



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Inteligencia artificial y autoaprendizaje para la mejora de modelos predictivos: caso de estudio con el índice S&P500

Samuel Murillo Ospina

Universidad Nacional de Colombia
Facultad de Minas, Área Curricular de Ingeniería de Sistemas e Informática
Medellín, Colombia
2024

Inteligencia artificial y autoaprendizaje para la mejora de modelos predictivos: caso de estudio con el índice S&P500

Samuel Murillo Ospina

Tesis presentada como requisito parcial para optar al título de:

Magister en Ingeniería – Analítica

Director (a):

María Constanza Torres Madroño

(Ph.D. Computing and Information Sciences and Engineering)

Universidad Nacional de Colombia

Facultad de Minas, Área Curricular de Ingeniería de Sistemas e Informática

Medellín, Colombia

2024

A mis mentores y a mi familia,

La búsqueda del conocimiento y la mejora constante son fundamentales en el viaje hacia la comprensión de la realidad y la innovación científica. En este trabajo enfocado al uso de Inteligencia Artificial y Autoaprendizaje para la mejora de modelos predictivos, me gustaría expresar mi profundo agradecimiento a todos aquellos que han compartido conmigo su apoyo y sabiduría en este camino.

Agradezco a mi familia por su apoyo incondicional y confiar en mis decisiones para llevar a cabo este proyecto de vida académico. Su amor y comprensión han sido mi mayor fortaleza.

Agradezco a los excelentes maestros que tuve en el transcurso de mi maestría, por ser mi puente entre el aprendizaje y el conocimiento científico, y por compartir amablemente sus valiosas experiencias y sugerirme una trayectoria de formación acorde con mis metas.

Agradezco a los pioneros y visionarios en el campo de la inteligencia artificial, cuyas contribuciones facilitaron mi comprensión de las temáticas afines, les doy las gracias por su visión y dedicación a la excelencia.

Que este trabajo sirva como un pequeño tributo a su influencia y un testimonio del compromiso de las personas con el avance del conocimiento y la creación de un futuro más brillante.

Con gratitud y admiración,

Samuel Murillo Ospina

Resumen

Título: Inteligencia artificial y autoaprendizaje para la mejora de modelos predictivos: caso de estudio con el índice S&P500

La realización de pronósticos precisos es crucial en la sociedad y en la ciencia, debido a que permiten anticipar tendencias y comprender el impacto de las variables consideradas en el comportamiento futuro de un fenómeno de interés. Este trabajo aborda el desafío de mejorar el desempeño de los modelos predictivos. Para ello, se llevó a cabo una investigación cuantitativa con el objetivo de evaluar la predicción del precio de cierre del índice bursátil S&P500 mediante la aplicación de un modelo Transformer, integrando técnicas de autoaprendizaje.

Acorde con lo anterior, se realizó la implementación y optimización del modelo, logrando un error porcentual promedio del 1,52% y un máximo del 4,04% en la predicción de los próximos 60 valores del S&P500. Para mejorar el desempeño se incorporó autoaprendizaje basado en la distancia de Mahalanobis. Así se mejoró el rendimiento al reducir el error a 1,22%, que corresponde a una disminución de 19,67% con respecto al valor antes de incorporar el método mencionado.

Comparado con métodos tradicionales, el Transformer demostró una precisión superior, capturando patrones complejos y fue posible ajustar sus resultados automáticamente a través de la distancia de Mahalanobis. Los hallazgos resaltan la importancia de la investigación continua en la predicción de series de tiempo utilizando inteligencia artificial en combinación con técnicas que permitan ajustar los modelos de acuerdo con información histórica y los registros de desempeño.

Palabras clave: inteligencia artificial, autoaprendizaje, pronósticos, desempeño, autoajuste.

Abstract

Title: Artificial intelligence and self-learning for enhancing predictive models: a case study with the S&P500 index

The realization of accurate forecasts is crucial in society and science, as they allow to anticipate trends and understand the impact of variables considered on the future behavior of an interest phenomenon. This work takes the challenge of improving the performance of predictive models. To achieve this, a quantitative research was developed with the objective of evaluating the prediction of the closing price of the S&P500 stock index using a Transformer model, integrating self-learning techniques.

In accordance with the above, the implementation and optimization of the model were carried out, achieving an average percentage error of 1.52% and a maximum of 4.04% in the prediction of the next 60 values of the S&P500. To improve performance, self-learning based on Mahalanobis distance was incorporated. Thus, performance was improved by reducing the error to 1.22%, which corresponds to a 19.67% decrease compared to the value before incorporating the mentioned method.

Compared to traditional methods, Transformer demonstrated superior accuracy, capturing complex patterns, and it was possible to automatically adjust its results through the Mahalanobis distance. The findings highlight the importance of continuous research in time series prediction using artificial intelligence in combination with techniques that allow models to be adjusted according to historical information and performance records.

Keywords: artificial intelligence, self-learning, forecasts, performance, self-adjustment.

Contenido

	Pág.
Resumen	I
Lista de figuras	V
Lista de tablas	VI
Introducción	1
1. Capítulo 1. Descripción del problema	5
1.1 Delimitación del problema	5
1.1.1 Hipótesis.....	6
1.2 Justificación.....	6
1.3 Objetivos	8
1.3.1 Objetivo general.....	8
1.3.2 Objetivos específicos	9
2. Capítulo 2. Revisión sistemática de literatura	11
2.1 Tipo de revisión.....	11
2.2 Protocolo de revisión.....	12
2.2.1 Criterios de inclusión y exclusión de artículos	12
2.3 Desarrollo de la revisión	13
2.4 Síntesis de la revisión de literatura	19
3. Capítulo 3. Marco teórico	23
3.1 Conceptos y teorías de soporte	23
3.1.1 ¿Qué son pronósticos?	23
3.1.2 Modelos predictivos	23
3.1.3 Inteligencia artificial	27
3.1.4 Aprendizaje automático	27
3.1.5 Autoaprendizaje.....	28
3.1.6 Optimizador RMSProp	29
3.1.7 Distancia de Mahalanobis.....	29
3.1.8 Automatización de modelos predictivos	30
3.1.9 Bases de datos	31
3.1.10 Modelo canónico de la arquitectura Transformer	32
3.1.11 Transformer para series de tiempo (Método de Pronóstico propuesto)	34
4. Capítulo 4. Metodología	37
4.1 Base de datos usada.....	37

4.2	Modelo Transformer.....	38
4.3	Evaluación de desempeño.....	40
4.3.1	Formulas.....	41
4.3.1.1	Error absoluto medio.....	41
4.3.1.2	Error cuadrático medio.....	41
4.3.1.3	Error porcentual.....	41
4.4	Modelo propuesto con autoaprendizaje.....	42
4.4.1	Formula de integración de autoaprendizaje en los pronósticos.....	43
5.	Capítulo 5. Resultados y discusión.....	47
5.1	Desarrollo de objetivo 1. Aplicar el modelo Transformer de series de tiempo para predecir el comportamiento del índice bursátil S&P500.....	47
5.1.1	Arquitectura resumida del modelo implementado.....	48
5.2	Desarrollo de objetivo 2. Incorporar técnicas de autoaprendizaje al modelo Transformer con el fin de incrementar su capacidad predictiva.....	54
5.3	Desarrollo de objetivo 3. Comparar los resultados obtenidos con los de otros enfoques de predicción del S&P500, identificando sus ventajas y limitaciones.....	60
6.	Conclusiones y recomendaciones.....	63
6.1	Conclusiones.....	63
6.2	Recomendaciones.....	65
	Bibliografía.....	75

Lista de figuras

	Pág.
Figura 2-1: Producción de documentos por año en la temática de interés.	14
Figura 2-2: Gráfico de producción de documentos por país.	15
Figura 2-3: Número de documentos por tipo.	16
Figura 2-4: Número de documentos por área temática.	16
Figura 3-1: Comparación entre 5 modelos de pronóstico comunes.	26
Figura 3-2: Diagrama de funcionamiento general de un modelo Transformer.	32
Figura 3-3: Esquema de funcionamiento de un modelo Transformer para pronóstico de series de tiempo.	34
Figura 4-1: Matriz de autoaprendizaje.	43
Figura 4-2: Diagrama del proceso de autoaprendizaje.	44
Figura 5-1: Dataframe de trabajo.	47
Figura 5-2: Flujo de la implementación posterior a la programación del modelo.	49
Figura 5-3: Gráfica de la función de pérdida.	50
Figura 5-4: Gráfica de valores reales vs pronósticos usando Transformer.	52
Figura 5-5: Gráfica de error porcentual para cada pronóstico respecto a las observaciones reales en valor absoluto.	53
Figura 5-6: Gráfica de error porcentual para cada pronóstico respecto a las observaciones reales sin valor absoluto.	54
Figura 5-7: Gráfica de valores reales vs pronósticos usando Transformer y Autoaprendizaje basado en distancia de Mahalanobis.	56
Figura 5-8: Gráfica de error porcentual para cada pronóstico con autoaprendizaje respecto a las observaciones reales en valor absoluto.	57
Figura 5-9: Gráfica de error porcentual para cada pronóstico con autoaprendizaje respecto a las observaciones reales sin valor absoluto.	57
Figura 5-10: Gráfica de error porcentual comparando Transformer y autoaprendizaje respecto a las observaciones reales en valor absoluto.	58
Figura 5-11: Gráfica de error porcentual comparando Transformer y autoaprendizaje respecto a las observaciones reales sin valor absoluto.	59

Lista de tablas

	Pág.
Tabla 2-1: Artículos seleccionados en la revisión de literatura.	18
Tabla 5-1: Valor de los pronósticos generados con Transformer.	51
Tabla 5-2: Valor de los pronósticos generados con Transformer y Autoaprendizaje. ...	55

Introducción

En la actualidad, la realización de pronósticos precisos se ha convertido en una actividad crucial en la sociedad y en cualquier campo de la ciencia, ya que proporcionan una visión aproximada del futuro de un fenómeno de interés con base en la medición y el análisis histórico de las variables que lo conforman. Para ello, es necesario construir ecosistemas analíticos que suplan las diferentes necesidades de información e implementación de modelos que anticipen tendencias, identifiquen patrones ocultos y permitan comprender mejor cómo los factores pasados y presentes afectan el futuro de una situación de interés.

Esto abre la puerta a la realización de innovaciones en la manera de automatizar procesos, la creación de modelos más precisos en la medición de variables, la mejora de los modelos de pronóstico y, en general, construir cimientos fuertes para entender la realidad desde los datos que son recogidos por los sistemas de información y las personas con base en el registro de fenómenos, sentimientos, conductas, comportamientos transaccionales, sociales y digitales.

Por otro lado, como mencionan Hanke & Wichern (2014, pp. 3, 469, 474), la efectividad de los diferentes métodos de pronóstico depende del horizonte de tiempo escogido y la temporalidad de los datos (diarios, mensuales, anuales, etc.), y no necesariamente los métodos avanzados o estadísticamente complejos brindan los mejores

resultados, debe haber un equilibrio y una configuración rigurosa de cada método, además de datos de calidad y que se actualicen constantemente, al igual que garantizar un monitoreo constante y usar herramientas que permitan controlar el ciclo de vida de los modelos analíticos.

Adicionalmente, es necesario que los modelos predictivos se ajusten de alguna manera a los cambios del entorno y los errores que surgen en sus ejecuciones, con el fin de mejorar los resultados y sacar el mayor provecho a la nueva información que se va obteniendo al realizar el seguimiento de las métricas desempeño utilizadas. Esto es análogo al proceso de aprendizaje que realizan los seres humanos, con el conocimiento de errores frecuentes se genera una nueva respuesta con el fin de no cometer las mismas equivocaciones o acercarse al valor o acciones esperadas.

Por ejemplo, Sun-Hee, et al. (2013, p. 11), en un estudio sobre la aplicación de un método avanzado para análisis de series de tiempo llamado modelo de auto regresión coercitivo ajustado, comenta que en la realización de predicciones “una dirección más ambiciosa sería reajustar automáticamente las ecuaciones de parámetros y coeficientes”, puesto que los modelos funcionan con información de un momento o periodo de tiempo específicos y a medida que hay nuevas observaciones o patrones se pierde la capacidad de predecir con exactitud lo que sucederá, a la vez que los parámetros se alejan de la realidad.

De modo que para lidiar con el hecho de que los modelos se descalibran con el tiempo es necesario considerar las razones de que esto suceda. Entre los motivos se encuentra el modo como se realiza su entrenamiento. Esto es a partir de información en una temporalidad específica con un conjunto de registros determinado, y a medida que pasa el tiempo los resultados obtenidos pueden alejarse de la realidad puesto que son generados con información vieja. Más aun, las técnicas utilizadas para realizar pronósticos siguen en desarrollo, y no hay garantía de que un mismo método funcionará siempre, ya que los fenómenos cambian y el enfoque sigue siendo usar datos pasados para hacer inferencias sobre el futuro.

Por lo anterior, se evidencia una necesidad de desarrollar modelos predictivos que sean capaces de ajustarse a cambios, es decir, que se autoajusten de acuerdo con las

variaciones en su entorno o en las variables implicadas. En este contexto, el presente trabajo tiene el objetivo de evaluar la predicción del comportamiento del índice bursátil S&P500 mediante la aplicación del modelo de inteligencia artificial Transformer, integrando técnicas de autoaprendizaje.

El alcance de la investigación es generar resultados de pronóstico junto con métricas de desempeño que permitan comparar lo obtenido con estudios similares y al mismo tiempo exponer que contribución puede tener incorporar técnicas para ajustar automáticamente las predicciones sin necesidad de repetir en entrenamiento, al menos, en el corto plazo.

Como limitaciones se destaca el hecho de que ningún modelo predictivo es perfecto, y siempre existe algún porcentaje de error debido a factores no medidos en las variables de entrada y del entorno, incluyendo las propias limitaciones de los modelos de pronóstico. Se busca contribuir en el área incrementando el conocimiento existente en cuanto a la aplicación de técnicas complementarias para el uso de Transformer, explorando alternativas para afinar en el tiempo la calidad predictiva. Se presentan los hallazgos identificados y el uso de la similitud estadística entre observaciones para optimizar la precisión y adaptarse a los errores registrados.

1. Capítulo 1. Descripción del problema

1.1 Delimitación del problema

El problema de investigación identificado es la limitada capacidad de un modelo de pronóstico a medida que se incrementa su horizonte de tiempo, puesto que es necesario estimar o indagar por el comportamiento futuro de los parámetros que lo componen, además de que las variables incluidas pueden ser insuficientes para realizarlo con precisión o ser menos explicativas de lo que se considera.

Dicho problema puede materializarse en la realización de predicciones erróneas que pueden obstaculizar la toma de decisiones o según el área donde ocurra incluso afectar procesos industriales, científicos, médicos, meteorológicos o administrativos con un alto costo económico y humano. Por otro lado, los errores mencionados pueden estar relacionados con la naturaleza y modo de recolección de los datos, o debido a sesgos sistemáticos como asumir en un pronóstico que “N = Todos”, siendo N el tamaño de la muestra de estudio (Harford, 2014, p. 18).

Esta situación se aborda en el presente trabajo a través de un caso de estudio generando pronósticos para el índice S&P500 usando un modelo Transformer e integrando técnicas de autoaprendizaje para mejorar su capacidad predictiva.

La investigación es relevante en el contexto de la inteligencia artificial y la ciencia de datos porque se centra en mejorar el desempeño de los modelos de pronóstico, lo cual es crucial en la toma de decisiones. Se espera que los resultados de este estudio proporcionen una mayor comprensión de cómo se puede mejorar la precisión de los modelos Transformer integrando técnicas de autoaprendizaje.

De este modo se considera la siguiente pregunta: ¿Cómo se puede mejorar la predicción de series de tiempo con Transformer y la integración de técnicas de autoaprendizaje, y en qué medida respecto a otros estudios similares?

1.1.1 Hipótesis

H_a = El pronóstico mejora su precisión respecto a las observaciones reales con la incorporación de autoaprendizaje.

Vs

H_0 = El pronóstico no mejora su precisión respecto a las observaciones reales con la incorporación de autoaprendizaje.

1.2 Justificación

La principal razón que motiva el desarrollo del presente trabajo es la necesidad de aplicar acciones para contrarrestar la limitada capacidad de un modelo de pronóstico a medida que se incrementa su horizonte de tiempo respecto al último dato utilizado para el entrenamiento, puesto que es necesario estimar o indagar por el comportamiento futuro de los parámetros que lo componen (Hanke & Wichern, 2014, p. 35). No obstante, también es posible hacer uso de los errores registrados para ajustar dichos parámetros o el resultado de modo que se reproduzca el comportamiento real del fenómeno en cuestión.

Así mismo Pérez, Cruz, Guatemala & Juárez (2018, pp. 101), mencionan que los pronósticos son importantes porque ayudan a los encargados de tomar decisiones a realizar juicios apoyados en datos acerca del corto, mediano y largo plazo que podrían afectar las operaciones, además de que tienen aplicaciones muy diversas desde la planificación de la producción hasta la estimación de las ventas para aplicar estrategias que mantengan la competitividad en las organizaciones.

En este sentido Mendieta & Sotolongo (2019, pp. 3), comentan que el pronóstico es una parte esencial de la ciencia moderna, y una prueba de la madurez de una disciplina científica radica en su capacidad para hacer pronósticos precisos.

De esta manera, la necesidad de realizar ajustes surge debido a que los modelos predictivos no piensan por sí solos y se descalibran conforme cambian los valores reales de los parámetros que se estiman para uso. Adicionalmente, diversos fenómenos estudiados por el ser humano tienen una naturaleza cambiante y pueden tener fluctuaciones estructurales importantes debido a acontecimientos sociales o de la naturaleza misma.

Otro aspecto a considerar es que cada modelo de pronóstico posee un desempeño determinado, que según Millea (2021, pp. 16 - 17), varía dependiendo de las técnicas usadas y la cantidad de datos considerados sobre la serie de tiempo en cuestión, y por esto implementar metodologías de corrección de errores puede dar mayor vida útil a los modelos predictivos emulando el proceso de aprendizaje del ser humano, el cual consiste en adaptarse a partir de la experiencia y anticiparse a lo que va a suceder evaluando el desenlace de situaciones similares en el pasado.

