

Estudio de la Red Neuronal Profunda Propuesta por J.P. Morgan Aplicada a la Predicción y Clasificación de Series Temporales Bursátiles

Aurora Pérez Pérez^{1,†} and Laura García Perrín^{2,†}

¹Affiliation of author one
²Affiliation of author three

This manuscript was compile on May 24, 2024

Abstract

Este estudio investiga el rendimiento de una red neuronal diseñada para la predicción y clasificación de series temporales del mercado de valores, inspirada en un modelo propuesto por J.P. Morgan.

Keywords: keyword 1, keyword 2, keyword 3, keyword 4, keyword 5

Corresponding author: Provide the corresponding author information and publisher here. *E-mail address:* example@organization.com.

DOI: <https://www.doi.org/exampledoi/XXXXXXXXXX>

Received: March 20, 2024 **Revised:** April 16, 2024 **Accepted:** April 20, 2024 **Published:** May 21, 2024

Rho LaTeX Class © This document is licensed under Creative Commons CC BY 4.0.

1. Introducción

1.1. Critpmonedas, ¿qué son?

Las criptomonedas, también conocidas como monedas digitales, son instrumentos de pago que no tienen una forma física y se basan en un algoritmo matemático llamado *blockchain* o cadena de bloques [49]. A diferencia del dinero tradicional, las criptomonedas no están respaldadas por un banco central que garantice su valor, y su precio puede cambiar mucho según la oferta y la demanda. No es obligatorio aceptarlas como pago, ni se consideran una forma segura de guardar valor o una unidad de cuenta estable. Por esta razón, las criptomonedas no compiten con el dinero tradicional, sino que se consideran activos especulativos de alto riesgo.

Al tratarse de activos especulativos de alto riesgo, suscita interés anticiparse a los movimientos de los precios de dichas acciones para poder tomar decisiones de inversión más informadas y, potencialmente, aspirar a mayores rendimientos. Sin embargo, esta tarea está llena de desafíos debido a la naturaleza intrínsecamente compleja y volátil [17] [8] de los mercados financieros o de valores. El mercado de valores está influenciado por una multitud de factores [2], incluyendo noticias económicas, eventos políticos, cambios en la política monetaria, y comportamientos inesperados [1] de los inversores. Esta complejidad hace que anticiparse a los precios (o al comportamiento) de las acciones sea un problema difícil de abordar con métodos tradicionales de análisis financiero.

En este contexto, el uso de redes neuronales y técnicas avanzadas de aprendizaje profundo emergen como una solución prometedora [30] [41] [35] al ser capaces de modelar las complejas relaciones no lineales presentes en los datos del mercado [14], así como de capturar patrones que podrían ser ignorados por los métodos tradicionales. Al capturar información más precisa sobre el comportamiento de los activos financieros, se pueden tomar decisiones más informadas, lo que se traduce en una mayor ventaja competitiva (o bien más significativa) así como en una reducción de los posibles

riesgos que acarrearán nuestras decisiones. Por otra parte, las redes neuronales serían capaces de trabajar con grandes cantidades de datos y de hacer frente a otras dificultades que se expondrán más adelante.

Este estudio se centra en evaluar el rendimiento de una red neuronal diseñada específicamente para la predicción y clasificación de series temporales bursátiles, inspirada en un modelo propuesto por J.P. Morgan [52]. Entre otros objetivos, está evaluar qué métodos de procesamiento de datos son más adecuados cuando se trata de lidiar con valores nulos o faltantes, muy característicos de las series temporales, así como comparar el rendimiento de dicha arquitectura de estudio frente a otras que son populares en la literatura.

1.2. Descripción General de las Series Temporales

Una serie temporal es una colección de datos que se registran en diferentes momentos y que se ordenan cronológicamente. [51] Las series temporales se utilizan para entender cómo cambian diferentes variables a lo largo del tiempo y cómo estas variables pueden influirse mutuamente. Por ejemplo, podríamos usar series temporales para estudiar cómo la temperatura afecta las ventas de helados o cómo los precios de las acciones fluctúan con las noticias económicas.

Los datos de series temporales y su análisis están adquiriendo mucha popularidad debido a la producción masiva de dichos datos a través de, por ejemplo, el Internet de las Cosas (IoT), la digitalización de la atención médica y el auge de las ciudades inteligentes [36]. En los próximos años, podemos esperar que la cantidad, calidad e importancia de los datos de series temporales crezcan rápidamente. [48] Así pues, a medida que se recopilan más datos, se vuelve necesario utilizar técnicas de análisis más eficientes y competentes que nos ayuden a extraer información de los mismos para entender los fenómenos que ocurren en el mundo, tomar decisiones más informadas y adquirir cierta capacidad de anticipación lanzando predicciones más precisas.

