



Facultad de Ciencias Económicas y Empresariales

LA CAPACIDAD PREDICTIVA DE LAS REDES NEURONALES LSTM RESPECTO DEL BITCOIN

*Diseño y optimización de una red neuronal LSTM para predecir
las variaciones en el precio del Bitcoin en el corto plazo.*

Autor: Iñigo Sánchez Zurdo
Directora: María Coronado Vaca

Resumen:

La creciente atención dedicada al Machine Learning en el ámbito de las finanzas se ha proyectado, como no podía ser de otro modo, en el ámbito de las criptomonedas. Sin embargo, la literatura disponible hasta el momento es escasa respecto de dos aspectos: la utilización de datos de alta frecuencia y la predicción de un valor en lugar de la dirección de la variación del precio. El presente trabajo aborda ambas dimensiones mediante el diseño y optimización de un modelo LSTM que pretende predecir el precio de cotización del bitcoin a 5 minutos vista en base a la cotización, volumen y distintos indicadores técnicos durante los 60 minutos anteriores.

El modelo LSTM diseñado no muestra sin embargo una capacidad predictiva superior a la de los modelos de referencia contra los que se compara su rendimiento, a pesar del potencial de las redes LSTM que se proclama en la literatura. Algunos cambios propuestos en la metodología podrían mejorar los resultados, pero por el momento, no podemos sino concluir que las redes LSTM no presentan una capacidad predictiva significativa respecto de la cotización del Bitcoin.

Palabras clave: Bitcoin, Machine Learning, Redes Neuronales, LSTM

Abstract:

The growing interest on stock prediction amongst Machine Learning practitioners has expanded into the cryptocurrency market. However, literature as of yet has not turned its attention to some interesting aspects: the use of minute instead of hourly or daily data, and the prediction of stock returns rather than the sign of the movement. As a result, the aim of this paper is to tackle both of the previous issues by building and optimizing an LSTM neural network that aims to predict the price of Bitcoin 5 minutes into the future from the stock prices, volume and technical indicators of the previous 60 minutes.

The results of the LSTM model show, despite its potential for stock prediction as seen on the literature, a worse performance than the models used as reference. Some of the proposed changes on the used methodology could improve upon these results, but as of now, we can only conclude that LSTM models show no significant predictive ability of Bitcoin future stock prices.

Keywords: Bitcoin, Machine Learning, Neural Networks, LSTM

ÍNDICE

1. INTRODUCCIÓN	1
2. REVISIÓN DE LA LITERATURA RELATIVA A LA UTILIZACIÓN DE MACHINE LEARNING PARA LA PREDICCIÓN DE PRECIOS DE COTIZACIÓN	3
2.1. Aproximación a la relación entre Machine Learning y finanzas.	3
2.2. La predicción de series temporales en mercados bursátiles.	6
2.2.1. <i>Modelos econométricos</i>	6
2.2.2. <i>Modelos supervisados</i>	10
2.2.3. <i>Modelos no supervisados</i>	11
2.3. La utilización de redes neuronales para la inversión.	11
2.3.1. <i>Redes Neuronales Artificiales (ANN)</i>	12
2.3.2. <i>Redes Neuronales Convolucionales (CNN)</i>	13
2.3.3. <i>Redes Neuronales Recurrentes (RNN)</i>	14
2.3.4. <i>Comparativa de la validez obtenida en la literatura por las Redes Neuronales y otros modelos predictivos</i>	15
2.4. Análisis de la aplicación los modelos predictivos en el ámbito de las criptomonedas...	17
2.4.1. <i>Arquitecturas empleadas en la literatura para la predicción de la cotización de criptomonedas</i>	17
2.4.2. <i>Posibilidades de aplicación de las redes neuronales para la predicción de la cotización de las criptomonedas</i>	19
2.4.3. <i>Limitaciones de la literatura disponible en el ámbito de las criptomonedas</i>	21
3. LA PREDICCIÓN DEL PRECIO DE COTIZACIÓN DEL BITCOIN MEDIANTE UN MODELO LSTM COMO OBJETIVO	22
4. ANÁLISIS PRÁCTICO DE LA CAPACIDAD PREDICTIVA DE LAS REDES LSTM EN EL ÁMBITO DE LAS CRIPTOMONEDAS	24
4.1. Conformación del dataset inicial	24
4.1.1. <i>Recolección de datos</i>	24
4.1.2. <i>Tratamiento de observaciones ausentes en el dataset inicial</i>	25
4.2. Tratamiento del dataset	25
4.2.1. <i>Incorporación de variables adicionales</i>	25
4.2.2. <i>Transformación de las variables</i>	30
4.3. Selección de variables	33
4.4. Arquitectura del modelo	38
4.4.1. <i>División del dataset</i>	38
4.4.2. <i>Construcción de los bloques de datos</i>	39
4.4.3. <i>La validación cruzada para garantizar la robustez de los resultados obtenidos</i>	40

4.4.4.	<i>Arquitectura base de un modelo LSTM</i>	42
4.4.5.	<i>Optimización de los parámetros del modelo LSTM</i>	43
5.	INTERPRETACIÓN DE LOS RESULTADOS	53
5.1.	Descripción de los modelos de referencia	53
5.2.	Comparativa de resultados entre el modelo LSTM y los modelos de referencia	53
6.	CONCLUSIONES Y RECOMENDACIONES METODOLÓGICAS PARA FUTUROS TRABAJOS.....	57
7.	BIBLIOGRAFÍA.....	59
	ANEXO I: ANÁLISIS DE LA CORRELACIÓN DE LOS INPUTS PRELIMINARES RESPECTO DE LA VARIABLE TARGET	68

1. INTRODUCCIÓN

Las criptomonedas se han convertido en un corto periodo de tiempo en un fenómeno social que va más allá de su utilidad práctica, derivando en un nivel de volatilidad que excede, y en una proporción sustancial, el de los mercados bursátiles tradicionales.

Este interés por este nuevo activo se ha proyectado también en el ámbito del Machine Learning. Cada vez son más los practicantes que tratan de proyectar sus modelos en este mercado para predecir la cotización futura de criptomonedas como el Bitcoin. Sin embargo, los esfuerzos predictivos en este campo se han restringido casi en su totalidad a la clasificación de los movimientos del precio.

La literatura disponible sobre la predicción del precio de cotización de acciones considera los modelos de red neuronal Lon-Short Term Memory (LSTM) como una de las arquitecturas con mayor potencial para la predicción de los precios de cotización, si bien su potencial en el ámbito de las criptomonedas no ha sido estudiado con tanta profundidad. Por ello, en el presente trabajo se ha elaborado un modelo LSTM con el que se pretende predecir la cotización futura del Bitcoin en tanto que criptomoneda por excelencia.

Otro aspecto novedoso respecto de la literatura disponible, además del planteamiento del problema en tanto que regresión en lugar de clasificación, es la utilización de datos de alta frecuencia a través de la cotización minuto a minuto de distintas criptomonedas, y la predicción en el corto plazo, a 5 minutos vista, frente a los periodos predictivos diarios que proliferan en la literatura. Adicionalmente, se incorporan una pluralidad de los indicadores más populares dentro del análisis técnico para analizar su capacidad predictiva.

Para ello, se aplican en primera instancia de manera sucesiva distintas técnicas de selección de variables tanto univariantes como multivariantes para determinar las variables de mayor importancia. A continuación, se divide el dataset en datos de evaluación y entrenamiento, y estos últimos a su vez en sets de entrenamiento y de validación.

Esta división en entrenamiento y validación resulta necesaria para proceder con la optimización de parámetros sin contaminar el modelo alimentándolo con los datos de evaluación. Tras determinar los valores óptimos para cada parámetro a optimizar, se puede proceder finalmente con el proceso predictivo.

Finalmente, para determinar de manera precisa la capacidad predictiva del modelo LSTM diseñado, se compararán sus resultados con los de distintos modelos de referencia, así como con otras arquitecturas de redes neuronales recurrentes.

Como se expondrá de nuevo más adelante, no se espera descubrir un modelo infalible ni mucho menos. Si ello fuera posible, sería uno de los muchos autores analizados en la revisión del estado de la cuestión quien lo desarrollase. En su lugar, el objetivo del presente trabajo no es otro que abordar y tratar de arrojar luz sobre algunos de los aspectos predictivos respecto del Bitcoin sobre los que la literatura no se ha pronunciado aún.

2. REVISIÓN DE LA LITERATURA RELATIVA A LA UTILIZACIÓN DE MACHINE LEARNING PARA LA PREDICCIÓN DE PRECIOS DE COTIZACIÓN

2.1. Aproximación a la relación entre Machine Learning y finanzas.

Puede definirse el Machine Learning (o ML) como el método de análisis de datos que permite automatizar el desarrollo de modelos analíticos mediante algoritmos. De manera iterativa, el modelo aprende de los datos que se le suministran para encontrar relaciones o secuencias ocultas *a priori* (Deshmukh et al., 2019; Simonian et al., 2019).

El pilar sobre el que se erige el análisis técnico es la identificación y extrapolación de tendencias, por lo que resultan evidentes las ventajas que los modelos de aprendizaje pueden proporcionar en el ámbito de la especulación y la predicción de cotizaciones futuras, al permitir analizar cantidades cuasi infinitas de datos e identificar ya no solo tendencias las series temporales, sino incluso el momento en que estas se revierten (Vijh et al., 2020; Harvey, 2021).

Hace ya tiempo que la inversión sistemática o cuantitativa dejó de ser un método de inversión oscuro e incomprensible para el inversor promedio. En la actualidad encontramos no sólo expertos en esta tipología de inversión, comúnmente denominados *quants*, sino incluso fondos dedicados de manera exclusiva a la inversión algorítmica, con unos resultados que incluso superan a los de los más reconocidos y laureados inversores de valor o fundamentales. Afirma Harvey (2021, p.i) en este sentido que “*en el entorno actual todos somos quants aunque no todos empleemos portfolios sistemáticos*”.

Durante las décadas de 1960 y 1970, las a menudo enfrentadas corrientes de la especulación cuantitativa y la inversión en valor, se vieron desafiadas por la hipótesis de la eficiencia de los mercados (Mallikarjuna & Rao, 2019), según la cual, en su versión más estricta, las cotizaciones de las acciones reflejan en todo momento de manera plena la información privada y pública disponible hasta ese momento (Fama, 1970; Sharma & Kaushik, 2017; Shen & Shafiq, 2020).

Siendo esto así, la teoría del paseo aleatorio o *random walk*, establece que los precios de una determinada acción no pueden ser predichos en base a los datos históricos al no existir correlación alguna entre sendos periodos temporales (Fama, 1970). Frente a esta postura,

sin embargo, los defensores del análisis técnico o ‘chartistas’ consideran que la cotización histórica esconde determinada información que puede proporcionar indicios acerca de los movimientos futuros de la cotización, y que dicha información puede ser conocida en base a una pluralidad de indicadores (Usmani et al., 2016).

En relación con la teoría del paseo aleatorio, Lo & MacKinlay (1988) llevaron a cabo un interesante análisis de los retornos semanales del mercado de valores entre 1962 y 1985, rechazando la hipótesis nula defendida por la teoría del *random walk* mediante un test de especificación basado en la volatilidad, encontrando además autocorrelación positiva en los retornos con periodos de tenencia semanales.

Así, con el paso del tiempo la hipótesis de la total eficiencia de los mercados y la incapacidad de predecir la cotización de las acciones se ha ido disipando, y en este proceso ha contribuido de manera notable el desarrollo y perfeccionamiento de las técnicas de Machine Learning.

Lo cierto es que los esfuerzos por parte de investigadores tanto del ámbito económico-financiero como del informático con vistas a hacer posible dicha predicción han sido constantes. El avance tecnológico ha transformado completamente el mundo de la inversión en bolsa, abriendo la puerta a la posibilidad de predecir los movimientos futuros de las cotizaciones (Rouf et al., 2021).

El abaratamiento de la computación junto con la mayor facilidad para codificar y sistematizar sistemas de inversión ha dado lugar a un nuevo paradigma en la especulación o, empleando la terminología anglosajona, *trading*, y el análisis técnico.

Y es que los modelos de Machine Learning han probado mejorar la eficiencia de los modelos predictivos en más de un 60% respecto a las metodologías predictivas tradicionales basadas en la estadística como las medias exponenciales o la regresión lineal (Vijh et al., 2020).

Una de las principales ventajas de los modelos de Machine Learning en el ámbito de las finanzas, es que los modelos no cuentan con sesgos del comportamiento ni se verán influenciados por las emociones humanas (Harvey, 2021).

Sin embargo, y como afirman López de Prado y Bailey (2021), dos de las principales referencias a nivel mundial en lo que a la aplicación del Machine Learning a las finanzas se refiere, lo cierto es que los resultados obtenidos empleando estos métodos a menudo no resultan mejores que una estrategia puramente aleatoria cuando se emplean en condiciones reales de mercado, por mucho que los resultados sobre el papel puedan parecer óptimos.

Estos autores aluden, entre los principales problemas que surgen en la aplicación del Machine Learning a las finanzas, entre otros, al overfitting o la teoría de las falsas estrategias, que supone en esencia un incremento artificial de los resultados mostrados por una estrategia o modelo determinado al aplicarse de manera repetida sobre los datos pasados, desapareciendo ese rendimiento positivo al extrapolarse la estrategia a condiciones reales de mercado.

Harvey (2021) comparte esta postura, considerando que la tendencia actual al overfitting, incluso en la literatura, se fundamenta en que la ratio ruido-signal (*'signal-to-noise ratio'*) en los retornos de los activos financieros es tan reducida que los investigadores terminan optimizando el ruido, llevando a resultados pobres en la práctica.

También se muestra escéptico respecto de la capacidad de predecir precios de cotización o retornos futuros Chan (2021). Este autor considera que uno de los elementos que caracterizan los modelos predictivos válidos es la 'no reflexividad' de la variable objetivo, es decir, que la posibilidad de predecir de manera acertada una variable no altere el comportamiento de la misma, algo que no sucede en los mercados de valores ya que si los retornos pudieran predecirse con precisión, estos mismos se transformarían a medida que los inversores comiencen a actuar sobre sus propias predicciones, generando la obsolescencia de los modelos.

Para finalizar con los problemas asociados a los modelos de Machine Learning financieros, cuya consideración es la base para un modelo efectivo, cabe mencionar lo que los autores denominan como modelos de 'caja negra'. La capacidad predictiva y analítica de los modelos de aprendizaje pueden llevar a los investigadores a desarrollar modelos basados casi exclusivamente en los datos y no en fundamentos económicos (Harvey, 2021).

Jim Simons, probablemente el inversor cuantitativo más laureado de la historia junto con su Medallion Fund, consideraba que no era tan importante el conocer todos los niveles que componían la maquinaria del mercado como el encontrar un sistema matemático capaz de imitarlos lo suficiente como para generar retornos constantes (Zuckerman 2019).

Todo inversor que carezca de los conocimientos matemáticos de Jim Simons y su equipo y de la capacidad computacional de gigantes de la inversión sistemática como Renaissance Technologies, haría bien en aplicar los consejos de Harvey antes de adentrarse en la predicción del mercado de valores.

Sin embargo, esta dificultad adicional que encontramos en la predicción de carácter financiero no ha impedido un desarrollo notable de literatura al respecto.

2.2. La predicción de series temporales en mercados bursátiles.

El concepto de Machine Learning es sumamente amplio, englobando una pluralidad de arquitecturas que varían entre sí en la forma en la que analizan los datos proporcionados.

Adicionalmente, los modelos econométricos y matemáticos más tradicionales siguen siendo empleados por numerosos economistas e inversores, y es que su validez resulta a menudo similar o incluso superior a la de los modelos de Machine Learning.

La cotización de las acciones puede definirse como una serie temporal, es decir, una secuencia cronológica de observaciones para una variable, en este caso el precio. Para la predicción de series temporales, se emplean tanto modelos lineales como no lineales, entre los que se encuentran los modelos de ‘Deep Learning’ (Selvin et al., 2017).

2.2.1. Modelos econométricos

Antes de entrar a analizar los modelos de Machine Learning, cabe hacer alusión a los métodos ‘clásicos’ de predicción. Los avances en el análisis de datos no han impedido, más bien han realzado, el análisis y estudio de métodos predictivos econométricos y matemáticos al permitir su aplicación en series cada vez más amplias.

También a la inversa, se produce que la incorporación de conceptos estadísticos avanzados ha contribuido en gran medida a pulir diversos algoritmos de aprendizaje (Moghar &

Hamiche, 2020). Dada la demostrada validez de los modelos econométricos, sus características podrían ser incorporadas a los modelos de Machine Learning para mejorar su capacidad predictiva.

Apunta Sen (2021) que la volatilidad que caracteriza los mercados bursátiles impide la utilización de modelos estadísticos simples como la regresión lineal o incluso la polinómica, pero otros modelos econométricos sí que producen niveles de precisión aceptables en el ámbito de la predicción de cotizaciones.

Entre las modalidades más básicas de este tipo de modelos tradicionales, siendo a su vez la base para otros modelos más complejos, destacan la media móvil simple y exponencial, denominadas comúnmente SMA y EMA respectivamente.

La predicción basada en medias móviles supone en esencia, determinar el precio futuro en base a una media aritmética en el caso de SMA o exponencial primando los valores más recientes en el caso de EMA, de las últimas x observaciones (Aygün & Kabaçi, 2021).