Con respecto al modelo de pronóstico a implementar para desarrollar el presente trabajo se escogió Transformer para series de tiempo. Trabajos previos han evidenciado la capacidad potencial de los Transformer para este tipo de aplicaciones. Por ejemplo, Kaeley, Qiao & Bagherzadeh (2023, p. 7), lograron una precisión (accuracy) del 89% al predecir el precio de apertura de las acciones de Facebook, Amazon, Apple, Netflix y Google, usando un periodo de retraso de 29 días, es decir, usando datos de n días hacia atrás.

Según Irabaruta (2021, cap. 5, p. 1), el modelo Transformer está por encima de otros métodos de machine learning como redes neuronales, bosque aleatorio y máquina de soporte vectorial, ya que estos permitieron obtener una precisión media de 80%, 73% y 81% al usarlos para pronóstico del precio de acciones de Walmart, J&P Morgan y Caterpillar respectivamente.

Por otra parte, la realización de pronósticos precisos toma relevancia debido a que los errores usualmente tienen un costo asociado en el contexto de las empresas y los

procesos industriales. Acorde con lo anterior disminuir los errores en las predicciones permite tomar mejores decisiones en cualquier área que cuente con suficientes datos y contribuye al uso eficiente de los recursos económicos y físicos.

Para realizar dicha disminución en los errores se incorpora autoaprendizaje. Un ejemplo de un trabajo previo que exitosamente ha empleado el autoaprendizaje es el presentado por Sallab & Rashwan (2011, pp. 224), los cuales demostraron una mejora apreciable en la realización de pronósticos. En su trabajo demostraron que la clasificación de correos electrónicos en categorías como "enviado", "bandeja de entrada", "eliminado", entre otros, puede pasar de una precisión promedio de 84,17% a una de 85,78% tras aplicar un método basado en aprendizaje no supervisado, sin la necesidad de tener etiquetas o categorías predefinidas junto con la actualización de los pesos de la red neuronal con cada nueva observación.

Con el propósito de potenciar el rendimiento del modelo propuesto, se busca identificar sus debilidades mediante el análisis de métricas de rendimiento, lo que permite obtener retroalimentación. Además, se pretende ajustar las predicciones del algoritmo utilizado en función de la realidad de los datos. Este enfoque no solo busca mejorar la precisión del modelo, sino también fomentar investigaciones futuras que amplíen el conocimiento existente. Estas investigaciones podrían explorar métodos más precisos y adecuados a la naturaleza de los fenómenos a predecir, contribuyendo así a un mayor avance en el campo.

1.3 Objetivos

1.3.1 Objetivo general

- Evaluar la predicción del comportamiento del índice bursátil S&P500 mediante la aplicación del modelo de inteligencia artificial Transformer, integrando técnicas de autoaprendizaje.

1.3.2 Objetivos específicos

- Aplicar el modelo Transformer de series de tiempo para predecir el comportamiento del índice bursátil S&P500.
- Incorporar técnicas de autoaprendizaje al modelo Transformer con el fin de incrementar su capacidad predictiva.
- Comparar los resultados obtenidos con los de otros enfoques de predicción del S&P500, identificando sus ventajas y limitaciones.

2. Capítulo 2. Revisión sistemática de literatura

2.1 Tipo de revisión

La necesidad de una revisión de literatura respecto a investigaciones que apliquen Transformer y autoaprendizaje para mejorar pronósticos del índice S&P500, recae en que es necesario evaluar los aspectos que ya han sido tratados por otras investigaciones para enfocar la presente en trabajar los tópicos de mayor contribución y novedad. Adicionalmente, para facilitar la comparación de los resultados obtenidos con los de otros enfoques identificando las ventajas y limitaciones.

Para esto, se utiliza como guía la metodología **PRISMA**, debido a que sus ítems son relevantes para las revisiones sistemáticas y ofrece una ruta metodológica completa para la selección y procesamiento de las publicaciones encontradas en las bases de datos consultadas (Pagea, et al., 2021, pp. 792 - 795).

Por otra parte, el tipo de revisión escogida es meta análisis con el fin de desarrollar el proceso de consulta de información de modo que las publicaciones tenidas en cuenta estén estrictamente relacionadas con el problema que se desea abordar, y para extraer características agregadas de los documentos y su evolución a través de los años (Sánchez, 2010, p. 53 - 63).

Para ello, se toma como preguntas de investigación las siguientes:

- ¿Cómo se ha abordado el pronóstico del S&P500 en los últimos años?
- ¿Qué modelos de pronóstico del S&P500 han sido los que han obtenido mejores resultados?

- ¿Cuáles han sido las limitaciones frecuentes en las investigaciones relacionadas con el problema de estudio?

2.2 Protocolo de revisión

2.2.1 Criterios de inclusión y exclusión de artículos

Las publicaciones deben contener los siguientes términos en su título, resumen o palabras clave: "transformer", "s&p500", "artificial intelligence", "auto learning", "self learning" o "forecast".

Los artículos deben contener al menos uno de los siguientes términos en su título, resumen o palabras clave: "predictive model", "forecast model", "prognostic model", "auto adjusting model", "transformer forecast", "forecast self adjust", "auto correct forecast", "transformer time series forecast", "time series transformer forecast", "neural network forecast", "deep learning forecast model" o "transformer model".

Los artículos que incluyan "s&p500" deben contener al menos uno de los siguientes términos en el título, resumen o palabras clave: "transformer", "self learning", "auto learning", "auto tuning", "self optimizing", "auto adjusting", "adaptative forecast adjust", "auto forecast" o "adaptive forecast".

Los artículos deben haber sido publicados desde el 2017 (año donde se publicó el artículo base del modelo Transformer) en adelante.

Se excluyen los artículos que no contienen en su título, resumen o palabras clave los términos “forecast” y “s&p500”, puesto que la investigación requiere comparar los resultados obtenidos con estudios similares, en este caso que hayan utilizado medidas de desempeño tras aplicar modelos de pronóstico a dicho índice en un periodo de tiempo y proporcionen detalles sobre los datos y técnicas usadas.

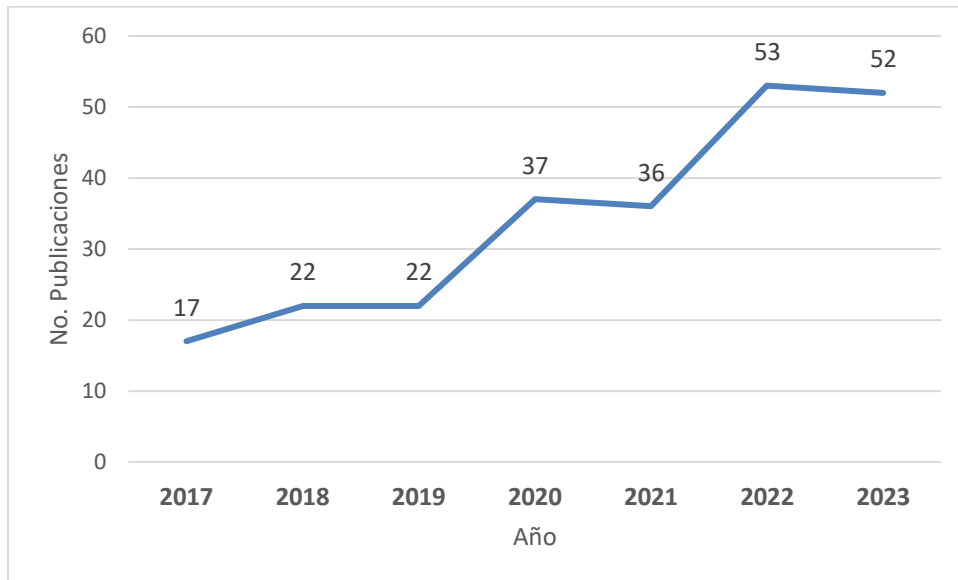
2.3 Desarrollo de la revisión

Para realizar el proceso de búsqueda se utilizó la base de datos Scopus, usando la siguiente ecuación de búsqueda:

```
( ( TITLE-ABS-KEY ( "transformer" OR "s&p500" OR "artificial intelligence" OR "auto learning" OR "self learning" OR "forecast" ) AND TITLE-ABS-KEY ( "transformer" OR "s&p500" OR "artificial intelligence" OR ( "predictive model" OR "forecast model" OR "prognostic model" OR "auto adjusting model" OR "transformer forecast" OR "forecast self adjust" OR "auto correct forecast" OR "transformer time series forecast" OR "time series transformer forecast" OR "neural network forecast" OR "deep learning forecast model" OR "transformer model" ) ) ) OR TITLE-ABS-KEY ( "s&p500" AND ( "transformer" OR "self learning" OR "auto learning" OR "auto tuning" OR "self optimizing" OR "auto adjusting" OR "adaptative forecast adjust" OR "auto forecast" OR "adaptive forecast" ) ) AND PUBYEAR > 2016 ) AND ( s&p500 ) AND ( forecast )
```

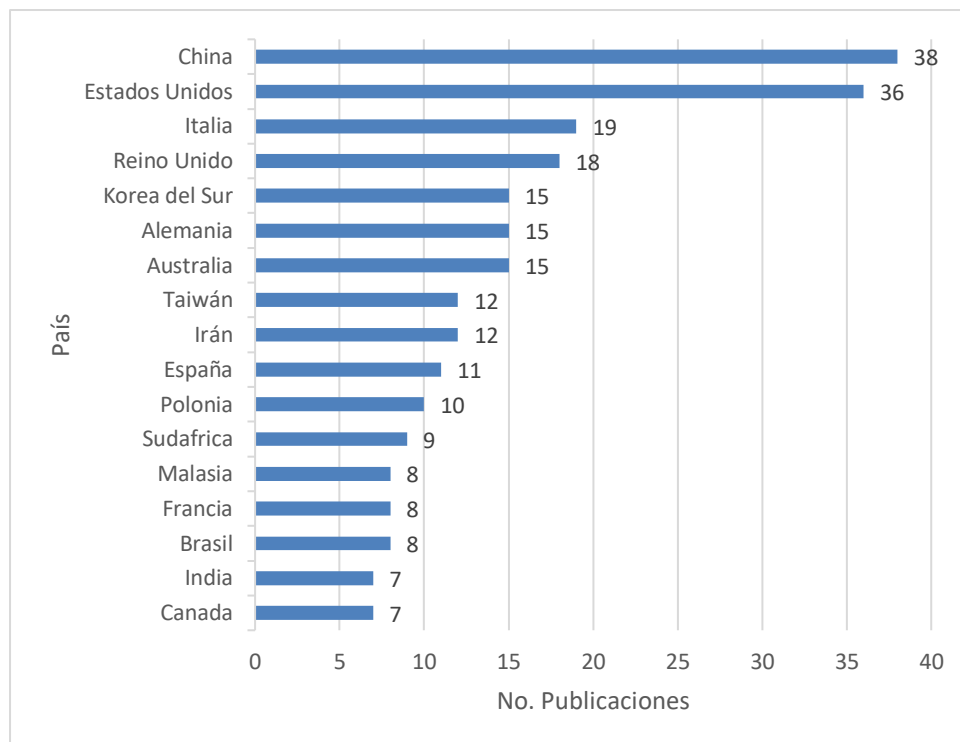
La anterior ecuación contiene los principales tópicos de interés y la segmentación que se requiere para filtrar solo estudios relevantes y que hayan sido publicados desde 2017 en adelante, a partir de esto se obtuvieron 239 documentos.

Figura 2-1: Producción de documentos por año en la temática de interés.



Fuente: Scopus (2023).

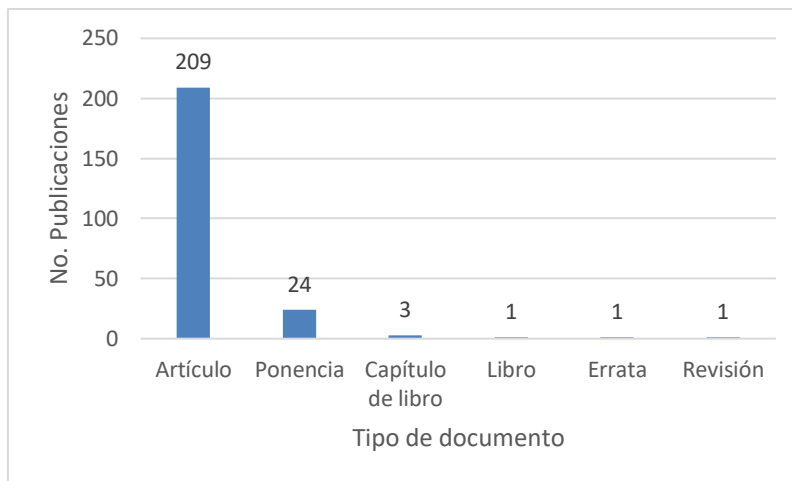
A partir de la figura 2-1 se puede identificar una tendencia creciente en la producción de publicaciones científicas desde 2017 (17) a 2023 (52) en los tópicos descritos en la ecuación de búsqueda, por lo que se evidencia gran interés de la comunidad científica en abordar el área problema.

Figura 2-2: Gráfico de producción de documentos por país.

Fuente: Scopus (2023).

A partir de la figura 2-2 se puede evidenciar que la mayor parte de la producción científica con respecto al número de publicaciones ha provenido principalmente de China (38), Estados Unidos (36) e Italia (19), con poca participación de los países de América Latina, siendo Brasil el que cuenta con mayor cantidad de contribuciones (8).

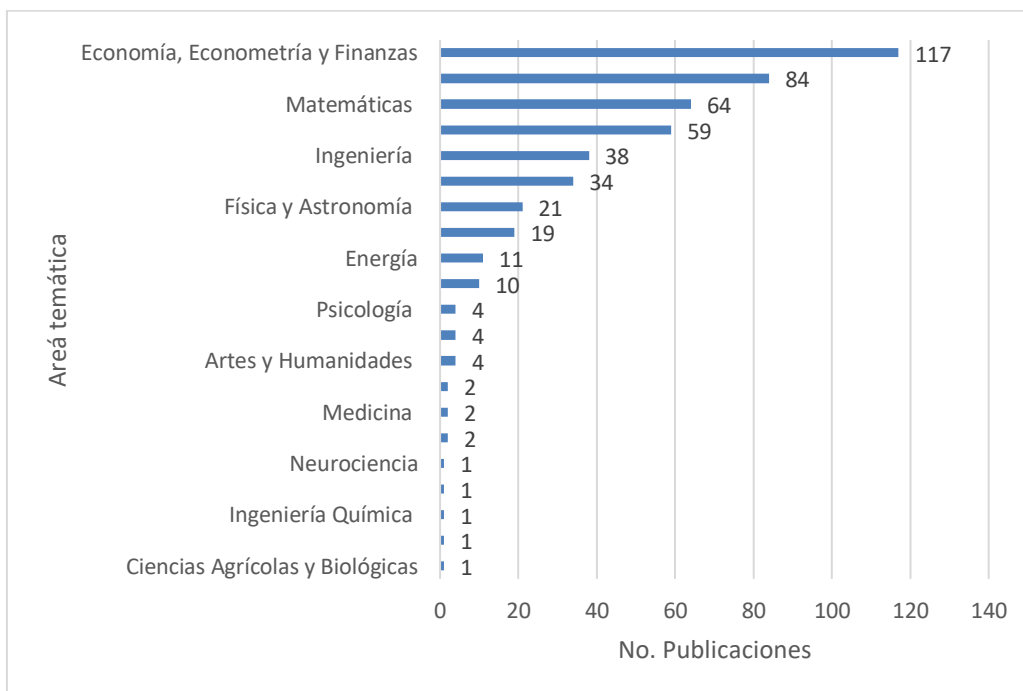
Figura 2-3: Número de documentos por tipo.



Fuente: Scopus (2023).

La figura 2-3 muestra que la mayoría de publicaciones encontradas corresponden a artículos científicos (209), seguido por artículo de ponencia (24), capítulo de libro (3), libro (1), errata (1) y revisión de artículo (1).

Figura 2-4: Número de documentos por área temática.



Fuente: Scopus (2023).

Por su parte, la Figura 2-4 muestra que la mayoría de documentos encontrados corresponden al área de economía, econometría y finanzas (117), ciencias de la computación (84), matemáticas (64), administración de negocios y contabilidad (59), ingeniería (38) y ciencias de la decisión (34), entre otros.

Se seleccionaron las 10 principales publicaciones con el fin de dirigir la investigación con base en la producción científica más acorde con el tema del trabajo y establecer la línea base para comparar los resultados obtenidos con otros similares. Para ello, se consideraron únicamente artículos, con los cuales se realizó lectura del título y el resumen de cada uno, descartando así todos aquellos que no contenían los términos Transformer, inteligencia artificial o auto aprendizaje aplicados a la predicción del S&P500.

A partir de esto, quedaron 105 artículos de los cuales se les dio prioridad a los que aplicaban Transformer en el pronóstico de series de tiempo, seguido de auto aprendizaje e inteligencia artificial tomando en consideración la existencia de información suficiente en cuanto a la presentación de resultados y el uso de métricas de desempeño. La Tabla 2-1 presenta los artículos seleccionados a partir de esta revisión sistemática de literatura:

Tabla 2-1: Artículos seleccionados en la revisión de literatura.