El análisis de series temporales aparece en numerosos campos y aplicaciones, entre los que podríamos destacar la salud y



Figure 1. Precio de Bitcoin (BTC) por día desde el 2 de septiembre de 2021 hasta el 28 de mayo de 2024. Fuente: [Stadista](#).

medicina, la meteorología, la economía y la astronomía, entre otros. El primer «problema» que podríamos encontrar es reside en llevar a cabo un modelado adecuado de series temporales, de tal forma que se alcanzase la construcción del mejor modelo que se ajustara a los datos. Ahora bien, suponiendo que hemos superado esta primera dificultad, nos encontramos con que estas series tienen características distintivas que son cruciales para su posterior análisis y predicción:

- **Estacionariedad:** Una serie temporal es estacionaria si sus propiedades estadísticas, como la media y la varianza, no cambian con el tiempo. La no estacionariedad puede manifestarse a través de tendencias o cambios de nivel.
- **Tendencia:** Un componente a largo plazo que muestra una dirección general en la serie temporal, ya sea creciente o decreciente.
- **Estacionalidad:** Fluctuaciones periódicas que se repiten en intervalos regulares, como días, meses o años.
- **Ruido:** Variaciones aleatorias que no pueden explicarse por tendencias, estacionalidades o otros componentes sistemáticos.
- **Autocorrelación:** La relación entre valores de la serie temporal en diferentes momentos. Una alta autocorrelación indica que los valores pasados influyen significativamente en los valores futuros.

Las series temporales en finanzas bursátiles son altamente volátiles y ruidosas en comparación con las de medicina, astronomía y meteorología. Mientras que las series médicas presentan alta variabilidad biológica y las series meteorológicas y astronómicas muestran estacionalidades claras y tendencias predecibles, las series bursátiles están sujetas a cambios bruscos debido a factores económicos y eventos externos. Además, en finanzas, la no estacionariedad y la influencia de noticias económicas y políticas son más pronunciadas, haciendo que las predicciones sean más complejas y desafiantes en comparación con otros campos donde las leyes naturales y biológicas predominan.

2. Revisión de la Literatura

2.1. Introducción a los Modelos Predictivos en Series Temporales

La predicción de series temporales ha sido una área de investigación activa debido a su aplicabilidad en diversos cam-

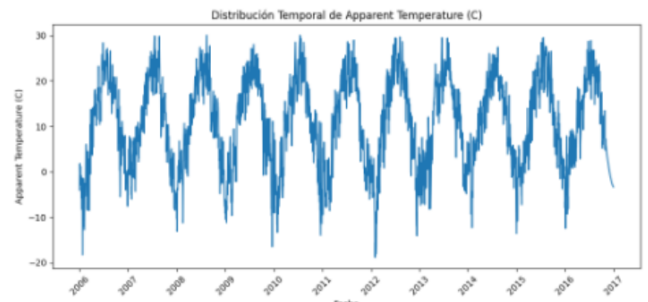


Figure 2. Ejemplo de un gráfico que muestra la evolución de la temperatura de cierta región a lo largo de los años. Podemos apreciar que existe un patrón cíclico por el cual, en el primer o segundo semestre del año, las temperaturas tienden a ser más altas que en el otro semestre (estacionalidad).

pos como la energía, meteorología, finanzas, y más. Tradicionalmente, los modelos estadísticos han dominado esta área, pero recientemente, los modelos de aprendizaje profundo han mostrado un desempeño superior, especialmente en la captura de dependencias complejas a largo plazo.

2.2. Evolución del Pronóstico en Series Temporales

La tarea de predecir el futuro a partir de datos históricos se conoce como «pronóstico». Esto, llevado al ámbito del análisis de series temporales, quiere decir que tratamos de predecir el futuro a través de datos históricos, es decir, que ya han ocurrido con anterioridad. Tradicionalmente, esta tarea se basaba en métodos estadísticos [3]–[7], [9], [31], pero con la aparición de los ordenadores en el siglo XX, el paradigma comenzó a cambiar y surgieron nuevas metodologías [7], [10]–[12], [31].

Entre los modelos tradicionales para la predicción de series temporales destaca el modelo ARIMA (*AutoRegressive Integrated Moving Average*), que ha sido la base de la predicción de series temporales durante décadas. Estos enfoques funcionan bien con series temporales que presentan patrones cíclicos o tendencias lineales [33]. Sin embargo, no son tan efectivos con series temporales altamente volátiles o no lineales, como ocurre con las series temporales bursátiles.

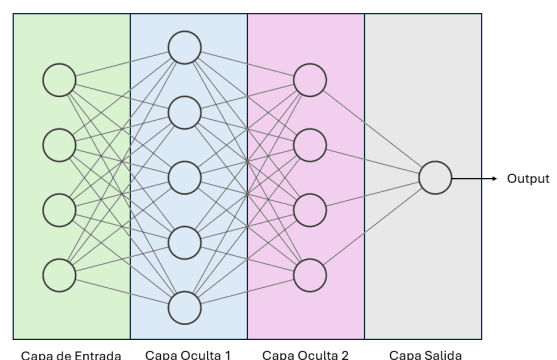


Figure 3. Diagrama FNCC que muestra un ejemplo de arquitectura de una red neuronal profunda con dos capas ocultas.