Numerosos autores han tratado de predecir el precio futuro de un activo financiero aplicando uno o varios de los modelos lineales que engloban a toda la familia de técnicas autorregresivas y que parten en cierta medida de las medias móviles (Adewuyi, 2016; Selvin et al., 2017).

Mediante modelos autorregresivos integrados de media móvil (ARIMA por sus siglas en inglés), en los que la predicción futura resulta de la combinación lineal de los valores previos, se han obtenido resultados dispares al proyectarse en los mercados bursátiles, si bien su validez en la predicción de movimientos cortoplacistas encuentra un sustento notable en la literatura (Adebiyi et al., 2014).

Particularmente, en Mallikarjuna & Rao (2019), el modelo ARIMA mostró los mejores resultados de entre todos los 5 modelos evaluados, en 7 de los 24 mercados analizados, afirmando los autores (p.14) que *“las técnicas ARIMA y SETAR pueden considerarse las claras ganadoras en la predicción de los retornos del mercado bursátil en mercados desarrollados, emergentes y frontera”*.

Sin embargo, la implementación efectiva de los modelos ARIMA queda supeditada al cumplimiento de determinados requisitos entre los que destacan la estacionariedad de los

datos o la necesidad de aplicar técnicas de derivación y diferenciación a los datos (Aygün & Kabaçi, 2021).

El modelo SETAR (Self-Exciting Threshold Autoregressive) por su parte, resulta especialmente preciso cuando se aplica a series temporales en las que existen rupturas estructurales (*structural breaks*) de los datos al emplearse un umbral para capturar las tendencias no lineales (Naik & Mohan, 2021). De esta manera solventa una de las principales problemáticas de los modelos econométricos más simples como ya se ha expuesto.

Selvin et al. (2017) afirman que la problemática de estos modelos lineales reside en que al emplear ecuaciones matemáticas para ajustar un modelo a una serie temporal de carácter univariante, no se logra capturar las interrelaciones y dinámicas existentes en el mercado de valores, una idea similar a la antes expuesta de Sen (2021).

Entre los modelos no lineales, una de las arquitecturas más primitivas es el análisis regresivo, mediante el que se pueden establecer relaciones estadísticas cuantitativas entre distintas variables (Xing et al., 2013; Vijn et al., 2020).

Se trata de un método no lineal en el que a partir del análisis de las variables disponibles se determina una ecuación de regresión en base a la cual se realizan las predicciones correspondientes para el periodo proyectado (Sharma & Kaushik, 2017), si bien resulta complicado afirmar su utilidad en la predicción del mercado bursátil ya que es incapaz de capturar todas las sutilezas y matices que terminan influyendo en la cotización de las acciones (Xing et al., 2013).

También resultan abundantes los estudios que han aplicado Modelos de Markov, el proceso matemático sobre el que se sustentan las redes neuronales, para tratar de predecir movimientos futuros en las cotizaciones.

En tanto existen en el mercado de valores elementos que no pueden ser observados de manera directa autores como Kavitha et al. (2013) o más recientemente Liu et al. (2021) y Venugopal et al. (2021) han tratado de asimilar dichos factores a los estados ocultos en los Modelos Ocultos de Markov y emplear las probabilidades de transición a cada uno de los estados posibles para predecir cambios en el precio. Los trabajos citados obtuvieron

resultados satisfactorios tanto en la predicción del signo del movimiento como en la entidad del mismo.

Un proceso oculto de Markov es aquel en que la cadena de eventos está gobernada por parámetros ocultos subyacentes, de manera que si bien puede observarse el resultado de la cadena, los estados que han conducido hasta dicho estado no son accesibles (Zuckerman 2019).

Cabe destacar que ya en la década de 1960 Jim Simons se planteaba la posibilidad de predecir los cambios futuros en la cotización en base al comportamiento pasado de la acción. Para sobreponerse a ese problema de ‘caja negra’ que caracteriza a los procesos ocultos de Markov, Simons implementó el algoritmo Baum-Welch para poder inferir si quiera parcialmente las reglas y probabilidades contenidas en el modelo (Zuckerman 2019).

El desarrollo computacional ha facilitado no solo la implantación sino también el entendimiento de los Modelos de Markov, haciéndolos más propicios como herramienta predictiva.

Xing et al. (2013) emplean un Modelo Oculto de Markov para estimar los valores de máxima verosimilitud del precio, junto con un algoritmo de media ponderada para aumentar la precisión del modelo. Estos autores muestran cómo este Modelo de Markov obtiene unos resultados predictivos ampliamente superiores al análisis de regresión lineal, con un error medio de 0,02 frente a 0,51.

De los Modelos de Markov también deriva toda la familia de modelos de heterocedasticidad condicional autorregresiva (ARCH). Los modelos GARCH, y todas las variaciones de este que se han ido desarrollando, han demostrado una gran precisión a la hora de modelar la volatilidad y por ende el riesgo, convirtiéndose en una de las herramientas con mayor acogida entre los autores de esta disciplina (Caporale & Zekokh, 2019; Li & Zhang, 2021).

Si bien la aplicación de estos métodos en contextos de volatilidad o aleatoriedad extrema, siendo el mercado cripto un ejemplo paradigmático de estas circunstancias, no siempre resulta óptimo, de ahí la preferencia por los modelos de Machine Learning (Sen, 2021).

Apuntan Fatima & Hussein (2008) que métodos como ARIMA, ARCH o GARCH son incapaces de proyectar de manera precisa series temporales ya que además de asumir normalidad en la distribución de los datos, se basan en la teoría de los procesos estocásticos estacionales, de manera que sus propiedades estadísticas no varían a lo largo del tiempo.

Sin embargo, concluyen estos autores que los resultados de su modelo de red neuronal artificial mejoran al desarrollar un modelo híbrido ANN-GARCH, en el que primero se aplica la metodología GARCH para estimar parámetros y predecir valores y, posteriormente, se emplea la red neuronal para modelar dichas predicciones.

En una línea similar trabajan Usmami et al. (2016), incorporando los resultados de técnicas estadísticas como la media móvil o la ARIMA como inputs en sus modelos de Machine Learning.

2.2.2. Modelos supervisados

Los modelos supervisados son aquellos en los que el algoritmo de aprendizaje es entrenado a partir de datos etiquetados, es decir, aquellos en los que las distintas observaciones o inputs tienen asociado un valor o output.

El objetivo de los modelos supervisados es desarrollar o inferir reglas que puedan ser aplicadas con suficiente fiabilidad a nuevos datos, tanto para su clasificación como para la realización de análisis de tipo regresivo (Simonian et al., 2019).

Su popularidad en la predicción de la cotización de activos financieros viene aumentando (Sharma & Kaushik, 2017), y es que sus características, entre las que destacan la ausencia de multicolinealidad o su capacidad para operar en conjuntos de datos multidimensionales, superan las limitaciones habituales de los modelos lineales y no lineales más simples (Sharma & Kaushik, 2017; Simonian et al., 2019). Adicionalmente, se trata de una técnica multifacética en cuanto a que puede realizar tanto tareas de regresión como de clasificación (Vijh et al., 2020).

Simonian et al. (2019) consideran la utilización de los mínimos cuadrados ordinarios, siendo el retorno de la cartera la variable dependiente, como el modelo más simple siempre

y cuando las variables independientes que se lleguen a emplear presenten una correlación lo suficientemente baja.

Entrando en arquitecturas más elaboradas y con demostrada validez en el campo de la predicción y proyección de series temporales podemos mencionar en primer lugar los *random forest*. Se trata de un modelo basado en la acumulación sucesiva de árboles de decisión para llegar a la predicción final (Simonian et al., 2019).

Por último, cabe también mencionar las arquitecturas más básicas como los árboles de decisión, la regresión logística y la regresión lineal, que para tareas complejas, como la que se pretende en el presente trabajo resultan en principio excesivamente limitados, aunque la regresión lineal se emplea en el presente trabajo como modelo de referencia para determinar la capacidad predictiva.

2.2.3. Modelos no supervisados

Los modelos no supervisados de Machine Learning tienen por objetivo el inferir o definir estructuras ocultas dentro de un conjunto de datos, caracterizándose, y esta es la diferencia esencial respecto de los modelos supervisados, por no encontrarse esos datos categorizados o clasificados. Su utilidad es por tanto la de proporcionar una cierta estructura a los datos (Simonian et al., 2019).

Si bien la predicción de cotizaciones futuras es, en el fondo, una predicción de series temporales en la que resultan de aplicación los modelos supervisados, los modelos no supervisados también encuentran algunos ámbitos de actuación propios.

López de Prado y Lewis (2018), desarrollaron un algoritmo no supervisado para la detección de estrategias de inversión falsas derivadas de modelos supervisados, además de concluir que su utilización podría extenderse a ámbitos como la gestión del riesgo (“risk management”) o en la identificación de grupos no correlacionados.

2.3. La utilización de redes neuronales para la inversión.

Las redes neuronales son una variedad de los modelos supervisados que pretende simular la funcionalidad de las neuronas en el cerebro humano, consistiendo en modelos de carácter

autoajustables pues tienen la capacidad de modificar los parámetros que la componen en base a nuevos datos que se le proporcionen a la red (Sharma & Kaushik, 2017).

Si bien las redes neuronales parecen la arquitectura óptima en el contexto de las finanzas, no podemos obviar que su utilización no ha aportado siempre resultados óptimos, y es que la precisión de estos modelos es sumamente sensible a la cantidad y calidad de las variables empleadas, la preparación de los datos empleados y la propia arquitectura de la red neuronal (Ersan et al., 2020).

Y es que dentro de las redes neuronales también existe una gran disparidad de arquitecturas que deriva en diferentes conclusiones en la literatura.

2.3.1. *Redes Neuronales Artificiales (ANN)*

Como exponen Yetis et al. (2014), las redes neuronales artificiales, en adelante ANN por sus siglas en inglés *artificial neural networks*, son sistemas de procesamiento de información inspirados en la construcción matemática de las neuronas humanas.

Su función esencial es la de simular y analizar patrones complejos en conjuntos de datos no estructurados, en base a la asignación de pesos a las neuronas que componen cada una de las capas (Vijh et al., 2020), lo que le distingue de los modelos predictivos convencionales.

Las redes neuronales artificiales han sido empleadas de manera asidua en el marco de la investigación financiera, con resultados positivos con carácter general.

Una de las primeras muestras del potencial de las ANN en el mundo de las finanzas se presenta en Kimoto et al. (1990). Dada la demostrada validez, ya entonces, de las ANN para el reconocimiento de patrones, los autores se plantearon la posibilidad de que estos modelos pudieran resultar también efectivos en la predicción del mercado de valores, demostrando la mayor validez del modelo ANN en comparación con el análisis de regresión múltiple.

Como explican Selvamuthu et al. (2019), los modelos ANN incorporan una serie de funciones límite o umbrales, de manera similar a los modelos SETAR tan aclamados en materia predictiva, junto con diversos algoritmos de aprendizaje que proporcionan mayores posibilidades de predicción.

Otra versión de este tipo de redes neuronales, sería la de regresión generalizada (GRNN) que además de no asumir ninguna distribución de los datos que se le proporcionan, cuenta con un único parámetro de activación por lo que puede resultar especialmente óptimo en problemas de clasificación.

2.3.2. *Redes Neuronales Convolucionales (CNN)*

También las redes neuronales convolucionales (convolutional neural networks o CNN) han tratado de implementarse en el ámbito financiero a pesar de que normalmente se empleen para tareas como el reconocimiento facial o la conducción autónoma. Y es que cabe plantear la posibilidad, si quiera a nivel teórico, de que las CNN fueran capaces de, viendo una imagen de la cotización reciente de una acción, predecir su comportamiento futuro.

Si bien la literatura al respecto es limitada, cabe mencionar el reciente trabajo de Zolotareva (2021) quien trató de predecir el comportamiento de determinadas acciones en el largo plazo mediante la digitalización de las imágenes de su cotización pasada que después fueron nutridas a una red CNN.

Afirma la autora que los resultados no son óptimos debido a las limitaciones del modelo y posibles inconsistencias en los datos originales empleados en la fase de entrenamiento, pero que a pesar de ello el modelo identificaba de manera satisfactoria los cambios de tendencia en las cotizaciones en el largo plazo.

Anteriormente, Siripurapu (2015) había realizado un ejercicio similar en el que afirmaba que los resultados obtenidos eran decepcionantes y que se obtendrían mejores resultados al azar que siguiendo su modelo ya que el R^2 obtenido no fue mayor a 0 en ningún caso.

En el contexto del análisis técnico, encontramos por ejemplo el análisis Gann, desarrollado por William Gann en base a su ‘Ley de la Vibración’, en el que las secuencias geométricas y ángulos de los gráficos de precios se emplean para tratar de predecir las variaciones futuras (Zuckerman, 2019).

Por ello, a pesar de la ausencia de resultados claros hasta la fecha, y de que las principales estrategias de análisis técnico pueden incorporarse de manera matemática a otros tipos de redes neuronales con una validez sustancialmente superior hasta el momento, la utilización

de modelos CNN para la predicción de cotizaciones sigue siendo un terreno prometedor que podría ganar adeptos en los próximos años, como también afirman Aygün & Kabaçi (2021) tras analizar la utilización de distintos modelos de ML para la predicción del precio de criptomonedas.

2.3.3. *Redes Neuronales Recurrentes (RNN)*

Las redes neuronales recurrentes (*Recurrent Neural Networks*), en adelante RNN, son ampliamente consideradas como la arquitectura óptima en problemas relacionados con series temporales como la cotización de las acciones.

Cabe definir las redes neuronales recurrentes como “*sistemas dinámicos con capacidad de representar estados temporales*” (Du & Swamy, 2019, p.351), con una capacidad predictiva superior a la de otros modelos de Machine Learning o estadísticos en el análisis de series temporales (Albariqi & Winarko, 2020; Dey et al., 2021).

Afirman Dey et al. (2021), que la propia estructura del mercado de capitales, eminentemente desestructurada, también refuerza la utilización de este tipo de redes neuronales capaces de capturar los patrones ocultos de las series temporales.

Una de las modalidades, o versiones mejoradas, de las RNN son los modelos GRU, acrónimo de gated recurrent unit, que incorporan un vector de actualización y otro de reseteo para tratar de solventar el problema del desvanecimiento del gradiente (Dey et al., 2021).

De entre los distintos algoritmos de aprendizaje que se abarcan bajo el paraguas de las redes neuronales recurrentes, destaca también el Long-Short Term Memory (LSTM), que permite resolver algunos de los problemas tradicionalmente asociados a las RNN como el desvanecimiento de gradiente (Dey et al., 2021) o su incapacidad para archivar memoria en el largo plazo (Moghar & Hamiche, 2020), aspecto este último en el que se funda su principal ventaja teórica respecto a la anterior arquitectura GRU.

Así, los modelos LSTM, defienden Greff et al. (2017), permiten escalar las redes neuronales recurrentes a series temporales de amplia duración y capturar tendencias en el largo plazo.

En su forma básica o “vanilla”, el modelo LSTM se compone de una célula, y puertas de entrada, salida y olvido, a las que puede aplicarse una pluralidad de funciones de activación (Greff et al., 2017), de manera que la célula registra y recuerda los valores de cada una de las puertas y estas últimas a su vez regulan la transmisión de la información asociada a la célula, componiendo lo que se conoce como un bloque de memoria (Van Houdt et al., 2020).

Los bloques de memoria se caracterizan por mantener su estado en el tiempo regulando el tránsito de la información, además de encontrarse conectado tanto al bloque de origen como a las puertas antes mencionadas, de ahí su carácter de recurrente (Graff et al., 2015; Van Houdt et al., 2020).

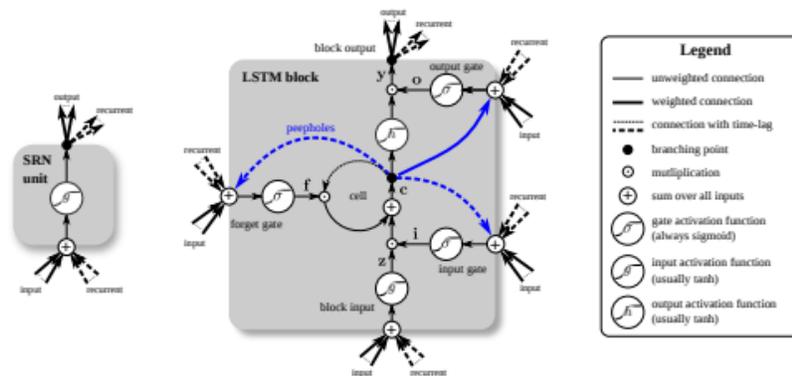


Ilustración 1: Representación gráfica de la arquitectura de una red LSTM

Fuente: Greff et al. (2015), p. 2

Van Houdt et al. (2020) concluyen su trabajo determinando como arquitectura recomendada para la predicción de series temporales el modelo básico de LSTM por su capacidad predictiva, facilidad de configuración y aplicabilidad tanto a problemas de regresión como de clasificación, si bien se trata de una aproximación meramente teórica.

