

ANÁLISIS COMPARATIVO DE REDES NEURONALES PROFUNDAS PARA LA PREDICCIÓN DEL PRECIO DE MERCADO DE BITCOIN

AARÓN LÓPEZ-GARCÍA

alopez@edem.es

*Centro Universitario EDEM, Valencia, Spain
Plaça de L'aigua, s/n, 46024*

Recibido (24/05/2023)

Revisado (24/07/2023)

Aceptado (25/07/2023)

RESUMEN: Uno de los problemas más desafiantes de los últimos años es el estudio financiero de las criptomonedas con el fin de conocer su comportamiento futuro. El auge de la inteligencia artificial nos está ofreciendo herramientas que nos permiten abordar el asunto con mayor precisión. En particular, el aprendizaje profundo nos permite generar redes neuronales que aprenden de grandes volúmenes de datos, identificando así patrones imperceptibles para los humanos. En este trabajo, nos centramos en el estudio de Bitcoin, la criptomoneda de mayor valor y popularidad del mercado actualmente. A la hora de pronosticar su precio, se han implementado tres tipos de redes neuronales profundas y un modelo comparativo. Los resultados del análisis comparativo concluyen que los patrones temporales resultan ser más representativos que los espaciales al estimar el comportamiento del precio de cierre futuro de Bitcoin.

Palabras Clave: Bitcoin, Redes neuronales, Series temporales, Predicción financiera.

ABSTRACT: One of the most challenging problems of recent years is the financial study of cryptocurrencies in order to know their future behavior. The rise of artificial intelligence is providing us with tools that allow us to address the issue with greater precision. In particular, deep learning allows us to generate neural networks that learn from large volumes of data, thus identifying patterns imperceptible to humans. In this paper, we focus on the study of Bitcoin, the most valuable and popular cryptocurrency on the market today. Three types of deep neural networks and a comparative model have been implemented when forecasting its price. The results of the comparative analysis conclude that temporal patterns turn out to be more representative than spatial patterns when estimating the behavior of Bitcoin's future closing price.

Keywords: Bitcoin, Neural Networks, Time series, Financial forecasting.

1. Introducción

A principios del siglo XX, el estudio de series temporales en finanzas fue una tarea que despertó el interés de muchos investigadores (Tsay, 2000; Chatfield, 2005). Con el fin de modelar y predecir el comportamiento dinámico de los fenómenos, se empezaron a desarrollar métodos estadísticos que sirvieran para hacer inferencia. En sus inicios, la regresión o ajuste lineal fue el modelo matemático más utilizado para aproximar la relación de dependencia entre una o más variables independientes. De esta manera, una variable dependiente continua Y podía ser estimada a partir de otras variables regresoras X_i . A pesar de que estos modelos son muy conocidos y suelen usarse como punto de referencia, existen varios problemas con la extrapolación que supone la tarea de predicción (Chatfield, 2000).

En la segunda mitad del siglo XX, se desarrollaron herramientas más potentes para compensar las limitaciones que presentaba el modelo de regresión lineal. En parte, esto se debía a que la naturaleza dinámica de las finanzas parece seguir un comportamiento no lineal más incierto y complejo. A partir de las contribuciones que supuso la introducción de las medias móviles, surgieron diferentes técnicas con implementaciones matemáticas más avanzadas ya que estas consideran procesos estocásticos a la hora de hacer pronóstico sobre los datos (Yule, 1927). Dos modelos a destacar son los modelos ARMA (AutoRegressive Moving Average) y ARIMA (AutoRegressive Integrated Moving Average) (Box y Jenkins, 1970), que permitían modelar mejor la dinámica de las series temporales porque las futuras predicciones se calculan a partir de ventanas de n datos pasados. Además, estos modelos son capaces de controlar la tendencia y la estacionalidad gracias a su variante SARIMAX (Elamin y Fukushige, 2018). Debido a su gran poder predictivo, la familia de modelos ARIMA ha sido estudiada durante los últimos 50 años (Newbold, 1983; Suganthi y Samuel, 2012). Un aspecto muy destacado es su eficiencia y transparencia a la hora de aplicarlos en tareas financieras (Ho y Xie, 1998; Mondal *et al.*, 2014).

En paralelo a la familia de modelos ARIMA, se desarrollaron también los conocidos modelos ARCH (AutoRegressive Conditional Heteroskedasticity, Engle (1982)) y GARCH (Generalized AutoRegressive Conditional Heteroskedasticity, Bollerslev (1986)). La incorporación de estos modelos tenía como finalidad modelar el comportamiento no lineal que aparece de manera natural en el análisis de series temporales.

A finales del siglo XX, se cambió el enfoque con el que se planteaba la predicción de series temporales. La formalización de las técnicas de aprendizaje automático para hacer análisis de series temporales supuso un gran avance para este área (Ahmed *et al.*, 2010; Cerqueira *et al.*, 2019). En particular, este cambio de enfoque está justificado por dos puntos fundamentales. El primero es la gran cantidad de datos que podemos recopilar para poder hacer futuros pronósticos, tanto variables endógenas como exógenas. El segundo fue el avance computacional que presentaban los nuevos ordenadores para poder hacer computación sobre los datos. Por estas dos contribuciones, los modelos de aprendizaje automático resultaron ser más apropiados a la hora de describir las relaciones no lineales que se encuentran implícitas a la hora de estudiar fenómenos económicos o financieros (Bontempi *et al.*, 2013). De este modo, el estudio puede ser exhaustivo y las predicciones pueden ser significativas a mayor largo plazo (Pavlyshenko, 2019; Masini *et al.*, 2023).

Llegados al siglo XXI, el auge de la inteligencia artificial (IA) ha sido imparable. El uso de herramientas de IA en problemas del día a día ha sido cada vez más recurrente, obteniendo resultados considerablemente mejores de los que se lograban de forma analítica o empírica (Borges *et al.*, 2021). Independientemente del área al que hagamos referencia, la IA ha servido para asistir y mejorar los procesos que ya existían, convirtiéndose en una herramienta indispensable (Pannu, 2015). Dentro de la implementación de técnicas de inteligencia artificial, el uso de redes neuronales para resolver problemas relacionados con IA ha resultado ser el estado del arte de esta ciencia. Para el caso del análisis de series temporales, el aprendizaje profundo compuesto por redes

neuronales profundas es un referente a nivel predictivo (Sezer *et al.*, 2020). Su implementación en este campo es cada vez más extendida al igual que los modelos encargados de ajustarse al comportamiento de los mercados. Para más referencias, consulte (Dingli y Fournier, 2017; Lim y Zohren, 2021; Torres *et al.*, 2021).

