cuenta que el valor de la posición o FOB (Free On Board) es un termino utilizado en el comercio internacional que indica que el vendedor cumple con su obligación de entrega cuando la mercancía ha sido cargada a bordo del buque designado en el puerto de embarque convenido. A partir de ese momento, todos los costos y riesgos de pérdida o daño de la mercancía son transferidos al comprador como lo indica of Commerce (2020) , este valor es de suma importancia debido a que las empresas pueden planificar de manera más eficiente los costos asociados a sus procesos de importación y exportación, como lo manifiesta Estevadeordal and Suominen (2004), es fundamental considerar lo relacionado con la posición arancelaria para lograr un mejor entendimiento del flujo del comercio.

El Objetivo de esta investigación es Construir modelos como Bosque Aleatorio, XGBoost (Aumento de Gradiente Extremo), y Red Neuronal LSTM (Memoria a Largo y Corto Plazo), además de implementar modelos clásicos como las series de tiempo ARIMA (Modelo Autorregresivo de Media Móvil), con el propósito de predecir el valor de la posición de exportación de flores en Colombia.

2. Modelos de Aprendizaje de Maquina.

2.1. Bosque Aleatorio y XGBOOST.

Para hablar de árboles aleatorios, primero se debe definir el algoritmo árbol de decisión que, según Flores Rodríguez (2021), consiste en nodos que forman un árbol dirigido con un nodo raíz que no tiene un nodo predecesor. Todos los demás nodos tienen exactamente un nodo predecesor. Un nodo con arcos de salida se conoce como nodo interno o nodo prueba, mientras que los nodos que no poseen sucesores son nodos hoja o nodos de decisión.

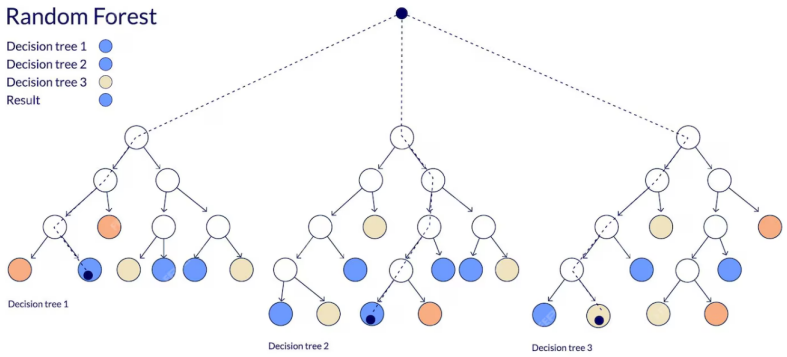

Figura 2: Composición de un bosque aleatorio. Cardellino (2021).

Mientras que los bosques aleatorios, según Jesus (2020), son un algoritmo tipo ensamble basado en árboles de decisión. Los métodos de ensamble combinan varios algoritmos de aprendizaje de máquina para mejorar la exactitud de los resultados. En este caso, en lugar de usar varios algoritmos diferentes sobre los mismos datos de entrenamiento, se utiliza el mismo algoritmo en diferentes subconjuntos aleatorios de los datos de entrenamiento. Por esta razón, el ensamble de bosque aleatorio emplea varios árboles de decisión, y cada uno de estos árboles es alimentado con un subconjunto aleatorio de datos de entrenamiento.

Por otra parte, el algoritmo XGBoost, como lo señala Espinosa-zúñiga (2020), consiste en un ensamblado secuencial de árboles, los cuales se agregan de manera secuencial con el objetivo de aprender del resultado de los árboles previos y corregir el error producido por estos, hasta que dicho error ya no pueda reducirse más. Este algoritmo ha sido utilizado en investigaciones como la de Dairu and Shilong (2021), en la que se desarrolló un modelo de árbol de decisión con XGBoost para predecir las ventas de uno de los principales almacenes minoristas en Estados Unidos, Walmart.

Matemáticamente los bosques aleatorios se rigen por la siguiente formula:

$$\hat{f}(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} f_b(\mathbf{x})$$

donde:

- $\hat{f}(\mathbf{x})$ es la predicción final del modelo. - B es el número total de árboles en el bosque. - $f_b(\mathbf{x})$ es la predicción del b-ésimo árbol de decisión.

como menciona Breiman (2001b) cada árbol f_b se constituye utilizando un subconjunto de los datos de entrenamiento mediante un muestreo con remplazo (bootstrap sampling). Además, en cada nodo del árbol, solo se considera un subconjunto aleatorio de características al momento de realizar la mejor división, lo que introduce diversidad entre los árboles y mejora la generalización del modelo.

mientras que para el modelo XGBOOST matemáticamente se representa de la siguiente forma segun Chen and Guestrin (2016).

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t)$$

donde:

- $\Omega(f_t)$ es un término de regularización que penaliza la complejidad del modelo.

Para mejorar la eficiencia computacional, la función de pérdida se aproxima mediante una expansión de segundo orden de Taylor:

$$L^{(t)} \approx \sum_{i=1}^{n} \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t)$$

donde:

$$g_i = \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}, \quad h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial \left(\hat{y}_i^{(t-1)} \right)^2}$$

Este enfoque permite un ajuste eficiente del modelo y mejora la precisión en la predicción.

2.2. Series de Tiempo.

Las series de tiempo son secuencias de datos recolectados y ordenados cronológicamente, utilizadas en diversas disciplinas para analizar y predecir comportamientos futuros basados en datos históricos. Estas series son fundamentales en campos como la economía, la meteorología, la medicina y las ciencias sociales, ya que permiten identificar patrones temporales y realizar previsiones que facilitan la toma de decisiones.

Las series de tiempo presentan desafíos únicos debido a la naturaleza temporal de los datos. Las observaciones no son independientes entre sí, ya que están ordenadas cronológicamente, lo que introduce autocorrelación. Además, las series pueden mostrar variaciones estacionales, tendencias a largo plazo y ciclos, así como ruido o fluctuaciones aleatorias. Estas características hacen necesario el uso de métodos específicos para su análisis y modelización (Box et al., 2015).

2.3. Series de Tiempo ARIMA (modelo auto regresivo de media móvil) .

ARIMA es un modelo de series de tiempo utilizado para predecir valores futuros basados en datos históricos. Combina tres componentes principales: la parte autorregresiva (AR), que considera la relación entre una observación y un número de observaciones anteriores; la parte integrada (I), que aplica diferencias a los datos para hacerlos estacionarios; y la parte de media móvil (MA), que modela la relación entre una observación y un número de errores de predicción anteriores. Es útil para series de tiempo no estacionales y requiere que los datos sean estacionarios o se transformen para serlo (Box et al., 2015).