Nombre de publicación	Autores	Año	Método
Prediction with Time-Series Mixer for the S&P500 Index	Ye J., Gu J., Dash A., Deek F.P., Wang G.G.	2023	Arquitectura Transformer para series de tiempo acoplado con MLP-Mixer
Candidate point selection using a self-attention mechanism for generating a smooth volatility surface under the SABR model	Kim H., Park K., Jeon J., Song C., Bae J., Kim Y., Kang M.	2021	Red Transformer como red central de procesamiento
Deep Reinforcement Learning with Comprehensive Reward for Stock Trading	Zhou Q., Qu T., Han Y., Duan F., Zhou, Qibin.	2023	Red convolucional temporal y red Transformer en aprendizaje profundo por refuerzo
Do the US president's tweets better predict oil prices? An empirical examination using long short-term memory networks	Beyer D. S., Coussement K., De Caigny A., Pérez L.F., Creemers S.	2023	Transformer (BERT) en combinación con red neuronal profunda de memoria a corto plazo (LSTM) con un retraso de cinco días tanto para el precio del petróleo como para datos textuales de Twitter
Support for Stock Trend Prediction Using Transformers and Sentiment Analysis	Kaeley., H, Qiao., Y & Bagherzadeh., N.	2023	Transformer para pronóstico de series de tiempo y análisis sentimental
Forecasting stock price using integrated artificial neural network and metaheuristic algorithms compared to time series models	Shahvaroughi Farahani M., Razavi Hajiagha S.H.	2021	Red neuronal artificial (ANN) entrenada con los algoritmos metaheurísticos de optimización de araña social (SSO) y el algoritmo bat (BA)
Financial Forecasting with Word Embeddings Extracted from News: A Preliminary Analysis	Barbaglia L., Consoli S., Wang S.	2021	Redes neuronales recurrentes autorregresivas que operan en entorno probabilístico
Application of machine learning in algorithmic investment strategies on global stock markets	Grudniewicz, J.	2023	Redes neuronales, K vecinos más cercanos, árboles de regresión, bosques aleatorios, clasificadores Naive Bayes, modelos lineales generalizados bayesianos y máquina de soporte vectorial
Time-sequencing European options and pricing with deep learning – Analyzing based on interpretable ALE method	Liang. L., Cai. X.	2022	Red neuronal convolucional (1D-CNN) y modelo LSTM

Fuente: elaboración propia.

Se identificó poca producción científica en la aplicación de Transformer para pronóstico del S&P500, por el contrario, se encontraron diversos estudios en la aplicación de machine learning, y un menor número relacionado con el uso de autoaprendizaje a partir de algoritmos metaheurísticos para optimizar métricas de desempeño.

2.4 Síntesis de la revisión de literatura

Acorde con el proceso de revisión sistemática se identificaron los métodos utilizados en otras investigaciones para abordar el problema de pronóstico del S&P500 y la incorporación de técnicas de auto aprendizaje. Se resalta la combinación de diversas metodologías computacionales y de inteligencia artificial, con métodos tradicionales desde las matemáticas y la estadística, evidenciando como principal aplicación la predicción de variables económicas y financieras en la búsqueda realizada.

En relación al uso de Transformer para pronóstico del S&P500 los autores Ye, Gu, Dash, Deek y Wang (2023), presentan la arquitectura TS-Mixer conformada por dos tipos de capas MLP (Multilayer Perceptron): una capa mezcladora de características y una capa mezcladora temporal. La capa de mezcla de características se aplica de manera independiente a cada punto de datos para capturar la correlación entre las características, y la segunda capa mezcladora extrae la dependencia temporal de cada característica a lo largo de toda la secuencia de entrada.

En comparación con las redes neuronales tradicionales utilizadas en la predicción de secuencias, TS-Mixer demuestra un mejor rendimiento según los autores, aunque no indican puntualmente las métricas de desempeño. En contraste, Kim, Park, Jeon, Song, Bae, Kim & Kang (2021), indican que se obtuvo un error medio de pronóstico de los precios de opciones de empresas europeas de menos de 3% usando datos de los 3 meses anteriores a partir de la predicción del índice S&P500 con Transformer.

En cuanto al mejoramiento de los modelos predictivos, Zhou, Qu, Han, Duan, Zhou & Qibin (2023), sugieren que el uso de deep reinforcement learning aplicado a redes

convolucionales y Transformer puede ser efectivo para mejorar las estrategias de trading al considerar tanto la predicción de precios del S&P500 a corto plazo como la tendencia a largo plazo, lo que puede llevar a reducir el riesgo en el trading de acciones.

Del mismo modo Beyer, Coussement, De Caigny, Pérez & Creemers (2023), presentan una arquitectura de redes neuronales de memoria a corto y largo plazo (LSTM) en combinación con Transformer para la predicción de los precios del petróleo, encontrando que la utilización de tweets del presidente de los Estados Unidos como entrada permitió lograr un rendimiento competitivo respecto otras investigaciones en la predicción del precio del petróleo en comparación con modelos tradicionales.

Por otro lado, Kaeley, Qiao & Bagherzadeh (2023), destacan una mejora en la precisión predictiva del precio de acciones de Facebook, Amazon, Apple, Netflix, Google con un conjunto de 3700 registros, desde el 18/03/2019 hasta el 18/02/2022 obtenidos del API de datos históricos de EOD. Se utiliza Transformer y análisis sentimental evidenciando que brindan mejores resultados en comparación con redes neuronales recurrentes a medida que aumenta la longitud de la secuencia desde los 5 días hábiles, obteniendo un accuracy de 89% usando un periodo de retraso de 29 días, es decir, usando datos de n días hacia atrás para predecir el precio de apertura del periodo $n+1$.

Del lado de las redes neuronales tradicionales Farahani & Hajiagha (2021), comentan que al incorporar a una red neuronal recurrente los algoritmos metaheurísticos BAT (Bat Algorithm) y SSO (Social Spider Optimization), se obtiene un error absoluto medio de 1×10^{-40} y 1×10^{-52} respectivamente en comparación con un valor 13,499 del modelo sin la mejora. Así mismo, Barbaglia, Consoli & Wang (2021), realizan la predicción de la varianza del S&P500 a un horizonte de 1 día usando redes neuronales recurrentes indicando la probabilidad de ocurrencia de cada pronóstico, evidenciando que el modelo "bert with punctuation" obtiene el mejor desempeño con un MSE de 0,983.

Respecto al uso de pronósticos para elaborar estrategias de inversión Grudniewicz (2023), realiza un estudio de comparación del desempeño de estrategias tradicionales de inversión (buy-and-hold) con estrategias basadas en machine learning, identificando que las segundas superaron a las estrategias de compra y retención en todos los índices de

mercado analizados incluyendo el S&P500, donde el modelo BGLM obtuvo la segunda mejor calificación en términos de IR promedio (average information ratio), con un puntaje de 6,11 después de SVM (support vector machine), que obtuvo el mejor puntaje con un valor de 6,28.

Por su parte, Liang & Cai (2022), realizan una predicción del precio de opciones ETF50 de China y opciones del S&P 500, aunque no indican las métricas de desempeño obtenidas respaldan la capacidad para capturar información de series temporales y su énfasis diferenciado en características de entrada y registros históricos usando modelos 1D-CNN y LSTM, lo que muestra su relevancia para tareas de pronóstico de opciones.

De este modo, se encontró como principal limitación la falta de publicaciones enfocadas al uso de Transformer para predecir el comportamiento del S&P500, por otro lado, se evidenció que en la mayoría de publicaciones fue necesaria la integración de varios métodos para producir el mejor resultado posible a partir de los datos disponibles, y esto obedece al hecho de que no hay un modelo universal para hacer predicciones.

Finalmente, esta revisión de literatura evidenció una producción científica cada vez mayor en el área de inteligencia artificial desde 2017 a 2023, con la mayoría de publicaciones enfocadas en el uso de la ciencia computacional para brindar mayor fortaleza y adaptabilidad a los modelos de pronóstico con datos cuantitativos y cualitativos, en conjunto con la incorporación de técnicas de autoaprendizaje y corrección de errores.

3. Capítulo 3. Marco teórico

3.1 Conceptos y teorías de soporte

3.1.1 ¿Qué son pronósticos?

Según Gallegos (2013, p. 6, cap. 1): “pronosticar es realizar un enunciado sobre el valor futuro de una variable de interés, fundamentado ya sea por el análisis de datos históricos disponibles, por el juicio de expertos en el tema o por una combinación de ambas cosas”, dichos análisis se realizan utilizando en conjunto o por separado modelos cuantitativos y cualitativos, con los que se pretende explicar algún comportamiento o fenómeno a partir de su información histórica y la aplicación de métodos desarrollados por la ciencia, por lo que finalmente es una estimación anticipada del valor de una variable que sirve en la toma de decisiones para manejar la incertidumbre.

Por otra parte, según Cadena, et al. (2018, p. 142), solo existen dos tipos de pronóstico, los de relación causal que explican el comportamiento de una variable dependiente en función de otras independientes (fórmulas matemáticas), y los de series de tiempo que pronostican una variable con base en su información histórica.

3.1.2 Modelos predictivos

Hacen referencia a modelos que implementan metodologías de pronóstico, y su implementación y métodos de ajuste han sido tratados de diferentes maneras en la literatura académica. Por ejemplo, Espino (2017), realiza un estudio sobre las técnicas y modelos utilizados en el análisis predictivo, encontrando que sus principales aplicaciones

se encuentran en los sectores de marketing, actuarial, servicios financieros, administración pública, y en general es usado en las empresas para predecir el comportamiento de sus clientes y métricas del negocio ante determinados estímulos.

Incluso pueden usarse para detectar fraudes o para ofrecer recomendaciones personalizadas frente a la compra de productos, además el autor resalta que para realizar análisis predictivo es necesario haber recolectado un conjunto suficientemente amplio de datos para que sea posible encontrar patrones que permitan construir reglas de decisión capaces de anticipar el comportamiento de los fenómenos de estudio.

Con respecto a casos de aplicación del análisis predictivo, Zúñiga (2021), realiza la implementación de un modelo de predicción del tiempo faltante para terminar de fabricar los productos que están en proceso en una célula de manufactura simulada, dicho modelo utiliza procesos de regresión lineal con una implementación basada en Python 3.6 conectado a las bases de datos de un proceso de manufactura simulado con el software CIROS.

Para seleccionar el modelo final se escogió entre 36 opciones de modelos de regresión propuestos con los criterios de mayor coeficiente de determinación y menor error cuadrático medio, encontrando que la predicción (modelo escogido) acertó al 76% frente a las observaciones reales (datos simulados).

Por otra parte, entre las herramientas contemporáneas de modelamiento y elaboración de pronósticos se encuentra el Machine Learning (aprendizaje de máquina o aprendizaje automático). Según Ramírez (2018), esta es una subárea de la inteligencia artificial que hace referencia a un conjunto de técnicas algorítmicas apoyadas en una suficiente potencia de cómputo para extraer patrones y características de los datos a través de procesos iterativos y de ajuste hasta lograr reproducir las características del fenómeno de interés. Adicionalmente, es una disciplina que ha tomado mayor relevancia en las últimas décadas debido al progreso de la informática y la disponibilidad de sistemas de cómputo cada vez más potentes y especializados.

Acorde con lo anterior, Maisueche (2019), presenta un estudio sobre la utilización de esta técnica en la industria, destacando que su relevancia actual se sustenta principalmente en su capacidad para reducir tiempos de ejecución de procesos y toma de decisiones.

Por consiguiente, es posible su aplicación a situaciones o procesos donde se requiere encontrar tendencias de las variables y acorde con esto realizar predicciones e incluso anticipar errores en procesos de producción, esto a través de la clasificación automatizada según el fin de cada proceso, como en el caso de la separación de residuos y productos con el uso de sensores.

No obstante, Nipun, Chaspari y Behzadan (2018, p. 21), argumentan que a pesar de que esta disciplina permite analizar datos y reconocer patrones de manera efectiva, son los humanos al configurar y desarrollar los modelos quienes determinan la calidad de los resultados y cuáles son las decisiones a tomar con base en ello.

Por su parte, Correa (2010), desarrolla una investigación para construir un modelo de predicción y control de la velocidad de corte con respecto a la rugosidad superficial en el proceso de micro fresado de alta precisión usando redes Bayesianas, las cuales son un método de aprendizaje automático que permite aprender las relaciones de dependencia y causalidad entre las variables del problema.

Dicho autor obtuvo un promedio de precisión entre 81.2 % y 90%, lo cual constituye evidencia sobre la utilidad para modelar procesos industriales usando aprendizaje automático para realizar tareas mecánicas como el corte de materiales.

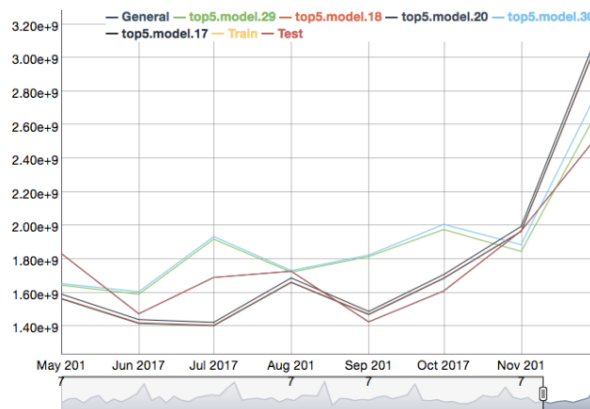
Del mismo modo, Rico y Sánchez (2018), implementan el algoritmo Naïve Bayes para automatizar la predicción del rendimiento académico en un curso de ecuaciones diferenciales de nivel universitario, obteniendo una exactitud de 66,28% en los pronósticos respecto a las observaciones reales.

Teniendo en cuenta lo anterior, se puede identificar como los modelos predictivos presentan un gran apoyo a los procesos industriales y en la obtención de información valiosa a partir de bases de datos, sin embargo, poseen un margen de error significativo y

es necesario su constante ajuste para evitar perder la calidad en los resultados, lo cual constituye un argumento adicional para el desarrollo del presente trabajo.

A continuación, se presenta una comparación de modelos de pronóstico comúnmente usados:

Figura 3-1: Comparación entre 5 modelos de pronóstico comunes.



Fuente: tomado de Suarez (2018, p. 100).

La figura 3-1 muestra la comparación de 5 modelos de pronósticos cuantitativos, siendo el #29 un modelo Extreme Gradient Boost Sin Tendencia y Estacionalidad (aprendizaje automático), el #18 HoltWinters Multiplicativo con Transformación Logarítmica (series de tiempo), el #20 HoltWinters Aditivo con Transformación Logarítmica (series de tiempo), el #30 XGBoost Sin Tendencia y Sin Estacionalidad (aprendizaje automático), y el #17 HoltWinters Multiplicativo (series de tiempo).

De este modo se encontró que el mejor modelo en este caso es el Extreme Gradient Boost Sin Tendencia y Estacionalidad, el cual ajusta un árbol de regresión o clasificación a los datos, posteriormente se construye un segundo modelo que se centra en predecir las observaciones que el primer modelo predijo mal y a través de varias iteraciones se logra mejorar los resultados.

3.1.3 Inteligencia artificial

Jaramillo (2012, p. 10), menciona que la inteligencia artificial hace referencia al diseño de sistemas que emulan características asociadas con el comportamiento humano haciendo uso de la potencia computacional y reglas impuestas a través de los lenguajes de programación existentes.

Una de sus ventajas reside en que permite manejar la complejidad e implementar sistemas que se pueden autoajustar según criterios expertos, dichos sistemas se vuelven inteligentes en la medida en que procesan información de entrada y mejoran su desempeño en la salida, además sirven para modelar fenómenos no lineales y de los que no se conoce de manera detallada su funcionamiento.

Cabe mencionar que la inteligencia artificial se ha definido desde diferentes perspectivas como sistemas que piensan como humanos, sistemas que piensan racionalmente, sistemas que actúan como humanos y sistemas que actúan racionalmente, a través de reglas de comportamiento, la robótica y las diferentes disciplinas relacionadas con la computación moderna, replicando comportamientos humanos a través de la programación, el uso de sensores y la capacidad de las máquinas para guardar información y usarla para responder correctamente a sus variables de entrada o “triggers” (Alvarado, 2015, p. 29).

3.1.4 Aprendizaje automático

Moreno, et al. (1994, p. 7), indican que el aprendizaje automático (machine learning) es un proceso donde un sistema elige las características más relevantes de la información o datos de entrada, las compara con otras conocidas de tener registro, y si detecta diferencias significativas, adapta su funcionamiento (parámetros) de modo que cada vez sea menor la diferencia entre la salida obtenida respecto al valor real o resultado deseado.

Una de las principales razones para la creación y desarrollo de sistemas de aprendizaje automático radica en el hecho de que en varios campos del conocimiento o situaciones hay una carencia de experiencia o datos relevantes, y la representación del conocimiento asociado a esta falta de experiencia es limitada, fragmentada y, en consecuencia, insuficiente o casi nula.

Además, proporcionar a una persona todo el conocimiento y las habilidades requeridas para volverse experto en la realización de alguna tarea plantea un desafío complejo y costoso en términos de tiempo y recursos. La corrección de posibles errores en el aprendizaje resulta complicada y a menudo requiere la intervención de individuos con atención especializada y oportuna.

3.1.5 Autoaprendizaje

A diferencia del aprendizaje de máquinas, según Sallab & Rashwan (2011, pp. 225), el autoaprendizaje se enfoca en hacer capaces a los sistemas de aprender con una supervisión mínima y adaptarse a los datos del mundo real a medida que se introducen más y más observaciones nuevas en lugar de solamente aprender los patrones y estructura pasados. Puede ser llamado de diversas maneras como aprendizaje adaptativo, máquinas de autoaprendizaje o máquinas de aprendizaje adaptativo, entre otros.

Los anteriores autores mencionan que uno de los métodos de autoaprendizaje consiste en realizar “Backpropagation” ajustando los pesos de una red neuronal incorporando datos nuevos al set de entrenamiento. Si se logra una mejora en la tasa de error se aplican los nuevos pesos; de lo contrario, se conservan los antiguos.

3.1.6 Optimizador RMSProp

Según Tieleman & Hinton (2012, p. 4), es un método de optimización utilizado en inteligencia artificial que se aproxima al valor absoluto del hessiano y posee un mejor rendimiento en problemas no convexos que el descenso por el gradiente. El RMSProp se utiliza especialmente en la optimización de la función de pérdida en redes neuronales con millones de parámetros puesto que converge con un menor número de iteraciones a los valores mínimos de la función.

A pesar de la popularidad del descenso por el gradiente estocástico (SGD), sigue representando un gran costo computacional en la implementación de modelos robustos con gran cantidad de parámetros y datos utilizados, RMSProp aborda esta problemática proporcionando una estimación sesgada de la diagonal del Hessiano absoluto, lo que se traduce en una mejora de la eficiencia en el entrenamiento de modelos de machine learning.

