

INFORME FINAL DE PROYECTO



“DISEÑO DE UNA RED NEURONAL ARTIFICIAL LSTM MULTIPASOS PARA LA PREDICCIÓN DEL VALOR DE ACTIVOS EN EL MERCADO DE CRIPTOMONEDAS”

INVESTIGADOR PRINCIPAL DEL PROYECTO: Dr. Miguel Jiménez
Carrión Piura / Perú

Tipo de proyecto: Pequeño

Líneas de investigación:

Economía y negocios, Informática, electrónica y telecomunicaciones, Procesos industriales, Matemática y estadística

Sub líneas de investigación:

Economía de las finanzas y del desarrollo financiero, Ciencia de datos, Gestión de los procesos tecnológicos administrativos industriales para el incremento de la productividad empresarial, Inteligencia artificial

Agosto 2025

1. RESUMEN DEL PROYECTO

A nivel global, se observa un entorno económico desafiante para las empresas y entidades productivas, marcado por la incertidumbre en los mercados nacionales e internacionales. Factores como la inflación y variables socioeconómicas afectan la gestión de proyectos e inversiones. En este contexto, se valoran más las situaciones en las que las entidades poseen más dinero del necesario para sus transacciones diarias y buscan invertir este capital de manera efectiva, lo que se considera según He y Ciccone (2020) como exceso de liquidez, lo cual puede ser ventajoso si se utiliza adecuadamente. Este fenómeno toma relevancia en un entorno de bajas tasas de interés y alta volatilidad económica, donde los inversores deben decidir entre mantener su dinero estático, incurriendo en costos de oportunidad significativos, o invertir en mercados no confiables debido a la inflación (Bhosale, 2018).

Específicamente se abordó la investigación desde varias aristas, como por ejemplo el uso de varias herramientas de machine learning para poder comparar los desempeños de estas, tomando en cuenta siempre un enfoque en el modelo de red LSTM, la cual es una herramienta que siempre brinda mejores desempeños, en particular el modelo LSTM alcanzó un R^2 del 99.41% durante el primer día de predicción y mantuvo un desempeño superior al 97% hasta el 7mo día, reflejando consistencia incluso en predicciones más allá de una semana. Durante el entrenamiento, obtuvo un RMSE de \$1187.14 y un MAPE de 2.20%, manteniendo esta última métrica por debajo del 10% en validación. En predicciones a 7 días, logró un RMSE de \$5038.46 y un MAPE de 6.83%. Comparativamente, modelos como Máquinas de Soporte Vectorial (SVM) tuvieron un RMSE de \$7718.16 y un MAPE de 11.32%, Impulso de Gradiente Extremo (XGBoost) y Bosques Aleatorios mostraron resultados menos favorables, con errores significativos; por ejemplo, Bosques aleatorios tuvo un RMSE de \$ 19,492.41 y un MAPE de 30.35%, por otra parte XGBoost alcanzó un RMSE de \$17,849.66 y un MAPE de 27.74%. Estos hallazgos destacan la eficacia del LSTM como una herramienta clave para enfrentar los desafíos de predicción en mercados financieros altamente volátiles como el de las criptomonedas.

En relación a los entregables en primer lugar se presentó ante el “VI Congreso Multidisciplinario de Ciencias Aplicadas en Latinoamérica” realizada por la Universidad de Veracruz en México del 4 al 8 de noviembre del 20224, la conferencia “Modelamiento y Predicción del Valor de Activos en el mercado de las Principales Criptomonedas, utilizando Redes LSTM, Máquinas de Soporte Vectorial y ARIMA”, de la misma forma un artículo científico titulado “Machine Learning Algorithms in Predicting Prices in Volatile Cryptocurrency Markets” presentada a la revista Hightech and Innovation Journal, cuya categoría es Q2 indexada en Scopus; por otro lado de desarrollaron dos tesis de pregrado tesis de pregrado de los alumnos Josué Geraldo Usquiza Colán con su investigación "Utilización de las redes neuronales recurrentes y la influencia de la red social Twitter en la predicción de criptomonedas BTC, LTC Y ETH", y la tesis del alumno Jherson Smith Ruiz López con su investigación “Análisis Comparativo de las Redes Neuronales Recurrentes y el Modelo ARIMA en la Predicción de las Criptomonedas BTC, LTC Y ETH”; esta tesis dio origen a otro artículo científico titulado: Closing Price Prediction of Cryptocurrencies BTC, LTC, and ETH Using a Hybrid ARIMA-LSTM Algorithm presentada y publicada por la revista Hightech and Innovation Journal cuya categoría es Q2 indexada en Scopus.

Palabras clave: redes neuronales, criptomonedas, blockchain, predicción

2. EQUIPO DEL PROYECTO

Dr. Miguel Jiménez Carrión:
Investigador Principal

Jherson Smith Ruiz Lopez
Alumno participante

Josué Geraldo Usquiza Colán
Alumno participante

3. ANTECEDENTES O ESTADO DEL ARTE

Las criptomonedas están adquiriendo cada vez más relevancia en el mundo financiero y pueden considerarse como un mercado emergente. La baja barrera de entrada en la participación de compra de criptomonedas y la alta disponibilidad de datos del mercado de las criptomonedas lo convierten en un excelente tema de estudio, del cual es posible obtener información sobre el comportamiento de los mercados mediante la aplicación de análisis de sentimiento y técnicas de aprendizaje automático para la desafiante tarea de predecir el valor de las criptomonedas. Si bien ha habido algunos estudios previos, la mayoría de ellos se han centrado exclusivamente en el comportamiento de Bitcoin. En particular algunos estudios proponen el uso de herramientas comunes de aprendizaje automático y datos de redes sociales disponibles para predecir el movimiento de precios del mercado de criptomonedas Bitcoin, Ethereum, Ripple y Litecoin. Se comparó la utilización de redes neuronales (NN), máquinas de vectores de soporte (SVM) y bosques aleatorios (RF) mientras utilizamos elementos de Twitter y datos de mercado como características de entrada. Los resultados mostraron que es posible predecir los mercados de criptomonedas utilizando el aprendizaje automático y el análisis de sentimientos, donde los datos de Twitter por sí solos podrían usarse para predecir ciertas criptomonedas y que NN supera a los otros modelos (Valencia et al., 2019)

El modelamiento y predicción de series temporales constituye una tarea ardua y esencial para los procedimientos de optimización financiera. Numerosos estudios han sido elaborados con la finalidad de reducir la incertidumbre del inversor, mediante el pronóstico de precio de monedas y acciones. Sin embargo, el surgimiento de un nuevo tipo de monedas con características propias, conocidas como cryptocurrencias, plantea retos adicionales. En este sentido, otra investigación planteó analizar en qué medida las publicaciones en las redes sociales pueden capturar las expectativas colectivas de los inversores, y afectar el valor futuro de la moneda. El objetivo fue pronosticar el desempeño diario de un mercado en base a dos componentes: aquellos que definen el comportamiento de la criptomoneda en sí (volumen, valor de apertura, valor de cierre, valor máximo y valor mínimo) y las expectativas e interacciones del entorno, obtenidas de los tweets recolectados; para tal efecto, propusieron el uso de un tipo de red neuronal recurrente, conocida como “Long Short Term Memory” (LSTM). La metodología empleada para el preprocesamiento de los datos y la aplicación de esta técnica de pronóstico de series temporales permitió obtener una predicción con un MAPE de 34.92%; lo que indica que la representación de la variable de percepción en redes social no fue la pertinente y, por lo tanto, motiva nuevos trabajos con la finalidad de modelar esta variable mediante el uso de otras técnicas de NLP (Regal, 2019).

Recientemente, las criptomonedas se han convertido en un componente importante y conocido con potencial tanto económico como financiero. Desafortunadamente, la adquisición de Bitcoin no es sencilla debido a los negocios desiguales y a las importantes fluctuaciones de las tasas. Los enfoques tradicionales de previsión de precios han demostrado ser incapaces de proporcionar datos y soluciones adecuadas porque ahora los precios se pueden pronosticar en tiempo real. Específicamente una investigación recomienda una alternativa basada en aprendizaje automático para un prestamista hipotecario en función de los problemas destacados en la previsión del precio de Bitcoin. El sistema propuesto incluye un algoritmo de aprendizaje por refuerzo para la estimación y previsión de precios, así como un marco blockchain para un entorno eficiente y seguro. La predicción propuesta, en comparación con otras estrategias de última generación en este sector, demostró un mejor desempeño. En este sistema, la predicción propuesta alcanzó una mayor consistencia, en comparación con otros sistemas, con respecto a Monero (XMR), Litecoin (LTC), Oryen (ORY) y Bitcoin (BTC) (Mariappan, 2023).

La criptomoneda es un nuevo tipo de activo que ha surgido como resultado del avance de la tecnología financiera y ha creado una gran oportunidad para las investigaciones. La previsión de precios de las criptomonedas es difícil debido a la volatilidad y el dinamismo de los precios. En todo el mundo, se utilizan cientos de criptomonedas; sobre el particular Hamayel & Owda (2021), proponen tres tipos de algoritmos de redes neuronales recurrentes (RNN) utilizados para predecir los precios de tres tipos de criptomonedas, a saber, Bitcoin (BTC), Litecoin (LTC) y Ethereum (ETH). Los modelos muestran excelentes predicciones dependiendo del error porcentual absoluto medio (MAPE). Los resultados obtenidos de estos modelos muestran que la unidad recurrente cerrada (GRU) tuvo un mejor desempeño en la predicción para todos los tipos de criptomonedas que los modelos de memoria a corto y largo plazo (LSTM) y LSTM bidireccional (bi-LSTM). Por tanto, puede considerarse el mejor algoritmo. GRU presenta la predicción más precisa para LTC con porcentajes MAPE de 0,2454%, 0,8267% y 0,2116% para BTC, ETH y LTC, respectivamente. El algoritmo bi-LSTM presenta el resultado de predicción más bajo en comparación con los otros dos algoritmos, ya que los porcentajes de MAPE son: 5,990%, 6,85% y 2,332% para BTC, ETH y LTC, respectivamente. Así mismo los autores sostienen, que los modelos de predicción de su investigación representan resultados precisos cercanos a los precios reales de las criptomonedas. La importancia de tener estos modelos es que pueden tener importantes ramificaciones económicas al ayudar a los inversores y comerciantes a identificar las ventas y compras de criptomonedas.