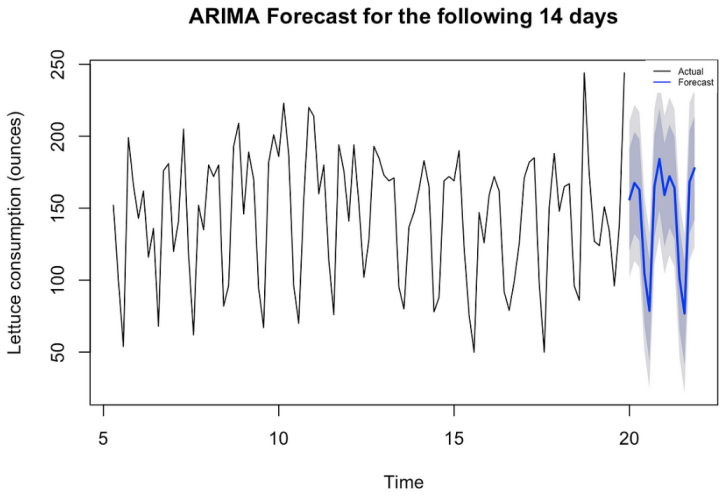

Figura 3: Serie de tiempo predicción con modelo ARIMA ejemplo. GeeksforGeeks (2020)

matemáticamente las serie de tiempo ARIMA según Khan et al. (2021) combina tres componentes principales:

- **AR (AutoRegresivo)**: Representado por un proceso de orden p, donde la variable depende de sus valores pasados:
$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \epsilon_t$$
- **I (Integrado)**: Indica el número de diferenciaciones necesarias (d) para hacer la serie estacionaria:
$$Y'_t = Y_t - Y_{t-1}, \quad (\text{para } d = 1)$$
- **MA (Media Móvil)**: Modelo de orden q que expresa la serie como una combinación de errores pasados:
$$Y_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q}$$

La ecuación general del modelo ARIMA(p, d, q) se expresa como:

$$\Phi_p(B)(1 - B)^d Y_t = \Theta_q(B)\epsilon_t$$

donde B es el operador de rezago, $\Phi_p(B)$ y $\Theta_q(B)$ son los polinomios de los coeficientes del modelo AR y MA respectivamente, y ϵ_t es un ruido blanco.

2.4. Red neuronal recurrente (RNN) LSTM (memoria largo corto plazo).

En el contexto del aprendizaje automático, las redes neuronales se definen como sistemas computacionales formados por capas de unidades interconectadas (neuronas artificiales) que, a través del ajuste iterativo de sus parámetros mediante algoritmos de optimización (como el gradiente descendente), son capaces de aproximar funciones complejas y aprender representaciones jerárquicas de los datos Schmidhuber (2015), las redes neuronales o neuroredes están motivadas por el objetivo de modelar el procesamiento de la información en sistemas nerviosos biológicos, especialmente por la forma de funcionamiento del cerebro humano, que es completamente distinta al funcionamiento de una computadora digital convencional.

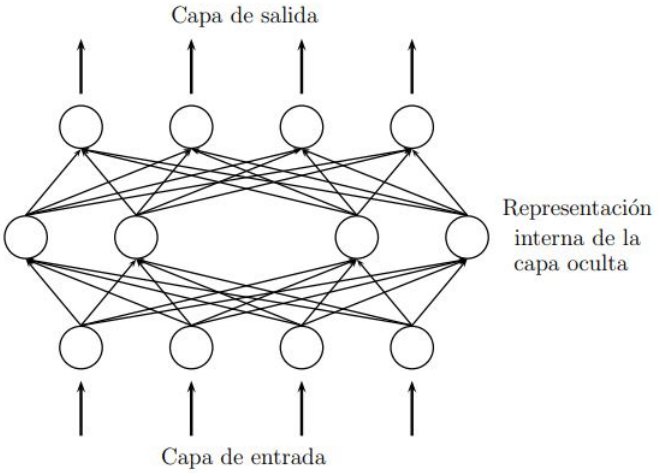

Figura 4: Red neuronal Santana (2006).

Las redes neuronales LSTM son un tipo de redes neuronales recurrentes (RNN) que basan su aprendizaje en la secuencialidad del tiempo como variable principal para desarrollar sus predicciones. Como menciona Hochreiter (1997), las redes neuronales LSTM (Long Short-Term Memory) se definen como una arquitectura de red neuronal recurrente diseñada específicamente para superar los problemas de retropropagación del error a lo largo del tiempo.

matemáticamente las redes neuronales LSTM tienen una celda de memoria c_t que es controlada por tres puertas que segun Hochreiter (1997) se comportan de la siguiente forma:

- **Puerta de olvido**: Decide qué información eliminar de la celda de memoria.

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$

- **Puerta de entrada**: Controla qué nueva información agregar a la celda de memoria.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

- **Puerta de salida**: Decide qué información de la celda de memoria se usa para la salida.

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$

La actualización de la celda de memoria y del estado oculto se define como:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

$$h_t = o_t \odot \tanh(c_t)$$

donde $\sigma(\cdot)$ es la función sigmoide, $\odot$ denota la multiplicación elemento a elemento, W y U son matrices de pesos, y b son los sesgos.

3. Metodología.

La medición de la calidad de los algoritmos de aprendizaje de maquina diseñados es sumamente importante, dado que contar con herramientas que nos permitan decidir qué modelo es mejor que otro fomenta la mejora continua en los procesos de aprendizaje de máquinas. Como métrica de evaluación de los algoritmos desarrollados se usó el error cuadrático medio (RMSE), que es un sistema de medición utilizado para medir las diferencias entre los valores de la variable y la predicción del algoritmo Freedman et al. (1993). En una regresión, este error se presenta como la distancia vertical que hay desde el punto predicho hasta la recta de regresión.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Fórmula 1: Ecuación del RMSE. Freedman et al. (1993)

Mientras que para el modelo de serie de tiempo se tuvieron en cuenta las mediciones AIC (Criterio de Información de Akaike) y BIC (Criterio de Información Bayesiano), con el fin de evaluar y comparar la calidad del ajuste entre diferentes configuraciones del modelo. El AIC estima la cantidad de información perdida por un modelo, mientras que el BIC incorpora además un castigo más fuerte por el número de parámetros. En ambos casos, un menor valor se traduce en un mejor equilibrio entre ajuste y simplicidad

Los datos utilizados en esta investigación fueron suministrados por Ministerio de Industria, Comercio y Turismo. (2022). La base de datos está constituida por las exportaciones realizadas por Colombia desde 2012 hasta 2019, para el desarrollo de la investigación se filtro la variable "POSARA3", que contiene los códigos de posiciones arancelarias de exportación de todos los productos, Esto con el fin de quedarse únicamente con la base de datos que contiene información sobre flores. Los datos arancelarios que identifican a las flores son:

Código Arancelario	**Descripción**
603110000	Rosas, frescas
603121000	Capullos de claveles frescos
603129000	Otros claveles frescos
603199010	Crisantemos frescos
603141000	Orquídeas frescas
603149000	Otras flores frescas que no sean orquídeas
603150000	Gladiolos frescos
603192000	Flor de lirio fresca
603193000	Flor de gerbera fresca
603194000	Flores de alstroemeria frescas
603199090	Otras flores frescas
603900000	Partes de flores frescas o secas
603130000	Capullos de tulipán frescos
603191000	Flores de lisianthus frescas

Tabla 1: Descripción de los códigos arancelarios de flores en Colombia

Con la implementación de este filtro se procedió a realizar un análisis de valores atípicos sobre la variable respuesta, "valor de la posición FOB-DODL3", dando como resultado final una base de datos de $830,330$ registros de exportaciones de flores entre los años 2012 y 2020.