3.1.7 Distancia de Mahalanobis

Díaz (2007, pp. 29), menciona que la distancia euclidiana ofrece una manera limitada de medir las diferencias o similitud entre variables u objetos, puesto que cada elemento perteneciente a su cálculo esta ponderado por un mismo factor, no es invariante frente a cambios de escala, asume que las variables son independientes entre sí y poseen varianza unitaria:

Dados dos vectores $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^p$, donde $\mathbf{X} = (X_1, \dots, X_p)'$ y $\mathbf{Y} = (Y_1, \dots, Y_p)'$, la distancia euclidiana entre ambos vectores se define como (ver ecuación (3.1)):

$$d(\mathbf{X}, \mathbf{Y}) = \sqrt{(X_1 - Y_1)^2 + \dots + (X_p - Y_p)^2} \quad (3.1)$$

En este sentido surge la distancia de Mahalanobis como una alternativa para incorporar la variabilidad en los datos asociados a variables que están sujetas a

fluctuaciones aleatorias de diferentes magnitudes junto con sus relaciones de dependencia, y de este modo medir su similitud acorde con la estructura de variabilidad que se tiene:

Dados dos vectores $X_i = x_{i1}, \dots, x_{ik}$ y $X_j = x_{j1}, \dots, x_{jk}$, donde la estructura de covarianzas no necesariamente es la misma, la distancia de Mahalanobis estimada entre ambos vectores está dada por (ver ecuación (3.2):

$$D_{Mahalanobis} = (X_i - X_j)' S_p^{-1} (X_i - X_j), \text{ con } i, j = 1, \dots, k, \text{ y } k = \text{número de variables en cada vector.} \quad (3.2)$$

Donde $S_p = \frac{((n-1)S_1 + (m-1)S_2)}{(n+m-2)}$ (3.3), es la matriz de covarianzas combinada de ambos grupos, con “n” y “m” siendo el número de observaciones respectivamente, y S_1, S_2 las matrices de covarianzas de cada vector (Díaz, 2007, pp. 29).

3.1.8 Automatización de modelos predictivos

Derby (2005, p. 17), menciona que la automatización consiste en algún “método para controlar automáticamente la operación de un aparato o artefacto, proceso o sistema integrado por diversos componentes a través de medios mecatrónicos, electrónicos y computacionales, que sustituyen los órganos sensitivos y la capacidad de decisión del ser humano”, de ahí que actualmente existe una tendencia hacia la automatización de los procesos relacionados con la aplicación de técnicas de analítica avanzada, aprovechando las capacidades computacionales actuales y la identificación de tareas que son repetitivas y susceptibles de replicar con algoritmos (Management solutions, 2020, p. 13).

A nivel practico la automatización de modelos predictivos, según Avendaño (2020, p. 3), hace referencia al uso de las herramientas tecnológicas y los lenguajes de programación para mecanizar la ejecución periódica de modelos de pronóstico, de este

modo se evitan las ejecuciones manuales y se pueden incorporar métricas relevantes de salida e incluso algún tipo de ajuste con base en el desempeño del modelo.

En concordancia con lo anterior (Nieto, 2006, p. 120), menciona que la razón de ser de la automatización recae en que los procedimientos lógicos se encomiendan a maquinas automatizadas especiales u ordenadores, puesto que estos procesan información mucho más rápido que el hombre, haciendo uso de modelos y procedimientos matemáticos que describen un fenómeno de interés según el conocimiento experto o el razonamiento humano, minimizando la intervención y manteniendo resultados precisos y en menor tiempo.

3.1.9 Bases de datos

Castrillón, et al. (2008, p. 98), indica que una base de datos “es la organización estructurada de un conjunto de información con al menos una característica en común que permite su agrupación”, del mismo modo Gil (1994, p. 4), menciona que “una base de datos es una colección de información organizada y presentada para servir a un propósito específico, y también se entiende como un archivo de datos interrelacionados, recolectados, que satisfacen las necesidades de información de una comunidad determinada de usuarios”.

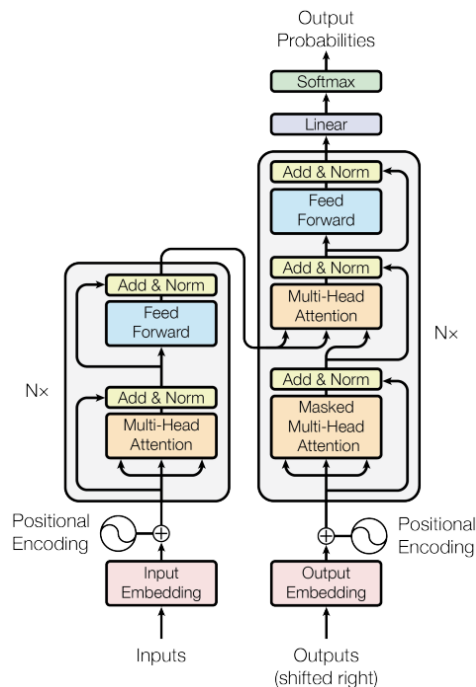
De esta manera cada unidad de información almacenada en una base de datos está compuesta por los atributos relevantes de la entidad que describe, dicha estructura facilita y se optimiza con el fin permitir las consultas de información de manera adecuada y usar relaciones entre los orígenes de datos para combinar información única de cada fuente y consolidarla para fines de gestión.

3.1.10 Modelo canónico de la arquitectura Transformer

Vaswani, et al. (2017, p. 1 - 2), indican que el modelo Transformer surgió como alternativa para mejorar el desempeño de las redes neuronales para la tarea de traducción de idiomas a partir de aprendizaje de maquina usando el mecanismo de atención, el cual permite modelar las dependencias globales entre el conjunto de entrada y el de salida a través de positional encoding.

Por otra parte, el Transformer es el primer modelo de traducción de palabras que se basa completamente en el mecanismo de autoatención para calcular representaciones de su entrada y salida sin usar redes neuronales recurrentes (RNN), alineados en secuencia o convoluciones.

Figura 3-2: Diagrama de funcionamiento general de un modelo Transformer.



Tomado de: Vaswani, et al. (2017, p. 3).

La figura 3-2 ilustra el funcionamiento del modelo canónico de la arquitectura Transformer, aquí el dato de entrada consiste en una oración en el idioma a traducir que se transforma en vectores de palabras llamados "embeddings", los cuales se enriquecen con información contextual a través de un mecanismo de atención que pondera la importancia de las palabras en función de su relación con otras palabras en la oración.

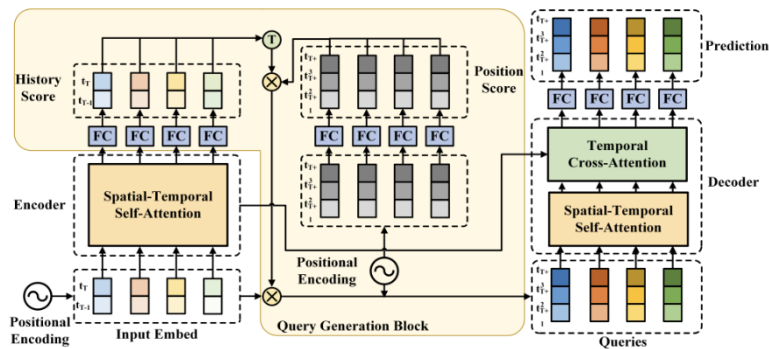
Esta atención múltiple permite que el modelo capture conexiones a lo largo de toda la secuencia. Luego, estas representaciones se procesan a través de múltiples capas de codificación y decodificación en paralelo, permitiendo al modelo aprender patrones y relaciones en las secuencias de palabras de entrada y salida. Durante la decodificación, otro mecanismo de atención se emplea para enfocarse en las partes relevantes de la oración de origen mientras genera palabras en el idioma destino, lo que lleva a una traducción coherente y precisa.

En el presente trabajo se utiliza un modelo Transformer aplicado a series de tiempo, el cual, según Chen, et al. (2021, pp. 1 - 3), consiste en la adaptación del modelo canónico de la arquitectura de AI Transformer que inicialmente se diseñó para pronosticar secuencias de palabras, haciendo la inclusión de un bloque de atención para el manejo de las dependencias temporales en combinación con el bloque de atención estándar para manejar las dependencias espaciales y de este modo lograr pronosticar secuencias temporales como ocurre con las series de tiempo.

Por ejemplo, si se desea pronosticar en qué medida el precio de una acción subirá, bajará o se mantendrá igual en el siguiente período de tiempo, después de haber procesado la secuencia de entrada a través de las capas del modelo Transformer, se obtienen como salidas secuencias transformadas que contienen representaciones enriquecidas y aprendidas de los patrones en la serie temporal, luego tras aplicar la función de activación ReLU u otra adecuada se obtiene como salida un valor numérico continuo que representa la predicción para el siguiente punto en la serie.

3.1.11 Transformer para series de tiempo (Método de Pronóstico propuesto)

Figura 3-3: Esquema de funcionamiento de un modelo Transformer para pronóstico de series de tiempo.



Fuente: tomado de Chen, et al. (2021, pp. 3).

La figura 3-3 ilustra el funcionamiento de un Transformer para series de tiempo, el cual es una arquitectura de red neuronal que se enfoca en modelar las relaciones temporales en secuencias numéricas, su funcionamiento se basa en descomponer la secuencia en vectores de entrada, que luego son procesados por capas de atención que permiten considerar la influencia de puntos de tiempo pasados y futuros sobre el valor actual.

A su vez, posee capas feed-forward (retroalimentación) que se encargan de proyectar y transformar las características temporales en un espacio de mayor dimensionalidad, ampliando la capacidad del modelo para aprender y representar estructuras temporales más sofisticadas y adaptarse a distintos patrones presentes en los datos en diferentes periodos en la serie y luego de fluir a través del modelo, se llega a la última capa donde se produce una predicción numérica directa basada en la representación transformada de la serie.

De manera adicional los resultados obtenidos en Zhou, et al. (2021), indican que el modelo Transformer para series de tiempo obtiene mejores medidas de desempeño que los modelos de redes neuronales recurrentes con perceptrón multicapa y los tradicionales modelos autorregresivos (AR), por lo que el mecanismo espacio-temporal es aplicable y efectivo para el pronóstico de series de tiempo.

4. Capítulo 4. Metodología

La investigación se desarrolla bajo un enfoque cuantitativo, el cual por su naturaleza parte de la información recolectada alrededor del problema o situación de estudio, generando mediciones y transformaciones a los datos para describir su comportamiento y permitir llegar a conclusiones sobre las hipótesis planteadas. Esto a través del uso de herramientas estadísticas y tecnológicas que hagan visible el comportamiento de las variables y permitan mejorar su entendimiento (Sampieri, 2014, p. 4).

4.1 Base de datos usada

Se utiliza la base histórica de contratos de futuros del S&P500 proporcionada por Investing (2023), debido a que proporciona una medida del valor de las acciones de las 500 compañías más importantes que se negocian en la Bolsa de Valores de Nueva York y sirve como un indicador del desempeño del mercado de valores de Estados Unidos.

La base trabajada contiene información desde el 1 de diciembre de 1997 hasta el 22 de septiembre de 2023 en la temporalidad diaria, para la evaluación de los pronósticos generados con el modelo Transformer se toma información del 25/09/2023 al 19/12/2023 (60 datos). Se escoge la base de futuros del S&P500 en lugar del índice S&P500, debido a que la información histórica del índice de futuros contiene el volumen de dinero negociado en cada sesión de la bolsa y así puede mejorar el poder predictivo del modelo a implementar.

Se consideran las variables de **fecha** (fecha del día de operación del índice en la bolsa de Nueva York), **precio de cierre** (valor que tenía el índice en el cierre de la jornada bursátil, es decir, a las 4:00 pm hora local de Nueva York), **apertura** (valor que tenía el índice en la apertura de la bolsa, es decir, a las 9:30 am hora local de Nueva York), **máximo** (valor más alto alcanzado en la jornada bursátil), **mínimo** (valor más bajo alcanzado en la jornada bursátil) y el **volumen** (monto de dinero en compras y ventas de contratos de futuros del S&P500 en la jornada bursátil) en dólares americanos.

4.2 Modelo Transformer

Para responder al primer objetivo específico se realiza la implementación del modelo Transformer. De este modo, se utiliza código Python y el dataset mencionado. Se procedió importando desde Google Colab la base de datos mencionada en el punto anterior, alojada en Google Drive, posteriormente se instalaron e importaron las librerías torch, numpy, pandas, sklearn, train_test_split, time_series_transformer, scikit-learn, tensorflow, scipy.spatial.distance, time, matplotlib, torch, keras_tuner y desde sklearn la librería metrics. Se realizó el procesamiento y transformación del dataset estableciendo como índice la variable “fecha” y convirtiendo las variables precio de cierre, apertura, máximo, mínimo y el volumen a formato Float.

Para la validación cruzada se dividieron los datos en conjunto de entrenamiento, validación y test tomando la base del 1 de diciembre de 1997 hasta el 22 de septiembre de 2023 en la temporalidad diaria, se configuro el parámetro shuffle = “True” en la función sklearn.model_selection.train_test_split() debido a que los modelos de Deep Learning como Transformer, trabajan mejor usando conjuntos de datos aleatorios a partir de los datos originales antes de dividirlos en conjuntos de entrenamiento, validación y prueba, teniendo en cuenta que después de cada época, se evalúa el rendimiento del modelo en el conjunto de validación. Esto ayuda a garantizar que cada conjunto sea representativo de la distribución general de los datos. En contraste con los métodos tradicionales de series de tiempo, en Deep Learning si los datos están ordenados por tiempo o categoría o en una

división secuencial, esto podría resultar en conjuntos de datos que no representan bien la distribución global, y al utilizar dicho modelo en pronósticos puede ser inútil para predecir eventos siguientes en la secuencia.

Se continuó dividiendo los datos en secuencias de entrada y salida (variable objetivo) usando el método `"torch.tensor(chunk)"`, luego se normalizó el dataset restando la media y dividiendo por la desviación estándar, se continuó aplicando la función `"train_test_split"` dividiendo el 30% del dataset en datos de test y validación, y el 70% en datos de entrenamiento (1959 muestras de test y 4571 muestras de entrenamiento). Tras esto se usó programación funcional para definir la capa de atención multi-cabeza, utilizada para capturar relaciones a largo plazo en secuencias temporales. Esto es acorde con lo realizado por Hestness, et al.(2017), quienes utilizan la secuencia de datos normalizados antes de realizar la división aleatoria en conjuntos de entrenamiento, validación y test. Esto puede introducir un data leakage en los métodos tradicionales, sin embargo, en Deep Learning los modelos mejoran su desempeño con la diversidad de datos de diferentes características en lugar de contar solo con una fracción pequeña que no contiene información reciente. Adicionalmente, el modelo Transformer en su bloque de positional encoding asigna a cada elemento de la serie total considerada su posición relativa y por ello a pesar de repartirse de manera aleatoria los datos para entrenamiento, validación y test se conserva la ubicación temporal de los datos a lo largo de la optimización del modelo.

Una vez definido el módulo anterior se crearon las variables asociadas a los parámetros del Transformer siendo estas: B (tamaño del lote o rejilla), S (longitud de la secuencia): 60, H (número de cabezas de atención), D (dimensionalidad o número de variables implicadas): 5. Ahora se define la codificación posicional para inyectar información sobre la posición de cada token (punto de datos en la secuencia temporal) en la secuencia analizada, de esta manera, el modelo puede capturar información sobre la posición de cada token en la secuencia, lo que mejora su capacidad para entender la estructura y las relaciones en los datos de entrada.

Para esto, se genera una matriz que contiene los valores de codificación posicional para cada posición en la secuencia y para cada dimensión de los embeddings o representaciones vectoriales de las secuencias, usando funciones seno y coseno de ángulos basados en la posición de cada token en la secuencia.

Llegado a este punto, se procede a definir la estructura del Transformer para aplicar optimización con búsqueda aleatoria a los parámetros número de capas, dimensión de la rejilla, número de cabezas de atención, número de capas ocultas, en conjunto con la incorporación del optimizador RMSprop con un learning rate de 0.00005. De este modo, tras ejecutar el entrenamiento se pueden utilizar los pesos aprendidos para realizar pronósticos según los datos de entrada que se proporcionen a través de multiplicación de matrices.

Para el entrenamiento del modelo se utilizan los datos del 1 de diciembre de 1997 hasta el 22 de septiembre de 2023 en la temporalidad diaria, realizando validación cruzada, y se toman 50 épocas de ejecución con el optimizador RMSProp para calibrar el modelo.

4.3 Evaluación de desempeño

Para evaluar el desempeño del Transformer se utiliza como función de pérdida el “error medio absoluto”, puesto que permite cuantificar el error entre las predicciones y los valores reales para entrenar el modelo disminuyendo dicha discrepancia y ajustarse a los datos suministrados, proporcionando una visión general de la capacidad para realizar predicciones precisas. Se escogió la métrica mencionada debido a que se probó usando el “error cuadrático medio” y se obtuvo mejores resultados usando el error medio absoluto.

4.3.1 Formulas

4.3.1.1 Error absoluto medio

$$MAD = \frac{1}{n} \sum_{t=1}^n |Y_t - \hat{Y}_t| \quad (4.1)$$

Donde “n” representa el número de observaciones, Y_t es lo observado y \hat{Y}_t es el pronóstico.

4.3.1.2 Error cuadrático medio

$$MSE = \frac{1}{n} \sum_{t=1}^n (Y_t - \hat{Y}_t)^2 \quad (4.2)$$

Donde “n” representa el número de observaciones, Y_t es lo observado y \hat{Y}_t es el pronóstico.

4.3.1.3 Error porcentual

$$E\% = \frac{(Y_t - \hat{Y}_t)}{Y_t} \quad (4.3)$$

Donde Y_t es lo observado y \hat{Y}_t es el pronóstico.

Las anteriores formulas fueron tomadas de Hanke & Wichern (2014, p. 36).

Acorde con esto, se toma el modelo Transformer para series de tiempo que obtuvo menor valor en la función de pérdida para predecir el comportamiento del S&P500, y se generan 60 pronósticos del 25/09/2023 al 19/12/2023. Después de generar la predicción se realiza una comparación entre los valores reales y los valores de pronóstico usando el error porcentual calculado como la resta entre lo observado y lo predicho dividido lo observado.

4.4 Modelo propuesto con autoaprendizaje

Debido a que los modelos de pronóstico no son perfectos y a que los fenómenos de estudio pueden evolucionar con el tiempo, es necesario incorporar procesos de corrección y alimentación de dichos modelos con base en los errores presentados, acorde con dicha necesidad en el presente trabajo se propone un procedimiento utilizando la distancia de Mahalanobis para ajustar el modelo de pronóstico a partir del registro de errores obtenidos y su conjunto de datos asociado, a través de la similitud con un nuevo conjunto de entrada (última observación de las variables de interés), incorporando así el error que se espera tener en la generación de un nuevo pronóstico.

