

Predicción de precios de acciones utilizando LSTM y GRU para el análisis de series temporales.

Álvaro Lafuente Sánchez-Migallón

Grado de Ingeniería Informática

Inteligencia Artificial

Dr. David Isern Alarcón

Dr. Friman Sánchez Castaño

01/2025



Esta obra está sujeta a una licencia de
Reconocimiento – NoComercial - SinObraDerivada

[3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	Predicción de precios de acciones utilizando LSTM y GRU para el análisis de series temporales
Nombre del autor:	Álvaro Lafuente Sánchez-Migallón
Nombre del consultor/a:	Dr. David Isern Alarcón
Nombre del PRA:	Friman Sánchez Castaño
Fecha de entrega (mm/aaaa):	01/2025
Titulación:	Grado de Ingeniería Informática
Área del Trabajo Final:	Inteligencia Artificial
Idioma del trabajo:	Español
Palabras clave	Predicción de acciones, redes neuronales recurrentes, series temporales
Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i>	
<p>Este trabajo se centra en la predicción de precios de acciones utilizando redes neuronales recurrentes (LSTM y GRU) aplicadas a series temporales financieras. Se emplearon datos históricos de Apple Inc., enriquecidos con indicadores técnicos, para construir y evaluar los modelos. Como línea base, se implementó una regresión lineal para establecer un punto de comparación inicial.</p> <p>La metodología incluyó un exhaustivo preprocesamiento de los datos, generación de secuencias temporales y optimización de hiperparámetros mediante <i>Grid Search</i>, considerando el tamaño de la ventana, unidades recurrentes, tamaño de lote y tasa de aprendizaje. Los modelos se evaluaron mediante métricas como MSE y MAE, y se analizaron las diferencias entre LSTM y GRU.</p>	

Los resultados indican que los modelos GRU superaron a los LSTM en términos de precisión, mostrando un menor MSE y MAE, lo cual concuerda con investigaciones previas que destacan la eficiencia de las GRU en la captura de dependencias temporales. Sin embargo, ambos modelos demostraron mejoras significativas respecto a la regresión lineal.

El análisis crítico destacó limitaciones como la falta de integración de variables externas y la dependencia de un conjunto de datos limitado. Se sugieren futuras líneas de investigación que incluyan modelos híbridos y fuentes de datos adicionales, como análisis de sentimiento y factores macroeconómicos, para mejorar la capacidad predictiva de los modelos.

En conclusión, el trabajo valida la efectividad de las redes neuronales recurrentes en la predicción de precios de acciones y establece un marco para investigaciones futuras en este dominio.

Abstract (in English, 250 words or less):

This project focuses on stock price prediction using recurrent neural networks (LSTM and GRU) applied to financial time series. Historical data from Apple Inc., enriched with technical indicators, were used to build and evaluate the models. A linear regression model was implemented as a baseline for initial comparison.

The methodology included comprehensive data preprocessing, temporal sequence generation, and hyperparameter optimization using Grid Search, considering window size, recurrent units, batch size, and learning rate. Models were evaluated using metrics such as MSE and MAE, and differences between LSTM and GRU were analyzed.

Results indicate that GRU models outperformed LSTM in terms of accuracy, showing lower MSE and MAE, aligning with prior research highlighting GRU efficiency in capturing temporal dependencies. Both models showed significant improvements compared to linear regression.

The critical analysis highlighted limitations such as the lack of external variable integration and reliance on a limited dataset. Future research is suggested to explore hybrid models and additional data sources, such as sentiment analysis and macroeconomic factors, to enhance predictive performance.

In conclusion, the study validates the effectiveness of recurrent neural networks in stock price prediction and sets a foundation for future research in this domain.

Índice

1. INTRODUCCIÓN	9
1.1. CONTEXTO Y JUSTIFICACIÓN DEL TRABAJO	9
1.2. OBJETIVOS DEL TRABAJO	10
1.3. ENFOQUE Y MÉTODO SEGUIDO	11
1.4. PLANIFICACIÓN DEL TRABAJO	11
1.5. BREVE SUMARIO DE PRODUCTOS OBTENIDOS	14
1.6. BREVE DESCRIPCIÓN DE LOS OTROS CAPÍTULOS DE LA MEMORIA	14
2. MARCO TEÓRICO.....	16
2.1. PREDICCIÓN DE PRECIOS DE ACCIONES: ESTADO DEL ARTE EN EL USO DE <i>MACHINE LEARNING</i>	16
2.2. REDES NEURONALES RECURRENTE (RNN, LSTM, GRU)	17
3. METODOLOGÍA.....	18
3.1. CONJUNTO DE DATOS.....	18
3.2. PREPROCESAMIENTO DE DATOS.....	19
3.3. GENERACIÓN DE INDICADORES TÉCNICOS	21
3.4. IMPLEMENTACIÓN DE LA LÍNEA BASE CON REGRESIÓN LINEAL.....	24
3.5. PREPARACIÓN DE LOS DATOS PARA REDES NEURONALES	25
3.6. OPTIMIZACIÓN DE HIPERPARÁMETROS	25
4. DESARROLLO Y RESULTADOS	27
4.1. RESULTADOS Y ANÁLISIS DE LA LÍNEA BASE	27
4.1.1. Métricas de rendimiento	27
4.1.2. Visualización de predicciones	27
4.1.3. Interpretación y conclusión	28
4.2. RESULTADOS Y ANÁLISIS DE LOS MODELOS LSTM Y GRU	28
4.2.1. Preparación de los datos	29
4.2.2. Configuración de los modelos LSTM y GRU	29
4.2.3. Resultados	30
4.2.4. Curvas de pérdida durante el entrenamiento	31
4.2.5. Visualización de predicciones	35
4.2.6. Interpretación y conclusión	36
4.3. OPTIMIZACIÓN DE HIPERPARÁMETROS Y EVALUACIÓN DE MODELOS	37
4.3.1. Resultados y análisis.....	38

4.3.2. <i>Conclusión</i>	39
5. CONCLUSIONES	40
6. GLOSARIO	43
7. BIBLIOGRAFÍA	46
8. ANEXOS	48
8.1. RESULTADOS DETALLADOS DE LAS PRUEBAS DE ENTRENAMIENTO PARA LSTM Y GRU	48
8.1.1. <i>Modelo LSTM</i>	48
8.1.2. <i>Modelo GRU</i>	49

Lista de figuras

ILUSTRACIÓN 1: PLAN DE TRABAJO	13
ILUSTRACIÓN 2: PREDICCIÓN DEL PRECIO DE CIERRE CON REGRESIÓN LINEAL.....	28
ILUSTRACIÓN 3: MODELO LSTM: EVOLUCIÓN DE LA PÉRDIDA (5 ÉPOCAS)	32
ILUSTRACIÓN 4: MODELO LSTM: EVOLUCIÓN DE LA PÉRDIDA (7 ÉPOCAS)	33
ILUSTRACIÓN 5: MODELO LSTM: EVOLUCIÓN DE LA PÉRDIDA (10 ÉPOCAS)	33
ILUSTRACIÓN 6: MODELO GRU: EVOLUCIÓN DE LA PÉRDIDA (5 ÉPOCAS).....	34
ILUSTRACIÓN 7: MODELO GRU: EVOLUCIÓN DE LA PÉRDIDA (7 ÉPOCAS).....	34
ILUSTRACIÓN 8: MODELO GRU: EVOLUCIÓN DE LA PÉRDIDA (10 ÉPOCAS).....	35
ILUSTRACIÓN 9: MODELO LSTM: PREDICCIONES VS. VALORES REALES.....	36
ILUSTRACIÓN 10: MODELO GRU: PREDICCIONES VS. VALORES REALES	36
ILUSTRACIÓN 11: COMPARACIÓN DE MSE EN LAS MEJORES CONFIGURACIONES DE GRU Y LSTM.....	40

1. Introducción

1.1. Contexto y justificación del Trabajo

El proyecto se centra en la predicción de precios de acciones utilizando variantes de Redes Neuronales Recurrentes (RNN) que han demostrado ser efectivas para analizar series temporales en el contexto financiero, como LSTM (*Long Short-Term Memory*) y GRU (*Gated Recurrent Units*) [1]. Estas arquitecturas son adecuadas para este tipo de análisis debido a su capacidad para capturar dependencias de largo plazo en datos secuenciales, lo cual es esencial en mercados bursátiles, donde las tendencias de precios pueden depender de valores pasados y de patrones históricos complejos [2]. Por ello, se descartan otras variantes de RNN que no se adecuan a las características específicas de este entorno de alta volatilidad y su necesidad de modelar relaciones a lo largo del tiempo.

Estudios recientes han destacado la utilidad de las RNN, en particular LSTM y GRU, para analizar series temporales financieras. En el trabajo de Fischer y Krauss (2018), se observó que las redes LSTM superan a los métodos tradicionales de regresión y análisis técnico en la predicción de precios de acciones, ya que pueden capturar patrones de no linealidad inherentes a los datos financieros [3]. Gao, Wang y Zhou (2021) también destacan que, tras optimizaciones específicas, los modelos LSTM y GRU pueden reducir errores de predicción en comparación con otros enfoques [1]. A nivel de series temporales, el uso de LSTM ha sido efectivo no solo en mercados bursátiles sino también en dominios como el análisis de señales en salud y meteorología, lo cual refuerza su aplicabilidad en contextos donde las dependencias temporales son críticas [4].

El objetivo principal de este proyecto es desarrollar un modelo predictivo que, a partir de datos históricos de precios de acciones (apertura, cierre, volumen, máximos y mínimos), sea capaz de anticipar movimientos de precios. Este tipo de información es de gran valor tanto para inversores individuales como para instituciones financieras, ya que proporciona una base para la toma de decisiones estratégicas en mercados complejos y de alta volatilidad. Sin embargo, el mercado bursátil es inherentemente volátil y está influido por factores externos difíciles de modelar, como eventos económicos globales y cambios repentinos en la política económica.

Con estas limitaciones en mente, el proyecto no solo busca desarrollar y entrenar modelos LSTM y GRU, sino también evaluar su efectividad en comparación con enfoques tradicionales, como la regresión lineal, que, aunque útil en algunas circunstancias, tiende a fallar en entornos de alta volatilidad [5]. Este análisis crítico permitirá explorar la viabilidad de implementar estas redes en predicciones de corto plazo, donde las dependencias temporales en los datos son más significativas y pueden aportar valor adicional en el proceso de toma de decisiones.