Las soluciones de *machine learning* (ML) han ganado popularidad para la predicción de series temporales bursátiles debido a su capacidad para manejar la complejidad de los datos [13], [22], [26], [29], [32]. Dentro de ML, el aprendizaje

profundo (*deep learning*) es de especial interés ya que se trata de una técnica muy flexible que aprende a partir de grandes volúmenes de datos, lo cual es beneficioso para el análisis de series temporales.

Los modelos de aprendizaje profundo, como las Redes Neuronales Recurrentes (RNN) y las Redes Neuronales Convolucionales (CNN), han revolucionado el campo de la predicción de series temporales. En particular, el modelo Multivariate Temporal Convolutional Network (M-TCN) desarrollado por Wan et al. (2019) [33] ha mostrado mejoras significativas en la precisión de predicción al manejar datos multivariados y no periódicos. Este modelo combina convoluciones dilatadas y bloques residuales asimétricos para mejorar la captura de patrones a largo plazo en series temporales multivariadas.

Un enfoque novedoso es la combinación de CNNs y Transformers para la predicción de series temporales financieras, como se describe en el artículo de Zeng et al. (2023) [47]. Este método aprovecha la capacidad de las CNNs para capturar patrones locales y la capacidad de los Transformers para aprender dependencias a largo plazo. Los experimentos demostraron que este enfoque supera a métodos tradicionales como ARIMA y EMA, así como a modelos de aprendizaje profundo autoregresivos como DeepAR [37].

Wu et al. (2023) exploraron la efectividad de estructuras 2D de series temporales en tareas de clustering, utilizando métodos como el Recurrent Plot (RP), Gramian Angular Summation Field (GASF), Gramian Angular Differential Field (GADF), y Markov Transition Field (MTF). Estos métodos permitieron clasificar índices como el CSI300 y el S&P500, reconociendo fluctuaciones anormales en las series temporales financieras, cruciales para el análisis de riesgo y estrategias de inversión [46].

2.3. Pronóstico Visual en el Dominio Financiero

Nuestro estudio se enfoca en una nueva metodología para predecir datos a lo largo del tiempo que ha ganado popularidad en los últimos años. Este método convierte los datos numéricos en imágenes para aprovechar los algoritmos utilizados en visión por computadora. La idea es que las representaciones visuales de datos financieros pueden ayudar a los comerciantes a tomar mejores decisiones. Investigaciones anteriores han explorado cómo representar los datos de series temporales como imágenes de gráficos de líneas, gráficos recurrentes y gráficos de velas para tareas de predicción y clasificación.

Dada la imprevisibilidad inherente en las series temporales financieras, varios investigadores han recurrido a técnicas de descomposición para obtener información más detallada en diferentes niveles de frecuencia, logrando así una mejor gestión de carteras y modelado de riesgos. Estos trabajos han explorado cómo predecir y clasificar dichos datos utilizando representaciones de frecuencia combinadas con técnicas de aprendizaje profundo [52].

2.4. Aplicaciones en el Sector Financiero

La aplicación de técnicas de aprendizaje profundo en el sector financiero ha sido extensivamente explorada. Por ejemplo, el

uso de codificación de series temporales a imágenes y técnicas de aprendizaje profundo, como se describe en «*Deep Learning and Time-Series-to-Image Encoding for Financial Forecasting*» [33], ha mejorado las predicciones en mercados financieros volátiles al transformar datos de series temporales en representaciones visuales para el análisis con CNNs. Otro ejemplo es el enfoque propuesto por Khalid et al. (2024), que utiliza una red neuronal convolucional profunda para convertir datos financieros en imágenes 2D y predecir tendencias de mercado. Este método demostró ser superior en precisión comparado con modelos tradicionales de aprendizaje automático [50].

Cohen et al. (2021) introdujeron un enfoque innovador donde la predicción de series temporales se aborda como una tarea de visión por computadora. Transforman los datos de entrada en imágenes y entrenan un modelo para producir la imagen subsecuente, prediciendo distribuciones en lugar de valores puntuales. Este método mostró ser efectivo para datos cíclicos, pero menos para datos irregulares como los precios de acciones. Sin embargo, superó a varios métodos basados en imágenes, incluyendo ARIMA, al evaluar las predicciones utilizando métricas basadas en imágenes [44].

El estudio de Bamford et al. (2023) propone un marco para la recuperación de series temporales financieras multimodales utilizando proyecciones de espacio latente. Este enfoque permite consultas amigables para el usuario, como texto en lenguaje natural o bosquejos de series temporales, mejorando la eficiencia computacional y la precisión en comparación con los enfoques tradicionales de recuperación de series temporales [45].