Por su parte Yang et al. (2023), manifiesta que los precios de las criptomonedas tienen la característica de alta volatilidad, lo que tiene una resistencia específica a la predicción del precio de las criptomonedas. Por lo tanto, el método adecuado de predicción del precio de las criptomonedas puede ayudar a reducir el riesgo de inversión de los inversores. En este estudio, propusieron un método de predicción novedoso utilizando un modelo gris fraccionario (FGM (1,1)) para predecir el precio de la criptomoneda blockchain. En concreto, este estudio estableció el FGM (1,1) a través del precio de cierre de tres criptomonedas blockchain representativas (Bitcoin (BTC), Ethereum (ETH) y Litecoin (LTC)). Adoptó el algoritmo Particle Swarm Optimization (PSO) para optimizar y obtener el orden óptimo del modelo, realizando así una investigación de predicción sobre el precio de la criptomoneda blockchain. Para verificar la precisión predictiva de FGM (1,1),

tomamos principalmente MAPE, MAE y RMSE como criterios de evaluación y comparamos la precisión predictiva del modelo con FGM (1,1) a través de experimentos. Los resultados de la investigación indican que, dentro del rango de datos estudiado, la precisión predictiva del FGM (1,1) en el precio de cierre de BTC, ETH y LTC ha alcanzado un nivel "altamente preciso". Además, a diferencia del FGM (1,1), el FGM (1,1) supera la capacidad predictiva en los experimentos. Este estudio proporciona un nuevo método factible para la predicción de precios de criptomonedas blockchain. Tiene referencias específicas e información para departamentos gubernamentales, inversores e investigadores en teoría y práctica.

Las criptomonedas son activos de inversión muy volátiles y difíciles de predecir, sobre el particular, Sung et al. (2022), en su estudio utilizan varios datos de criptomonedas como características para predecir el precio de retorno logarítmico de las principales criptomonedas. La contribución original de un estudio es la selección de las características principales más influyentes para cada criptomoneda utilizando las características de volatilidad de la criptomoneda, derivadas de los modelos de heterocedasticidad condicional autorregresiva (ARCH) y heterocedasticidad condicional autorregresiva generalizada (GARCH), junto con el precio de cierre de la criptomoneda. Además, se buscó predecir el precio de rendimiento logarítmico de las criptomonedas mediante la implementación de varios tipos de modelos de series temporales. Con base en las características principales seleccionadas, el precio de retorno logarítmico de la criptomoneda se predijo mediante el modelo de predicción de series de tiempo de promedio móvil integrado autorregresivo (ARIMA) y el modelo de predicción de series de tiempo basado en redes neuronales artificiales. Como resultado de la predicción de precios de rendimiento logarítmico, los modelos de predicción de series temporales basados en redes neuronales mostraron un poder predictivo superior en comparación con el modelo tradicional de predicción de series temporales.

Así mismo con relación a la alta volatilidad de las criptomonedas han atraído una gran atención hasta ahora, y Bitcoin es la más notable; por ello el interés de Maleki et al. (2023), plantean predecir sus fluctuaciones aun cuando es una tarea desafiante. Aunque varias investigaciones utilizaron métodos estadísticos y económicos tradicionales para descubrir las variables determinantes de los precios de las criptomonedas, el progreso en el desarrollo de modelos de predicción para herramientas de toma de decisiones en técnicas de inversión aún se encuentra en sus primeras etapas. Muchos métodos diferentes de predicción de precios de criptomonedas cubren propósitos, como pronosticar un enfoque de un solo paso que se puede realizar mediante análisis de series de tiempo, redes neuronales y algoritmos de aprendizaje automático. Sin embargo, es necesario darse cuenta de la tendencia de una moneda a largo plazo. En este artículo, el objetivo fue investigar y pronosticar los precios de Bitcoin en función de otras tres monedas conocidas (es decir, Ethereum, Zcash y Litecoin), asumiendo poca información sobre los precios de Bitcoin utilizando algoritmos de aprendizaje automático. Además, propusieron un nuevo método para predecir el precio de Bitcoin considerando los precios de diferentes criptomonedas. Los resultados demostraron que Zcash tuvo el mejor desempeño en la predicción del precio de Bitcoin sin información sobre las fluctuaciones del precio de Bitcoin entre otras criptomonedas.

En el mismo sentido Jin & Li (2023), refiriéndose a la sustancial volatilidad y no estacionariedad de los precios de las criptomonedas, manifiestan que pronosticarlos se ha convertido en una tarea compleja dentro del ámbito del análisis de series de tiempo financieras y plantea su estudio en donde presenta un modelo de predicción híbrido innovador, VMD-AGRU-RESVMD-LSTM, que combina el marco de desintegración-integración con técnicas de aprendizaje profundo para una predicción precisa del precio de las criptomonedas. El proceso comienza descomponiendo la serie de precios de criptomonedas en un número finito de subseries, cada una caracterizada por patrones de volatilidad relativamente simples, utilizando el método de descomposición de modo variacional (VMD). A continuación, la red neuronal de unidad recurrente cerrada (GRU), en combinación con un mecanismo de atención, predice la secuencia de cada componente modal por separado. Además, la secuencia residual, obtenida después de la descomposición, sufre una descomposición adicional. Los componentes de la secuencia residual resultantes sirven como entrada para una red GRU (AGRU), que predice los valores futuros de la secuencia residual. En última instancia, la red neuronal LSTM integra las predicciones de componentes modales y residuales para generar el precio final pronosticado. Los resultados empíricos obtenidos para los datos diarios de Bitcoin y Ethereum muestran métricas de rendimiento prometedoras. El error cuadrático medio (RMSE) se informa como 50,651 y 2,873, el error absoluto medio (MAE) se sitúa en 42,298 y 2,410, y el error porcentual absoluto medio (MAPE) se registra en 0,394% y 0,757%, respectivamente. En particular, los resultados predictivos del modelo VMD-AGRU-RESVMD-LSTM superan a los de los modelos LSTM y GRU independientes, así como a otros modelos híbridos, lo que confirma su rendimiento superior en la previsión de precios de criptomonedas.

Las monedas virtuales han sido declaradas como uno de los activos financieros ampliamente reconocidos como monedas de cambio. Las operaciones con criptomonedas llamaron la atención de los inversores, ya que las criptomonedas pueden considerarse inversiones muy rentables. Para optimizar las ganancias de las inversiones en criptomonedas, es esencial una predicción precisa de los precios. En vista de que la predicción de precios es una tarea de series de tiempo, una investigación propone un modelo híbrido de aprendizaje profundo para predecir el precio futuro de la criptomoneda. El modelo híbrido integra una red neuronal convolucional unidimensional y una unidad recurrente cerrada apilada (1DCNN-GRU). Dados los datos del precio de las criptomonedas a lo largo del tiempo, la red neuronal convolucional unidimensional codifica los datos en una representación discriminativa de alto nivel. Posteriormente, la unidad recurrente cerrada apilada captura las dependencias de largo alcance de la representación. El modelo híbrido propuesto se evaluó en tres conjuntos de datos de criptomonedas diferentes, Bitcoin, Ethereum y Ripple. Los resultados experimentales demostraron que el modelo 1DCNN-GRU propuesto superó a los métodos existentes con los valores RMSE más bajos de 43,933 en el conjunto de datos de Bitcoin, 3,511 en el conjunto de datos de Ethereum y 0,00128 en el conjunto de datos de Ripple (Kang et al., 2022).

Sarıkaya y Aslan (2022) en su investigación "Deep Learning and Machine Learning-Based Sentiment Analysis on BitCoin (BTC) Price Prediction", tuvieron como objetivo examinar los sentimientos que las personas tienen sobre Bitcoin y proporcionar una visión general

de su efecto sobre el valor de la criptomoneda utilizando el poder de las arquitecturas de aprendizaje profundo y los métodos de aprendizaje automático. Para cumplir con este objetivo, se usó una metodología de tipo cuantitativa, utilizando algoritmos de aprendizaje automático y arquitecturas de redes neuronales convolucionales (CNN), memoria a corto y largo plazo (LSTM) y memoria a corto y largo plazo bidireccional (BiLSTM). Entre los resultados, se encontró que la polaridad general del sentimiento era positiva, con aproximadamente el doble de tweets positivos que negativos. Los modelos de aprendizaje profundo basados en la incrustación de palabras Glove demostraron tener mejor rendimiento que los basados en TF-IDF. En comparación con los algoritmos tradicionales de aprendizaje automático, como SVM (0.688), Regresión Lineal (0.694) y Random Forest (0.71), los modelos de aprendizaje profundo (especialmente el modelo BiLSTM) mostraron una precisión de 0.967, significativamente mayor usando la matriz de coocurrencia global palabra a palabra (Glove). Finalmente, se concluyó que los resultados experimentales de este estudio facilitan la detección de información poco fiable, inexacta e incompleta sobre Bitcoin y otras criptomonedas, ayudando así a las personas a tomar decisiones de inversión más informadas. Esta investigación servirá como referente para evaluar la precisión de las predicciones empleando el Error Porcentual Absoluto Medio (MAPE) y se aplican técnicas de Procesamiento de Lenguaje Natural (NLP) para entender la influencia social en los precios.