1.2. Objetivos del Trabajo

Objetivo General: Desarrollar un modelo de predicción de precios de acciones que emplee redes LSTM y GRU, comparando su rendimiento con métodos tradicionales de regresión lineal y evaluando su efectividad en la predicción de series temporales financieras.

Objetivos Específicos:

- Evaluar la capacidad de LSTM y GRU para modelar series temporales de datos financieros y predecir el precio de acciones, centrándose en las características de los datos disponibles.
- Comparar el rendimiento de las arquitecturas LSTM y GRU con el de un modelo de regresión lineal, utilizando métricas de error como MSE (Mean Squared Error) y MAE (Mean Absolute Error).
- Analizar el impacto de incluir indicadores técnicos (como medias móviles y RSI) en el rendimiento de los modelos.
- Identificar las limitaciones de los modelos en entornos de alta volatilidad y, en función de los resultados obtenidos, ofrecer recomendaciones para posibles futuras mejoras.
- Establecer recomendaciones prácticas sobre la aplicabilidad de LSTM y GRU para la predicción de precios de acciones en entornos reales.

1.3. Enfoque y método seguido

Los precios son notoriamente difíciles de predecir debido a la alta volatilidad y numerosos factores externos, como eventos económicos, medios globales y procesos de mercado. A pesar de esto, los métodos más antiguos de análisis técnico básico pocas veces abordan la complejidad del mercado y la no linealidad de los datos. Por lo tanto, los inversores y comerciantes buscan nuevas herramientas impulsadas por los avances en el aprendizaje automático y el análisis de *big data*.

Para este proyecto, se utilizarán datos históricos de precios de acciones del NASDAQ que incluyen variables como la fecha de transacción, precios de apertura y cierre, precios máximo y mínimo diarios, y volumen de transacciones [6]. Estos valores son fundamentales para entender el comportamiento de las acciones en el mercado, ya que reflejan tanto la actividad de compra y venta como las variaciones de precio que se producen dentro de cada sesión de mercado.

Con estos datos, los modelos LSTM y GRU procesarán la secuencia histórica de precios y volúmenes para aprender patrones que se puedan relacionar con los movimientos de precios futuros. Al emplear una estructura de series temporales, se espera que estos modelos capten las relaciones temporales inherentes a los datos financieros, ofreciendo una predicción más ajustada en comparación con enfoques lineales, y evaluando así su precisión en entornos de alta volatilidad.

1.4. Planificación del Trabajo

- **Fase 0: Definición de los contenidos del trabajo** (25/09/2024 - 07/10/2024).
 - Definir la temática del TFG relacionada con la predicción de precios de acciones utilizando técnicas de *machine learning*.
 - Elaborar una propuesta inicial que incluya la justificación del tema, los objetivos generales y específicos.
 - Buscar y analizar estudios previos para entender el estado del arte en predicción de series temporales y machine learning en el ámbito bursátil.

- **Hito 1:** Propuesta del TFG presentada y feedback inicial recibido.
- **Fase 1: Plan de trabajo** (08/10/2024 – 22/10/2024).
 - Redactar el plan de trabajo, detallando los objetivos del proyecto, la metodología a seguir y la planificación temporal.
 - Acordar el plan de trabajo con el tutor y realizar ajustes en función del feedback recibido.
 - **Hito 2:** Plan de trabajo finalizado y entregado.
- **Fase 2: Desarrollo del trabajo** (23/10/2024 - 27/11/2024)
 - **23/10 - 31/10:** Revisión de literatura y búsqueda de datasets en *Kaggle*, Yahoo Finanzas y *Alpha Vantage*, asegurando la calidad y la adecuación de los datos para el análisis.
 - **01/11 - 09/11:** Preprocesamiento de datos y selección de indicadores técnicos que se incluirán como características.
 - **10/11 - 19/11:** Implementación inicial del modelo de regresión lineal para establecer una línea base.
 - **20/11 - 27/11:** Implementación de un modelo LSTM y un modelo GRU, ajustando hiperparámetros básicos.
 - **Hito 3:** Implementación completa de los modelos de regresión lineal, LSTM y GRU, y preprocesamiento de datos.
- **Fase 3: Desarrollo del trabajo** (28/11/2024 - 23/12/2024)
 - **28/11 - 05/12:** Entrenamiento de modelos LSTM y GRU, comparando su rendimiento con la línea base de la regresión lineal.
 - **06/12 - 15/12:** Optimización de hiperparámetros y evaluación de modelos utilizando técnicas como validación cruzada.

- **16/12 - 23/12:** Análisis de resultados y comparación con estudios previos, buscando identificar mejoras y limitaciones del enfoque.
- **Hito 4:** Comparación de resultados entre LSTM, GRU y regresión lineal, análisis de métricas de error.
- **Fase 4: Redacción de la memoria (25/09/2024 - 12/01/2025)**
 - **25/09 - 03/01:** Redacción del marco teórico, descripción de la metodología y análisis de resultados.
 - **04/01 - 12/01:** Revisión del documento y ajustes finales, inclusión de referencias bibliográficas gestionadas mediante Mendeley o Zotero, asegurando el uso de la plantilla oficial de la UOC.
 - **Hito 5:** Versión preliminar de la memoria para revisión.
- **Fase 5: Presentación y defensa ante el tribunal (13/01/2025 - 26/01/2025)**
 - **13/01 - 18/01:** Preparación de la presentación para la defensa, incluyendo gráficos y visualizaciones clave del desempeño de los modelos.
 - **19/01 - 26/01:** Simulación de la defensa, revisión de preguntas y presentación final.
 - **Hito 6:** Entrega final de la memoria y defensa del TFG.



Ilustración 1: Plan de trabajo

1.5. Breve resumen de productos obtenidos

- **PEC1 (Plan de trabajo):** Definición del TFG, objetivos generales y específicos, planificación con hitos y temporalización.
- **PEC2:** Implementación inicial de los modelos y preprocesamiento de datos.
- **PEC3:** Evaluación de los modelos, ajuste de hiperparámetros y análisis de resultados.
- **PEC4:** Redacción de la memoria y elaboración de conclusiones.
- **Entrega final:** Documento completo del TFG, incluyendo memoria, código y presentación.

1.6. Breve descripción de los otros capítulos de la memoria

- **Introducción**
 - Contexto y justificación del trabajo: Descripción del mercado bursátil y la relevancia de la predicción de precios de acciones. Discusión sobre la necesidad de herramientas predictivas más avanzadas y la motivación para utilizar LSTM y GRU.
 - Objetivos del trabajo: Definición clara de los objetivos generales y específicos, incluyendo el desarrollo de modelos predictivos y la comparación de su rendimiento.
 - Enfoque y método seguido: Justificación de la elección de LSTM y GRU como enfoques principales para la predicción de series temporales, y comparación con métodos tradicionales como la regresión lineal.
 - Planificación del trabajo: Resumen de la organización temporal del proyecto, destacando las diferentes fases de desarrollo y evaluación.

- Breve resumen de productos obtenidos: Resumen de los resultados esperados, como la implementación de los modelos y la obtención de métricas de rendimiento.
- **Marco Teórico**
 - Revisión del estado del arte en la predicción de precios de acciones utilizando *machine learning*.
 - Explicación de redes neuronales recurrentes, con especial énfasis en LSTM y GRU.
- **Metodología**
 - Selección y descripción de los datos: Detalle sobre la recolección de datos históricos de precios de acciones e indicadores técnicos.
 - Preprocesamiento de datos: Procedimientos de limpieza, normalización y generación de características.
 - Implementación de modelos: Descripción detallada de la implementación de los modelos LSTM, GRU y la regresión lineal.
 - Estrategia de evaluación: Descripción de las métricas de evaluación y el enfoque utilizado para comparar el rendimiento de los modelos.
- **Desarrollo y Resultados**
 - Implementación y ajuste de modelos: Detalle del proceso de entrenamiento y ajuste de hiperparámetros de los modelos LSTM y GRU.
 - Comparación de resultados: Presentación de los resultados de cada modelo, utilizando gráficos y métricas como MAE y MSE.
 - Análisis crítico: Discusión sobre los resultados obtenidos, sus implicaciones y cómo se comparan con estudios previos.
- **Conclusiones**

- Resumen de los principales hallazgos del proyecto y lecciones aprendidas.
 - Reflexión crítica sobre el logro de los objetivos planteados inicialmente.
 - Evaluación del seguimiento de la planificación y la efectividad de la metodología aplicada.
 - Recomendaciones para futuras investigaciones y potenciales mejoras de los modelos utilizados.
- **Glosario**
 - Definición de términos técnicos y acrónimos utilizados a lo largo de la memoria, como LSTM, GRU, MAE, MSE, entre otros.
- **Bibliografía**
 - Relación completa de las fuentes bibliográficas consultadas, citadas de forma adecuada utilizando herramientas como Mendeley o Zotero.
- **Anexos**
 - Material adicional relevante para el desarrollo del proyecto, como detalles técnicos de la implementación, código fuente, o resultados complementarios que no se incluyen en el cuerpo principal de la memoria.

2. Marco teórico

2.1. Predicción de precios de acciones: Estado del arte en el uso de *Machine Learning*

La predicción de precios de acciones ha sido un tema ampliamente investigado debido a la importancia que tiene para los inversores y los mercados financieros en general. En las últimas décadas, el uso de técnicas de aprendizaje automático (machine learning) ha ganado popularidad para predecir el comportamiento de los precios de las acciones debido a su capacidad para procesar grandes volúmenes de datos y encontrar patrones no lineales que no son fácilmente detectables por métodos tradicionales.

- **Modelos tradicionales vs. modelos basados en machine learning**

Los métodos tradicionales de análisis técnico, como el análisis de series temporales y el uso de indicadores financieros, han sido utilizados por décadas. Sin embargo, estos modelos a menudo no capturan la complejidad y no linealidad del mercado financiero. Por el contrario, los métodos basados en aprendizaje automático, especialmente las redes neuronales, han mostrado una gran capacidad para modelar relaciones complejas y no lineales entre los factores que afectan el precio de las acciones [3].

- **Modelos de machine learning utilizados en predicción financiera**

En la predicción de precios de acciones, se han utilizado una amplia gama de modelos de machine learning. Entre ellos destacan los modelos lineales como la regresión lineal, los modelos basados en árboles de decisión como *Random Forest* y *XGBoost*, y los modelos de redes neuronales. Sin embargo, las redes neuronales recurrentes (RNN) y sus variantes, como LSTM (*Long Short-Term Memory*) y GRU (*Gated Recurrent Unit*), han sido particularmente exitosas en la predicción de series temporales debido a su capacidad para capturar dependencias temporales y a largo plazo en los datos [4][20].

