



Universidad del
Rosario

| Escuela de Ingeniería,
Ciencia y Tecnología

**PREDICCIÓN DEL PRECIO DEL BITCOIN UTILIZANDO ALGORITMOS DE
APRENDIZAJE PROFUNDO**

Presentado para obtener el título de:
Magister en Matemáticas Aplicadas y Ciencias de la Computación

Emanuelle Alejandro Moreno Quintero

Dirección:
Yiby Karolina Morales Pinto

Universidad del Rosario
Escuela de Ingeniería, Ciencia y Tecnología
Maestría en Matemáticas Aplicadas y Ciencias de la Computación
2023

Abstract

El mercado de criptomonedas está experimentando un rápido crecimiento, lo que lo convierte en una alternativa potencialmente más lucrativa que los mercados financieros convencionales. No obstante, esta expansión va de la mano con una significativa volatilidad, presentando así un desafío crucial. En el contexto de esta tesis de maestría, se desarrollaron modelos de predicción de series temporales para el precio de cierre de Bitcoin mediante el uso de algoritmos de aprendizaje profundo, tales como LSTM y GRU. Además, se llevó a cabo una comparación con modelos tradicionales como ARIMA, con el propósito de analizar y evaluar su rendimiento.

The cryptocurrency market is experiencing rapid growth, making it a potentially more lucrative alternative to conventional financial markets. However, this expansion goes hand in hand with significant volatility, thus presenting a crucial challenge. In the context of this master's thesis, time series prediction models for the closing price of Bitcoin were developed using deep learning algorithms such as LSTM and GRU. In addition, a comparison was carried out with traditional models such as ARIMA, with the purpose of analyzing and evaluating their performance.

Índice

1. INTRODUCCIÓN	5
2. OBJETIVOS	6
2.1. Objetivo general	6
2.2. Objetivos específicos	6
3. PROBLEMA Y JUSTIFICACIÓN	7
4. MARCO TEÓRICO Y ESTADO DEL ARTE	9
4.1. Marco Teórico	9
4.2. Estado del arte	16
5. Metodología	17
5.1. Obtención de datos	18
5.2. Análisis exploratorio de datos	18
5.3. Preparación de datos	21
5.4. Entrenamiento de modelos	21
5.5. Análisis de resultados:	21
5.6. Configuración de hiperparámetros:	23
6. Conclusiones	25
7. REFERENCIAS	26

Lista de tablas

1. Tendencias de la volatilidad del Bitcoin en los últimos años. 8
2. Métricas de los modelos en testeo 23

Lista de figuras

1.	Arquitectura Fundamental, Figura tomada de: https://www.europeanvalley.es/noticias/crear-red-neuronal-desde-las-matematicas/	9
2.	Funciones de activación, Figura tomada de: https://www.researchgate.net/	10
3.	Arquitectura RNN, Figura tomada de: https://www.researchgate.net/	12
4.	Arquitectura LSTM, Figura tomada de: https://www.themachinelearners.com/	13
5.	Arquitectura GRU, Figura tomada de: https://www.themachinelearners.com/	13
6.	Diagrama para la predicción del precio del Bitcoin	18
7.	Muestra de la tabla de datos	19
8.	Serie temporal desde 2018 hasta 2022	19
9.	Serie temporal filtrada	20
10.	Distribución variable objetivo	20
11.	ACF del precio de cierre	20
12.	Predicción de los modelos	22
13.	Distribución de los errores con LSTM	22
14.	Distribución de los errores con GRU	23

1. INTRODUCCIÓN

Una criptomoneda es una moneda digital que utiliza un sistema de registro para rastrear las transacciones. Este sistema es mantenido por una red de computadoras, en lugar de un organismo central. Este activo se diferencia de las monedas fiduciarias, como el dólar estadounidense o el euro, en que no están respaldadas por un gobierno o una institución financiera. En cambio, su valor se basa en la oferta y la demanda. Este mercado maneja la tecnología blockchain que permite registrar información en un libro de contabilidad compartido, de forma segura y transparente. Se utiliza un algoritmo de consenso para verificar la autenticidad de las transacciones. Esto hace que estas monedas sean más seguras y confiables que las monedas fiduciarias, ya que las transacciones son muy difíciles de falsificar o manipular.

Las criptomonedas han irrumpido en el panorama financiero global en los últimos años, desafiando las normas tradicionales del sistema monetario. Estas monedas virtuales descentralizadas, basadas en la tecnología blockchain, han experimentado un crecimiento significativo y han capturado la atención de inversores, empresas e instituciones financieras en todo el mundo. Sin embargo, la volatilidad extrema del mercado de las criptomonedas plantea desafíos tanto para los inversores como para los analistas financieros.

El mercado de las monedas digitales se caracteriza por cambios de precios drásticos y rápidos, lo que dificulta la toma de decisiones informadas y la predicción de las fluctuaciones del mercado. En este entorno altamente volátil, la necesidad de herramientas efectivas que permitan analizar y predecir los precios de estas se ha vuelto crucial. Afortunadamente, el desarrollo de modelos de aprendizaje automático y técnicas estadísticas avanzadas ha brindado oportunidades para abordar este desafío y mejorar la comprensión del comportamiento del mercado criptográfico.

La predicción del precio de las divisas digitales se presenta como un desafío considerable debido a su naturaleza única y disruptiva en comparación con las monedas tradicionales. Estas exhiben patrones de auge y caída, impulsados en gran medida por su naturaleza especulativa, y cuentan con una base de inversores predominantemente orientada hacia el corto plazo. Como resultado, el uso de métodos tradicionales para predecir sus precios se muestra como una aproximación poco adecuada y precisa.

En este contexto, el presente trabajo se inspira en los destacados resultados obtenidos por los algoritmos de Deep Learning, como las redes LSTM (Memoria a Corto y Largo Plazo) y GRU (Unidad Recurrente con Compuertas), con el objetivo de abordar la predicción del precio de cierre del Bitcoin. Este valor es especialmente complejo de anticipar debido a la naturaleza dinámica y especulativa del entorno criptográfico mencionado anteriormente. Para superar este desafío, se emplearán modelos autorregresivos adaptados con los algoritmos LSTM y GRU. Estos modelos se entrenarán utilizando una ventana de tiempo de 15 días hacia atrás para predecir el precio de cierre en el siguiente día.

La importancia de este estudio radica en su contribución al campo emergente de las finanzas digitales y su relevancia para los inversores y participantes del este mercado. Se espera que los resultados obtenidos permitan mejorar la toma de decisiones en este entorno incierto y proporcionen una mayor comprensión de los factores que influyen en los precios de este dinero digital.

En cuanto a la estructura del trabajo, se presenta una visión general de los capítulos y secciones que abordarán aspectos clave de la investigación, incluyendo la metodología utilizada, los datos analizados, los modelos implementados y los resultados obtenidos. Además, se discutirán las implicaciones de los hallazgos y se plantearán posibles recomendaciones y áreas de investigación futura.

Con esta investigación, se espera contribuir al conocimiento existente sobre la predicción de precios de criptomonedas y proporcionar a los inversores y analistas financieros herramientas valiosas para mejorar su comprensión del mercado criptográfico en constante evolución.

2. OBJETIVOS

2.1. Objetivo general

Desarrollar y aplicar modelos autorregresivos basados en los algoritmos LSTM,GRU y CNN para predecir el precio de cierre del Bitcoin en el mercado de las criptomonedas, comparandolo con los algoritmos tradicionales de series temporales como ARIMA.

2.2. Objetivos específicos

1. Construir los modelos propuestos haciendo uso de los algoritmos de aprendizaje profundo propuestos.
2. Evaluar el rendimiento de los modelos utilizando métricas adecuadas para la predicción de series temporales, como el error cuadrático medio (MSE) o el error absoluto medio (MAE).
3. Comparar los resultados obtenidos con los modelos autorregresivos basados en LSTM, GRU y CNN con los resultados de algoritmos tradicionales de series temporales, como ARIMA o modelos de suavizado exponencial.

3. PROBLEMA Y JUSTIFICACIÓN

Las criptomonedas son monedas virtuales utilizadas en los sistemas financieros actuales. A diferencia de las monedas tradicionales, no son emitidas por ningún gobierno, autoridad central o banco, lo que las hace descentralizadas. Todas las criptomonedas se basan en la tecnología Blockchain, una infraestructura altamente compleja que garantiza la seguridad y la integridad de los datos almacenados en ella, de tal manera que es difícil de piratear y modificar. En los últimos años, las criptomonedas han experimentado un crecimiento dramático, ganando popularidad y aumentando su capitalización. Bitcoin, desarrollada por Satoshi Nakamoto [1], fue la primera criptomoneda descentralizada y se ha convertido en la más valiosa del mundo. El primer Bitcoin fue utilizado por un programador de Florida para comprar una pizza en mayo de 2010 [2]. A medida que el volumen de transacciones aumentaba, se introdujeron diversos tipos de criptomonedas en el mundo de la criptografía. Algunas de las criptomonedas más conocidas incluyen Ethereum y Ripple.

En 2019, el mercado de criptomonedas rompió su silencio desde que estalló la burbuja en 2017, cuando se disparó el precio de este activo, en particular, el precio de Bitcoin alcanzó nuevos niveles récord bajo el estímulo de las políticas globales de relajación. El 7 de marzo del 2021, el precio de Bitcoin alcanzó los \$51 206,69, marcando un hito significativo. Mientras tanto, se negociaron casi 6 000 criptomonedas diferentes y la capitalización total del mercado alcanzó los \$1,48 billones [3]. Las 10 principales criptomonedas representan aproximadamente el 85% de la cuota de mercado total, con Bitcoin dominando con aproximadamente el 64% de la capitalización de mercado [4]. Dado que las transacciones de criptomonedas están descentralizadas, están disponibles en muchas plataformas, lo que las hace más seguras en comparación con las transacciones centralizadas. La inversión en este tipo de mercados aporta una buena liquidez al ecosistema. Las criptomonedas tienen un futuro prometedor debido a los múltiples avances que han llevado a importantes industrias a invertir en ellas, incluyendo reconocidos bancos, empresas e inversores institucionales como JP Morgan, MicroStrategy, Tesla y PayPal, entre otros [5].

