



Ingeniería y competitividad

ISSN: 0123-3033

ISSN: 2027-8284

Facultad de Ingeniería, Universidad del Valle

Lambis-Alandete, Erick; Jiménez-Gómez, Miguel; Velásquez-Henao, Juan D.
Comparación de algoritmos de Deep Learning para pronósticos en los precios de Criptomonedas
Ingeniería y competitividad, vol. 25, núm. 3, e-21312845, 2023, Agosto-Diciembre
Facultad de Ingeniería, Universidad del Valle

DOI: <https://doi.org/10.25100/iyc.v25i3.12845>

Disponible en: <https://www.redalyc.org/articulo.oa?id=291377795014>

- ▶ [Cómo citar el artículo](#)
- ▶ [Número completo](#)
- ▶ [Más información del artículo](#)
- ▶ [Página de la revista en redalyc.org](#)

UNIVERSIDAD DEL VALLE
redalyc.org

Sistema de Información Científica Redalyc
Red de Revistas Científicas de América Latina y el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso
abierto

Comparación de algoritmos de Deep Learning para pronósticos en los precios de criptomonedas

Comparison of Deep Learning algorithms for cryptocurrencies price forecasting

Erick Lambis-Alandete¹, Miguel Jiménez-Gómez¹, Juan D. Velásquez-Henao¹

¹ Universidad Nacional de Colombia, Facultad de Minas, Medellín, Colombia

Resumen

Debido al alto atractivo de las criptomonedas los inversionistas y los investigadores han prestado mayor atención en la previsión de sus precios. Con el desarrollo metodológico del Deep Learning, la previsión de las criptomonedas ha tenido mayor importancia en los últimos años. En este artículo, se evalúan cuatro modelos de Deep Learning: RNN, LSTM, GRU y CNN-LSTM con el objetivo de evaluar el desempeño en el pronóstico del precio de cierre diario de las dos criptomonedas más importantes: Bitcoin y Ethereum. Se utilizaron métricas de análisis de desempeño como MAE, RMSE, MSE y MAPE y como métrica de ajuste, el R2. Cada modelo de Deep Learning fue optimizado a partir de un conjunto de hiperparámetros y para diferentes ventanas de tiempo. Los resultados experimentales mostraron que el algoritmo RNN tuvo un rendimiento superior en la predicción del precio de Bitcoin y el algoritmo LSTM en el precio de Ethereum. Incluso, ambos métodos presentaron mejor desempeño con dos modelos de la literatura evaluados. Finalmente, la confiabilidad del pronóstico de cada modelo se evaluó analizando la autocorrelación de los errores y se encontró que los dos modelos más eficientes tienen alto poder de generalización.

Abstract

Due to the growth and interest that cryptocurrencies have generated nowadays, investors and researchers have paid more attention to forecasting the prices of cryptocurrencies. With the methodological development of Deep Learning, forecasting cryptocurrencies has become more important in recent years. In this paper, four Deep Learning models RNN, LSTM, GRU and CNN-LSTM are evaluated with the aim of evaluating the performance in forecasting the daily closing price of the two most important cryptocurrencies: Bitcoin and Ethereum. Performance analysis metrics such as MAE, RMSE, MSE and MAPE were used and as a fitting metric, the R2. Each Deep Learning model was optimized from a set of hyperparameters and for different time windows. Experimental results showed that the RNN algorithm had superior performance in predicting the Bitcoin price and the LSTM algorithm in predicting the Ethereum price. Even, both methods presented better performance with two literature models evaluated. Finally, the forecast reliability of each model was evaluated by analyzing the autocorrelation of the errors and the two most efficient models were found to have high generalization power.

Keywords:

Deep Learning, Cryptocurrencies, Time Series, investor

Palabras clave:

Deep Learning, Criptomonedas, Series de Tiempo, inversionista.

Cómo citar:

Lambis-Alandete, E., Jiménez-Gómez, M., Velásquez-Henao, J.D. Comparación de algoritmos de Deep Learning para pronósticos en los precios de criptomonedas. *Ingeniería y Competitividad*, 2023, 25(3); e-21312845. doi:

<https://doi.org/10.25100/iyv.v25i3.12845>

Recibido: 06-03-23

Aceptado 18-09-23

Correspondencia:

elambis@unal.edu.co, slumjimenezgo@unal.edu.co, jdvelasq@unal.edu.co

Este trabajo está licenciado bajo una licencia internacional Creative Commons Reconocimiento-No Comercial-CompartirIgual4.0.



Conflicto de intereses:

Ninguno declarado



¿Por qué se realizó?

El estudio se llevó a cabo con el objetivo de evaluar y comparar el rendimiento de varios métodos de Deep Learning, incluyendo RNN, LSTM, GRU, CNN y CNN-LSTM, en la predicción de los precios diarios de las criptomonedas BTC (Bitcoin) y ETH (Ethereum). El motivo principal detrás de este estudio fue abordar la creciente importancia de las criptomonedas en los mercados financieros y la necesidad de desarrollar modelos de pronóstico precisos para ayudar a los inversionistas y tomadores de decisiones a comprender y anticipar las fluctuaciones en los precios de estas monedas digitales altamente volátiles. Además, el estudio se propuso optimizar los hiperparámetros de cada modelo para maximizar su rendimiento y comparar los resultados con los métodos previamente reportados en la literatura científica para determinar si los modelos desarrollados en este estudio superaban a las técnicas existentes.

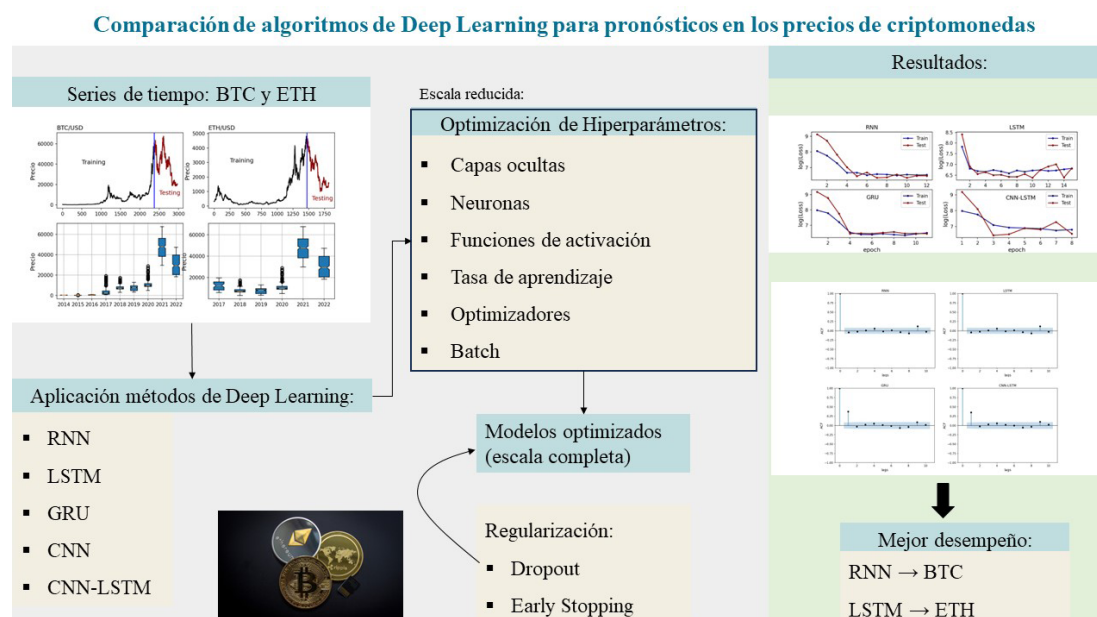
¿Cuáles fueron los resultados más relevantes?