Desde la llegada de Bitcoin, el panorama de las criptomonedas ha visto el surgimiento de varias monedas virtuales que rápidamente han establecido su presencia en el mercado global. La dinámica de este mercado, influenciada por una multitud de factores que son difíciles de predecir, plantea un desafío para comprender plenamente sus ideas subyacentes. En mención a un artículo que propone una metodología para sugerir cuándo es apropiado comprar o vender criptomonedas, con el fin de maximizar las ganancias. A partir de grandes conjuntos de datos de mercado y redes sociales, usaron una metodología que combina diferentes técnicas estadísticas, de análisis de texto y de aprendizaje profundo para respaldar un algoritmo de comercio de recomendaciones. En particular, se explotó información adicional como la correlación entre las publicaciones en las redes sociales y las fluctuaciones de precios, la conexión causal entre los precios y el sentimiento de los usuarios de las redes sociales con respecto a las criptomonedas. Se llevaron a cabo varios experimentos con datos históricos para evaluar la efectividad del algoritmo comercial, logrando una ganancia promedio general del 194% sin tarifas de transacción y del 117% si se consideran las tarifas. En particular, entre los diferentes tipos de criptomonedas consideradas (es decir, alta capitalización, proyectos sólidos y monedas meme «las monedas meme o memecoin es una criptomoneda que se basa en el concepto de un meme y que nace como alternativa de inversión de dinero fácil»). Por tanto, un memecoin no tiene un valor intrínseco y rara vez es útil.), el algoritmo de negociación demostró ser muy eficaz para predecir las tendencias de precios de monedas meme influyentes, generando ganancias considerablemente mayores en comparación con otros tipos de criptomonedas (Belcastro et al., 2023)

Quiroga & Villalobos (2023), Manifiestan que con el análisis de datos históricos en criptomonedas sería posible generar escenarios que favorezcan el entendimiento del fenómeno en las criptomonedas y sostienen que serviría de apoyo para la gestión de portafolios; por tanto, plantean un estudio estadístico descriptivo e inferencial en un centenar de criptomonedas y proponen una metodología cuantitativa, prospectiva y

longitudinal. La hipótesis que se planteó es que al analizar los datos históricos sería posible generar escenarios que favorezcan el entendimiento del fenómeno en estudio; además, podría ser de apoyo para la gestión de portafolios. El periodo de análisis correspondió al 28 de abril de 2013 al 4 de agosto de 2022. La fuente de los datos fue la información contenida en la plataforma de Coingecko. El aporte teórico se sustentó en el estudio de un fenómeno emergente con implicaciones sociales que ha ganado impulso mundial, influenciado por el dinamismo tecnológico y agentes gubernamentales y privados. Los resultados de este artículo permitieron describir el comportamiento histórico del mercado de cien criptomonedas; también hizo posible prospectar escenarios e identificar correlaciones entre las criptodivisas, lo cual es importante para la creación de portafolios de inversión, desde un enfoque de diversificación de riesgos. En conclusión, este artículo genera un marco de referencia para el entendimiento de la evolución del mercado de criptomonedas a partir de una muestra por conveniencia de cien activos; igualmente, con el análisis de clúster, se realizó una clasificación de estas en función de la correlación, esto, desde un enfoque de teoría de portafolio, permitiría diversificar el riesgo.

Debido a que las predicciones de precios de criptomonedas altamente precisas son de suma importancia para inversores e investigadores; sin embargo, debido a la no linealidad del mercado de las criptomonedas, es difícil evaluar la naturaleza distintiva de los datos de series temporales, lo que genera desafíos a la hora de generar predicciones de precios adecuadas. Estos escenarios obligaron a realizar numerosos estudios sobre la predicción del precio de las criptomonedas utilizando diferentes algoritmos basados en Deep Learning (DL). Entre ellos se encuentran Seabe et al. (2023), quienes en su estudio proponen tres tipos de redes neuronales recurrentes (RNN), memoria a corto y largo plazo (LSTM), unidad recurrente cerrada (GRU) y LSTM bidireccional (Bi-LSTM) para predicciones de tipos de cambio de tres criptomonedas principales en del mundo, medida por su capitalización de mercado: Bitcoin (BTC), Ethereum (ETH) y Litecoin (LTC). Los resultados experimentales en las tres principales criptomonedas utilizando tanto la raíz del error cuadrático medio (RMSE) como el error porcentual absoluto medio (MAPE), mostrando que Bi-LSTM tuvo un mejor desempeño en la predicción que LSTM y GRU. Por tanto, puede considerarse el mejor algoritmo. Bi-LSTM presentó la predicción más precisa en comparación con GRU y LSTM, con valores MAPE de 0,036, 0,041 y 0,124 para BTC, LTC y ETH, respectivamente. El artículo sugiere que los modelos de predicción que se presentan en él son precisos a la hora de predecir los precios de las criptomonedas y pueden resultar beneficiosos para inversores y comerciantes.

En el mismo sentido y con el propósito de proporcionar un marco de comparación que supere las limitaciones de incertidumbre, volatilidad y dinamismo y que sea capaz de reproducir las predicciones no solo en las criptomonedas más comunes, sino que al mismo tiempo sea consistente y con capacidad de generalización Murray et al. (2023) [23], propusieron crear un marco de comparación que supere estas limitaciones y utilizar ese marco para realizar experimentos extensos en los que se pueda comparar el rendimiento de enfoques estadísticos, ML y DL ampliamente utilizados en la literatura para predecir el precio de cinco criptomonedas populares, es decir, XRP, Bitcoin (BTC), Litecoin (LTC), Ethereum. (ETH) y Monero (XMR). Más adelante los investigadores sostienen que son los primeros en proponer el uso del transformador de fusión temporal

(TFT) en su investigación. Además, ampliaron su investigación a modelos y conjuntos híbridos para evaluar si la combinación de modelos individuales aumenta la precisión de la predicción. Su evaluación muestra que los enfoques DL son los mejores predictores, particularmente el LSTM, y esto es consistentemente cierto en todas las criptomonedas examinadas. LSTM alcanza un RMSE promedio de 0,0222 y un MAE de 0,0173.

Samson (2024) presenta un estudio relevante en esta área, evaluando la eficacia de tres algoritmos de aprendizaje automático (ML) —Gradient Boosting (GB), Random Forest (RF) y Bagging— en la predicción de los precios de cierre diario de seis criptomonedas principales: Binance, Bitcoin, Ethereum, Solana, USD y XRP. A diferencia de los enfoques tradicionales que utilizan los precios de apertura, máximo y mínimo como características predictivas, este estudio adoptó una metodología innovadora al emplear precios rezagados como características de entrada. Este enfoque se basó en la suposición de que los precios rezagados proporcionan una mejor representación de la dinámica temporal de los precios de las criptomonedas, en comparación con los precios convencionales. El análisis utilizó un conjunto de datos históricos que abarcó desde 2015 hasta 2024, dependiendo de la criptomoneda, y dividió los datos en un conjunto de entrenamiento (80%) y un conjunto de prueba (20%) para evaluar el rendimiento de los algoritmos. Los resultados mostraron que el algoritmo GB obtuvo el mejor desempeño en la predicción de los precios de Bitcoin y Solana, mientras que RF fue más efectivo para predecir los precios de Ethereum, USD y XRP. Este hallazgo resalta las diferencias en la eficacia de los algoritmos según la criptomoneda y las características del mercado, sugiriendo que el uso de RF puede ser más adecuado en ciertos contextos, mientras que GB podría ofrecer mejores resultados en otros.

Fang et al. (2024) se inspiran en el éxito reciente de la aplicación del aprendizaje automático (ML) en la predicción del mercado de valores. En este estudio, analizan y presentan las características específicas del mercado de criptomonedas en un contexto de negociación de alta frecuencia. Específicamente, aplican un enfoque de ML para predecir la dirección de los cambios en el precio medio en el siguiente tick. Sus resultados indican que existen características universales en todas las criptomonedas que permiten que los modelos superen a aquellos específicos de cada activo. Además, demuestran que el uso de secuencias largas de puntos de datos no mejora las predicciones, lo que resalta la ineficiencia de alimentar modelos con conjuntos de datos extensos. También abordan el desafío técnico de diseñar un predictor ligero capaz de funcionar de manera efectiva con datos en vivo de los intercambios de criptomonedas. Para mejorar el rendimiento del modelo, presentan un nuevo método de reentrenamiento. Finalmente, examinan la compensación entre la precisión del modelo y la frecuencia de reentrenamiento en el contexto de la predicción de múltiples etiquetas. En general, sus hallazgos muestran que se pueden lograr resultados prometedores con datos en vivo, como lo demuestra una precisión constante del 78% en la predicción de los movimientos de los precios medios de Bitcoin frente al dólar estadounidense.

Kiranmai (2025) presenta un estudio importante que aborda la creciente popularidad de los mercados de criptomonedas, que, hasta 2023, incluían más de 23,000 criptomonedas y una valoración total de mercado de 870.81 mil millones de dólares. A pesar de su creciente importancia, las criptomonedas siguen siendo altamente susceptibles a la

volatilidad, lo que convierte la predicción de precios en un desafío crítico para los inversores que buscan tomar decisiones informadas. Este estudio tiene como objetivo desarrollar un modelo de pronóstico dinámico utilizando un enfoque de ensamble y evaluar la precisión de las predicciones para las 15 principales criptomonedas. Se evalúa la precisión de modelos estadísticos y econométricos después del ajuste de hiperparámetros, extrayendo información de estos modelos para construir un modelo de ensamble mediante algoritmos de aprendizaje automático. Específicamente, el estudio emplea Gradient-Boosted Regressor (GBR), Random Forest Regressor (RFR), Support Vector Regression (SVR) y Multi-Layer Perceptron (MLP), utilizando curvas de validación para optimizar los parámetros del modelo y mejorar la precisión de las predicciones. Los hallazgos del estudio revelan que cuando los movimientos de precios exhiben autocorrelación, tanto el modelo Autoregressive Integrated Moving Average (ARIMA) como el modelo de ensamble superan a otros métodos. Modelos como ARIMA, Regresión Lineal Simple (SLR), Random Forest (RF), Árbol de Decisión (DT), Gradient Boosting (GB) y Regresión Multi-Modelo (MLR) demostraron un buen desempeño con criptomonedas, sugiriendo que las tendencias, la estacionalidad y los patrones de precios históricos desempeñan un papel significativo en la predicción de precios. Notablemente, el enfoque MLR proporcionó predicciones más precisas para criptomonedas con mayor volatilidad y patrones de precios irregulares, destacando su potencial para la predicción en el impredecible panorama de los activos digitales.