Estas ventajas atraen a los comerciantes a invertir en estas, pero su precio cambia cada día como las monedas fiduciarias. La “volatilidad excepcional de los precios” es una de las principales características de los blockchains [6]. El cuadro 2 muestra los cambios en el precio de Bitcoin en diferentes períodos de tiempo, junto con el correspondiente cambio porcentual [7]. Se observa que el precio de Bitcoin ha experimentado cambios significativos en distintos intervalos de tiempo. En períodos más cortos, ha mostrado fluctuaciones notables, con aumentos en el último mes, últimos 6 meses y último año. No obstante, también ha habido períodos de disminución en los últimos 2 años. Estas bruscas variaciones en el precio de Bitcoin resaltan la volatilidad inherente en el mercado de las criptomonedas, lo que puede presentar oportunidades de ganancia, pero también implica riesgos considerables para los inversores. A pesar de las grandes fluctuaciones de precios desde 2016, el interés en las criptomonedas se ha mantenido más o menos constante [8].

La volatilidad de las criptomonedas puede ser resultado de diversas causas, como la interacción entre inversores a través de plataformas como Twitter, que puede generar burbujas [9], así como las búsquedas en Google Trends y Wikipedia [10]. En especial los dos fenómenos anteriores, incitan a la compra o venta de estas divisas, lo cual en una ventana de tiempo dada también genera volatilidad [11]. No obstante, resulta difícil determinar exactamente qué impulsa el precio de estas monedas digitales a lo largo del tiempo. Como ocurre con la mayoría de las fluctuaciones de precios en el mundo de las monedas criptográficas, las causas exactas son difíciles de identificar. Con el creciente interés en este mercado y su relevancia en el mundo financiero, resulta crucial contar con un conjunto de herramientas efectivas para predecir las variaciones al alza o a la baja en los precios de las monedas digitales de interés. Por lo tanto, en los últimos años, los investigadores han propuesto diversas técnicas para predecir y modelar el precio de estas y analizar la volatilidad del mercado criptográfico. Algunos de ellos han empleado métodos estadísticos, mientras que otros han utilizado el Aprendizaje Profundo como enfoque para lograr una predicción más precisa del precio de Bitcoin [12].

El análisis predictivo con Aprendizaje Profundo se ha convertido en una herramienta clave para abordar la necesidad de proporcionar precios actualizados y estimar el rendimiento de las inversiones en

Periodo	Variación (\$)	Variación (%)	Tendencia
Últimos 30 días	+4,068.35	+15.04	↗
Últimos 6 meses	+14,470.46	+86.87	↗
Último año	+12,002.84	+62.76	↗
Últimos 2 años	-3,617.44	-10.41	↘
Últimos 3 años	+21,993.44	+240.79	↗
Últimos 4 años	+19,428.93	+166.08	↗
Últimos 5 años	+24,442.34	+365.64	↗
Últimos 6 años	+28,582.01	+1,123.02	↗
Últimos 7 años	+30,452.94	+4,517.03	↗
Últimos 8 años	+30,867.87	+11,906.60	↗
Últimos 9 años	+30,488.83	+4,776.64	↗

Cuadro 1: Tendencias de la volatilidad del Bitcoin en los últimos años.

mercado digitales. Con el aumento de la demanda de criptomonedas en todo el mundo, es importante contar con soluciones que permitan conocer el monto neto a recibir en caso de invertir en ellas. A medida que los precios de las monedas criptográficas experimentan aumentos significativos, se generan oportunidades para los inversores a nivel mundial. Por esta razón, en este documento, se trata de poner en práctica esos modelos de Aprendizaje Profundo de última generación para ver qué tan bien funcionan en la búsqueda de señales comerciales empleando como entrada el precio de cierre.

4. MARCO TEÓRICO Y ESTADO DEL ARTE

4.1. Marco Teórico

Redes Neuronales (NN) Las redes neuronales son un componente esencial del campo del aprendizaje automático e inteligencia artificial. Su diseño y funcionamiento están inspirados en el sistema nervioso humano, específicamente en la forma en que las neuronas en el cerebro se comunican y procesan la información. Las redes neuronales son una clase de modelos computacionales altamente flexibles y adaptables, capaces de aprender de manera autónoma a partir de datos y mejorar su rendimiento en tareas específicas a medida que reciben más información.

El concepto fundamental detrás de una red neuronal es la neurona artificial, que es la unidad básica de procesamiento. Cada neurona artificial recibe una o más entradas, las pondera por medio de pesos y un sesgo (bias), y luego aplica una función de activación para producir una salida. La interconexión de múltiples de estas neuronas artificiales da lugar a una arquitectura de red neuronal.

Las redes neuronales pueden tener diferentes estructuras y topologías, dependiendo de la naturaleza de la tarea que deben resolver. Algunas de las arquitecturas más comunes incluyen las redes neuronales feedforward, donde la información fluye en una dirección, de la entrada hacia la salida; las redes neuronales recurrentes, que permiten conexiones cicladas y se utilizan en tareas que involucran secuencias; y las redes neuronales convolucionales, especialmente adecuadas para problemas de visión por computadora y procesamiento de imágenes.

El entrenamiento de una red neuronal implica ajustar los pesos y sesgos para minimizar el error entre las salidas predichas por la red y las salidas reales conocidas, lo que se conoce como el proceso de aprendizaje. Esto se logra utilizando algoritmos de optimización, como el descenso del gradiente, que ajustan gradualmente los parámetros de la red en función del error calculado durante el entrenamiento.

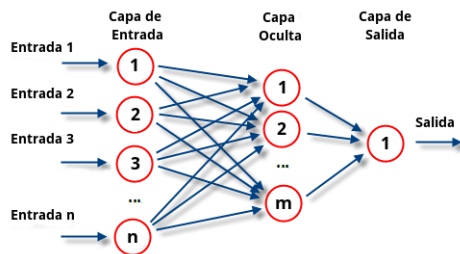


Figura 1: [Arquitectura Fundamental](https://www.europeanvalley.es/noticias/crear-red-neuronal-desde-las-matematicas/), Figura tomada de: <https://www.europeanvalley.es/noticias/crear-red-neuronal-desde-las-matematicas/>

La explicación de las diferentes capas y conexiones que se muestran en la figura 1 se muestran a continuación.

Neurona Artificial: La neurona artificial es el bloque fundamental que conforma una red neuronal. Está diseñada para imitar el comportamiento de las neuronas biológicas presentes en el cerebro humano, y su función es procesar y transmitir información en una red neuronal [13].

La estructura básica de una neurona artificial consiste en las siguientes partes:

- **Capa de entradas:** La neurona recibe múltiples entradas, cada una de las cuales está asociada con un valor numérico. Estas entradas representan la información que la neurona debe procesar. En una red neuronal, las entradas pueden provenir de otras neuronas en capas previas o directamente de los datos de entrada del problema.
- **Capa Oculta:** Es una capa intermedia entre la capa de entrada y la capa de salida. En esta capa, las neuronas (nodos) realizan cálculos y transformaciones en los datos de entrada antes de

pasarlos a la capa de salida. La capa oculta es fundamental para que la red pueda aprender y modelar relaciones y patrones complejos en los datos. Cuantas más capas ocultas tenga una red neuronal, más profunda se considera la red, lo que puede aumentar su capacidad para aprender representaciones abstractas y realizar tareas más sofisticadas

- **Pesos:** Cada entrada está multiplicada por un peso correspondiente. Los pesos representan la fuerza de la conexión entre la neurona actual y las neuronas que la preceden. A través del proceso de entrenamiento, la red neuronal ajustará estos pesos para capturar patrones y características relevantes en los datos de entrada.
- **Sesgo:** Además de las entradas ponderadas, una neurona también tiene un valor llamado sesgo (bias). El sesgo actúa como un término independiente que permite que la neurona ajuste su salida incluso cuando todas las entradas son cero.
- **Función de activación:** Después de sumar las entradas ponderadas y el sesgo, la neurona aplica una función de activación a la suma para producir la salida de la neurona. La función de activación introduce no linealidad en la neurona, lo que le permite aprender y representar relaciones complejas en los datos.

Existen diferentes tipos de funciones de activación utilizadas en las redes neuronales, entre las cuales se incluyen [13]:

- **Función Sigmoide:** Produce valores en un rango entre 0 y 1, y fue ampliamente utilizada en el pasado, aunque ha sido reemplazada en muchas ocasiones por otras funciones debido a ciertos problemas asociados con el desvanecimiento del gradiente.
- **Función ReLU (Rectified Linear Unit):** Es la función de activación más utilizada en la actualidad. Es simple y computacionalmente eficiente, retornando el valor de la entrada si es positiva y cero si es negativa.
- **Función Tanh (Tangente hiperbólica):** Produce valores en un rango entre -1 y 1. Es similar a la función sigmoide pero con un rango más amplio, lo que puede mejorar el rendimiento en ciertos casos.

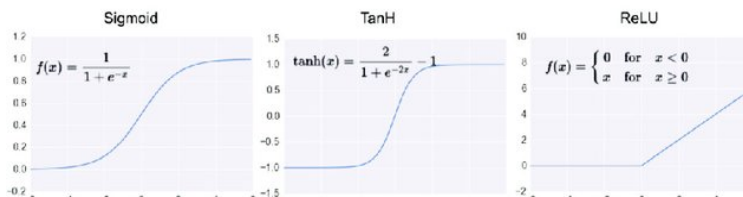


Figura 2: **Funciones de activación**, Figura tomada de: <https://www.researchgate.net/>

Una vez que una neurona ha calculado su salida, esta información se transmite a través de las conexiones de la red neuronal, y el proceso se repite en las capas siguientes hasta que se obtiene la salida final de la red, que representa la predicción o clasificación del problema en cuestión.