2.5. Desafíos y Oportunidades Futuras

A pesar de los avances, el pronóstico de series temporales bursátiles sigue enfrentando desafíos significativos. La volatilidad del mercado, el riesgo de *overfitting* y la necesidad de grandes cantidades de datos etiquetados son problemas persistentes.

Las investigaciones actuales se enfocan en optimizar estos modelos para hacerlos más eficientes y robustos. Por ejemplo, el uso de capas convolucionales 1D en lugar de convoluciones causales y la implementación de bloques residuales asimétricos presentan oportunidades para mejorar la precisión y reducir el tiempo de entrenamiento [33]. Además, el enfoque híbrido que combina CNNs y Transformers muestra un gran potencial para mejorar aún más la precisión y eficiencia de las predicciones de series temporales [47].

3. Arquitectura Considerada

A continuación, se describen las arquitecturas de redes neuronales utilizadas en la investigación, incluyendo tanto la arquitectura propuesta por J.P. Morgan [52] como la arquitectura adaptada para este estudio.

3.1. Arquitectura Propuesta por J.P. Morgan

El artículo de J.P. Morgan propone un nuevo enfoque para la predicción de series temporales al utilizar como entrada a la red neuronal representaciones visuales en forma de espectro-

gramas de tiempo-frecuencia y transformadores (*transformers*) de visión.

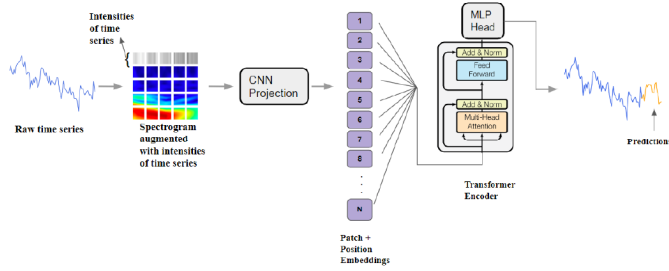


Figure 4. Descripción general del enfoque propuesto por J.P. Morgan

Entre sus componentes clave, se encuentran:

- **Espectrograma de Tiempo-Frecuencia:** Se utilizan las transformadas *wavelet* para generar espectrogramas que capturan información tanto en la dimensión temporal como frecuencial.
- **Transformer de Visión (ViT):** Un transformer de visión adaptado se usa para procesar las imágenes del espectrograma, dividiendo la imagen en parches y aplicando atención multi-cabeza para aprender las dependencias temporales y frecuenciales.
- **Cabeza MLP (Multilayer Perceptron):** Añadida al *transformer*, realiza la predicción final.

Las ventajas de considerar esta arquitectura son que se permite el aprendizaje simultáneo en los dominios de tiempo y frecuencia, lo que mejora la precisión de predicción y de clasificación ya que se tiene en cuenta más información, y, probablemente, sea capaz de capturar patrones complejos que podrían ser ignorados por métodos tradicionales.

3.2. Arquitectura Adaptada para Predicción de Series Temporales Bursátiles

Nuestra arquitectura propuesta toma inspiración del enfoque de J.P. Morgan, adaptándolo para la predicción y clasificación de series temporales bursátiles.

En primer lugar, la construcción del modelo comienza con la definición de las entradas, que son imágenes de tamaño 128×128 en escala de grises. Estas imágenes se tokenizan mediante la extracción de parches utilizando una capa Conv2D, que convierte las imágenes en *tokens* para el procesamiento del Transformer. Se aplica un bloque Transformer compuesto por atención multi-cabeza y una red *feed-forward*, acompañado de normalización de capas y *dropout* para la regularización. Después, se aplanan los resultados y se pasan por capas densas finales para realizar la clasificación. El modelo se compila utilizando el optimizador Adam y se entrena para minimizar la pérdida de entropía cruzada binaria, con la precisión como métrica de evaluación para la red neuronal en el entrenamiento y en el conjunto de pruebas.

Por otra parte, el mejor modelo general utiliza la imputación exponencial con una precisión en el test de 0.5942 (contando con un *dataset* de 720 instancias). Los resultados del estu-

dio de hiperparámetros utilizando la imputación exponencial mostraron que los mejores parámetros fueron los siguientes:

- `num_patches`: 64
- `patch_size`: 15
- `num_heads`: 16
- `ff_dim`: 128
- `embed_dim`: 192
- `batch_size`: 128

A considerar que otros métodos de imputación de nulos que se emplearon fueron k-NN, MICE, interpolación lineal y *spline*, entre otros. En su mayorá, la mejor precisión que lograron alcanzar en el conjunto de pruebas fue de 0.5072, aproximadamente.

3.3. Modelos Comparativos

Finalmente, se describen brevemente otras arquitecturas de redes neuronales que se utilizan en el estudio con el objetivo de ser comparadas y así evaluar el rendimiento de nuestra arquitectura propuesta. Es decir, estos modelos sirven como puntos de referencia (*baselines*) a la hora de medir la efectividad y apreciar posibles mejoras proporcionadas por nuestra metodología.