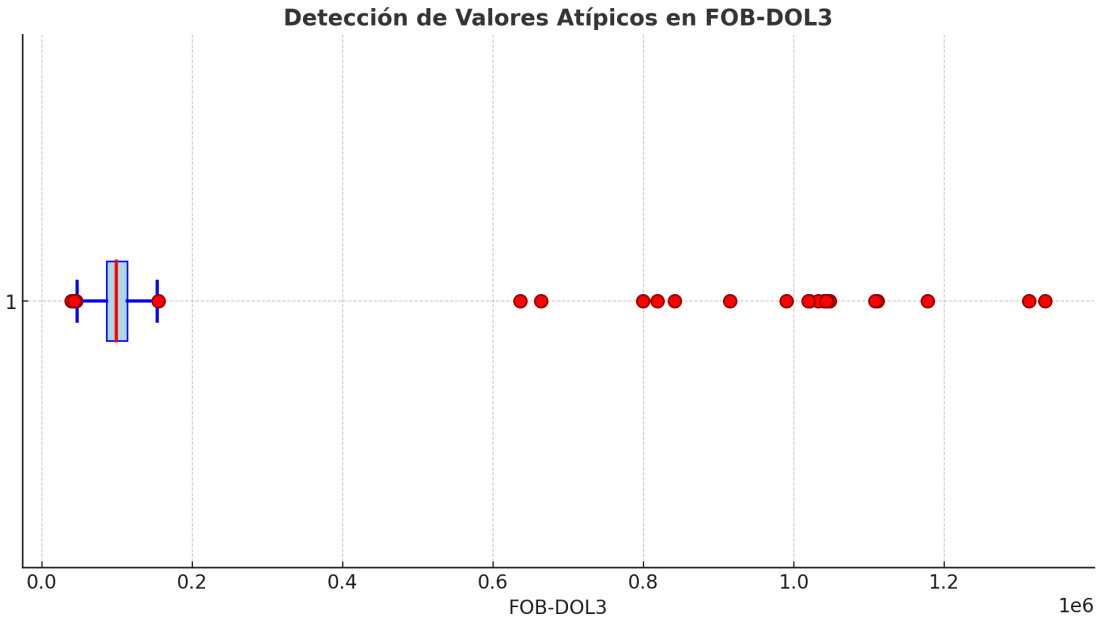

Figura 5: Detección de valores atípicos, gráfica creada por el investigador.

Por ultimo se procedió a excluir variables como la razón social del exportador, dirección del exportador, número de NIT del declarante, razón social del declarante, nombre o razón social del importador, y dirección del país de destino (dirección del importador), con el fin de mantener la anonimización de la información del exportador. resultando en una base de datos final de $830,330$ registros de exportaciones de flores con 48 variables.

A continuación se presenta el análisis de correlación.

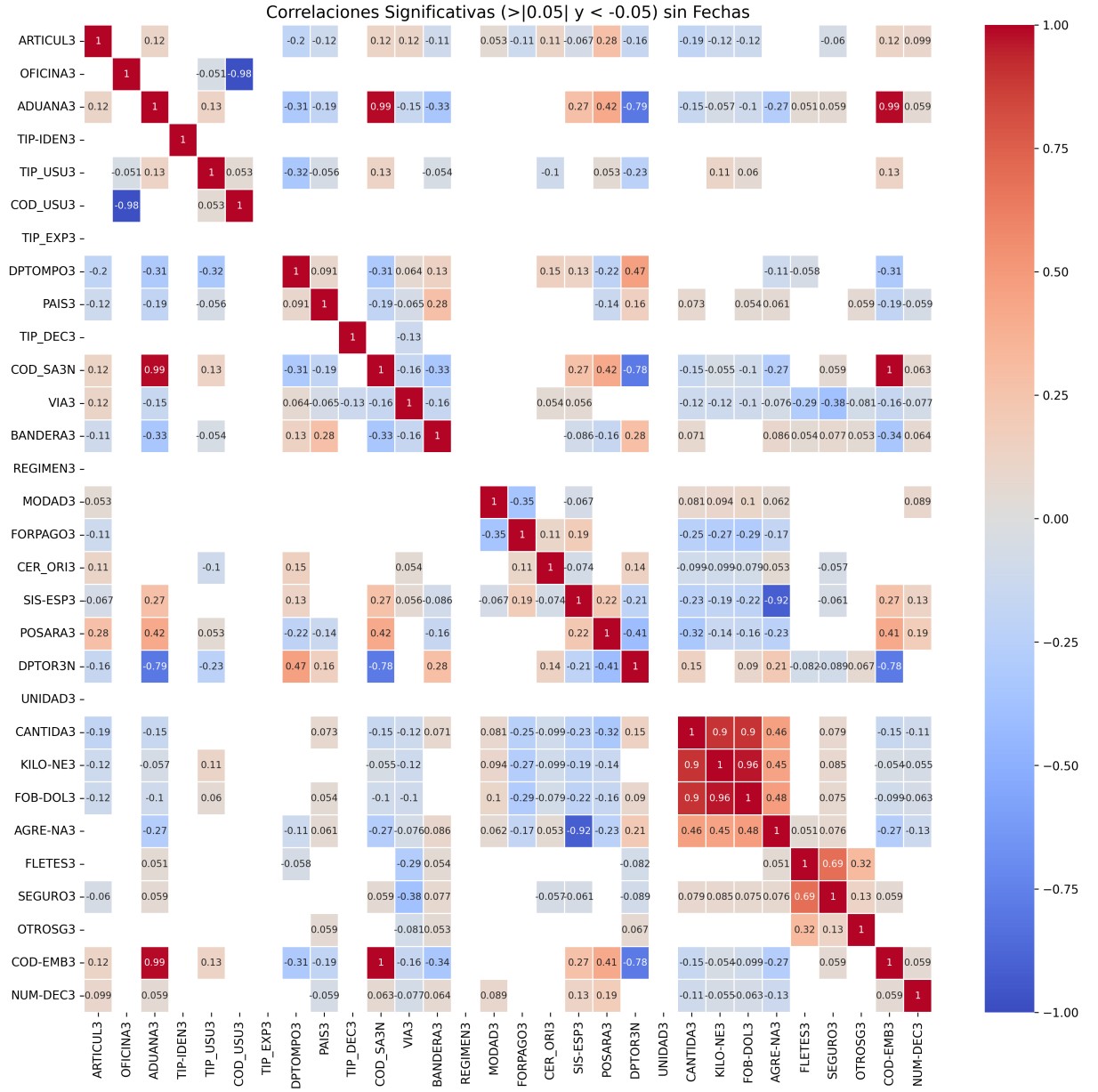

Figura 6: Detección de valores atípicos.

El resultado de esta evaluación tiene un rango de -1 a 1, considerando una relación lineal nula o baja entre 0 y 0.3, media entre 0.3 y 0.5, y alta cuando es mayor a 0.5. Además, en el análisis desarrollado se utilizó la correlación de Spearman, la cual, según el mismo autor, se emplea cuando las variables no siguen una distribución normal o son variables ordinales, lo que permite realizar un análisis más robusto.

Para garantizar la capacidad de generalización de los modelos de aprendizaje de maquina y evitar la fuga de datos, se implementó un esquema de separación temporal. De este total, $651,227$ registros ($78.43\,\%$) fueron destinados al entrenamiento de los modelos, abarcando los datos desde 2012 hasta 2018, mientras que los $179,103$ registros restantes ($21.57\,\%$) se reservaron exclusivamente para la evaluación del rendimiento en

datos no vistos correspondientes al año 2019.

para el modelo de serie de tiempo se implemento un análisis con frecuencia semanal en el modelado ARIMA, siguiendo recomendaciones como las de Ludvigson et al. (2014), quienes aplicaron modelos ARIMA series semanales de reclamos de seguros por desempleo en EE.UU. La elección se justifica por la presencia de menor ruido que en series diarias y una mejor representación de dinámicas operativas reales.

Además, el conjunto de datos cuenta con un total de 21 variables, las cuales son:

Variable	**Descripción**
ARTICUL3	Número total de artículos que trae la declaración. Debe ser consecutivo
ADUANA3	Código de la aduana
FORPAGO3	Forma de pago (1= Con Reintegro, 2= Sin Reintegro)
CANTIDA3	Cantidad de unidades de la posición
AGRE-NA3	Total valor agregado nacional de la posición
TIP_USU3	Tipo de usuario (39 opciones posibles)
CER_ORI3	Tipo de certificado de origen (8 opciones)
KILO-BR3	Total kilos brutos de la posición
SEGURO3	Valor seguro de la posición
PAIS3	Código país destino (numérico)
SIS-ESP3	Sistemas especiales (1= Sí, 2= No)
COD-EMB3	Código de administración de aduana de embarque
COD_SA3N	Código lugar de salida (numérico)
POSARA3	Posición Arancelaria
VIA3	Código de la vía de transporte
FECH-DE3	Fecha de la declaración de exportación definitiva (año, mes, día)
NUM-DEC3	Número de declaración de exportación definitiva

Tabla 2: Descripción de las Variables utilizadas para el desarrollo de los modelos propuestos

4. Resultados.

A continuación, se presentan los hiperparámetros utilizados en los modelos propuestos, así como los resultados obtenidos durante la ejecución de los mismos, destacando aquellos con el mejor desempeño.