Arquitectura de las redes neuronales La arquitectura de una red neuronal se refiere a su estructura y organización, incluyendo el número de capas, la cantidad de neuronas en cada capa y la forma en que están interconectadas. La elección de la arquitectura juega un papel crucial en el rendimiento y la eficacia de la red para resolver problemas específicos. A continuación, se describen algunas de las arquitecturas de redes neuronales más comunes:

- **Redes Neuronales Feedforward:** Las redes neuronales feedforward son el tipo más básico de arquitectura y también se les conoce como redes de propagación hacia adelante. En este tipo de red, la información fluye en una sola dirección, desde la capa de entrada, a través de una o varias capas ocultas (si las hay), hasta la capa de salida. Cada neurona en una capa está conectada a todas las neuronas de la capa siguiente, y las conexiones no forman ciclos. Estas redes son adecuadas para problemas de clasificación, regresión y otras tareas donde no hay dependencias temporales en los datos.

- **Redes Neuronales Convolucionales (CNN):** Las redes neuronales convolucionales son especialmente efectivas en tareas que involucran datos estructurados, como imágenes y videos. Están diseñadas para capturar patrones locales y características en datos bidimensionales o tridimensionales. Una característica distintiva de las CNN es el uso de operaciones de convolución y capas de agrupación (pooling) para reducir la dimensionalidad y extraer características relevantes. Estas redes son ampliamente utilizadas en aplicaciones de visión por computadora, detección de objetos y reconocimiento de imágenes.

En la práctica, se pueden combinar diversas arquitecturas de redes neuronales para abordar problemas complejos y lograr un rendimiento óptimo. La elección de la arquitectura adecuada depende del tipo de datos, la naturaleza del problema y la cantidad de datos disponibles para el entrenamiento. La evolución constante de las arquitecturas de redes neuronales y el desarrollo de nuevas técnicas de modelado continúan impulsando el progreso en el campo del aprendizaje automático y la inteligencia artificial.

Algoritmos de aprendizaje El aprendizaje en las redes neuronales es un proceso fundamental mediante el cual los pesos y sesgos de las neuronas se ajustan iterativamente para mejorar el rendimiento de la red en una tarea específica. Existen varios algoritmos de aprendizaje utilizados en el entrenamiento de redes neuronales. A continuación, se describen algunos de los más comunes [13]:

- **Descenso del Gradiente (Gradient Descent):** El descenso del gradiente es el algoritmo de aprendizaje más básico y ampliamente utilizado en las redes neuronales. Funciona ajustando los pesos y sesgos de la red para minimizar la función de pérdida, que mide el error entre las salidas predichas y las salidas reales. En cada iteración, se calcula el gradiente de la función de pérdida con respecto a los pesos y sesgos, y luego se actualizan los parámetros en la dirección opuesta del gradiente para reducir el error.
- **Descenso del Gradiente Estocástico (Stochastic Gradient Descent, SGD):** El SGD es una variante del descenso del gradiente que se utiliza comúnmente en conjunción con grandes conjuntos de datos de entrenamiento. En lugar de calcular el gradiente sobre todos los datos de entrenamiento en cada iteración, el SGD selecciona un subconjunto aleatorio (lote) de datos para calcular el gradiente y actualizar los parámetros. Esto hace que el proceso de entrenamiento sea más rápido y eficiente, aunque con mayor variabilidad en las actualizaciones de los pesos.
- **Descenso del Gradiente con Momento (Gradient Descent with Momentum):** Este enfoque incorpora el concepto de momentum para acelerar el proceso de entrenamiento. El momentum acumula una fracción de los gradientes pasados para guiar la dirección de la actualización de los pesos, lo que puede ayudar a superar áreas de mínimos locales y converger más rápidamente.

Cada algoritmo de aprendizaje tiene sus ventajas y desventajas, y la elección del algoritmo depende de factores como la naturaleza del problema, el tamaño del conjunto de datos, la arquitectura de la red y las preferencias del desarrollador. La selección adecuada del algoritmo de aprendizaje puede tener un impacto significativo en el rendimiento y la eficiencia de la red neuronal entrenada.

Funciones de pérdida (Loss Functions) Las funciones de pérdida son una parte esencial del proceso de entrenamiento de redes neuronales, ya que son utilizadas para medir el error entre las salidas predichas por la red y las salidas reales (etiquetas) proporcionadas durante el entrenamiento. La elección adecuada de la función de pérdida depende del tipo de problema que se esté abordando, como clasificación, regresión o tareas de generación.

- **Error Cuadrático Medio (Mean Squared Error, MSE):** El error cuadrático medio es una función de pérdida comúnmente utilizada en problemas de regresión. Calcula la diferencia cuadrática promedio entre las salidas predichas y las salidas reales. MSE es sensible a los valores atípicos y penaliza en gran medida los errores grandes.

La comparación de varios modelos o métodos de predicción con MSE es útil. En el mismo conjunto de datos, se cree que un modelo con un MSE más bajo es más preciso. Esto permite evaluar la eficacia relativa de varios modelos para predecir el precio de cierre de Bitcoin, no obstante se debe tener en cuenta que esta medida se encuentra en unidades cuadradas, en nuestro caso serían

dolares al cuadrado y se podría dificultar la interpretación. Es por esto que también se utiliza la siguiente métrica de evaluación.

- **Error Absoluto Medio (MAE):** Es una función de pérdida utilizada en problemas de regresión. Representa la diferencia absoluta promedio entre las predicciones de un modelo y los valores reales de los datos de prueba. En otras palabras, el MAE mide cuánto se desvían, en promedio, las predicciones del modelo de los valores verdaderos. Cuanto menor sea el valor del MAE, mejor será el ajuste del modelo a los datos, ya que indica que las predicciones se acercan más a los valores reales.

El MAE está en la misma unidad de medida que la variable objetivo, a diferencia del MSE que está en unidades cuadradas. Esto facilita la interpretación porque el MAE representa el error promedio en la predicción en la misma unidad monetaria que se está tratando de predecir. El MAE asigna el mismo peso a todos los errores, sin importar su magnitud, a diferencia del MSE, que penaliza de manera más significativa los errores grandes debido al término al cuadrado. Esto significa que un MAE alto no penaliza más los errores grandes, pero indica que las predicciones del modelo están, en promedio, alejadas de los valores reales en términos absolutos.

Redes Neuronales Recurrentes (RNN) Son un tipo de arquitectura de redes neuronales diseñadas específicamente para procesar secuencias de datos, donde la información tiene una dependencia temporal. A diferencia de las redes neuronales feedforward, las RNNs contienen conexiones recurrentes que permiten que la información fluya a través de ciclos, como se muestra en la figura 3 lo que les da la capacidad de mantener un estado oculto que encapsula la información relevante sobre secuencias anteriores. Esta propiedad las convierte en una herramienta poderosa para abordar problemas con datos secuenciales, como el procesamiento del lenguaje natural, el reconocimiento de voz, la generación de texto y las predicciones de series temporales.

La Celda Básica de RNN: La unidad básica en una RNN es la celda recurrente, que se repite en el tiempo para procesar cada elemento de la secuencia de entrada. La celda recurrente toma como entrada el dato actual de la secuencia y el estado oculto anterior, y produce una nueva salida y un nuevo estado oculto. El estado oculto actúa como una representación de la información pasada en la secuencia y se comparte en todas las etapas de tiempo, lo que le permite capturar las dependencias a largo plazo en los datos.

Una limitación importante en las RNNs es el problema del desvanecimiento o explosión del gradiente durante el entrenamiento. Cuando la red se entrena con secuencias largas, los gradientes que retropropagan a través de los ciclos recurrentes pueden hacerse muy pequeños (desvanecimiento) o muy grandes (explosión). Esto puede dificultar el aprendizaje efectivo de dependencias a largo plazo. Para abordar este problema, se han propuesto técnicas como gradient clipping y las redes LSTM, que se describirán a continuación.

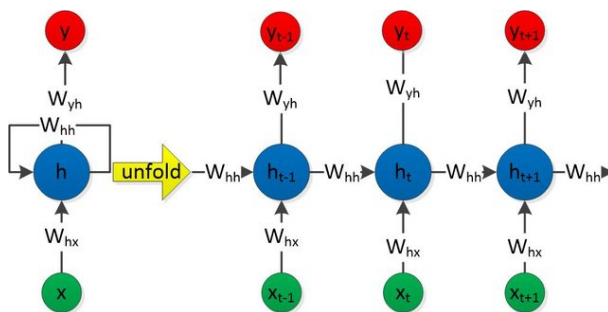


Figura 3: [Arquitectura RNN](https://www.researchgate.net/), Figura tomada de: <https://www.researchgate.net/>

Redes Long Short Term Memory (LSTM) Las redes LSTM son un tipo especializado de redes neuronales recurrentes diseñadas para superar el problema de desvanecimiento o explosión del gradiente, que es común en las RNNs tradicionales al entrenar con secuencias largas. Fueron propuestas por primera vez por Sepp Hochreiter y Jürgen Schmidhuber en 1997 en [14]. Las redes LSTM son particularmente adecuadas para modelar y procesar secuencias de datos con dependencias a largo plazo y

patrones complejos.