2.2. Redes neuronales recurrentes (RNN, LSTM, GRU)

Las **redes neuronales recurrentes** (RNNs) son un tipo de red neuronal especialmente diseñada para trabajar con datos secuenciales o temporales. A diferencia de las redes neuronales tradicionales, que asumen que las entradas son independientes entre sí, las RNNs tienen conexiones cíclicas que permiten que la salida de una capa se retroalimente como entrada para la siguiente capa, lo que les permite recordar información de pasos anteriores. Esto las convierte en una herramienta poderosa para la predicción de series temporales, como los precios de acciones [14].

- **Alternativas:** Aunque existen otros enfoques para la predicción de series temporales, como las máquinas de soporte vectorial o las redes neuronales tradicionales, las RNNs y sus variantes, como LSTM (*Long Short-Term Memory*) y GRU (*Gated Recurrent Unit*), son las más adecuadas para modelar las dependencias temporales a largo plazo en los datos financieros [3][4].

- **Criterios de decisión:** Se seleccionaron las RNNs debido a su capacidad única para modelar dependencias temporales, lo que es esencial para el análisis de datos financieros. Además, se eligieron las variantes LSTM y GRU, que están diseñadas para superar las limitaciones de las RNNs tradicionales, como el problema del desvanecimiento del gradiente [19].
- **Decisión tomada:** Se optó por utilizar LSTM y GRU debido a su capacidad para manejar secuencias largas y a su mejor desempeño en comparación con modelos más simples, como la regresión lineal, en tareas de predicción de series temporales, especialmente en entornos de alta volatilidad como los mercados financieros [20].

3. Metodología

3.1. Conjunto de datos

El conjunto de datos utilizado en este proyecto contiene información histórica de los precios de acciones de Apple, Inc. (*AAPL ticker*), obtenida del *dataset* público *Stock Market Dataset* en la plataforma *Kaggle* [11]. Este dataset incluye datos financieros diarios desde el 12 de diciembre de 1980 hasta el 1 de abril de 2020, abarcando casi 40 años de actividad bursátil.

- **Estructura del dataset**

El dataset incluye las siguientes columnas:

- *Date*: Fecha de la transacción bursátil.
- *Open*: Precio de la apertura de acción.
- *High*: Precio máximo alcanzado durante el día.
- *Low*: Precio mínimo alcanzado durante el día.
- *Close*: Precio de cierre ajustado para reflejar divisiones de acciones.

- *Adj Close*: Precio de cierre ajustado tanto para divisiones como para dividendos.
- *Volume*: Número de acciones negociadas durante el día.

- **Tamaño y formato**

El dataset tiene un tamaño aproximado de 2.75 GB para todos los tickers en NASDAQ y se almacenan en formato CSV. Para reducir la complejidad y el análisis y ajustarse a los objetivos del proyecto, se seleccionó exclusivamente el archivo correspondiente a Apple, Inc. (AAPL.csv), que incluye miles de registros diarios de dicha empresa.

- **Razonamiento detrás de la selección**

Apple, Inc. fue seleccionada como caso de estudio por las siguientes razones:

- Alta liquidez: Las acciones de Apple son altamente negociadas, lo que las hace representativas de un mercado eficiente.
- Evolución histórica significativa: Desde su salida a bolsa, Apple ha mostrado múltiples tendencias, eventos significativos y periodos de alta volatilidad, ofreciendo un caso ideal para evaluar modelos predictivos.
- Disponibilidad de los datos: Los datos de Apple están ampliamente documentados y son fáciles de obtener desde múltiples fuentes, lo que facilita la validación del análisis.

3.2. Preprocesamiento de datos

Para trabajar con los datos históricos de la empresa seleccionada, se realizó un preprocesamiento exhaustivo. Este proceso se diseñó para garantizar la calidad y consistencia de los datos, preparándolos para su uso en modelos predictivos.

- **Conversión de fechas:** La columna Date, que especifica la fecha de cada transacción, se transformó al formato de fecha estándar y se estableció como

índice principal del conjunto de datos. Esta operación facilita la manipulación y el análisis temporal de las series de precios históricos.

- **Verificación de valores nulos:** Se verificó la existencia de valores faltantes en las columnas clave (*Open*, *High*, *Low*, *Close*, *Adj Close*, *Volume*). Este paso fue crucial para evitar inconsistencias en el conjunto de datos. No se encontraron valores nulos, lo que permitió proceder directamente con los cálculos posteriores.
- **Normalización de datos mediante Min-Max Scaling:** Para garantizar que todas las variables estuvieran en una escala comparable, se aplicó el método de escalado Min-Max a las columnas de precios (*Open*, *High*, *Low*, *Close*, *Adj Close*) y al volumen (*Volume*). Este método es particularmente adecuado cuando los datos tienen un rango conocido o limitado, como es el caso de los precios y volúmenes en los mercados financieros.

Fórmula:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Donde:

- X' : Valor escalado entre 0 y 1.
- X : Valor original.
- X_{min} : Valor mínimo de la columna.
- X_{max} : Valor máximo de la columna.

Este proceso garantiza que todas las variables tengan el mismo rango, lo cual mejora la estabilidad numérica y el rendimiento de los algoritmos de aprendizaje automático. Además, evita que las variables con rangos más amplios dominen a las de menor rango [5][7].

3.3. Generación de indicadores técnicos

A partir de los precios normalizados y el volumen de transacciones, se generaron una serie de indicadores técnicos. Estos indicadores se seleccionaron por su relevancia en el análisis financiero y su potencial para mejorar la precisión de las predicciones de precios.

- **Media Móvil Simple (SMA)**

La SMA calcula el promedio de los precios de cierre durante una ventana de tiempo específica, ayudando a identificar tendencias generales del mercado [8]. Este indicador se implementó para ventanas de 5, 10 y 20 días.

Fórmula:

$$SMA_t = \frac{1}{n} \sum_{i=0}^{n-1} P_{t-i}$$

Donde:

- SMA_t : Media móvil simple en el tiempo t .
- n : Tamaño de la ventana temporal.
- P_{t-i} : Precio de cierre en el día $t - i$.

- **Media Móvil Exponencial (EMA)**

La EMA, al igual que la SMA, mide las tendencias generales del mercado, pero da mayor peso a los datos más recientes, lo que la hace más sensible a los cambios rápidos [9]. Este indicador se calculó para una ventana de 10 días.

Fórmula:

$$EMA_t = \alpha \cdot P_t + (1 - \alpha) \cdot EMA_{t-1}$$

Donde:

- $\alpha = \frac{2}{n+1}$: Factor de suavizado.
- EMA_t : Media móvil exponencial en el tiempo t .

- **Índice de Fuerza Relativa (RSI)**

EL RSI mide la velocidad y magnitud de los cambios de precios para identificar zonas de sobrecompra ($RSI > 70$) o sobreventa ($RSI < 30$) Este indicador fue introducido por J. Welles Wilder en su libro *New Concepts in Technical Trading Systems*, donde se establece un período estándar de 14 días para su cálculo [9].

Fórmula:

$$RSI = 100 - \frac{100}{1 + RS}$$

Donde:

- $RS = \frac{\text{Ganancia Promedio}}{\text{Pérdida Promedio}}$: Relación entre ganancias y pérdidas promedio en los últimos n días.

- **Volatilidad histórica**

Este indicador refleja la variabilidad del precio en una acción, calculada como la desviación estándar de los precios de cierre en una ventana de 10 días [10]. La volatilidad histórica se utiliza comúnmente para medir el riesgo asociado a un activo financiero.

Fórmula:

$$\sigma_t = \sqrt{\frac{\sum_{i=0}^{n-1} (P_{t-i} - \mu)^2}{n}}$$

Donde:

- σ_t : Volatilidad histórica en el tiempo t .
- μ : Media de los precios en la ventana.

- **Rango Verdadero Promedio (ATR)**

El ATR mide la volatilidad diaria del precio considerando máximos, mínimos y precios de cierre. Es ampliamente utilizado en mercados de alta fluctuación para evaluar el riesgo y determinar niveles de entrada y salida en estrategias de trading [8]. Fue calculado para una ventana de 14 días.

Fórmula:

$$ATR = \frac{1}{n} \sum_{i=0}^{n-1} TR_i$$

Donde:

- $TR_i = \max (alto-bajo, |alto-cierre\ previo|, |bajo-cierre\ previo|)$: Rango verdadero.

Tras el preprocesamiento de los datos originales e inclusión de los indicadores técnicos, se generaron valores nulos en las columnas de los indicadores debido a los cálculos basados en ventanas. Estas filas iniciales, sin datos completos para realizar los cálculos, fueron eliminadas para mantener la consistencia del conjunto de datos.

Una vez limpiadas las filas con valores nulos, los indicadores técnicos se normalizaron utilizando el método de escalado Min-Max. Este proceso asegura que los valores de los indicadores (SMA_5 , EMA_{10} , RSI_{14} , etc.) estén en el mismo rango $[0,1]$, permitiendo que todos los datos sean comparables y facilitando su uso en modelos predictivos. Este paso también es esencial para evitar que indicadores con valores más amplios, como el ATR o la volatilidad histórica, influyan desproporcionadamente en el modelo [5][7].

3.4. Implementación de la línea base con regresión lineal

Para establecer una línea base en el análisis predictivo de los precios de acciones, se ha implementado un modelo de regresión lineal simple. Este modelo tiene como objetivo predecir el precio de cierre ajustado (*Close*) del día siguiente a partir de los datos preprocesados y los indicadores técnicos generados previamente. La regresión lineal fue seleccionada debido a su simplicidad y a su capacidad para servir como punto de comparación con modelos más avanzados como las redes LSTM y GRU [12].

- **Variable objetivo (*Target*):** Se ha definido como el precio de cierre del día siguiente, obtenido desplazando la columna *Close* una posición hacia arriba en el conjunto de datos. Esto es una práctica común en problemas de predicción de series temporales [5].
- **Características (*X*):** Se han seleccionado todas las columnas que contenían los indicadores técnicos calculados (*SMA_5*, *EMA_10*, *RSI_14*, etc.), así como las columnas originales (*Open*, *High*, *Low*, *Volume*, etc.).
- **División de los datos:** El conjunto de datos se ha dividido en dos subconjuntos. El de entrenamiento (80%), para ajustar los parámetros del modelo, y el de prueba (20%), para evaluar su rendimiento.