Se define el siguiente arreglo de filas y columnas como la matriz de aprendizaje a considerar, en la cual cada fila corresponde a una predicción del modelo:

Figura 4-1: Matriz de autoaprendizaje.

$$\begin{pmatrix} X_1 & X_2 & X_3 & \dots & X_p & e = Y_O - Y_P \\ X_{11} & X_{21} & X_{31} & \dots & X_{p1} & e_1 \\ X_{12} & X_{22} & X_{32} & \dots & X_{p2} & e_2 \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ X_{1n} & X_{2n} & X_{3n} & \dots & X_{pn} & e_n \end{pmatrix}$$

Fuente: elaboración propia.

La figura 4-1 muestra la matriz a usar en el proceso de autoaprendizaje, donde el primer elemento de la primera fila corresponde a X_{11} , siendo $X_1 \dots X_p$ las etiquetas de cada variable, $(X_1, X_2, \dots, X_n)'$ el vector de características de entrada o variables del fenómeno de interés, y e el error obtenido al restar Y_o (valor observado o deseado) con Y_p (valor del pronóstico).

4.4.1 Formula de integración de autoaprendizaje en los pronósticos

$$\hat{Y}'_t = \hat{Y}_p + \min\left(\frac{(Y_{tk} - \hat{Y}_{tk})}{Y_{tk}}\right) \tag{4.4}$$

Donde \hat{Y}'_t es el pronóstico con autoaprendizaje, Y_t es lo observado, \hat{Y}_p es el pronóstico que se va a ajustar y $\min\left(\frac{(Y_{tk} - \hat{Y}_{tk})}{Y_{tk}}\right)$ representa el error porcentual obtenido al consultar en el dataframe la fila con los datos de entrada usados para generar un pronóstico previo que cumplieron con el menor valor de la resta respecto a la distancia de Mahalanobis de este conjunto y el valor de dicha distancia calculada a partir de los datos de entrada usados en el pronóstico a ajustar.

El proceso de autoaprendizaje consiste en que al realizar un pronóstico y obtener un valor, a este se le adicione el error obtenido con el conjunto de entrada que tenga la menor distancia de Mahalanobis, es decir, se selecciona un valor de ajuste comparando una nueva observación o conjunto de entrada con todos los registros anteriores de manera iterativa hasta encontrar el que posee menor distancia de Mahalanobis versus el valor del conjunto con el cual se busca generar el pronóstico para asignar el error obtenido en dicho momento anterior como valor de ajuste del nuevo pronóstico, y se espera que con el tiempo se posea suficiente información para ajustar el pronóstico bajo la máxima similitud posible (mínima distancia de Mahalanobis) suponiendo que observaciones similares tendrán errores de pronóstico similares para un mismo modelo, dicho de otro modo se integra el error de pronóstico esperado.

Figura 4-2: Diagrama del proceso de autoaprendizaje.



Fuente: elaboración propia usando Visme.

La figura 4-2 resume el modo como se integró el modelo Transformer con autoaprendizaje para la predicción del S&P500. De manera adicional, para iniciar con el procedimiento se calculan los pronósticos de la variable de salida a un horizonte de tiempo escogido, y a partir de ese momento se guarda registro de las diferencias entre el valor real y el calculado poblando la matriz de autoaprendizaje con cada valor real nuevo, y a partir de este punto se continua ajustando el pronóstico con el criterio de menor distancia de Mahalanobis, con lo cual se espera que cada vez sea más robusto y preciso el resultado al realizar el ajuste de la predicción.

Por otro lado, se recalca que en el autoaprendizaje no se incluyeron datos del conjunto de test obtenido al realizar la división en entrenamiento, validación y test, porque es un procedimiento diferente a dicho entrenamiento. El insumo del autoaprendizaje fueron 420 registros de pronósticos con su valor y sus datos de entrada. En ese sentido, para incrementar los datos a utilizar con el procedimiento mencionado ya que solo se tienen 60 datos de predicciones, se entrenó un modelo Transformer para pronosticar 360 valores futuros desde el 19/05/2022, y de esta forma se contó con 420 registros para aplicar el proceso de ajuste basado en distancia de Mahalanobis sobre las variables de entrada. Adicionalmente, el autoaprendizaje es un proceso que utiliza la información de los errores de pronóstico con cada conjunto de entrada para incorporar el error esperado asignando dicho valor a través de algún criterio, el cual en el caso de la presente investigación es que el valor de ajuste se asigna con base en la menor distancia de Mahalanobis entre el conjunto de entrada nuevo y el conjunto de entrada de los pronósticos ya generados con el que dicha distancia sea la mínima según los datos obtenidos.

En el pronóstico no se incluye la distancia calculada, lo que se hace es con base en la ecuación 4.4 sumar el valor de error esperado a partir del conjunto de entrada que tuvo una menor distancia de Mahalanobis respecto al conjunto de entrada con el que se va a calcular un nuevo pronóstico, es decir, en la generación de un nuevo pronóstico se evalúa tras haber calculado la distancia de Mahalanobis en todos los registros cual es el que posee el menor valor con respecto a la distancia calculada del conjunto de entrada

para generar una nueva predicción y sumo este error esperado al nuevo pronóstico, bajo el supuesto de que observaciones similares tendrán errores de pronóstico similares.

Por lo tanto, para responder al segundo objetivo específico, se genera el pronóstico de 60 valores futuros usando el modelo Transformer con la integración de autoaprendizaje efectuando una comparación entre los valores reales y los valores de pronóstico para determinar si la precisión mejora o empeora.

Para el tercer objetivo se contrastan los resultados obtenidos con los de otros enfoques de predicción del S&P500, identificando sus ventajas y limitaciones, tomando como base la revisión de literatura realizada y los principales hallazgos encontrados en cuanto a la mejora de modelos predictivos con el uso de inteligencia artificial y la incorporación de técnicas de autoaprendizaje.

5. Capítulo 5. Resultados y discusión

5.1 Desarrollo de objetivo 1. Aplicar el modelo Transformer de series de tiempo para predecir el comportamiento del índice bursátil S&P500

Para este punto se comenzó realizando el preprocesamiento y lectura de la base de datos de contratos de futuros del índice S&P500 desde el 1/12/1997 hasta el 22/09/2023 generando el siguiente dataframe en Google Colab:

Figura 5-1: Dataframe de trabajo.

Fecha	Último	Apertura	Máximo	Mínimo	Vol.
1997-01-12	979.599976	955.50	979.75	955.00	10500.0
1997-02-12	975.099976	980.00	982.00	972.00	10020.0
1997-03-12	977.500000	974.75	983.75	967.75	12160.0
1997-04-12	976.900024	977.50	985.75	972.50	12240.0
1997-05-12	986.599976	976.25	989.00	963.00	14020.0

Fuente: elaboración propia usando Google Colab.

La figura 5-1 muestra los primeros 5 registros del dataframe utilizado. A los datos se les aplica un proceso de normalización, restando la media y desviación estándar del dataset completo para facilitar el entrenamiento del modelo Transformer donde se usa como índice la fecha y las demás variables en formato float.

Las variables último, apertura, máximo, mínimo se encuentran inicialmente en términos de valor del índice S&P500 respectivamente, y el volumen en dólares estadounidenses. Posteriormente se dividen los datos en secuencias de entrada y salida usando tensores de la librería “torch”, se realiza la separación en datos de entrenamiento (70%), y se destina 30% para datos de test (evaluación del rendimiento del modelo).

De este modo, ya contando con la estructura completa para ejecutar el entrenamiento del modelo se utilizó una búsqueda aleatoria para optimizar el número de capas ocultas y los demás hiperparámetros, a diferencia del dropout el cual se probó variándolo de forma manual entre 30% y 5%, junto con el learning rate del RMSProp que se varió entre 0.00005 y 0.00001, y se obtuvo la siguiente configuración:

5.1.1 Arquitectura resumida del modelo implementado

Número de capas ocultas (num_layers) = 1

Dimensión de la rejilla (D) = 32

Número de cabezas de atención (H) = 8

Dimensión de las capas ocultas (hidden_mlp_dim) = 32

Número de características de entrada (inp_features) = 5

Número de encoder aplicados a la entrada (num_layers) = 0

Número de características de salida (out_features) = 1

Número de decoder aplicados a la salida (num_layers) = 1

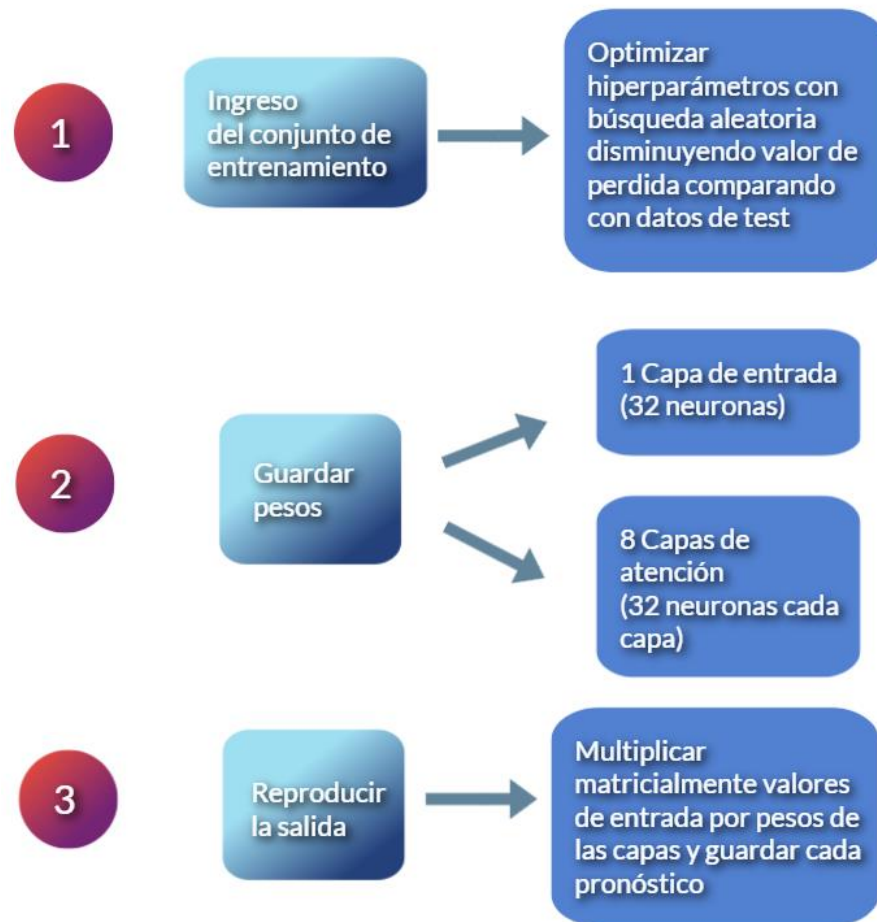
Porcentaje de neuronas apagadas (dropout_rate) = 0.05

Optimizador = torch.optim.RMSprop(transformer.parameters(), lr=0.00005)

*lr = learning rate

Acorde con lo anterior, el flujo que sigue el modelo es el siguiente:

Figura 5-2: Flujo de la implementación posterior a la programación del modelo.

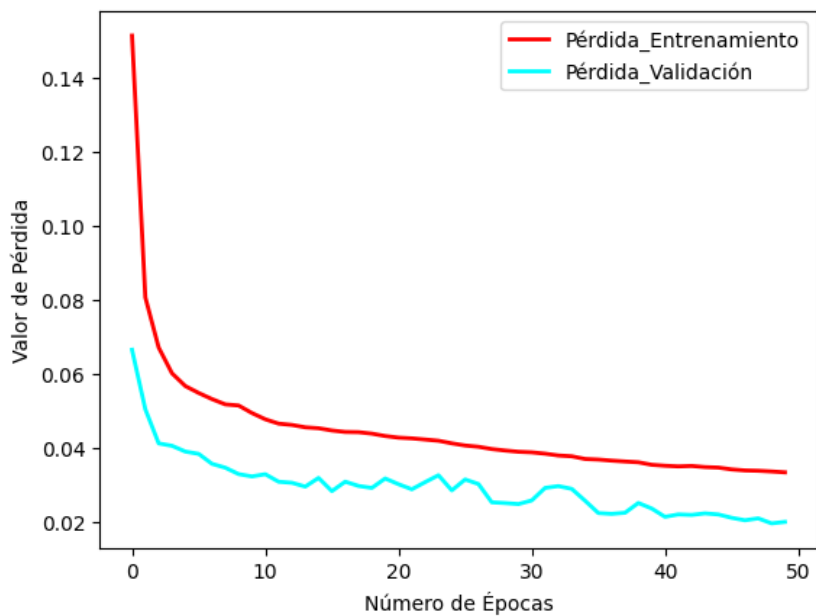


Fuente: elaboración propia usando Google Colab.

La figura 5-2 muestra los pasos a seguir tras la configuración en código Python del modelo Transformer, comenzando por el ingreso del conjunto de entrenamiento, luego ejecutando 50 épocas de entrenamiento con la integración de optimización de hiperparámetros con búsqueda aleatoria disminuyendo el error absoluto en cada iteración. Una vez terminadas las iteraciones se selecciona el mejor modelo según el menor valor de la función de perdida en entrenamiento y test, al mismo tiempo quedan guardados los

pesos de la capa de entrada y las 8 capas de atención seleccionadas en la optimización, posteriormente se apagan el 5% de las neuronas y se realiza la multiplicación matricial de los valores de entrada con los pesos de las neuronas generando el pronóstico de los siguientes 60 valores del precio de cierre del S&P500.

Figura 5-3: Gráfica de la función de pérdida.



Fuente: elaboración propia usando Google Colab.

La figura 5-3 muestra que tras el entrenamiento se obtuvo un valor de 0,0204 para la función de pérdida. Esto quiere decir que, en promedio, las predicciones del modelo “Transformer time series” tienen un error absoluto de aproximadamente 0,0204 unidades en la escala original de los datos.

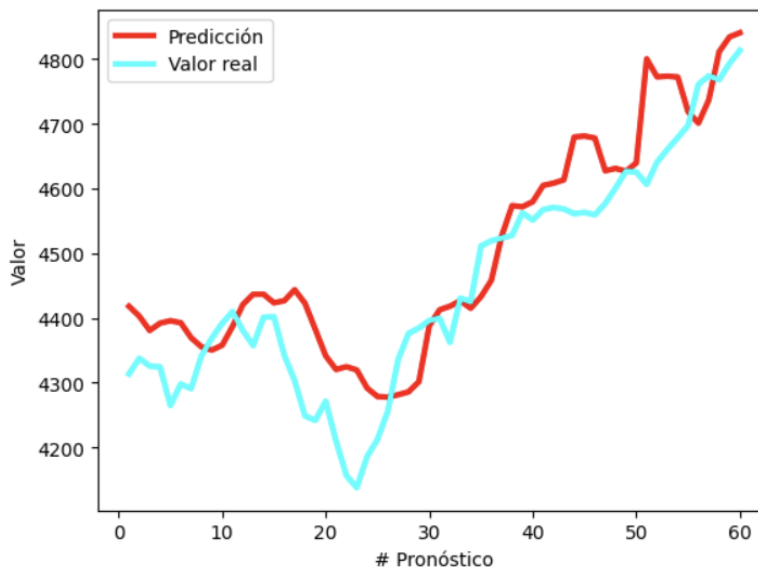
De este modo, con el modelo ya entrenado se genera el pronóstico para los siguientes 60 valores de cierre del S&P500 por fuera del dataset, y se aplica la operación inversa a la normalización efectuada para conocer las predicciones en la escala original:

Tabla 5-1: Valor de los pronósticos generados con Transformer.

Índice	Valor real	Valores predichos
1	4313,5	4417,92
2	4337,5	4402,49
3	4325,5	4380,6
4	4324,25	4391,8
5	4264,75	4395,65
6	4297,75	4392,46
7	4290,75	4369,33
8	4341,5	4355,17
9	4368,75	4350,26
10	4391,5	4357,75
11	4409,75	4386,08
12	4380,5	4420,62
13	4357,25	4436,54
14	4401	4436,62
15	4401,75	4423,25
16	4342,25	4426,55
17	4303	4443,59
18	4248,5	4422,05
19	4241,75	4382,01
20	4271,25	4341,36
21	4209,75	4320,26
22	4156,5	4324,56
23	4137,75	4319,56
24	4185,75	4291,57
25	4212,25	4278,47
26	4256	4277,68
27	4335,75	4281,24
28	4376	4285,77
29	4384,25	4301,39
30	4396	4387,62
31	4399,5	4412,52
32	4362,25	4417,98
33	4430,5	4426,51
34	4425,25	4414,89
35	4511	4432,08
36	4519,25	4458,21
37	4523,25	4526,66
38	4527,5	4573,44
39	4562,25	4571,66
40	4551,25	4579,56
41	4567,25	4604,07
42	4570,75	4608,12
43	4568,25	4613,04
44	4561	4679,05
45	4563	4681,24
46	4559,25	4677,88
47	4576,75	4627,04
48	4600,75	4630,97
49	4626,5	4626,64
50	4625,25	4639,49
51	4606,25	4800,24
52	4640,5	4772,41
53	4660,25	4773,48
54	4678,5	4772,23
55	4697,25	4718,6
56	4760,75	4700,93
57	4774	4736,88
58	4768	4811,27
59	4793	4834,12
60	4813,25	4840,57

Fuente: elaboración propia usando Microsoft Excel.

Figura 5-4: Gráfica de valores reales vs pronósticos usando Transformer.