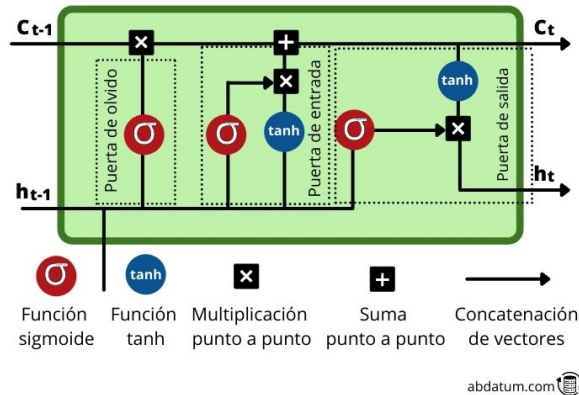


Figura 4: [Arquitectura LSTM](https://www.themachinelearners.com/), Figura tomada de: <https://www.themachinelearners.com/>

La Celda LSTM: Es la unidad básica de una red LSTM es la celda LSTM, que es una estructura que contiene varias compuertas (gates) que controlan el flujo de información y regula cómo se almacena y olvida la información en la memoria de la celda. Las principales compuertas en una celda LSTM son:

- Compuerta de Entrada (Input Gate): Controla la cantidad de nueva información que se debe agregar a la memoria de la celda.
- Compuerta de Olvido (Forget Gate): Decide qué información debe olvidarse de la memoria de la celda.
- Compuerta de Salida (Output Gate): Regula la cantidad de información que se debe leer desde la memoria de la celda y utilizar como salida de la celda LSTM.

La característica clave de las redes LSTM es su capacidad para mantener y utilizar información relevante a largo plazo a través del mecanismo de memoria a largo plazo. Esto se logra mediante la combinación de las compuertas y una serie de operaciones matemáticas que permiten que la celda LSTM aprenda cuándo almacenar nueva información, cuándo olvidar información obsoleta y cuándo leer y utilizar la información relevante en tareas posteriores.

Redes GRU Las Redes GRU (Gated Recurrent Units) son otra variante de las Redes Neuronales Recurrentes (RNNs) que fueron desarrolladas para abordar los desafíos del desvanecimiento y explosión del gradiente presentes en las RNNs tradicionales. Al igual que las LSTM, las GRU incorporan mecanismos de compuertas para controlar el flujo de información a lo largo del tiempo y permiten el aprendizaje efectivo de dependencias a largo plazo en secuencias de datos.

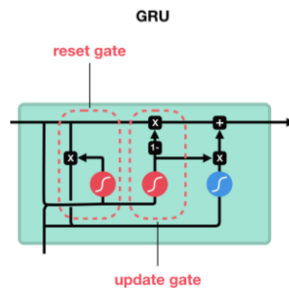


Figura 5: [Arquitectura GRU](https://www.themachinelearners.com/), Figura tomada de: <https://www.themachinelearners.com/>

Compuertas en las Redes GRU: Las Redes GRU contienen dos tipos de compuertas principales: la compuerta de reinicio (reset gate) y la compuerta de actualización (update gate). Estas compuertas trabajan juntas para controlar la cantidad de información que se mantiene en el estado oculto de la red y qué información se debe olvidar o actualizar en función de la entrada actual y el estado anterior.

- **Compuerta de Reinicio (Reset Gate):** La compuerta de reinicio determina qué parte del estado oculto anterior debe olvidarse antes de calcular la nueva información que se incorporará en el estado actual.
- **Compuerta de Actualización (Update Gate):** La compuerta de actualización controla cuánta información nueva se incorporará al estado oculto actual y cuánta información se mantendrá del estado anterior.

El mecanismo de adaptación en las Redes GRU permite que estas redes aprendan cuándo mantener y cuándo olvidar información relevante en secuencias pasadas. Esto les permite capturar dependencias a largo plazo en los datos de manera eficiente, similar a las LSTM, pero con menos parámetros y, en algunos casos, menos costo computacional.

Comparación con Redes LSTM: Aunque comparten similitudes en el control del flujo de información a lo largo del tiempo, las Redes GRU y las Redes LSTM tienen diferencias en su estructura y número de parámetros. Mientras que las LSTM son más adecuadas para capturar patrones más complejos y tienen un mayor costo computacional debido a su arquitectura más compleja, las GRU ofrecen una alternativa más ligera y eficiente, que puede ser suficiente y adecuada en muchos escenarios.

En resumen, las Redes GRU (Gated Recurrent Units) son una variante eficiente y efectiva de las Redes Neuronales Recurrentes (RNNs) que superan el problema del desvanecimiento y explosión del gradiente. Su mecanismo de compuertas permite capturar dependencias a largo plazo en secuencias de datos y las hace especialmente útiles en tareas que involucran procesamiento de datos secuenciales y dependencias temporales.

Los dos enfoques principales en el campo del aprendizaje profundo y la predicción de series temporales son los modelos autoregresivos (AR), las redes LSTM y GRU. Los modelos autoregresivos se basan en la idea de que los valores futuros de una serie temporal dependen linealmente de sus valores anteriores, mientras que las redes LSTM son una variante avanzada de las redes neuronales recurrentes (RNN) que se utilizan para capturar y modelar relaciones a largo plazo en secuencias de datos. Aunque ambos métodos tienen sus propias ventajas y desventajas, su uso se ha vuelto esencial para una variedad de aplicaciones, como el procesamiento del lenguaje natural y la predicción de precios financieros.

Definición del Modelo AR(p): Un modelo AR(p) asume que el valor de una serie de tiempo en un momento t (denotado como y_t) se puede explicar como una combinación lineal de sus p valores anteriores, más un término de error ϵ_t [15]:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \dots + \beta_p y_{t-p} + \epsilon_t \quad (1)$$

En donde p es la cantidad de pasos hacia atrás en el tiempo, β son los coeficientes de regresión y ϵ es un ruido blanco.

El objetivo principal al utilizar un modelo AR(p) es estimar los valores de los coeficientes $\beta_0, \beta_1, \dots, \beta_p$ que mejor se ajusten a los datos observados en la serie de tiempo. Esto se logra a través de técnicas de estimación, como el método de mínimos cuadrados o el método de máxima verosimilitud, que buscan minimizar la suma de los cuadrados de los errores (suma de los residuos al cuadrado) o maximizar la probabilidad de los datos observados, respectivamente.

El modelo AR(p) asume que la serie de tiempo es estacionaria, lo que significa que su media y varianza no cambian a lo largo del tiempo. Si la serie de tiempo no es estacionaria, es posible que sea necesario aplicar técnicas de transformación, como diferenciación o descomposición estacional, para hacerla estacionaria antes de ajustar el modelo.

Además, el modelo AR(p) se basa en la suposición de que la relación autoregresiva entre los valores pasados y el valor actual es lineal. Si la relación es no lineal, es posible que otros modelos, como modelos de regresión no lineal o modelos ARIMA, sean más adecuados.

En resumen, un modelo autoregresivo AR(p) es una herramienta valiosa para analizar y predecir series de tiempo, siempre que se cumplan las suposiciones de estacionariedad y linealidad en la relación autoregresiva. La elección adecuada del orden del modelo (p) y la estimación precisa de los coeficientes autoregresivos son fundamentales para obtener resultados.

La función de autocorrelación (ACF) es una herramienta crucial para el análisis de series temporales y es esencial para la creación de modelos autoregresivos (AR). La ACF permite visualizar y cuantificar la relación entre los valores actuales y los valores previos en una serie temporal, lo que es esencial para determinar el orden adecuado del modelo AR. Podemos encontrar patrones de correlación y dependencia en la ACF al examinarla. Estos patrones nos ayudan a determinar cuántos pasos en el tiempo pasado deben incluirse en la regresión autoregresiva. En este contexto, hablaremos sobre cómo la ACF ayuda a crear modelos AR precisos y efectivos para la modelización y predicción de datos secuenciales.

Función de AutoCorrelación (ACF): Representa la correlación entre una serie de tiempo y sus valores pasados (retardos/delay) a diferentes intervalos de tiempo (eje x). En otras palabras, el ACF muestra cómo una observación en un momento dado se correlaciona (eje y) con sus valores previos a lo largo del tiempo.

Función de AutoCorrelación Parcial (PACF) Es una función que representa la relación entre dos valores en una serie temporal sin tener en cuenta los valores intermedios. En otras palabras, el PACF solo mide las correlaciones directas entre dos puntos a lo largo del tiempo, ignorando las correlaciones indirectas generadas por los valores intermedios. Debido a que la estacionariedad es una suposición esencial para que los modelos autorregresivos (AR) sean válidos y eficaces, la implementación de pruebas de estacionariedad es crucial al trabajar con estos modelos. La estacionariedad es cuando las propiedades estadísticas de una serie temporal, como su media y varianza, permanecen constantes a lo largo del tiempo.

Test de Dickey-Fuller: Es una prueba estadística utilizada para determinar si una serie de tiempo es estacionaria o no estacionaria. Una serie de tiempo estacionaria es aquella cuyas propiedades estadísticas, como la media y la varianza, son constantes a lo largo del tiempo. Tomaremos nuestra significancia con un valor $\alpha = 0,05$.

- H_0 : Si $p > \alpha$. La serie no es estacionaria.
- H_a : Si $p \leq \alpha$. La serie es estacionaria.

Con el fin de incluir más variables explicativas en el desarrollo de modelos se describen las siguientes características las cuales son variables económicas y que son utilizadas en el ajuste de un modelo GRU con una ventana de tiempo de 5 días, estas características fueron extraídas mediante la librería TA-Lib (Technical Analysis) de Python:

- Media Móvil Exponencial (EMA - Exponential Moving Average): La EMA es similar a la SMA (Simple Moving Average - Promedio Móvil Simple) pero da más peso a los precios más recientes, lo que la hace sensible a los cambios recientes en la tendencia.
- Índice de Fuerza Relativa (RSI - Relative Strength Index): El RSI mide la velocidad y el cambio de los movimientos de precios y puede ayudar a identificar condiciones de sobrecompra o sobreventa en el mercado.

El EMA puede capturar de manera efectiva la dirección general de la tendencia del precio del Bitcoin al incluir el EMA en un modelo de predicción. La tendencia es una parte importante de la predicción de precios porque indica si el precio está en una fase alcista, bajista o lateral. De manera similar el RSI en nivel extremo puede indicar un posible cambio de tendencia. El uso de RSI en un modelo de predicción puede dar señales significativas sobre la fuerza dominante en el mercado de Bitcoin y ayudar a predecir posibles cambios de precios.