En los últimos años, el análisis de mercados y de productos financieros ha evolucionado considerablemente. Una de las preocupaciones actuales es el estudio de los activos digitales no tangibles (Alzahrani y Daim, 2019; Wu *et al.*, 2021). En particular, el caso de las criptomonedas supone una descentralización total de ciertos mercados. A pesar de que surgen para ofrecer una mayor accesibilidad e inclusión financiera, la población mundial no tiene una perspectiva clara sobre cómo se generan las criptomonedas, generando así un escenario de incertidumbre al respecto (Kim *et al.*, 2016; Patil *et al.*, 2018).

En lo que respecta a criptomonedas, podemos afirmar que Bitcoin (BTC) es el referente del mercado (Chohan, 2022). Esta criptomoneda se creó en 2008 por una persona que actuaba bajo el pseudónimo de Satoshi Nakamoto. Sin embargo, se cree que esta moneda ha sido creada más bien por un grupo de personas. En enero de 2009, se comercializó la primera versión de BTC con la idea de servir principalmente para realizar humildes transacciones online. Pasados cuatro años, el precio de Bitcoin superó los 1.000 dólares estadounidenses, de manera que hubo un aumento de interés hacia esta criptomoneda. Desde entonces, su popularidad se ha disparado. En octubre de 2021, la unidad de Bitcoin alcanzó su máximo histórico, con precio de 66.000 dólares estadounidenses (Roa, 2021). A pesar de su gran fama a nivel mundial, tanto la volatilidad como las fluctuaciones que sufre Bitcoin afectan a la seguridad de este activo financiero. De todos modos, el interés que genera hace que desde entonces se haya convertido en objeto de estudio, tanto para inversores como para analistas de datos (Mudassir *et al.*, 2020). Por tanto, creemos que es necesario desarrollar herramientas inteligentes que estén a la altura del problema que supone predecir el comportamiento del precio de Bitcoin.

2. Revisión bibliográfica

El análisis de series temporales es un problema clásico dentro de las matemáticas financieras. En lo que respecta a la implementación de modelos estadísticos clásicos, podemos encontrar varios enfoques y estrategias. Para el caso de la regresión lineal, aparte de encontrar referencias sobre ajustar el comportamiento de un suceso en particular, también existen alternativas que estudian el pronóstico de tendencias en los datos (Gardner y McKenzie, 1985) o la incorporación de lógica difusa para el modelado de los datos de entrada al modelo (Zhang *et al.*, 2020). En cuanto a las implementaciones ARIMA, estas son normalmente consideradas como punto de referencia a la hora de comparar resultados (Mehrmolaei y Keyvanpour, 2016). A pesar de no ser tan efectivas como las redes neuronales artificiales, han demostrado ser un gran modelo para generar pronósticos (Zhang, 2003; Kihoro *et al.*, 2004; Siami-Namini *et al.*, 2018) llegando incluso a combinarse para generar modelos híbridos (Wang *et al.*, 2013). Entre muchas aplicaciones, podemos destacar su uso para el pronóstico de demanda (Falatouri *et al.*, 2022) o predicción de consumo y precio de la electricidad (Jakaša *et al.*, 2011). Otra técnica ampliamente conocida son los modelos de Holt-Winters (Chatfield, 1978), donde su robustez también ha sido comparada con los modelos ARIMA (Pongdatu y Putra, 2018). Por último, los modelos GARCH también han supuesto un gran interés para los investigadores. Su puesta en práctica involucra la predicción de precios al por menor (Aradhyula y Holt, 1988), la volatilidad de mercados (Franses y Van Dijk, 1996) o el modelado del tráfico en red (Anand *et al.*, 2008); así como la comparación con otras técnicas como ARIMA (Crawford y Fratantoni, 2003).

Dentro de las técnicas de aprendizaje automático supervisado, la implementación de sistemas inteligentes ha supuesto la mayor parte de la atención a la hora de pronosticar precios en el mer-

cado bursátil (Abraham *et al.*, 2001). Un factor determinante es el número de modelos diferentes que funcionan correctamente bajo condiciones muy específicas. Entre ellos, los modelos de aprendizaje estadístico más representativos son las Máquinas de vectores de soporte (“*Support vector machine*” SVM, Kim (2003)), Bosques aleatorios (“*Random Forest*” RF, Ho (1995)), K vecinos más próximos (KNN, Cover y Hart (1967); Altman (1992)), Naive-Bayes (NB, Lowd y Domingos (2005)), Perceptrón Multicapa (“*Multi-Layer Perceptron*” MLP, Pasero *et al.* (2010)). Este listado de modelos suele utilizarse para generar puntos de partida para mejorar por modelos más complejos (Krauss *et al.*, 2017; Turabieh, 2019; Dino y Abdulrazzaq, 2019).

El aprendizaje profundo mediante la composición de redes neuronales profundas es un enfoque innovador y efectivo para la predicción de series temporales. En general, podemos distinguir los modelos dependiendo del tipo de capa que las componen. Nosotros nos centraremos en las redes neuronales Prealimentadas (“*Feed-Forward Neural Network*” FFNN, Svozil *et al.* (1997)), Convolucionales (“*Convolutional Neural Networks*” CNN, Selvin *et al.* (2017)) y Recurrentes (“*Recurrent Neural Networks*” RNN, Zhang y Xiao (2000); Althelaya *et al.* (2018); Siami-Namini *et al.* (2019)). Un punto a favor de las redes neuronales es que han demostrado ser mejor modelo que los métodos estadísticos clásicos al procesar datos no estacionarios (Lachtermacher y Fuller, 1995).

Con respecto al análisis financiero del valor de Bitcoin, se han hecho muchas investigaciones al respecto (Velankar *et al.*, 2018). Entre ellas, se pueden encontrar implementaciones de los modelos mencionados previamente con el fin de pronosticar su precio de cierre (Ji *et al.*, 2019). Hasta la fecha, se puede asegurar que es mucho más conveniente implementar redes neuronales profundas para estimar el precio de cierre diario de Bitcoin debido al gran poder predictivo que presentan este tipo de arquitecturas (McNally *et al.*, 2018; Mudassir *et al.*, 2020). Además de este tipo de enfoques, esta criptomoneda también ha sido objeto de estudio con respecto al tipo de minado empleando cadena de bloques (Ahram *et al.*, 2017), la huella de carbono que supone la generación de Bitcoins (O’Dwyer y Malone, 2014) y la seguridad detrás de las transacciones digitales que lo envuelven (Rahouti *et al.*, 2018).

Un factor a tener en cuenta es el impacto que ha generado el estudio predictivo dentro del mundo de la investigación. Hoy en día existen competiciones de predicción de series temporales en diferentes áreas, donde se evalúa el pronóstico de cierto modelo sobre varios sucesos dinámicos. Una de las competiciones más destacadas son las Competiciones-M (Hyndman, 2020; Önköl, 2020), que cuentan ya con cuatro ediciones. Gracias a este tipo de eventos, el análisis de series temporales no para de evolucionar, dando así con enfoques teóricos cada vez más consistentes.