2.3.4. Comparativa de la validez obtenida en la literatura por las Redes Neuronales y otros modelos predictivos

Vijh et al. (2020) muestran que la red neuronal artificial mejora la predicción de un modelo Random Forest en 4 de las 5 acciones empleadas para el análisis. Por su parte Usmani et

al. (2016) concluyen que la técnica MLP consigue mejores predicciones que el SLP o *single layer perceptron* y que el modelo SVM.

También Gheyas & Smith (2009) realizaron un análisis comparativo de distintas arquitecturas, en el que concluyeron que el modelo GRNN muestra una capacidad predictiva ampliamente superior a la de otros algoritmos como ARIMA o GARCH.

Adebiyi et al. (2014) adoptan la postura contraria considerando que el modelo ANN es superior al ARIMA en la predicción de los precios de cotización, particularmente si se trata de predecir un valor y no solo la dirección del movimiento. Los propios autores sin embargo aluden a la contradictoria literatura existente en lo relativa a la capacidad predictiva de los modelos ARIMA y ANN.

Por su parte, Mallikarjuna & Rao (2019) obtienen resultados predictivos claramente inferiores con una red neuronal artificial (ANN) que con los modelos econométricos ARIMA o SETAR, mientras que Remus y O'Connor (2001) afirman una capacidad predictiva similar en el corto plazo pero mayor en el largo plazo, especialmente en series en las que existan discontinuidades.

En cambio, Yetis et al. (2014) obtuvieron resultados esperanzadores con su red neuronal, presentando un error inferior al 2% en la predicción de la cotización del índice NASDAQ mediante redes de perceptrón multicapa (MLP o multi layer perceptron)

A su vez, Ersan et al. (2020) realizan una extensa comparación entre los métodos de aprendizaje de Support Vector Machines, k-Nearest Neighbour (en adelante, SVM y k-NN respectivamente) y redes neuronales artificiales, concluyendo que estos dos últimos muestran un rendimiento superior al modelo SVM, siendo a su vez el modelo k-NN ligeramente más preciso que la red neuronal artificial.

Estos resultados contradicen de manera frontal los hallazgos de Cao & Tay (2003) que afirmaban la superioridad de los modelos SVM respecto del modelo k-NN y las redes neuronales artificiales en la proyección de series temporales financieras.

Sin embargo, también encontramos en la literatura estudios en los que las redes neuronales muestran una capacidad predictiva claramente superior a la de los modelos SVM (Sharma & Kaushik, 2017).

Por otro lado, Dey et al. (2021) llevan a cabo un análisis comparativo práctico entre las arquitecturas RNN básica, LSTM y GRU, afirmando que estos dos últimos modelos, muestran una capacidad predictiva ampliamente superior al modelo básico. El modelo GRU muestra además mayor capacidad predictiva en contextos de alta y baja volatilidad, mientras que las predicciones del modelo LSTM resultan más precisas en contextos de volatilidad media.

2.4. Análisis de la aplicación los modelos predictivos en el ámbito de las criptomonedas

Tras la aproximación realizada en los apartados anteriores acerca de las ventajas, inconvenientes y validez de los distintos modelos tradicionales y de Machine Learning en lo relativo a la predicción de precios de cotización, se procede a continuación a focalizar el estudio en el mercado de las criptomonedas.

Lo cierto es que la información disponible es, como cabría esperar, limitada en comparación con la predicción de acciones, además de presentar muchos de ellos distintas limitaciones en cuanto a la metodología aplicada.

Cabe destacar que precisamente la limitada atención que los practicantes del Machine Learning han prestado a este campo una de las razones que fundamentan la realización del presente trabajo.

2.4.1. Arquitecturas empleadas en la literatura para la predicción de la cotización de criptomonedas

Madan et al. (2014) fueron pioneros en la implementación de técnicas de Machine Learning para la predicción de variaciones en el precio de cotización del Bitcoin, en base a diversa información obtenida del blockchain para complementar los precios de cotización del activo, el input más habitual en la predicción de cotizaciones.

Atendiendo a sus hallazgos, el modelo lineal generalizado (GLM), se muestra superior al modelo de Random Forest, si bien, y como los propios autores puntualizan, estos resultados podrían estar contaminados por el hecho de que el dataset empleado muestra una proporción de movimientos positivos muy superior a los movimientos negativos.

Por su parte, Greaves y Au (2015) fueron de los primeros en realizar un análisis comparativo entre una pluralidad técnicas de Deep Learning para la predicción de la cotización del Bitcoin. Además de realizar un interesante proceso de selección de parámetros, basándose de nuevo en la información del blockchain de Bitcoin, analizan la capacidad predictiva de modelos de regresión lineal, regresión logística, SVM y ANN.

Comparando los resultados obtenidos por los modelos anteriores contra una referencia base proporcionada por el incremento porcentual medio del precio del Bitcoin, obtienen un rendimiento mejor al de referencia, pero sólo ligeramente. Cabe destacar que los autores concluyen que la información proporcionada por el blockchain no resulta excesivamente informativa y que, de hecho, el precio actual del Bitcoin es la variable de mayor relevancia para la predicción de su cotización futura.

Los esfuerzos predictivos en el campo de las criptomonedas se han incrementado exponencialmente, profundizando además en la distinción entre las predicciones basadas en los aspectos tecnológicos de las criptomonedas y el blockchain, y las basadas en sus aspectos más financieros como el volumen y la cotización (Aygün y Kabaçi, 2021).

Desde entonces, son numerosos los trabajos que incorporan este tipo de variables a la predicción del precio de criptomonedas. Cabe mencionar a Jang y Lee (2018), Chen et al. (2019), Albariqi y Winarko (2020), o Vaidehi et al. (2021), que también incorporan información procedente del blockchain de Bitcoin para predecir su cotización.

Chen et al. (2019) focalizan su trabajo en la dimensión de la muestra y en la granularidad del precio del Bitcoin, comparando el error de predicción con una granularidad de 5 minutos y de 1 día, pero manteniendo el foco en variables como el tamaño medio de cada bloque, la dificultad de minado o el Hash rate.

También cabe incluir los modelos basados en el sentiment analysis de las redes sociales y que han proliferado mucho en el ámbito de la predicción financiera y, en concreto, del Bitcoin.

Autores como Aygün y Kabaçi (2021) emplean modelos ARIMA para predecir la cotización de criptomonedas, si bien no prolifera en la literatura, como sí sucede en el

ámbito de la predicción de mercados bursátiles tradicionales, la utilización de modelos GARCH (Mudassir et al., 2020)

2.4.2. Posibilidades de aplicación de las redes neuronales para la predicción de la cotización de las criptomonedas

Greaves y Au (2015) muestran que la red neuronal muestra la máxima precisión de entre los modelos de clasificación empleados, superando ampliamente a los modelos de regresión logística y, sobre todo, el SVM que a penas mejora el modelo de referencia.

Aygiün y Kabaçi (2021) realizan una comparativa entre las principales modalidades de redes neuronales: ANN con 1 capa, ANN con 3 capas, CNN y RNN, concluyendo que la RNN muestra un rendimiento superior al del resto de redes neuronales, si bien esta diferencia no resulta estadísticamente significativa respecto al modelo CNN. Adicionalmente, la red neuronal presenta una capacidad predictiva ampliamente superior a la de mecanismos predictivos tradicionales como la media móvil simple (SMA) o los modelos ARIMA.

Munim et al. (2019) ya habían realizado anteriormente un ejercicio similar, comparando la capacidad predictiva de un modelo ARIMA con una modalidad particular de las ANN que denominan red neuronal de auto regresión (NNAR), llegando sin embargo a una conclusión opuesta y es que el modelo ARIMA se muestra superior al modelo NNAR en los datos de test, a pesar de rendir mejor en el proceso de entrenamiento.

A una conclusión similar llegan Chen et al. (2019) concluyendo que los modelos estadísticos muestran una capacidad predictiva significativamente superior a la de los modelos basados en redes neuronales para problemas de clasificación con datos de baja frecuencia, mientras que el rendimiento de las redes neuronales resulta superior al emplear datos de alta frecuencia. Cabe además destacar que de entre las arquitecturas de red neuronal analizadas, es la LSTM la que presenta una mayor precisión.

Al considerar posibles estudios futuros, los autores aluden a estudios anteriores en los que se había determinado la inferioridad de los modelos NNAR respecto a las arquitecturas de RNN, y que la predicción con RNN debía de ser un aspecto a considerar en el futuro para mejorar los resultados obtenidos en su trabajo.

Jiang (2020) realiza un estudio focalizado precisamente en las estructuras de RNN, concluyendo que la arquitectura GRU con 2 capas obtiene la menor raíz del error cuadrático medio (RMSE), seguido de cerca por el modelo LSTM con 2 capas. Entre los modelos GRU y LSTM no se observan diferencias notables, afirmando el mismo autor que cada modalidad podría ser válida en función de las características del mercado, pero ambas arquitecturas sí mejoran significativamente los resultados obtenidos por modelos de perceptrón multicapa (MLP).

Sin embargo, los modelos MLP sí que muestran una capacidad predictiva notable en problemas de clasificación (Sin & Wang, 2021; Albariqi & Winarko, 2020), a pesar de no poder extrapolarse este desempeño favorable al problema de regresión que nos atañe en el presente trabajo.

Uno de los trabajos de mayor calidad y profundidad en este ámbito es el presentado por Cocco et al. (2021), en el que no sólo se distingue entre distintas modalidades de redes neuronales, sino también entre modelos de una sola etapa, los modelos que podemos considerar ‘habituales’ en la literatura, y modelos de dos etapas en los que se combinan distintas arquitecturas.

Los autores concluyen que los modelos de dos etapas muestran una capacidad predictiva mayor que los modelos de una sola etapa y que, en concreto, la combinación de los modelos LSTM y SVR se erige como una de las mejores arquitecturas.

Una arquitectura similar se muestra en Li & Dai (2019), los cuales presentan un modelo híbrido CNN-LSTM que mejora la capacidad predictiva de cada una de las dos modalidades de red neuronal por separado.

Por su parte, Mudassir et al. (2020), en un trabajo extremadamente completo y preciso, concluyen que de entre los modelos ANN, SANN, SVM y LSTM, es este último el que presenta los mejores resultados generales.

En definitiva, las redes neuronales son una opción, si no óptima, por lo menos razonable para la predicción del precio de criptomonedas, particularmente si se trata como un problema de regresión y no de clasificación.

De entre las distintas modalidades de redes neuronales, destacan las RNN a pesar de que su precisión suele situarse por debajo del 60% en la literatura (Albariqi & Winarko, 2020), observándose del mismo modo una clara preferencia por las redes LSTM y en menor medida GRU.

2.4.3. *Limitaciones de la literatura disponible en el ámbito de las criptomonedas*

Sin hacer de menos el trabajo de los autores expuestos hasta el momento, cabe realizar diversas puntualizaciones a su trabajo, las cuales nos proporcionan a su vez posibles explicaciones a los resultados obtenidos en estos.

Así, por ejemplo, que Greaves & Au (2015), no mejoren en exceso el modelo de referencia, podría deberse a que tratan de predecir el precio de cotización en lugar de los retornos del Bitcoin.

Albariqi & Winarko (2020) también ponen el foco en estas circunstancias criticando que gran parte de la literatura disponible promueve resultados positivos sin ningún tipo de validación cruzada, lo que deriva probablemente en overfitting del set de entrenamiento y en una incógnita no resuelta respecto de la posibilidad de generalizar los resultados obtenidos.

Podemos destacar sin embargo, por la calidad que muestra la metodología desarrollada en los mismos, los trabajos de Cocco et al. (2021), Mudassir et al. (2020) o los propios Albariqi & Winarko (2020) que llevan a cabo las necesarias acciones para asegurar la replicabilidad del modelo incluyendo la validación cruzada, el preprocesamiento de los datos, la predicción de retornos en lugar de precios, y la optimización de parámetros o *hyperparameter tuning*.

3. LA PREDICCIÓN DEL PRECIO DE COTIZACIÓN DEL BITCOIN MEDIANTE UN MODELO LSTM COMO OBJETIVO

En el apartado anterior se ha puesto de manifiesto como numerosos autores han centrado sus esfuerzos en la predicción de precios de cotización. Una parte de estos, han puesto a su vez el foco en el ámbito de las criptomonedas y en concreto en el Bitcoin. Así mismo, se ha expuesto en detalle la capacidad de las redes LSTM para la predicción de series temporales y su potencial en el ámbito financiero.

Por todo ello, objetivo principal del presente trabajo es el de analizar la capacidad predictiva de estas redes LSTM en la predicción del precio del Bitcoin. Sin embargo, para diferenciar este estudio de otros ya realizados por autores destacados al respecto, en lugar de tratar de predecir el signo del movimiento del precio se va a tratar de predecir la variación del mismo, por lo que se trata de un problema de regresión respecto del que no existe literatura excesiva en el campo de las criptomonedas.

El perímetro del trabajo queda limitado en gran medida por la capacidad computacional de la que se dispone para desarrollar el mismo, por lo que no se espera obtener un modelo que prediga casi a la perfección los movimientos del Bitcoin, sino más bien el objetivo es reafirmar el antes mencionado potencial predictivo de las redes LSTM en este contexto, y para ello se compararán los resultados obtenidos entre distintas arquitecturas a efectos de afirmar o negar dicha capacidad predictiva.

Otro de los objetivos de este análisis, es el estudio de la utilización de indicadores técnicos como inputs para la predicción. En tanto un número importante de inversores se basa en estos para tomar sus decisiones de inversión, consideramos que estas variables podrían albergar información predictiva relevante.

Por último, también se pretende proyectar el ámbito de estudio de la predicción de criptomonedas a los datos de alta frecuencia. La mayoría de los trabajos existentes tratan horizontes de tiempo basados en horas o incluso días, mientras que nuestro análisis trabaja con datos minuto a minuto y un periodo de predicción de 5 minutos.

En definitiva, se busca abordar algunos de los aspectos que menor atención han recibido hasta el momento en la predicción de la cotización del Bitcoin, un campo que de por sí no

ha recibido toda la atención que podría por parte de los practicantes de Machine Learning, algo que resulta lógico por otra parte dado lo novedosos de su generalización el ámbito financiero.

4. ANÁLISIS PRÁCTICO DE LA CAPACIDAD PREDICTIVA DE LAS REDES LSTM EN EL ÁMBITO DE LAS CRIPTOMONEDAS

Tras estudiar la literatura disponible y los modelos desarrollados en las publicaciones a las que se ha tenido acceso, se ha concluido que las redes neuronales presentan un enorme potencial predictivo. A diferencia de lo que sucede con el análisis predictivo de acciones, la utilización de redes LSTM no ha sido estudiada con tanto detalle, y la mayoría de los trabajos existentes lo abordan en tanto que problema de clasificación.

El presente trabajo pretende determinar la capacidad predictiva de las redes LSTM para la cotización del Bitcoin, tratándolo como problema de regresión y no de clasificación, y para reducidos periodos de tiempo.

En el presente apartado se pretende describir los pasos previos necesarios para diseñar y afinar la red LSTM, la arquitectura final seleccionada, y los resultados predictivos obtenidos conforme a la misma, comparándolos contra una serie de modelos base que se expondrán en el subapartado correspondiente.

4.1. Conformación del dataset inicial

4.1.1. Recolección de datos

Dado que el objetivo del modelo es la predicción del precio del Bitcoin en el corto plazo, en primer lugar, es necesario disponer de los datos de cotización de esta criptomoneda minuto a minuto.

Se ha acudido a la página Crypto Data Download¹ para descargar las cotizaciones de las siguientes criptomonedas: Bitcoin (BTCUSD), Ethereum (ETHUSD), Litecoin (LTCUSD) y Ripple (XRPUSD). El dataset inicial cuenta con las siguientes variables respecto de estas monedas: apertura, máximo, mínimo, cierre, volumen en dólares y número de monedas intercambiadas cada minuto entre el 16 de agosto de 2017 y el 6 de junio de 2022. La fecha

¹ Todos los datos mencionados de esta página pueden descargarse en <https://www.cryptodatadownload.com/data/bitstamp/>

de inicio se debe a que no se dispone de observaciones anteriores a dicha fecha para ETHUSD.

Sin embargo, debido a las exigencias computacionales y a la limitada capacidad computacional de la que se dispone el tamaño del dataset ha sido reducido considerablemente. De hecho, todas las técnicas que se expondrán de aquí en adelante no han podido ser aplicadas con datos superiores a 1 mes. Por ello, el tamaño del dataset empleado es de 44.640 observaciones que se corresponden con el periodo entre ‘2022-01-01 00:00:00’ y ‘2022-31-01 23:59:00’, es decir, el mes de enero de 2022.

4.1.2. Tratamiento de observaciones ausentes en el dataset inicial

Dado que estos datos serán la base para otras posibles variables, es necesario asegurar que son correctos, así como corregir algunas deficiencias que pueda haber en el dataset. A su vez, dado que la mayoría de estos datos serán transformados posteriormente a su variación porcentual o logarítmica respecto de la observación anterior, el valor 0 puede plantear dificultades ya que derivaría en un resultado de infinito cuando tratemos de calcular dichas variaciones.

Tras observar el dataset, son las columnas relativas al volumen intercambiado las que suelen contener valor 0. En el periodo seleccionado, 131 observaciones contaban con este problema por lo que tras su eliminación el dataset se compone de 44.509 observaciones.