Hossain (2024) explora el papel crítico de la predicción de series temporales en los mercados financieros, particularmente en la predicción de precios de activos y la orientación de decisiones de inversión. La volatilidad inherente a los mercados de criptomonedas, como Bitcoin (BTC) y Ethereum (ETH), complica la predicción debido a las fluctuaciones extremas de precios impulsadas por factores como el sentimiento del mercado, los cambios tecnológicos y las regulaciones gubernamentales. Tradicionalmente, la predicción en los mercados financieros se basaba en métodos estadísticos, pero a medida que los mercados se volvieron más complejos, la aparición de modelos de aprendizaje profundo como Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM) y, más recientemente, FinBERT-LSTM, ofreció un nuevo enfoque para capturar patrones y dinámicas intrincadas dentro de los datos. En respuesta a los desafíos que plantea la alta volatilidad de las criptomonedas, Mabsur propone un modelo híbrido que integra redes Bi-LSTM con FinBERT, un modelo conocido por sus capacidades de análisis de sentimiento. Este enfoque híbrido busca mejorar la precisión de la predicción al combinar modelos avanzados de predicción de series temporales con el análisis de sentimiento, un método que considera tanto los datos históricos de precios como los factores emocionales y psicológicos que influyen en el comportamiento del mercado. Este estudio llena una brecha significativa en la predicción financiera al ofrecer un modelo capaz de navegar por las complejidades y la imprevisibilidad de los mercados volátiles de criptomonedas. El modelo híbrido proporciona información valiosa para inversores y analistas, permitiéndoles tomar decisiones más informadas ante las condiciones de mercado impredecibles.

Zahidul et al. (2024) exploran la naturaleza dinámica y volátil del mercado de criptomonedas, que ha influido significativamente en los ecosistemas financieros a nivel mundial. En su investigación, se centran en la creciente importancia de las

criptomonedas, las cuales han pasado de ser activos digitales de nicho a convertirse en oportunidades de inversión convencionales, como Bitcoin y Ethereum. El estudio tiene como objetivo investigar la efectividad de diversos algoritmos de aprendizaje automático (ML) en la predicción de los precios de criptomonedas dentro del volátil mercado financiero estadounidense. Al identificar qué técnicas de ML proporcionan las predicciones más precisas y confiables bajo diferentes condiciones de mercado, la investigación contribuye a comprender las fortalezas y limitaciones de estos enfoques. El conjunto de datos utilizado para el análisis de la predicción de precios de criptomonedas incluye una amplia gama de puntos de datos, obtenidos de los principales intercambios de criptomonedas como Binance, Coinbase y Kraken, además de métricas comerciales esenciales para definir la dinámica del mercado. Los autores utilizan datos agregados de bases de datos financieras de renombre, como Coin-Market-Cap, Crypto-Compare y Yahoo Finance, asegurando una base sólida para los modelos de aprendizaje automático. Los modelos considerados en el estudio abarcan desde métodos lineales más simples hasta algoritmos de ensamble y de optimización por gradiente más complejos. Los autores evalúan el desempeño predictivo de estos modelos utilizando varias métricas, incluyendo Precisión, Recall, F1-Score, Error Absoluto Medio (MAE), Raíz del Error Cuadrático Medio (RMSE) y R-cuadrado. Entre los algoritmos probados, el modelo de Gradient Boosting mostró un rendimiento superior en términos de precisión, recall y F1-score. Además, los tres modelos evaluados exhibieron valores relativamente bajos de MAE y RMSE, lo que indica su efectividad en la predicción de movimientos de precios de criptomonedas.

Los hallazgos subrayan la importancia de los modelos de aprendizaje automático en el ámbito de la predicción de precios de criptomonedas, particularmente para inversores y actores del mercado financiero. La investigación destaca que las criptomonedas se han convertido en componentes clave de las carteras de inversión individuales e institucionales, así como en estrategias de trading. Al integrar modelos de ML en la gestión de inversiones, estos pueden proporcionar información valiosa sobre puntos de entrada y salida, diversificación de carteras y gestión de riesgos. Zahidul et al. (2024) sugieren que la consolidación de técnicas de aprendizaje automático dentro del sistema financiero marca un cambio significativo hacia la toma de decisiones basada en datos en el comercio de criptomonedas.

4. JUSTIFICACIÓN DE LA INVESTIGACIÓN

Durante los últimos 150 años se han presentado ciertos periodos de crisis económica, como la de 1929, la de los años 1970, la crisis del 2008 y la última del 2020 que origino una contracción de 6,2%, estos periodos de inflexión de un ciclo en donde la prosperidad económica pasa a recesión representan una perturbación en la estructura de la sociedad (Rapoport y Brenta, 2010), teniendo en cuenta esto las unidades productoras que dinamizan las económicas locales se ven afectadas directamente por lo que deben agilizar el desarrollo de nuevos proyectos o aplicar inversiones con un mayor riesgo, con la finalidad de poder recuperarse de manera eficiente en un periodo de tiempo corto y no caer en quiebra. Tal es el caso en donde países como la Federación Rusa aumento sus inversiones hasta en un 4,3% con la finalidad de encontrar un refugio económico (Chkalova et al., 2020). Las personas o empresas que busquen este tipo de refugio teniendo un exceso de liquidez o riqueza, deben decidir de forma acertada la fuente en

donde invertir para mejorar el rendimiento de su capital. Se debe considerar que el solo hecho de hacer una inversión en un mercado de renta variable, es decir tener una postura de compra y venta, ya es un problema que contiene alta complejidad, tal como indica Cantú (2005), ya que supone elementos aleatorios, una necesidad a satisfacer, decisiones que tomar, así como le evaluación de las posibles consecuencias de estas decisiones. Frente a lo mencionado entre las opciones se puede tomar los costos de oportunidad, entre los cuales se puede reservar el dinero, absorbiendo la naturaleza de las variaciones inflacionarias, también se puede realizar una inversión libre de riesgo al colocar el dinero en una entidad financiera, o finamente, tal como expresa Ames (2012), se puede invertir en un mercado de capitales, entonces esta última opción puede verse expresada en una forma alternativa como son las criptomonedas, ya que son un mercado respaldado por una tecnología emergente que expone el valor de las mismas mediante una valorización similar a la renta variable. Este mercado ha ido creciendo permitiendo a los agentes económicos invertir su dinero sin una restricción por lo que el sector financiero en cuanto al crecimiento de hedge funds (fondos especulativos), creció entre 1990 a 2007 de 610 a 10000 veces sus valores iniciales (Rapoport & Brenta, 2010).

En el escenario mencionado, esta forma de generar rendimientos a partir de la inversión entrando al mercado de las criptomonedas debe estar sujeta al buen uso de herramientas tecnológicas que aseguren que los rendimientos tendrán un comportamiento conveniente para el inversor, esta tarea resulta compleja en varios puntos entre los que destacan la obtención de datos de calidad, en donde dependiendo de su naturaleza y comportamiento deben ser analizados, evaluados y reprocesados para mitigar la mayor cantidad de ruido posible que perturbe las predicciones resultantes, este procedimiento puede contemplar el análisis de la temporalidad, completar datos faltantes o la eliminación de outliers; otro punto relevante para el análisis es la elección del algoritmo a emplearse en el proceso de predicción, ya que si bien es cierto que muchos modelos sirven para generar proyecciones, no todos obtienen datos que se ajusten a la real naturaleza de los datos, esto debido a que factores como la cantidad de datos, patrones atípicos o factores externos que influyen en los comportamientos de los mismos alteran la precisión, siendo las proyecciones poco congruentes con la realidad que se busca expresar mediante los modelos evaluados; finalmente otro punto importante es seleccionar dentro de un conjunto de modelos, cuál genera la mayor precisión y el menor error posible, por lo que es de vital importancia evaluar dentro de la estructura de los modelos un conjunto de parámetros hasta conseguir cuales son los valores que generan el mayor rendimiento y eficiencia en términos de error y precisión.

Usualmente se han usado herramientas como modelos de predicción se series de tiempo, tal como indica Agudelo et al. (2015), entre los más usados están los modelos ARIMA, GARCH, método de Monte Carlo, métodos híbridos, o modelos más complejos como son las redes neuronales. Estos últimos han demostrado su gran utilidad, ya que entre sus características más destacadas resaltan el hecho de que generan predicciones a partir de patrones internos de los datos, por lo que se puede generalizar el comportamiento global de una serie de tiempo aun frente la presencia de ruido, otro atributo destacable es que la naturaleza de los datos no necesariamente debe cumplir con los supuestos de normalidad e independencia, y finalmente la potencia no lineal para realizar las predicciones en base a las fronteras de decisión generadas que ayudan a encontrar predicciones con una precisión extremadamente alta.

La Universidad Nacional de Piura entre otras cosas está comprometida con la producción y creación científica, humanística, técnica, y con responsabilidad social por lo que cuenta con dieciséis líneas de investigación, entre las que se encuentra: Economía y negocios, Informática electrónica y telecomunicaciones, Matemática y estadística y Procesos industriales, entre otras. El número de investigadores RENACYT calificados por CONCYTEC es de 33, los cuales están vinculados a diferentes campos del conocimiento, la ciencia y la tecnología. Desde el marco de las líneas de investigación mencionadas nace este proyecto de investigación aplicada que lo que se buscó aportar a las metas establecidas para los objetivos del desarrollo sostenible N°9: Industria, innovación e infraestructura, utilizando los recursos con mayor eficacia, aumentar la investigación científica y mejorar la capacidad tecnológica de los sectores industriales y aumentando el número de personas que trabajen en investigación y desarrollo.

5. OBJETIVO GENERAL

Analizar las herramientas más relevantes utilizadas en la predicción de criptomonedas y hacer una comparativa y proponer una red neuronal artificial LSTM de tipo multistep para predecir el comportamiento de las criptomonedas.

6. OBJETIVOS ESPECÍFICOS

- a) Seleccionar y evaluar la data histórica de las criptomonedas para generar un dataset de calidad.
- b) Hacer una evaluación de las herramientas más relevantes utilizadas en la predicción de criptomonedas, utilizando las métricas RMSE, MAE y MAPE.
- c) Diseñar topologías de redes neuronales LSTM para la predicción del valor de las criptomonedas en varios pasos de tiempo
- d) Evaluar el desempeño de las distintas topologías de red, así como sus hiperparámetros para predecir el valor de las criptomonedas en varios pasos de tiempo.

7. METODOLOGÍA

La metodología a nivel teórico corresponde a una investigación de tipo aplicada, ya que a partir de teoría existente se aplicó a un caso real como es la predicción del valor de criptomonedas mediante series de tiempo, así mismo debido a que el rendimiento de las redes propuestas fueron medidas en su representación numérica así como el valor de las criptomonedas la investigación tiene un enfoque cuantitativo, y finalmente un diseño experimental correspondiente a la evaluación de hiperparámetros de la red que son modificados para encontrar el mejor desempeño.