Los resultados más relevantes del estudio incluyen la identificación de los modelos de Deep Learning óptimos para predecir los precios de BTC y ETH, así como la demostración de que estos modelos superaron a las técnicas previamente publicadas en la literatura científica. Entre los resultados específicos, se encontró que ciertos modelos, como RNN y LSTM, destacaron como los más efectivos en la predicción de los precios de criptomonedas. Estos modelos mostraron un rendimiento superior en términos de métricas de desempeño como el Error Absoluto Medio (MAE), el Error Cuadrático Medio (MSE), la Raíz del Error Cuadrático Medio (RMSE), el Error Porcentual Absoluto Medio (MAPE) y el coeficiente de determinación (R^2). Además, se logró optimizar los hiperparámetros de los modelos para mejorar aún más su precisión en las predicciones.

¿Qué aportan estos resultados?

Los resultados de este estudio aportan varias contribuciones significativas. En primer lugar, proporcionan a los inversionistas y tomadores de decisiones en el ámbito de las criptomonedas una herramienta más precisa y confiable para pronosticar los precios de BTC y ETH. Esto puede ayudar en la toma de decisiones financieras más informadas y en la gestión de riesgos en el mercado de las criptomonedas. Además, al comparar los modelos desarrollados en el estudio con los métodos existentes en la literatura, se demuestra que la aplicación de técnicas de Deep Learning puede superar a enfoques tradicionales en la predicción de series de tiempo, lo que tiene implicaciones más amplias en la aplicación de estas técnicas en otros campos financieros y de predicción de datos. En resumen, los resultados de este estudio contribuyen al avance en la precisión de las predicciones de criptomonedas y respaldan la eficacia de los modelos de Deep Learning en esta área.

Graphical Abstract



Introducción

Las criptomonedas son una moneda digital donde cada transacción se realiza de forma segura y se almacena en un bloque conocido como *Blockchain* (1). Las transacciones en la *Blockchain* son una forma nueva, novedosa y segura de transferir dinero saltándose las regulaciones monetarias. Cada una de estas transacciones están disponibles para todos los usuarios para su verificación, lo que da transparencia en el proceso (2). La primera criptomoneda se desarrolló en el año 2008 y se ha convertido en la moneda digital más valiosa. A través de los años se han introducido muchos tipos de criptomonedas y actualmente, la segunda en transacciones es *Ethereum*.

Las criptomonedas se han convertido en un activo importante en los mercados financieros debido a que ha aumentado el interés para realizar inversiones especulativas. Adicionalmente, se están popularizando como medio de pago para compras de bienes y servicios. El precio de las criptomonedas tiene un rango de movimiento más amplio que el precio del oro, los índices bursátiles y las materias primas. Esto se debe a que son independientes porque no tienen una institución que las controle lo que elimina la gestión monetaria de los bancos centrales (3).

Debido a esta independencia, los precios de las criptomonedas son ampliamente volátiles, son inestables y cambian rápidamente con el tiempo, por tanto, la predicción de los precios es una tarea desafiante y crucial para los investigadores (2,4). De esta manera, la alta volatilidad y el comportamiento errático en los precios de las criptomonedas han causado interés en la literatura científica por tratar de predecir los precios con diferentes técnicas. A continuación, se mencionan algunos estudios encontrados en la base de datos de *Scopus* por medio de la siguiente ecuación de búsqueda: *TITLE-ABS (cryptocurrency AND {deep learning} AND price)*.

Algunos autores (5) predicen los precios de las criptomonedas *Bitcoin* (BTC), *Digital Cash* y *Ripple* con los métodos de *Deep Learning* LSTM y una Red Neuronal de Regresión Generalizada (GRNN-*Generalized regression neural networks*), la cual es un sistema paralelo y basado en memoria que estima la superficie de regresión de una variable continua. Esta investigación encuentra que el modelo LSTM es de mayor carga computacional, pero obtiene mejores resultados en el reconocimiento de patrones no lineales, por tanto, es más eficiente para pronosticar la dinámica caótica inherente en los mercados de criptomonedas.

Otros autores (4) proponen un esquema híbrido entre LSTM y GRU con el cual obtuvieron mayor precisión en el pronóstico de los precios de las criptomonedas *Litecoin* y *Monero*. Por su parte, otros investigadores (6) aplican tres tipos de enfoques para predecir el precio de *Bitcoin*: modelo LSTM convencional, modelo LSTM con AR(2) (autorregresivo de segundo orden para eliminar la estacionariedad) y aplicación de un modelo AR(2) (modelo convencional de modelamiento de series de tiempo). Los autores determinaron que el modelo LSTM AR(2) superó al modelo LSTM convencional y al modelo AR(2).

Estos autores (7) implementaron modelos híbridos de CNN-LSTM y CNN-BiLSTM (LSTM Bidireccional) en los precios de *Bitcoin*, *Ethereum* y XRP. Con estos modelos los autores afirman que se puede obtener beneficios para desarrollar modelos de pronóstico fuertes, estables y confiables. (8) demostraron eficacia para pronosticar el precio del *Bitcoin* con modelos LSTM y GRU para manejar la volatilidad.

Entre los estudios más recientes se encuentran estos investigadores (9) quienes estudiaron la evaluación del rendimiento de un algoritmo genético con técnicas de *Deep Learning*

para predecir el precio de seis criptomonedas: *Bitcoin*, *Ethereum*, *Binance*, *Litecoin*, *Stellar Lumens* y *Dogecoin*. Los modelos de *Deep Learning* considerados fueron, CNN, redes neuronales directas profundas y GRU. Concluyeron que el mejor modelo es CNN. (10) otra investigación obtiene que el mejor modelo es el híbrido GRU-BiLSTM para pronosticar el precio de *Bitcoin*, *Ethereum*, *Ripple* y *Binance*. Por su parte, otros autores (1) con un modelo híbrido entre CNN-GRU, consideran que supera a los métodos existentes para pronosticar el precio de *Bitcoin*, *Ethereum* y *Ripple*. No obstante, en otra investigación (3) se presenta el modelo LSTM como el de mayor rendimiento y más efectivo para las criptomonedas AMP, *Ethereum*, *m*, *ElectroOptical System* y *XRP*. Por último, ellos (2) utilizaron LSTM para conocer los patrones dentro del precio de cierre de *EOS*, *Bitcoin*, *Ethereum* y *Dogecoin*. El rendimiento de este modelo de *Deep Learning* fue comparado con un modelo ARIMA. Similarmente, se comparó el método LSTM con otros modelos por medio de métricas como RMSE, MAE, MAPE y R^2 (11).

Esta investigación representa una contribución significativa al campo de la predicción de series de tiempo univariadas de precios de criptomonedas, mediante la aplicación de métodos avanzados de *Deep Learning*. En particular, este trabajo se destaca por su enfoque exhaustivo que abarca cinco métodos diferentes: *RNN (Recurrent Neural Networks)*, *LSTM (Long Short-Term Memory)*, *GRU (Gated Recurrent Units)*, *CNN (Convolutional Neural Networks)* y un método híbrido que combina CNN y LSTM. Una de las principales contribuciones radica en la rigurosa optimización de hiperparámetros realizada en este estudio. Esta optimización permite una comparación objetiva y fundamentada entre los métodos mencionados, lo que arroja luz sobre cuál de ellos presenta el mejor rendimiento en términos de precisión en la predicción de la serie de tiempo de precios de criptomonedas. Este enfoque metodológico es particularmente valioso, ya que proporciona una base sólida para la elección del modelo más adecuado, a diferencia de otros estudios en la literatura que pueden carecer de este nivel de análisis comparativo. Además, en este artículo se demuestra la mitigación del sobreajuste (*overfitting*), un problema crítico en la predicción de series de tiempo. Para abordar este desafío, se emplean técnicas de regularización como la detención temprana (*early stopping*) y el *Dropout*, que contribuye a mejorar la generalización de los modelos desarrollados. Otra contribución notable es la evaluación de la confiabilidad de los modelos propuestos mediante la aplicación de la prueba de ACF (*Autocorrelation Function*) a los residuales. Esta evaluación aporta una capa adicional de validación de la calidad de las predicciones, lo que aumenta la confianza en los resultados obtenidos. Por último, este estudio va más allá al comparar de manera sistemática los métodos de mejor desempeño identificados con otros métodos encontrados en la literatura. Esta comparación proporciona una perspectiva valiosa sobre el estado actual de la investigación en el campo y puede ayudar a orientar futuras investigaciones en la dirección más prometedora.