4.1. Serie de Tiempo ARMA

Para desarrollar el modelo ARMA, primero se realizo una interpolación de datos para llevar los datos que se presentan de manera diaria a semanal , esto se realizo con la función **resample** de la librería pandas.

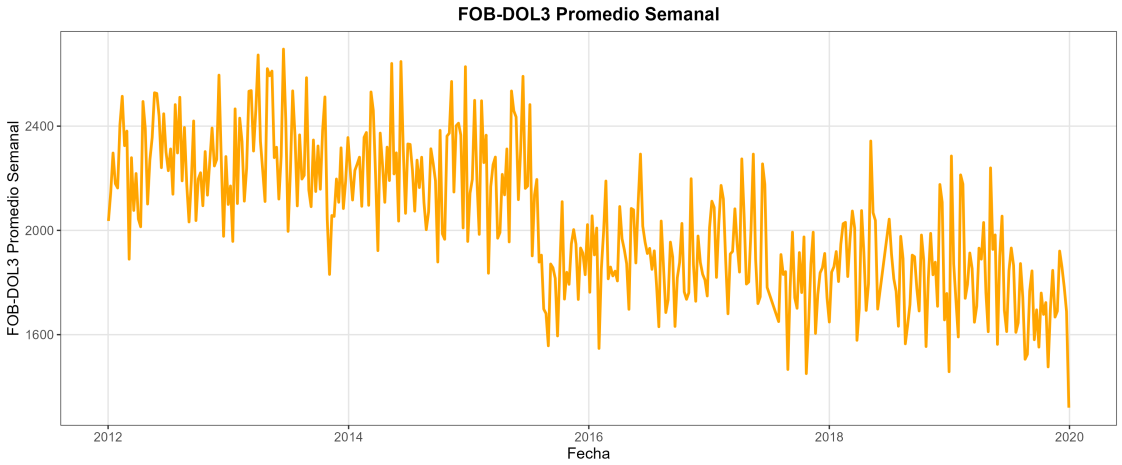

Figura 7: Comportamiento semanal promedio de la variable FOB-DOL3.

se realiza a continuación un análisis de estacionaridad mediante la prueba de Dickey-Fuller. Esta prueba, según Lizarazu-Alanez and Villaseñor-Alva, 2007, evalúa si una serie de tiempo tiene una tendencia persistente lo que implicaría una no estacionaridad o si al contrario los datos fluctúan alrededor de una media constante lo que implicaría una estacionaridad de la serie.

Estadístico de prueba ADF	-4.752176
Valor p	0.01
Valores Críticos	
1 %	-3.430
5 %	-2.862
10 %	-2.567

Tabla 3: Resultados de la Prueba Dickey-Fuller Aumentada (ADF) aplicada a la serie FOB-DOL3 promedio semanal.

Como el estadístico ADF es menor que los tres valores críticos, podemos concluir que los datos presentan estacionaridad. A continuación se ejecutan múltiples modelos ARMA ya que la serie de tiempo al ser estacionaria no requiere diferenciación I, por lo cual se genera una grilla de hiperparámetros que se presenta a continuación.

Hiperparámetro	**Descripción**	**Rango de Valores**
p	Orden de autorregresión	0, 1, 2, 3, 4, 5
q	Orden de media móvil	0, 1, 2, 3, 4, 5

Tabla 4: Hiperparámetros utilizados en el modelo ARIMA.

Modelo	MSE	MAE	AIC	BIC
ARMA(3,5)	22577.25	110.76	5284.49	5324.43
ARMA(4,1)	28627.62	119.71	5302.59	5330.55
ARMA(4,2)	28558.02	114.75	5302.62	5334.57
ARMA(5,0)	28325.28	117.93	5299.85	5327.81
ARMA(5,1)	28232.91	116.53	5301.08	5333.03
ARMA(5,2)	27905.64	110.28	5302.49	5338.43
ARMA(5,3)	27643.52	114.91	5303.66	5343.60

Tabla 5: Métricas de error y criterios de información para modelos ARMA evaluados sobre la serie FOB-DOL3.

Modelo	Ljung-Box_p	Jarque-Bera_p	ARCH_p	Durbin-Watson	DW_p
ARMA(3,5)	0.7208	0.5235	0.0523	1.9949	0.4797
ARMA(4,1)	0.2914	0.8853	0.2023	2.0070	0.5280
ARMA(4,2)	0.6240	0.8499	0.1038	2.0152	0.5608
ARMA(5,0)	0.5051	0.8267	0.1633	1.9783	0.4140
ARMA(5,1)	0.6022	0.8236	0.2201	1.9946	0.4783
ARMA(5,2)	0.6888	0.8002	0.1012	2.0050	0.5199
ARMA(5,3)	0.7616	0.8388	0.2354	1.9997	0.4987

Tabla 6: Resultados de las pruebas diagnósticas aplicadas a los residuos de los modelos ARMA.

Prueba	Resultado	Límite para cumplir	Interpretación (dummy)
Ljung-Box	p = 0.7208	p >0.05	No hay patrones repetidos en los errores. El modelo capturó bien la estructura temporal. Ljung and Box (1978)
Jarque-Bera	p = 0.5235	p >0.05	Los errores tienen forma normal (campana). Se pueden usar herramientas estadísticas. Jarque and Bera (1987)
ARCH	p = 0.0523	p >0.05	No hay cambios bruscos en la varianza de los errores. El modelo es estable en el tiempo. Engle (1982)
Durbin-Watson	DW = 1.995, p = 0.4797	DW 2, p >0.05	Los errores son independientes entre sí. El modelo no deja correlaciones lineales. Durbin and Watson (1950)

Tabla 7: Pruebas diagnósticas aplicadas al modelo ARMA(3,5).

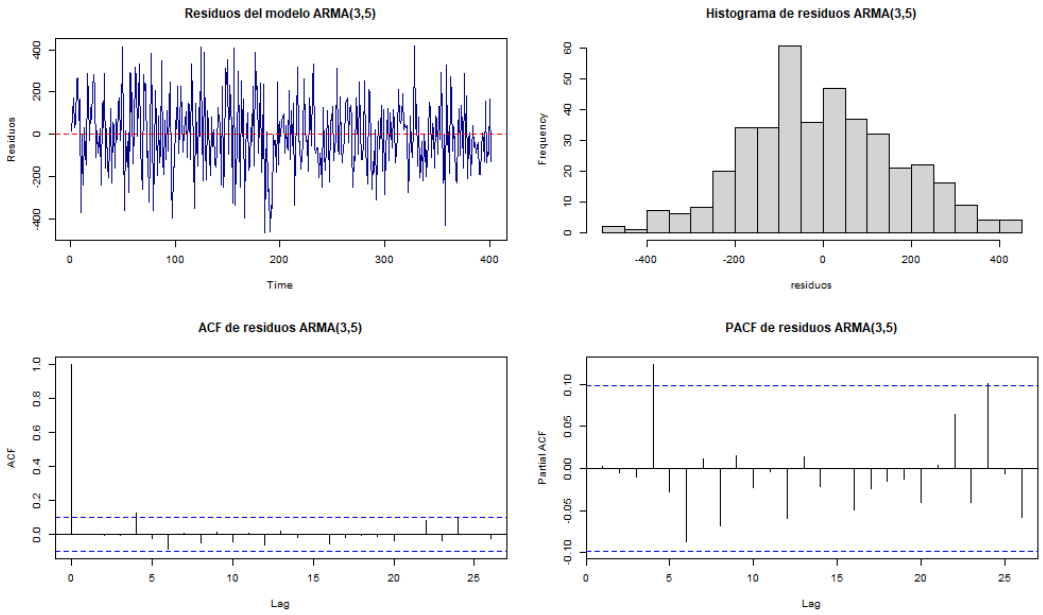

Figura 8: Diagnóstico gráfico de residuos para el modelo ARMA(3,5): serie de residuos, histograma, ACF y PACF.