3. Metodología

3.1. *Regresor Prophet*

Dado que la información sobre el mercado de valores se ha seguido día a día, es interesante ver si la predicción de series temporales puede abordarse mediante modelos aditivos de regresión no lineal. Para ese tipo de tareas, puede aplicarse el modelo Prophet (Taylor y Letham, 2018), un software desarrollado por el grupo Meta capaz de producir predicciones a escala. En parte, hemos seleccionado Prophet como modelo de referencia porque se ha demostrado que este algoritmo es robusto al aplicarse sobre conjuntos de datos con valores faltantes y efectos estacionarios. Se trata de un modelo aditivo basado en tres componentes: tendencia (g), estacionalidad (s) y el efecto festivo que ocurre de forma irregular (h). Se combinan en la siguiente ecuación:

$$\hat{y}_t = g(t) + s(t) + h(t) + \varepsilon_t, \quad (1)$$

de manera que ε_t es un error con distribución normal. La motivación de este modelo era mejorar los modelos estadísticos clásicos de regresión dándoles una mayor flexibilidad a partir de las tres

componentes mencionadas previamente. Un muestra clara del avance que supone Prophet es que no necesita considerar ni ventanas fijas ni interpolar valores faltantes para hacer predicciones.

Tribe *et al.* (2021) presentaron NeuralProphet, el sucesor de Prophet para hacer análisis de series temporales siguiendo los estándares de explicabilidad, interpretabilidad y transparencia. El modelo se basa en una solución híbrida entre la metodología estadística clásica y las redes neuronales escalables. Las componentes básicas de NeuralProphet para realizar predicciones son tendencia (T), estacionalidad (S), efecto festivo (E), regresión del impacto de variables exógenas futuras (F), autoregresión de eventos pasados (A), regresión del impacto de variables exógenas futuras con retraso (L). Esto se representa de la siguiente forma:

$$\hat{y}_t = T(t) + S(t) + E(t) + F(t) + A(t) + L(T), \quad (2)$$

En la práctica, tanto Prophet como NeuralProphet han demostrado ser modelos contundentes para el análisis de series temporales financieras. Para Prophet podemos consultar en Almazroue *et al.* (2020); Aditya Satrio *et al.* (2021); Kumar Jha y Pande (2021), para NeuralProphet en Shohan *et al.* (2022); Arias Velásquez (2022) y una combinación de ambos en ChikkaKrishna *et al.* (2022); Long *et al.* (2023). Además, su uso también está vinculado con la predicción del precio de cierre de Bitcoin Yenidoğan *et al.* (2018).

3.2. Redes neuronales profundas

3.2.1. Red neuronal prealimentada - Feedforward neural network

En 1957, Rosenblatt desarrolló el concepto de perceptrón (Rosenblatt, 1957, 1958, 1963). Su implementación no tardó en hacerse ampliamente popular debido al gran avance que suponía para tareas de reconocimiento de patrones. Este tipo de neurona artificial se diseñó para calcular una clasificación binaria utilizando funciones de activación sobre una combinación lineal de características. La idea principal de la función es dividir la entrada en regiones y determinar cuáles de ellas son representativas a la hora de tomar una decisión. La formulación matemática de una función perceptrón es la siguiente:

$$\rho(x) = \begin{cases} 1 & \text{si } \phi(w^\top \cdot x + b) \geq 0, \\ 0 & \text{cualquier otro caso,} \end{cases} \quad (3)$$

de manera que ϕ es la función de activación que transforma la combinación lineal de los parámetros y los datos de entrada.

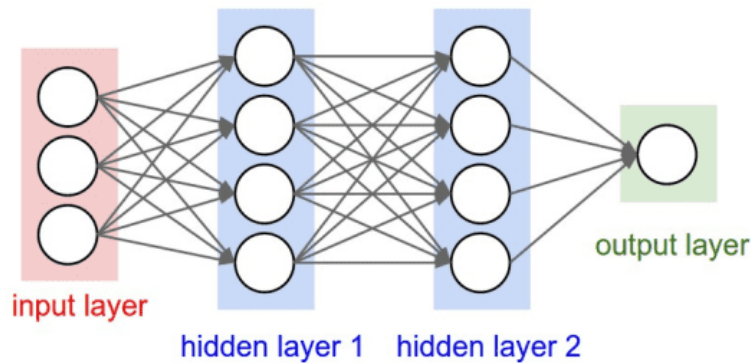


Figura 1. Arquitectura de una red neuronal prealimentada con dos capas de profundidad (Fuente).

Un red neuronal prealimentada (FFNN, Schmidhuber (2015)) es una estructura artificial formada por perceptrones que conectan todos los nodos de la capa anterior para alimentar a la siguiente capa. El concepto de FFNN es posiblemente uno de las arquitecturas más sencillas dentro del aprendizaje profundo. Para el caso en que la red neuronal esté compuesta por una única capa, se denomina perceptrón unicapa. Para cualquier otro caso, este tipo de redes pueden componerse de l capas (“*Fully connected layer*” FCL) donde cada una la componen $\{k_i\}_{k=1}^l$ neuronas. La arquitectura de una FFNN tiene la forma que vemos en la Figura 1.

3.2.2. Red neuronal convolucional - *Convolutional neural network*

Una red neuronal convolucional (CNN) es un tipo de red neuronal que implementa operaciones convolucionales sobre los datos de entrada mediante un producto interno matricial (LeCun y Bengio, 1998). El objeto más importante es el núcleo de la operación ya que define las dimensiones (altura, anchura y profundidad) que se transmiten mediante un mapa de características seguido de un número de pasos que determina la dimensionalidad espacial a operar. Dichos núcleos contienen los pesos, por lo que son los encargados de aprender de los datos (O’Shea y Nash, 2015; Albawi *et al.*, 2017). Como resultado, en cada fase del proceso convolucional del entrenamiento de modelos de IA se procesa la entrada mediante transformaciones locales. De esta manera, cada modelo puede aprender patrones espaciales que caracterizan el conjunto de datos de entrada. En lo que respecta al aprendizaje profundo, las CNN se consideran uno de los tipos de RNA más utilizados (Li *et al.*, 2021).

La agregación del operador de convolución, el núcleo, la activación, la normalización y la agrupación se denomina bloque convolucional. Para el caso de dos dimensiones $n \times m$, se representa del siguiente modo:

$$h_{ij}^{l+1} = \sum_{a=0}^{n-1} \sum_{b=0}^{m-1} w_{ab}^l h_{i+a, j+b}^l. \quad (4)$$

Acto seguido, se aplica una activación ϕ y una normalización ψ . Normalmente, estas funciones son la ReLU y la llamada normalización estadística o por lote.

Un ejemplo de implementación de un modelo CNN se representa en la Figura 2, donde una tarea de clasificación de imágenes motiva el uso de una red de este tipo.