Esta división 80%-20% es una práctica estándar en el aprendizaje automático, ya que permite utilizar una proporción adecuada de datos para el entrenamiento mientras se reserva una cantidad suficiente para la evaluación [13]. Además, al trabajar con series temporales, respeta el orden temporal para evitar fugas de información y garantizar la validez del modelo [14].

- **Entrenamiento del modelo:** Se ha utilizado la implementación de regresión lineal de la biblioteca *scikit-learn* [15], una herramienta ampliamente aceptada y utilizada en la comunidad de aprendizaje automático debido a su facilidad de uso y eficiencia. El modelo ha sido entrenado con los datos del conjunto de entrenamiento y luego evaluado con el conjunto de prueba, utilizando el método de mínimos cuadrados, que minimiza la suma de los errores cuadrados entre las predicciones y los valores reales [16].

3.5. Preparación de los datos para redes neuronales

Para implementar los modelos LSTM y GRU, es necesario estructurar los datos históricos en forma de secuencias, adecuadas para procesarse en redes neuronales recurrentes.

- **Creación de secuencias:** Se han generado ventanas deslizantes (*sliding windows*) con un tamaño de 10 días consecutivos (*window_size*). Cada ventana se utilizó como entrada para predecir el precio de cierre del día siguiente. Este enfoque es estándar en el modelado de series temporales y permite capturar patrones locales en los datos [3][4].
- **División en entrenamiento y prueba:** El conjunto de datos se ha dividido en un 80% para entrenamiento y un 20% para prueba, respetando el orden temporal. Esto asegura que los datos de prueba no estén presentes en el entrenamiento, evitando fugas de información y simulando condiciones reales de predicción [5].
- **Formato de los datos:** Los datos de entrada (X_{train} y X_{test}) tienen dimensiones de $(n_samples, window_size, n_features)$, donde:
 - $n_samples$ es el número de secuencias generadas.
 - $window_size$ es el tamaño de la ventana temporal (10 días).
 - $n_features$ es el número de características (13, que incluye precios y los indicadores técnicos generados.)

Las etiquetas (y_{train} , y_{test}) son vectores 1D que contienen los valores de cierre a predecir para cada secuencia.

3.6. Optimización de hiperparámetros

Para los modelos LSTM y GRU, se seleccionaron los siguientes parámetros a optimizar:

- **Tamaño de la ventana** (*window_size*): Determina cuántos días consecutivos de datos se utilizan como entrada para predecir el precio del día siguiente. Un tamaño demasiado pequeño puede perder contexto temporal, mientras que uno demasiado grande puede introducir ruido. Valores evaluados: 10, 20, 30.
- **Número de unidades recurrentes** (*units*): Define la capacidad de memoria del modelo para capturar dependencias temporales, el número de neuronas en las capas ocultas de las redes LSTM y GRU. Un valor alto de neuronas permite capturar patrones más complejos, aunque incrementa el riesgo de sobreajuste. Valores evaluados: 32, 64, 128.
- **Tamaño del lote** (*batch_size*): Influye en la estabilidad del entrenamiento y la eficiencia computacional. Valores evaluados: 32, 64.
- **Tasa de aprendizaje** (*learning_rate*): Controla la velocidad de ajuste de los pesos del modelo. Un valor pequeño asegura una convergencia estable, mientras que uno grande puede llevar a oscilaciones en el proceso de optimización. Valores evaluados: 0.001, 0.0005.

El proceso de optimización siguió los siguientes pasos:

1. **División de los datos:** Los conjuntos de entrenamiento y prueba fueron utilizados como en iteraciones anteriores, garantizando que los datos de prueba permanecieran sin tocar durante la optimización.
2. **Definición de la métrica de evaluación:** Se utilizó el MSE (Error Cuadrático Medio) como criterio principal para comparar el rendimiento de las combinaciones de hiperparámetros. Esta métrica es adecuada para problemas de regresión en series temporales, ya que penaliza los errores grandes [23].
3. **Evaluación iterativa:** Para cada combinación de hiperparámetros, se entrenaron y evaluaron los modelos LSTM y GRU en el conjunto de validación.

Para la optimización de los hiperparámetros, se utilizó *Grid Search*, un método exhaustivo que evalúa todas las posibles combinaciones de valores especificados para

cada parámetro. Este enfoque permite identificar la configuración que produce los mejores resultados según la métrica definida, en este caso, el MSE (Error Cuadrático Medio). Si bien *Grid Search* puede ser computacionalmente costoso, resulta adecuado para este trabajo debido al número moderado de hiperparámetros evaluados. Además, su simplicidad y efectividad lo convierten en una elección común en investigaciones relacionadas con modelos de aprendizaje profundo [24].

4. Desarrollo y Resultados

Esta sección se centra en el análisis y comparación de los modelos implementados, incluyendo sus resultados, interpretaciones y conclusiones.

4.1. Resultados y análisis de la línea base

Con la implementación de la regresión lineal descrita en la sección 3.4, aquí se presentan los resultados obtenidos y su interpretación.

4.1.1. Métricas de rendimiento

- **Error Cuadrático Medio (MSE):** 0.00205. El MSE mide la media de los errores al cuadrado, penalizando más los errores grandes, lo que lo convierte en una métrica robusta para evaluar la precisión del modelo [17].
- **Error Absoluto Medio (MAE):** 0.0337. El MAE mide el error promedio en términos absolutos, proporcionando una interpretación directa del error medio en el rango normalizado [18].

Estas métricas indican que el modelo presenta un error promedio absoluto del 3.37% en el rango normalizado de $[0,1]$, lo que refleja un rendimiento aceptable para un modelo base.

4.1.2. Visualización de predicciones

A continuación, se presenta un gráfico comparativo entre los valores reales y los valores predichos para el conjunto de prueba:

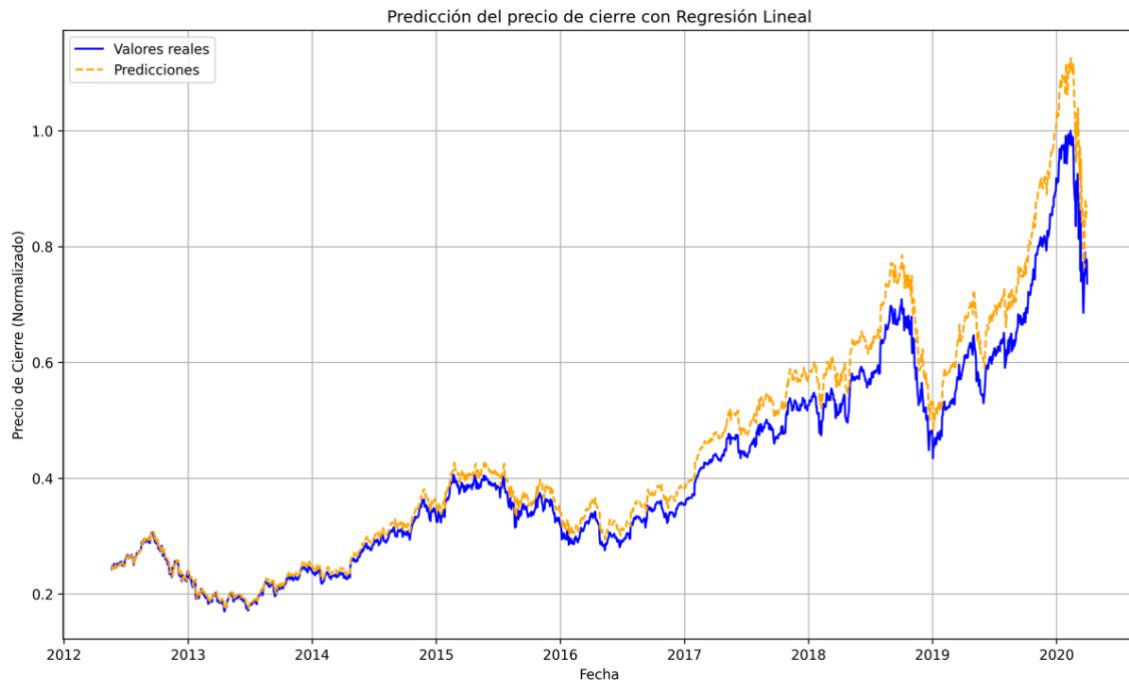


Ilustración 2: Predicción del precio de cierre con Regresión Lineal

En el gráfico se observa que el modelo sigue las tendencias generales de los precios, aunque tiene limitaciones en la precisión, especialmente durante los periodos de alta volatilidad.

4.1.3. Interpretación y conclusión

El modelo de regresión lineal ha demostrado ser capaz de capturar algunas tendencias generales en los precios de cierre ajustados. Sin embargo, debido a la simplicidad del enfoque, no logra modelar adecuadamente la complejidad de los patrones temporales en los datos financieros. Este resultado refuerza la necesidad de utilizar modelos más sofisticados, como LSTM y GRU, para capturar las relaciones temporales y no lineales inherentes a los precios de acciones.

4.2. Resultados y análisis de los modelos LSTM y GRU

Esta sección se dedica a la implementación y evaluación de los modelos LSTM y GRU. A continuación, se detallan los pasos de preparación, entrenamiento y resultados.

4.2.1. Preparación de los datos

Para la implementación de los modelos LSTM y GRU, los datos estructurados en secuencias de 10 días y 13 características fueron divididos en un conjunto de entrenamiento (80%) y uno de prueba, como se explicó en la sección 3.5. Las dimensiones de los datos para el entrenamiento y prueba son las siguientes:

- **Tamaño de X_train:** (7904, 10, 13)
- **Tamaño de y_train:** (7904,)
- **Tamaño de X_test:** (1976, 10, 13)
- **Tamaño de y_test:** (1976,)

Estas dimensiones corresponden a 7904 secuencias de entrenamiento y 1976 secuencias de prueba, con 13 características por secuencia, que incluyen los precios y los indicadores técnicos generados.

4.2.2. Configuración de los modelos LSTM y GRU

Los modelos LSTM y GRU han sido configurados con las siguientes características:

- **Número de unidades recurrentes:** 50 unidades en las capas LSTM y GRU.
- **Capa densa de salida:** Una capa densa con una sola neurona de salida, utilizando activación lineal para predecir el precio de cierre.
- **Optimizador:** Adam, que es comúnmente utilizado para redes neuronales debido a su eficiencia y capacidad para manejar gradientes en redes profundas.