Dentro de las actividades asociadas al desarrollo de la presente investigación en primer lugar se obtuvo la data histórica de los valores de renta variable de las criptomonedas, para esto se usaron bases de datos confiables como Yahoo! finances y/o investing, así mismo para el caso del análisis de sentimiento se usaron una base de datos de Twitter (Ahora red social "X"). Después de haber obtenido la data se realizó el análisis exploratorio de datos, en donde por medio de técnicas como la visualización y análisis de distribución

estadística de identificar los datos anómalos o faltantes dentro del data set inicial, en este punto se realizó la limpieza de datos, a partir de la identificación de outliers, y también evaluar el método de imputación, este procedimiento fue muy importante debido a que dependiendo de la cantidad de datos faltantes y al comportamiento de los mismos se procedió a evaluar el método de imputación más adecuado considerando que los datos deben mantener su distribución de probabilidad, y alguna medida de tendencia central que no difiera después de la imputación de datos. Después de haber realizado el análisis exploratorio de datos se procedió a diseñar la red neuronal artificial y programarla en Python, usando sus respectivas librerías de inteligencia artificial y análisis estadístico, para este caso específico las redes programadas son de tipo LSTM, y también se usaron otras herramientas de predicción de series de tiempo como XGBoost, ARIMA, Bosque Aleatorio y Maquinas de Soporte Vectorial (SVM). Antes de poder generar las predicción se segmentaron los datos en set de entrenamiento, set de prueba y de validación, así también se pasó por un proceso de escalamiento de datos para normalizarlos en una rango que pueda facilitar el aprendizaje de la red; Se busco la optimización de hiperparámetros, para tal fin se debe también definir los valores de óptimos, como el número de unidades LSTM en cada capa, el número de capas de la red, el ratio de aprendizaje, el optimizador a usar, el porcentaje de dropout, las métricas de error y los periodos de tiempo a predecir. Luego de haber generado las predicciones y obtenido las métricas de error se procedió a evaluar mediante un diseño experimental el comportamiento o rendimiento de la red más eficiente y así conocer que hiperparámetros pueden optimizarse dependiendo del nivel de significancia que se tenga de las fuentes de variación.

8. DESARROLLO DE LOS OBJETIVOS ESPECÍFICOS DEL PROYECTO

Para el desarrollo del proyecto principal se usaron los objetivos planteados como base fundamental para la ejecución del artículo científico y de las tesis por lo tanto fueron el punto inicial para los objetivos particulares de cada investigación, los resultados se muestran a continuación por objetivo:

a) Selección de la data histórica de las criptomonedas para generar un dataset de calidad.

Tabla 1: Datos históricos de criptomonedas

Criptomoneda	Fecha inicial de datos	Fecha final de datos	Cantidad de datos
Bitcoin	18/07/2010	11/07/2024	5108
Ethereum	10/03/2016	11/07/2024	3046
Polkadot	08/02/2021	11/07/2024	1250
Shiba Inu	12/05/2021	11/07/2024	1157
BNB	09/11/2017	11/07/2024	2437
Avalanche	03/01/2021	11/07/2024	1286

TRX	13/06/2018	11/07/2024	2221
Dogecoin	03/06/2017	11/07/2024	2596
Cardano	31/12/2017	11/07/2024	2385
XRP	22/01/2015	11/07/2024	3458
Solana	13/07/2020	11/07/2024	1453
Litecoin	24/08/2016	11/07/2024	2879

Por cada criptomoneda se obtuvo la cantidad total de valores diarios desde el inicio de las operaciones hasta el corte temporal del inicio de la ejecución del proyecto. Se hizo una imputación de datos con el algoritmo K-vecinos y los outliers se dejaron en el dataset principal debido a la naturaleza volátil de las criptomonedas.

Seudocódigo del algoritmo KNN

Importar librerías numpy, pandas, sklearn

Leer dataset con valores faltantes

Seleccionar características relevantes

Normalizar los datos

Definir número de vecinos $K = 3$

Mientras existan valores faltantes en el dataset:

Para cada fila con valores faltantes:

Encontrar las $K (=3)$ filas más cercanas sin valores faltantes

Calcular la media de los valores de las K filas más cercanas

Imputar el valor faltante con la media calculada

Guardar dataset con valores imputados

b) Evaluación de las herramientas más relevantes utilizadas en la predicción de criptomonedas, utilizando las métricas RMSE, MAE y MAPE.

Tabla 2. Comparación de modelos de regresión

Algoritmo	RMSE	MAE	MAPE
LSTM Networks	\$ 5,038.46	\$ 4,425.35	6.83%
XGBoost	\$17,849.66	\$17,687.05	27.74%
Random Forest	\$19,492.41	\$19,343.61	30.35%
SVM	\$7718.16	\$7192.06	11.32%

Se hizo un comparativo de los principales modelos de machine learning para evaluar su desempeño. Tal como se puede observar en la tabla 2 las redes neuronales LSTM

tuvieron un desempeño mucho mayor en comparación con los otros modelos en cuanto a las métricas RMSE, MAE, MAPE.

- c) Diseño de topologías de redes neuronales LSTM para la predicción del valor de las criptomonedas en varios pasos de tiempo

Tabla 3. Hiperparámetros de prueba para los modelos

Hiperparámetros	Configuraciones						
Número de pasos de entrada	1	7	15	21			
Capas LSTM	1	2	3	4			
Capas Dropout	1	2	3	4			
Tasa Dropout	0.1	0.2	0.3	0.4	0.5		
Unidades LSTM	50	100	200	350	500	750	1000
Tasa de aprendizaje	0.00001	0.00005	0.00009	0.00010	0.00050		
Tamaño de lote	20	35	70	100	150	250	
Épocas	30	40	50	70	100	200	
Porcentaje de entrenamiento	70%	80%	90%				

Mediante un ajuste de Hiperparámetros se diseñaron varios modelos de red LSTM para comparar cual brindaba el mejor desempeño.

- d) Evaluar el desempeño de las distintas topologías de red, así como sus hiperparámetros para predecir el valor de las criptomonedas en varios pasos de tiempo.

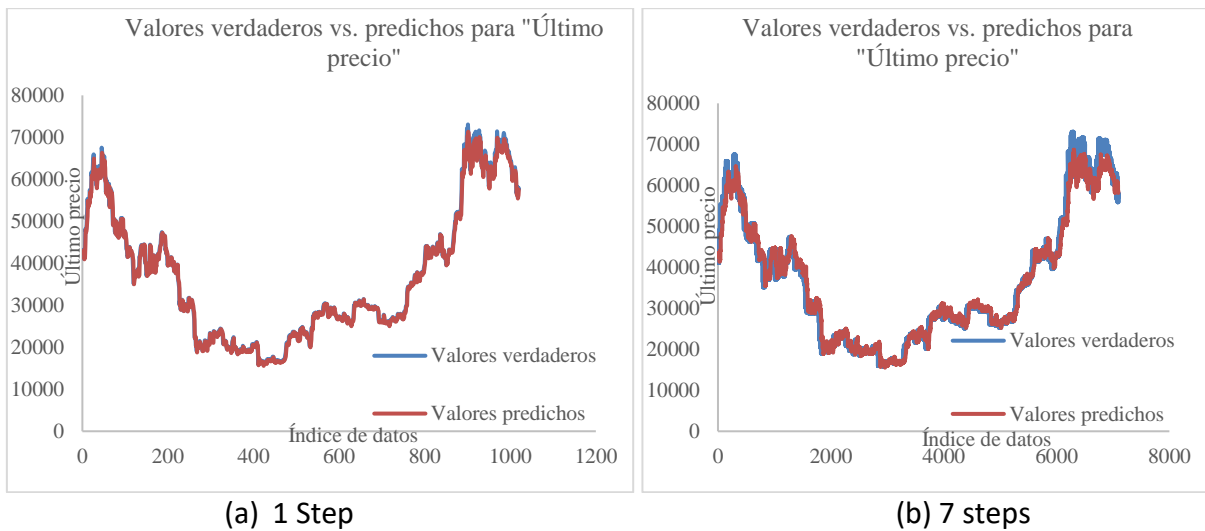
Tabla 7. Desempeño multipasos de la Red LSTM

Steps	RMSE	MAE	MAPE	R2
1	\$ 1,187.14	\$ 813.81	2.20%	99.41%
2	\$ 1,445.75	\$ 957.25	2.57%	99.17%
3	\$ 1,625.89	\$ 1,068.28	2.87%	98.95%
4	\$ 1,801.73	\$ 1,182.84	3.18%	98.71%
5	\$ 1,982.72	\$ 1,299.89	3.49%	98.44%
6	\$ 2,143.41	\$ 1,405.56	3.79%	98.17%
7	\$ 2,497.82	\$ 1,676.43	4.45%	97.52%

Se demuestra que el modelo de red LSTM es altamente eficiente al encontrarse métricas de error consistentes, considerando que para el primer día de predicción se alcanzó un R2 de 99.41% y se mantiene por encima del 97% después del step 7, lo cual evidencia la alta capacidad de predicción de la red LSTM con un buen ajuste a pesar de la incertidumbre generada por las predicciones mayores a 1 día, además otras métricas como el MAPE se mantienen por debajo del umbral del 10% que según la literatura es el máximo permisible [31], recalando la buena capacidad de la red en generar predicciones hasta una semana después con un buen desempeño.