4.2. Tratamiento del dataset

4.2.1. Incorporación de variables adicionales

Con vistas a maximizar la capacidad predictiva del modelo que se desarrolle, se ha decidido incorporar una serie de variables adicionales cuya incorporación definitiva al modelo se considera en el apartado 0.

En primer lugar, se ha decidido añadir una variable adicional para cada criptomoneda secundaria que represente el múltiplo de BTCUSD respecto de la misma criptomoneda. Puede que una variable que relacione la cotización del par de monedas resulta más informativa que la cotización individual de cada criptomoneda a efectos de la predicción del precio del Bitcoin. Estas ratios incorporadas, proporcionan además información que

sería esencial en una estrategia de ‘pairs trading’, muy común entre los inversores institucionales, por lo que podría incrementar la capacidad predictiva del modelo.

El grueso de las restantes variables incorporadas encuentran su fundamento en el análisis técnico. El racional para su incorporación sería el siguiente: si numerosos inversores emplean estos indicadores para decidir operar con un determinado activo, la incorporación de estos mismos indicadores podría resultar relevante para modelizar la evolución de la cotización del mismo.

Dado que el objetivo del presente trabajo es predecir variaciones de la cotización en el corto plazo, y que el Bitcoin carece de elementos en los que se basan los inversores en valor como cuentas de resultados, el análisis técnico podría aportar incluso más información en el ámbito de las criptomonedas. La literatura apoya la incorporación de estas variables.

El principal trabajo al respecto es el realizado por Huang et al. (2019) en el que emplean hasta 124 indicadores técnicos para tratar de predecir el precio del Bitcoin. Si bien los autores no tratan de predecir un valor concreto sino el rango de variación de la cotización de un día al siguiente, su modelo de Random Forests muestra una capacidad predictiva razonable.

Entre los indicadores empleados, podemos encontrar ampliamente reconocidas bandas de Bollinger, diversos indicadores relacionados con la media móvil, osciladores de momento, la volatilidad del activo, y patrones o figuras tradicionales del análisis técnico.

Por su parte, Cocco et al. (2021) emplean como indicadores técnicos la media móvil simple, la media móvil exponencial, el momentum, la media móvil de convergencia/divergencia (indicador MACD) y el índice de fuerza relativa (RSI).

Otros indicadores técnicos empleados en la literatura incluyen el índice de flujo de dinero (Money Flow Index o MFI), el volumen en balance (OBV), la desviación típica, el rate of change (ROC), la media móvil exponencial triple (Li & Dai, 2019; Mudassir et al., 2020).

Resulta necesario realizar una puntualización, y es que la mayoría de indicadores técnicos actúan en ventanas diarias, mientras que en el presente trabajo se está empleando datos en minutos.

Una posible solución sería simplemente mantener el número de periodos previsto por el indicador en minutos en lugar de horas o días (Lien, 2022). Lo cierto es que la literatura no parece haberse pronunciado en exceso sobre este aspecto, por lo que este será el mecanismo empleado para incorporar los indicadores.

En primer lugar, se han incorporado la media móvil exponencial (EMA) con una pluralidad de periodos siguiendo la literatura consultada (Huang et al., 2019; Cocco et al., 2021). Como indica Adewuyi (2016), la media móvil exponencial es en el fondo una versión simplificada de los modelos ARCH y GARCH, la base sobre la que se construyen dichos modelos, por lo que su inclusión en el modelo tendría un impacto similar, aunque computacionalmente mucho más barato, al diseño y ajuste de un modelo autorregresivo.

Esta media móvil se ha calculado tanto respecto del precio de cierre (EMA_c) como del volumen (EMA_v), y adicionalmente se ha incorporado una ratio para comparar el cierre y volumen en N_t con EMA_c y EMA_v .

El racional de la incorporación preliminar de estas ratios junto con la media móvil, es que relativizar el valor de la observación actual con la media de x observaciones anteriores, podría resultar más útil en términos predictivos que ambas variables por separado.

La media móvil exponencial también es la base del indicador MACD, desarrollado por Gerald Appel en la década de 1970, que consiste en comparar una señal que habitualmente es la EMA de 9 días, contra el MACD, calculado a su vez en su versión original como la diferencia entre la EMA de 12 días y la de 26 días. Los cruces entre la señal y el MACD, así como las 2 líneas que componen el MACD, conceden numerosas oportunidades de actuación a los inversores por lo que se ha convertido en uno de los indicadores más empleados en el análisis técnico (Hoang Hung, 2016; Kang, 2021).

Otro de los pilares del análisis técnico es el RSI, un indicador de momento mediante el que se determina la fortaleza de un determinado activo en base a los movimientos positivos y negativos del mismo. El RSI proporciona un valor entre 0 y 100, considerándose como puntos de entrada y salida los umbrales de 30 y 70 respectivamente (Morosan, 2011; Panigrahi et al. 2021).

Morosan (2011) propone una versión modificada en la que dicha fortaleza se calcula no sólo en base al precio del activo, sino también al volumen intercambiado, la cual permite no tanto identificar una reversión de la tendencia sino la continuación de la misma, pero que en cualquier caso proporciona mejores resultados que la versión original del RSI. La formulación de este indicador sería por tanto la siguiente: $RSI = 50 * (1 + \frac{1}{1+RS_p} - \frac{1}{1+RS_r})$.

Para el presente trabajo el RSI para BTCUSD y ETHUSD han sido calculados y, adicionalmente, también se incorpora como variable la ratio entre ambos RSI de la siguiente manera: $RSI_{ratio} = RSI_{BTC}/RSI_{ETH}$.

Por último, una de las notas características de las criptomonedas es su volatilidad. La literatura ha concluido que las criptomonedas de mayor capitalización propagan además su volatilidad, siendo las responsables de los principales shocks en el mercado cripto (Yi et al., 2018).

La modelización de la volatilidad del Bitcoin es otro campo tratado en profundidad por la literatura. Bergsli et al. (2022) determinan la superioridad de los modelos de autorregresión heterogénea (heterogeneous autorregressive models o HAR) respecto a la familia de modelos GARCH, especialmente en predicciones en el corto plazo, debido a que los modelos HAR se basan en la varianza realizada, la cual presenta una capacidad predictiva de la volatilidad superior, además de acercarse a una distribución normal al aplicar una transformación logarítmica a la misma.

Se incorpora al modelo la varianza realizada del Bitcoin únicamente, ya que atendiendo a Yi et al. (2018), los shocks de volatilidad del resto de criptomonedas recogidas en el dataset serían consecuencia de la volatilidad del Bitcoin. Esta decisión se toma atendiendo además a las limitaciones computacionales a las que nos enfrentamos para realizar este trabajo.

Por último, es necesario incorporar la variable más relevante, aquella que el modelo va a tratar de predecir. Esta variable ‘target’ es el precio de Bitcoin a 5 minutos vista. Como ya se ha expuesto en reiteradas ocasiones, el objetivo de este trabajo es poner a prueba la capacidad predictiva de las redes neuronales con datos de alta frecuencia.

La literatura sobre la distancia hasta el periodo a predecir es algo contradictoria. Li & Dai (2019) muestran como el error de predicción aumenta a medida que se incrementa el

periodo predictivo, si bien Albariqi & Winarko (2020) obtienen una precisión ampliamente superior (81.3% vs. 70.0%) prediciendo a plazos más prolongados.

Y es que estos últimos autores consideran que una de las razones por la que los satisfactorios resultados obtenidos en la predicción de la cotización de acciones no han podido ser replicado en tal medida con las criptomonedas se debe a un excesivo cortoplacismo observable de manera generalizada en la literatura.

En cualquier caso, para el presente trabajo se ha optado por predecir 5 minutos en el futuro en lugar de 1 minuto ya que si en el futuro se quisiera emplear las predicciones del modelo para desarrollar una estrategia de inversión, se disponga de una ventana de tiempo suficiente para que el modelo realice la predicción.

En la Tabla 1 que se presenta a continuación, se proporciona un resumen de las 116 variables incorporadas en el momento inicial, junto con el periodo de aplicación de cada una de ellas.

Tabla 1: Variables que conforman el dataset inicial

Variable	Periodo	Criptomonedas empleadas
Apertura	1 minuto	BTC, ETH, LTC y XRP
Máximo		
Mínimo		
Cierre		
Volumen en monedas		
Volumen en USD		
Ratio de cotización respecto a BTC	1 minuto	ETH, LTC y XRP
Media móvil exponencial del Cierre (EMA_c)	5, 15, 30, 60, 120, 240, 360, 720 y 1.440 minutos	BTC y ETH
Ratio del precio de cierre respecto la media móvil del cierre (C_t/EMA_c)		
Media móvil exponencial del Volumen en USD (EMA_v)		
Ratio Volumen en USD respecto media móvil del volumen (Vol_t/EMA_v)		
MACD(12,26,9)	12, 26 y 9 minutos	BTC
RSI (RSI_{BTC} y RSI_{ETH})	15, 30, 60, 120, 360, 720 y	BTC, ETH
Ratio RSI de BTC y ETH ($RSI_{BTC/ETH}$)	1.440 minutos	-
Varianza real	15, 30 y 60 minutos	BTC
Target	-	BTC

4.2.2. Transformación de las variables

La utilización de los precios de cotización presenta numerosas dificultades para el proceso de aprendizaje de los modelos de ML. Una solución habitual es la de transformar los precios en retornos, calculando la variación del precio respecto de la observación anterior.

Por un lado, la utilización de retornos permite escalar las variables a un rango homogéneo, evitando así sobrevalorar una determinada variable por el mero hecho de que su cotización sea más elevada. De esta manera se facilita la comparación entre distintos activos (Viernes, 2021).

Sin embargo, los practicantes de ML habitualmente van un paso más allá y emplean la transformación logarítmica de los retornos. Por un lado, asumiendo que los precios siguen una distribución log-normal, los retornos logarítmicos se acercan a una distribución normal,

abriendo la posibilidad de aplicar una multitud de técnicas a los datos (Kamalov et al., 2021).

Apunta Meucci (2010, p.2) que *“los retornos acumulativos o logarítmicos siguen la regla de la raíz cuadrada: la media y la varianza crecen linealmente con el horizonte temporal, y la volatilidad crece como la raíz del horizonte”*.

Adicionalmente, a diferencia de los retornos lineales cuya propiedad aditiva se predica respecto de distintos activos, siendo por tanto óptimo para la gestión de portfolios, los retornos logarítmicos son aditivos en el tiempo (Meucci, 2010; Viernes, 2021) lo que les hace idóneos para el tratamiento de series temporales (Kim & Kim, 2019; Cocco et al., 2021), a pesar de presentar algunas dificultades en la construcción de plataformas de riesgo (Meucci, 2010).

La discusión sobre la incorporación de precios o retornos para la predicción en el ámbito financiero ha sido tratada en profundidad por numerosos estudios. Entre ellos cabe destacar el desarrollado por Kamalov et al. (2021), en el que se concluye, en contradicción con la postura general de los practicantes de Machine Learning, que, para la predicción de la dirección de la cotización, el precio muestra una mejor capacidad predictiva que los retornos.

Los autores hacen sin embargo especial énfasis en que su experimento es aplicable a la predicción de la dirección del precio, es decir, a un problema de clasificación. También apuntan los autores que la superioridad del precio como input respecto a los retornos se disipa al incorporar indicadores técnicos al modelo.

Por todo lo expuesto hasta ahora, consideramos óptimo para la finalidad del presente trabajo, que se ha diseñado como problema de regresión cabe recordar y que incorpora una pluralidad de indicadores técnicos, transformar las variables relativas a los precios de cotización, las ratios de cada moneda respecto a BTC y, por supuesto, la variable Target en sus retornos logarítmicos.

A pesar de que en un momento posterior las variables serán escaladas, otras variables han sido transformadas a su variación porcentual respecto de la observación anterior, sin transformación logarítmica en este caso.

Aquellas variables que representan ratios, como: C_t/EMA_c , Vol_t/EMA_v , $MACD(12,26,9)$, RSI_{BTC} , RSI_{BTC} , $RSI_{BTC/ETH}$ no serán transformadas ya que se considera que en su caso es el valor del ratio y no la variación respecto del periodo anterior lo que podría albergar capacidad predictiva. Tampoco se procede a la transformación a las variables que representan la varianza real ya que tanto de los retornos logarítmicos como de la variación porcentual derivan en errores de cálculo y a valores extremos.

La

Tabla 2 presenta de manera resumida la lista de variables junto con la transformación a la que han sido sometidas.

Tabla 2: Resumen de las transformaciones a las que se han sometidos las distintas variables

Variable	Transformación
Apertura	Logaritmo de los retornos
Máximo	
Mínimo	
Cierre	
Volumen en monedas	
Volumen en USD	
Ratio de cotización respecto a BTC	Logaritmo de los retornos
Media móvil exponencial del Cierre (EMA _c)	Variación porcentual
Media móvil exponencial del Volumen en USD (EMA _v)	
Ratio del precio de cierre respecto la media móvil del cierre (C _t /EMA _c)	-
Ratio Volumen en USD respecto media móvil del volumen (Vol _t /EMA _v)	
MACD(12,26,9)	-
RSI (RSI _{BTC} y RSI _{ETH})	-
Ratio RSI de BTC y ETH (RSI _{BTC} /ETH)	-
Varianza real	-
Target	Logaritmo de los retornos

Es necesario puntualizar que el cálculo de medias móviles, así como la transformación logarítmica o porcentual, generan observaciones sin valores para algunas variables. Sin embargo, al haberse reducido el tamaño del dataset a enero de 2022, la eliminación de las variables nulas generadas por estos procesos no afecta al tamaño del dataset empleado.

4.3. Selección de variables

En la práctica, son numerosos los autores que incorporan a sus modelos cientos de variables de diversas plataformas con un elevado nivel de correlación entre sí y tratan de mostrarlos posteriormente como ‘factores’ de inversión (Simonian et al., 2019).

Afirma Chan (2021) que una de las principales problemáticas en la predicción financiera mediante modelos de Machine Learning es que, si bien parece que la cantidad de

información disponible es prácticamente ilimitada, lo cierto es que la información financiera estadísticamente independiente y que no presente correlaciones elevadas es mucho más limitada.

Por ello, resulta esencial para garantizar la robustez y validez de los modelos de Machine Learning aplicar técnicas de selección de variables o *'feature selection'*.

Para llevar a cabo este cometido, se ha procedido en primer lugar a calcular la correlación de Pearson de cada variable respecto de la variable target. Se ha optado por prescindir de las variables cuya correlación es inferior a 0,001 para aligerar el dataset y poder proceder a realizar análisis multivariantes con posterioridad.

Este proceso ha resultado en la eliminación de las 27 variables que se presentan a continuación.

Tabla 3: Variables preliminares eliminadas en base a su baja correlación respecto a la variable Target.

Variable	Correlación	Variable	Correlación
Open_XRP_LogRet	-0,0009	ETH_RSI_60m	-0,0004
BTC/ETH_Ratio_720m	-0,0009	BTC/ETH_Ratio_120m	-0,0004
ETH_30m_VolEMA	0,0008	BTC/ETH_Ratio_60m	-0,0003
BTC_RSI_30m	0,0008	ETH_RSI_1440m	-0,0003
BTC_RSI_15m	-0,0008	BTC_RSI_120m	0,0003
ETH_30m_EMA/Close	-0,0007	BTC_RSI_1440m	-0,0002
BTC_RSI_720m	0,0007	ETH_RSI_30m	0,0001
ETH_RSI_15m	-0,0007	ETH_RSI_720m	0,0001
BTC_RSI_60m	-0,0007	ETH_15m_VolEMA	0,0001
Open_ETH_LogRet	0,0006	ETH_5m_VolEMA	0,0000
ETH_30m_EMA	-0,0006	BTC/ETH_Ratio_1440m	0,0000
ETH_RSI_120m	0,0006	BTC/ETH_Ratio_15m	0,0000
BTC/ETH_Ratio_360m	-0,0005	MACD_26_12_9	0,0000
ETH_RSI_360m	-0,0004		

Respecto de los resultados obtenidos, destaca la eliminación del indicador MACD, siendo la variable menos correlacionada con la cotización futura del bitcoin de las analizadas.

También se prescinde de un número relevante de variaciones del indicador RSI tanto para BTC como ETH.

De esta manera se logra no sólo reducir el número de variables totales a 87 además de la target, para poder como ya se ha expuesto un análisis multivariante de las variables a mantener, sino que además se elimina parte de los indicadores con mayor número de variantes y que por tanto mostraban una correlación notable entre sí.

Esta primera reducción de la dimensión del dataset permite comenzar a emplear algunas de las técnicas de selección de variables previstas en el paquete sklearn.

En primer lugar, se ha procedido a realizar un f-test sobre. Se trata de un análisis univariante que determina la capacidad predictiva de cada variable bajo la hipótesis nula de que todas las variables tienen coeficiente 0, es decir, carecen de capacidad predictiva. Por tanto, un p-valor lo suficientemente bajo permite rechazar la hipótesis nula indicando que dicha variable sí goza de capacidad predictiva (Lee, 2021).

Empleando un nivel de significación alpha del 0,05, se procede a la eliminación de 24 variables adicionales que se enumeran en la siguiente imagen (no se proporciona en formato de tabla ya que Excel tiene dificultades para computar de manera correcta los valores reales del p-valor, distorsionando algunos de los resultados) junto con el p-valor obtenido en el f-test.