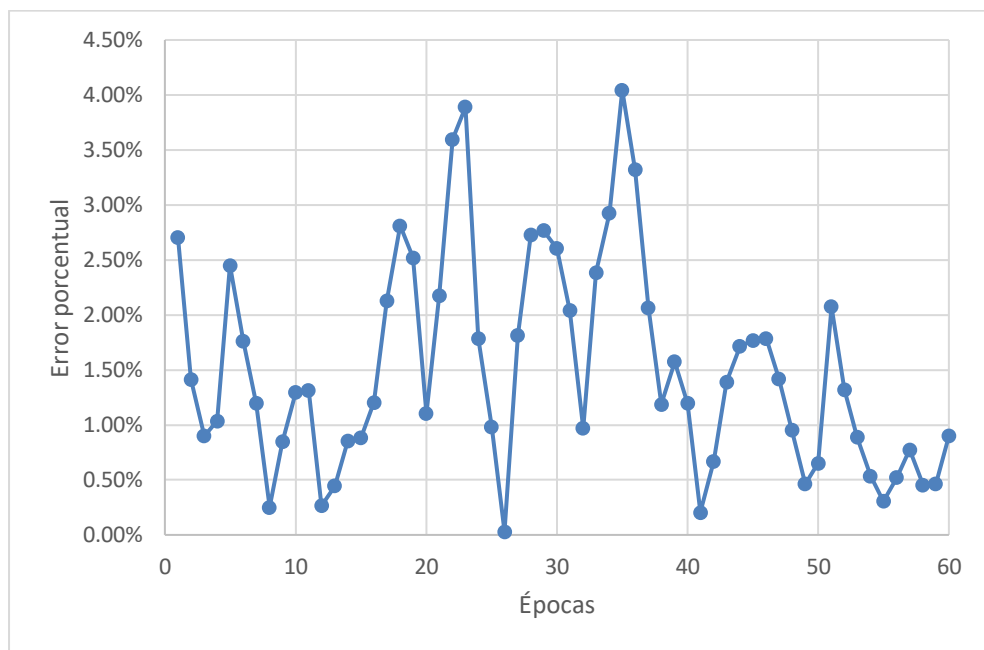
Fuente: elaboración propia usando Google Colab.

La figura 5-4 muestra las predicciones para los siguientes 60 valores de cierre del índice S&P500 junto con los datos reales, donde se puede evidenciar que los pronósticos logran seguir la tendencia de la serie, sin embargo, no capturan completamente la volatilidad del índice o cambios que se pueden presentar con el paso del tiempo, esto se puede mejorar incorporando un modelo de predicción de la volatilidad con el cual se realice un ajuste a la salida.

Se puede evidenciar una tendencia general al alza durante el periodo considerado, la gráfica presenta una volatilidad moderada, con algunos picos de subida y bajada. Se observa un valor máximo de 4840,56 y un valor mínimo de 4277,67, a la vez que se evidencia una forma consistente sin grandes variaciones y valores similares a los reales acorde con la tabla 5-4.

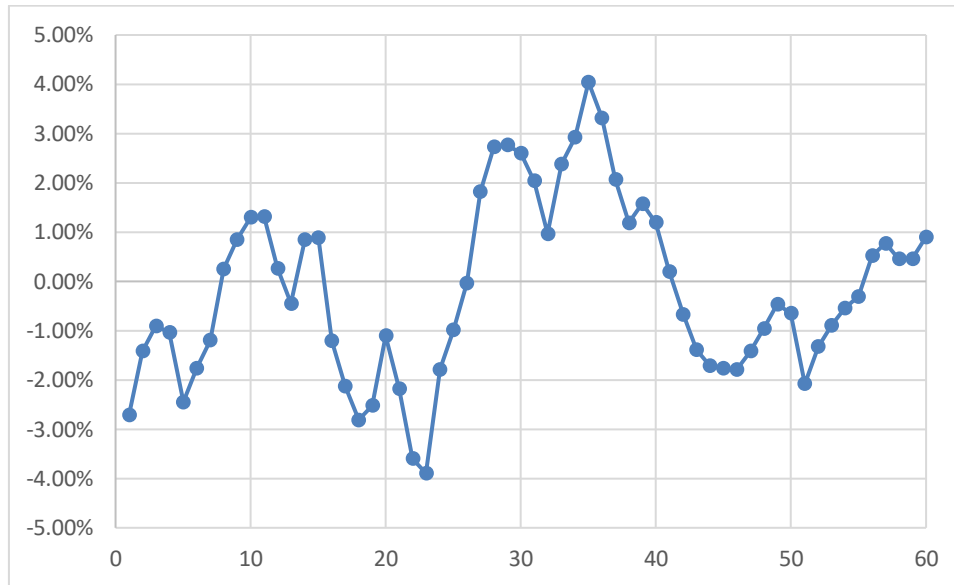
A grandes rasgos se observa que el modelo Transformer logra brindar resultados acordes con la realidad, donde se destaca el hecho de que los pronósticos se alejaron como máximo un 4,04% y se acercaron en algunos casos con una diferencia mínima de 0,03% respecto al dato real como se muestra a continuación:

Figura 5-5: Gráfica de error porcentual para cada pronóstico respecto a las observaciones reales en valor absoluto.



Fuente: elaboración propia usando Microsoft Excel.

Figura 5-6: Gráfica de error porcentual para cada pronóstico respecto a las observaciones reales sin valor absoluto.



Fuente: elaboración propia usando Microsoft Excel.

Las figuras 5-5 y 5-6 muestran para cada una de las predicciones el error, calculado en este caso como la resta entre lo observado y lo predicho dividido lo observado, donde se evidencia un error promedio por pronóstico del 1,52%, esto permite cuantificar las diferencias que pueden existir al aplicarlo y tener una idea del desempeño del modelo a medida que pasa el tiempo para establecer un plan de acción con base en la máxima y mínima precisión que posee en el horizonte analizado.

5.2 Desarrollo de objetivo 2. Incorporar técnicas de autoaprendizaje al modelo Transformer con el fin de incrementar su capacidad predictiva

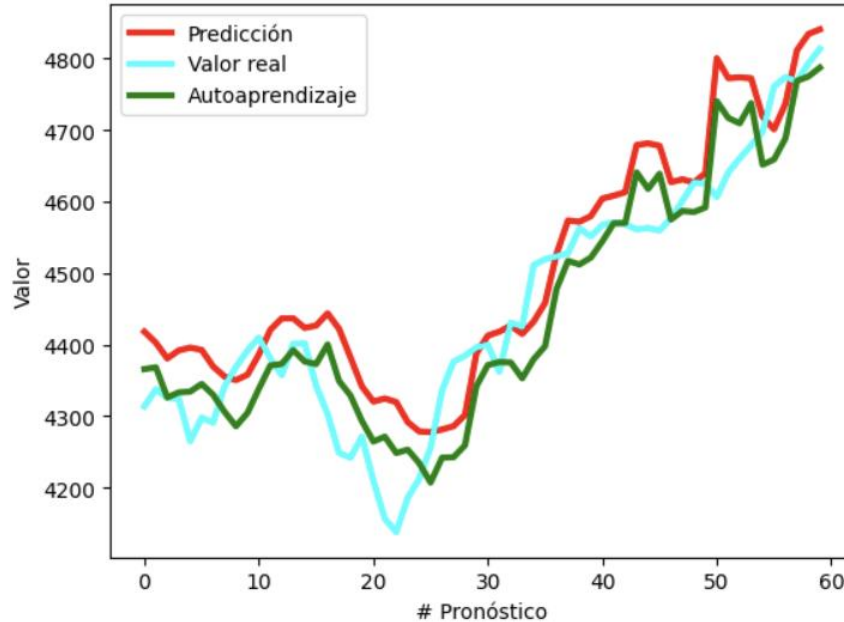
Para este punto se partió de los 60 pronósticos generados con el modelo Transformer, aplicando el proceso de autoaprendizaje basado en distancia de Mahalanobis descrito en la sección de metodología, de este modo, se aplicó el ajuste propuesto a los 60 valores predichos y se obtuvieron los siguientes resultados:

Tabla 5-2: Valor de los pronósticos generados con Transformer y Autoaprendizaje.

Índice	Valor real	Transformer	Autoaprendizaje
1	4313,5	4417,92	4368,77
2	4337,5	4402,49	4362,43
3	4325,5	4380,6	4330,52
4	4324,25	4391,8	4339,59
5	4264,75	4395,65	4342,07
6	4297,75	4392,46	4345,85
7	4290,75	4369,33	4326,84
8	4341,5	4355,17	4307,92
9	4368,75	4350,26	4295,38
10	4391,5	4357,75	4308,34
11	4409,75	4386,08	4339,36
12	4380,5	4420,62	4372,67
13	4357,25	4436,54	4381,68
14	4401	4436,62	4391,31
15	4401,75	4423,25	4376,73
16	4342,25	4426,55	4376,58
17	4303	4443,59	4398,73
18	4248,5	4422,05	4362,59
19	4241,75	4382,01	4332,58
20	4271,25	4341,36	4294,64
21	4209,75	4320,26	4269,86
22	4156,5	4324,56	4275,4
23	4137,75	4319,56	4261,47
24	4185,75	4291,57	4250,01
25	4212,25	4278,47	4234,11
26	4256	4277,68	4220,16
27	4335,75	4281,24	4239,29
28	4376	4285,77	4241,81
29	4384,25	4301,39	4257,86
30	4396	4387,62	4341,83
31	4399,5	4412,52	4368,96
32	4362,25	4417,98	4373,78
33	4430,5	4426,51	4377,71
34	4425,25	4414,89	4360,85
35	4511	4432,08	4382,71
36	4519,25	4458,21	4404,76
37	4523,25	4526,66	4478,61
38	4527,5	4573,44	4521,41
39	4562,25	4571,66	4518,02
40	4551,25	4579,56	4526,78
41	4567,25	4604,07	4550,23
42	4570,75	4608,12	4564,87
43	4568,25	4613,04	4567,52
44	4561	4679,05	4635,54
45	4563	4681,24	4625,03
46	4559,25	4677,88	4633,98
47	4576,75	4627,04	4576,84
48	4600,75	4630,97	4584,79
49	4626,5	4626,64	4581,9
50	4625,25	4639,49	4591,42
51	4606,25	4800,24	4745,23
52	4640,5	4772,41	4719,77
53	4660,25	4773,48	4716,55
54	4678,5	4772,23	4730,05
55	4697,25	4718,6	4660,46
56	4760,75	4700,93	4655,4
57	4774	4736,88	4688,02
58	4768	4811,27	4764,9
59	4793	4834,12	4779,27
60	4813,25	4840,57	4788,66

Fuente: elaboración propia usando Google Colab.

Figura 5-7: Gráfica de valores reales vs pronósticos usando Transformer y Autoaprendizaje basado en distancia de Mahalanobis.

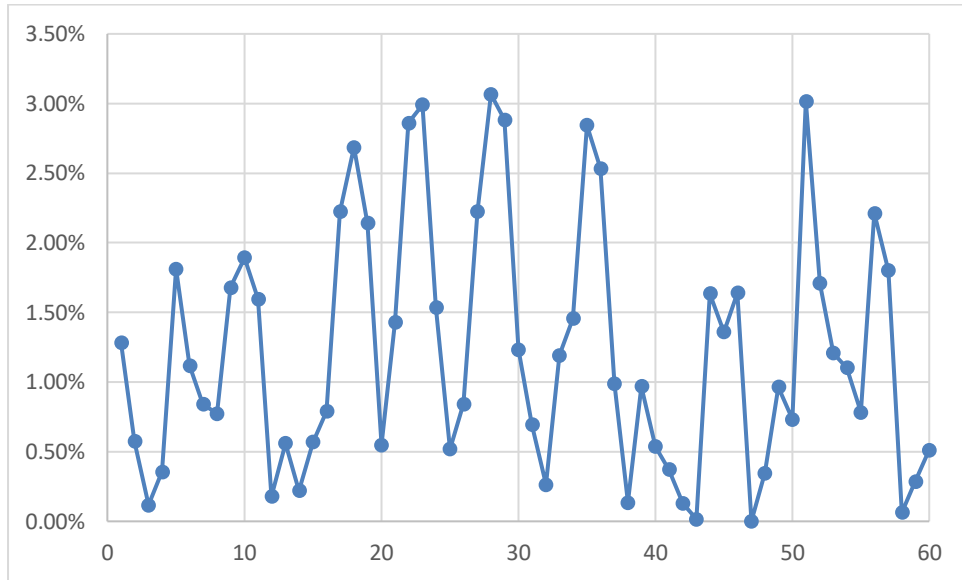


Fuente: elaboración propia usando Google Colab.

La figura 5-7 muestra las predicciones para los siguientes 60 valores de cierre del índice S&P500 junto con los datos reales y los pronósticos luego de aplicar autoaprendizaje, se puede observar que el resultado mejoró y se ajustó de una manera más precisa respecto a los datos reales, siguiendo la tendencia de la serie y su forma, no obstante, sigue existiendo una oportunidad de mejora para lograr con la evolución del proceso de autoaprendizaje un ajuste óptimo y con mayor cantidad de datos de error histórico.

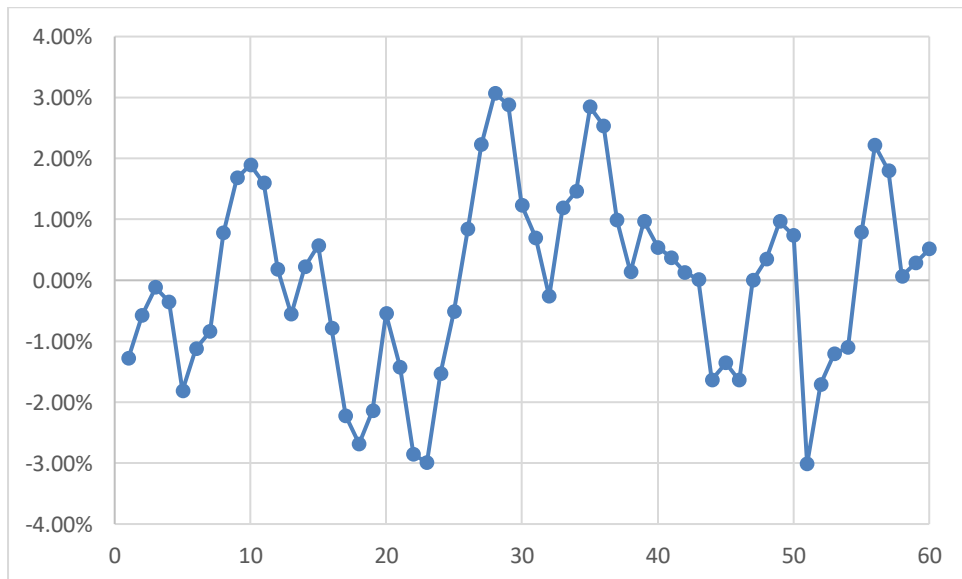
De acuerdo al proceso aplicado se evidencia que el modelo Transformer con la adición de autoaprendizaje logra brindar resultados acordes con la realidad, donde se destaca el hecho de que bajo esta metodología los pronósticos se alejaron como máximo un 3,07% y se acercaron en algunos casos con una diferencia mínima de 0,0019% respecto al dato real como se muestra a continuación:

Figura 5-8: Gráfica de error porcentual para cada pronóstico con autoaprendizaje respecto a las observaciones reales en valor absoluto.



Fuente: elaboración propia usando Microsoft Excel.

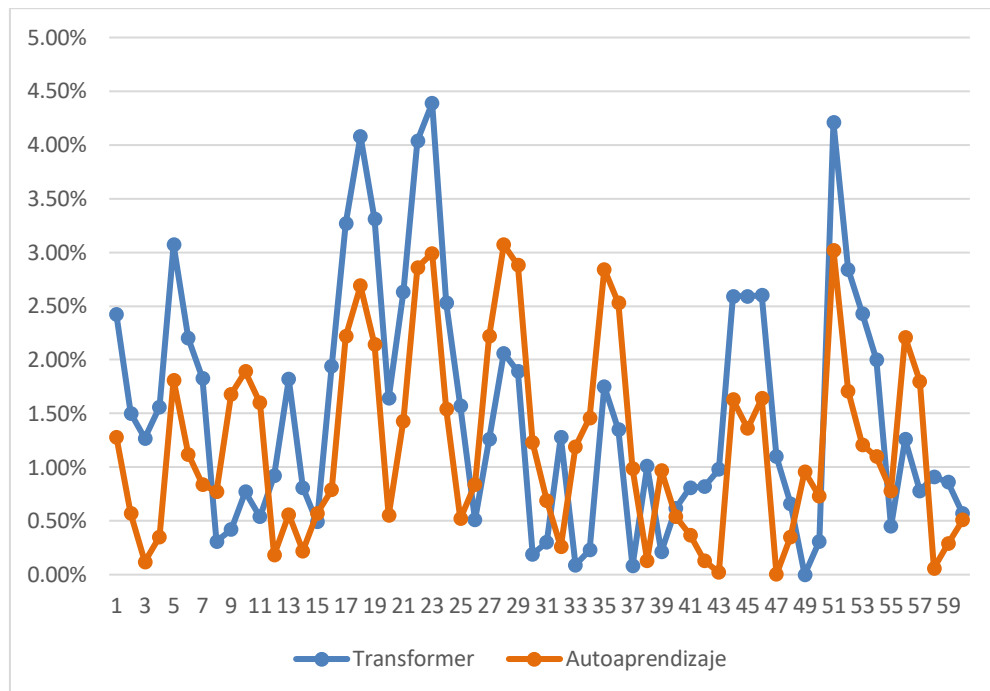
Figura 5-9: Gráfica de error porcentual para cada pronóstico con autoaprendizaje respecto a las observaciones reales sin valor absoluto.



Fuente: elaboración propia usando Microsoft Excel.