En conjunto, la aplicación y análisis de múltiples métodos de *Deep Learning*, la optimización de hiperparámetros, la atención al sobreajuste, la evaluación de la confiabilidad y la comparación con la literatura existente, hacen de este trabajo una valiosa contribución científica que avanza en la comprensión y el uso efectivo de la predicción de series de tiempo de precios de criptomonedas en un contexto financiero y tecnológico en constante evolución. Por su parte, la predicción precisa de los precios de las criptomonedas es esencial para diferentes agentes; por ejemplo, para los inversionistas que buscan maximizar sus ganancias y minimizar sus pérdidas, para las compañías de inversión en la gestión del riesgo y para las entidades de control para tomar decisiones en el ámbito regulatorio.

El resto de este artículo se estructura de la siguiente manera: en la sección 2 se presenta la metodología implementada con la exposición de los datos y los métodos de *Deep Learning*. Luego, en la sección 3 se discuten los resultados. Finalmente, en la sección 4 se presentan las conclusiones.

Metodología

Datos

En este documento se evalúa el rendimiento de varios métodos de *Deep Learning* para el pronóstico de las dos principales criptomonedas: *Bitcoin* (BTC) y *Ethereum* (ETH). Del sitio web <https://finance.yahoo.com> se obtuvieron los precios de cierre diarios de las dos criptomonedas. Para BTC se utilizaron los precios desde el 17 de septiembre de 2014 hasta el primero de noviembre de 2022, obteniendo un total de 2,968 precios. El conjunto de datos de ETH es de 1,819 precios con una ventana de tiempo desde el 9 de noviembre de 2017 hasta el primero de noviembre de 2022.

Los modelos de *Deep Learning* se prepararon con el conjunto de entrenamiento (*training*) y se evaluaron con el conjunto de prueba (*testing*) de cada criptomoneda. En BTC se evaluaron los modelos con los 600 precios más recientes, desde el 13 de marzo de 2021 hasta la fecha más reciente, primero de noviembre de 2022. Los demás precios se usaron como conjunto de *training*. En ETH se usaron los 350 precios más recientes como conjunto de *testing*, desde el 17 de noviembre de 2021 hasta el primero de noviembre de 2022. El conjunto de *training* de BTC es aproximadamente el 80% del conjunto de datos y el 20% aproximadamente es el conjunto de *testing*. Lo mismo se aplicó de forma aproximada para los conjuntos de datos de ETH.

La figura 1 muestra el comportamiento de cada criptomoneda, el conjunto de entrenamiento y de prueba, y la forma de la distribución de los precios por año representada por un *boxplot*. Las dos criptomonedas tuvieron un comportamiento similar en los últimos dos años: generaron precios máximos en el año 2021, pero disminuyeron en el último período. En los años anteriores al 2021, cada criptomoneda tenía un comportamiento con precios atípicamente altos cada año. De esta manera, se destaca que los conjuntos de *training* tienen patrones diferentes que los conjuntos de *testing*. Los modelos de *Deep Learning* se evaluaron (*testing*) donde los patrones en el precio mostraron rangos amplios y mayor dispersión.

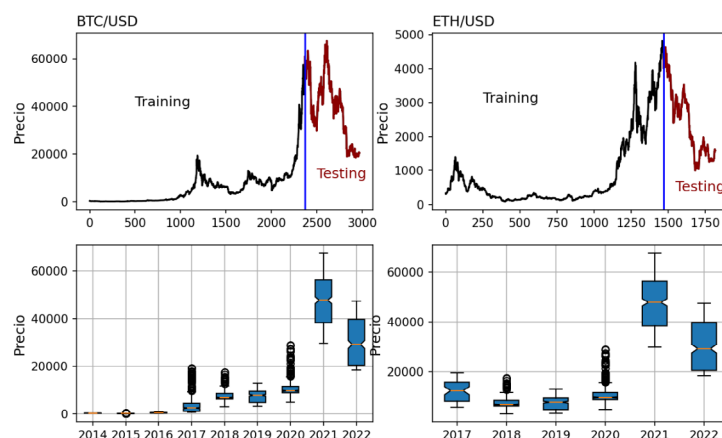


Figura 1. Precios diarios de BTC y ETH, y forma de la distribución por año.

Métodos de *Deep Learning*

En esta sección se introducirán los métodos de *Deep Learning* empleados para el pronóstico de las series de tiempo. En este artículo, evaluamos algunos de los algoritmos de aprendizaje profundo más exitosos y ampliamente utilizados para pronosticar los precios de las criptomonedas.

RNN

Una RNN (*Recurrent Neural Network*) es un tipo de arquitectura de una red neuronal profunda que tiene una estructura en la dimensión temporal, se parece a la red neuronal *Feedforward*, excepto que tiene conexiones que apuntan hacia atrás y envía la salida a sí misma. En las redes neuronales tradicionales, los *batch* o vectores de los datos de entrada son independientes entre sí, como consecuencia, no logran hacer uso de la información secuencial. Por el contrario, el método RNN agrega un estado oculto que almacena la información secuencial y arroja una salida dependiente de este estado oculto (12). Para series temporales, en cada paso de tiempo, RNN recibe el vector de entradas x_t y su propia salida del paso de tiempo anterior, es decir, la salida \hat{y} que también es una entrada para el siguiente estado de la neurona.

En cada paso de tiempo t , la neurona RNN recibe el vector de entrada x_t y la salida del paso de tiempo anterior, y_{t-1} . El resultado de la neurona, y_t , se calcula en la Ec. (1).

$$y_t = \phi(W_x x_t + W_y y_{t-1} + b) \quad (1)$$

Los pesos de las entradas son el vector W_x , los pesos para las salidas de los pasos anteriores se representan con W_y , b es el bias, y ϕ representa la función de activación.

LSTM

LSTM (*Long Short-Term Memory*) es una de las variaciones de la arquitectura de RNN. LSTM es un método de *Deep Learning* útil para predicción de datos secuenciales como las series de tiempo. Las redes RNN no tienen la capacidad de reconocer patrones a largo plazo debido al problema del desvanecimiento o explosión del gradiente, esto ocurre en el método de *backpropagation* cuando aprende los pesos y debe propagar la pendiente calculada con la regla de la cadena. A medida que los valores de la pendiente se propagan hacia atrás en la función de activación, la pendiente se vuelve extremadamente pequeña o extremadamente grande, lo que genera que el gradiente desaparezca o explote. LSTM utiliza celdas de memoria (*memory cells*) para superar este problema y puede aprender de dependencias a largo plazo. En 1997 se introdujo la primera LSTM que usaba celdas y compuerta de memoria para almacenar información durante largos períodos de tiempo o para olvidar información innecesaria (13).

Los LSTM actuales están compuestos por una celda de memoria (c_t) y tres compuertas: una compuerta de entrada (i_t), una compuerta de olvido (f_t), y una compuerta de salida (o_t). Cada compuerta tiene una función única: sigmoide (14). En el tiempo t , x_t representa la entrada de los datos secuenciales y h_t el estado oculto (13).