Ilustración 2: variables eliminadas en base al f-test*

BTC_15m_VolEMA_pctchange	2.709945	9.972504e-02
BTC_15m_VolEMA/Vol	2.573106	1.086948e-01
BTC_30m_VolEMA/Vol	1.810575	1.784399e-01
BTC_60m_VolEMA/Vol	1.385583	2.391530e-01
BTC_120m_VolEMA/Vol	1.107487	2.926293e-01
BTC_5m_VolEMA_pctchange	0.905295	3.413664e-01
BTC_240m_VolEMA/Vol	0.752839	3.855792e-01
ETH_60m_VolEMA_pctchange	0.656178	4.179129e-01
BTC/ETH_Ratio_30m	0.477819	4.894122e-01
BTC_360m_VolEMA/Vol	0.471015	4.925210e-01
ETH_1440m_VolEMA_pctchange	0.286387	5.925456e-01
BTC_1440m_VolEMA/Vol	0.169134	6.808828e-01
ETH_120m_VolEMA_pctchange	0.158961	6.901149e-01
Volume_XRP_LogRet	0.142832	7.054814e-01
Vol_USD_XRP_LogRet	0.139867	7.084131e-01
Volume_LTC_LogRet	0.080878	7.761105e-01
Vol_USD_LTC_LogRet	0.079718	7.776805e-01
ETH_720m_VolEMA_pctchange	0.049795	8.234207e-01
BTC_RSI_360m	0.024368	8.759528e-01
Vol_USD_ETH_LogRet	0.012978	9.092989e-01
Volume_ETH_LogRet	0.011821	9.134220e-01
BTC_720m_VolEMA/Vol	0.006807	9.342477e-01
ETH_360m_VolEMA_pctchange	0.003583	9.522702e-01
ETH_240m_VolEMA_pctchange	0.000646	9.797197e-01

*En orden de izquierda a derecha se muestra el nombre de la variable el valor F y el p-valor obtenido con la realización del test.

Tras la aplicación de las dos primeras técnicas de selección, el dataset se ha reducido a 63 variables, excluyendo el target, respecto de las 115 iniciales.

Por último, se ha aplicado un mecanismo de selección multivariante denominado ‘recursive feature elimination’, que determina el número de variables óptimo, ordenándolas además en función de su importancia, mediante un procedimiento iterativo en el que se busca el subconjunto de variables que optimizan el estimador que se seleccione, procediendo en cada iteración a eliminar la variable de menor relevancia y reajustando el modelo hasta maximizar el rendimiento del estimador.

Se ha realizado este proceso empleando como estimador un modelo de regresión lineal. El ranking de importancia de las variables resultante se expone a continuación. Todas aquellas que cuentan con ranking 1 son las seleccionadas por el método RFE, mientras que el ranking del número 2 en adelante determina el orden de importancia de aquellas variables que no han sido seleccionadas.

Tabla 4: Ránking de importancia de las distintas variables según el método RFE

Variable	Ranking RFE	Variable	Ranking RFE
BTC_15m_EMA/Close	1	ETH_5m_EMA/Close	26
Close_LTC_LogRet	1	ETH_60m_EMA_pctchange	27
BTC_15m_EMA_pctchange	1	ETH_60m_EMA/Close	28
BTC_240m_EMA_pctchange	1	BTC_5m_EMA/Close	29
BTC_LTC_ratio_LogRet	1	RealVar_BTC_15m	30
Close_BTC_LogRet	1	High_LTC_LogRet	31
BTC_60m_EMA_pctchange	1	Low_LTC_LogRet	32
BTC_120m_EMA_pctchange	1	Open_LTC_LogRet	33
BTC_360m_EMA_pctchange	2	High_ETH_LogRet	34
BTC_30m_EMA_pctchange	3	Close_ETH_LogRet	35
BTC_30m_EMA/Close	4	Low_ETH_LogRet	36
ETH_360m_EMA_pctchange	5	Open_BTC_LogRet	37
ETH_240m_EMA_pctchange	6	RealVar_BTC_60m	38
BTC_60m_EMA/Close	7	Close_XRP_LogRet	39
BTC_120m_EMA/Close	8	Vol_USD_BTC_LogRet	40
BTC_240m_EMA/Close	9	Volume_BTC_LogRet	41
BTC_360m_EMA/Close	10	BTC_XRP_ratio_LogRet	42
ETH_240m_EMA/Close	11	BTC_ETH_ratio_LogRet	43
ETH_360m_EMA/Close	12	Low_BTC_LogRet	44
BTC_720m_EMA_pctchange	13	Low_XRP_LogRet	45
BTC_1440m_EMA_pctchange	14	BTC_1440m_VolEMA_pctchange	46
ETH_720m_EMA_pctchange	15	BTC_720m_VolEMA_pctchange	47
BTC_720m_EMA/Close	16	BTC_360m_VolEMA_pctchange	48
ETH_720m_EMA/Close	17	High_XRP_LogRet	49
BTC_1440m_EMA/Close	18	RealVar_BTC_30m	50
ETH_120m_EMA_pctchange	19	High_BTC_LogRet	51
ETH_120m_EMA/Close	20	BTC_240m_VolEMA_pctchange	52

ETH_1440m_EMA_pctchange	21	BTC_120m_VolEMA_pctchange	53
ETH_1440m_EMA/Close	22	BTC_60m_VolEMA_pctchange	54
ETH_15m_EMA_pctchange	23	BTC_30m_VolEMA_pctchange	55
ETH_15m_EMA/Close	24	BTC_5m_VolEMA/Vol	56
ETH_5m_EMA_pctchange	25		

El modelo RFE ajustado por regresión lineal considera que únicamente han de emplearse 8 variables, lo que parece reducir excesivamente el número de inputs. Por ello se ha optado por comparar el resultado obtenido con 2 datasets distintos incorporando respectivamente las 20 y 30 variables de mayor ranking RFE. Se ha tratado de analizar también un dataset con 50 de las variables para determinar si la calidad del modelo se reducía de manera significativa por incorporar un número tan elevado de variables, pero no ha sido posible por motivos de capacidad computacional.

4.4. Arquitectura del modelo

4.4.1. División del dataset

El primer paso para garantizar la validez de los resultados del modelo, es la división de los datos en sets de entrenamiento y test, de manera que el modelo sea evaluado respecto de unos datos con los que no haya sido ajustado previamente.

Y es que la finalidad de los modelos de Machine Learning como apunta López de Prado (2021) es la de realizar predicciones respecto de características u observaciones desconocidas para el modelo. Lo contrario, la evaluación del modelo empleando datos con los que el mismo ha sido entrenado, llevaría al modelo a mostrar unos resultados ampliamente superiores a los reales además de una elevada velocidad de aprendizaje, sin que por ello el modelo sea extrapolable a situaciones reales de mercado.

También resulta esencial, como expone Chan (2021) que no se añadan nuevos parámetros o condiciones, ni se ajusten sus valores, para mejorar la predicción en los datos de test, pues en ese caso se estaría empleando el set de evaluación como set de entrenamiento de manera efectiva, dando lugar al sesgo de *data snooping*.

De esta manera se estaría reforzando de manera artificial el resultado del modelo, incluso respecto del set de evaluación, pero decrecería su validez efectiva y capacidad de generalizar a contextos reales de mercado.

Para llevar a cabo esta división, se ha empleado el paquete Scikit-learn, cuya función `train_test_split`² permite además de separar fácilmente un dataset en sets de entrenamiento y test, reordenar las observaciones de manera que el modelo no se beneficie de posibles tendencias para acelerar el aprendizaje, así como establecer una semilla para garantizar la replicabilidad del modelo.

Sin embargo, la reordenación no se ha incorporado en la separación de entrenamiento y evaluación, ya que es necesario proceder a la creación de los bloques de predicción antes de alterar el orden de las observaciones, sino en la validación cruzada.

Se ha optado por reservar un 20% del dataset, 8.902 observaciones, para la evaluación del modelo, empleándose el 80% restante de observaciones (35.607) para el entrenamiento y ajuste del modelo.

Esta distribución del dataset se encuentra en línea con lo recogido en la literatura. El tamaño del dataset suele situarse en el 20% (Kim & Kim, 2019; Mudassir et al., 2020; Moghar & Hamiche, 2020) con algunos autores empleando hasta el 30% (Cocco et al., 2021) del total de observaciones disponibles.

Otra parte de la literatura consultada sin embargo emplea una porción incluso inferior al 10% con modelos LSTM (Greff et al. 2017). Incluso, Li & Dai (2020) emplean a penas un 3,2% de sus observaciones para evaluar su modelo, lo que por lo menos *a priori* se antoja insuficiente.

4.4.2. Construcción de los bloques de datos

El objetivo del modelo que se pretende elaborar es la predicción, mediante los precios de cotización y otra serie de indicadores técnicos durante los ‘p’ minutos anteriores a la

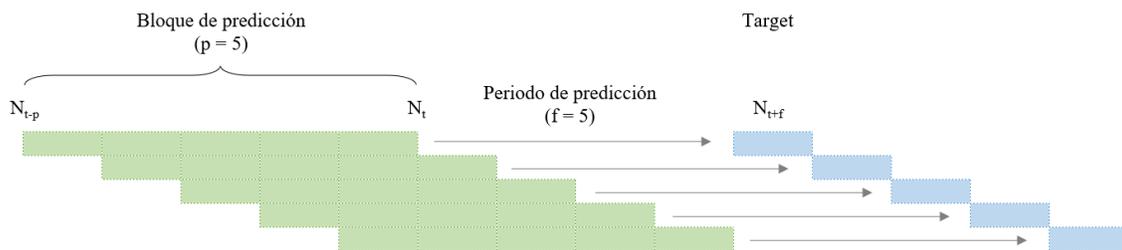
² Puede obtenerse información adicional sobre la misma en https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

observación actual en tiempo ‘t’, de la variación en el precio de cotización del Bitcoin a ‘f’ minutos vista (N_{t+f}).

Con independencia del periodo ‘p’ que se emplee para predecir, parámetro que será optimizado en el apartado 4.4.5, han de agruparse las observaciones durante dicho periodo, conformando un bloque de predicción al que ha de asignársele valor correspondiente de la variable target ‘f’ periodos en el futuro.

En la imagen que se proporciona a continuación puede observarse el funcionamiento de este proceso iterativo, donde cada línea verde representa un bloque, asumiendo un tamaño de 5 periodos a efectos meramente ilustrativos, y el bloque azul correspondiente es el valor real para dicho bloque, que será posteriormente comparado con la predicción que realice el modelo para determinar el rendimiento del mismo.

Ilustración 3: Esquema del procedimiento de construcción de los bloques



4.4.3. La validación cruzada para garantizar la robustez de los resultados obtenidos

Como indica López de Prado (2018, p.103) la finalidad de la validación cruzada o ‘cross-validation’, es “*determinar el error de generalización de un algoritmo de Machine Learning para evitar el overfitting*”.

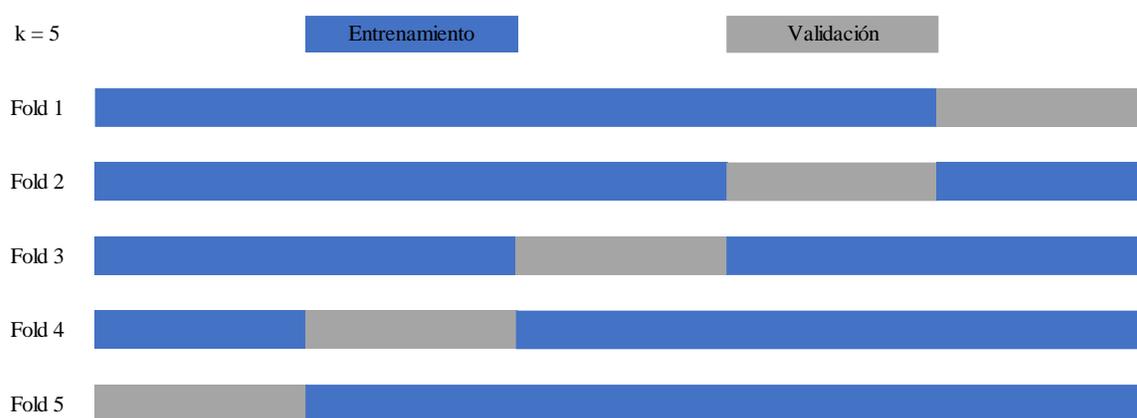
Una de las técnicas más afamadas dentro de la validación cruzada es la del ‘k-fold cross validation’ que consiste en reservar la proporción resultante de la operación $\frac{1}{k}$ del set de entrenamiento como set de validación para testear el modelo una vez ajustado, manteniéndose el restante $\frac{1-k}{k}$ por ciento de las observaciones para el entrenamiento del modelo (López de Prado, 2021).

Posteriormente, se procede a alternar las observaciones que conforman los sets de entrenamiento y validación, de manera que el modelo estaría siendo entrenado y validado en términos prácticos sobre k muestras distintas, lo que proporciona una aproximación mucho más precisa de su verdadera validez que una única iteración sobre los datos de entrenamiento.

El set de validación, actúa por tanto de manera similar al set de test o evaluación, pero permitiendo ajustar diversos parámetros del modelo sin contaminar el set de evaluación. Por ello, es importante que toda observación sea asignada alternativamente a uno sólo de los sets de entrenamiento o validación (López de Prado, 2021).

La Ilustración 4 proporciona un esquema del funcionamiento del k -fold cross validation para el $k = 5$ que se ha empleado para la realización del presente trabajo.

Ilustración 4: Funcionamiento del k -fold cross validation



De esta manera, el set de entrenamiento que como se había explicado anteriormente componía el 80% del total de observaciones, se divide a su vez en 5 pliegues de los cuales 1 se reserva para a validación. Por tanto, un 20% del set de entrenamiento, que equivale a un 16% de la muestra total o 7.121 observaciones, conforman el set de validación, reduciéndose el set de entrenamiento a 28.486 observaciones.

Entre las principales utilidades de la validación cruzada encontramos la optimización de parámetros que se estudiará en el apartado 4.4.5.

Las ventajas, y la necesidad, de la utilización del procedimiento de k-fold cross-validation para la optimización de los parámetros también ha sido plasmada en algunos de los estudios de mayor relevancia en el ámbito de la predicción del Bitcoin.

Cocco et al. (2021) que también emplean una validación cruzada con 5 pliegues ($k = 5$), afirman los autores (p.7) que “*según nuestro mejor saber, nuestro trabajo es el primer intento de predicción del precio del Bitcoin mediante la investigación de la mejor arquitectura mediante el denominado método de valoración cruzada k-fold*”.

Y es que lo cierto es que si bien encontramos trabajos anteriores en el mismo campo como el de Albariqi & Winarko (2020) donde también se hace uso de la validación cruzada, esta se limita a garantizar la validez de los resultados obtenidos y no para optimizar los parámetros.

Para concluir este apartado relativo a la validación cruzada, cabe resaltar que López de Prado (2021) expone en profundidad algunas de las limitaciones de esta técnica en el ámbito financiero, particularmente al emplearse en la fase de diseño del modelo.

El autor alude a la imposibilidad de asumir que las observaciones son aleatorias, independientes e idénticamente distribuidas (proceso I.I.D.) y la reutilización de un mismo set de validación como las principales causas de dichas limitaciones, y propone la utilización de una estrategia k-fold purgada eliminando aquellas observaciones que puedan verse duplicadas.

A pesar de lo acertado de su formulación, las dificultades para purgar la estrategia de validación cruzada k-fold y el hecho de que la superposición se produce en menos de un 1% de las observaciones que componen el set de validación y el set de evaluación, nos hacen decantarnos por aplicar la validación cruzada en la forma descrita en la Ilustración 4.

Sin embargo, y con vistas a posibles estudios futuros, esta sería una de las posibles mejoras que podrían realizarse respecto de la arquitectura que se diseña en el presente trabajo.

4.4.4. Arquitectura base de un modelo LSTM

Con carácter previo a la optimización de parámetros, resulta necesario comprender precisamente qué parámetros requieren ser optimizados en un modelo LSTM.

La base sobre la que se parte en este trabajo consiste en una única neurona LSTM, sometida a una tasa de dropout y a un proceso de Batch Normalization para minimizar las posibilidades de overfitting del modelo al set de entrenamiento, y una neurona Dense final con una sola unidad, ya que al ser la neurona de cierre del modelo es necesario que realice una única predicción, y con una función de activación lineal.

La función de activación no ha sido sometida a optimización ya que la otra alternativa, la función ReLu, únicamente regresa valores iguales o superiores a 0, algo que la hace muy popular para otro tipo de tareas (particularmente para problemas de clasificación) pero que imposibilita predecir movimientos negativos en el precio del Bitcoin (Ronaghan, 2018).

4.4.5. Optimización de los parámetros del modelo LSTM

La optimización de parámetros se ha realizado en base a la división entre los sets de entrenamiento y validación que se han descrito en el apartado 4.4.3, pero reduciéndose el tamaño de ambos sets por motivos de capacidad computacional. Para poder ejecutar el código necesario para proceder con este ‘hiperparameter tuning’ ha sido necesario reducir el tamaño de los sets de entrenamiento y validación a un 20% de su extensión original.

Trabajos de referencia en el campo predictivo de criptomonedas como los de Cocco et al. (2021) o Albariqi & Winarko (2020) recalcan la importancia de este proceso, optimizándose parámetros como el número de neuronas por capa, la tasa de aprendizaje, y el algoritmo de aprendizaje u optimizador.