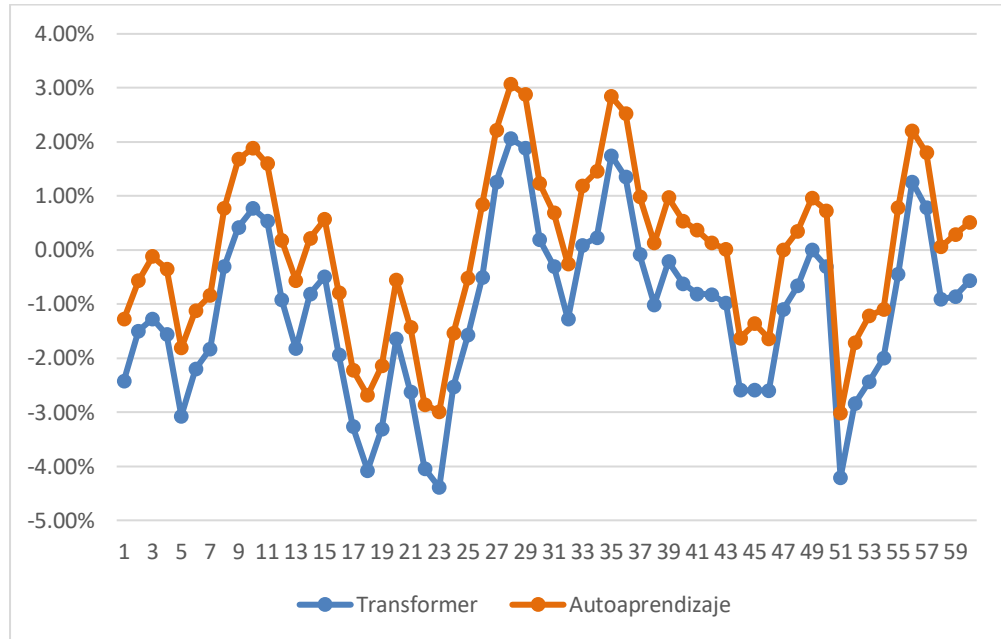
Las figuras 5-8 y 5-9 muestran para cada una de las predicciones un error porcentual promedio por pronóstico del 1,22%, respecto a 1,52% obtenido sin aplicar autoaprendizaje, esto representa una mejora del 19,67% o en otras palabras una reducción del error porcentual en dicha cuantía.

Figura 5-10: Gráfica de error porcentual comparando Transformer y autoaprendizaje respecto a las observaciones reales en valor absoluto.



Fuente: elaboración propia usando Microsoft Excel.

Figura 5-11: Gráfica de error porcentual comparando Transformer y autoaprendizaje respecto a las observaciones reales sin valor absoluto.



Fuente: elaboración propia usando Microsoft Excel.

Las figuras 5-10 y 5-11 permiten comparar los resultados de pronóstico usando únicamente el modelo Transformer y el modelo combinado, en general se evidencia un menor error porcentual de pronóstico con la adición de autoaprendizaje, de este modo, se encuentra una gran oportunidad de mejora en futuras investigaciones profundizando y perfeccionando el método propuesto con una mayor cantidad de datos disponibles y un plan de pruebas que permita sacar el mayor provecho a los registros de errores, bajo el supuesto de que variables de entrada similares producirán errores de pronóstico similares bajo un mismo modelo.

5.3 Desarrollo de objetivo 3. Comparar los resultados obtenidos con los de otros enfoques de predicción del S&P500, identificando sus ventajas y limitaciones

Con relación a los resultados presentados se evidencia un mejor desempeño al comparar con lo expuesto por Kim, Park, Jeon, Song, Bae, Kim & Kang (2021), donde se obtuvo un error medio de pronóstico de los precios de opciones de empresas europeas de alrededor de 3% usando datos de los 3 meses anteriores a partir de la predicción del índice S&P500 con Transformer.

Dichos autores también mencionan que una de las principales limitaciones al realizar predicción de precios es la volatilidad implícita de los activos, la cual varía dependiendo de la liquidez del mercado y el volumen negociado, y aunque está sea cubierta de manera precisa durante un cierto periodo de tiempo, tarde o temprano ocurren eventos macroeconómicos o cambios estructurales en la serie que implican un ajuste minucioso de los modelos o repetir el entrenamiento con los nuevos datos, e incluso interpolar o extrapolar los resultados previos con el nuevo cambio de volatilidad introduciendo ruido que es difícil de controlar.

Por otro lado, en comparación con Kaeley, Qiao & Bagherzadeh (2023), se evidencia una mayor precisión con el método propuesto, debido a que dichos autores obtuvieron un error porcentual del 11% usando Transformer y análisis sentimental aplicado a la predicción del precio de acciones de Facebook, Amazon, Apple, Netflix y Google con un conjunto de 3700 registros, aun así recalcan haber obtenido mejores resultados que al usar redes neuronales recurrentes usando datos de los últimos 30 días para pronosticar el precio del día siguiente.

Respecto a la métrica ($MAE = 0,0204$) utilizada en la presente investigación los resultados fueron similares a los obtenidos por Farahani & Hajiagha (2021), quienes

obtuvieron un menor valor de error absoluto medio de 1×10^{-40} y 1×10^{-52} al incorporar a una red neuronal recurrente los algoritmos metaheurísticos BAT y SSO respectivamente, en comparación con un valor 13,499 del modelo sin la mejora. Así mismo, Barbaglia, Consoli & Wang (2021), realizan la predicción de la varianza del S&P500 a un horizonte de 1 día usando redes neuronales recurrentes indicando la probabilidad de ocurrencia de cada pronóstico, encontrando que el modelo Transformer “bert with punctuation” obtiene el mejor desempeño con un MSE de 0,983.

Los anteriores estudios comparten como limitaciones en común el hecho de que se obtienen mejores resultados con Transformer que al usar métodos tradicionales de Machine learning, sin embargo, la calidad y abundancia de los datos juegan un papel importante en el desempeño de los modelos lo cual no logra superarse simplemente al aplicar métodos sofisticados de optimización de hiperparámetros y limpieza de datos, puesto que como se menciona en Sallab & Rashwan (2011, pp. 224), al aplicar autoaprendizaje se consiguió una mejora en la clasificación de correos electrónicos en categorías como “enviado”, “bandeja de entrada”, y “eliminado” entre otros, pasando de un accuracy promedio de 84,17% a uno de 85,78%, lo cual representa un cambio del 1,91%, en comparación con un 19,67% obtenido en el presente trabajo, por lo que no siempre se mejora significativamente la calidad de los pronósticos al usar optimización o autoaprendizaje.

Comparando lo obtenido por Correa (2010), Rico y Sánchez (2018), Zúñiga (2021), se identifica que los métodos de machine learning tradicionales consiguen un error porcentual de pronóstico de entre 10 a 30%. A su vez, según lo expuesto por Kim, Park, Jeon, Song, Bae, Kim & Kang (2021), Barbaglia, Consoli & Wang (2021), Kaeley, Qiao & Bagherzadeh (2023), los métodos de pronóstico con Transformer consiguen un error porcentual de entre 3 a 15%, en contraste con lo obtenido por Broz y Viego (2014), al aplicar un modelo autorregresivo de media móvil (ARIMA) para predecir el precio de productos forestales industrializables, obteniendo un error porcentual del 12%, que a su vez se puede comparar con lo obtenido por Guzmán y Hurtado (2021), quienes obtuvieron un error porcentual de 37,60% en entrenamiento y 35,96% en test en el pronóstico del

índice S&P500, recalcando que el modelo no lograba predecir nada muy diferente al valor de cierre del día anterior.

De acuerdo con lo anterior, se logra identificar que los modelos Transformer poseen un mejor desempeño que los modelos tradicionales de Machine learning, y estos a su vez un mejor desempeño que los modelos autorregresivos para predicción de series de tiempo, esto puede variar en cada caso y según la rigurosidad con que se hayan aplicado, recolectado y limpiado los datos, a la vez que se hace necesario incorporar métodos de optimización y corrección de errores, por lo que se reitera la importancia de realizar investigaciones que provean cada vez mejores herramientas.

6. Conclusiones y recomendaciones

6.1 Conclusiones

- A la luz de los resultados obtenidos se ha demostrado que el modelo Transformer, originalmente diseñado para tareas de procesamiento de lenguaje natural, puede adaptarse con éxito para la predicción de series de tiempo. A través de su arquitectura basada en el mecanismo de autoatención, el modelo puede capturar patrones complejos y aprender representaciones precisas de una serie temporal, logrando así en la presente investigación un error porcentual promedio de 1,52% y un error porcentual máximo de 4,04% en la predicción de los siguientes 60 días del índice S&P500 usando datos desde el 1/12/1997 hasta el 22/09/2023.
- En vista de que los modelos de pronóstico no aciertan completamente todos los valores se realizó una búsqueda aleatoria para optimizar los hiperparámetros del modelo, incluyendo el número de capas ocultas, la dimensión de la rejilla y el número de cabezas de atención, entre otros. Esta optimización permitió encontrar una configuración que maximizara el rendimiento predictivo, sin embargo, el desempeño se incrementó notablemente al aplicar el método de autoaprendizaje propuesto basado en distancia de Mahalanobis, pasando de un error porcentual promedio de 1,52% a 1,22%, representando una mejora de 19,67%, que fue superior a lo encontrado en la revisión de literatura en estudios similares.
- Al comparar los resultados obtenidos en este estudio con los de investigaciones previas, se ha observado una mejora significativa en la precisión de las predicciones, identificando que los métodos de machine learning tradicionales consiguen un error porcentual de entre 10 a 30%, los métodos de pronóstico con

Transformer consiguen un error porcentual de entre 3 a 15%, en contraste con lo encontrado sobre modelos autorregresivos y métodos tradicionales para series de tiempo con un error porcentual de entre 12% a 35,96%, pero a pesar de las mejoras en la precisión usando métodos modernos, los modelos de predicción, incluido el Transformer, enfrentan desafíos comunes, como la dependencia de la calidad y la cantidad de datos, la sensibilidad a cambios en la volatilidad del mercado y la necesidad de ajustes periódicos para mantener su precisión.

- En general en la presente investigación se identificó que el modelo Transformer para series de tiempo ha demostrado ser superior a los métodos tradicionales de machine learning, así como a los modelos autorregresivos, en términos de precisión en la predicción del índice S&P500. Esto se atribuye a la capacidad del modelo para aprender representaciones complejas de la serie temporal, capturar patrones no lineales y de corto y largo plazo, que en conjunto con optimización de hiperparámetros y autoaprendizaje permiten que se ajuste automáticamente con el paso del tiempo y la ingesta de nuevos datos.
- Los resultados de este estudio subrayan la importancia de la investigación continua en el campo de la predicción numérica utilizando modelos de inteligencia artificial. A medida que surgen nuevos enfoques y técnicas, es fundamental seguir explorando y evaluando la efectividad para mejorar la precisión de las predicciones y brindar herramientas más confiables para la toma de decisiones en el mercado financiero. En resumen, este estudio ha contribuido a la comprensión de la eficacia del modelo Transformer en la predicción del índice S&P500 y ha resaltado su superioridad en comparación con otros enfoques de predicción. Sin embargo, se reconoce la necesidad de continuar investigando y refinando estos modelos para abordar sus limitaciones y mejorar aún más su rendimiento en entornos complejos y dinámicos.

6.2 Recomendaciones

- Se sugiere comenzar la implementación de modelos de pronóstico con un enfoque en la calidad de los datos, debido a que los Transformer y las técnicas de machine learning son muy susceptibles a la presencia de caracteres especiales y tipos de dato no aptos para análisis cuantitativo como en el caso de los textos.
- Es necesario contar con suficiente volumen de registros para que los modelos logren identificar patrones históricos y también se requiere prestar atención a la temporalidad usada, puesto que se pueden obtener errores al intentar realizar predicciones en la temporalidad diaria usando datos semanales o de una temporalidad diferente a la que se tiene de objetivo.
- Más importante que implementar modelos complejos o modernos es realizar un monitoreo constante de sus métricas y como estas evolucionan a medida que transcurre tiempo desde el entrenamiento del modelo, y por eso se recomienda incorporar técnicas que permitan ajustar automáticamente los pronósticos de acuerdo con datos y errores históricos.

A. Anexo: código del entrenamiento del modelo Transformer

##Codigo de entrenamiento con optimización de hiperparámetros

```
import torch
import torch.nn as nn
import math
from kerastuner import HyperParameters
from tqdm import tqdm

# Definición de la función create_look_ahead_mask
def create_look_ahead_mask(size):
    mask = torch.tril(torch.ones((size, size))).to(device)
    return mask
```

##Implementacion de Transformer TSA

```
import torch.nn as nn
import math

class MultiHeadAttention(nn.Module):
    """Multi-head self-attention module"""
    def __init__(self, D, H):
        super(MultiHeadAttention, self).__init__()
        self.H = H # number of heads
        self.D = D # dimension
```

```
self.wq = nn.Linear(D, D*H)
```

```
self.wk = nn.Linear(D, D*H)
```

```
self.wv = nn.Linear(D, D*H)
```

```
self.dense = nn.Linear(D*H, D)
```

```
def concat_heads(self, x):
```

```
    """(B, H, S, D) => (B, S, D*H)"""
```

```
    B, H, S, D = x.shape
```

```
    x = x.permute((0, 2, 1, 3)).contiguous() # (B, S, H, D)
```

```
    x = x.reshape((B, S, H*D)) # (B, S, D*H)
```

```
    return x
```

```
def split_heads(self, x):
```

```
    """(B, S, D*H) => (B, H, S, D)"""
```

```
    B, S, D_H = x.shape
```

```
    x = x.reshape(B, S, self.H, self.D) # (B, S, H, D)
```

```
    x = x.permute((0, 2, 1, 3)) # (B, H, S, D)
```

```
    return x
```

```
def forward(self, x, mask):
```

```
    q = self.wq(x) # (B, S, D*H)
```

```
    k = self.wk(x) # (B, S, D*H)
```

```
    v = self.wv(x) # (B, S, D*H)
```

```
    q = self.split_heads(q) # (B, H, S, D)
```

```
    k = self.split_heads(k) # (B, H, S, D)
```

```
    v = self.split_heads(v) # (B, H, S, D)
```

```
    attention_scores = torch.matmul(q, k.transpose(-1, -2)) #(B,H,S,S)
```

```
    attention_scores = attention_scores / math.sqrt(self.D)
```

```
# add the mask to the scaled tensor.
if mask is not None:
    attention_scores += (mask * -1e9)

attention_weights = nn.Softmax(dim=-1)(attention_scores)
scaled_attention = torch.matmul(attention_weights, v) # (B, H, S, D)
concat_attention = self.concat_heads(scaled_attention) # (B, S, D*H)
output = self.dense(concat_attention) # (B, S, D)

return output, attention_weights

class TransformerLayer(nn.Module):
    def __init__(self, D, H, hidden_mlp_dim, dropout_rate):
        super(TransformerLayer, self).__init__()
        self.dropout_rate = dropout_rate
        self.mlp_hidden = nn.Linear(D, hidden_mlp_dim)
        self.mlp_out = nn.Linear(hidden_mlp_dim, D)
        self.layernorm1 = nn.LayerNorm(D, eps=1e-9)
        self.layernorm2 = nn.LayerNorm(D, eps=1e-9)
        self.dropout1 = nn.Dropout(dropout_rate)
        self.dropout2 = nn.Dropout(dropout_rate)

        self.mha = MultiHeadAttention(D, H)

    def forward(self, x, look_ahead_mask):
        attn, attn_weights = self.mha(x, look_ahead_mask)
        attn = self.dropout1(attn)
        attn = self.layernorm1(attn + x)

        mlp_act = torch.relu(self.mlp_hidden(attn))
        mlp_act = self.mlp_out(mlp_act)
        mlp_act = self.dropout2(mlp_act)

        output = self.layernorm2(mlp_act + attn)
```

```
return output, attn_weights
```

```
class Transformer(nn.Module):
    def __init__(self, num_layers, D, H, hidden_mlp_dim, inp_features, out_features,
dropout_rate):
        super(Transformer, self).__init__()
        self.sqrt_D = torch.tensor(math.sqrt(D)).to(device)
        self.num_layers = num_layers
        self.input_projection = nn.Linear(inp_features, D)
        self.output_projection = nn.Linear(D, out_features)
        self.pos_encoding = positional_encoding(D)
        self.dec_layers = nn.ModuleList([
            TransformerLayer(D, H, hidden_mlp_dim, dropout_rate=dropout_rate)
            for _ in range(num_layers)
        ])
        self.dropout = nn.Dropout(dropout_rate)

    def forward(self, x, mask):
        B, S, D = x.shape
        attention_weights = {}
        x = self.input_projection(x)
        x *= self.sqrt_D

        x += self.pos_encoding[:, :S, :]

        x = self.dropout(x)

        for i in range(self.num_layers):
            x, block = self.dec_layers[i](x=x, look_ahead_mask=mask)
            attention_weights[f'decoder_layer{i + 1}'] = block

        x = self.output_projection(x)
        return x, attention_weights
```



```
def model_builder(hp):
    transformer = Transformer(
        num_layers=hp.Int('num_layers', min_value=1, max_value=5, step=1),
        D=hp.Choice('D', values=[16, 32, 64, 128]),
        H=hp.Choice('H', values=[4, 8, 16]),
        hidden_mlp_dim=hp.Choice('hidden_mlp_dim', values=[16, 32, 64, 128]),
        inp_features=5,
        out_features=1,
        dropout_rate=hp.Float('dropout_rate', min_value=0.0, max_value=0.5, step=0.05)
    ).to(device)

    optimizer = torch.optim.RMSprop(
        transformer.parameters(),
        lr=hp.Choice('learning_rate', values=[0.03, 1e-2, 1e-3, 1e-4])
    )

    return transformer, optimizer

# Entrenamiento con optimización de parámetros
n_epochs = 50
losses, val_losses = []

# Uso de HyperParameters
hp = HyperParameters()

for e in tqdm(range(n_epochs)):
    model, optimizer = model_builder(hp)

    model.train()
    sum_train_loss = 0.0
    for x, y in train_dataset:
        x, y = x.to(device), y.to(device)
        S = x.shape[1]
```

```
mask = create_look_ahead_mask(S)
optimizer.zero_grad()
out, _ = model(x, mask)
loss = torch.nn.L1Loss()(out, y)
sum_train_loss += loss.item()
loss.backward()
optimizer.step()
losses.append(sum_train_loss / len(train_dataset))

model.eval()
sum_val_loss = 0.0
with torch.no_grad():
    for i, (x, y) in enumerate(val_dataset):
        x, y = x.to(device), y.to(device)
        S = x.shape[1]
        mask = create_look_ahead_mask(S)
        out, _ = model(x, mask)
        loss = torch.nn.L1Loss()(out, y)
        sum_val_loss += loss.item()
    val_losses.append(sum_val_loss / (i + 1))

print("Entrenamiento completado.")
print("Pérdida de entrenamiento:", losses)
print("Pérdida de validación:", val_losses)
```