4.2. Estado del arte

Esta sección analiza los diversos algoritmos de aprendizaje automático disponibles para predecir los precios de criptomonedas. También muestra los estudios y comparaciones realizadas entre los algoritmos de deep learning para la predicción de precios por varios autores.

Varios factores afectan el precio de las criptomonedas. Algunos de estos factores se centran en los valores de precios de apertura y cierre, también en los valores de precios más altos y más bajos alcanzados en un día, lo que preocupa a los traders e inversionistas en varias partes del mundo [16]. El precio de las criptomonedas está relacionado con los datos de transacción de la moneda digital, es decir, un aumento en el volumen de negociación influye en el precio. Entonces, las criptomonedas se usan parcialmente como medio de intercambio. Esto implica que el principal impulsor del precio es la demanda especulativa y, por lo tanto, debe considerarse un activo invertible [17].

Los precios de las criptomonedas tienen fluctuaciones frecuentes, por lo tanto, se han incorporado muchos modelos de aprendizaje automático y aprendizaje profundo para la predicción de sus precios. Hay muchos modelos de aprendizaje profundo que se adoptan para la predicción de precios [18–30]. Por ejemplo, en [31] propusieron un algoritmo de pronóstico que se puede aplicar a diferentes tareas financieras, de ingeniería y médicas. El algoritmo integró una red neuronal artificial (ANN) y un perceptrón multicapa (MLP). A partir de los experimentos, la incorporación de MLP en ANN aumentó el poder de la predicción del precio de Bitcoin. En [32], el cambio del precio de Bitcoin se modela como un problema de clasificación binomial (subida o bajada) utilizando algoritmos de aprendizaje automático como modelos lineales generalizados y bosques aleatorios. Velankar et al. [33] se centran en las características óptimas que afectan los precios de Bitcoin y el enfoque de predicción de precios propuesto utilizando diferentes algoritmos de aprendizaje automático como la regresión bayesiana y el bosque aleatorio.

Yenidogan et al. adoptaron un algoritmo de aprendizaje profundo, conocido como Facebook Prophet para la predicción de precios de Bitcoin [34]. Se llevó a cabo una técnica de división triple para proporcionar proporciones óptimas para los conjuntos de entrenamiento, prueba y validación. Los resultados experimentales mostraron que el algoritmo PROPHET superó al algoritmo ARIMA. McNally et al. [35] aprovecharon varios algoritmos de aprendizaje profundo para predecir el precio de Bitcoin. En el proceso inicial, los patrones funcionales se extraían de los datos mediante ingeniería de características. Los resultados experimentales demostraron que el modelo de memoria a corto plazo (LSTM) logró un mejor rendimiento, mientras que las redes neuronales recurrentes (RNN) lograron un rendimiento mucho más bajo. Politis et al. [36] aprovecharon múltiples modelos de aprendizaje profundo para predecir el precio de Ethereum. La selección de características se realizó para reducir la complejidad y las anomalías del conjunto de datos. El modelo de conjunto se implementó con la combinación de LSTM, GRU y/o redes convolucionales temporales (TCN). En el resultado experimental del pronóstico diario, el modelo de conjunto con LSTM, GRU y Hybrid GRU-TCN tuvo el mejor desempeño.

En la literatura se han propuesto tres tipos de modelos de predicción de series temporales: modelos estadísticos, modelos de inteligencia artificial y modelos híbridos. En las últimas décadas, los investigadores han utilizado modelos estadísticos convencionales, como los modelos autorregresivos (AR) [37], la media móvil autorregresiva (ARMA) [38], la media móvil integrada autorregresiva (ARIMA) [2] y la regresión lineal multivariada [39]. para pronosticar el precio de Bitcoin. Los modelos estadísticos no son apropiados para sistemas caóticos (como el mercado de criptomonedas) con muchas incertidumbres porque requieren que los datos de series temporales estén sujetos a suposiciones específicas a priori, como la estacionariedad [40].

Wu et al. [41] construyó un marco de pronóstico para la predicción de precios de bitcoin mediante la incorporación de un modelo de serie temporal. En su estudio, los autores propusieron una red neuronal LSTM con un modelo autorregresivo AR(2) que usa la función de autocorrelación (ACF) y la función de autocorrelación parcial (PACF) de los precios de bitcoin para encontrar desfases de precios y periodos de promedio móvil y luego usa volumen de traducción como variable predictora. El rendimiento predictivo del modelo se comparó con el LSTM convencional utilizando cuatro métricas de error: errores cuadráticos medios (MSE), errores cuadráticos medios (RMSE), errores absolutos me-

dios (MAE) y errores porcentuales absolutos medios (MAPE). Los resultados mostraron que el modelo LSTM con AR (2) propuesto tenía errores de predicción más pequeños que el LSTM convencional. Esta investigación también ofreció una aplicación LSTM para ayudar a superar y mejorar el problema de selección de variables de entrada sin depender de suposiciones estadísticas estrictas sobre los datos.

Se observa que el problema de predecir el precio de una criptomoneda se puede abordar desde diferentes frentes, ya sea regresión o clasificación binaria (subida o bajada del precio), sin embargo este último enfoque tiene el problema que no indica el grado de cambio que va a tener la criptomoneda. También lo podemos abordar desde la regresión con el fin de predecir el precio de cierre de tal forma que se tenga una idea de si será un buen momento de comprar o vender los activos. Dado que las redes LSTM y GRU se utilizan para predecir los precios de las criptomonedas y este estudio sigue en desarrollo. Por lo tanto, la construcción de un marco para predecir los precios de Bitcoin usando LSTM y GRU es valioso y contribuyente.

5. Metodología

Para llevar a cabo la propuesta metodológica se usó Python como software principal para la preparación de los datos y también para desarrollar las diferentes arquitecturas y modelos encontrados en la literatura, estos serán ejecutados en un computador con las siguientes características: Intel Core i5-1035G1, 8,00 GB de RAM, 500GB SSD y una tarjeta gráfica NVIDIA GeForce MX330.

Para el desarrollo de este trabajo se realizaron los siguientes pasos:

1. Obtención de los datos: Se obtuvieron los datos del sitio web Yahoo Finanzas. Los datos se encontraban en formato CSV con la data historica del precio del Bitcoin.
2. Análisis exploratorio de datos: Se realizó un análisis exploratorio de los datos para comprender su distribución, valores atípicos y realizar test estadísticos para determinar su posible estacionariedad.
3. Preparación de datos: Se aplicó normalización de los datos dada la escala en miles de dolares de la variable objetivo a modelar.
4. Entrenamiento de modelos: Se entrenaron los algoritmos LSTM, GRY y CNN con las ventanas de tiempo de 5,10 y 15 días hacia atrás. También se probaron modelos con variables como el EMA y el RSI.
5. Análisis de resultados: Se estudian las métricas obtenidas por los diferentes modelos entrenados, se comparan y se selecciona el mejor modelo.

En la figura 6 se ilustra el diagrama de flujo para realizar predicciones del precio el Bitcoin con el enfoque de este trabajo (izquierda) y por otro lado el enfoque tradicional (derecha), vemos que el enfoque tradicional tiene un paso muy relevante a la hora de poder predecir el precio del Bitcoin, el cual es verificar que los datos que se van a utilizar cumplan con los supuestos de estacionariedad, independencia y ausencia de tendencias con los cuales fue construido el modelo tradicional, en caso que no se cumplan el modelo no debería ser usado, dado que su ajuste se hizo bajo otras condiciones diferentes a las de producción. Por otro lado podemos ver que el enfoque propuesto posee la ventaja que no necesita cumplir con las condiciones de estacionariedad o de distribución, solo es necesario seleccionar de manera adecuada la ventana de tiempo a utilizar para realizar la predicción vía ACF (AutoCorrelation Function) como bien se encuentra en la literatura.

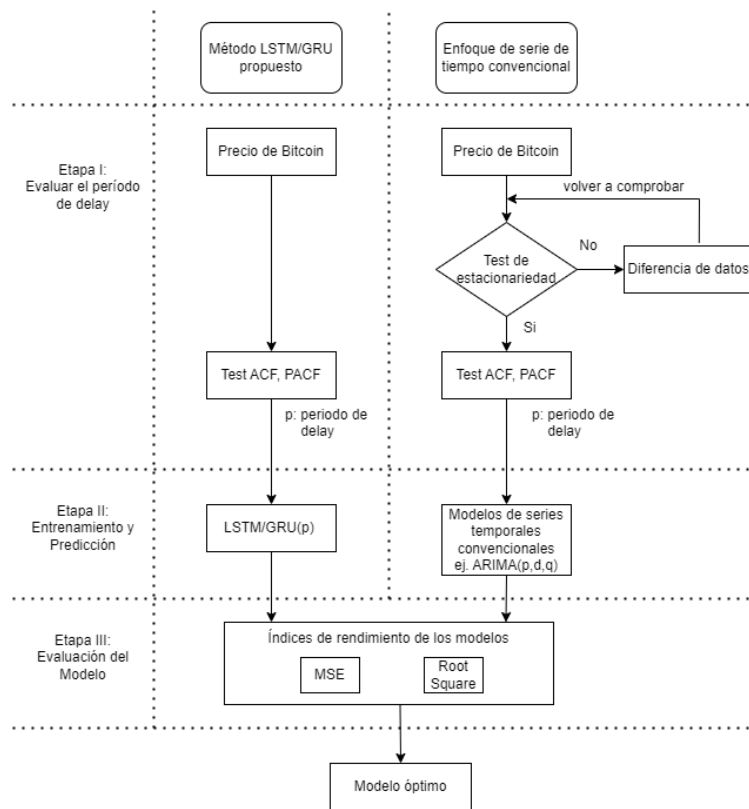


Figura 6: Diagrama para la predicción del precio del Bitcoin

5.1. Obtención de datos

Los datos históricos del precio del Bitcoin fueron obtenidos de Yahoo Finanzas el cual permite una obtención muy fácil de diferentes criptomonedas, en este caso descargamos el histórico diario desde el inicio del Bitcoin, enero del 2018 hasta enero de 2022.