Se evidencio que el mejor modelo para predecir el valor FOB-DOL3 semanal fue el ARMA(3,5) con un MAE de 110.76, un MSE de 22,577.25, y los menores valores de AIC (5284.49) y BIC (5324.43) entre los modelos evaluados.

$$\begin{aligned} y_t = \mu &+ \phi_1 y_{t-1} + \phi_2 y_{t-2} + \phi_3 y_{t-3} + \varepsilon_t \\ &+ \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \theta_3 \varepsilon_{t-3} + \theta_4 \varepsilon_{t-4} + \theta_5 \varepsilon_{t-5} \end{aligned} \tag{1}$$

remplazando los coeficientes obtenemos:

$$\begin{aligned} y_t = 2013.2834 &+ 1.2374\, y_{t-1} - 1.2428\, y_{t-2} + 0.9829\, y_{t-3} \\ &- 1.0902\, \varepsilon_{t-1} + 1.1844\, \varepsilon_{t-2} - 0.7732\, \varepsilon_{t-3} \\ &+ 0.0140\, \varepsilon_{t-4} + 0.0491\, \varepsilon_{t-5} + \varepsilon_t \end{aligned} \tag{2}$$

4.2. Red neuronal recurrente (RNN) LSTM.

Una LSTM, según Staudemeyer and Morris (2019), es una arquitectura de red neuronal recurrente diseñada para abordar los problemas de explosión y desvanecimiento del gradiente en secuencias largas. Para comprender el modelo de red neuronal recurrente, es importante mencionar que los datos analizados incluyeron características como el día, el mes, el día de la semana, el día del año y los fines de semana, con el fin de captar patrones estacionales. A partir de esto, se procedió a crear una función de secuencia temporal, la cual, según Staudemeyer and Morris (2019), consiste en un conjunto de datos ordenados en el tiempo que permite a las redes aprender dependencias a largo plazo en secuencias, lo cual es crucial para problemas como la predicción de series temporales.

Teniendo en cuenta que Staudemeyer and Morris (2019) menciona que el uso de capas LSTM apiladas permite aumentar la capacidad del modelo, donde cada capa sucesiva recibe la salida secuencial completa

de la capa anterior, se decidieron crear dos capas LSTM: la primera con **return sequences** (True) para obtener las secuencias completas, y la segunda con **return sequences** (False) para obtener solo el último valor de la secuencia. Posteriormente, se aplicó la técnica Dropout, que según Staudemeyer and Morris (2019) es una técnica de regularización que evita el sobreajuste apagando de manera aleatoria algunas neuronas durante el entrenamiento. La salida del modelo consistió en una sola capa densa con un nodo para predecir el valor numérico de la posición FOB-DOL3. En el proceso de ejecución, se implementó la técnica de parada temprana, que según Barone et al. (2017), ayuda a detener el entrenamiento cuando el modelo comienza a sobreajustarse. Para entrenar el modelo, se construyó una grilla de hiperparámetros, los cuales, según Barone et al. (2017), deben estar constituidos de la siguiente forma:

Hiperparámetro	**Rango**	**Descripción**
Dropout	0.2 - 0.4	Regularización
Aprendizaje	10^{-6} - 1 (0.01)	Estabilidad en gradientes
Batch Size	1 - 256 (32)	Balance eficiencia/convergencia
Épocas	20 - 100	Early Stopping para evitar sobre ajuste

Tabla 8: Rangos recomendados de hiperparámetros según Barone et al. (2017)

Con base en esta referencia, se procedió a construir la grilla de hiperparámetros, la cual se utilizó para probar múltiples modelos. A continuación, se presenta la grilla de hiperparámetros utilizada.

Units	**Batch Size**	**Epochs**	**Learning Rate**	**Dropout Rate**
200	64	30	0.0001	0.3
200	64	50	0.0001	0.4
300	64	50	0.0001	0.2
400	64	100	0.0001	0.3
300	32	50	0.0005	0.2
400	32	100	0.0005	0.3
200	64	50	0.0001	0.4
300	128	50	0.0001	0.2
400	128	100	0.0001	0.3
250	32	50	0.0003	0.2
300	64	100	0.0003	0.25
350	128	50	0.0001	0.4
150	16	100	0.0005	0.2
200	256	50	0.001	0.3
50	256	20	0.001	0.3
100	256	20	0.001	0.3
150	256	30	0.0005	0.4
200	256	30	0.0001	0.4
50	64	50	0.001	0.3
100	64	50	0.001	0.4
150	64	50	0.0005	0.4
200	64	50	0.0001	0.4
50	128	50	0.001	0.4
100	128	50	0.001	0.4
150	128	50	0.0005	0.4
200	128	50	0.0005	0.4

Tabla 9: Grilla de hiperparámetros para el modelo LSTM generada por Investigador..

A continuación, se presentan los resultados más significativos:

Unidades	Batch Size	Epochs	R^2	MSE	MAE
200	64	30	0.0432	5212084.08	1654.32
200	64	50	0.0431	5212632.25	1681.39

Tabla 10: Resultados más significativos de la ejecución LSTM

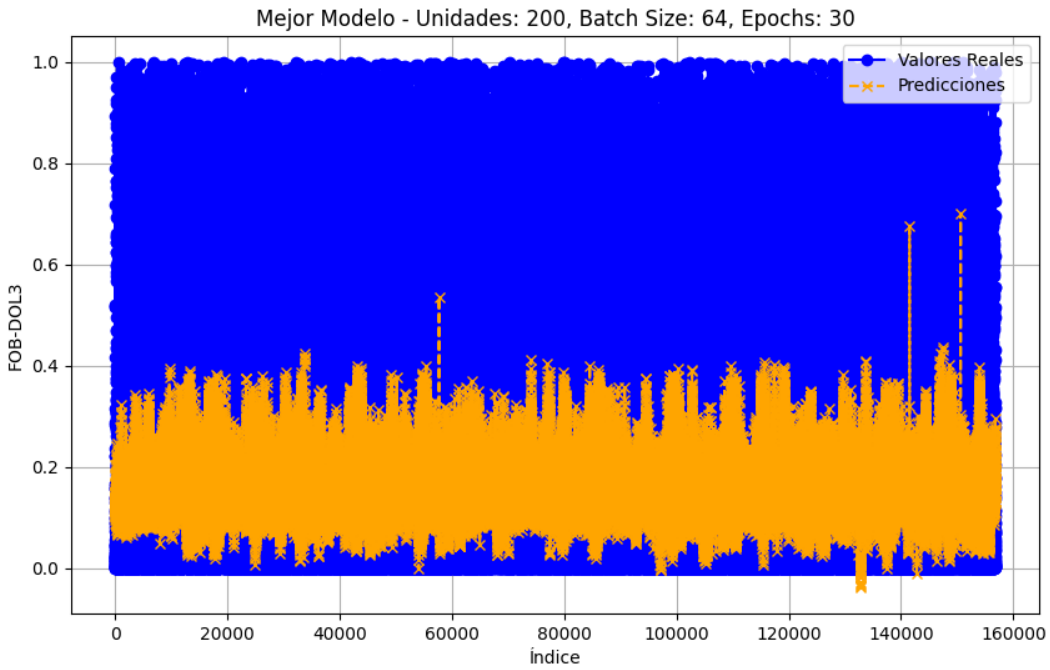

Figura 9: Resultados ejecución LSTM comparación valores reales vs valores de predicción, gráfica elaborada por investigador.