Los parámetros que se procede a optimizar en el presente trabajo, junto con los valores que serán analizados para cada uno durante el proceso de optimización, se exponen en la siguiente tabla.

Tabla 5: Parámetros optimizados de la red LSTM

Variable a optimizar	Valores analizados
Unidades por neurona	64, 128, 256
Dropout rate	0,01; 0,1; 0,2
Learning rate	0,001; 0,01; 0,1
Optimizador	Adam, SGD, RMS
Batch size	32, 64, 128, 256

Tamaño del bloque de datos	10, 15, 30, 60, 120, 240
Escalado de datos	MinMax(0,1), MinMax(-1,1), StandardScaler, RobustScaler, QuantileTransformer
Número de épocas	-

Un aspecto a considerar a lo largo del presente apartado, es que la sensibilidad del modelo al valor de los parámetros, esto es, si cualquier valor distinto del óptimo reduce drásticamente su *'performance'*, es un claro indicio de la presencia de overfitting y del sesgo de *data snooping* antes mencionado (Chan, 2021).

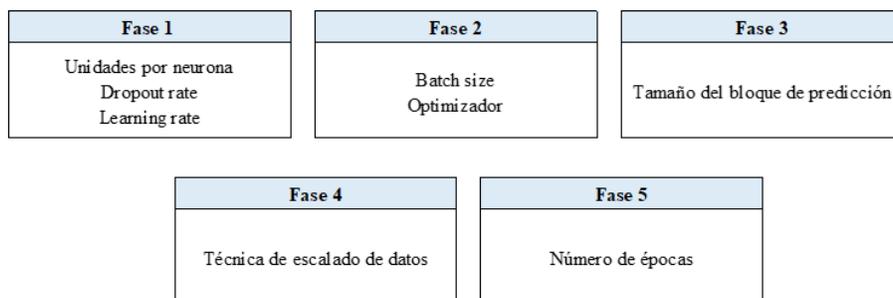
Originalmente, se pretendía optimizar todos los parámetros al mismo tiempo, iterando respecto de los distintos valores de cada uno de ellos. La ventaja de este método es doble ya que por un lado se incrementa el número de iteraciones para cada valor, lo que unido a la validación cruzada 5-fold proporciona unos resultados mucho más robustos acerca de la posible validez de cada arquitectura, y adicionalmente permite capturar la relación de los distintos parámetros entre sí en lugar de aislados.

La desventaja, sin embargo, es la capacidad computacional requerida para llevar a cabo un análisis de este tipo, incluso habiéndose limitado el tamaño de los sets de entrenamiento y validación. El total de combinaciones para los valores que se pretende analizar es de 9.721, cada una de las cuales será evaluada 5 veces mediante validación cruzada durante 10 épocas, resultando en un total de 486.050 iteraciones sobre un set de entrenamiento que cuenta con alrededor de 28.000 observaciones.

Ante la imposibilidad de llevar a cabo un análisis tan profundo, se ha optado por dividir la optimización en fases consecutivas, limitando el número de parámetros, y por tanto también de combinaciones, que se someten a optimización, pero permitiendo realizar un análisis multivariante en el que se recogen por lo menos parcialmente las relaciones de los distintos parámetros entre sí.

En cada fase, se seleccionará un único valor para cada parámetro analizado, en base a la pérdida media que se obtenga del proceso de validación cruzada 5-fold, empleándose como función de pérdida el error medio cuadrático. Las distintas fases junto con los parámetros optimizados en cada una de ellas se exponen en la siguiente Ilustración.

Ilustración 5: División en fases del proceso de optimización de parámetros



Para la primera fase, se establece un modelo base LSTM con 10 épocas, suficientes para optimizar en gran medida el modelo a expensas de la optimización posterior de este parámetro, con el que se busca optimizar los siguientes parámetros:

- **Unidades por capa:** representan la dimensión del estado oculto de la red, el número de parámetros que alimentan a la siguiente neurona ya que las redes neuronales son modelos ‘feed-forward’ (Karim, 2019). A más unidades, más capaz es la red neuronal de captar y modelar secuencias, pero, al mismo tiempo, también aumentan considerablemente las posibilidades de que el modelo sobreajuste. Dado que nuestro modelo incorpora una única neurona LSTM, este parámetro se refiere exclusivamente al número de unidades que componen esta capa.
- **Dropout rate:** se trata de un mecanismo de regularización que pretende evitar el overfitting en las redes neuronales desprendiéndose de la proporción que se determine mediante esta ratio de inputs de manera aleatoria durante la fase de entrenamiento (Srivastava et al. 2014).
- **Learning rate:** determina la velocidad a la que la red neuronal optimiza sus parámetros para adaptarse al problema que se le presenta. Tasas de aprendizaje amplias derivan en inestabilidad en los resultados obtenidos por la red y, probablemente, en overfitting, mientras que tasas excesivamente reducidas impiden optimizar el aprendizaje de la red neuronal (Brownlee, 2019).

La pérdida media para cada parámetro se expone en la tabla que se presenta a continuación. En azul se resalta el valor que minimiza la pérdida de cada parámetro, donde se observa

que los valores que minimizan la pérdida para cada parámetro son 128 unidades, un dropout de 0,2 y una tasa de aprendizaje de 0,01.

Tabla 6: Pérdida media por parámetro en la Fase 1

Uds. por capa	Pérdida media	Dropout Rate	Pérdida media	Learning Rate	Pérdida media
64	0,002048	0,01	0,002828	0,001	0,002348
128	0,001794	0,1	0,001871	0,01	0,002024
256	0,002701	0,2	0,001843	0,1	0,002171

Las diferencias resultan especialmente relevantes en el número de unidades por neurona, mientras que las diferencias en la tasa de aprendizaje son bastante reducidas.

En cuanto al learning rate, si bien las diferencias son reducidas, particularmente entre 0,01 y 0,1, se opta por 0,01 para evitar como se ha expuesto anteriormente utilizar tasas excesivamente elevadas o reducidas. Finalmente, respecto del dropout rate, a pesar de la casi nula diferencia entre los valores de 0,1 y 0,2 se opta por este último para minimizar las posibilidades de overfitting del modelo.

También Albariqi & Winarko (2020) concluyeron que la tasa de aprendizaje óptima era 0,01, si bien emplearon un dropout de 0.5 en su modelo RNN que, cabe destacar, seguía una arquitectura simple y no LSTM como en el presente trabajo.

En la fase 2, los parámetros optimizados han sido:

- **Batch Size:** representa el número de observaciones con los que entrena el modelo en cada iteración antes de actualizar sus pesos y parámetros, de manera que a mayor batch size, menos iteraciones habrán de realizarse en cada época para completar el set de entrenamiento o validación (Brownlee, 2018).
- **Optimizador:** distintos algoritmos de aprendizaje en base a los cuales el modelo actualiza los pesos de cada atributo del modelo con vistas a minimizar la pérdida.

Los resultados, que se proporcionan en la Tabla 7, muestran como parámetros óptimos el optimizador Adam y un batch size de 32 unidades. De nuevo, en el **Error! Reference source not found.** se proporciona un desglose de todas las arquitecturas analizadas.

Tabla 7: Pérdida media por parámetro en la Fase 2

Optimizador	Pérdida media	Batch Size	Pérdida media
SGD	0,005378	32	0,003096
RMS	0,004854	64	0,003839
Adam	0,001969	128	0,003725
		256	0,005609

Estos resultados pueden considerarse esperados y acordes al conocimiento presente en la literatura. Y es que son varios los trabajos en los que de entre los principales algoritmos de aprendizaje, es el optimizador Adam el que minimiza la función de pérdida (Albariqi & Winarko, 2020; Mudassir et al., 2020).

Respecto del batch size, también resultaría lógica la elección de un valor menor ya que esto implicaría que la red neuronal se actualiza con mayor frecuencia y, por ende, tiene una mayor capacidad de optimización.

Sin embargo, analizando la pérdida media obtenida con el optimizador Adam para cada batch size (Tabla 8), observamos como la elección de 32 unidades parece equivocada. De hecho, el batch size de 32 presenta la mayor pérdida, y de manera significativa de entre todos los valores analizados del parámetro al combinarlo con el optimizador Adam.

Tabla 8: Comparativa de las arquitecturas basadas en el optimizador Adam.

Optimizador	Batch size	Pérdida media
Adam	32	0,002790
Adam	64	0,001507
Adam	128	0,001614
Adam	256	0,001966

Dada la amplia superioridad del optimizador Adam respecto a las alternativas SGD y RMS, se considera oportuno incorporar al modelo el batch size que minimice la pérdida de este optimizador, por lo que se seleccionan 64 unidades en lugar de 32 para incorporar a la arquitectura final del modelo.

En la tercera fase, se optimiza el tamaño del bloque, es decir, el número de observaciones en el pasado (*'backsteps'*) que se emplean para predecir la variable objetivo. El racional

para proceder a la optimización de este parámetro, es que si bien a mayor tamaño de bloque más información se proporciona para optimizar la red neuronal, emplear plazos excesivamente prolongados también podría introducir ruido en el modelo deteriorando su capacidad predictiva.

Los resultados obtenidos muestran el razonamiento anterior a la perfección, y es que el periodo de 60 observaciones, muestra una pérdida notablemente inferior a la de los periodos menos prolongados. A partir de las 60 observaciones se produce el efecto contrario, incrementándose de manera significativa la pérdida, por lo que se seleccionan 60 observaciones como el tamaño de bloque óptimo.

Tabla 9: Resultados del proceso de optimización del tamaño de bloque.

Backsteps	Pérdida media
10	0,002801
15	0,001960
30	0,001922
60	0,001474
120	0,002595
240	0,003220

A continuación, se procede a comparar distintas técnicas de escalado de datos. Ya anteriormente se ha hecho alusión a la importancia de que todas las variables se encuentren en un rango de valores similares para evitar que el modelo ‘sobrepondere’ algunas de ellas basándose exclusivamente en lo elevado de sus valores. El propio paquete sklearn contiene diversas herramientas de escalado, además de una detallada descripción del funcionamiento de cada una de ellas en su documentación.

La técnica de escalado por excelencia es la de MinMax Scaling que consiste en dividir el valor de cada variable por el valor máximo de la misma, de manera que todos los valores queden contenidos en el rango (0,1). De manera alternativa, cabría realizar un escalado en el que los datos queden contenidos en el rango (-1,1). Estas alternativas pueden ser denotadas respectivamente como MinMax(0,1) y MinMax(-1,1).

Sin embargo, existen otras alternativas al MinMax Scaling, particularmente la estandarización, que supone la transformación de los valores en una distribución de media 0 y desviación típica 1, o incluso la uniformización, en la que además de recogerse todos los valores en el rango (0,1) estos siguen una distribución normal (Coqueret & Guida, 2020). Incluso, la función QuantileTransformer de sklearn permite transformar un set de datos en una distribución gaussiana.

Mudassir et al. (2020) por su parte, proceden con un preprocesamiento dual en el que el método MinMax (0,1) viene precedido de la aplicación de la función RobustScaling.

Si bien para el presente trabajo se considera oportuno proceder con una única instancia de escalado, se ha procedido a comparar la pérdida obtenida con distintas funciones de escalado.

Como puede apreciarse, ambas modalidades de MinMax Scaling se muestran claramente superiores y entre estas, es el MinMax(-1,1) la técnica que minimiza la función de pérdida.

Tabla 10: Comparativa de distintas técnicas de escalado de datos.

Técnica de escalado	Pérdida media
MinMax(0,1)	0,002170
MinMax(-1,1)	0,001688
StandardScaler	0,003640
RobustScaler	0,004921
QuantileTransform	0,003212

En todas las arquitecturas analizadas, el escalado de los datos se ha realizado en base al set de entrenamiento exclusivamente. Jiang (2020) demuestra como la normalización de los datos por ventanas en lugar de empleando la totalidad de los datos deriva en un rendimiento claramente superior de los modelos predictivos, con independencia de la arquitectura empleada.

Y es que si normalizásemos los datos de entrenamiento con la totalidad de los datos disponibles, es decir, con el conjunto de entrenamiento y de test, estaríamos de manera efectiva alimentando al modelo con datos a los que no debería de tener acceso por estar reservados a la validación, cometiendo lo que Chan (2021) denomina ‘*look-ahead bias*’.

Finalmente, se reserva la quinta y última fase de optimización para la determinación del número de épocas óptimo. Este parámetro representa las iteraciones que realiza el modelo sobre el dataset, es decir, el número de veces que el modelo entrena sobre la totalidad de observaciones que componen el set de entrenamiento (Brownlee, 2019).

A diferencia de los parámetros anteriores, para el número de épocas no se procede a iterar el mismo modelo variando el número de épocas. En su lugar, se ejecuta el modelo una única vez durante 50 épocas y posteriormente se analiza la evolución de la función de pérdida en evaluación para determinar el valor óptimo de este parámetro.

En el gráfico que se expone a continuación puede observarse la evolución de la función de pérdida, los errores medios cuadráticos como se vienen empleando hasta el momento, a lo largo de las 50 épocas.

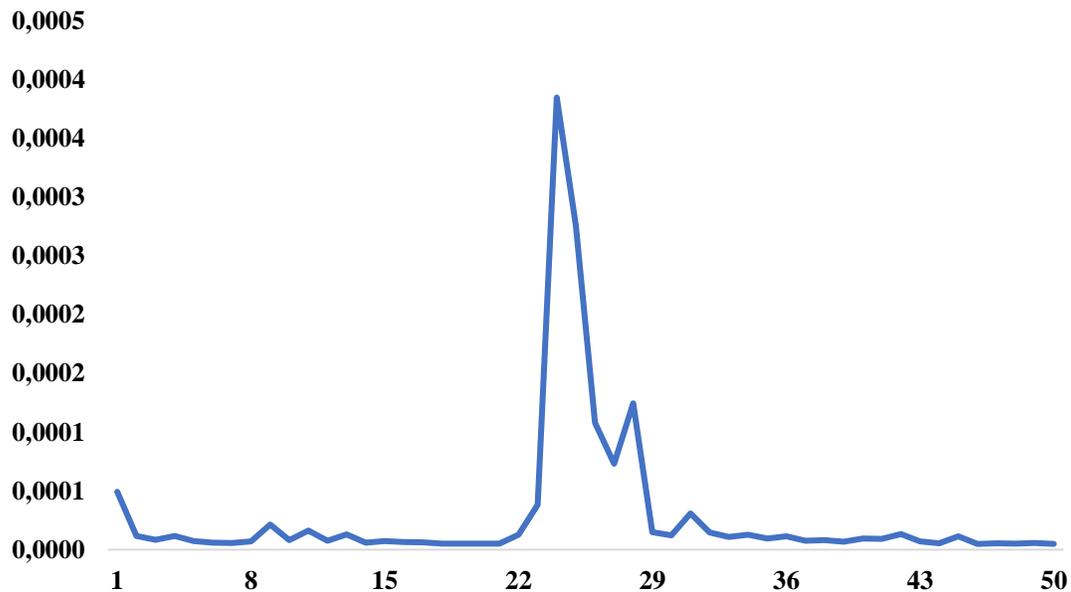
Tabla 11: Evolución de la función de pérdida en validación y entrenamiento durante 50 épocas

Época	Pérdida en entrenamiento	Pérdida en validación	Época	Pérdida en entrenamiento	Pérdida en validación
1	0,049785	0,000049	26	0,001143	0,000108
2	0,000145	0,000012	27	0,000122	0,000073
3	0,000052	0,000008	28	0,000258	0,000124
4	0,000031	0,000012	29	0,000420	0,000015
5	0,000028	0,000007	30	0,000325	0,000012
6	0,000024	0,000006	31	0,000087	0,000031
7	0,000021	0,000006	32	0,000074	0,000015
8	0,000022	0,000007	33	0,000052	0,000011
9	0,000022	0,000021	34	0,000038	0,000013
10	0,000023	0,000008	35	0,000044	0,000010
11	0,000018	0,000016	36	0,000026	0,000011
12	0,000025	0,000008	37	0,000019	0,000008
13	0,000014	0,000013	38	0,000014	0,000008
14	0,000017	0,000006	39	0,000038	0,000007
15	0,000013	0,000007	40	0,000018	0,000010
16	0,000015	0,000007	41	0,000015	0,000009
17	0,000011	0,000006	42	0,000011	0,000013
18	0,000014	0,000005	43	0,000012	0,000007
19	0,000010	0,000005	44	0,000010	0,000005
20	0,000012	0,000005	45	0,000013	0,000011
21	0,000010	0,000005	46	0,000010	0,000005
22	0,000017	0,000013	47	0,000024	0,000005
23	0,000260	0,000038	48	0,000021	0,000005
24	0,000952	0,000385	49	0,000011	0,000006
25	0,002750	0,000276	50	0,000032	0,000005

Las distintas tonalidades de azul y verde (cuanto más oscuro, menor pérdida) indican las 3 épocas con menor pérdida en entrenamiento y validación respectivamente. Centrándonos en esta última, observamos como es precisamente la época 50 la que minimiza el error medio cuadrático.

Sin embargo, se observa como en la decimonovena época la pérdida de validación se sitúa muy cercana a la obtenida en la época 50, antes de un inusual pico que puede observarse en el siguiente gráfico.