5.2. Análisis exploratorio de datos

El conjunto de datos está conformado por las siguientes columnas:

- Date: Representa la fecha en la que fue capturada la información.
- Open: Indica el valor de precio de apertura en dolares del Bitcoin en ese día.
- Close: Indica el valor de precio de cierre del Bitcoin en ese día.
- High: Indica el valor de precio más alto registrado en ese día.
- Low: Indica el valor de precio más bajo registrado en ese día.
- Volume: Indica la cantidad de transacciones (venta o compra) realizadas en ese día.

	Date	Open	High	Low	Close	Volume
0	2014-09-17	465.864014	468.174011	452.421997	457.334015	21056800
1	2014-09-18	456.859985	456.859985	413.104004	424.440002	34483200
2	2014-09-19	424.102997	427.834991	384.532013	394.795990	37919700
3	2014-09-20	394.673004	423.295990	389.882996	408.903992	36863600
4	2014-09-21	408.084991	412.425995	393.181000	398.821014	26580100

Figura 7: Muestra de la tabla de datos

La Figura 8 muestra que el precio de cierre de Bitcoin alcanzó un máximo histórico el 06 de enero de 2018 el valor de Bitcoin fue \$17172,3, pero empezó a caer bruscamente. Cayó a \$5883,5 el 28 de junio del mismo año, y cayó en un 65,7% en aproximadamente medio año. Además, la figura 8 también muestra que la variabilidad, fluctuaciones o cambios en el precio de cierre del Bitcoin es muy alta (volatilidad). Este fenómeno se presenta en todos los instrumentos financieros como el dolar y euro, sin embargo en las criptomonedas se presencia mucho más dado que este mercado tiende a ser especulativo, no centralizado y el interés del público en un tiempo determinado genera estos cambios abruptos en el precio de las criptomonedas.



Figura 8: Serie temporal desde 2018 hasta 2022

Teniendo en cuenta los cambios drásticos y que esto puede introducir ruido al modelo, ya que no permite una correcta identificación de patrones y con el fin de mejorar la precisión del modelo se filtra la base desde marzo del 2021 hasta enero del 2022, ya que en ese intervalo al valor máximo de diferencia es de 20K, mientras que desde 2018 hasta 2021 se evidencia un cambio de hasta 40K. La base final quedó conformada con 333 registros, los cuales 239 fueron utilizados para entrenamiento y 94 para testeo.

La figura 10 muestra la distribución del precio de cierre (variable objetivo) entre el periodo de tiempo seleccionado anteriormente, como era de esperarse dada la figura 9 se observa que la distribución es muy variable dada la volatilidad que posee el fenómeno que queremos modelar. Al realizar el test de normalidad de Kolmogorov-Smirnov se obtuvo un p valor de $7,83e - 58$ y con un nivel de significancia de 0,05 concluimos que la distribución del precio de cierre no proviene de una distribución normal. También se aplicó la prueba de Dickey-Fuller para probar estacionariedad sobre la serie temporal y se obtuvo un p valor de 0,6879 y con una significancia de 0,05 podemos concluir que la serie no es estacionaria, por lo tanto, aplicar modelos estadísticos clásicos para este caso en particular no sería viable teniendo en cuenta que este es el supuesto fundamental en estos modelos.

La figura 11 muestra la función de autocorrelación del precio de cierre, esto con el fin de saber que tan amplia debe ser la ventana de tiempo para predecir nuestra variable objetivo.

Podemos ver que los retrasos 1 y 2 capturan una alta correlación, en [41] seleccionaron las dos pri-



Figura 9: Serie temporal filtrada

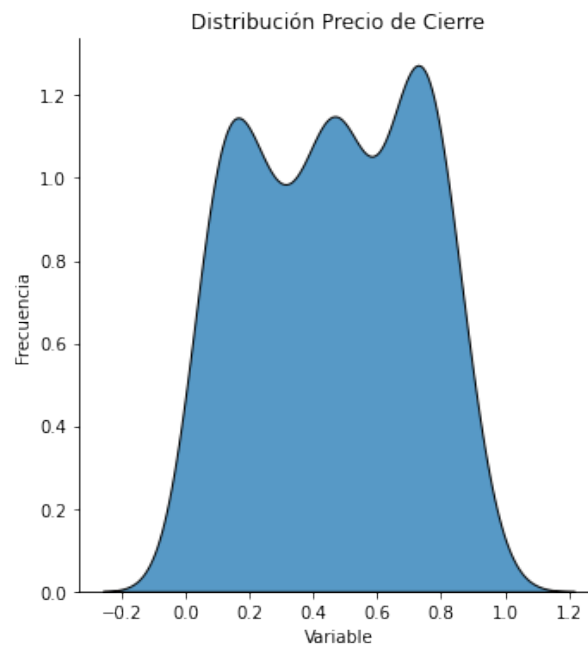


Figura 10: Distribución variable objetivo

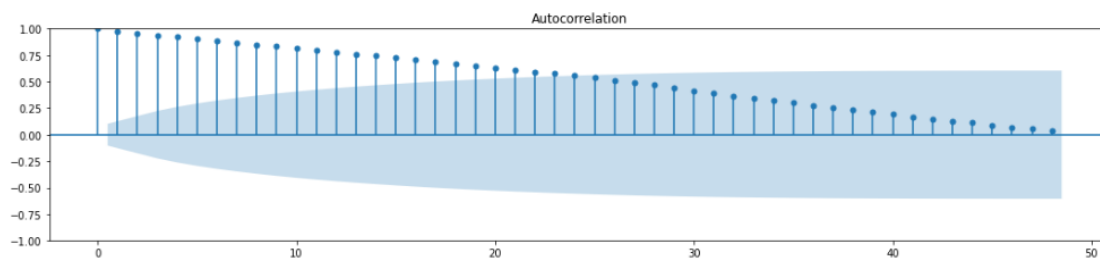


Figura 11: ACF del precio de cierre

meras y obtuvieron buenos resultados, sin embargo seleccionar solo dos hace que el modelo capture los comportamientos más evidentes y de mayor presencia en los datos dado que se toman tiempo muy recientes y no se tienen en cuenta efectos de días anteriores que se pueden reflejar de manera tardía, lo que hace que el modelo no sea robusto frente a cambios en los patrones de los datos dado que no puede recoger aquellos diferentes a los usuales, lo cual no es muy útil teniendo en cuenta la gran volatilidad que tienen las criptomonedas, en este caso, el Bitcoin. Por otro lado, seleccionar una ventana de tiempo

muy amplia hace que el modelo sea computacionalmente más costoso dado que se deben calcular tantos parámetros como el tamaño de la ventana de tiempo indique en el caso de modelos tradicionales. Por lo tanto, a partir de la gráfica de 11 se seleccionaron probar ventanas de tiempo de 5,10 y 15 días.

5.3. Preparación de datos

Como buena práctica realizamos una normalización de los datos, se utiliza la función `MinMaxScaler` de la librería `Scikit-Learn` de Python. Esto con el fin de realizar un mejor calculo de los pesos de las redes dado que la escala de nuestra variable objetivo es en miles de dolares y a la hora de realizar la optimización podría demorarse bastante dado que estos algoritmos están basados en distancia y por lo tanto se dificultaría la convergencia de los métodos.

5.4. Entrenamiento de modelos

Para el entrenamiento de los modelos se creo una función en Python que toma como ventana de tiempo 5,10 y 15 días hacia atrás para predecir el precio de cierre en el siguiente día. Se probaron los algoritmos de aprendizaje profundo LSTM, GRU, GRU con variables económicas las cuales fueron el EMA y el RSI, estas fueron obtenidas con la librería `TA-Lib` y `CNN`. Para el caso de modelos tradicionales se ajustó `ARIMA` con y sin suavizamiento exponencial. A continuación se describe la configuración de los algoritmos para las tres ventanas de tiempo definidas anteriormente.

Los modelos LSTM se configuraron de la siguiente manera:

La capa LSTM tiene 256 unidades, lo que significa que se generarón 256 nodos LSTM en la capa, una capa densa, función de perdida MSE, optimizador ADAM. batch size 16 y 200 épocas de entrenamiento.

Los modelos GRU se configuraron de la siguiente manera:

La capa GRU tiene 256 unidades, lo que significa que se generarón 256 nodos GRU en la capa, una capa densa, función de perdida MSE, optimizador ADAM. batch size 16, 200 épocas de entrenamiento y una tasa de aprendizaje de $1e-2$. Para el caso del modelo GRU con variables económicas se tuvieron en cuenta como variables explicativas el EMA y el RSI, las cuales fueron calculadas con la librería `TA-Lib`.

Los modelos CNN se configuraron de la siguiente manera:

Una capa convolucional de tamaño 64, `MaxPooling1D`, una capa densa de 50 unidades y por último la capa densa de tamaño 1, función de perdida MSE y optimizador ADAM.

Modelo ARIMA:

Se configuraron dos modelos ARIMA con una ventana de 5 días hacía atrás, sin suavizamiento y con suavizamiento exponencial, debemos tener en cuenta que la serie del precio de cierre del Bitcoin no supero el test de estacionalidad, por lo tanto, estos modelos se esperan que sean regulares.

5.5. Análisis de resultados:

Se obtuvieron los siguientes resultados para los diferentes modelos:

Los resultados de los modelo LSTM y GRU para predecir el precio de cierre de Bitcoin a 94 días (conjunto de testeo) se muestran en las Figuras 14c y 12b, respectivamente. Se puede ver en las dos figuras que los valores predichos de los dos modelos están cerca de los valores reales, y la dirección de la tendencia de los cambios también es muy consistente e incluso parecieran modelos muy similares en predicciones.