Se evidencia en los resultados de R^2 que el modelo no explica ni captura la variabilidad de los datos de FOB-DOL3. Sin embargo, los datos obtenidos a partir del MSE con los hiperparámetros propuestos indican que el modelo se ajusta adecuadamente a los datos, aunque presenta un error elevado en la predicción.

4.3. Bosques Aleatorios

Las grillas de hiperparámetros son fundamentales a la hora de construir modelos de inteligencia artificial pues permite con ello la optimización de los resultados. Para realizar el modelo Bosque Aleatorio el autor Sumathi et al. (2020) recomienda la construcción de una grilla de hiperparámetros la cual se presenta a continuación:

Con esta información se procedió a construir una grilla de hiperparámetros.

Como resultado del entrenamiento y ejecución del modelo, se obtuvieron los siguientes resultados:

Hiperparámetro	Valores sugeridos	Descripción
n_estimators	100 a 2000	Cantidad de árboles, para mejora precisión
max_depth	10, 20, None	Profundidad máxima con el fin de evitar sobreajuste
min_samples_split	2, 5, 10	Muestras mínimas aceptada para dividir nodo
min_samples_leaf	1, 2, 5	Muestras mínimas en hoja
max_features	sqrt, log2	parámetros a considerar en cada división

Tabla 11: Rangos de hiperparámetros recomendados para Random Forest por autor Sumathi et al. (2020)

Hiperparámetro	Valores	Descripción
n_estimators	100, 200	Número de árboles en el bosque
max_depth	10, 20, None	Profundidad máxima de los árboles
min_samples_split	2, 5	Muestras mínimas necesarias para dividir un nodo
min_samples_leaf	1, 2	Muestras mínimas en una hoja

Tabla 12: Grilla de Hiperparámetros para Random Forest construida por investigador.

Parámetro	Valor
n_estimators	200
max_depth	None
min_samples_split	2
min_samples_leaf	1
Resultados	
Coeficiente de determinación (R^2)	0.9435
Error cuadrático medio (MSE)	307361.05
Error absoluto medio (MAE)	248.266

Tabla 13: Resultados de ejecución del modelo Random Forest

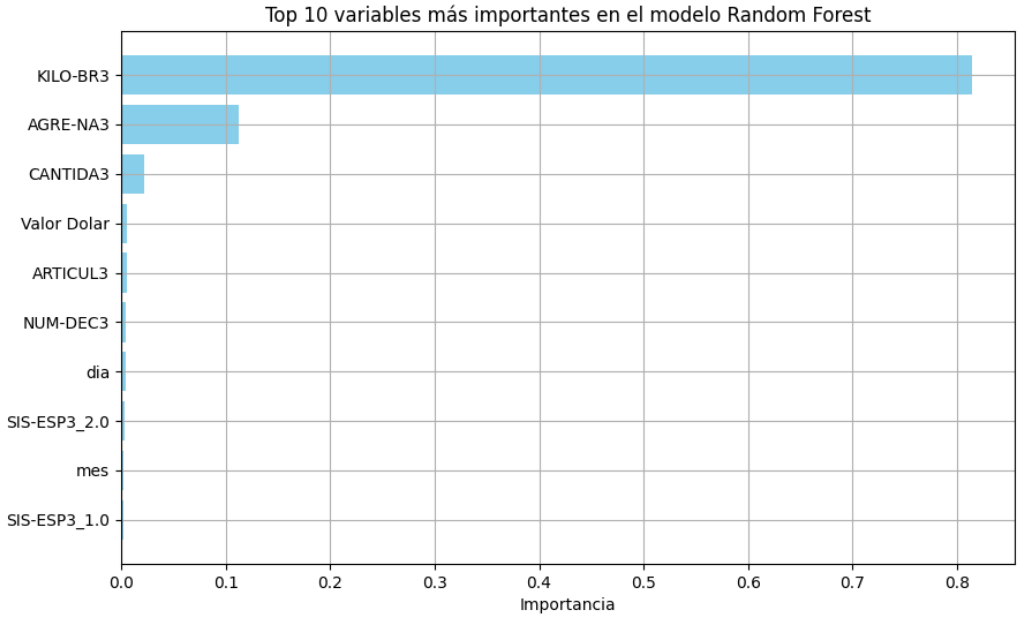

Figura 10: Top 10 variables mas importantes utilizadas por el modelo Random Forest, gráfica elaborada por investigador.

Se observa que la variable con mayor peso al momento de realizar el entrenamiento del modelo Random Forest fue "Kilos Brutos", seguida por el "Total Valor Agregado Nacional de la posición AGRE-NA3", y, en tercer lugar, la Cantidad de Elementos Constituidos en la posición".

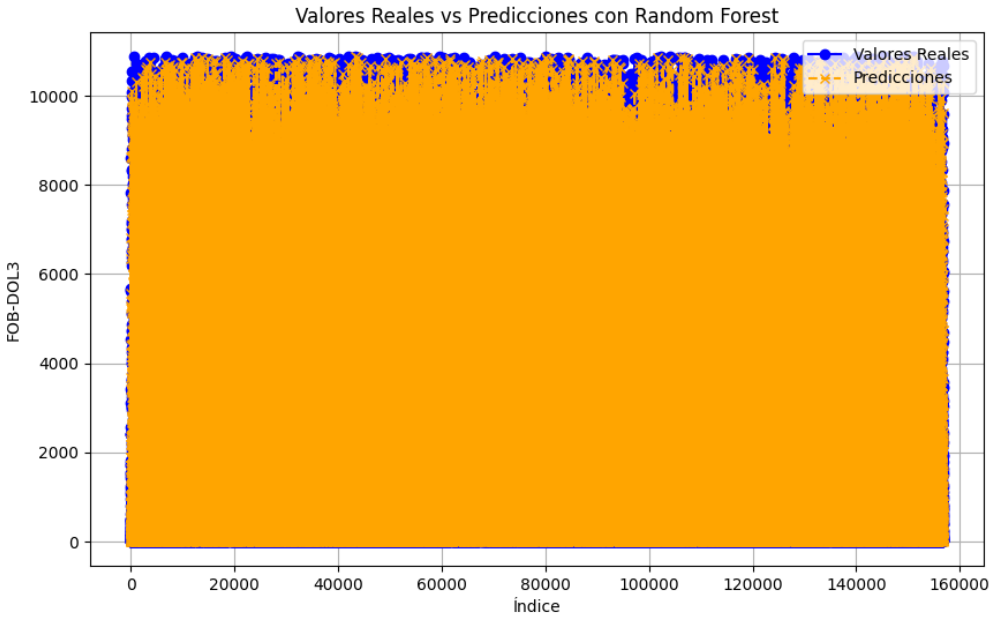

Figura 11: Comparación resultados valores reales y valores de predicción , gráfica elaborada por investigador.

Con los resultados obtenidos mediante la medición de R^2, se puede afirmar que el modelo de Random Forest con los hiperparámetros establecidos explica aproximadamente el 94.35 % de la variabilidad de los datos. Además, al observar el valor relativamente bajo del MSE, podemos concluir que el modelo está realizando predicciones cercanas al valor real.

Estos resultados reflejan un alto nivel de precisión en las predicciones. Al realizar la comparación con estudios previos, encontramos como por ejemplo Yuliana et al. (2024) en su investigación Hyperparameter Optimization of Random Forest Algorithm to Enhance Performance Metric Evaluation of 5G Coverage Prediction con la cual lograron mejoras significativas en la precisión de predicciones de cobertura 5G al optimizar los hiperparámetros del algoritmo de Bosques Aleatorios, tras la optimización de resultados la investigación en mención alcanzo un RMSE de 0.66.