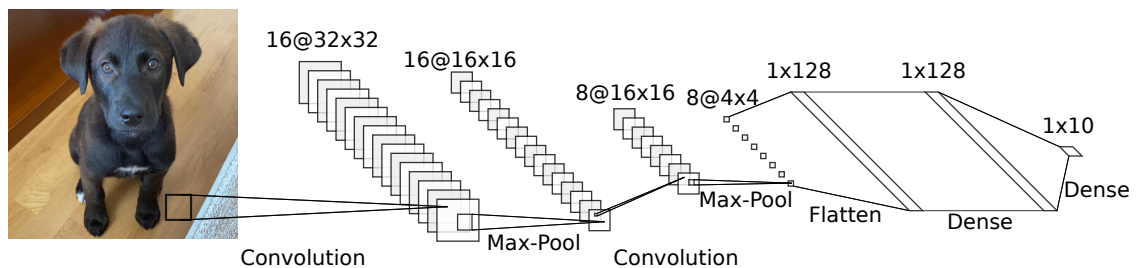


Figura 2. Topología de un modelo CNN diseñado para clasificar una imagen entre 10 animales diferentes.

3.2.3. Red neuronal recurrente - *Recurrent neural network*

Como las redes neuronales tradicionales no eran capaces de transmitir el factor temporal que presentan las series temporales, se diseñaron las redes neuronales recurrentes (RNN, Elman (1990)).

Su estructura interna conecta los datos previamente procesados con la siguiente entrada, como se muestra en la Figura 3, de modo que pueden aprender las características temporales subyacentes y aplicarlas sucesivamente. Esto le permite transmitir los patrones dinámicos de los conjuntos de datos, así como conectarlos recursivamente con una retroalimentación automática. Esta dirección de avance descrita se encarga de analizar los datos de entrada en el dominio serie-tiempo.

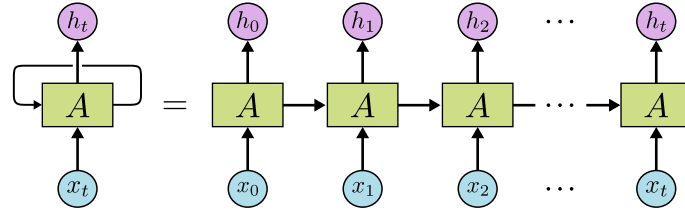


Figura 3. Arquitectura de las capas de una red neuronal recurrente y su serialización sobre t pasos.

Un caso particular de RNNs son las capas de memoria de largo y corto plazo (long short-term memory, LSTM) introducidas por Hochreiter y Schmidhuber (1997). Son un modelo basado en gradientes que se desarrolló para incorporar la información temporal a largo plazo a través de varias celdas que se unen para formar el bloque. La fusión de las propiedades a corto y largo plazo de una serie proporciona a las capas LSTM más ventajas a la hora de aprender de los datos en comparación con las capas RNN simples. Así, son capaces de conectar adecuadamente desfases temporales mayores con un error menor (Schuster y Paliwal, 1997). La formulación de cada una de sus componentes es la siguiente:

Puerta de olvido	$f_t = \sigma(W_f h_{t-1} + V_f x_t + b_f)$
Puerta de entrada	$i_t = \sigma(W_i h_{t-1} + V_i x_t + b_i)$
Memoria actualizada	$\tilde{C}_t = \tanh(W_C h_{t-1} + V_C x_t + b_C)$
Memoria interna	$C_t = f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t$
Puerta de salida	$o_t = \sigma(W_o h_{t-1} + V_o x_t + b_o)$
Puerta oculta	$h_t = o_t \otimes \tanh(C_t)$

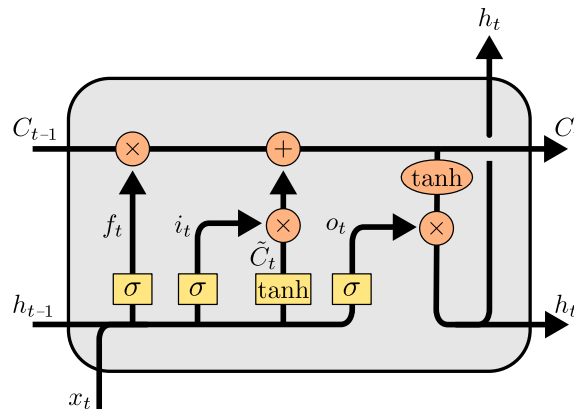


Figura 4. Arquitectura de una capa LSTM con su flujo de trabajo y sus distintas celdas y puertas.

4. Caso de estudio: Predicción del precio de cierre de Bitcoin

La motivación de este artículo es presentar un análisis comparativo de las redes neuronales descritas en la anterior sección para la particular tarea de pronosticar el precio de cierre de Bitcoin. Para ello, hemos realizado un estudio del mercado de las criptomonedas utilizando Python y la API de Yahoo! Finance. Al centrarnos en Bitcoin, hemos extraído los datos financieros en dólares y los hemos preparado para que esta información pueda ser procesada por nuestros modelos. A partir de estos datos, hemos generado un punto predictivo empleando NeuralProphet y la implementación de las redes neuronales con TensorFlow. Por último, después de haber entrenado las redes neuronales, hemos evaluado cada uno de los modelos para conocer su poder predictivo. El esquema de esta implementación está ilustrado en la Figura 5.

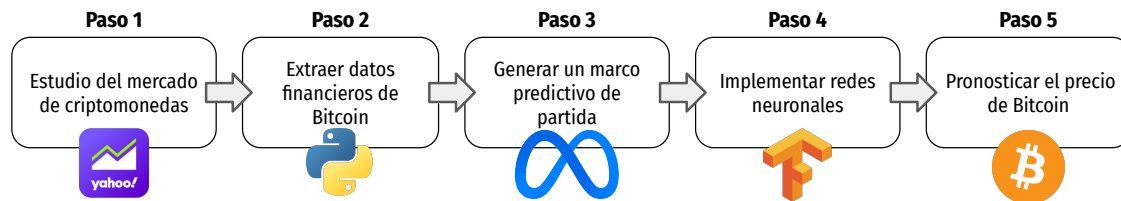


Figura 5. Procedimiento aplicado para realizar el estudio predictivo de Bitcoin.

4.1. *Procesamiento de los datos*

A partir de la API de Yahoo! Finance, hemos extraído información de carácter financiero de Bitcoin desde 1 de enero de 2017 hasta el 1 de mayo de 2023. Con estos datos, hemos seleccionado el precio de cierre de esta criptomoneda para realizar una estudio de series temporales considerando un único regresor. Estos datos se presentan en la Figura 6.