La compuerta de olvido (f_t) permite que la red neuronal LSTM ignore la información de la celda de estado del paso de tiempo anterior (c_{t-1}) o la conserve. Esta compuerta de olvido es alimentada por los valores de entrada x_t y el estado oculto del paso de tiempo anterior (h_{t-1}), este último, a su vez, es la salida del paso de tiempo anterior (\hat{y}_{t-1}). Estas dos

entradas se combinan (+) y pasan por una función sigmoide que genera un número entre 0 y 1 que multiplica elemento por elemento. Resultados cercanos a 1 en la compuerta de olvido permiten que la información se mantenga, en cambio, la información se olvida con un resultado cercano a 0 (15). En la Ec. (2) se muestra la forma de calcular el valor de la compuerta de olvido, donde w_{hf} es la matriz de pesos para la celda de estado del paso anterior (c_{t-1}), w_{if} es la matriz de pesos para los valores de entrada x_t y los bias son b_{hf} y b_{if} (11).

$$f_t = \sigma(w_{hf}h_{t-1} + w_{if}x_t + b_{hf} + b_{if}) \quad (2)$$

Luego, la celda de memoria (c_t) se actualiza mediante la Ec. (3). En esta fase la información anterior se recuerda o se olvida en c_t para activar la memoria a largo plazo correspondiente (16).

$$c_t = c_{t-1} \otimes f_t \quad (3)$$

Nuevamente las dos entradas, h_{t-1} y x_t pasan a través de una función tangente hiperbólica (tanh) con sus respectivos pesos, w_{hg} y w_{ig} , y sus respectivos bias, b_{hg} y b_{ig} . Este cálculo se muestra en la Ec. (4) y representa el almacenamiento de la información nueva del paso de tiempo actual (11). El objetivo de pasar la información por la función tangente hiperbólica es comprimir los valores entre -1 y 1 (15).

$$g_t = \tanh(w_{hg}h_{t-1} + w_{ig}x_t + b_{hg} + b_{ig}) \quad (4)$$

La compuerta de entrada (i_t) evalúa la información actual contenida en g_t . Esta compuerta funciona de forma similar a la compuerta de olvido, con las mismas entradas (h_{t-1} y x_t) que pasan a una función sigmoide. El resultado de i_t reduce el efecto de g_t al arrojar un valor cercano a 0 o permite pasar la información con un valor cercano a 1. En este último caso, se almacena la información en la memoria de largo plazo (c_t). La Ec. (5) muestra cómo se forma la salida de la compuerta de entrada (i_t) y la Ec. (6) muestra cómo se actualiza la celda de memoria (c_t) (11). Las matrices de pesos son w_{hi} y w_{ii} , y los bias son b_{hi} y b_{ii} .

$$i_t = \sigma(w_{hi}h_{t-1} + w_{ii}x_t + b_{hi} + b_{ii}) \quad (5)$$

$$c_t = c_{t-1} \otimes i_t g_t \quad (6)$$

El valor de c_t actualizado es el estado actual de la celda LSTM y es transferido al siguiente paso de tiempo.

La última compuerta, la compuerta de salida (o_t), determina cuánta información de la memoria a largo plazo (c_t) debe transferirse a la salida ($h_t = \hat{y}_t$). Esta compuerta también tiene una función sigmoide, pero tiene tres entradas: h_{t-1} , x_t y $\tanh(c_t)$. El valor de c_t primero pasa por la función tangente hiperbólica antes de multiplicarse por la compuerta de salida para generar la salida h_t . Las Ec. (7) y (8) muestran cómo se generan estos resultados (11).

$$o_t = \sigma(w_{ho}h_{t-1} + w_{io}x_t + b_{ho} + b_{io}) \quad (7)$$

$$h_t = o_t \otimes \tanh(c_t) \quad (8)$$

Por lo tanto, las redes neuronales LSTM superan el problema de la memoria de corto plazo con este diseño de tres compuertas y la celda de memoria de largo plazo. Así que la información en los pasos de tiempo anteriores puede reflejarse en el estado actual de la celda (15).

GRU

Los modelos GRU (*Gated Recurrent Unit*) son otra variación a la red neuronal recurrente. El algoritmo GRU tiene una arquitectura simplificada que LSTM y también permite la memoria de largo plazo. GRU tiene dos compuertas: la compuerta de actualización (*update gate*) z_t y la compuerta de reinicio (*reset gate*) r_t .

En las dos compuertas de la red GRU, las entradas son las mismas: resultado del paso de tiempo anterior (h_{t-1}) y los valores de entrada (x_t) (17). La función de activación de las compuertas es la sigmoide y el resultado se genera con las Ec. (9) y (10):

$$z_t = \sigma(w_{iz}x_t + w_{hz}h_{t-1} + b_r) \quad (9)$$

$$r_t = \sigma(w_{ir}x_t + w_{hr}h_{t-1} + b_r) \quad (10)$$

Donde w_{iz} y w_{ir} son las matrices de pesos para los valores de entrada x_t , w_{hz} y w_{hr} son las matrices de pesos para la memoria de largo plazo del paso de tiempo anterior h_{t-1} y b_r es el bias.

En el resultado de la red GRU, h_t , se muestra en la Ec. (11).

$$h_t = z_t \otimes h_{t-1} + (1 - z_t)g_t \quad (11)$$

Donde g_t es la función de activación tangente hiperbólica por defecto. pero se podría cambiar por cualquier otra. En la Ec. (12) se muestra cómo se genera g_t .

$$g_t = \tanh(w_{ig}x_t + w_{hg}(r_t \otimes h_{t-1}) + b_g) \quad (12)$$

CNN

La arquitectura de las redes CNN (*Convolutional Neural Networks*) se compone de tres capas principales: capa de convolución, capa de agrupamiento o *pooling* y capa totalmente conectada. La capa de convolución extrae características de los datos de entrada. Esta capa se compone de varios *kernels* de convolución que se utilizan para calcular varios *Feature Maps* (18). La operación de la capa de convolución se expresa en la Ec. (13).

$$h = f(x * W + b) \quad (13)$$

Donde x son los datos de entrada, W los pesos, $*$ operación de convolución, y el bias es b . $f(\cdot)$ es la función de activación de la capa de convolución (19). Por medio de esta operación se crean los *Feature Maps*. Para datos secuenciales, se usa la estructura convolucional unidimensional (14).

La capa de agrupación o *pooling* reduce la dimensión de los *Feature Maps*. Esta reducción acelera el cálculo y evita el sobreajuste (14). Las operaciones de agrupamiento de mayor uso son la media y agrupación máxima (*max pooling*). En este artículo se utiliza el *max pooling* debido a que reduce efectivamente la dimensionalidad de los datos y mejora la solidez de los resultados (14).

Los *kernels* de las primeras capas de convolución están diseñados para detectar características de bajo nivel, mientras que los *kernels* de las capas superiores aprenden a identificar características más abstractas. Al apilar varias capas de convolución y de agrupamiento, se puede detectar características de nivel superior.

Después de apilar varias capas de convolución y de agrupamiento, sigue una o varias capas totalmente conectadas. Estas capas totalmente conectadas son opcionales y pueden reemplazarse por una sola capa de convolución de 1×1 para generar la salida de la red neuronal (18).

Método híbrido: CNN-LSTM

El método híbrido combina la red neuronal CNN con la arquitectura LSTM para realizar el pronóstico. El conjunto de datos primero pasa por las capas de convolución para luego pasar por las capas de una red tipo LSTM.

Optimización de hiperparámetros

Los modelos de *Deep Learning* tienen diferentes hiperparámetros que determinan la estructura de la red neuronal y la manera de entrenamiento. El rendimiento de los modelos varía según el conjunto de hiperparámetros elegidos, por tanto, para lograr un rendimiento óptimo, se deben seleccionar los hiperparámetros que generan mejores resultados (19). En la búsqueda de los hiperparámetros se prueba la red neural con las combinaciones posibles de hiperparámetros.