B. Anexo: código para el cálculo de la distancia de Mahalanobis

##Codigo para calcular la distancia de Mahalanobis en cada fila con las variables de entrada con las que se realizaron los pronósticos

```
import pandas as pd
import numpy as np
from scipy.spatial.distance import Mahalanobis
variables = ['Apertura', 'Máximo', 'Mínimo', 'Vol.']
# Eliminar las primeras dos filas de df2
df2 = df2.iloc[2:]
# Calcular la distancia de Mahalanobis para cada fila
mean_vector = df2[variables].mean().values
cov_matrix = df2[variables].cov().values
inv_cov_matrix = np.linalg.inv(cov_matrix)

# Calcular la distancia de Mahalanobis para cada fila
df4 = pd.DataFrame()
df4['D.Mahalanobis2'] = [Mahalanobis(row[variables].values, mean_vector,
inv_cov_matrix) for _, row in df2.iterrows()]

# Restablecer el índice para que comience en cero
df4.reset_index(drop=True, inplace=True)

print(df4)
```


Bibliografía

- Ahmad A. Al Sallab & Mohsen A. Rashwan. (2011). Self learning machines using Deep Networks. Cairo University. Obtenido de:
https://www.researchgate.net/publication/241624295_Self_learning_machines_using_Deep_Networks
- Alvarado., R. E. (2015). Una mirada a la inteligencia artificial. Revista Ingeniería, Matemáticas y Ciencias de la Información, ISSN-e 2357-3716, Vol. 2, Nº. 3. Obtenido de: <https://dialnet.unirioja.es/servlet/articulo?codigo=7894426>
- Ashour., M.A.H y Ahmed., A.S. (2023). Bitcoin prediction with a hybrid model. Periodicals of Engineering and Natural Sciences Original Research Vol. 11, No. 1, pp.186-196. Obtenido de: DOI: 10.21533/pen.v11i1.3452
- Avendaño., Q. J. (2020). Automatización del mecanismo de funcionamiento de la válvula grabber para manejo de tubería. Obtenido de:
http://repositorio.uan.edu.co/bitstream/123456789/7333/3/2022_JohanAlexanderAvenda%C3%B1oQuiroga.pdf
- Azarnejad., A y Khaloozadeh., H. (2022). Stock return system identification and multiple adaptive forecast algorithm for price trend forecasting. Obtenido de: DOI: 10.1016/j.eswa.2022.116685
- Barbaglia L., Consoli S., Wang S. (2021). Financial Forecasting with Word Embeddings Extracted from News: A Preliminary Analysis. Obtenido de:
https://www.researchgate.net/publication/358703709_Financial_Forecasting_with_Word_Embeddings_Extracted_from_News_A_Preliminary_Analysis
- Base de datos Scopus. (2023). Advanced search. Disponible en: <https://www-scopus-com.ezproxy.unal.edu.co/search>
- Beyer D. S., Coussement K., De Caigny A., Pérez L.F & Creemers S. (2023). Do the US president's tweets better predict oil prices? An empirical examination using long short-term memory networks. Obtenido de: <https://www.x-mol.net/paper/article/1675716590483259392>
- Broz., R. D y Viego., N. V. (2014). Predicción de precios de productos de Pinus spp con modelos ARIMA. vol.20 no.1, ISSN 1405-0471. Obtenido de:

https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-04712014000100004

- Cadena., J. B, Ariza., M. J., y Palomo., R. J. (2018). La gestión de pronóstico en las decisiones empresariales: un análisis empírico. *Revista Espacios*, 39(13), 1-1. Obtenido de: https://www.cesa.edu.co/wp-content/uploads/2020/08/2020_Grupode-Investigacion.pdf
- Cai., B y Di., Q. (2023). Different Forecasting Model Comparison for Near Future Crash Prediction. *Applied Sciences (Switzerland)*, 13 (2), art. no. 759. Obtenido de: DOI: 10.3390/app13020759
- Callejas, C. M & Vélez G. M. (2019). Predicción electoral usando un modelo híbrido basado en análisis sentimental: elecciones presidenciales de Colombia. Obtenido de: https://www.researchgate.net/publication/339581272_Prediccion_electoral_usando_un_modelo_hibrido_basado_en_analisis_sentimental_y_seguimiento_a_encuestas_elecciones_presidenciales_de_Colombia
- Castrillón., E. J, García., D. J, Anaya., T. M, Rodríguez., B. D, De la Rosa., B. D y Caballero., U. C. (2008). Bases de datos, motores de búsqueda e índices temáticos: herramientas fundamentales para el ejercicio médico. *Salud Uninorte*, vol. 24, núm. 1, pp. 95-119. Universidad del Norte Barranquilla, Colombia. Obtenido de: <https://www.redalyc.org/pdf/817/81724111.pdf>
- Chen., K, Chen., G, Xu., D, Zhang., L, Huang., Y & Knoll., A. (2021). NAST: Non-Autoregressive Spatial-Temporal Transformer for Time Series Forecasting. Obtenido de: https://www.researchgate.net/publication/349195112_NAST_Non-Autoregressive_Spatial-Temporal_Transformer_for_Time_Series_Forecasting
- Cheng., M. Y, Fang., Y. C y Wang., C. Y. (2021). Auto-tuning SOS Algorithm for Two-Dimensional Orthogonal Cutting Optimization. *KSCE Journal of Civil Engineering*, 25 (10), pp. 3605-3619. Obtenido de: DOI: 10.1007/s12205-021-0522-y
- Cong., N. B, Pérez., R. L & Morell. C. (2015). Aprendizaje supervisado de funciones de distancia: estado del arte. Obtenido de: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992015000200002
- Correa., V. M. (2010). Inteligencia artificial para la predicción y control del acabado superficial en procesos de fresado a alta velocidad. Obtenido de: https://oa.upm.es/37172/1/TD_CORREA_%20VALENCIA_MARITZA.pdf
- Departamento de ciencias de la computación e I. A (DECSAI). (s. f). Modelado de datos. Fundamentos de diseño de bases de datos. Universidad de Granada. Obtenido de: <https://elvex.ugr.es/idbis/db/docs/intro/C%20Modelado%20de%20datos.pdf>

- Derby., S. J. (2005). Design of Automatic Machinery. Obtenido de:
https://www.researchgate.net/publication/329026602_Design_of_Automatic_Machinery
- Díaz., M. L. (2007). Estadística multivariada: inferencia y métodos. Universidad Nacional de Colombia. Sede Bogotá. Departamento de Estadística. Facultad de ciencias. Obtenido de:
<https://repositorio.unal.edu.co/bitstream/handle/unal/79907/Estad%C3%ADstica%20Multivariada%209789587011951.pdf?sequence=2&isAllowed=y>
- Espino., T. C. (2017). Análisis predictivo: técnicas y modelos utilizados y aplicaciones del mismo - herramientas Open Source que permiten su uso. Obtenido de:
<https://openaccess.uoc.edu/bitstream/10609/59565/6/caresptimTFG0117mem%C3%B2ria.pdf>
- Gallegos., M. E. (2013). Métodos de pronósticos para negocios. Editorial digital Tecnológico de Monterrey. Instituto Tecnológico y de Estudios Superiores de Monterrey, México. Obtenido de:
<http://prod77ms.itesm.mx/podcast/EDTM/P196.pdf>
- Gil., R. M. (1994). La base de datos. Importancia y aplicación en educación. Perfiles Educativos, núm. 65. Obtenido de: <https://www.redalyc.org/pdf/132/13206506.pdf>
- Grudniewicz, J. (2023). Application of machine learning in algorithmic investment strategies on global stock markets. Obtenido de:
<https://www.sciencedirect.com/science/article/pii/S0275531923001782>
- Guzmán., A. S y Hurtado., F. J. (2021). Predicción de la tendencia del indicador S&P500. Universidad de Antioquia, Facultad de Ingeniería, Departamento de Posgrados Medellín, Colombia. Obtenido de:
https://bibliotecadigital.udea.edu.co/bitstream/10495/20116/7/GuzmanMarcelan_2021_PrediccionS%26P500.pdf
- Hanke, J. E. and Wichern, D. W. (2014). Business Forecasting (9th ed.). Harlow, England: Pearson Education Limited. Obtenido de:
<https://pdfcoffee.com/qdownload/hanke-john-e-wichern-dean-w-business-forecasting-pearson-2014-pdf-free.html>
- Harford, T. (2014). Big data: are we making a big mistake? The Royal Statistical Society Journal, Volume 11, Issue 5, p. 18. Obtenido de:
<https://rss.onlinelibrary.wiley.com/doi/epdf/10.1111/j.1740-9713.2014.00778.x>
- Hestness., J, Narang., S, Ardalani., N, Diamos., G, Jun., H, Kianinejad., H, Patwary., M, Yang., Y & Zhou., Y. (2017). Deep Learning scaling is predictable, empirically. Baidu Research. Obtenido de: <https://arxiv.org/pdf/1712.00409>

- Hu., J. (2022). Intelligent Decisionmaking System through LSTM Prediction Model and DQN algorithm. Obtenido de: DOI: 10.1109/ICDSCA56264.2022.9988461
- Investing. (2023). Datos históricos S&P 500. Obtenido de:
<https://es.investing.com/indices/us-spx-500-historical-data>
- Irabaruta., J. (2021). Stock Trading with Machine Learning Methods and Technical Indicators. Obtenido de:
https://www.researchgate.net/publication/357049879_Stock_Trading_with_Machine_Learning_Methods_and_Technical_Indicators
- Iyer., S y Mahajan., A. (2023). Short-term Adaptive Forecast Model for TEC over equatorial low latitude region. Dynamics of Atmospheres and Oceans, 101, art. no. 101347. Obtenido de: DOI: 10.1016/j.dynatmoce.2022.101347
- Jaramillo., R. L. (2012). Métodos cualitativos y cuantitativos versus métodos de inteligencia artificial. Obtenido de:
<https://repository.cesa.edu.co/bitstream/handle/10726/428/ADM2012-00624.pdf?sequence=8&isAllowed=y>
- Kaeley., H, Qiao., Y & Bagherzadeh., N. (2023). Support for Stock Trend Prediction Using Transformers and Sentiment Analysis. Obtenido de:
<https://arxiv.org/pdf/2305.14368.pdf>
- Kim H., Park K., Jeon J., Song C., Bae J., Kim Y & Kang M. (2021). Candidate Point Selection using a Self-Attention Mechanism for Generating a Smooth Volatility Surface under the SABR model. Obtenido de:
https://www.researchgate.net/publication/349192855_Candidate_Point_Selection_using_a_Self-Attention_Mechanism_for_Generating_a_Smooth_Volatility_Surface_under_the_SABR_model
- Liang. L & Cai. X. (2022). Time-sequencing European options and pricing with deep learning – Analyzing based on interpretable ALE method. Obtenido de:
<https://www.sciencedirect.com/science/article/abs/pii/S095741742101304X>
- Maisueche C. A. (2019). Utilización del machine learning en la industria 4.0. Máster en ingeniería industrial, escuela de ingenierías industriales Universidad de Valladolid. Obtenido de: <https://core.ac.uk/download/pdf/228074134.pdf>
- Management solutions. (2020). Auto Machine learning, hacia la automatización de los modelos. Obtenido de:
<https://www.managementolutions.com/sites/default/files/publicaciones/esp/auto-machine-learning-sp.pdf>
- Matthew J. Pagea, Joanne E. McKenzie, Patrick M. Bossuyt, Isabelle Boutron, Tammy C. Hoffmann, Cynthia D. Mulrow, Larissa Shamseer, Jennifer M. Tetzlaff, Elie A. Akl,

- Sue E. Brennana, Roger Chou, Julie Glanville, Jeremy M. Grimshaw, Asbjørn Hróbjartsson, Manoj M. Lalu, Tianjing Li, Elizabeth W. Loder, Evan Mayo-Wilson, Steve McDonald, Luke A. McGuinness, Lesley A. Stewart, James Thomas, Andrea C. Tricco, Vivian A. Welch, Penny Whiting, David Moher. (2021). Declaración PRISMA 2020: una guía actualizada para la publicación de revisiones sistemáticas. Obtenido de: <https://www.revespcardiol.org/es-declaracion-prisma-2020-una-guia-articulo-S0300893221002748>
- Mendieta., P. M & Sotolongo., L. C. (2019). El pronóstico y su importancia en la práctica clínica. Obtenido de: <http://scielo.sld.cu/pdf/med/v58n3/1561-302X-med-58-03-e1322.pdf>
- Millea., A. (2021). Deep Reinforcement Learning For Trading. A Critical Survey. Obtenido de: https://www.researchgate.net/publication/356279401_Deep_Reinforcement_Learning_For_Trading-A_Critical_Survey
- Moreno, R. N. (2018). Aplicaciones de la Analítica y la Minería de Datos en la Gestión de Recursos Humanos. Trabajo de grado - Maestría. Universidad Nacional de Colombia. Facultad de Minas, Departamento de Ingeniería de la Organización. Medellín, Colombia. Obtenido de: <https://repositorio.unal.edu.co/handle/unal/68928>
- Moreno., A, Armengol., E, Béjar., J, Belanche., L, Cortés., U, Gavaldá., R, Gimeno., J. M, López., B, Martín., M y Sánchez., M. (1994). Aprendizaje automático. Ediciones de la Universidad Politécnica de Catalunya, SL Jordi Girona Salgado. Barcelona. Obtenido de: <https://upcommons.upc.edu/bitstream/handle/2099.3/36157/9788483019962.pdf>
- Nieto., C. E. (2006). Manufactura y automatización. Ingeniería e Investigación, vol. 26, núm. 3, pp. 120-128. Obtenido de: <https://www.redalyc.org/pdf/643/64326315.pdf>
- Nipun., D. N, Chaspari., T y Behzadan., H. A. (2018). Automated ergonomic risk monitoring using body-mounted sensors and machine learning. Obtenido de: <https://www.sciencedirect.com/science/article/am/pii/S1474034618300454>
- Pérez., P. A, Cruz., J, Guatemala., V. A, Juárez., F. V. (2018). Importancia de los pronósticos en la toma de decisiones en las MIPYMES. Obtenido de: <https://revistageon.unillanos.edu.co/index.php/geon/article/view/17/11>
- Poddubnaya., A.A y Keller., A.V. (2020). "forecast" - Adaptive forecast diagnostic intelligent system for vehicles. IOP Conference Series: Materials Science and Engineering, 819 (1), art. no. 012008. Obtenido de: DOI: 10.1088/1757-899X/819/1/012008

- Pouls, M., Ahuja, N., Glock, K., Meyer, A. (2022). Adaptive forecast-driven repositioning for dynamic ride-sharing. Obtenido de: DOI: 10.1007/s10479-022-04560-3
- Ramírez., H. D. (2018). El machine Learning a través de los tiempos, y los aportes a la humanidad. Universidad Libre Seccional Pereira. Facultad De Ingenierías. Programa De Ingeniería De Sistemas. Obtenido de:
<https://repository.unilibre.edu.co/bitstream/handle/10901/17289/EL%20MACHINE%20LEARNING.pdf?sequence=1>
- Rico., P. A y Sánchez., G. D. (2018). Diseño de un modelo para automatizar la predicción del rendimiento académico en estudiantes del IPN. Revista Iberoamericana para la investigación y el desarrollo educativo, ISSN 2007-7467. Obtenido de:
<https://www.scielo.org.mx/pdf/ride/v8n16/2007-7467-ride-8-16-00246.pdf>
- Sampieri, H. R. (2014). Metodología de la investigación. McGraw-Hill / Interamericana editores, S.A. de C.V. México D.F. ISBN: 978-1-4562-2396-0. Obtenido de:
<https://www.uca.ac.cr/wp-content/uploads/2017/10/Investigacion.pdf>
- Sánchez., M. J. (2010). Cómo realizar una revisión sistemática y un meta-análisis. Obtenido de: <http://www.um.es/metaanalysis/pdf/5030.pdf>
- Shahvaroughi Farahani M & Razavi Hajiagha S.H. (2021). Forecasting stock price using integrated artificial neural network and metaheuristic algorithms compared to time series models. Obtenido de: <https://link.springer.com/article/10.1007/s00500-021-05775-5>
- Suarez., B. L. (2018). Modelo de pronóstico de series de tiempo basado en técnicas de analítica predictiva en la mejora del proceso de definición del plan y presupuesto de ventas. Obtenido de:
<https://repositorio.escuelaing.edu.co/bitstream/handle/001/798/Luis%20Carlos%20Suarez%20Bernal.pdf?sequence=2&isAllowed=y>
- Sun-Hee, K. Christos, F & Hyung, J. Y. (2013). Coercively Adjusted Auto Regression Model for Forecasting in Epilepsy EEG. Computational and Mathematical Methods in Medicine, Volume 2013. Article ID 545613. Obtenido de:
<https://doi.org/10.1155/2013/545613>
- Tieleman., T & Hinton., G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 4, 2012.
- Vaswani., A, Shazeer., N, Parmar., N, Uszkoreit., J, Jones., L, Gomez., A, Kaiser., L, & Polosukhin., I. (2017). Attention Is All You Need. 31st Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA. Obtenido de:
https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

- Visme. (2024). Editor de contenido e imágenes. Obtenido de: <https://my.visme.co/editor>
- Ye J., Gu J., Dash A., Deek F.P & Wang G.G. (2023). Prediction with Time-Series Mixer for the S&P500 Index. Obtenido de: DOI: 10.1109/ICDEW58674.2023.00008
- Yu, N., Chen, J. (2022). Design of Machine Learning Algorithm for Tourism Demand Prediction. Obtenido de: DOI: 10.1155/2022/6352381
- Zhou Q., Qu T., Han Y., Duan F & Zhou, Qibin. (2023). Deep Reinforcement Learning with Comprehensive Reward for Stock Trading. Obtenido de: DOI: 10.1007/978-981-99-1648-1_44
- Zhou., H, Zhang., S, Peng., J, Zhang., S, Li., J, Xiong., H, & Zhang., W. (2021). Informer: Beyond efficient transformer for long sequence timeseries forecasting. In AAAI, 2021.
- Zúñiga., Q. L. (2021). Desarrollo de un modelo predictivo para un sistema de manufactura orientado a la industria 4.0. Pontificia Universidad Javeriana de Colombia, Facultad de Ingeniería, Maestría en Ingeniería Industrial, Bogotá D.C. Obtenido de: <https://repository.javeriana.edu.co/bitstream/handle/10554/55631/123-attachment-1627079269Lina%20Zu%C3%B1iga%20-%20Trabajo%20de%20Grado.pdf?sequence=4>