Ambos modelos han sido entrenados los utilizando datos de entrenamiento, y su rendimiento fue evaluado con el conjunto de prueba utilizando las métricas Error Cuadrático Medio (MSE) y Error Absoluto Medio (MAE), como con la regresión lineal.

4.2.3. Resultados

Se llevaron a cabo múltiples pruebas para los modelos LSTM y GRU, variando el número de épocas (5, 7 y 10). Esto permitió analizar la evolución del modelo frente al fenómeno de sobreajuste y su capacidad para generalizar correctamente.

Resultados del modelo LSTM

El modelo LSTM fue configurado y entrenado utilizando un tamaño de lote de 64. El entrenamiento se realizó durante 5, 7 y 10 épocas, alcanzando los siguientes resultados en el conjunto de prueba:

Épocas	MSE (Conjunto de Prueba)	MAE (Conjunto de prueba)
5	0.0042	0.0386
7	0.0017	0.0233
10	0.0012	0.0190

El modelo mostró un buen equilibrio entre precisión y generalización en las configuraciones de 7 y 10 épocas. En la configuración de 10 épocas, el LSTM alcanzó su mejor desempeño, con un MAE de 0.0190, lo que indica que la mayor duración del entrenamiento no resultó en sobreajuste significativo en este caso.

Resultados del modelo GRU

El modelo GRU, configurado de manera similar al LSTM, también fue entrenado durante 5, 7 y 10 épocas. Los resultados obtenidos en el conjunto de prueba son:

Épocas	MSE (Conjunto de Prueba)	MAE (Conjunto de prueba)
5	0.00033	0.0253
7	0.00053	0.0187
10	0.00119	0.0195

El modelo GRU presentó una pérdida consistentemente más baja que el LSTM durante el entrenamiento, destacándose especialmente con 7 épocas, donde alcanzó un

MAE de 0.0187. Sin embargo, en la configuración de 10 épocas, el MAE incrementó ligeramente a 0.0195, sugiriendo una leve tendencia al sobreajuste.

Análisis comparativo

Al comparar ambos modelos, se observa que el GRU supera al LSTM tanto en términos de MSE como de MAE, mostrando una capacidad superior para capturar las relaciones temporales en los datos:

Modelo	Épocas	MSE (Conjunto de prueba)	MAE (Conjunto de prueba)
LSTM	5	0.0042	0.0386
GRU	5	0.00033	0.0253
LSTM	7	0.0017	0.0233
GRU	7	0.00053	0.0187
LSTM	10	0.0012	0.0190
GRU	10	0.00119	0.0195

Además, el modelo GRU demostró ser más robusto y menos propenso al sobreajuste en comparación con LSTM. Este resultado refuerza la idea de que las GRU pueden ser más efectivas que las LSTM en ciertos contextos, especialmente cuando se trata de datos de series temporales con características limitadas. Esto es consistente con estudios previos [1], donde se ha destacado la eficiencia computacional y capacidad de generalización de GRU frente a LSTM.

4.2.4. Curvas de pérdida durante el entrenamiento

En esta sección se analizan las curvas de pérdida (MSE) en los conjuntos de entrenamiento y validación durante el entrenamiento de los modelos LSTM y GRU, utilizando diferentes números de épocas (5, 7 y 10). Este análisis permite identificar tendencias de sobreajuste o subajuste en cada configuración y evaluar cómo los modelos responden a diferentes duraciones de entrenamiento.

Gráfico del modelo LSTM

El modelo LSTM mostró un comportamiento estable en las primeras épocas en todas las configuraciones. Sin embargo, se observó que:

- Con 5 épocas, el modelo no alcanzó su máximo potencial, con un MSE final de 0.0042 y un MAE de 0.0386 en el conjunto de prueba. Esto indica que el modelo podría beneficiarse de entrenamientos más prolongados para capturar mejor las relaciones temporales:

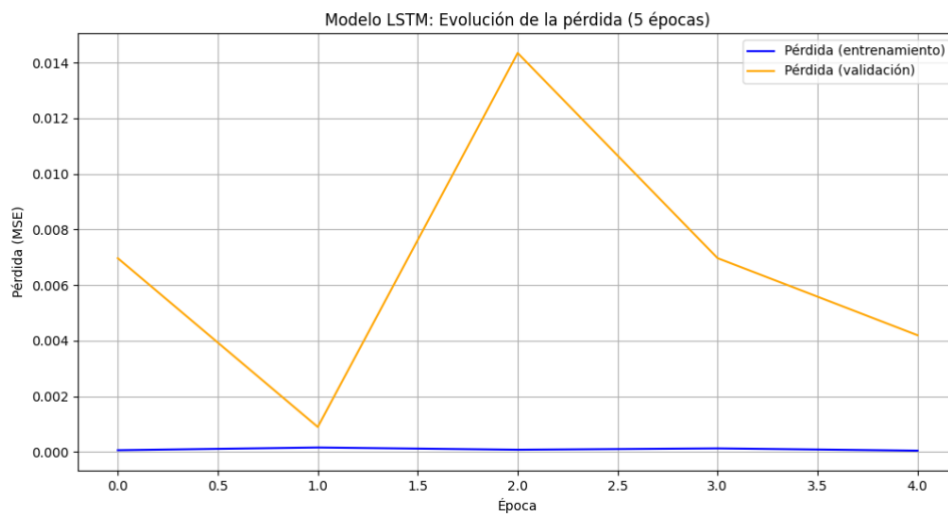


Ilustración 3: Modelo LSTM: Evolución de la pérdida (5 épocas)

- Con 7 épocas, se logró el mejor equilibrio entre generalización y precisión, con un MSE de 0.0017 y un MAE de 0.0233. Se muestra una clara mejora respecto a la configuración anterior, sin señales significativas de sobreajuste:



Ilustración 4: Modelo LSTM: Evolución de la pérdida (7 épocas)

- Con 10 épocas, el modelo alcanzó su mejor rendimiento global, con un MSE de 0.00116 y un MAE de 0.0190 en el conjunto de prueba. Las curvas de pérdida sugieren una ligera tendencia al sobreajuste en las últimas épocas, aunque el impacto en las métricas finales fue mínimo:

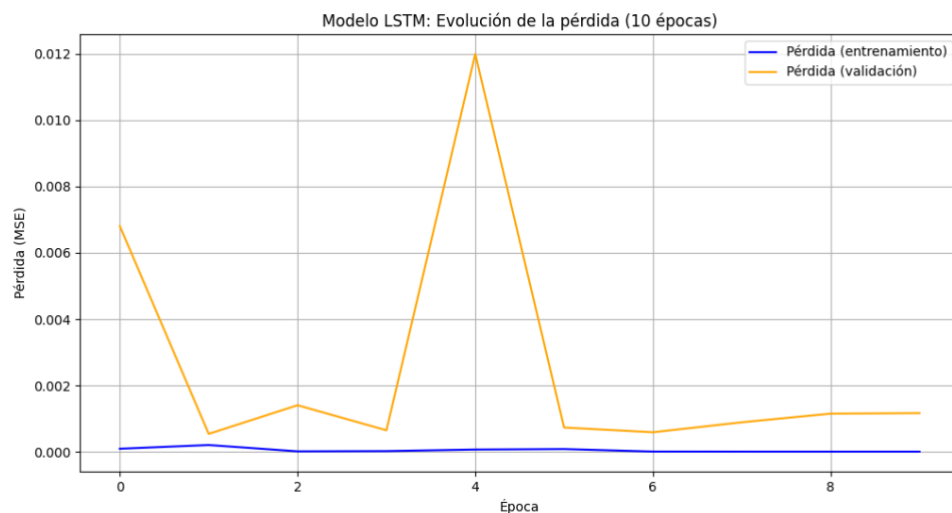


Ilustración 5: Modelo LSTM: Evolución de la pérdida (10 épocas)

Gráfico del modelo GRU

El modelo GRU presentó una convergencia más consistente, con curvas de pérdida más suaves y una menor diferencia entre el entrenamiento y la validación. Este

comportamiento sugiere que el GRU manejó mejor las dependencias temporales de los datos sin caer en sobreajuste significativo.

- Con 5 épocas, presentó un rendimiento aceptable con un MSE de 0.00033 y un MAE de 0.0253. Esto lo posiciona como una opción eficiente en términos computacionales para escenarios con limitaciones de tiempo o recursos:

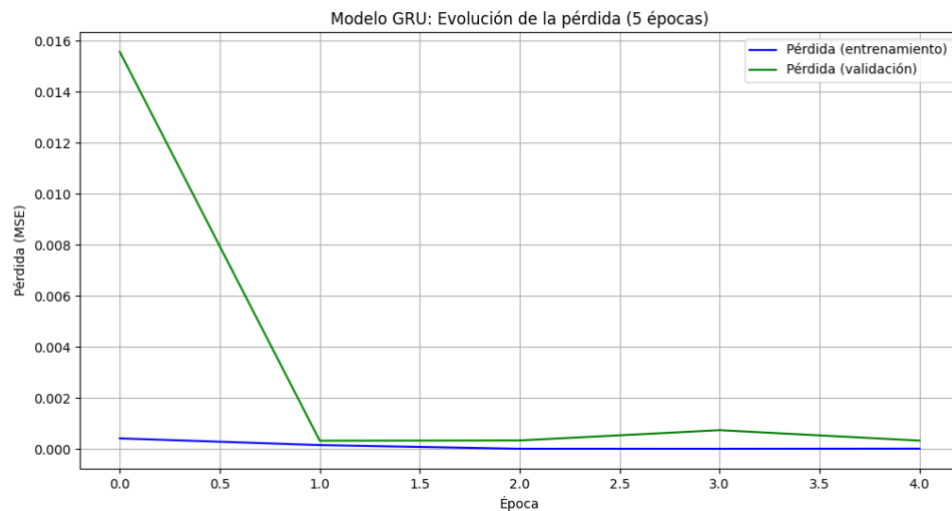


Ilustración 6: Modelo GRU: Evolución de la pérdida (5 épocas)

- Con 7 épocas, alcanzó el mejor rendimiento global, con un MSE de 0.00053 y un MAE de 0.0187. En esta configuración, GRU superó significativamente a LSTM, destacándose como el modelo más eficiente en este trabajo:



Ilustración 7: Modelo GRU: Evolución de la pérdida (7 épocas)

- Con 10 épocas, aunque las métricas finales fueron competitivas (MSE de 0.00119 y MAE de 0.0196), se observó un leve incremento en la pérdida de validación hacia las últimas épocas. Esto sugiere una tendencia al sobreajuste en configuraciones más largas.