4.4. XGBOOST Boosting por gradientes extremos.

Para ejecutar el modelo XGBOOST, de manera similar a lo realizado con el Random Forest, se procedió a construir una grilla de hiperparámetros, los cuales, según Putatunda and Rama (2019), son los presentados a continuación:

Hiperparámetro	Descripción	Valores recomendados
`n_estimators`	Número de árboles en el modelo.	100 - 300
`learning_rate`	Tasa de aprendizaje del modelo.	0.1, 0.2
`max_depth`	Profundidad máxima de cada árbol	5, 7, 9
`subsample`	Proporción de muestras usadas en cada árbol.	0.8, 1.0
`colsample_bytree`	Proporción de columnas usadas para cada árbol.	0.8, 1.0
`gamma`	Pérdida mínima para crear partición adicional en un nodo.	0, 0.1, 0.2
`min_child_weight`	Peso mínimo de una hoja, previene nodos con pocas observaciones.	1, 5, 10
`reg_alpha`	Regularización L1, reduce la complejidad del modelo.	0, 0.01, 0.1
`reg_lambda`	Regularización L2, ayuda a evitar sobre ajuste.	1, 1.5, 2

Tabla 14: Hiperparámetros recomendados para XGBoost por Putatunda and Rama (2019)

Con base en estas recomendaciones, se procedió a construir la siguiente grilla de hiperparámetros.

Hiperparámetro	Valores en la grilla
`n_estimators`	[100, 200, 300]
`learning_rate`	[0.1, 0.2]
`max_depth`	[5, 7, 9]
`subsample`	[0.8, 1.0]
`colsample_bytree`	[0.8, 1.0]

Tabla 15: Grilla de hiperparámetros construida para XGBoost por investigador

Los resultados obtenidos fueron los siguientes:

Parámetro / Métrica	Valor
`colsample_bytree`	0.8
`learning_rate`	0.1
`max_depth`	9
`n_estimators`	300
`subsample`	1.0
Coeficiente de determinación (R^2)	0.9428
Error cuadrático medio (MSE)	311555.33
Error absoluto medio (MAE)	261.46

Tabla 16: Resultados del mejor modelo XGBoost ejecutado.

Hiperparámetro	Valores en la grilla
`n_estimators`	[100, 200, 300]
`learning_rate`	[0.1, 0.2]
`max_depth`	[5, 7, 9]
`subsample`	[0.8, 1.0]
`colsample_bytree`	[0.8, 1.0]

Tabla 17: Grilla de hiperparámetros para XGBoost construida por investigador.

El mejor resultado obtenido de la ejecución de XGBoost, evaluado mediante el R^2, indica que este modelo puede explicar aproximadamente el 94.28 % de la variabilidad de los datos de FOB-DOL3. Además, al observar los resultados del MSE, podemos concluir que el modelo está realizando predicciones cercanas al valor real.

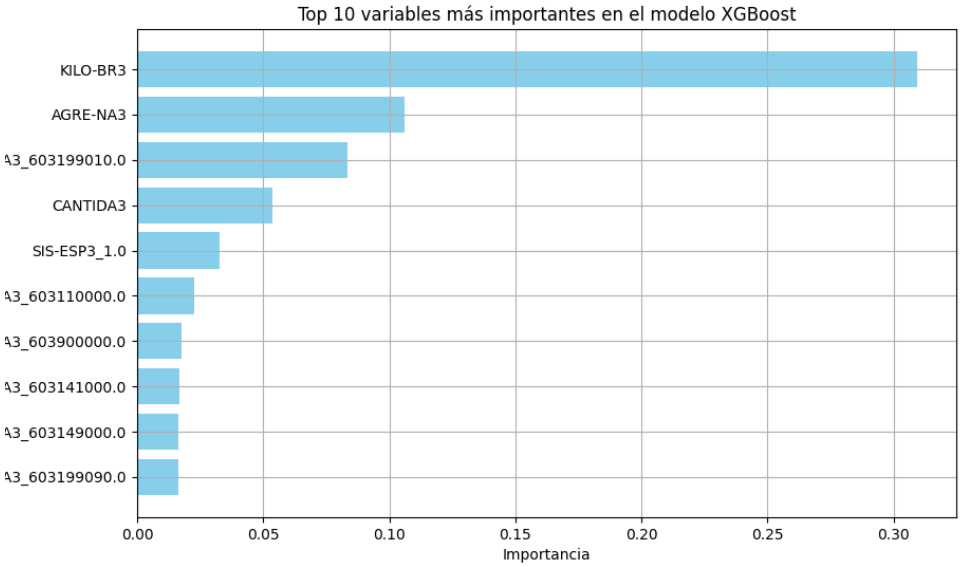

Figura 12: Variables más importantes para el modelo XGBOOST, gráfica elaborada por Investigador.

Se observa que, al igual que en el modelo de Random Forest, para el modelo XGBoost, la variable más importante es "Kilos Brutos", aunque con una menor proporción e influencia. Esta es seguida por la variable AGRE-NA3.

5. Conclusiones.

Modelo	**MAE**	**AIC**	**BIC**
ARMA(3,5)	110.76	5284.49	5324.43

Tabla 18: Desempeño del modelo ARMA(3,5) según MAE, AIC y BIC.

Modelo	**MSE**	**MAE**	**R^2**
LSTM (Mejor resultado)	5,212,084.08	1,654.32	0.0432
Bosques Aleatorios	307,361.05	248.27	0.9435
XGBoost	311,555.33	261.46	0.9428

Tabla 19: Comparación de modelos de Machine Learning según MSE, MAE y R^2.

El modelo de machine learning que obtuvo los mejores resultados en términos de precisión y capacidad explicativa fue el **Random Forest**, configurado con los siguientes hiperparámetros: *n-estimators = 200*,

min-samples-split = 2 y *min-samples-leaf = 1*. Este modelo logró explicar un notable 94.35 % de la varianza presente en los datos, evidenciando su capacidad para capturar patrones significativos dentro del conjunto analizado. Además, el error cuadrático medio (MSE) alcanzado por este modelo fue de 307,361, un resultado altamente aceptable en el contexto del problema de predicción abordado. Cabe destacar que el modelo fue entrenado y evaluado utilizando conjuntos de datos diferentes, con el fin de evitar el sobreajuste. Estos resultados refuerzan la eficacia de los modelos basados en árboles y técnicas de ensamble, como *Random Forest*, para manejar datos complejos y variables interdependientes.

Por otra parte, el modelo de series de tiempo **ARMA(3,5)** presentó un desempeño aceptable dentro del enfoque clásico, logrando un error absoluto medio (MAE) de 110.76. Si bien no alcanzó los niveles de precisión observados en los modelos de Machine Learning por su configuración lineal que no le permite observar interelaciones de variables, su fortaleza radica en la capacidad para capturar la estructura temporal lineal de la serie. Además, superó satisfactoriamente las principales pruebas diagnósticas —como Ljung-Box, Jarque-Bera y Durbin-Watson— lo que respalda la validez estadística del modelo y su correcta especificación.

Sin embargo, la **prueba ARCH presentó un valor p = 0.0523**, muy cercano al umbral crítico de 0.05. Este resultado sugiere una posible presencia de *heterocedasticidad condicional* (cambios en la varianza del error), por lo que se recomienda que, en trabajos futuros, se explore el uso de modelos como **ARCH** o **GARCH**, los cuales están diseñados específicamente para modelar series financieras o económicas donde la varianza no es constante a lo largo del tiempo.