En la tabla 2 vemos las métricas obtenidas por los modelos autorregresivos basados en las arquitecturas LSTM, GRU, CNN y ARIMA en el conjunto de prueba. Se observa un compartamiento particular en las métricas de los modelos LSTM en donde el MAE va disminuyendo a medida que la ventana de

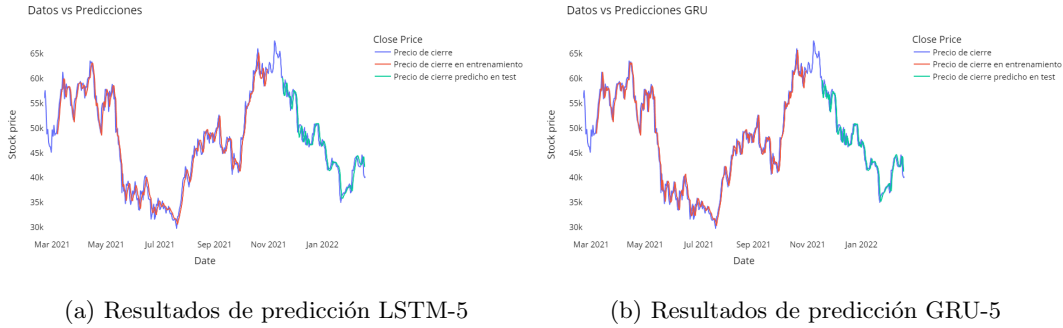


Figura 12: Predicción de los modelos

tiempo se hace más corta, esto se debe a que el algoritmos probablemente estaba recibiendo mucho ruido a medida que se iba más hacia atrás en el tiempo, se obtuvo mejor métrica con LSTM-5. Para el caso de GRU vemos que este no se ve afectado de manera considerable como si lo sufrió LSTM con el tamaño de la ventana, se mantuvo muy similar en las tres ventanas de tiempo, esto era de esperarse dado que como se encuentra en literatura esta arquitectura resultar ser más liviana a la hora de realizar los cálculos ya que solo utiliza dos compuertas, la compuerta de reinicio y de actualización. Para el caso de CNN tuvo rendimiento muy similar a la obtenidos con LSTM. Las CNN están destinadas principalmente al procesamiento de imágenes y datos espaciales en los que la ubicación espacial de los datos es crucial. Las CNN no tienen una estructura autoregresiva que les permita capturar relaciones temporales en series temporales secuenciales (Incluso haciendo el preprocesamiento de las ventanas de tiempo), a diferencia de las redes neuronales recurrentes (RNN) y modelos autoregresivos como ARIMA. Las CNN operan sobre regiones de datos específicas utilizando ventanas de contexto fijas y filtros convolucionales. Para la modelización autoregresiva, donde la dependencia temporal puede abarcar una longitud variable de pasos anteriores, esto no es adecuado. En el caso de GRU con variables económicas tuvo un desempeño muy malo dado, las variables explicativas escogidas no aportan suficiente información para predecir el valor de cierre del Bitcoin, sin duda los modelos autoregresivos obtenidos con LSTM o incluso CNN obtuvieron mejor desempeño, es importante recordar que los indicadores técnicos, como el EMA y el RSI, no son infalibles y no debería ser la única información a tener en cuenta para la toma de decisiones de inversión. Los mercados financieros son influenciados por una amplia gama de factores, incluyendo eventos geopolíticos, noticias, regulaciones, y más. Analizando los resultados de los modelos clásicos como ARIMA era de esperarse un rendimiento malo dado que la serie temporal del precio de cierre del Bitcoin no es estacionaria y por lo tanto el ajuste de estos modelos se vio seriamente afectado dado que este es de la hipótesis más fuertes que se tiene para este tipo de modelos.

La figura 13 muestra la distribución de los errores de los modelos ajustados con el algoritmo LSTM en sus diferentes ventanas, se observa que a medida que la ventana de tiempo va disminuyendo la distribución de los errores tiende a ser normal, esto indica que el modelo se ajusta de mejor manera y sus predicciones son confiable dado que no se estarían aleatorizando y que el modelo está entendiendo el fenómeno que se está describiendo.

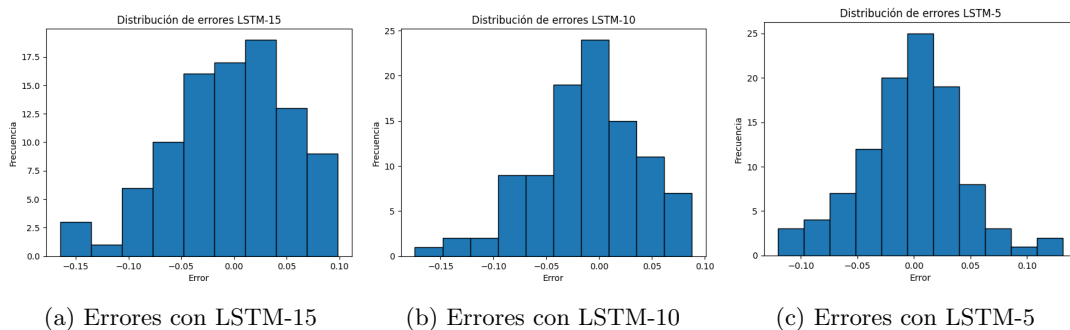


Figura 13: Distribución de los errores con LSTM

Modelo	MAE	MSE	R ²
LSTM-15	1725.15	4668949.65	0.88
LSTM-10	1561.58	4021529.90	0.92
LSTM-5	1298.03	2998882.36	0.95
GRU-15	1148.81	2427251.02	0.94
GRU-10	1165.07	2496796.72	0.95
GRU-5	1177.43	2655988.15	0.96
CNN-15	1888.95	5592265.82	0.86
CNN-10	1314.39	3605090.97	0.82
CNN-5	1643.20	4495474.98	0.93
ARIMA-5	1465.68	11785270.20	NA
ARIMA-5 EXP	47820.63	2371728017.21	NA
GRU5-VarEcono	179164728.67	3.21e+16	0.75

Cuadro 2: Métricas de los modelos en testeo

La figura 14 muestra la distribución de los errores obtenidos con el algoritmo GRU con las dife-

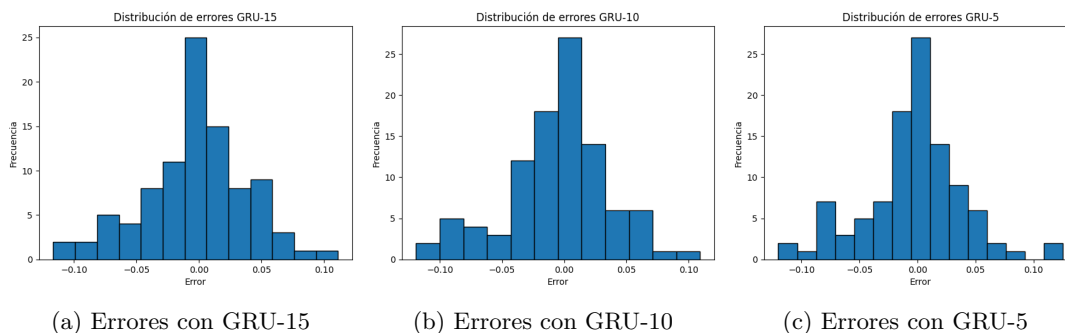


Figura 14: Distribución de los errores con GRU

rentes ventanas de tiempo, en este caso observamos que la distribución de los errores tiende a ser normal sin importar el tamaño del paso hacia atrás, a diferencia del algoritmo LSTM en donde si se vio seriamente afectado en distribución y métricas. Esto muestra que el algoritmo GRU es más robusto y posee un mejor rendimiento que el algoritmo LSTM frente a los cambios tan abruptos que puede llegar a tener el precio de cierre del Bitcoin.

5.6. Configuración de hiperparámetros:

Al analizar los resultados obtenidos en el conjunto de testeo de los diferentes modelos se selecciona el modelo GRU-5 para realizar la configuración de hiperparámetros mediante la técnica Grid-Search, dado que los tres modelos GRU tienen métricas muy similares y este utiliza menos ventanas de tiempo para realizar las predicciones.

Se probaron los siguientes valores:

- Unidades GRU: min value=64, max value=512, step=64
- learning rate: 1e-2, 1e-3, 1e-4
- batch size: min value=8, max value=64, step=8

- número de épocas: min value=50, max value=200, step=50
- Optimizador: ADAM
- Función de pérdida: MSE

Al ejecutar la configuración de hiperparámetros se obtuvieron los siguientes resultados:

Mejor configuración de hiperparámetros:

- Unidades en GRU: 384
- Learning rate: 0.001
- batch size: 40
- Número de épocas: 150

Del ajuste de hiperparámetros se cambió el número de épocas pasando de 256 a 384, batch size de 16 a 40 y se redujo el número de épocas de 200 a 150.

6. Conclusiones

Las principales conclusiones y contribuciones de este estudio se describen a continuación:

Primero, Se lograron configurar dos modelos modelos con buen poder predictivo como lo fueron LSTM y GRU, que incluso no cumpliendo con los supuestos de estacionariedad de la serie temporal lograron ajustarse bien al problema propuesto, lo cual brinda un mejor enfoque que el tradicional para modelar este tipo de fenómenos. También se descartó el uso de las CNN y acercamientos tradicionales como los modelos ARIMA.

Segundo, el marco propuesto en este trabajo permite utilizar nuevas técnicas de modelamiento de series temporales apoyadas en el gran avance que ha tenido el desarrollo del machine learning, en este caso exploramos las muy bien conocidas redes LSTM, las cuales como se pudo apreciar en la revisión de la literatura son muy usadas en este tipo de problemas, sin embargo se logró demostrar con nuestro conjunto de datos que las redes GRU tuvieron un mejor rendimiento en cuanto a la predicción, teniendo en cuenta que se configuraron de igual forma y además que se les aplico el mismo preprocesamiento de datos. Los valores pronosticados de los dos modelos están cerca de los reales y la dirección de la tendencia de los cambios también son coherentes.