Los hiperparámetros a optimizar son: capas ocultas, cantidad de neuronas por capa, funciones de activación, tasa de aprendizaje, optimizadores y tamaño del batch. La tabla 1 muestra los resultados experimentales de las configuraciones óptimas.

Tabla 1. Hiperparámetros óptimos de los modelos para BTC y ETH

Hiperparámetros óptimos	BTC			ETH		
	RNN	LSTM	GRU	RNN	LSTM	GRU
Capas ocultas	2	2	1	1	1	1
Neuronas por cada	50	50	50	20	50	20
Funciones de activación	ELU	selu	selu	selu	ReLU	ReLU
Tasa de aprendizaje	0.001	0.001	0.1	0.01	0.001	0.1
Optimizadores	Adam	RMSProp	Adam	Adam	RMSProp	Adam
Tamaño del Batch	128	32	64	32	128	128

Resultados y discusión

Comparación modelos optimizados

En esta sección se evalúa el desempeño de cada modelo de *Deep Learning* optimizado y se hacen comparaciones con los modelos de predicción encontrados en la literatura. Cada modelo fue entrenado de nuevo, pero a escala completa con cada uno de los hiperparámetros optimizados.

La tabla 2 presenta la comparación de las métricas de desempeño de los métodos de *Deep Learning* para el *Bitcoin* y *Ethereum*. Cada método optimizado se entrenó con el conjunto de *train* correspondiente y se calcularon las métricas de desempeño sobre el conjunto de *test* de cada activo.

Tabla 2. Métricas de desempeño de los métodos para BTC y ETH.

	MAE	MSE	RMSE	MAPE	R ²
BTC					
RNN	1,048		1,508	2.64%	0.9873
LSTM	1,066		1,535	2.71%	0.9868
GRU	1,205		1,666	3.06%	0.9844
CNN-LSTM	1,226		1,666	3.08%	0.9844
ETH					
RNN	95	15,334	124	4.13%	0.9830
LSTM	77	10,975	105	3.31%	0.9885
GRU	93	15,351	124	4.13%	0.9835
CNN-LSTM	122	23,214	152	5.29%	0.9751

Conforme a los resultados presentados en la tabla 2, se observa un rendimiento destacado en los cuatro métodos DL en lo que respecta a la predicción de los precios de las criptomonedas BTC y ETH, evidenciado por la obtención de coeficientes de determinación (R²) superiores a 0.97. Específicamente, en el caso de BTC, se destaca el rendimiento sobresaliente del método RNN. Este método exhibe los valores más bajos en métricas tales como el MAE, MSE, RMSE y MAPE, y además, obtiene el R² más elevado. La magnitud del MAE, que se sitúa en 1,048 USD, denota que, en el proceso de pronóstico de los precios de BTC, el modelo presenta una desviación promedio de 1,048 USD por día. En consonancia, se constata que el método RNN exhibe los valores más favorables en las métricas MSE y RMSE, lo cual indica una mayor calidad en las proyecciones. Conviene señalar que el MSE, al calcular el promedio de los cuadrados de los errores, manifiesta una mayor sensibilidad a valores atípicos en comparación con el MAE. Por último, el MAPE correspondiente al modelo, se cifra en 2.64%, reflejando el porcentaje de error en las predicciones con respecto a los valores reales.

En el marco de los resultados para el precio de la criptomoneda ETH, se determinó que el mejor método para el pronóstico es el de LSTM. En primer lugar, el MAE de 77 indica que, en promedio, las predicciones generadas por el modelo LSTM se desvían aproximadamente en 77 unidades de la moneda con respecto a los valores reales de ETH. Este bajo nivel de error absoluto es un indicativo de la capacidad del modelo para predecir los precios con precisión. El MSE de 10,975, aunque es un valor más elevado que el MAE, es de suma relevancia. El MSE promedia los cuadrados de las diferencias entre las predicciones y los valores reales, lo que lo hace más sensible a las desviaciones significativas. A pesar de este aspecto, el MSE es considerablemente bajo, lo que subraya

la capacidad del modelo LSTM para minimizar errores sustanciales en la predicción. El RMSE de 105, que se obtiene al calcular la raíz cuadrada del MSE, ofrece una métrica más intuitiva, ya que se expresa en la misma unidad que los valores originales de la serie de tiempo. En este contexto, el RMSE refleja que las predicciones del modelo tienen una desviación promedio de aproximadamente 105 unidades de la moneda ETH con respecto a los valores reales. El MAPE del 3.31% es un indicador particularmente destacado. Este valor representa el porcentaje de error promedio en las predicciones en comparación con los valores reales. Un MAPE bajo demuestra que el modelo LSTM es capaz de realizar pronósticos altamente precisos y que las desviaciones porcentuales de las predicciones con respecto a los valores reales son mínimas. Por último, el coeficiente de determinación (R^2) de 0.9885 es una métrica que evalúa cuánta varianza en los datos es explicada por el modelo. En este caso, un R^2 tan cercano a 1 indica que el modelo LSTM es altamente efectivo en capturar y explicar la variabilidad presente en la serie de tiempo de ETH, lo que sugiere un ajuste excepcional del modelo a los datos observados. En resumen, el método LSTM demuestra ser altamente competente en la predicción de la serie de tiempo de ETH, respaldado por sus bajos valores de MAE, MSE, RMSE, MAPE y un elevado coeficiente R^2 . Estos resultados respaldan su elección como el método más efectivo en este contexto de pronóstico.

En el código en *Python* se usó un *Callbacks* con *patience* igual a 5 con el fin de optimizar el tiempo de ejecución. De esta manera, el entrenamiento de capa modelo se detenía si después de cinco *epochs* no se obtenía una mejora en la función de pérdida (*Loss*). La figura 2 muestra la evolución del entrenamiento a medida que aumenta la cantidad de *epochs* por cada modelo de *Deep Learning* para BTC, en esta figura se transformó cada función de pérdida aplicando logaritmo con el fin de visualizarlo mejor. El modelo que más *epochs* necesitó fue LSTM con 14 *epochs*, por el contrario, el modelo CNN-LSTM necesitó ocho *epochs*. Se observa que las funciones de pérdida de los cuatro modelos en el conjunto de *test* convergen rápidamente a la función de pérdida del conjunto de *train*, entre tres y cuatro *epochs* los modelos convergen. Adicionalmente, no se evidenció sobre ajuste (*overfitting*) debido al uso del *Callbacks* y al *Dropout* de 0.20 aplicado en las capas ocultas de cada una de las arquitecturas.

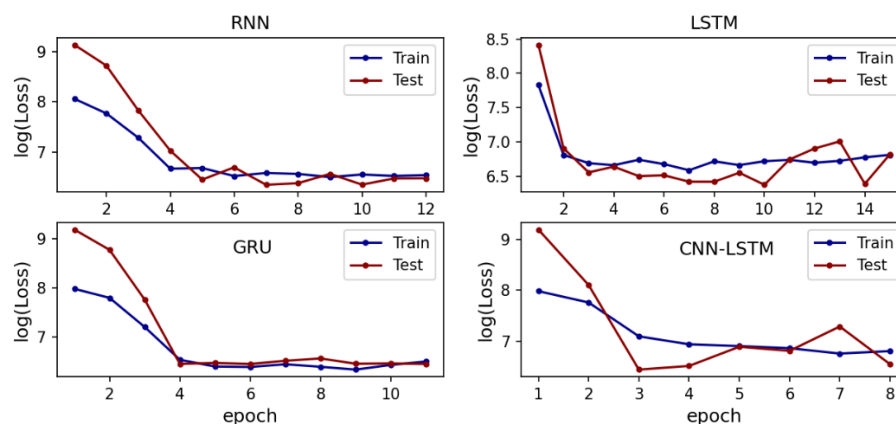


Figura 2. Logaritmo de las funciones de pérdida con respecto a la cantidad de *epoch* de los modelos optimizados para BTC.