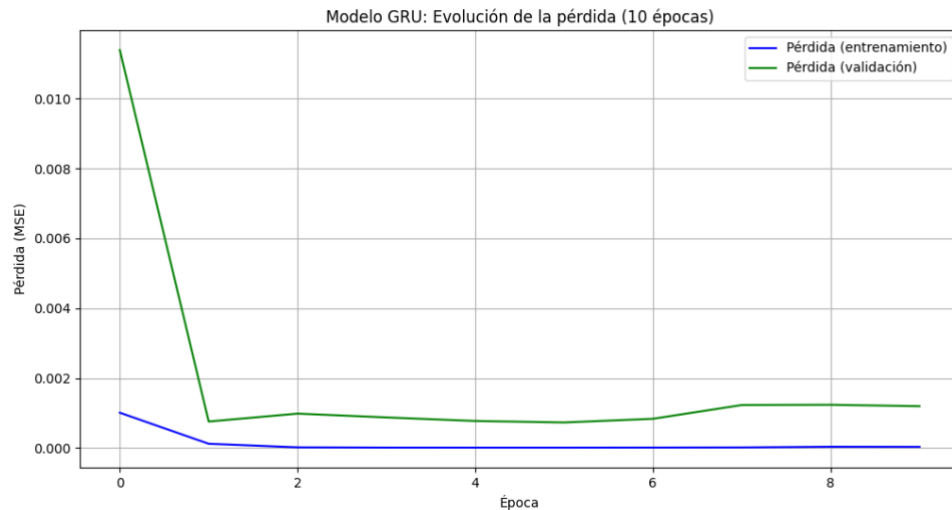


Ilustración 8: Modelo GRU: Evolución de la pérdida (10 épocas)

4.2.5. Visualización de predicciones

Se han generado gráficos comparativos que muestran las predicciones de los modelos LSTM y GRU frente a los valores reales del conjunto de prueba. Estos gráficos permiten evaluar visualmente el rendimiento de los modelos en la tarea de predicción del precio de cierre.

Gráfico del modelo LSTM

El modelo LSTM sigue las tendencias generales de los valores reales, aunque presenta desviaciones en periodos de alta volatilidad. Esto indica que es capaz de capturar patrones temporales, pero con limitaciones al modelar las dinámicas más complejas.

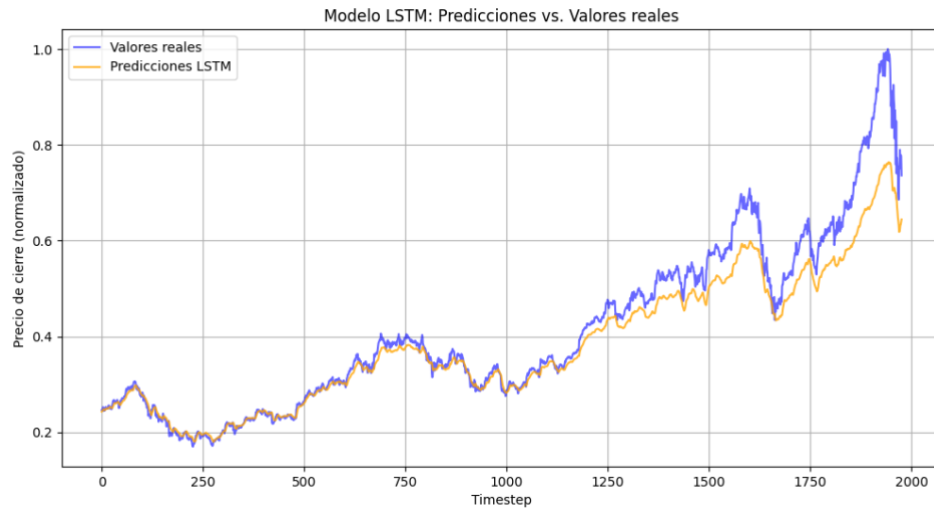


Ilustración 9: Modelo LSTM: Predicciones vs. Valores reales

Gráfico del modelo GRU

El modelo GRU muestra un rendimiento más consistente, con predicciones que se alinean mejor con los valores reales en periodos de alta volatilidad. Este resultado refuerza su superioridad en términos de error cuadrático medio y error absoluto medio.

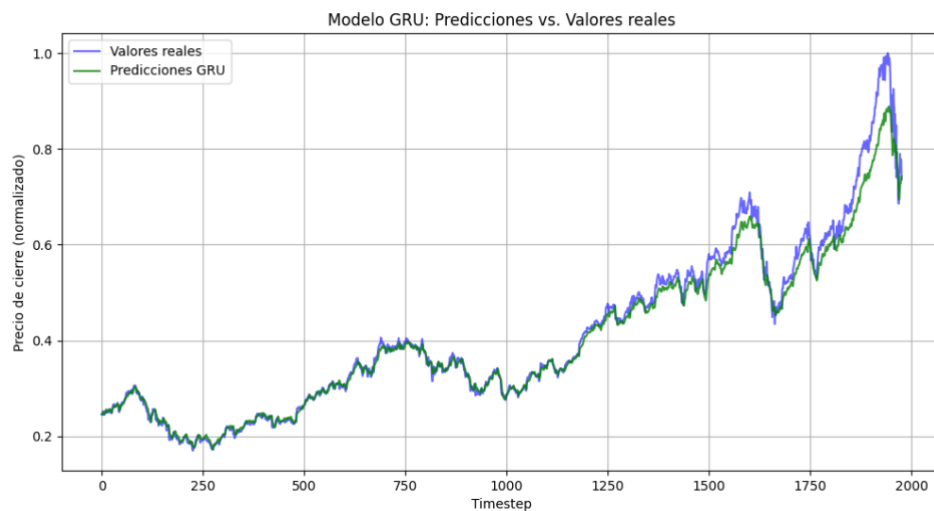


Ilustración 10: Modelo GRU: Predicciones vs. Valores reales

4.2.6. Interpretación y conclusión

El análisis realizado resalta la capacidad de los modelos GRU y LSTM para manejar problemas de predicción en series temporales, mostrando mejoras significativas frente a

la regresión lineal utilizada como línea base. Ambos modelos demostraron ser herramientas efectivas, aunque con diferencias en su rendimiento según la configuración de entrenamiento.

El modelo GRU destacó como el más eficiente y preciso en este trabajo. Su mejor desempeño se observó con 7 épocas, donde alcanzó un MSE de 0.00053 y un MAE de 0.0187 en el conjunto de prueba. Esto refuerza la idea de que las GRU son menos propensas al sobreajuste y pueden capturar eficientemente las relaciones temporales en los datos, lo que está alineado con investigaciones previas que resaltan su simplicidad y capacidad de convergencia más rápida.

Por otro lado, el modelo LSTM mostró un rendimiento competitivo, logrando su mejor configuración también con 10 épocas (MSE de 0.00116, MAE de 0.0190). A pesar de ser ligeramente menos eficiente que el GRU en este caso, el LSTM sigue siendo una opción robusta para problemas de series temporales, especialmente en casos donde se requieren modelos con mayor capacidad para manejar dependencias a largo plazo.

Finalmente, ambos modelos superaron significativamente a la regresión lineal, que alcanzó un MSE de 0.0020 y un MAE de 0.0337, validando así la hipótesis inicial de que las redes neuronales recurrentes son superiores para la predicción de precios de acciones.

Modelo	MSE (Conjunto de prueba)	MAE (Conjunto de prueba)
LSTM	0.00116	0.0190
GRU	0.00053	0.0187
Regresión Lineal	0.0020	0.0337

Estos resultados destacan la importancia de ajustar correctamente los hiperparámetros y las configuraciones de entrenamiento para optimizar el rendimiento y evitar fenómenos de sobreajuste o subajuste.

4.3. Optimización de hiperparámetros y evaluación de modelos

La optimización de hiperparámetros es una etapa crucial en el diseño de modelos de aprendizaje profundo, ya que permite ajustar parámetros clave para maximizar el

rendimiento del modelo. En este trabajo, se utilizó *Grid Search* [21], un enfoque sistemático que evalúa diferentes combinaciones de hiperparámetros para encontrar la configuración óptima. Este método ha sido ampliamente utilizado en investigaciones relacionadas con redes neuronales debido a su simplicidad y eficacia para problemas con un número limitado de combinaciones posibles [22].

Tras aplicar la metodología descrita en el apartado 3.6, se obtuvieron los siguientes resultados y análisis.

4.3.1. Resultados y análisis

La optimización de hiperparámetros mediante *Grid Search* permitió explorar diversas combinaciones de parámetros clave para los modelos LSTM y GRU. Los resultados obtenidos muestran un desempeño notablemente superior de los modelos GRU frente a los LSTM para este conjunto de datos, especialmente con configuraciones que emplean mayores tamaños de ventana y unidades recurrentes.

A continuación, se presentan las diez mejores configuraciones obtenidas, ordenadas por el MSE (Error Cuadrático Medio) en el conjunto de prueba:

Modelo	Tamaño de Ventana	Unidades	Tamaño de Lote	Tasa de Aprendizaje	MSE
GRU	30	128	32	0.0010	0.000111
GRU	20	128	32	0.0010	0.000117
GRU	30	128	64	0.0010	0.000138
GRU	30	128	64	0.0005	0.000155
GRU	10	64	32	0.0010	0.000157
GRU	10	128	64	0.0010	0.000159
GRU	20	128	32	0.0005	0.000170
GRU	20	64	64	0.0010	0.000175
GRU	10	128	32	0.0005	0.000176
LSTM	10	128	32	0.0010	0.000179

El mejor desempeño se obtuvo con el modelo GRU configurado con una ventana de 30 días, 128 unidades recurrentes, un tamaño de lote de 32 y una tasa de aprendizaje de 0.001. Este modelo alcanzó un MSE de 0.000111, destacándose como la configuración más precisa en la predicción de los precios de cierre.

Aunque el modelo LSTM mostró un rendimiento inferior en general, su mejor configuración (ventana de 10 días, 128 unidades, tamaño de lote de 32 y tasa de aprendizaje de 0.001) alcanzó un MSE de 0.000179. Esta diferencia sugiere que los modelos GRU son capaces de beneficiarse de ventanas de tiempo más amplias, capturando dependencias temporales más largas de manera eficiente. En contraste, el modelo LSTM parece optimizar mejor su rendimiento con ventanas más pequeñas, lo que podría estar relacionado con su arquitectura más compleja y su mayor propensión al sobreajuste en ventanas largas [25].