Dentro del entrenamiento se pudo realizar una comparación del comportamiento del modelo al predecir el set de validación tanto para la predicción de 1 step y de 7 steps la cual se muestra gráficamente a continuación en la figura 1:

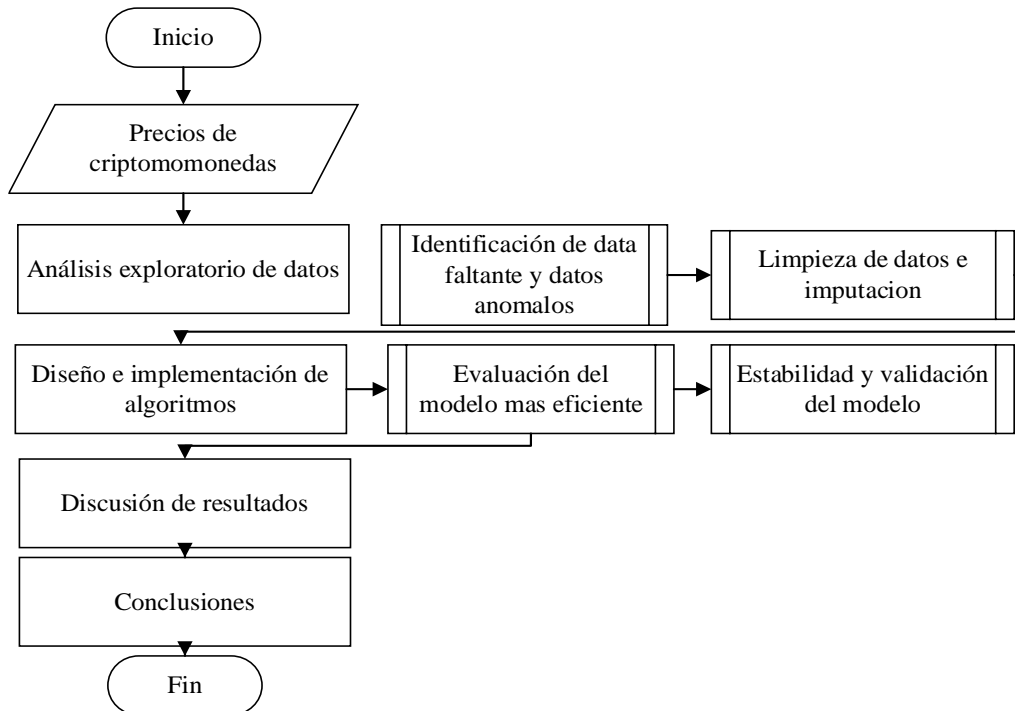
Figura 1. Comparación para Bitcoin entre red LSTM uni-step vs red LSTM multi-step



Se puede observar la nula o poca diferencia entre los datos reales y predichos para el modelo de 1 step, sin embargo, para el modelo donde se consideran 7 steps, la diferencia es aún más notoria y se puede diferenciar los datos reales y predichos, sin embargo el comportamiento es bastante similar lo cual evidencia el ajuste del modelo, lo que se justifica por las métricas encontradas, específicamente es el caso del coeficiente de determinación que se mantiene por encima del 97%.

En particular las metodologías de cada uno de los entregables se consideran detalles propios de cada uno de casos de estudio, artículo científico, conferencia y/o Tesis de alumnos durante el desarrollo del proyecto desarrolladas:

Artículo Científico: “Machine Learning Algorithms in Predicting Prices in Volatile Cryptocurrency Markets”

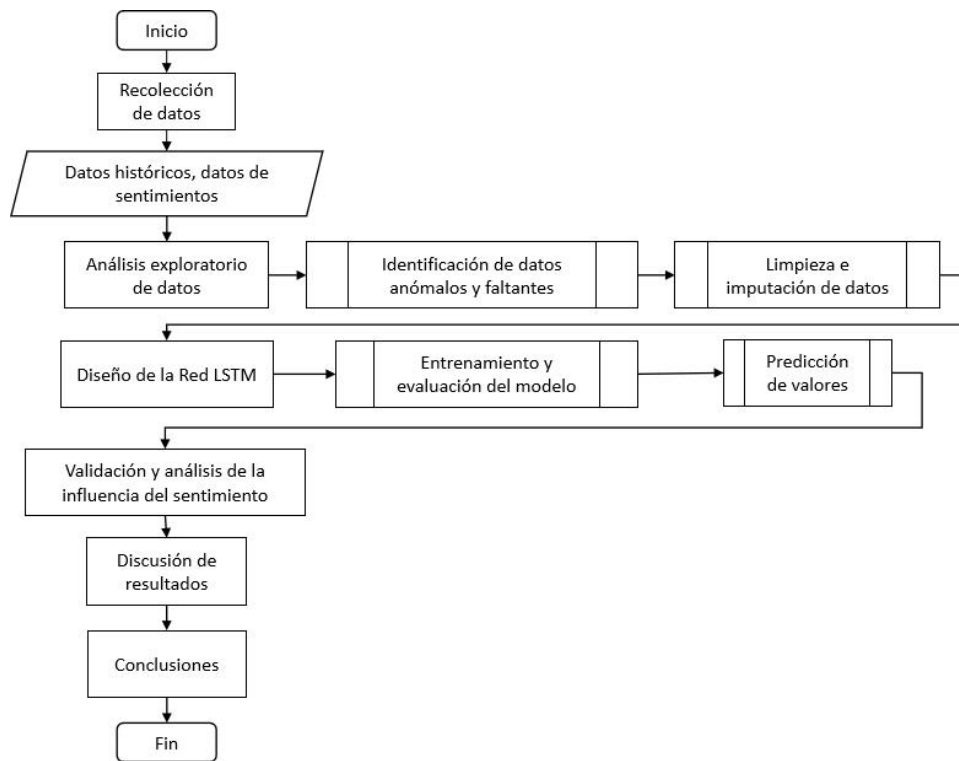


Conclusiones del artículo científico

- Las redes LSTM demostraron ser altamente efectivas en la predicción de precios de criptomonedas, específicamente para Bitcoin. Durante el primer día de predicción, el modelo alcanzó un R2 de 99.41%, manteniéndose por encima de 97% en los días sucesivos hasta el step 7. Esto indica una capacidad predictiva consistente y precisa, incluso para predicciones a más de una semana de anticipación.
- En términos de error, el modelo LSTM durante el entrenamiento alcanzó un RMSE de \$1187.14 y un MAPE de 2.20%, ambos valores dentro de los límites aceptables según la literatura. El MAPE se mantuvo por debajo del 10% en todos los pasos, lo que respalda la capacidad de la red para realizar predicciones precisas y confiables en el corto y mediano plazo.
- En comparación con otros algoritmos evaluados, el modelo LSTM demostró ser superior en las predicciones realizadas para los 7 días posteriores al conjunto de entrenamiento y validación. Este modelo alcanzó un RMSE de \$5038.46, inferior al RMSE de \$7718.16 del modelo MSV, al \$17,849.66 del modelo XGBoost. Estos resultados destacan la capacidad del LSTM para manejar series temporales no lineales y volátiles de manera más efectiva que los otros modelos. Además, el LSTM obtuvo un MAPE de 6.83%, menor que el 11.32% del algoritmo MSV, el 27.74% del modelo XGBoost y el 30.35% del Bosque Aleatorio. Este desempeño resalta la precisión del LSTM en la predicción de precios de criptomonedas, superando de manera significativa a los demás algoritmos evaluados.

- Los hiperparámetros óptimos para el modelo LSTM fueron un ratio de aprendizaje de 0.00009, 50 épocas y un tamaño de lote de 150. Estos valores permitieron obtener el mejor desempeño en términos de error, logrando una reducción significativa en el RMSE y manteniendo un bajo error relativo en las predicciones. Estos resultados sugieren que la optimización de los hiperparámetros es crucial para mejorar la precisión del modelo.

Tesis: "Utilización de las redes neuronales recurrentes y la influencia de la red social Twitter en la predicción de criptomonedas BTC, LTC Y ETH"

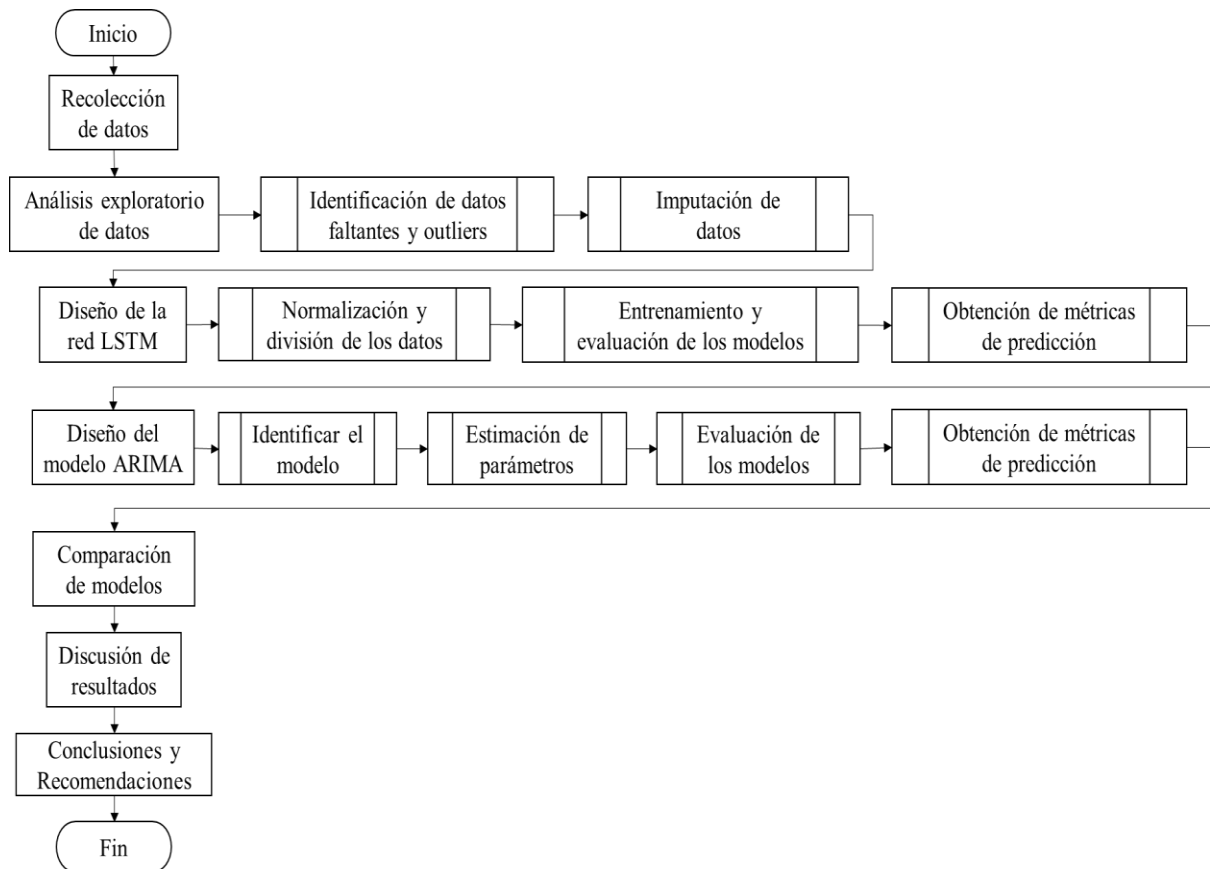


Conclusiones de la tesis

- El análisis exploratorio de las menciones en Twitter relacionadas con Bitcoin, Ethereum y Litecoin permitió caracterizar el volumen y la estructura de los datos recopilados, alcanzando un total de 862,968 tweets para BTC, 207,111 para ETH y 162,732 para LTC. Mediante técnicas de web scraping con Selenium, se superaron las restricciones de la API oficial de Twitter, obteniendo no solo el contenido de los tweets, sino también información complementaria como el usuario, la ubicación, la fecha de creación de la cuenta, el número de seguidores e interacciones, y la fuente de publicación. Los tweets obtenidos presentaron una gran variabilidad en su contenido, abarcando desde actualizaciones de precios y análisis de mercado hasta anuncios de adopción y discusiones generales sobre criptomonedas. Se identificó un uso frecuente de hashtags específicos (#BTC, #ETH, #LTC), referencias cruzadas a otras criptomonedas y presencia de cuentas automatizadas que publican información periódica. Este comportamiento refleja cómo se percibe y comparte el contenido relacionado con BTC, ETH y LTC, proporcionando una perspectiva valiosa sobre la dinámica y evolución del mercado de criptomonedas.