3.3.1. Modelos Estadísticos

- **EMA (Exponential Moving Average):** El promedio móvil exponencial es un método de suavizado que asigna más peso a las observaciones recientes. Es utilizado comúnmente en el análisis técnico de series temporales financieras para identificar tendencias. La principal ventaja es que es simple de implementar; la contrapartida es que difícilmente captura tendencias que no sean lineales [31].
- **ARIMA (AutoRegressive Integrated Moving Average):** ARIMA combina componentes autorregresivos (AR), de promedios móviles (MA) y de integración (I) para modelar series temporales. Es ampliamente utilizado para el análisis y la predicción de series temporales debido a su flexibilidad en la captura de diversas dinámicas de datos. Entre sus ventajas está que es capaz de modelar tanto la tendencia como la estacionalidad en los datos. Sin embargo, requiere de un ajuste cuidadoso de parámetros y puede ser computacionalmente intensivo [5], [6].

3.3.2. Modelos de Deep Learning

- **DeepAR:** DeepAR es un método basado en LSTM que genera predicciones probabilísticas para series temporales univariadas. Este modelo entrena en múltiples series temporales relacionadas y utiliza la arquitectura de redes neuronales recurrentes para capturar dependencias temporales. Es capaz de manejar múltiples series temporales relacionadas y proporcionar intervalos de predicción. La contrapartida es que requiere de una gran cantidad de datos de entrenamiento y es más complejo de implementar y entrenar que los modelos estadísticos tradicionales [40].
- **ViT-Lineplot:** Esta variante de nuestro modelo propuesto utiliza *transformers* de visión para procesar visualizaciones de gráficos de líneas de series temporales en lugar de espectrogramas. Los gráficos de líneas se generan a partir de los datos de series temporales y se utilizan

como entrada al transformer de visión. La ventaja es que permite evaluar la efectividad del *transformer* de visión utilizando una representación visual más simple. La limitación es que los gráficos de líneas no pueden capturar toda la complejidad de las series temporales, especialmente las relaciones de frecuencia [38].

- **Multivariate Temporal Convolutional Network (M-TCN):** Este modelo utiliza convoluciones dilatadas y bloques residuales asimétricos para manejar la predicción de series temporales multivariadas no periódicas. Ha demostrado mejoras significativas en la precisión de predicción al capturar patrones a largo plazo en series temporales multivariadas [33].
- **Transformer:** Los Transformers han demostrado ser altamente efectivos para tareas de series temporales debido a su capacidad para capturar dependencias a largo plazo y complejas. Se utilizan en combinación con CNNs para mejorar la precisión y eficiencia de las predicciones de series temporales [28], [47].

Modelo	Ventajas	Limitaciones
EMA	Simple de implementar.	Solo captura tendencias lineales [31].
ARIMA	Modela la tendencia y estacionalidad.	Requiere ajuste de hiperparámetros y es computacionalmente intensivo [5], [6].
DeepAR	Proporciona intervalos de predicción y maneja múltiples series temporales.	Necesita gran cantidad de datos de entrenamiento y es complejo de implementar [40].
ViT-Lineplot	Representación visual más simple.	Entradas pueden ser reduccionistas y no capturar toda la complejidad de las series temporales [38].
M-TCN	Captura patrones a largo plazo en series multivariadas.	Requiere una implementación compleja [33].
Transformer	Eficaz en capturar dependencias complejas a largo plazo.	Computacionalmente intensivo y requiere gran cantidad de datos [28], [47].

Table 1. Modelos comparativos utilizados en el estudio.

4. Marco Experimental

4.1. Lenguaje de Programación y Librerías Usadas

Para este estudio, se utiliza Python como lenguaje de programación junto con varias librerías especializadas. Las principales incluyen Numpy y Pandas para la manipulación y análisis de datos [18], [39], Matplotlib y Seaborn para la visualización de datos [16], [43], y Scikit-learn para tareas de preprocesamiento y modelado, como la imputación de datos faltantes y el escalado de características [20]. También se emplean las librerías Statsmodels y Scipy para el análisis estadístico y el suavizado de datos [19], [42]. Para la construcción y entrenamiento de modelos de aprendizaje profundo, se usa TensorFlow y Keras, junto con optimizadores y diversas capas para diseñar modelos complejos [23], [24]. Además, se incorpora Pyts para la transformación de series temporales [27] y keras_tuner para la optimización de hiperparámetros.

Finalmente, se utiliza Pywt para el análisis de ondas y transformaciones de señales [34].