Tercero, se logró mostrar que las redes GRU pueden llegar a ser mejores que las redes LSTM para este ejercicio de modelamiento, es importante recalcar esto dado que en la literatura actual el modelamiento de este tipo de fenómeno por lo general se hace con las redes LSTM, por lo tanto, se ha logrado mostrar un nuevo enfoque que puede llegar a ser más poderoso con las redes GRU.

Cuarto, se propone ampliar y mejorar el modelo de predicción del precio de cierre de Bitcoin utilizando algoritmos LSTM y GRU en trabajos futuros. Es importante mejorar el modelo para incluir variables económicas y exógenas relevantes, aunque estos algoritmos han demostrado ser eficaces en la captura de patrones temporales en los datos. Actualmente, eventos macroeconómicos, políticas gubernamentales, tasas de interés y tendencias de mercado globales afectan el precio del Bitcoin. Añadir estas variables como características de entrada al modelo permitirá una mejor comprensión y predicción de las fluctuaciones del precio del Bitcoin teniendo en cuenta su histórico y las dinámicas económicas y financieras que lo rodean.

7. REFERENCIAS

- [1] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.
- [2] Anupriya and S. Garg, “Autoregressive Integrated Moving Average Model based Prediction of Bitcoin Close Price,” 12 2018. [Online]. Available: <https://doi.org/10.1109/icssit.2018.8748423>
- [3] Y. Wang, C. Wang, A. Sensoy, S. Yao, and F. Cheng, “Can investors’ informed trading predict cryptocurrency returns? evidence from machine learning,” *Research in International Business and Finance*, p. 101683, 2022.
- [4] CoinMarketCap, “Cryptocurrency Prices, Charts And Market Capitalizations — CoinMarket-Cap.” [Online]. Available: <https://coinmarketcap.com/>
- [5] S. Freeda, T. C. E. Selvan, and I. Hemanandhini, “Prediction of bitcoin price using deep learning model,” in *2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2021, pp. 1702–1706.
- [6] M. Al Guindy, “Cryptocurrency price volatility and investor attention,” *International Review of Economics Finance*, vol. 76, no. C, pp. 556–570, 2021.
- [7] “Community-driven Bitcoin Statistics and Services.” [Online]. Available: <https://coin.dance/stats>
- [8] A. M. Khedr, I. Arif, P. R. P, M. El-Bannany, S. M. Alhashmi, and M. Sreedharan, “Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey,” *International Journal of Intelligent Systems in Accounting, Finance Management*, vol. 28, no. 1, pp. 3–34, 1 2021. [Online]. Available: <https://doi.org/10.1002/isaf.1488>
- [9] M. Al Guindy, “Cryptocurrency price volatility and investor attention,” *International Review of Economics & Finance*, vol. 76, pp. 556–570, 2021.
- [10] L. Kristoufek, “Bitcoin meets google trends and wikipedia: Quantifying the relationship between phenomena of the internet era,” *Scientific reports*, vol. 3, no. 1, pp. 1–7, 2013.
- [11] S. Sapuric and A. Kokkinaki, “Bitcoin is volatile! isn’t that right?” in *International conference on business information systems*. Springer, 2014, pp. 255–265.
- [12] P. K. Singh, A. K. Pandey, and S. Bose, “A new grey system approach to forecast closing price of Bitcoin, Bionic, Cardano, Dogecoin, Ethereum, XRP Cryptocurrencies,” *Quality Quantity*, vol. 57, no. 3, pp. 2429–2446, 7 2022. [Online]. Available: <https://doi.org/10.1007/s11135-022-01463-0>
- [13] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2010.
- [14] . S. J. Hochreiter, S., “Long Short-Term Memory,” *Neural Computation*, vol. 11, no. 15, p. 46, 11 2019. [Online]. Available: <https://doi.org/10.1162/neco.1997.9.8.1735>
- [15] P. Alvarez de Toledo, T. Saavedra, A. Crespo Marquez, F. Núñez, C. Usabiaga, and Y. Rebollo, “Autoregressive models and system dynamics. a case study for the labor market in spain.” 01 2002.
- [16] A. Aggarwal, I. Gupta, N. Garg, and A. Goel, “Deep learning approach to determine the impact of socio economic factors on bitcoin price prediction,” *2019 Twelfth International Conference on Contemporary Computing (IC3)*, pp. 1–5, 2019.
- [17] Y. Sovbetov, “Factors influencing cryptocurrency prices: Evidence from bitcoin, ethereum, dash, litcoin, and monero,” *Journal of Economics and Financial Analysis*, vol. 2, no. 2, pp. 1–27, 2018.
- [18] J. Y. Lim, K. M. Lim, and C. P. Lee, “Stacked bidirectional long short-term memory for stock market analysis,” in *2021 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAJET)*. IEEE, 2021, pp. 1–5.
- [19] L. S. Chong, K. M. Lim, and C. P. Lee, “Stock market prediction using ensemble of deep neural networks,” in *2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAJET)*. IEEE, 2020, pp. 1–5.

- [20] M. R. Islam and N. Nguyen, "Comparison of financial models for stock price prediction," *Journal of Risk and Financial Management*, vol. 13, no. 8, p. 181, 2020.
- [21] P. Koukaras, C. Nousi, and C. Tjortjis, "Stock market prediction using microblogging sentiment analysis and machine learning," in *Telecom*, vol. 3, no. 2. MDPI, 2022, pp. 358–378.
- [22] J. Park and Y.-S. Seo, "A deep learning-based action recommendation model for cryptocurrency profit maximization," *Electronics*, vol. 11, no. 9, p. 1466, 2022.
- [23] B. Manujakshi, M. G. Kabadi, and N. Naik, "A hybrid stock price prediction model based on pre and deep neural network," *Data*, vol. 7, no. 5, p. 51, 2022.
- [24] Z. Shahbazi and Y.-C. Byun, "Knowledge discovery on cryptocurrency exchange rate prediction using machine learning pipelines," *Sensors*, vol. 22, no. 5, p. 1740, 2022.
- [25] M. M. Patel, S. Tanwar, R. Gupta, and N. Kumar, "A deep learning-based cryptocurrency price prediction scheme for financial institutions," *Journal of information security and applications*, vol. 55, p. 102583, 2020.
- [26] E. Pintelas, I. E. Livieris, S. Stavroyiannis, T. Kotsilieris, and P. Pintelas, "Investigating the problem of cryptocurrency price prediction: a deep learning approach," in *IFIP International conference on artificial intelligence applications and innovations*. Springer, 2020, pp. 99–110.
- [27] P. Gao, R. Zhang, and X. Yang, "The application of stock index price prediction with neural network," *Mathematical and Computational Applications*, vol. 25, no. 3, p. 53, 2020.
- [28] S. Carta, A. Medda, A. Pili, D. Reforgiato Recupero, and R. Saia, "Forecasting e-commerce products prices by combining an autoregressive integrated moving average (arima) model and google trends data," *Future Internet*, vol. 11, no. 1, p. 5, 2018.
- [29] J. Abraham, D. Higdon, J. Nelson, and J. Ibarra, "Cryptocurrency price prediction using tweet volumes and sentiment analysis," *SMU Data Science Review*, vol. 1, no. 3, p. 1, 2018.
- [30] A. Dutta, S. Kumar, and M. Basu, "A gated recurrent unit approach to bitcoin price prediction," *Journal of risk and financial management*, vol. 13, no. 2, p. 23, 2020.
- [31] E. Sin and L. Wang, "Bitcoin price prediction using ensembles of neural networks," in *2017 13th International conference on natural computation, fuzzy systems and knowledge discovery (ICNC-FSKD)*. IEEE, 2017, pp. 666–671.
- [32] I. Madan, "Automated bitcoin trading via machine learning algorithms," 2014.
- [33] S. Velankar, S. Valecha, and S. Maji, "Bitcoin price prediction using machine learning," in *2018 20th International Conference on Advanced Communication Technology (ICACT)*, 2018, pp. 144–147.
- [34] I. Yenidoğan, A. Çayir, O. Kozan, T. Dağ, and Ç. Arslan, "Bitcoin forecasting using arima and prophet," in *2018 3rd International Conference on Computer Science and Engineering (UBMK)*. IEEE, 2018, pp. 621–624.
- [35] S. McNally, J. Roche, and S. Caton, "Predicting the price of bitcoin using machine learning," in *2018 26th euromicro international conference on parallel, distributed and network-based processing (PDP)*. IEEE, 2018, pp. 339–343.
- [36] A. Politis, K. Doka, and N. Koziris, "Ether price prediction using advanced deep learning models," in *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 2021, pp. 1–3.
- [37] H. Zhang and N. Rudholm, "Modeling and Forecasting Regional Gdp in sweden Using Autoregressive Models," 2013. [Online]. Available: <https://doi.org/10.1155/2021/1767708>

- [38] S. S. Pappas, L. Ekonomou, D. Karamousantas, G. E. Chatzarakis, S. K. Katsikas, and P. Liatsis, "Electricity demand loads modeling using AutoRegressive Moving Average (ARMA) models," *Energy*, vol. 33, no. 9, pp. 1353–1360, 9 2008. [Online]. Available: <https://doi.org/10.1016/j.energy.2008.05.008>
- [39] Z. Ismail, A. Yahya, and A. Shabri, "Forecasting gold prices using multiple linear regression method," *American Journal of Applied Sciences*, vol. 6, pp. 1509–1514, 2009.
- [40] X. Xu and W. Ren, "A Hybrid Model Based on a Two-Layer Decomposition Approach and an Optimized Neural Network for Chaotic Time Series Prediction," *Symmetry*, vol. 11, no. 5, p. 610, 5 2019. [Online]. Available: <https://doi.org/10.3390/sym11050610>
- [41] C.-H. Wu, C.-C. Lu, Y.-F. Ma, and R.-S. Lu, "A new forecasting framework for bitcoin price with lstm," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2018, pp. 168–175.