- El análisis de sentimientos aplicado a las menciones en Twitter sobre Bitcoin, Ethereum y Litecoin evidenció un patrón uniforme en las tres criptomonedas, a pesar de la diferencia en el volumen de menciones recopiladas. Se identificó un predominio de sentimientos neutrales, con ligeras inclinaciones hacia la positividad, reflejando promedios diarios de 0.2186 para BTC, 0.2187 para ETH y 0.2194 para LTC (siendo 0 el valor neutro). Aproximadamente el 39% de los tweets analizados fueron neutros y menos del 15% mostraron emociones negativas, lo que sugiere que la percepción del mercado sobre estas criptomonedas no se ve marcada por reacciones extremas. Estos resultados permitieron cuantificar la percepción del público hacia los cryptoactivos y establecieron la base para su integración en modelos predictivos. Se determinó estudiar el uso del análisis de sentimientos como variable predictora del precio último, ya sea de forma independiente o en combinación con datos históricos. Es así como para evaluar su influencia de manera precisa, se consideró necesario normalizar la cantidad de datos históricos de las criptomonedas a las menciones obtenidas en Twitter y analizar su relación con la volatilidad de los precios, garantizando así una comparación equitativa dentro del modelo.
- Para los modelos, se evaluaron múltiples configuraciones de hiperparámetros, incluyendo unidades LSTM (50-750), número de capas (1-4), tasas de aprendizaje (0.01 a 0.00009), tamaños de lote (20-500), capas de Dropout (0.2-0.5) y particiones de entrenamiento/validación (80%-20%). Estas configuraciones se aplicaron en tres modelos principales: dos univariados (basados en el análisis de sentimientos y el precio de cierre) y un modelo multivariado que combinó ambas variables. Los resultados iniciales indicaron que, para BTC y ETH, el modelo multivariado presentó un desempeño superior al integrar variables clave, mientras que, para LTC, el modelo univariado basado en el precio de cierre mostró resultados comparables. Asimismo, el análisis de varianza confirmó las configuraciones óptimas para los modelos predictivos, identificando como ideales una tasa de aprendizaje de 0.00009, 100 unidades LSTM y un tamaño de lote de 70. Estas combinaciones mejoraron la capacidad predictiva del modelo multivariado y univariado (precio último) en BTC, LTC y ETH, mientras que el modelo univariado basado en el análisis de sentimientos no resultó suficiente para ninguna de las tres criptomonedas.
- Los modelos basados en el precio de cierre (último) demostraron ser los más efectivos, alcanzando valores de MAPE del 2% y coeficientes de determinación (R^2) de 78% para BTC, 89% para ETH y 75% para LTC. En comparación, el modelo multivariado mostró un desempeño similar sin ofrecer ventajas adicionales significativas. Por otro lado, los modelos basados exclusivamente en el análisis de sentimientos presentaron resultados deficientes, con valores de R^2 negativos y errores relativos (MAPE entre 4% y 23%). Estos hallazgos confirman que los datos de sentimiento, aunque relevantes como variable complementaria, no son adecuados como principal predictor de precios. El modelo basado en el precio de cierre destacó por su simplicidad y alta precisión, consolidándose como la opción más eficiente para predecir las fluctuaciones del mercado de criptomonedas.

Tesis: “Análisis Comparativo de las Redes Neuronales Recurrentes y el Modelo ARIMA en la Predicción de las Criptomonedas BTC, LTC Y ETH”.

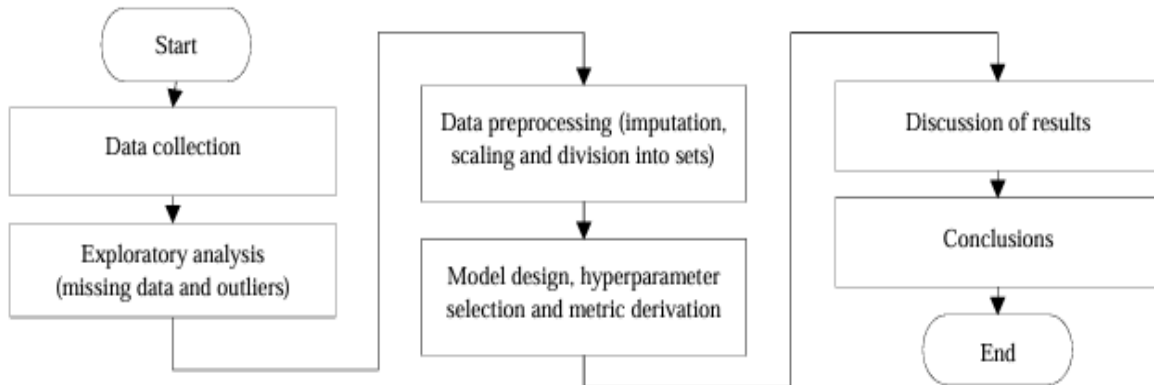


Conclusiones de la tesis

- Se determinó que las redes LSTM presentaron un gran desempeño en la predicción de los precios de las criptomonedas estudiadas sustentado en los valores favorables obtenidos para las 4 métricas de desempeño propuestas (RMSE, MAE, MAPE y R^2). Ello sugiere que las redes LSTM son una técnica lo suficientemente robusta para capturar patrones complejos en los precios de las criptomonedas debido a la volatilidad inherente de las mismas.
- Los modelos ARIMA en contraste presentaron un bajo desempeño en la predicción de los precios de las criptomonedas BTC, LTC y ETH, obteniendo valores desfavorables para las 4 métricas propuestas (RMSE, MAE, MAPE y R^2) e inferiores a los obtenidos por las redes LSTM. Ello implica que el modelo ARIMA no es suficientemente eficaz para la predicción de series temporales con patrones no lineales o complejos y es recomendable utilizar modelos más robustos para esta tarea.
- Los hiperparámetros más influyentes en las redes LSTM fueron la tasa de aprendizaje, el tamaño de lote, número de pasos de entrada, unidades LSTM y el porcentaje de entrenamiento, las configuraciones de hiperparámetros completas para las redes LSTM se muestran en la tabla 4.44. Con respecto a los modelos ARIMA, según el análisis de varianza aplicado, se observó que el factor “d” era el más influyente obteniendo mejores medias con una diferenciación.

Artículo Científico: «Closing Price Prediction of Cryptocurrencies BTC, LTC, and ETH Using a Hybrid ARIMA-LSTM Algorithm»

En la figura que sigue, se muestra el procedimiento seguido



Conclusiones del artículo científico

- El modelo híbrido ARIMA-LSTM demostró un desempeño competitivo al predecir los precios de criptomonedas, destacándose tanto en predicciones de un paso, múltiples pasos (5) y extrapolando fuera de la data histórica. En términos comparativos, el enfoque híbrido presentó métricas de precisión similares a las redes LSTM puras, superando consistentemente al modelo ARIMA. Las redes LSTM, debido a su capacidad para capturar dependencias temporales complejas, lograron un desempeño notable, alcanzando valores de MAPE cercanos al 2% y coeficientes de determinación (R^2) superiores al 92% para Litecoin (LTC), 98% para Ethereum (ETH) y 99% para Bitcoin (BTC). Estos resultados evidencian su capacidad para adaptarse a la alta volatilidad inherente a las criptomonedas, reconociendo patrones no lineales y tendencias ocultas.
- En contraste, el modelo ARIMA mostró limitaciones significativas, reflejadas en un MAPE superior al 10% y valores de R^2 por debajo del 70%, incluso negativos para LTC. Esto sugiere que su capacidad para modelar series temporales no lineales es insuficiente, especialmente en contextos de alta volatilidad. El desempeño inferior de ARIMA frente a LSTM y el modelo híbrido confirma que los métodos tradicionales de modelado de series temporales son inadecuados para datos complejos como los precios de las criptomonedas. En conclusión, el modelo híbrido ARIMA-LSTM se posiciona como una solución robusta, combinando las fortalezas de ambos enfoques. No obstante, se reconoce que la optimización de hiperparámetros y la exploración de arquitecturas alternativas podrían mejorar aún más su precisión.