4.2. Conjunto de Datos y de Características

El conjunto de datos utilizado en este estudio corresponde a la serie temporal del precio de cierre del activo LUNA. Incluye registros diarios de precios de apertura, cierre, máximo, mínimo y volumen de transacciones. Los datos provienen de fuentes financieras confiables y cubren un período desde el 12 de septiembre de 2019 hasta octubre del año 2021. Otras características técnicas a tener en cuenta del conjunto de datos son las siguientes:

- Cuenta con 720 instancias y 61 columnas o atributos.
- Contiene 3 tipos de variables: `int64`, `M8[ns]` y `float64`.
- Inicialmente, se parte con 10,979 valores nulos o faltantes. Tal y como se muestra en la figura [5], el número de valores NaN es más alto en las columnas relacionadas con YouTube y Medium, y más bajo en las columnas relacionadas con el ranking de mercado y el puntaje alternativo.

La matriz de correlación muestra cómo se relacionan las diferentes variables entre sí. Los valores oscilan entre -1 y 1. Un valor cercano a 1 indica una fuerte correlación positiva (cuando una variable sube, la otra también), mientras que un valor cercano a -1 indica una fuerte correlación negativa (cuando una variable sube, la otra baja). En la figura 6 observamos que hay varias áreas rojas y azules, lo que sugiere que algunas variables están altamente correlacionadas positiva o negativamente entre sí. Las variables que están fuertemente correlacionadas pueden ser útiles para predecir el comportamiento de una variable en función de otra.

Por otra parte, la descomposición de la serie temporal separa una serie en tres componentes: tendencia, estacionalidad y residuo. Por ejemplo, para la figura 7 tenemos lo siguiente:

- **Tendencia:** Muestra la dirección general a lo largo del tiempo. Vemos que hubo un gran aumento en el precio de cierre, especialmente a partir de finales de 2020.
- **Estacionalidad:** Muestra patrones repetitivos en intervalos regulares. Observamos un patrón estacional claro que se repite aproximadamente cada 30 días.
- **Residuo:** Muestra la variación en la serie que no puede ser explicada por la tendencia o la estacionalidad. Hay más variabilidad en los residuos hacia el final de la serie.

Finalmente, el *boxplot* de la figura 8 visualiza la distribución de los precios de cierre, donde observamos que la mayoría de los precios de cierre están concentrados en un rango bajo, pero que existen varios *outliers* que son precios mucho más altos.

4.3. Preprocesamiento de Datos

El preprocesamiento de datos es un paso crucial para asegurar la calidad del conjunto de datos antes de entrenar los modelos. Las etapas principales del preprocesamiento incluyen:

- **Limpieza de Datos:** Manejo de valores faltantes mediante imputación o eliminación, y eliminación de valores atípicos que pueden sesgar los resultados [20].

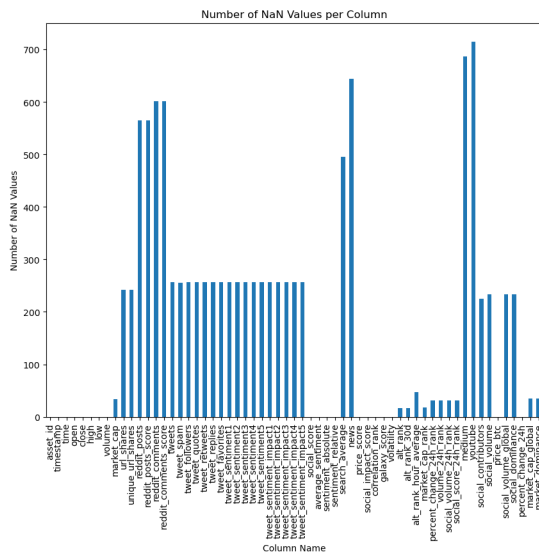


Figure 5. Histograma que muestra el número de valores nulos por columna dentro del conjunto de datos utilizado para este estudio.

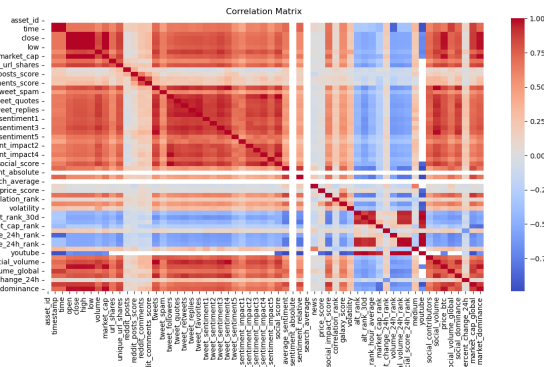


Figure 6. Matriz de correlaciones para el conjunto de datos problema.

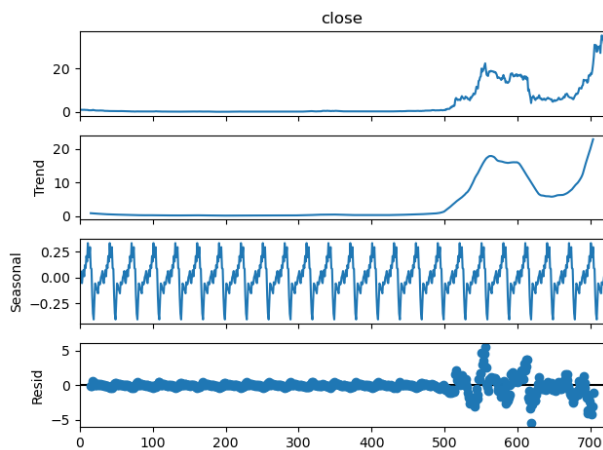


Figure 7. Descomposición del precio de cierre para un periodo de 30 días.