Gráfico 1: Evolución de la pérdida en validación a lo largo de 50 épocas



Teniendo en cuenta lo anterior, y atendiendo al incremento de la presión computacional a medida que se incrementa el número de épocas, se opta por seleccionar 20 épocas para entrenar el modelo final.

Recapitulando lo establecido en el presente apartado, la arquitectura óptima para el modelo LSTM tendría los siguientes valores para los distintos parámetros.

Tabla 12: Resumen de los valores óptimos para cada parámetro

Variable a optimizar	Valor óptimo
Unidades por neurona	128
Dropout rate	0,2
Learning rate	0,01
Optimizador	Adam
Batch size	64
Tamaño del bloque de datos	60
Escalado de datos	MinMax(-1,1)
Número de épocas	20

5. INTERPRETACIÓN DE LOS RESULTADOS

5.1. Descripción de los modelos de referencia

Para determinar realmente la capacidad predictiva de la red LSTM, no es suficiente con calcular su función de pérdida, ni si quiera compararla con otras arquitecturas de red neuronal. Es necesario comparar los resultados de estos modelos con otros modelos base para realmente determinar si el modelo que se ha construido cuenta realmente con una capacidad predictiva.

Para el presente trabajo, se han seleccionado 4 modelos base o de referencia.

- El primero de ellos es un modelo construido bajo la hipótesis del random walk en el que la mejor predicción para la cotización futura, es la cotización actual. Por tanto, este modelo predice para todas y cada una de las observaciones del set de evaluación el valor 0 que representa un retorno nulo, al no variar el precio respecto de la observación anterior.
- El segundo se ha construido como una media 'naïve', de manera que su predicción para cada observación será el retorno medio obtenido en el set de entrenamiento. Esta referencia se fundamenta en lo establecido por Greaves & Au (2015).
- Por último, se ha construido un modelo de regresión lineal.

5.2. Comparativa de resultados entre el modelo LSTM y los modelos de referencia

Se procede a continuación a realizar una comparativa entre los distintos modelos analizados para los datasets con 10, 20 y 30 variables. Las redes neuronales presentadas emplean como función de pérdida el error medio cuadrático al tratarse de un problema de regresión. Sin embargo, para determinar el rendimiento de los distintos modelos se proporciona el error medio (ME), la raíz de los errores medios cuadráticos (RMSE) y el error absoluto medio (MAE).

A este respecto, Mudassir et al. (2020) puntualizan que es común en la literatura emplear la transformación logarítmica de los precios y que ello resulta engañoso por resultar inferiores a las métricas que se obtendrían empleando los precios reales.

En lo que al presente trabajo respecta, las métricas empleadas en todos los modelos se calculan respecto del retorno logarítmico del precio de cotización del Bitcoin por lo que la comparación es homogénea y, por ende, válida a efectos de determinar la capacidad predictiva del modelo LSTM desarrollado.

Se expone a continuación una tabla resumen de las distintas métricas para cada una de las arquitecturas. En azul se resalta la arquitectura de todos los modelos con menor error en cada métrica, y en verde la arquitectura de RNN que minimiza el error en cada dataset.

Tabla 13: Comparativa de los resultados obtenidos en el set de evaluación por las distintas arquitecturas

Arquitectura	ME	RMSE	MAE
Variación nula (20)	-0,000031	0,002236	0,001479
Variación nula (30)	-0,000031	0,002236	0,001479
Naïve mean (20)	-0,000013	0,002236	0,001478
Naïve mean (30)	-0,000013	0,002236	0,001478
Regresión lineal (20)	-0,00642	0,061574	0,002133
Regresión lineal (30)	-0,006545	0,614392	0,008012
LSTM (20)	-0,000033	0,002257	0,001500
LSTM (30)	-0,00376	0,002414	0,001671
GRU (20)	0,001595	0,002791	0,002125
GRU (30)	-0,000505	0,002372	0,001640
Simple RNN (20)	0,002479	0,003546	0,002820
Simple RNN (30)	0,029242	0,031784	0,031220

Se observa como el modelo LSTM muestra mayor capacidad predictiva que las restantes redes neuronales, como cabía esperar ya que se ha optimizado esta arquitectura, pero sin embargo muestra peor capacidad predictiva que los modelos de referencia de variación nula y de naïve mean.

Adicionalmente, los resultados con el dataset de 20 variables mejoran a los obtenidos con 30 variables, reforzando la importancia de seleccionar las variables adecuadas.

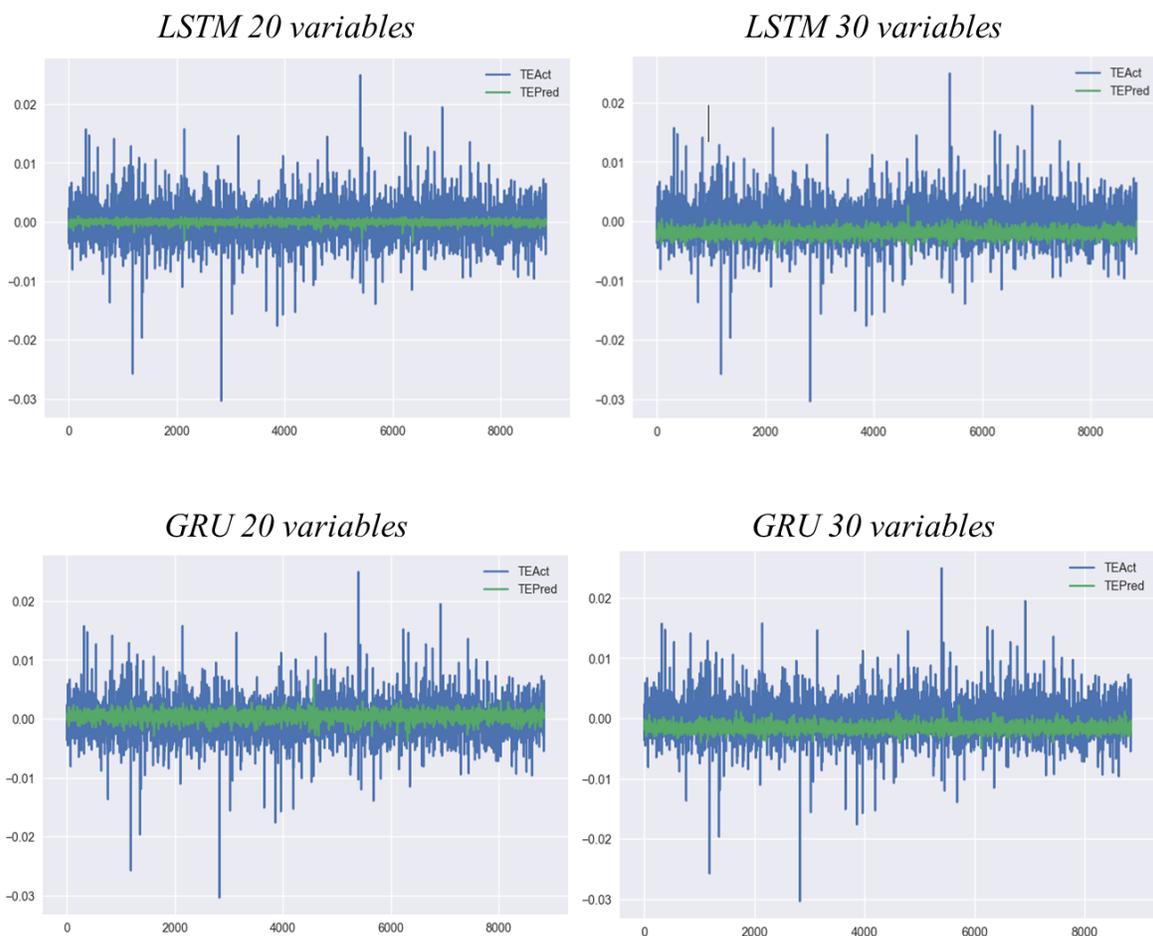
Destacar que los resultados entre estos dos modelos de referencia son prácticamente idénticos entre sí y el retorno logarítmico medio del Bitcoin en el set de entrenamiento es

de $-1.9e-05$, por lo que es un valor que se aproxima mucho al 0 con el que se modela la variación nula. Además, sus resultados no varían entre datasets ya que se calculan exclusivamente con la variable target.

También podemos concluir en base a los datos antes expuestos que la regresión lineal y la SimpleRNN no son modelos adecuados para la predicción del Bitcoin, presentando un nivel de error ampliamente superior al de las restantes arquitecturas.

Analizando las predicciones realizadas por el modelo LSTM, con independencia de utilizarse el dataset con 10, 20 o 30 variables, se observa como estas se encuentran en un rango mucho más limitado que el de las observaciones reales.

Gráfico 2: Visualización de la comparativa entre los valores reales y los predichos por las redes neuronales



Ello deriva necesariamente en un error notable y generalizado a lo largo de la totalidad del set de evaluación. De hecho, es prácticamente imposible que con predicciones de este tipo la red LSTM supere las arquitecturas de referencia pues en la práctica, está prediciendo prácticamente un mismo valor de manera continua al igual que los modelos base basados de variación nula o naïve mean.

Adicionalmente, el hecho de que la arquitectura GRU muestre la misma problemática, a pesar de predecir un rango ligeramente más amplio, que el LSTM puede indicar que hay algún elemento dentro de la arquitectura del modelo que esté provocando esta situación.

En definitiva, los modelos de redes neuronales, particularmente la arquitectura LSTM, no arrojan los resultados esperados e, incluso, muestran una capacidad predictiva inferior a la teoría del random walk

6. CONCLUSIONES Y RECOMENDACIONES METODOLÓGICAS PARA FUTUROS TRABAJOS

Con los resultados obtenidos aplicando las distintas arquitecturas de red neuronal recurrentes, no podemos afirmar una capacidad predictiva superior a la de distintos modelos de referencia incluidos la teoría del random walk o una media naïve.

Ello se debe a que como se ha descubierto al analizar la representación gráfica de las predicciones realizadas por la red LSTM, sobre la que se ha focalizado la presente investigación, que este modelo predice un rango muy limitado de valores. Puede incluso decirse que en la práctica el método predictivo sería similar al de estos modelos de referencia ya que la predicción es casi siempre un valor muy cercano a 0, que es precisamente cómo se construyó el modelo de variación nula.

Como se expuso al principio de este documento, no se pretendía ni esperaba obtener un modelo ampliamente superior a la base, ya que ni si quiera practicantes mucho más experimentados lo han logrado, particularmente tratando la predicción del Bitcoin como regresión, pero los resultados resultan algo decepcionantes.

Entre las posibles causas, puede que utilizar datos de alta frecuencia no sea la opción óptima, especialmente en un mercado tan volátil como el de las Criptomonedas. Adicionalmente, se ha expuesto a lo largo del trabajo en reiteradas ocasiones que la metodología que se pretendía inicialmente ha debido de ser adaptada para hacer frente a las limitaciones computacionales con las que se contaba, siendo este sin duda el principal escollo encontrado durante la realización del proyecto.

Por ello, con vistas a posibles trabajos futuros, sería interesante determinar si por ejemplo la utilización de un número superior de datos mejoraría la capacidad predictiva del modelo LSTM. Cabe recordar que únicamente se han podido emplear alrededor del 2% de las observaciones disponibles ante la imposibilidad de ejecutar el código con series temporales más prolongadas.

A su vez, también sería recomendable realizar el proceso iterativo completo para la optimización de parámetros en lugar de dividirla por fases, y es que es esta la manera de determinar realmente y con mayor precisión la arquitectura óptima del modelo.

Adicionalmente, hay numerosos indicadores técnicos que no han sido incorporados al presente trabajo, en parte por las ya de por sí dificultades para tratar los datos disponibles. Entre estos cabe mencionar algunas de las técnicas empleadas con mayor asiduidad como las bandas de Bollinger. Y en este sentido otro posible aspecto metodológico a perfeccionar sería el de la selección de variables mediante procesos iterativos multivariantes sin necesidad de aligerar previamente la muestra mediante técnicas univariantes más sencillas, pero menos efectivas.

En definitiva, lo decepcionante de los resultados atendiendo a las expectativas depositadas por la literatura en la capacidad predictiva de las redes neuronales no supone en modo alguno una negación de su potencial, sino la reafirmación que su optimización y ajuste es sumamente complicado.

7. BIBLIOGRAFÍA

- Adebiyi, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction. *Journal of Applied Mathematics*, 2014, 1–7. <https://doi.org/10.1155/2014/614342>
- Adewuyi, A. W. (2016). Modelling Stock Prices with Exponential Weighted Moving Average (EWMA). *Journal of Mathematical Finance*, 06(01), 99–104. <https://doi.org/10.4236/jmf.2016.61011>
- Albariqi, R., & Winarko, E. (2020). Prediction of Bitcoin Price Change using Neural Networks. 2020 International Conference on Smart Technology and Applications (ICoSTA). <https://doi.org/10.1109/icosta48221.2020.1570610936>
- AYGUN, B., & KABAKCI GUNAY, E. (2021). Comparison of Statistical and Machine Learning Algorithms for Forecasting Daily Bitcoin Returns. *European Journal of Science and Technology*, 21, 444–454. <https://doi.org/10.31590/ejosat.822153>
- Bailey, D. H., & López De Prado, M. (2021). Finance is Not Excused: Why Finance Should Not Flout Basic Principles of Statistics. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3895330>
- Bergsli, L. V., Lind, A. F., Molnár, P., & Polasik, M. (2022). Forecasting volatility of Bitcoin. *Research in International Business and Finance*, 59, 101540. <https://doi.org/10.1016/j.ribaf.2021.101540>
- Brownlee, J. (2018, 20 julio). Difference Between a Batch and an Epoch in a Neural Network. *Machine Learning Mastery*. Recuperado 9 de junio de 2022, de <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/>
- Brownlee, J. (2019, 25 enero). Understand the Impact of Learning Rate on Neural Network Performance. *Machine Learning Mastery*. Recuperado 9 de junio de 2022, de <https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/>

- Cao, L., & Tay, F. (2003). Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on Neural Networks*, 14(6), 1506–1518. <https://doi.org/10.1109/tnn.2003.820556>
- Caporale, G. M., & Zekokh, T. (2019). Modelling volatility of cryptocurrencies using Markov-Switching GARCH models. *Research in International Business and Finance*, 48, 143–155. <https://doi.org/10.1016/j.ribaf.2018.12.009>
- Chan, E. P. (2021). *Quantitative Trading: How to Build Your Own Algorithmic Trading Business* (2nd ed.). Wiley.
- Chen, Z., Li, C., & Sun, W. (2020). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics*, 365, 112395. <https://doi.org/10.1016/j.cam.2019.112395>
- Cocco, L., Tonelli, R., & Marchesi, M. (2021). Predictions of bitcoin prices through machine learning based frameworks. *PeerJ Computer Science*, 7, e413. <https://doi.org/10.7717/peerj-cs.413>
- Coqueret, G., & Guida, T. (2020). *Machine Learning for Factor Investing: R Version*. CRC Press.
- Deshmukh, Y., Saratkar, D., & Tiwari, Y. (2019). Stock Market Prediction Using Machine Learning. *IJARCCCE*, 8(1), 31–35. <https://doi.org/10.17148/ijarccce.2019.8107>
- Dey, P., Hossain, E., Hossain, M. I., Chowdhury, M. A., Alam, M. S., Hossain, M. S., & Andersson, K. (2021). Comparative Analysis of Recurrent Neural Networks in Stock Price Prediction for Different Frequency Domains. *Algorithms*, 14(8), 251. <https://doi.org/10.3390/a14080251>
- Doshi, S. (2019, 13 enero). Various Optimization Algorithms For Training Neural Network. *Towards Data Science*. Recuperado 9 de junio de 2022, de <https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6>
- Du, K., & Swamy, M. N. S. (2019). *Neural Networks and Statistical Learning* (2nd 2019 ed.). Springer.