La figura 3 muestra el comportamiento de la función de pérdida por cada *epoch* para los modelos aplicados en ETH. Se aplicó el logaritmo al eje y para visualizar mejor el cambio por *epoch*. Se destaca que la función de pérdida sobre el conjunto de prueba es menor que el conjunto de entrenamiento en los cuatro métodos. Lo anterior es una muestra de que no se tienen modelos sobre entrenados (con *overfitting*), esto se evitó con una tasa de *Dropout* de 0.20 en cada capa oculta.

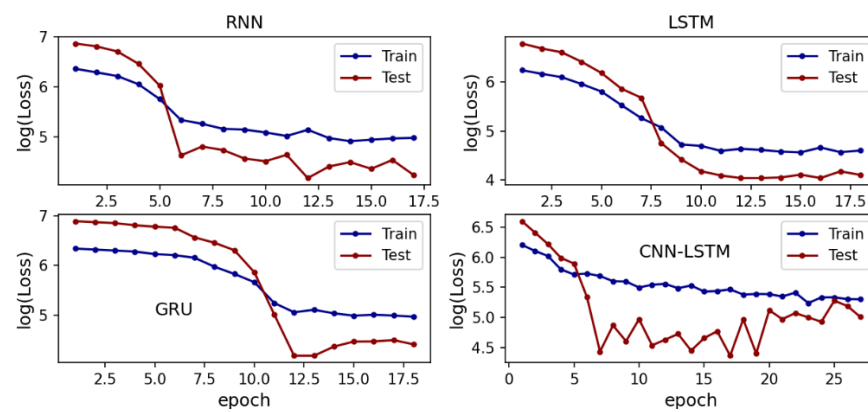


Figura 3. Logaritmo de las funciones de pérdida con respecto a la cantidad de *epoch* de los modelos optimizados para ETH.

Para evaluar la confiabilidad del pronóstico, se calculó la Función de Autocorrelación (*ACF-Autocorrelation Function*) para cada uno de los modelos optimizados. El objetivo del ACF es probar la existencia de autocorrelación entre los residuos. La ACF se obtiene a partir de la correlación lineal de cada residuo con los demás en diferentes rezagos (*lags*) para examinar si el modelo se ha ajustado correctamente a la serie de tiempo y si los residuos están idénticamente distribuidos y asintóticamente independientes (20). La figura 4 presenta la función ACF para los cuatro modelos optimizados para BTC, los puntos por fuera de las bandas muestran la intensidad de la autocorrelación temporal. Las bandas se construyen suponiendo que los residuos siguen una distribución de probabilidad gaussiana. De esta forma, en la figura 4 se observa que existe autocorrelación con el noveno rezago en los modelos RNN y LSTM; sin embargo, esta autocorrelación es de baja intensidad. Lo anterior se contrasta con lo hallado en las métricas de desempeño en que son los dos modelos de mejor rendimiento para BTC. En contraste, los modelo GRU y CNN-LSTM mostraron autocorrelación alta con el primer rezago, esto indica ineficiencias para la predicción y es evidencia de que no capturan toda la estructura determinística de la serie de tiempo.

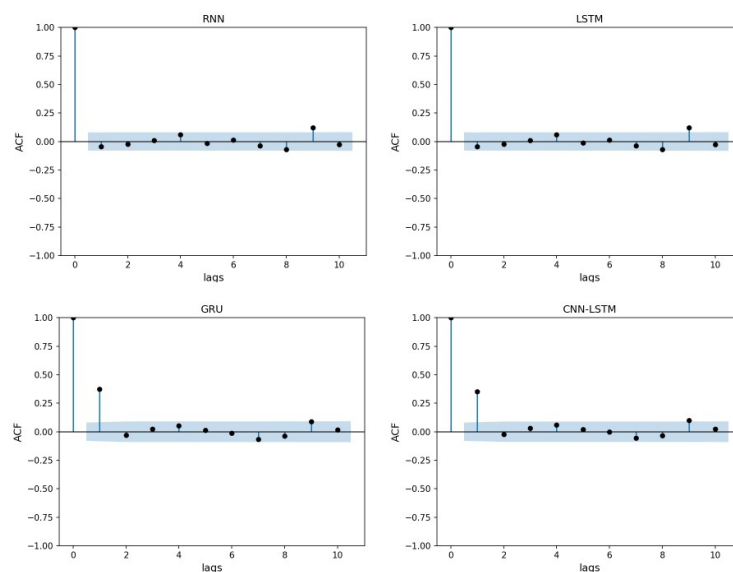


Figura 4. Función ACF de los residuos sobre el conjunto de test de Bitcoin de los modelos optimizados.

La ACF de los residuales de los modelos aplicados en ETH se muestran en la figura 5. Se observa que los modelos RNN, GRU y CNN-LSTM tienen presencia de rezagos en los residuales. Existe correlación alta entre el primer rezago en RNN y en los dos primeros rezagos en GRU y CNN-LSTM; sin embargo, el modelo optimizado LSTM no presenta autocorrelación entre los rezagos de los residuos, en consecuencia, este modelo es confiable para pronosticar el precio de *Ethereum* y tiene la capacidad de capturar la estructura determinística de la serie de tiempo.

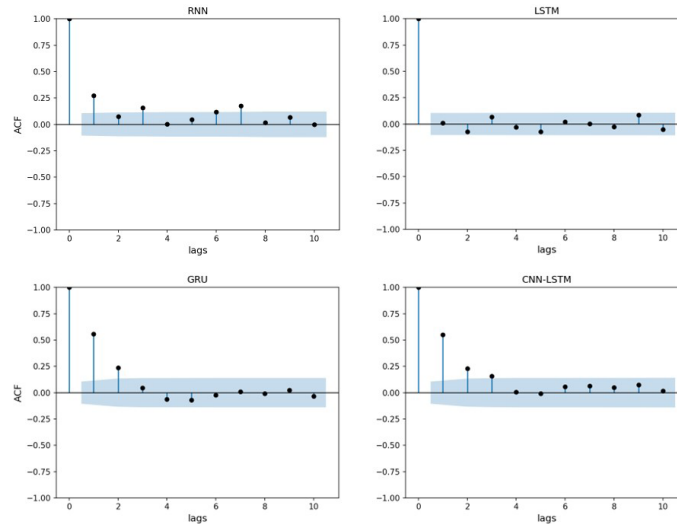


Figura 5. Función ACF de los residuos sobre el conjunto de test de Ethereum de los modelos optimizados.

Comparación modelos de la literatura

En esta sección, se compara el método de *Deep Learning* optimizado de mejor desempeño hallado por cada criptomoneda con los métodos encontrados en la literatura en el último año y que realizaban pronóstico sobre el precio del *Bitcoin* y *Ethereum*. Todos los modelos fueron entrenados con el mismo conjunto de datos.

El modelo de estos autores (2) contiene una arquitectura LSTM con una capa oculta con 250 neuronas, seguido de una capa de *Dropout* con una tasa de 0.20, la ventana de tiempo para pronosticar es de 10 días para predecir el siguiente día, las neuronas tienen la función de activación ReLU, el optimizador es Adam, tasa de aprendizaje de 0.001 y tamaño del *Batch* de 64. Este modelo se entrenó con 100 *epochs*.

El modelo presentado por estos investigadores (1) es un híbrido entre CNN y GRU. Tiene una sola capa de convolución con 256 filtros, tamaño del *Kernel* de 1. En GRU tiene dos capas ocultas, cada una con 256 neuronas, función de activación ReLU. Usa el optimizador Adam, ventana de tiempo de un día, *Batch* de 64 y el modelo se entrenó con 100 *epochs*.