4.3.2. Conclusión

Los resultados reflejan que los tamaños de ventana más amplios (20-30 días) y un mayor número de unidades recurrentes (128) favorecen el rendimiento de los modelos, especialmente en el caso de GRU. Las tasas de aprendizaje más altas (0.001) lograron una convergencia más eficiente en comparación con valores más pequeños (0.0005), posiblemente debido a la cantidad moderada de datos disponibles.

En comparación con los estudios previos, los resultados obtenidos en este trabajo muestran un rendimiento competitivo en cuanto a los valores de MSE (Error Cuadrático Medio) y MAE (Error Absoluto Medio). Los modelos GRU fueron más efectivos en términos de rendimiento general, con un MSE más bajo, lo que está alineado con lo observado en investigaciones previas que indican que las GRU son más eficientes en cuanto a la capacidad de convergencia y la propensión a evitar el sobreajuste [25].

Por otro lado, aunque el modelo LSTM mostró buenos resultados, su desempeño fue ligeramente inferior al de GRU en este caso específico. Este resultado puede ser explicado por la mayor complejidad y el mayor número de parámetros del LSTM, que pueden hacer que el modelo sea más propenso al sobreajuste si no se ajustan adecuadamente los hiperparámetros [4].

En la siguiente figura, se comparan las mejores configuraciones para los modelos GRU y LSTM según el MSE obtenido:

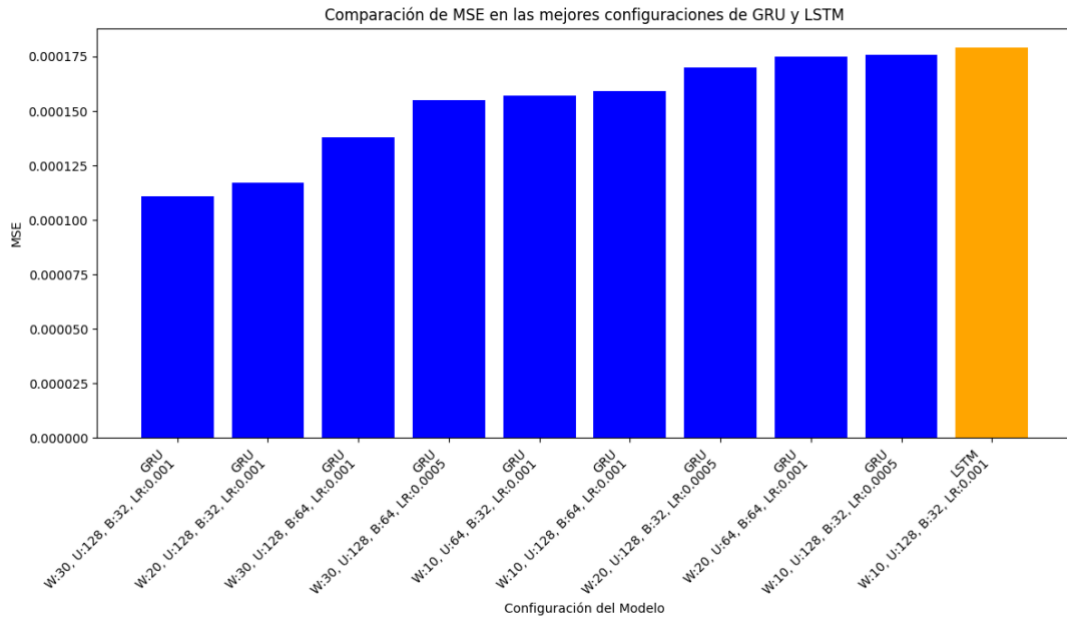


Ilustración 11: Comparación de MSE en las mejores configuraciones de GRU y LSTM

El gráfico resalta que las configuraciones óptimas del modelo GRU consistentemente superan en rendimiento a las del modelo LSTM. Esto refuerza la conclusión de que GRU es más adecuado para la predicción de series temporales en este caso.

5. Conclusiones

Este trabajo ha demostrado la efectividad de las redes neuronales recurrentes (LSTM y GRU) para abordar la predicción de precios de acciones utilizando datos históricos y características técnicas derivadas. Los modelos GRU destacaron por su precisión y menor complejidad computacional, consolidándose como la opción más eficiente en este estudio. Además, la regresión lineal utilizada como línea base permitió contextualizar las mejoras obtenidas con los modelos recurrentes, validando su superioridad en la captura de dependencias temporales y patrones no lineales inherentes a los mercados financieros.

Se cumplieron los objetivos principales planteados: preprocesamiento de datos, implementación de modelos avanzados y comparación con una línea base. Sin embargo, no se abordaron aspectos como la integración de variables macroeconómicas o análisis

de sentimiento, que podrían haber enriquecido las predicciones. Este desafío se debió a limitaciones de tiempo y alcance.

La planificación fue seguida rigurosamente, logrando cumplir con los hitos establecidos en el cronograma. No obstante, se realizaron ajustes metodológicos significativos, como la introducción de *Grid Search* para optimización de hiperparámetros, lo que mejoró el rendimiento de los modelos y su capacidad de generalización.

En investigaciones previas, se ha demostrado que las redes neuronales recurrentes (RNN), en particular LSTM y GRU, son modelos efectivos para la predicción de series temporales y el comportamiento de activos financieros. Modelos similares a los utilizados en este trabajo han sido aplicados con éxito en la predicción de precios de acciones, mostrando una mejora significativa en comparación con métodos tradicionales, como la regresión lineal [3][25]. Este trabajo confirma dichas observaciones, con un rendimiento superior de las RNN frente a la línea base de regresión lineal, especialmente en términos de MSE. Sin embargo, en algunos estudios, los resultados obtenidos por las redes neuronales recurrentes pueden ser influenciados por factores como el tamaño de la ventana de entrada, la arquitectura de la red y los hiperparámetros, lo que limita su capacidad de generalización a nuevos datos no vistos.

A pesar de los buenos resultados obtenidos, existen algunas limitaciones en este estudio. En primer lugar, los modelos entrenados mostraron un rendimiento aceptable en condiciones de baja volatilidad, siguiendo tendencias generales del mercado de manera efectiva. Sin embargo, durante periodos de alta volatilidad, como los observados en eventos como el desplome del mercado en marzo de 2020 debido a la pandemia de COVID-19, la capacidad predictiva de los modelos disminuyó significativamente. Este fenómeno ha sido ampliamente documentado en la literatura, destacando que las redes neuronales recurrentes tienden a tener un peor desempeño en escenarios con alta fluctuación de precios debido a la dificultad para capturar patrones erráticos en los datos [24]. En estos casos, las fluctuaciones rápidas en los precios tienden a generar errores más altos. Este comportamiento sugiere que la falta de sensibilidad de los modelos a eventos externos limita su aplicabilidad en escenarios de mayor inestabilidad.

Por otro lado, se propone que futuras investigaciones incluyan variables adicionales que puedan enriquecer el conjunto de datos. Por ejemplo, el análisis de sentimiento aplicado a redes sociales o noticias financieras podría proporcionar una señal temprana sobre cambios en el comportamiento del mercado, especialmente en periodos de incertidumbre. Esto impactaría positivamente en la capacidad de los modelos para identificar patrones previos a eventos volátiles, mejorando tanto su precisión como su capacidad para reaccionar ante fluctuaciones repentinas. Además, incluir indicadores macroeconómicos como tasas de interés o inflación podría complementar las características actuales y fortalecer las predicciones en mercados más amplios o diversificados.

Otra limitación es el uso de un conjunto de datos limitado. Mientras que estudios previos han utilizado grandes volúmenes de datos de múltiples fuentes y de diferentes activos, la generalización de este modelo a otros mercados o tipos de activos puede no ser inmediata.

Además, aunque las redes neuronales utilizadas son robustas, otros enfoques híbridos que combinan RNN con métodos como *SVM* o *XGBoost* han demostrado ser efectivos para capturar patrones no lineales y características complementarias en datos financieros [26].

Para mejorar los resultados de predicción, se sugiere la inclusión de otras fuentes de datos, como análisis de sentimiento o datos macroeconómicos, que podrían proporcionar señales adicionales sobre el comportamiento de los activos. Además, futuras investigaciones podrían beneficiarse del uso de modelos híbridos que combinen redes neuronales con otros enfoques de aprendizaje automático, como las máquinas de soporte vectorial (*SVM*), para capturar patrones no lineales en los datos.

6. Glosario

- **RNN (Red Neuronal Recurrente):** Tipo de red neuronal que se utiliza para procesar datos secuenciales, como series temporales. A diferencia de las redes tradicionales, las RNNs tienen conexiones que permiten que la salida de una capa se retroalimente como entrada para la siguiente capa.
- **LSTM (Long Short-Term Memory):** Variante de las RNN que se caracteriza por su capacidad para aprender dependencias a largo plazo en las series temporales, resolviendo el problema del desvanecimiento del gradiente en las RNN tradicionales.
- **GRU (Gated Recurrent Unit):** Variante de las RNN similar a LSTM, pero más simple en términos de arquitectura. Utiliza puertas (*gates*) para controlar el flujo de información y es más eficiente en términos computacionales.
- **SMA (Simple Moving Average):** Promedio de los precios de cierre durante una ventana de tiempo específica, utilizado para identificar tendencias de mercado.
- **EMA (Exponential Moving Average):** Promedio ponderado de los precios de cierre, que da mayor peso a los precios más recientes. Utilizado para identificar tendencias en mercados financieros.
- **RSI (Relative Strength Index):** Indicador que mide la velocidad y el cambio de los movimientos de precios, utilizado para identificar zonas de sobrecompra o sobreventa.
- **ATR (Average True Range):** Indicador que mide la volatilidad del mercado considerando la diferencia entre el máximo y el mínimo de cada día, así como el precio de cierre anterior.