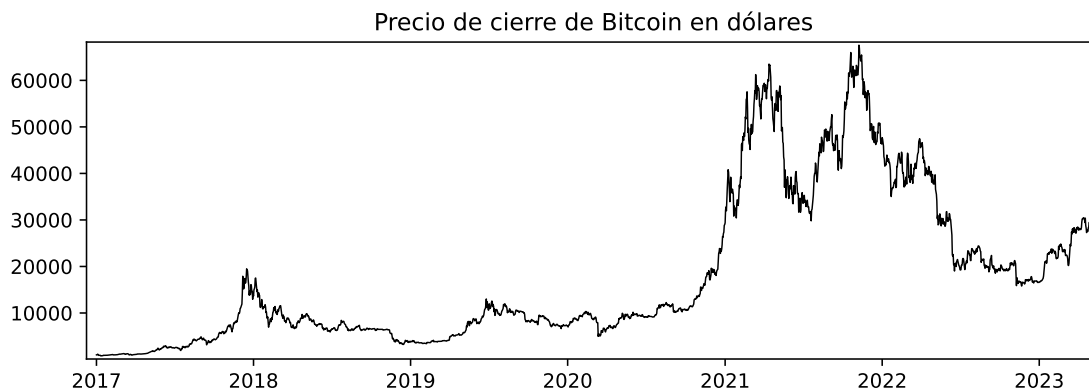


Figura 6. Precio de cierre de Bitcoin desde el 1/1/2017 hasta el 1/5/2023.

Para hacer el procesamiento de los datos, hemos decidido aplicar la normalización “min-max” para quitar las unidades (dólares) y cambiar el rango de los valores de $[777.75, 67566.82]$ a $[0, 1]$. Después, hemos serializado esta información en secuencias de 30 elementos, es decir, predecimos teniendo en cuenta los datos desde el día anterior al anterior mes. Finalmente, hemos dividido

estas secuencias de la siguiente manera: 75% de las series se destinan al entrenamiento, 15% a la validación y el 10% restantes al test final de evaluación.

4.2. Implementación de las redes neuronales

Todo el proceso de implementación de redes neuronales profundas se ha realizado utilizando la librería de TensorFlow. Desde la definición de los modelos hasta el proceso de aprendizaje se ha usado la API de esta librería. A la hora de realizar el proceso de ajuste de las redes neuronales, hemos seleccionado el error cuadrático medio (MSE) como función de pérdida y Adam como algoritmo de optimización (Kingma y Ba, 2017).

La motivación de este trabajo es realizar un marco comparativo para tres distintos tipos de modelos de aprendizaje profundo. Por tanto, se ha decidido que el diseño de las respectivas arquitecturas deben tener un número similar de parámetros de aprendizaje. Esto está justificado porque este número de parámetros es el que determina el poder predictivo del modelo y, a su vez, lo propenso que es el modelo a generar sobreajustes. Para nuestras tres redes neuronales, sus respectivas arquitecturas son las siguientes:

Tabla 1. Arquitectura de las redes neuronales implementadas para el análisis de series temporales de Bitcoin, donde ω indica el número de parámetros entrenables.

FFNN			CNN			LSTM		
Capa	Salida	ω	Capa	Salida	ω	Capa	Salida	ω
FCL	(1, 32)	992	Conv1D	(1, 28, 30)	120	LSTM	(1, 30, 20)	1760
FCL	(1, 32)	1056	Max-Pool	(1, 14, 30)	0	LSTM	(1, 4)	400
FCL	(1, 1)	33	Flatten	(1, 420)	0	FCL	(1, 1)	5
			FCL	(1, 1)	2105			
			FCL	(1, 1)	6			
		2081			2231			2165

4.3. Evaluación de los modelos

Con el fin de poder evaluar tanto el poder predictivo como el ajuste de nuestros modelos, hemos seleccionado seis métricas que nos servirán para evaluar las predicciones obtenidas. Por una parte, hemos elegido tres funciones que nos indican el error del modelo: MAE (error absoluto medio), RMSE (raíz del error cuadrático medio) y MedAE (error absoluto medio). Por otra parte, hemos seleccionado otras tres que nos indican el acierto de nuestras predicciones: MAPE (error absoluto medio en porcentaje), SMAPE (error medio absoluto en porcentaje simétrico) y DA (precisión direccional). La formulación matemática de estas medidas de error es la siguiente:

$$\begin{aligned}
 \text{MAE} & \quad \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| & \text{MAPE} & \quad \frac{100}{N} \sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{|y_i|} \\
 \text{RMSE} & \quad \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} & \text{SMAPE} & \quad \frac{100}{N} \sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i|} \\
 \text{MedAE} & \quad \text{median}\{|y_i - \hat{y}_i|\}_{i=1}^N & \text{DA} & \quad \frac{100}{N-1} \sum_{i=1}^{N-1} \sigma(y_i, \hat{y}_i)
 \end{aligned}$$

de manera que la función σ se define como:

$$\sigma(y_i, \hat{y}_i) = \begin{cases} 1 & \text{si } \text{signo}(y_{i+1} - y_i) = \text{signo}(\hat{y}_{i+1} - \hat{y}_i), \\ 0 & \text{cualquier otro caso.} \end{cases} \quad (5)$$

La selección de la función de pérdida DA nos sirve para conocer la precisión del modelo con respecto a las subidas y bajadas del precio de cierre.

5. Resultados

En esta sección presentamos los resultados obtenidos para la tarea de predecir el precio de cierre de Bitcoin medido en dólares estadounidenses. Como la motivación de este trabajo es realizar un análisis comparativo, la evaluación de los pronósticos obtenidos se ha dividido en dos etapas. Primero, generar un punto de partida utilizando el modelo NeuralProphet. Segundo, entrenar las redes neuronales descritas anteriormente para analizar sus pronósticos entre sí mismas y con NeuralProphet.

Para generar un marco predictivo de partida, hemos entrenado el modelo NeuralProphet con los ajustes por defecto de la librería de Python, excepto que el número de elementos que generan la salida es de 30. Además, hemos implementado la versión que devuelve el intervalo de incertidumbre atribuido a la predicción del modelo para comprobar las desviaciones que se obtienen al pronosticar la serie temporal de BTC. Los resultados para el conjunto test están en la última fila de la Tabla 2. A pesar de que el error medio absoluto (MAE) es bastante elevado, por encima de 600 dólares, vemos que el modelo sí que es preciso porque el error en porcentaje es del 3.12% y el acierto de subida-bajada del precio está por encima del 90%.

Para realizar el análisis comparativo de las tres redes neuronales implementadas en este trabajo, hemos entrenado los tres modelos empleando las mismas condiciones. La evaluación de las redes profundas FFNN, CNN y LSTM están incluidas en las tres primeras filas de la Tabla 2. A partir de estos resultados podemos afirmar que el modelo LSTM presenta un mejor ajuste a la hora de predecir el precio de cierre de la criptomoneda BTC, alcanzando un error absoluto por debajo de los 400 dólares, un error en porcentaje simétrico por debajo del 1% y un acierto direccional mayor al 97%. Para los otros dos modelos, juzgando por la propia evaluación en el conjunto de test, vemos que la red neuronal convolucional (CNN) tiene un poder predictivo mayor pronosticando el precio de Bitcoin que la red neuronal prealimentada (FFNN). Un factor a destacar es que todos los modelos han superado el marco de partida establecido por NeuralProphet. La explicación más viable para justificar estos resultados es que es más conveniente realizar un análisis de esta serie temporal considerando patrones temporales de largo y corto plazo, en lugar de patrones espaciales o combinaciones entre las propias series.