- **Normalización y Escalado:** Transformación de las características para que tengan una distribución uniforme, lo cual es esencial para el correcto funcionamiento de muchos algoritmos de aprendizaje automático [21].
- **Transformaciones Adicionales:** Incluye la codifi-

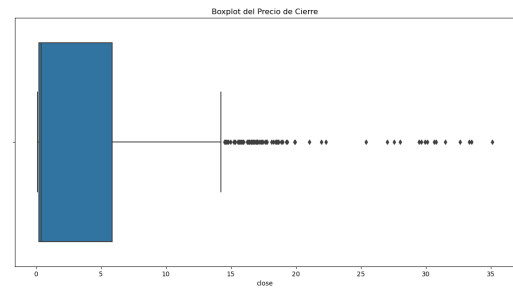


Figure 8. Boxplot para el precio de cierre.

cación de variables categóricas y la generación de nuevas características derivadas de las existentes para enriquecer el conjunto de datos.

- **División del Conjunto de Datos:** Separación del conjunto de datos en subconjuntos de entrenamiento, validación y prueba para evaluar el rendimiento de los modelos de manera imparcial [25].

4.4. Métricas de Evaluación

Las métricas de evaluación utilizadas para medir el rendimiento de los modelos incluyen:

- **Precisión:** Mide la exactitud de las predicciones en comparación con los valores reales.
- **Sign Accuracy:** Evalúa la precisión de la dirección de las predicciones, es decir, si el modelo predice correctamente si el valor subirá o bajará.
- **SMAPE (Symmetric Mean Absolute Percentage Error):** Una métrica de error relativa que es menos sensible a las escalas de los datos [15].
- **MASE (Mean Absolute Scaled Error):** Proporciona una escala de error que es independiente de la escala de los datos [15].

5. Resultados y Discusión

- **Comparación de rendimiento:** presentar y comparar el rendimiento de la red neuronal con otros modelos utilizando varias métricas (por ejemplo, precisión).
- **Ajuste de hiperparámetros:** analizar el proceso y los resultados del ajuste de hiperparámetros para el modelo.
- **Análisis de resultados:** proporcionar un análisis en profundidad de los resultados, incluido por qué nuestro modelo funciona mejor o peor que otros.
- **Visualización:** incluir gráficos y tablas relevantes para visualizar el rendimiento y el comportamiento de los modelos.

6. Limitaciones

7. Resumen y Conclusión

- **Resumen:** resumir los hallazgos clave de la investigación.
- **Conclusiones:** Sacar conclusiones basadas en los resultados y análisis experimentales.
- **Trabajo futuro:** sugerir posibles direcciones futuras para la investigación, como mejorar el modelo, aplicarlo a diferentes conjuntos de datos o probar con otros formatos de datos.