La tabla 3 presenta la comparación entre los dos modelos anteriores (2) y (1) y los dos modelos optimizados que se proponen en este artículo: RNN para BTC y LSTM para ETH. El modelo RNN optimizado presenta mejor desempeño en el pronóstico en el precio del *Bitcoin* que los dos modelos de la literatura que se componen de una estructura LSTM (2) y una estructura híbrida CNN-GRU (1). Incluso, los demás modelos optimizados, LSTM, GRU y CNN-LSTM son más precisos para modelar el precio de BTC. Por su parte, el modelo

LSTM optimizado propuesto en este artículo tiene mejores resultados en las métricas de desempeño, pero con resultados aproximados que el método (1) CNN-GRU de la literatura. Estos dos métodos tienen mejor desempeño que el modelo de la literatura LSTM (2) el cual tiene un RMSE de 144, valor más alto que los modelo RNN y GRU optimizado, pero un mejor valor que el híbrido CNN-LSTM optimizado.

Tabla 3. Comparación de modelos.

	MAE	MSE	RMSE	MAPE	R ²
BTC					
LSTM (2)	1,958		2,643	4.93%	0.9603
CNN-GRU (1)	1,495		1,971	3.84%	0.9783
RNN optimizado	1,048		1,508	2.64%	0.9873
ETH					
LSTM (2)	108	20,700	144	4.86%	0.9767
CNN-GRU (1)	76.59	10,964.5	104.7	3.309%	0.98852
LSTM optimizado	76.58	10,974.6	104.8	3.310%	0.98851

El análisis de los residuales de los dos modelos de la literatura se muestra en la figura 6 con el AFC de los residuales de BTC. El modelo CNN-GRU (1) presenta baja autocorrelación con el noveno rezago, por lo que se determina que el modelo es confiable para predecir el precio del *Bitcoin*. Por el contrario, el modelo de la literatura LSTM (2) no tiene la capacidad de predecir de forma confiable el precio de BTC debido a que se encontró alta autocorrelación entre los primeros cinco rezagos. Esto último es evidencia de que este modelo no es confiable en la predicción del *Bitcoin*.

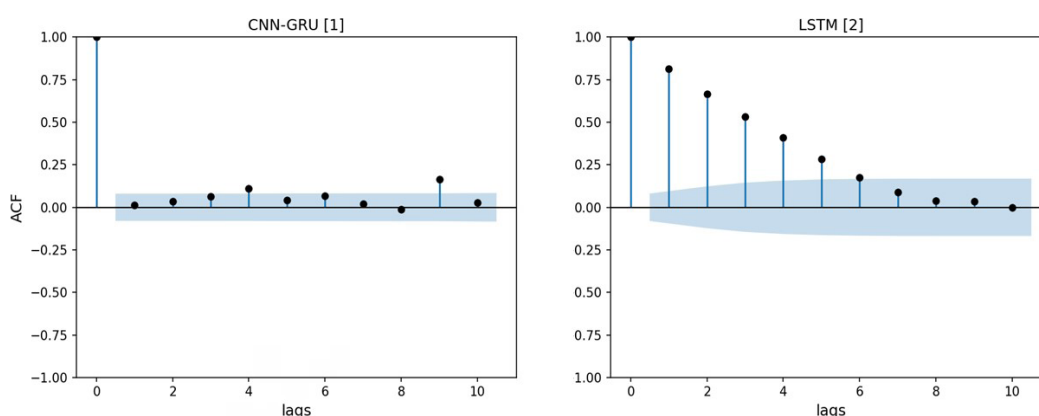


Figura 6. Función ACF de los residuos sobre el conjunto de test de BTC de los modelos de la literatura.

Finalmente, en la figura 7 el modelo CNN-GRU (1) no tiene rezago con autocorrelación, pero el modelo LSTM (2) tiene los residuos de los dos primeros rezagos con alta autocorrelación. Esta evidencia es similar a la encontrada con estos dos modelos aplicados sobre el precio de BTC.

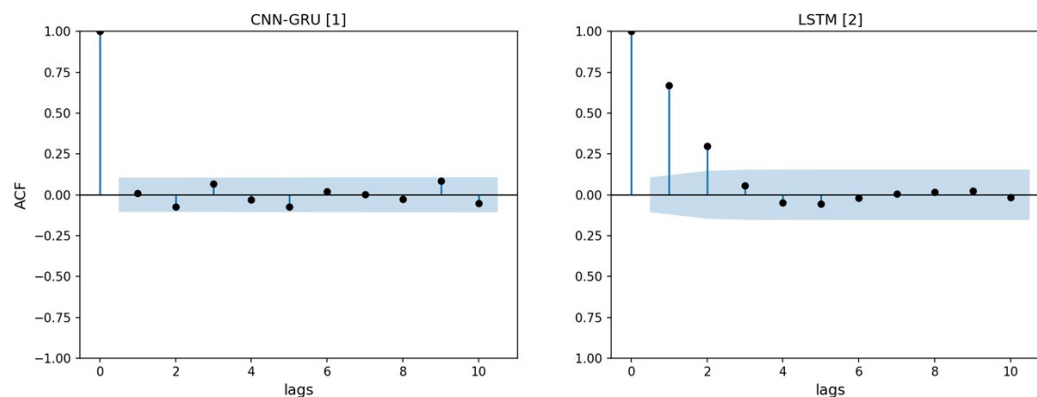


Figura 7. Función ACF de los residuos sobre el conjunto de test de ETH de los modelos de la literatura.

Conclusiones

Pronosticar el mercado de criptomonedas es una tarea desafiante y una preocupación para los inversionistas debido a su comportamiento altamente volátil. De esta manera, en los últimos años ha aumentado el interés de los investigadores por tratar de predecir el precio de las criptomonedas con técnicas de *Deep Learning*, las cuales tienen la capacidad de capturar patrones no lineales y con dependencias de largo plazo.

En este artículo se utilizó una metodología innovadora de inteligencia artificial en lugar de los modelos tradicionales de series de tiempo para estimar el precio de las dos criptomonedas de mayor transacción: *Bitcoin* y *Ethereum*. Se realizó un análisis experimental extenso y detallado teniendo en cuenta la evaluación del desempeño de los siguientes modelos de *Deep Learning*: RNN, LSTM, GRU y CNN-LSTM. Además, se estudió la fiabilidad y eficiencia de las predicciones de cada modelo examinando la autocorrelación de los residuos.

Este estudio ha arrojado luz sobre las notables implicaciones prácticas de la aplicación de métodos DL en el pronóstico de series de tiempo univariadas. Nuestra investigación ha revelado que estos enfoques representan una herramienta poderosa y efectiva para modelar y predecir con precisión las dinámicas de datos temporales. Específicamente, los modelos RNN y LSTM demostraron un rendimiento sobresaliente en términos de precisión de pronóstico, minimizando el MAE, el MSE, el RMSE y el MAPE. También, el estudio ofrece perspectivas valiosas sobre las implicaciones prácticas que los métodos de DL tienen para los tomadores de decisiones e inversionistas en el contexto del pronóstico de series de tiempo univariadas. Nuestros hallazgos resaltan que la aplicación de estas técnicas avanzadas pueden revolucionar la toma de decisiones en una variedad de industrias y sectores financieros. Para los tomadores de decisiones, los modelos de DL proporcionan la capacidad de obtener predicciones más precisas y detalladas sobre las tendencias futuras. En el ámbito de la inversión, los métodos de DL ofrecen una herramienta poderosa para analizar y pronosticar los movimientos de los activos financieros. La precisión mejorada de las predicciones puede llevar a estrategias de inversión más informadas y rentables. Sin embargo, es importante destacar que la adopción de estos requiere un enfoque prudente y una comprensión sólida de su funcionamiento, ya que los modelos de DL son inherentemente complejos.