Tabla 2. Evaluación de las redes neuronales para la tarea de predicción del precio de cierre de Bitcoin en dólares.

	MAE	RMSE	MedAE	MAPE	SMAPE	DA
FFNN	519.01	778.10	347.84	2.4007	1.2008	94.37
CNN	476.81	721.57	297.49	2.2241	1.1087	94.80
LSTM	397.44	604.27	239.59	1.8418	0.9192	97.40
NeuralProphet	662.01	905.40	486.10	3.1237	1.5600	91.77

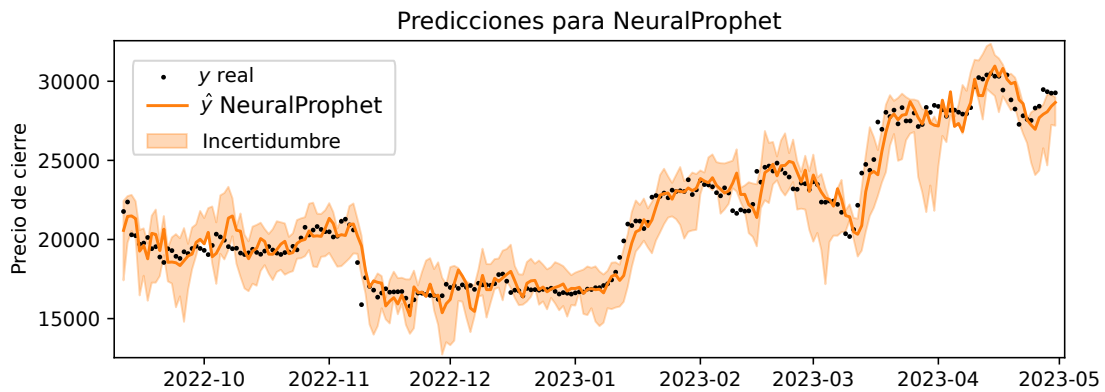


Figura 7. Precio de cierre de Bitcoin para el modelo NeuralProphet desde el 1/1/2017 hasta el 1/5/2023.

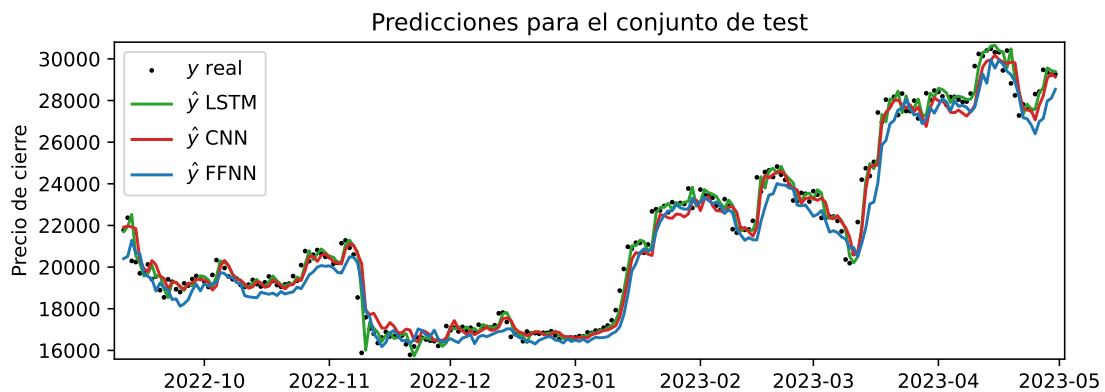


Figura 8. Precio de cierre de Bitcoin para nuestros modelos de aprendizaje profundo para predicción de series temporales desde el 1/1/2017 hasta el 1/5/2023.

6. Conclusiones y trabajos futuros

Al realizar un análisis comparativo de redes neuronales profundas, hemos visto que el modelo LSTM ha demostrado ser más efectivo para la tarea de pronosticar precios de cierre futuros de Bitcoin. En parte, esto está justificado porque los efectos temporales (corto y largo plazo) parecen ser determinantes a la hora de estimar el comportamiento del precio de la criptomoneda BTC. A su vez, hemos encontrado evidencias de que hay características espaciales (subidas-bajadas de precio, tendencias, estacionalidad, etc) que resultan significativas para pronosticar el precio de estas criptomonedas ya que el modelo CNN también ha devuelto buenos resultados. Por último, tanto el modelo FFNN como NeuralProphet parece que no son capaces de determinar ni extraer patrones que sirvan para aproximar el precio de mercado de BTC. Lo cual nos indica que la estrategia de predicción debe basarse en patrones que afectan directamente a los componentes de las series temporales de Bitcoin.

Cabe destacar que la implementación de los modelos de aprendizaje profundo se ha realizado con la idea de obtener modelos con la misma complejidad computacional, es decir, tener un número parecido de parámetros de aprendizaje. No obstante, para un estudio futuro podría considerarse el uso de modelos de memorias de largo y corto plazo (LSTMs) junto con una estrategia de

optimización de hiperparámetros (Nakisa *et al.*, 2018; Gorgolis *et al.*, 2019).

Otra línea de investigación futura que podría surgir a partir de este trabajo, es realizar marcos comparativos de redes neuronales para analizar series temporales en otros escenarios o problemas financieros. Un problema que surge a partir de este trabajo, sería estudiar el pronóstico de este tipo activos empleando múltiples series temporales. La finalidad de este estudio podría tener dos objetivos. Por una parte, ver si la cantidad de información con la que se alimenta una red neural está directamente relacionada con su poder predictivo. Por otra parte, comprobar si existen elementos externos (variables exóneas) que sean significativos a la hora de entrenar las redes neuronales.