- **MSE (*Mean Squared Error*):** Métrica de evaluación que calcula el promedio de los errores al cuadrado entre las predicciones y los valores reales. Es especialmente sensible a errores grandes.
- **MAE (*Mean Absolute Error*):** Métrica de evaluación que calcula el promedio de los errores absolutos, proporcionando una interpretación más directa del error medio.
- ***scikit-learn*:** Biblioteca de Python ampliamente utilizada para aprendizaje automático. Ofrece implementaciones eficientes de algoritmos de regresión, clasificación, y *clustering*, entre otros [15].
- ***Grid Search*:** Técnica sistemática utilizada en la optimización de hiperparámetros. Evalúa todas las combinaciones posibles de un conjunto predefinido de valores de hiperparámetros, seleccionando aquella que maximiza el rendimiento del modelo en una métrica específica.
- **SVM (*Support Vector Machines*):** Método de aprendizaje supervisado ampliamente utilizado para problemas de clasificación y regresión. Emplea funciones *kernel* para capturar relaciones no lineales en datos complejos y es comúnmente aplicado en series temporales financieras.
- ***XGBoost*:** Algoritmo de *boosting* de gradiente diseñado para alta eficiencia y flexibilidad. Este algoritmo entrena modelos en secuencia, donde cada modelo subsiguiente intenta corregir los errores cometidos por los anteriores. Es conocido por su capacidad para manejar grandes conjuntos de datos y sus técnicas avanzadas, como el manejo automático de datos faltantes y el soporte para paralelización.
- **Validación Cruzada (*Cross-Validation*):** Técnica para evaluar el rendimiento de un modelo dividiendo los datos en subconjuntos de entrenamiento y prueba de forma iterativa. Ayuda a evitar el sobreajuste y proporciona estimaciones más precisas del rendimiento del modelo en datos no vistos.

- ***Overfitting* (Sobreajuste):** Situación en la que un modelo aprende patrones específicos de los datos de entrenamiento, incluyendo ruido, en lugar de generalizar a datos nuevos, lo que resulta en un rendimiento pobre en datos de prueba.
- ***Underfitting* (Subajuste):** Ocurre cuando un modelo es demasiado simple para capturar las relaciones subyacentes en los datos, lo que lleva a un bajo rendimiento tanto en el conjunto de entrenamiento como en el de prueba.
- **Regularización:** Técnicas utilizadas para evitar el sobreajuste en modelos de aprendizaje automático, como L1 (Lasso), L2 (Ridge), o Dropout en redes neuronales.
- ***Learning Rate* (Tasa de Aprendizaje):** Parámetro que controla la magnitud de los ajustes que se hacen a los pesos del modelo durante el entrenamiento.
- ***Batch Size* (Tamaño de Lote):** Número de muestras procesadas juntas antes de actualizar los pesos del modelo durante el entrenamiento.

7. Bibliografía

[1] Ya Gao, Tong Wang, Enmin Zhou. (2021). *Stock Prediction Based on Optimized LSTM and GRU Models*.

[2] Bontempi, G., Ben Taieb, S., & Le Borgne, Y. A. (2013). *Machine learning strategies for time series forecasting*.

[3] Fischer, T., & Krauss, C. (2018). *Deep learning with long short-term memory networks for financial market predictions*.

[4] Hochreiter, S., & Schmidhuber, J. (1997). *Long short-term memory*.

[5] Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*.

[6] Oleh Onyshchak, Artem Burenok, Chanchal Alam. (2020). *Stock Market Dataset. Historical daily prices of Nasdaq-traded stocks and ETFs*.

[7] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer.

[8] Murphy, J. J. (1999). *Technical Analysis of the Financial Markets*.

[9] Wilder, J. W. (1978). *New Concepts in Technical Trading Systems*.

[10] Hull, J. (2018). *Options, Futures, and Other Derivatives*.

[11] Oleh Onyshchak. (2020). *Stock Market Dataset*.
<https://www.kaggle.com/datasets/jacksoncrow/stock-market-dataset/data>

[12] Seber, G. A. F., & Lee, A. J. (2012). *Linear Regression Analysis*.

[13] Kohavi, R. (1995). *A study of cross-validation and bootstrap for accuracy estimation and model selection*.

- [14] Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2015). *Time Series Analysis: Forecasting and Control*.
- [15] Pedregosa, F., et al. (2011). Scikit-learn: *Machine Learning in Python*.
- [16] Draper, N. R., & Smith, H. (1998). *Applied Regression Analysis*.
- [17] Chai, T., & Draxler, R. R. (2014). *Root mean square error (RMSE) or mean absolute error (MAE)?*.
- [18] Willmott, C. J., & Matsuura, K. (2005). *Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance*.
- [19] Cho, K., et al. (2014). *Learning phrase representations using RNN encoder-decoder for statistical machine translation*.
- [20] Graves, A., & Schmidhuber, J. (2005). *Framewise phoneme classification with bidirectional LSTM and other neural network architectures*.
- [21] Bergstra, J., & Bengio, Y. (2012). *Random Search for Hyper-Parameter Optimization*.
- [22] Snoek, J., Larochelle, H., & Adams, R. P. (2012). *Practical Bayesian Optimization of Machine Learning Algorithms*.
- [23] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*.
- [24] Chong, E., Han, C., & Park, F. C. (2017). *Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies*.
- [25] Chung, J., Gulcehre, C., Cho, K., Bengio, Y., & Vinyals, O. (2014). *Empirical evaluation of gated recurrent neural networks on sequence modeling*.
- [26] Shi, Z., Hu, Y., Mo, G., Wu, J. (2022). *Attention-based CNN-LSTM and XGBoost hybrid model for stock prediction*.

8. Anexos

8.1. Resultados detallados de las pruebas de entrenamiento para LSTM y GRU

8.1.1. Modelo LSTM

Entrenando el modelo LSTM para 5 épocas...

Epoch 1/5: loss: 1.2218e-04 - val_loss: 0.0070

Epoch 2/5: loss: 3.4557e-04 - val_loss: 8.9713e-04

Epoch 3/5: loss: 9.5401e-05 - val_loss: 0.0143

Epoch 4/5: loss: 4.0205e-04 - val_loss: 0.0070

Epoch 5/5: loss: 1.3795e-04 - val_loss: 0.0042

Error cuadrático medio (MSE) del modelo LSTM: 0.0041967579163610935

Error absoluto medio (MAE) del modelo LSTM para 5 épocas: 0.0385587178170681

Entrenando el modelo LSTM para 7 épocas...

Epoch 1/7: loss: 5.5028e-04 - val_loss: 0.0048

Epoch 2/7: loss: 1.3404e-04 - val_loss: 0.0037

Epoch 3/7: loss: 7.9271e-05 - val_loss: 0.0313

Epoch 4/7: loss: 4.9960e-04 - val_loss: 0.0064

Epoch 5/7: loss: 4.6708e-05 - val_loss: 0.0035

Epoch 6/7: loss: 8.2703e-06 - val_loss: 0.0018

Epoch 7/7: loss: 2.9859e-05 - val_loss: 0.0017

Error cuadrático medio (MSE) del modelo LSTM: 0.0017222205642610788

Error absoluto medio (MAE) del modelo LSTM para 7 épocas:
0.023261910304427147

Entrenando el modelo LSTM para 10 épocas...

Epoch 1/10: loss: 1.2874e-04 - val_loss: 0.0068

Epoch 2/10: loss: 6.7862e-04 - val_loss: 5.3808e-04

Epoch 3/10: loss: 1.0476e-05 - val_loss: 0.0014

Epoch 4/10: loss: 3.3498e-05 - val_loss: 6.4662e-04

Epoch 5/10: loss: 1.6945e-05 - val_loss: 0.0120

Epoch 6/10: loss: 2.5757e-04 - val_loss: 7.2981e-04

Epoch 7/10: loss: 2.9247e-06 - val_loss: 5.8672e-04

Epoch 8/10: loss: 1.6616e-06 - val_loss: 8.8819e-04

Epoch 9/10: loss: 6.0411e-07 - val_loss: 0.0011

Epoch 10/10: loss: 1.0049e-06 - val_loss: 0.0012

Error cuadrático medio (MSE) del modelo LSTM: 0.0011643166653811932

Error absoluto medio (MAE) del modelo LSTM para 10 épocas:
0.01901007816195488

8.1.2. Modelo GRU

Entrenando el modelo GRU para 5 épocas...

Epoch 1/5: loss: 9.4539e-04 - val_loss: 0.0156

Epoch 2/5: loss: 4.7929e-04 - val_loss: 3.2137e-04

Epoch 3/5: loss: 9.6332e-06 - val_loss: 3.3414e-04

Epoch 4/5: loss: 2.9597e-06 - val_loss: 7.3537e-04

Epoch 5/5: loss: 2.3037e-05 - val_loss: 3.2717e-04

Error cuadrático medio (MSE) del modelo GRU: 0.00032716908026486635

Error absoluto medio (MAE) del modelo GRU para 5 épocas:
0.025268886238336563

Entrenando el modelo GRU para 7 épocas...

Epoch 1/7: loss: 7.4794e-05 - val_loss: 0.0027

Epoch 2/7: loss: 0.0011 - val_loss: 9.1441e-04

Epoch 3/7: loss: 6.3669e-05 - val_loss: 0.0033

Epoch 4/7: loss: 2.0796e-04 - val_loss: 0.0016

Epoch 5/7: loss: 8.9887e-05 - val_loss: 0.0019

Epoch 6/7: loss: 1.1236e-04 - val_loss: 3.6153e-04

Epoch 7/7: loss: 1.7031e-05 - val_loss: 5.3315e-04

Error cuadrático medio (MSE) del modelo GRU: 0.0005331467837095261

Error absoluto medio (MAE) del modelo GRU para 7 épocas: 0.01867387443780899

Entrenando el modelo GRU para 10 épocas...

Epoch 1/10: loss: 0.0030 - val_loss: 0.0114

Epoch 2/10: loss: 3.0957e-04 - val_loss: 7.5524e-04
Epoch 3/10: loss: 2.3477e-05 - val_loss: 9.8120e-04
Epoch 4/10: loss: 6.3729e-06 - val_loss: 8.7280e-04
Epoch 5/10: loss: 8.2663e-06 - val_loss: 7.7081e-04
Epoch 6/10: loss: 6.0905e-06 - val_loss: 7.2820e-04
Epoch 7/10: loss: 1.8345e-05 - val_loss: 8.3359e-04
Epoch 8/10: loss: 7.8077e-06 - val_loss: 0.0012
Epoch 9/10: loss: 6.6633e-05 - val_loss: 0.0012
Epoch 10/10: loss: 6.6762e-05 - val_loss: 0.0012
Error cuadrático medio (MSE) del modelo GRU: 0.0011947033926844597
Error absoluto medio (MAE) del modelo GRU para 10 épocas:
0.019587621092796326