Referencias

K. Abdullah, C. Haruna, I. Muhammad, K. Asfandyar, B. J. Iqbal, A. Muhammad, H. M. F., and A. Hanan. Forecasting electricity consumption based on machine learning to improve performance: A case study for the organization of petroleum exporting countries (OPEC). *Computers and Electrical Engineering*, 86, 2020. ISSN 00457906. doi: 10.1016/j.compeleceng.2020.106737.

I. Administrativos. MODELO DEEP LEARNING PARA LA ESTIMACIÓN DEL SANTIAGO MOLINA AGUDELO JUANITA VILLEGAS RAMIREZ Trabajo de grado para optar al título de : Ingenieros Administrativos Juan Alejandro Peña Palacio. 2020a.

I. Administrativos. MODELO DEEP LEARNING PARA LA ESTIMACIÓN DEL SANTIAGO MOLINA AGUDELO JUANITA VILLEGAS RAMIREZ Trabajo de grado para optar al título de : Ingenieros Administrativos Juan Alejandro Peña Palacio. 2020b.

A. V. M. Barone, B. Haddow, U. Germann, and R. Sennrich. Regularization techniques for fine-tuning in neural machine translation. *arXiv preprint arXiv:1707.09920*, 2017.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001a. doi: 10.1023/A:1010933404324.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001b.

F. Cardellino. Tutorial para un clasificador basado en bosques aleatorios: cómo utilizar algoritmos basados en árboles para el aprendizaje automático. *freeCodeCamp*, 2021. URL `https://www.freecodecamp.org/espanol/news/random-forest-classifier-tutorial-how-to-use-tree-based-algorithms-for-machine-learning/`. Consultado el 17 de noviembre de 2024.

T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016. doi: 10.1145/2939672.2939785.

X. Dairu and Z. Shilong. Machine Learning Model for Sales Forecasting by Using XGBoost. *2021 IEEE International Conference on Consumer Electronics and Computer Engineering, ICCECE 2021*, (Iccece): 480–483, 2021. doi: 10.1109/ICCECE51280.2021.9342304.

DANE. Departamento nacional de estadistica.

Departamento Administrativo Nacional de Estadística (DANE). Microdatos de exportaciones de flores en Colombia, 2004. URL `https://microdatos.dane.gov.co/index.php/catalog/555`. Consultado el 9 de marzo de 2025.

Departamento Administrativo Nacional de Estadística (DANE). Boletín Técnico de Exportaciones. Technical report, 2019. URL `{https://www.dane.gov.co/index.php/estadisticas-por-tema/comercio-internacional/exportaciones}`.

J. Durbin and G. S. Watson. Testing for serial correlation in least squares regression: I. *Biometrika*, 37(3/4): 409–428, 1950. doi: 10.2307/2332391.

R. F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987–1007, 1982. doi: 10.2307/1912773.

J. Espinosa-zúñiga. Aplicación de algoritmos Random Forest y XGBoost en una base de solicitudes de tarjetas de crédito Application of Random Forest and XGBoost algorithms based on a credit card applications database. XXI(número 3):16, 2020.

A. Estevadeordal and K. Suominen. Rules of origin: a world map and trade effects. 2004.

C. Flores Rodríguez. Generación de árboles de decisión usando un algoritmo inspirado en la Física. 1:5–16, 2021.

D. Freedman, R. Pisani, and R. Purves. *ESTADISTICA*. Antoni Bosh editor, Barcelona - España, segunda ed edition, 1993.

GeeksforGeeks. Time series analysis using arima model in r programming. *GeeksforGeeks*, 2020. URL `https://www.geeksforgeeks.org/time-series-analysis-using-arima-model-in-r-programming/`. Consultado el 17 de noviembre de 2024.

S. Hochreiter. Long short-term memory. *Neural Computation MIT-Press*, 1997.

C. M. Jarque and A. K. Bera. A test for normality of observations and regression residuals. *International Statistical Review*, 55(2):163–172, 1987. doi: 10.2307/1403192.

B. Jesus. *Machine Learning y Deep Learning Usando Python, Scikit y Keras*. Edicion de la U, Bogota - Colombia, primera ed edition, 2020.

M. A. Khan et al. A comprehensive review of the application of arima model in time series forecasting. *Journal of Statistical Theory and Practice*, 15(4):1–20, 2021. doi: 10.1007/s42519-021-00268-9.

N. M. Khiem, Y. Takahashi, K. T. P. Dong, H. Yasuma, and N. Kimura. Predicting the price of Vietnamese shrimp products exported to the US market using machine learning. *Fisheries Science*, 87(3):411–423, 2021. ISSN 14442906. URL `https://doi.org/10.1007/s12562-021-01498-6`.

E. Lizarazu-Alanez and J. A. Villaseñor-Alva. Efectos de rompimientos bajo la hipótesis nula de la prueba dickey-fuller para raíz unitaria. *Agrociencia*, 41(2):193–203, 2007.

G. M. Ljung and G. E. P. Box. On a measure of lack of fit in time series models. *Biometrika*, 65(2):297–303, 1978. doi: 10.1093/biomet/65.2.297.

S. Ludvigson, S. Ng, et al. Dynamics of unemployment insurance claims: An arima–garch analysis. *Federal Reserve Bank of New York Staff Reports*, (701), 2014. URL https://www.newyorkfed.org/medialibrary/media/research/staff_reports/sr701.pdf.

Ministerio de Agricultura y Desarrollo Rural de Colombia. Colombia se expande a nuevos mercados de flores, 2024. URL https://www.minagricultura.gov.co/noticias/Paginas/Colombia-se-expande-a-nuevos-mercados-flores.aspx. Consultado el 17 de noviembre de 2024.

Ministerio de Industria, Comercio y Turismo. MINCIT, 2022. URL https://www.mincit.gov.co/.

J. D. Márquez González and L. V. Montaño Pérez. Comparación de métodos de predicción para el pronóstico de precios de venta de productos agrícolas en santander. *Repositorio UNAB*, 2018. URL https://repository.unab.edu.co/handle/20.500.12749/21921.

I. C. of Commerce. *Incoterms 2020: ICC Rules for the Use of Domestic and International Trade Terms*. ICC Publishing, Paris, France, 2020. ISBN 978-92-842-0511-0.

ProColombia. Exportaciones de flores colombianas llegan a la cifra más alta en su historia, 2021. URL https://procolombia.co/sala-de-prensa/noticias/exportaciones-de-flores-colombianas-llegan-la-cifra-mas-alta-en-su-historia. Consultado el 9 de marzo de 2025.

S. Putatunda and K. Rama. A modified bayesian optimization based hyper-parameter tuning approach for extreme gradient boosting. In *2019 Fifteenth International Conference on Information Processing (ICINPRO)*, pages 1–6. IEEE, 2019.

J. C. Santana. Predicción de series temporales con redes neuronales: una aplicación a la inflación colombiana. *Revista Colombiana de Estadística*, 29(1):77–92, 2006.

J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

J. Silvaa, J. R. Borréb, A. P. P. neres Castillo, L. Castrod, and N. Varela. Integration of data mining classification techniques and ensemble learning for predicting the export potential of a company. *Procedia Computer Science*, 151(2018):1194–1200, 2019. ISSN 18770509. URL https://doi.org/10.1016/j.procs.2019.04.171.

R. C. Staudemeyer and E. R. Morris. Understanding lstm–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*, 2019.

B. Sumathi et al. Grid search tuning of hyperparameters in random forest classifier for customer feedback sentiment prediction. *International Journal of Advanced Computer Science and Applications*, 11(9), 2020.

H. Yuliana, S. Basuki, M. R. Hidayat, A. Charisma, and H. Vidyaningtyas. Hyperparameter optimization of random forest algorithm to enhance performance metric evaluation of 5g coverage prediction. *Buletin Pos dan Telekomunikasi*, 22(1):75–90, 2024. URL https://doaj.org/article/6385bf8e80b74f5eaccaf7dcec3cbfb8.