Referencias bibliográficas

- Abraham, A., Nath, B., y Mahanti, P. K. (2001). Hybrid intelligent systems for stock market analysis. En *Proceedings of the International Conference on Computational Science-Part II, ICCS '01*, p. 337–345, Berlin, Heidelberg. Springer-Verlag.
- Aditya Satrio, C. B., Darmawan, W., Nadia, B. U., y Hanafiah, N. (2021). Time series analysis and forecasting of coronavirus disease in indonesia using arima model and prophet. *Procedia Computer Science*, 179:524–532. 5th International Conference on Computer Science and Computational Intelligence 2020.
- Ahmed, N. K., Atiya, A. F., Gayar, N. E., y El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29(5-6):594–621.
- Ahram, T., Sargolzaei, A., Sargolzaei, S., Daniels, J., y Amaba, B. (2017). Blockchain technology innovations. En *2017 IEEE Technology & Engineering Management Conference (TEMSCON)*, pp. 137–141.
- Albawi, S., Mohammed, T. A., y Al-Zawi, S. (2017). Understanding of a convolutional neural network. En *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6.
- Almazrouee, A. I., Almeshal, A. M., Almutairi, A. S., Alenezi, M. R., y Alhajerj, S. N. (2020). Long-term forecasting of electrical loads in kuwait using prophet and holt-winters models. *Applied Sciences*, 10(16).
- Althelaya, K. A., El-Alfy, E.-S. M., y Mohammed, S. (2018). Evaluation of bidirectional lstm for short-and long-term stock market prediction. En *2018 9th International Conference on Information and Communication Systems (ICICS)*, pp. 151–156.
- Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185.
- Alzahrani, S. y Daim, T. U. (2019). Analysis of the cryptocurrency adoption decision: Literature review. En *2019 Portland International Conference on Management of Engineering and Technology (PICMET)*, pp. 1–11.
- Anand, N. C., Scoglio, C., y Natarajan, B. (2008). Garch — non-linear time series model for traffic modeling and prediction. En *NOMS 2008 - 2008 IEEE Network Operations and Management Symposium*, pp. 694–697.
- Aradhyula, S. V. y Holt, M. T. (1988). Garch time-series models: An application to retail livestock prices. *Western Journal of Agricultural Economics*, 13(2):365–374.
- Arias Velásquez, R. M. (2022). A case study of neuralprophet and nonlinear evaluation for high accuracy prediction in short-term forecasting in pv solar plant. *Heliyon*, 8(9):e10639.
- Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327.
- Bontempi, G., Ben Taieb, S., y Le Borgne, Y.-A. (2013). *Machine Learning Strategies for Time Series Forecasting*, pp. 62–77. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Borges, A. F., Laurindo, F. J., Spínola, M. M., Gonçalves, R. F., y Mattos, C. A. (2021). The strategic use of artificial intelligence in the digital era: Systematic literature review and future

- research directions. *International Journal of Information Management*, 57:102225.
- Box, G. y Jenkins, G. (1970). *Time Series Analysis: Forecasting and Control*. Holden-Day series in time series analysis and digital processing. Holden-Day.
- Cerqueira, V., Torgo, L., y Soares, C. (2019). Machine learning vs statistical methods for time series forecasting: Size matters.
- Chatfield, C. (1978). The holt-winters forecasting procedure. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 27(3):264–279.
- Chatfield, C. (2000). *Time-series forecasting*. CRC press.
- Chatfield, C. (2005). Time-series forecasting. *Significance*, 2(3):131–133.
- ChikkaKrishna, N. K., Rachakonda, P., y Tallam, T. (2022). Short-term traffic prediction using fb-prophet and neural-prophet. En *2022 IEEE Delhi Section Conference (DELCON)*, pp. 1–4.
- Chohan, U. W. (2022). A history of bitcoin. Available at SSRN 3047875.
- Cover, T. y Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27.
- Crawford, G. W. y Fratantoni, M. C. (2003). Assessing the forecasting performance of regime-switching, arima and garch models of house prices. *Real Estate Economics*, 31(2):223–243.
- Dingli, A. y Fournier, K. S. (2017). Financial time series forecasting-a deep learning approach. *International Journal of Machine Learning and Computing*, 7(5):118–122.
- Dino, H. I. y Abdulrazzaq, M. B. (2019). Facial expression classification based on svm, knn and mlp classifiers. En *2019 International Conference on Advanced Science and Engineering (ICOASE)*, pp. 70–75.
- Elamin, N. y Fukushige, M. (2018). Modeling and forecasting hourly electricity demand by sarimax with interactions. *Energy*, 165:257–268.
- Elman, J. L. (1990). *Finding Structure in Time*, volumen 14, pp. 179–200. .
- Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50:987–1007.
- Falatouri, T., Darbanian, F., Brandtner, P., y Udokwu, C. (2022). Predictive analytics for demand forecasting – a comparison of sarima and lstm in retail scm. *Procedia Computer Science*, 200:993–1003. 3rd International Conference on Industry 4.0 and Smart Manufacturing.
- Franses, P. H. y Van Dijk, D. (1996). Forecasting stock market volatility using (non-linear) garch models. *Journal of Forecasting*, 15(3):229–235.
- Gardner, E. S. y Mckenzie, E. (1985). Forecasting trends in time series. *Management Science*, 31(10):1237–1246.
- Gorgolis, N., Hatzilygeroudis, I., Istenes, Z., y Gyenne, L.-G. (2019). Hyperparameter optimization of lstm network models through genetic algorithm. En *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pp. 1–4.
- Ho, S. y Xie, M. (1998). The use of arima models for reliability forecasting and analysis. *Computers & Industrial Engineering*, 35(1):213–216.
- Ho, T. K. (1995). Random decision forests. En *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volumen 1, pp. 278–282 vol.1.
- Hochreiter, S. y Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.
- Hyndman, R. J. (2020). A brief history of forecasting competitions. *International Journal of Forecasting*, 36(1):7–14. M4 Competition.
- Jakaša, T., Andročec, I., y Sprčić, P. (2011). Electricity price forecasting — arima model approach. En *2011 8th International Conference on the European Energy Market (EEM)*, pp. 222–225.
- Ji, S., Kim, J., e Im, H. (2019). A comparative study of bitcoin price prediction using deep learning. *Mathematics*, 7(10).