- Ersan, D., Nishioka, C., & Scherp, A. (2019). Comparison of machine learning methods for financial time series forecasting at the examples of over 10 years of daily and hourly data of DAX 30 and S&P 500. *Journal of Computational Social Science*, 3(1), 103–133. <https://doi.org/10.1007/s42001-019-00057-5>
- Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*, 25(2), 383–417. <https://doi.org/10.2307/2325486>
- Fatima, S., & Hussain, G. (2008). Statistical models of KSE100 index using hybrid financial systems. *Neurocomputing*, 71(13–15), 2742–2746. <https://doi.org/10.1016/j.neucom.2007.11.044>
- Gheyas, I., & Smith, L. (2009). A neural network approach to time series forecasting. *Lecture Notes in Engineering and Computer Science*, 2177. http://iaeng.org/publication/WCE2009/WCE2009_pp1292-1296.pdf
- Greaves, A., & Au, B. (2015, diciembre). Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin. http://snap.stanford.edu/class/cs224w-2015/projects_2015/Using_the_Bitcoin_Transaction_Graph_to_Predict_the_Price_of_Bitcoin.pdf
- Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), 2222–2232. <https://doi.org/10.1109/tnnls.2016.2582924>
- Harvey, C. (Ed.). (2021). Foreword - Why Is Systematic Investing Important? *Journal of Systematic Investing*, 1(1), i-iv. <https://doi.org/10.52354/jsi.1.1.i>
- Hoang Hung, N. (2016). Various moving average convergence divergence trading strategies: a comparison. *Investment Management and Financial Innovations*, 13(2), 363–369. [https://doi.org/10.21511/imfi.13\(2-2\).2016.11](https://doi.org/10.21511/imfi.13(2-2).2016.11)
- Huang, J. Z., Huang, W., & Ni, J. (2019). Predicting bitcoin returns using high-dimensional technical indicators. *The Journal of Finance and Data Science*, 5(3), 140–155. <https://doi.org/10.1016/j.jfds.2018.10.001>

- Jang, H., & Lee, J. (2018). An Empirical Study on Modeling and Prediction of Bitcoin Prices With Bayesian Neural Networks Based on Blockchain Information. *IEEE Access*, 6, 5427–5437. <https://doi.org/10.1109/access.2017.2779181>
- Jiang, X. (2020). Bitcoin Price Prediction Based on Deep Learning Methods. *Journal of Mathematical Finance*, 10(01), 132–139. <https://doi.org/10.4236/jmf.2020.101009>
- Kamalov, F., Gurrib, I., & Rajab, K. (2021). Financial Forecasting with Machine Learning: Price Vs Return. *Journal of Computer Science*, 17(3), 251–264. <https://doi.org/10.3844/jcssp.2021.251.264>
- Kang, B. K. (2021). Improving MACD Technical Analysis by Optimizing Parameters and Modifying Trading Rules: Evidence from the Japanese Nikkei 225 Futures Market. *Journal of Risk and Financial Management*, 14(1), 37. <https://doi.org/10.3390/jrfm14010037>
- Karim, R. (2019, 21 enero). Counting No. of Parameters in Deep Learning Models by Hand. *Towards Data Science*. Recuperado 9 de junio de 2022, de <https://towardsdatascience.com/counting-no-of-parameters-in-deep-learning-models-by-hand-8f1716241889#192e>
- Kavitha, G., Udhayakumar, A., & Nagarajan, D. (2013). Stock Market Trend Analysis Using Hidden Markov Models. <https://arxiv.org/ftp/arxiv/papers/1311/1311.4771.pdf>
- Kim, T., & Kim, H. Y. (2019). Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data. *PLOS ONE*, 14(2), e0212320. <https://doi.org/10.1371/journal.pone.0212320>
- Kimoto, T., Asakawa, K., Yoda, M., & Takeoka, M. (1990). Stock market prediction system with modular neural networks. *International Joint Conference on Neural Networks*, 1, 1–6. <http://web.ist.utl.pt/~adriano.simoese/tese/referencias/Papers%20-%20Adriano/NN.pdf>
- Lee, W. M. (2021, 18 octubre). Statistics in Python - Using ANOVA for feature selection. *Towards Data Science*. Recuperado 10 de junio de 2022, de <https://towardsdatascience.com/tagged/f-test>

- Li, Y., & Dai, W. (2020). Bitcoin price forecasting method based on CNN-LSTM hybrid neural network model. *The Journal of Engineering*, 2020(13), 344–347. <https://doi.org/10.1049/joe.2019.1203>
- Li, Y., & Zhang, L. (2021). Modelling and Forecasting Cyts Stock Prices Using Garch Model. *Frontiers in Economics and Management Research*, 32–37. <https://taylorandwales.com/uploads/jpapers/202104/fogR8BZAdII3wZNnzJWeh3CV69OTnrKe7odMOnWO.pdf>
- Lien, K. (2022, 28 marzo). The 5-Minute Trading Strategy. Investopedia. [https://www.investopedia.com/articles/forex/08/five-minute-momo.asp#:~:text=First%2C%20traders%20lay%20on%20two,average%20convergence%20divergence%20\(MACD\)](https://www.investopedia.com/articles/forex/08/five-minute-momo.asp#:~:text=First%2C%20traders%20lay%20on%20two,average%20convergence%20divergence%20(MACD)).
- Liu, M., Huo, J., Wu, Y., & Wu, J. (2021). Stock Market Trend Analysis Using Hidden Markov Model and Long Short Term Memory. <https://arxiv.org/pdf/2104.09700.pdf>
- Lo, A. W., & MacKinlay, A. C. (1988). Stock Market Prices Do Not Follow Random Walks: Evidence from a Simple Specification Test. *Review of Financial Studies*, 1(1), 41–66. <https://doi.org/10.1093/rfs/1.1.41>
- López De Prado, M. (2018). *Advances in Financial Machine Learning (Illustrated ed.)*. Wiley.
- Lopez De Prado, M., & Lewis, M. J. (2018). Detection of False Investment Strategies Using Unsupervised Learning Methods. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3167017>
- M, V., Pandit, A., Jindal, B., Kumari, M., & Singh, R. (2021). BITCOIN PRICE PREDICTION USING MACHINE LEARNING. *International Journal of Engineering Technologies and Management Research*, 8(5), 20–28. <https://doi.org/10.29121/ijetmr.v8.i5.2021.953>
- Madan, I., Saluja, S., & Zhao, A. (2014). Automated Bitcoin Trading via Machine Learning Algorithms. <http://cs229.stanford.edu/proj2014/Isaac%20Madan,%20Shaurya%20Saluja,%20Aojia>

%20Zhao,Automated%20Bitcoin%20Trading%20via%20Machine%20Learning%20Algorithms.pdf

- Mallikarjuna, M., & Rao, R. P. (2019). Evaluation of forecasting methods from selected stock market returns. *Financial Innovation*, 5(40). <https://doi.org/10.1186/s40854-019-0157-x>
- Moghar, A., & Hamiche, M. (2020). Stock Market Prediction Using LSTM Recurrent Neural Network. *Procedia Computer Science*, 170, 1168–1173. <https://doi.org/10.1016/j.procs.2020.03.049>
- Mudassir, M., Bennbaia, S., Unal, D., & Hammoudeh, M. (2020). Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach. *Neural Computing and Applications*. <https://doi.org/10.1007/s00521-020-05129-6>
- Munim, Z. H., Shakil, M. H., & Alon, I. (2019). Next-Day Bitcoin Price Forecast. *Journal of Risk and Financial Management*, 12(2), 103. <https://doi.org/10.3390/jrfm12020103>
- Naik, N., & Mohan, B. R. (2021). Stock Price Volatility Estimation Using Regime Switching Technique-Empirical Study on the Indian Stock Market. *Mathematics*, 9(14), 1595. <https://doi.org/10.3390/math9141595>
- Panigrahi, A. K., Vachhani, K., & Chaudhury, S. K. (2021). Trend identification with the relative strength index (RSI) technical indicator –A conceptual study. *Journal of Management Research and Analysis*, 8(4), 159–169. <https://doi.org/10.18231/j.jmra.2021.033>
- Remus, W., & O'Connor, M. (2001). Neural Networks for Time-Series Forecasting. *International Series in Operations Research & Management Science*, 245–256. https://doi.org/10.1007/978-0-306-47630-3_12
- Ronaghan, S. (2018, 16 julio). Deep Learning: Which Loss and Activation Functions should I use? Towards Data Science. Recuperado 8 de junio de 2022, de <https://towardsdatascience.com/deep-learning-which-loss-and-activation-functions-should-i-use-ac02f1c56aa8>

- Rouf, N., Malik, M. B., Arif, T., Sharma, S., Singh, S., Aich, S., & Kim, H. C. (2021). Stock Market Prediction Using Machine Learning Techniques: A Decade Survey on Methodologies, Recent Developments, and Future Directions. *Electronics*, 10(21), 2717. <https://doi.org/10.3390/electronics10212717>
- Selvamuthu, D., Kumar, V., & Mishra, A. (2019). Indian stock market prediction using artificial neural networks on tick data. *Financial Innovation*, 5(1). <https://doi.org/10.1186/s40854-019-0131-7>
- Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 1643–1647. <https://doi.org/10.1109/icacci.2017.8126078>
- Sen, J. (Ed.). (2021). *Machine Learning - Algorithms, Models and Applications*. IntechOpen. <https://doi.org/10.5772/intechopen.94615>
- Sharma, S., & Kaushik, B. (2017). Quantitative Analysis of Stock Market Prediction for Accurate Investment Decisions in Future. *Journal of Artificial Intelligence*, 11(1), 48–54. <https://doi.org/10.3923/jai.2018.48.54>
- Shen, J., & Shafiq, M. O. (2020). Short-term stock market price trend prediction using a comprehensive deep learning system. *Journal of Big Data*, 7(1). <https://doi.org/10.1186/s40537-020-00333-6>
- Simonian, J., Wu, C., Itano, D., & Narayanam, V. (2019). A Machine Learning Approach to Risk Factors: A Case Study Using the Fama–French–Carhart Model. *The Journal of Financial Data Science*, 1(1), 32–44. <https://doi.org/10.3905/jfds.2019.1.032>
- Sin, E., & Wang, L. (2017). Bitcoin price prediction using ensembles of neural networks. 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 666–671. <https://doi.org/10.1109/fskd.2017.8393351>

- Siripurapu, A. (2015). Convolutional Networks for Stock Trading. <https://www.semanticscholar.org/paper/Convolutional-Networks-for-Stock-Trading-Siripurapu-Stanford/86afcb8f6c9e182d79a25071371e6b567619b6dc>
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research*, 15(56), 1929–1958. <https://jmlr.org/papers/v15/srivastava14a.html#:~:text=Dropout%20is%20a%20technique%20for,%C3%A2%E2%82%AC%C5%93thinned%C3%A2%E2%82%AC%20networks.>
- Usmani, M., Adil, S. H., Raza, K., & Ali, S. S. A. (2016). Stock market prediction using machine learning techniques. 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), 322–327. <https://doi.org/10.1109/iccoins.2016.7783235>
- Vaidehi, M., Pandit, A., Jindal, B., Kumari, M., & Singh, R. (2021). Bitcoin Price Prediction Using Machine Learning. *International Journal of Engineering Technologies and Management Research*, 8(5), 20–28. <https://doi.org/10.29121/ijetmr.v8.i5.2021.953>
- van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, 53(8), 5929–5955. <https://doi.org/10.1007/s10462-020-09838-1>
- Venugopal, D., Kannan Kaliyaperumal, S., & Muthu Niraikulathan, S. (2021). Stock Market Trend Prediction Using Hidden Markov Model. *Forecasting in Mathematics - Recent Advances, New Perspectives and Applications*. <https://doi.org/10.5772/intechopen.93988>
- Viernes, F. A. (2021, 31 junio). When is Log Transformation Necessary for Financial Returns? Medium. Recuperado 8 de junio de 2022, de <https://medium.datadriveninvestor.com/when-is-log-transformation-necessary-for-financial-returns-4b3f5bb58e62>

- Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock Closing Price Prediction using Machine Learning Techniques. *Procedia Computer Science*, 167, 599–606. <https://doi.org/10.1016/j.procs.2020.03.326>
- Xing, T., Sun, Y., Wang, Q., & Yu, G. (2013). The analysis and prediction of stock price. 2013 IEEE International Conference on Granular Computing (GrC), 368–373. <https://doi.org/10.1109/grc.2013.6740438>
- Yetis, Y., Kaplan, H., & Jamshidi, M. (2014). Stock market prediction by using artificial neural network. 2014 World Automation Congress (WAC), 718–722. <https://doi.org/10.1109/wac.2014.6936118>
- Yi, S., Xu, Z., & Wang, G. J. (2018). Volatility connectedness in the cryptocurrency market: Is Bitcoin a dominant cryptocurrency? *International Review of Financial Analysis*, 60, 98–114. <https://doi.org/10.1016/j.irfa.2018.08.012>
- Zolotareva, E. (2021). Applying Convolutional Neural Networks for Stock Market Trends Identification. <https://arxiv.org/ftp/arxiv/papers/2104/2104.13948.pdf>
- Zuckerman, G. (2019). *The Man Who Solved the Market: How Jim Simons Launched the Quant Revolution* (Illustrated ed.). Penguin Random House.

ANEXO I: ANÁLISIS DE LA CORRELACIÓN DE LOS INPUTS PRELIMINARES RESPECTO DE LA VARIABLE TARGET

Se presentan, ordenados de mayor a menor en valor absoluto a pesar de que se muestra el valor real de la correlación, los resultados obtenidos del análisis de correlación para cada una de las variables preliminares.

Variable	Correlación	Variable	Correlación
BTC_ETH_ratio	-0,0466	BTC_5m_VolEMA/Vol	0,0034
BTC_5m_EMA/Close	-0,0386	BTC_120m_VolEMA/Vol	0,0033
BTC_15m_EMA/Close	-0,0374	BTC_240m_VolEMA/Vol	0,0031
BTC_15m_EMA	-0,0373	ETH_60m_EMA/Close	-0,0029
BTC_30m_EMA/Close	-0,0339	BTC_360m_VolEMA/Vol	0,0029
BTC_30m_EMA	-0,0338	ETH_60m_EMA	-0,0029
BTC_LTC_ratio	-0,0332	Volume_BTC_LogRet	0,0028
Close_BTC	-0,0325	Vol_USD_BTC_LogRet	0,0028
BTC_XRP_ratio	-0,0307	BTC_720m_VolEMA/Vol	0,0027
BTC_60m_EMA/Close	-0,0306	Low_XRP_LogRet	0,0026
BTC_60m_EMA	-0,0306	BTC_1440m_VolEMA/Vol	0,0024
BTC_120m_EMA	-0,0283	BTC_5m_VolEMA	0,0022
BTC_120m_EMA/Close	-0,0283	ETH_720m_VolEMA	0,0022
BTC_240m_EMA	-0,0258	ETH_360m_VolEMA	0,0022
BTC_240m_EMA/Close	-0,0257	ETH_1440m_VolEMA	0,0022
BTC_360m_EMA	-0,0235	ETH_240m_VolEMA	0,0021
BTC_360m_EMA/Close	-0,0235	ETH_120m_VolEMA	0,0019
BTC_720m_EMA	-0,0182	ETH_60m_VolEMA	0,0015
BTC_720m_EMA/Close	-0,0182	Open_LTC_LogRet	0,0014
Close_ETH_LogRet	0,0172	ETH_15m_EMA	0,0013
RealVar_BTC_30m	0,0162	Vol_USD_LTC_LogRet	0,0013
RealVar_BTC_15m	0,0148	Volume_LTC_LogRet	0,0013
RealVar_BTC_60m	0,0133	ETH_15m_EMA/Close	0,0013
BTC_1440m_EMA	-0,0132	BTC/ETH_Ratio_30m	0,0011
BTC_1440m_EMA/Close	-0,0132	BTC_RSI_360m	-0,0011
Close_LTC_LogRet	0,0116	Vol_USD_XRP_LogRet	0,0011
BTC_1440m_VolEMA	0,0106	Volume_XRP_LogRet	0,0011
Close_XRP_LogRet	0,0101	Vol_USD_ETH_LogRet	0,0011

Low_BTC_LogRet	-0,0099	Volume_ETH_LogRet	0,0010
BTC_720m_VolEMA	0,0094	Open_XRP_LogRet	-0,0009
BTC_360m_VolEMA	0,0083	BTC/ETH_Ratio_720m	-0,0009
ETH_5m_EMA	0,0081	ETH_30m_VolEMA	0,0008
High_ETH_LogRet	0,0081	BTC_RSI_30m	0,0008
ETH_5m_EMA/Close	0,0080	BTC_RSI_15m	-0,0008
BTC_240m_VolEMA	0,0077	ETH_30m_EMA/Close	-0,0007
ETH_360m_EMA	-0,0076	BTC_RSI_720m	0,0007
ETH_360m_EMA/Close	-0,0075	ETH_RSI_15m	-0,0007
ETH_240m_EMA	-0,0074	BTC_RSI_60m	-0,0007
ETH_240m_EMA/Close	-0,0074	Open_ETH_LogRet	0,0006
High_LTC_LogRet	0,0073	ETH_30m_EMA	-0,0006
High_BTC_LogRet	-0,0071	ETH_RSI_120m	0,0006
BTC_120m_VolEMA	0,0069	BTC/ETH_Ratio_360m	-0,0005
ETH_720m_EMA	-0,0064	ETH_RSI_360m	-0,0004
ETH_720m_EMA/Close	-0,0063	ETH_RSI_60m	-0,0004
BTC_60m_VolEMA	0,0062	BTC/ETH_Ratio_120m	-0,0004
ETH_120m_EMA	-0,0056	BTC/ETH_Ratio_60m	-0,0003
ETH_120m_EMA/Close	-0,0056	ETH_RSI_1440m	-0,0003
Low_ETH_LogRet	0,0054	BTC_RSI_120m	0,0003
ETH_1440m_EMA	-0,0053	BTC_RSI_1440m	-0,0002
ETH_1440m_EMA/Close	-0,0053	ETH_RSI_30m	0,0001
BTC_30m_VolEMA	0,0051	ETH_RSI_720m	0,0001
Low_LTC_LogRet	0,0043	ETH_15m_VolEMA	0,0001
High_XRP_LogRet	0,0040	ETH_5m_VolEMA	0,0000
BTC_15m_VolEMA	0,0036	BTC/ETH_Ratio_1440m	0,0000
BTC_30m_VolEMA/Vol	0,0035	BTC/ETH_Ratio_15m	0,0000
BTC_60m_VolEMA/Vol	0,0035	MACD_26_12_9	0,0000
BTC_15m_VolEMA/Vol	0,0035		