■ References

- [1] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *The Journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.
- [2] N.-F. Chen, R. Roll, and S. Ross, "Economic forces and the stock market," *The Journal of Business*, vol. 59, pp. 383–403, Jul. 1986.
- [3] C. W. J. Granger and P. Newbold, *Forecasting Economic Time Series*. Academic Press, 1986.
- [4] T. C. Mills, *Time Series Techniques for Economists*. Cambridge University Press, 1990.
- [5] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*. Prentice Hall, 1994.
- [6] J. D. Hamilton, *Time Series Analysis*. Princeton University Press, 1994.
- [7] S. Makridakis, S. C. Wheelwright, and R. J. Hyndman, *Forecasting: Methods and Applications*. John Wiley & Sons, 1998.
- [8] R. N. Mantegna and H. E. Stanley, *Introduction to Econophysics: Correlations and Complexity in Finance*. Cambridge University Press, 1999.
- [9] C. Chatfield, *Time-Series Forecasting*. Chapman & Hall/CRC, 2000.
- [10] C. Chatfield, *Time-Series Forecasting*. Chapman & Hall/CRC, 2001.
- [11] R. F. Engle, "Garch 101: The use of arch/garch models in applied econometrics," *Journal of Economic Perspectives*, vol. 15, no. 4, pp. 157–168, 2001.
- [12] J. H. Stock and M. W. Watson, "Forecasting using principal components from a large number of predictors," *Journal of the American Statistical Association*, vol. 97, no. 460, pp. 1167–1179, 2002.
- [13] G. P. Zhang, "Time series forecasting using a hybrid arima and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.
- [14] R. Tsay, *Analysis of financial time series*. Wiley-Interscience, 2005.
- [15] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International journal of forecasting*, vol. 22, no. 4, pp. 679–688, 2006.
- [16] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
- [17] R. Cont, "Stylized properties of asset returns," in May 2010, pp. 223–236, ISBN: 9780470057568. DOI: [10.1002/9780470061602.eqf19027](https://doi.org/10.1002/9780470061602.eqf19027).
- [18] W. McKinney, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, vol. 445, 2010, pp. 51–56.
- [19] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with python," in *Proceedings of the 9th Python in Science Conference*, Scipy, 2010.
- [20] F. Pedregosa, G. Varoquaux, A. Gramfort, et al., "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [21] G. Hinton, *Neural networks for machine learning*, Coursera, video lectures, 264, 2012.
- [22] K. Chen, Y. Zhou, and F. Dai, "A lstm-based method for stock returns prediction: A case study of china stock market," in *2015 IEEE International Conference on Big Data (Big Data)*, 2015, pp. 2823–2824.
- [23] F. Chollet, *Keras*, <https://github.com/fchollet/keras>, 2015.
- [24] M. Abadi, P. Barham, J. Chen, et al., "Tensorflow: A system for large-scale machine learning," in *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, 2016, pp. 265–283.
- [25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [26] W. Bao, J. Yue, and Y. Rao, "A lstm-based stock forecasting model with leading indicators," in *2017 IEEE International Conference on Big Data (Big Data)*, 2017, pp. 2355–2360.
- [27] G. Forestier, F. Petitjean, H. A. Dau, G. I. Webb, and E. Keogh, "Generating synthetic time series to augment sparse datasets," in *2017 IEEE International Conference on Data Mining (ICDM)*, IEEE, 2017, pp. 865–870.
- [28] A. e. a. Vaswani, "Attention is all you need. advances in neural information processing systems," *Curran Associates, Inc.*, 2017.
- [29] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.
- [30] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0377221717310652>.
- [31] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018, Available online at <https://otexts.com/fpp3/>.
- [32] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PLOS ONE*, vol. 13, no. 3, e0194889, 2018.
- [33] R. W. et al., "Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting," *MDPI*, 2019.
- [34] G. Lee, J. Han, J. Kim, and H. Ahn, "Pywavelets: A python package for wavelet analysis," *Journal of Open Source Software*, vol. 4, no. 36, p. 1237, 2019.
- [35] J. Liu, F. Chao, Y. Lin, and C. Lin, "Stock prices prediction using deep learning models," *CoRR*, vol. abs/1909.12227, 2019. [Online]. Available: <http://arxiv.org/abs/1909.12227>.

- [36] A. Nielsen, *Practical Time Series Analysis*. O'Reilly Media, Inc., 2019.
- [37] P. B. Bairui Du Delmiro Fernandez-Reyes, "Image processing tools for financial time series classification," *Cornell University*, 2020.
- [38] A. e. a. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv*, 2020.
- [39] C. R. Harris, K. J. Millman, S. J. van der Walt, *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.
- [40] D. e. a. Salinas, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, 2020.
- [41] J. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, and A. Troncoso, "Deep learning for time series forecasting: A survey," *Big Data* 9, vol. 9, 2020.
- [42] P. Virtanen, R. Gommers, T. E. Oliphant, *et al.*, "Scipy 1.0: Fundamental algorithms for scientific computing in python," *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.
- [43] M. L. Waskom, "Seaborn: Statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021.
- [44] N. C. et al., "Visual time series forecasting: An image-driven approach," *Association for Computing Machinery*, 2022.
- [45] B. et al., "Multi-modal financial time-series retrieval through latent space projections," *Association for Computing Machinery*, 2023.
- [46] J. W. et al., "Imaging feature-based clustering of financial time series," *PLoS ONE*, 2023.
- [47] Z. Z. et al., "Financial time series forecasting using cnn and transformer," *J. P. Morgan AI Research*, 2023.
- [48] P. Taylor, *Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025*, 2023. [Online]. Available: <https://www.statista.com/statistics/871513/worldwide-data-created/>.
- [49] B. de España, *¿qué son las criptomonedas?* 2024. [Online]. Available: <https://www.bde.es/wbe/es/areas-actuacion/politica-monetaria/preguntas-frecuentes/definicion-funciones-del-dinero/que-son-criptomonedas.html>.
- [50] Z. T. Tatane Khalid Sahib Mohamed Rida, "From time series to images: Revolutionizing stock market predictions with convolutional deep neural networks," *International Journal of Advanced Computer Science and Applications*, 2024.
- [51] Wikipedia, *Temporal series*, 2024. [Online]. Available: https://es.wikipedia.org/wiki/Serie_temporal.
- [52] Z. Zeng, R. Kaur, S. Siddagangappa, T. Balch, and M. Veloso, *From pixels to predictions: Spectrogram and vision transformer for better time series forecasting*, 2024. arXiv: 2403.11047 [cs.CV].