- Kihoro, J., Otieno, R. O., y Wafula, C. (2004). Seasonal time series forecasting: A comparative study of arima and ann models. .
- Kim, K. J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55:307–319.
- Kim, Y. B., Kim, J. G., Kim, W., Im, J. H., Kim, T. H., Kang, S. J., y Kim, C. H. (2016). Predicting fluctuations in cryptocurrency transactions based on user comments and replies. *PLOS ONE*, 11(8):1–17.
- Kingma, D. P. y Ba, J. (2017). Adam: A method for stochastic optimization.
- Krauss, C., Do, X. A., y Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the s&p 500. *European Journal of Operational Research*, 259:689–702.
- Kumar Jha, B. y Pande, S. (2021). Time series forecasting model for supermarket sales using fb-prophet. En *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 547–554.
- Lachtermacher, G. y Fuller, J. D. (1995). Back propagation in time-series forecasting. *Journal of Forecasting*, 14(4):381–393.
- LeCun, Y. y Bengio, Y. (1998). *Convolutional Networks for Images, Speech, and Time Series*, p. 255–258. MIT Press, Cambridge, MA, USA.
- Li, Z., Liu, F., Yang, W., Peng, S., y Zhou, J. (2021). A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21.
- Lim, B. y Zohren, S. (2021). Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2194):20200209.
- Long, B., Tan, F., y Newman, M. (2023). Forecasting the monkeypox outbreak using arima, prophet, neuralprophet, and lstm models in the united states. *Forecasting*, 5(1):127–137.
- Lowd, D. y Domingos, P. (2005). Naive bayes models for probability estimation. En *Proceedings of the 22nd International Conference on Machine Learning, ICML '05*, p. 529–536, New York, NY, USA. Association for Computing Machinery.
- Masini, R. P., Medeiros, M. C., y Mendes, E. F. (2023). Machine learning advances for time series forecasting. *Journal of Economic Surveys*, 37(1):76–111.
- McNally, S., Roche, J., y Caton, S. (2018). Predicting the price of bitcoin using machine learning. En *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pp. 339–343.
- Mehrmolaei, S. y Keyvanpour, M. R. (2016). Time series forecasting using improved arima. En *2016 Artificial Intelligence and Robotics (IRANOPEN)*, pp. 92–97.
- Mondal, P., Shit, L., y Goswami, S. (2014). Study of effectiveness of time series modeling (arima) in forecasting stock prices. *International Journal of Computer Science, Engineering and Applications*, 4(2):13.
- Mudassir, M., Bennbaia, S., Unal, D., y Hammoudeh, M. (2020). Time-series forecasting of bitcoin prices using high-dimensional features: a machine learning approach. *Neural Computing and Applications*.
- Nakisa, B., Rastgoo, M. N., Rakotonirainy, A., Maire, F., y Chandran, V. (2018). Long short term memory hyperparameter optimization for a neural network based emotion recognition framework. *IEEE Access*, 6:49325–49338.
- Newbold, P. (1983). Arima model building and the time series analysis approach to forecasting. *Journal of Forecasting*, 2(1):23–35.
- O'Dwyer, K. J. y Malone, D. (2014). Bitcoin mining and its energy footprint. En *25th IET*

- Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014)*. IET.
- O'Shea, K. y Nash, R. (2015). An introduction to convolutional neural networks.
- Pannu, A. (2015). Artificial intelligence and its application in different areas. *Artificial Intelligence*, 4(10):79–84.
- Pasero, E., Raimondo, G., y Ruffa, S. (2010). Mulp: A multi-layer perceptron application to long-term, out-of-sample time series prediction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6064 LNCS:566–575.
- Patil, A. P., Akarsh, T., y Parkavi, A. (2018). A study of opinion mining and data mining techniques to analyse the cryptocurrency market. En *2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS)*, pp. 198–203.
- Pavlyshenko, B. M. (2019). Machine-learning models for sales time series forecasting. *Data*, 4(1).
- Pongdatu, G. A. N. y Putra, Y. H. (2018). Seasonal time series forecasting using sarima and holt winter's exponential smoothing. *IOP Conference Series: Materials Science and Engineering*, 407(1):012153.
- Rahouti, M., Xiong, K., y Ghani, N. (2018). Bitcoin concepts, threats, and machine-learning security solutions. *IEEE Access*, 6:67189–67205.
- Roa, M. M. (2021). Bitcoin alcanza un nuevo máximo histórico y supera los 66.000 dólares.
- Rosenblatt, F. (1957). *The Perceptron, a Perceiving and Recognizing Automaton*. Report 85-60-1. Cornell Aeronautical Laboratory, Buffalo, New York.
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408.
- Rosenblatt, F. (1963). Principles of neurodynamics. perceptrons and the theory of brain mechanisms. *American Journal of Psychology*, 76:705.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117.
- Schuster, M. y Paliwal, K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.
- Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., y Soman, K. P. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. En *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1643–1647.
- Sezer, O. B., Gudelek, M. U., y Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning : A systematic literature review: 2005–2019. *Applied Soft Computing*, 90:106181.
- Shohan, M. J. A., Faruque, M. O., y Foo, S. Y. (2022). Forecasting of electric load using a hybrid lstm-neural prophet model. *Energies*, 15(6).
- Siami-Namini, S., Tavakoli, N., y Namin, A. S. (2019). A comparison of arima and lstm in forecasting time series. *Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018*, pp. 1394–1401.
- Siami-Namini, S., Tavakoli, N., y Siami Namin, A. (2018). A comparison of arima and lstm in forecasting time series. En *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1394–1401.
- Suganthi, L. y Samuel, A. A. (2012). Energy models for demand forecasting—a review. *Renewable and Sustainable Energy Reviews*, 16(2):1223–1240.
- Svozil, D., Kvasnicka, V., y Pospichal, J. (1997). Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems*, 39(1):43–62.
- Taylor, S. J. y Letham, B. (2018). Forecasting at scale. *The American Statistician*, 72(1):37–45.

- Torres, J. F., Hadjout, D., Sebaa, A., Martínez-Álvarez, F., y Troncoso, A. (2021). Deep learning for time series forecasting: A survey. *Big Data*, 9(1):3–21. PMID: 33275484.
- Triebe, O., Hewamalage, H., Pilyugina, P., Laptev, N., Bergmeir, C., y Rajagopal, R. (2021). Neuralprophet: Explainable forecasting at scale.
- Tsay, R. S. (2000). Time series and forecasting: Brief history and future research. *Journal of the American Statistical Association*, 95(450):638–643.
- Turabieh, H. (2019). Hybrid machine learning classifiers to predict student performance. *2019 2nd International Conference on New Trends in Computing Sciences, ICTCS 2019 - Proceedings*.
- Velankar, S., Valecha, S., y Maji, S. (2018). Bitcoin price prediction using machine learning. En *2018 20th International Conference on Advanced Communication Technology (ICACT)*, pp. 144–147.
- Wang, L., Zou, H., Su, J., Li, L., y Chaudhry, S. (2013). An arima-ann hybrid model for time series forecasting. *Systems Research and Behavioral Science*, 30(3):244–259.
- Wu, J., Liu, J., Zhao, Y., y Zheng, Z. (2021). Analysis of cryptocurrency transactions from a network perspective: An overview. *Journal of Network and Computer Applications*, 190:103139.
- Yenidoğan, I., Çayır, A., Kozan, O., Dağ, T., y Arslan, C. (2018). Bitcoin forecasting using arima and prophet. En *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, pp. 621–624.
- Yule, G. U. (1927). On a method of investigating periodicities in disturbed series, with special reference to wolfer's sunspot numbers. *Philosophical Transactions of the Royal Society of London*, 226:267–298.
- Zhang, G. (2003). Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175.
- Zhang, J.-S. y Xiao, X.-C. (2000). Predicting chaotic time series using recurrent neural network. *Chinese Physics Letters*, 17:88–90.
- Zhang, Y., Qu, H., Wang, W., y Zhao, J. (2020). A novel fuzzy time series forecasting model based on multiple linear regression and time series clustering. *Mathematical Problems in Engineering*, 2020:9546792.
- Önkal, D. (2020). M4 competition: What's next? *International Journal of Forecasting*, 36(1):206–207. M4 Competition.

Agradecimientos

Este trabajo está financiado por el centro universitario EDEM - Escuela De Empresarios.