Los resultados experimentales mostraron que, para el precio diario del *Bitcoin*, el modelo RNN tiene mayor rendimiento que los otros métodos, incluyendo los dos métodos de la literatura debido a que generó menos errores predictivos (MAE, RMSE, MSE y MAPE) y una evaluación más precisa (R^2). El modelo RNN presentó el RMSE más bajo de 1,508 para el conjunto de datos de Bitcoin. No obstante, los cuatro métodos evaluados presentaron R^2 mayores a 0.98, demostrando alta precisión de pronóstico, pero los modelos GRU y CNN-LSTM no capturan adecuadamente la estructura determinística del precio de la criptomoneda porque presentaron alta autocorrelación en el primer rezago de los errores. Posteriormente, los dos modelos de la literatura evaluados fueron el de CNN-GRU (1) y LSTM (2), los cuales lograron buen ajuste, pero en menor medida que los modelos optimizados y además, el modelo LSTM (2) presentó alta autocorrelación en los primeros cinco rezagos de los errores.

Por su parte, para el precio de cierre de ETH, los cuatro modelos optimizados tienen R^2 alto, siendo CNN-LSTM el de más bajo rendimiento con un R^2 de 0.97. En este caso, el modelo LSTM es de mejores resultados en las cinco métricas de desempeño y presentó un RMSE de 105. Comparando estos métodos con los de la literatura, se encontró que CNN-GRU (1) tiene un desempeño similar, pero LSTM (2) tiene rendimiento contrario siendo similar al modelo CNN-LSTM. Ambos métodos presentaron problemas en el pronóstico por la presencia de autocorrelación en los residuos.

Como trabajo futuro se podría evaluar y optimizar el modelo híbrido CNN-GRU e implementar la variación Bidireccional para GRU y LSTM y que incluya un modelo híbrido con CNN. En los datos de entrada se podría analizar primero la serie de tiempo si presenta autocorrelación para aplicar diferenciación antes de entrar a las redes neuronales y así ayudar a los modelos para no capturar más patrones. Por último, se podría combinar técnicas de pronóstico de series de tiempo tradicionales con los modelos de *Deep Learning* con el fin de aumentar la capacidad de pronóstico.

Bibliografía

1. Kang CY, Lee CP, Lim KM. Cryptocurrency Price Prediction with Convolutional Neural Network and Stacked Gated Recurrent Unit. *Data*. 2022;7(11). <https://doi.org/10.3390/data7110149>
2. Fleischer JP, von Laszewski G, Theran C, Bautista YJP. Time Series Analysis of Cryptocurrency Prices Using Long Short-Term Memory. *Algorithms*. 2022;15(7). <https://doi.org/10.3390/a15070230>
3. Ammer MA, Aldhyani THH. Deep Learning Algorithm to Predict Cryptocurrency Fluctuation Prices: Increasing Investment Awareness. *Electron*. 2022;11(15). <https://doi.org/10.3390/electronics11152349>
4. Patel MM, Tanwar S, Gupta R, Kumar N. A Deep Learning-based Cryptocurrency Price Prediction Scheme for Financial Institutions. *J Inf Secur Appl*. 2020;55. <https://doi.org/10.1016/j.jisa.2020.102583>
5. Lahmiri S, Bekiros S. Cryptocurrency forecasting with deep learning chaotic neural networks. *Chaos, Solitons and Fractals*. 2019;118:35-40. <https://doi.org/10.1016/j.chaos.2018.11.014>
6. Wu C-H, Lu C-C, Ma Y-F, Lu R-S. A new forecasting framework for bitcoin price with LSTM. In: *IEEE International Conference on Data Mining Workshops, ICDMW*. 2019. p. 168-75. <https://doi.org/10.1109/ICDMW.2018.00032>
7. Pintelas E, Livieris IE, Stavroyiannis S, Kotsilieris T, Pintelas P. Investigating the Problem of Cryptocurrency Price Prediction: A Deep Learning Approach. Vol. 584 IFIP, *IFIP Advances in*

- Information and Communication Technology. 2020. 99-110 p. https://doi.org/10.1007/978-3-030-49186-4_9
8. Awoke T, Rout M, Mohanty L, Satapathy SC. Bitcoin Price Prediction and Analysis Using Deep Learning Models. Vol. 134, Lecture Notes in Networks and Systems. 2021. 631-640 p. https://doi.org/10.1007/978-981-15-5397-4_63
9. Oyedele AA, Ajayi AO, Oyedele LO, Bello SA, Jimoh KO. Performance evaluation of deep learning and boosted trees for cryptocurrency closing price prediction. *Expert Syst Appl.* 2023;213. <https://doi.org/10.1016/j.eswa.2022.119233>
10. Ferdiansyah, Othman SH, Radzi RZM, Stiawan D, Sutikno T. Hybrid gated recurrent unit bidirectional-long short-term memory model to improve cryptocurrency prediction accuracy. *IAES Int J Artif Intell.* 2023;12(1):251-61. <https://doi.org/10.11591/ijai.v12.i1.pp251-261>
11. Nasirtafreshi I. Forecasting cryptocurrency prices using Recurrent Neural Network and Long Short-term Memory. *Data Knowl Eng.* 2022;139. <https://doi.org/10.1016/j.datak.2022.102009>
12. Bao W, Yue J, Rao Y. A deep learning framework for financial time series using stacked autoencoders and longshort term memory. *PLoS One.* 2017;12(7):1-24. <https://doi.org/10.1371/journal.pone.0180944>
13. Kim HY, Won CH. Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Syst Appl [Internet].* 2018;103:25-37. Available from: <https://doi.org/10.1016/j.eswa.2018.03.002>
14. Zha W, Liu Y, Wan Y, Luo R, Li D, Yang S, et al. Forecasting monthly gas field production based on the CNN-LSTM model. *Energy [Internet].* 2022;260(August):124889. Available from: <https://doi.org/10.1016/j.energy.2022.124889>
15. Luo H, Wang D, Cheng J, Wu Q. Multi-step-ahead copper price forecasting using a two-phase architecture based on an improved LSTM with novel input strategy and error correction. *Resour Policy.* 2022;79(August). <https://doi.org/10.1016/j.resourpol.2022.102962>
16. Lawi A, Mesra H, Amir S. Implementation of Long Short-Term Memory and Gated Recurrent Units on grouped time-series data to predict stock prices accurately. *J Big Data [Internet].* 2022;9(1). Available from: <https://doi.org/10.1186/s40537-022-00597-0>
17. Tang H, Ling X, Li L, Xiong L, Yao Y, Huang X. One-shot pruning of gated recurrent unit neural network by sensitivity for time-series prediction. *Neurocomputing [Internet].* 2022;512:15-24. Available from: <https://doi.org/10.1016/j.neucom.2022.09.026>
18. Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, et al. Recent advances in convolutional neural networks. *Pattern Recognit [Internet].* 2018;77:354-77. Available from: <https://doi.org/10.1016/j.patcog.2017.10.013>
19. Ghimire S, Deo RC, Casillas-Pérez D, Salcedo-Sanz S, Sharma E, Ali M. Deep learning CNN-LSTM-MLP hybrid fusion model for feature optimizations and daily solar radiation prediction. *Meas J Int Meas Confed [Internet].* 2022;202(August):111759. Available from: <https://doi.org/10.1016/j.measurement.2022.111759>
20. Livieris IE, Stavroyiannis S, Pintelas E, Pintelas P. A novel validation framework to enhance deep learning models in time-series forecasting. *Neural Comput Appl.* 2020;32(23):17149-67. <https://doi.org/10.1007/s00521-020-05169-y>