9. RESULTADOS OBTENIDOS DEL PROYECTO

Actividad	Acción	Resultados	Nivel de cumplimiento
1. Seleccionar y evaluar la data histórica de las criptomonedas para generar un dataset de calidad	1.1 Adquisición de data histórica del valor de las criptomonedas	Data histórica de criptomonedas en formato .xlsx	100%
	1.2 Identificación de datos anómalos y faltantes de la data set	Data histórica de criptomonedas con anomalías identificadas en formato .xlsx	100%
	1.3 Normalización de outliers	Data histórica de criptomonedas sin outliers en formato .xlsx	100%
	1.4 Imputación de datos faltantes	Data histórica de criptomonedas completa en formato .xlsx	100%
2. Hacer una evaluación de las herramientas más relevantes utilizadas en la predicción de criptomonedas, utilizando las métricas RMSE, MAE y MAPE.	2.1. Diseño de diversas herramientas de inteligencia artificial para la predicción del valor de las criptomonedas.	Modelos definidos de herramientas para la predicción del valor de las criptomonedas.	100%
	2.2. Evaluación mediante métricas RMSE, MAE y MAPE de las herramientas inteligencia artificial para la predicción del valor de las criptomonedas.	Reporte de resultados de métricas RMSE, MAE y MAPE	100%
3. Diseñar topologías de redes neuronales LSTM para la predicción del valor de las criptomonedas en varios pasos de tiempo	3.1 Segmentación de data set en datos de entrenamiento, validación y prueba	Reporte de resultados de métricas de precisión de las redes neuronales entrenadas	100%
	3.2 Determinación del número de modelos a evaluar		100%
	3.3 Definición de los hiperparámetros de la red LSTM		100%
	3.4. Entrenamiento de la red neuronal artificial LSTM		100%
4. Evaluar el desempeño de las distintas topologías de red, así como sus hiperparámetros para predecir el valor de las criptomonedas en varios pasos de tiempo.	4.1 Guardar en un archivo las métricas de desempeño de las redes neuronales evaluadas	Red neuronal artificial LSTM con mejor desempeño y configuración	100%
	4.2 Generar el diseño experimental con las métricas de las redes neuronales LSTM		100%
	4.3 Evaluar las fuentes de variación y su significancia estadística		100%
	4.4 Evaluar las diferencias significativas de los niveles de los factores significativos		100%
	4.5 Definir la mejor configuración de hiperparámetros de la red neuronal artificial LSTM con mejor desempeño		100%
5.. Cierre del Proyecto	5.1 Elaboración de dos tesis	Tesis sustentadas *	100%
	5.2 Elaboración y Publicación de artículos científicos	Artículos aceptados	100%
	5.3 Conferencia Internacional	Conferencia evento Int.	100%

* Recién el 01/08/2025 fue aprobado el grado de Bachiller, por lo que el proceso de sustentación continuará su trámite cuando obtengan el grado de Bachiller.

10. CRONOGRAMA DE ACTIVIDADES

Actividad	Fecha de inicio	Fecha de fin	2024							2025					
			Junio	Julio	Agosto	Setiembre	Octubre	Noviembre	Diciembre	Enero	Febrero	Marzo	Abril	Mayo	Junio
Desarrollo de proyectos de tesis - Análisis comparativo de las redes neuronales recurrentes y el modelo ARIMA	21/06/2024	16/09/2024	■	■	■	■									
Desarrollo de proyectos de tesis - Utilización de las redes neuronales recurrentes y la influencia de Twitter	21/06/2024	16/09/2024	■	■	■	■									
Desarrollo de artículo científico - "Uso de algoritmos de Machine Learning para predecir los precios en los mercados altamente volátiles de las criptomonedas"	21/06/2024	28/11/2024	■	■	■	■	■	■							
Publicación y Disertación de un Artículo Científico en COMCAPLA 2025 México	05/11/2024	08/11/2024						■	■						
Publicación en revista indizada - "Uso de algoritmos de Machine Learning para predecir los precios en los mercados altamente volátiles de las criptomonedas"	30/12/2024	1/03/2025								■	■	■	■		
Desarrollo de informe final de tesis - Análisis comparativo de redes neuronales y ARIMA	28/11/2024	1/01/2025							■	■	■				
Desarrollo de informe final de tesis - Utilización de redes neuronales y Twitter	28/11/2024	1/01/2025							■	■	■				
Presentación de informe final de tesis - Análisis comparativo de redes neuronales y ARIMA	20/02/2025	20/02/2025										■			
Presentación de informe final de tesis - Utilización de redes neuronales y Twitter	20/02/2025	20/02/2025										■			
Desarrollo de artículo científico - Análisis comparativo de redes neuronales y ARIMA	20/02/2025	31/03/2025										■	■		
Publicación en revista indizada - Análisis comparativo de redes neuronales y ARIMA	1/04/2025	30/06/2025												■	■

11. EJECUCIÓN DEL GASTO:

PARTIDA PRESUPUESTAL	ITEM	CLASIFICADOR DE GASTO	DESCRIPCIÓN	CANTIDAD	UNIDAD	COSTO UNITARIO	TOTAL PRESUPUESTADO	Totales por Rubros	TOTAL EJECUTADO	SALDO AL 25/03/2025
Equipos de Laboratorio	1	2.6.32.31	Laptops de alto rendimiento	3	und	S/ 18,466.07	S/ 55,398.21	S/ 60,297.21	S/ 57,949.00	2,348.20
	2	2.6.32.31	Impresora	1	und	S/ 4,899.00	S/ 4,899.00			
Publicación	3	2.3.22.4	Publicación de Artículos Científicos	2*	paper	S/ 3,952.01	S/ 11,856.04	S/ 19,702.79	S/ 18,127.43**	1,575.40
	4	2.3.27.101	Taller de difusión de resultados	1	taller	S/ 7,846.75	S/ 7,846.75			
Servicios de Asesoría	5	23.27.1498	Servicios de asesoría y consultoría	4	und	S/ 2,500.00	S/ 10,000.00	S/ 10,000.00	S/ 10,000.00	0.00
Asistencia de Investigación Alumnos	6	23.27.299	Tesistas	2	tesis	S/ 2,500.00	S/ 5,000.00	S/ 5,000.00	S/ 4,375.00	625.00
Pasajes y viáticos	7	2.3.21.11	Conferencia Internacional	1	total	S/ 5,000.00	S/ 5,000.00	S/ 5,000.00	S/ 0.00***	5,000.00
		23.21.12								
* : Se publicaron 2 papers						TOTAL	100000	S/ 100,000.00	S/ 90,451.43	S/ 9,548.60
** : Se devolvió en tesorería en Caja S/. 1830.22, Según Recibo N°25-001169 del 21/03/2025							Saldo del Proyecto		S/ 9,548.60	
***: No se utilizó la Resolución N° 665-2024, solo autorizó la licencia con gose de haber no se pudo viajar, la conferencia se hizo virtual										
Sin embargo se financió el monto de la inscripción al Congreso por otra fuente de financiamiento (450 US\$)										

Fuente de Financiamiento: Recursos Determinados

Resolución de Autorización de Gastos: N° 629-R-2024

Resolución de Modificación Presupuestal: N° 505-R-2025

Bases del Concurso 2024: N° 161-CU-2024

Reglamento: N° 644-CU-2023

REFERENCIAS

1. Agudelo, A. P., López-Lezama, J. M., & Velilla, E. (2015). Predicción del Precio de la Electricidad en la Bolsa mediante un Modelo Neuronal No-Lineal Autorregresivo con Entradas Exógenas. *Informacion Tecnologica*, 26(6), 99–108. <https://doi.org/10.4067/S0718-07642015000600012>
2. Ames Santillán, J. C. (2012). Alternativas de diversificación internacional para portafolios de acciones de la Bolsa de Valores de Lima. 7, 13. <https://www.redalyc.org/articulo.oa?id=281623577003>
3. Belcastro, L., Carbone, D., Cosentino, C., Marozzo, F., & Trunfio, P. (2023). Enhancing Cryptocurrency Price Forecasting by Integrating Machine Learning with Social Media and Market Data. *Algorithms*, 16(12), 542. <https://doi.org/10.3390/a16120542>
4. Bhosale, J., & Mavale, S. (2018). Volatility of select crypto-currencies: A comparison of Bitcoin, ETH and Litecoin. *Annu. Res. J. SCMS, Pune*, 6(1), 132-141.
5. Cantú, L. S. (2005). Los mercados bursátiles y el paradigma de complejidad. *Contaduría y Administración*, 215, 73–87.
6. Chkalova, O., Efremova, M., Kiryushin, S., & Muranova, T. (2020). Management system for innovation and investment projects in trade. *IOP Conference Series: Materials Science and Engineering*, 940(1). <https://doi.org/10.1088/1757-899X/940/1/012060>
7. Fang, F., Chung, W., Ventre, C., Basios, M., Kanthan, L., Li, L., & Wu, F. (2024). Ascertaining price formation in cryptocurrency markets with machine learning. *The European Journal of Finance*, 30(1), 78–100. <https://doi.org/10.1080/1351847X.2021.1908390>
8. Hamayel, M. J., & Owda, A. Y. (2021). A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms. *AI*, 2(4), 477-477-496. <https://doi.org/10.3390/ai2040030>
9. He, Z., & Ciccone, S. (2020). Too much liquidity? Seemingly excess cash for innovative firms. *Financial Review*, 55(1), 121-144. <https://doi.org/10.1111/fire.12210>
10. Hossain, M. F. B., Lamia, L. Z., Rahman, M. M., & Khan, M. M. (2024). FinBERT-BiLSTM: A Deep Learning Model for Predicting Volatile Cryptocurrency Market Prices Using Market Sentiment Dynamics. *arXiv preprint arXiv:2411.12748*.
11. Jin, C., & Li, Y. (2023). Cryptocurrency Price Prediction Using Frequency Decomposition and Deep Learning. *Fractal & Fractional*, 7(10), 708-708-736. <https://doi.org/10.3390/fractalfract7100708>

12. Kang, C. Y., Lee, C. P., & Lim, K. M. (2022). Cryptocurrency Price Prediction with Convolutional Neural Network and Stacked Gated Recurrent Unit. *Data* (2306-5729), 7(11), 149-149-161. <https://doi.org/10.3390/data7110149>
13. Kiranmai Balijepalli, N.S.S., & Thangaraj, V. (2025). Prediction of cryptocurrency's price using ensemble machine learning algorithms. *European Journal of Management and Business Economics*. <https://doi.org/10.1108/EJMBE-08-2023-0244>
14. Maleki, N., Nikoubin, A., Rabbani, M., & Zeinali, Y. (2023). Bitcoin price prediction based on other cryptocurrencies using machine learning and time series analysis. *Scientia Iranica. Transaction E, Industrial Engineering*, 30(1), 285-285-301. <https://doi.org/10.24200/sci.2020.55034.4040>
15. Mariappan, L. T., Pandian, J. A., Kumar, V. D., Geman, O., Chiuchisan, I., & Năstase, C. (2023). A Forecasting Approach to Cryptocurrency Price Index Using Reinforcement Learning. *Applied Sciences* (2076-3417), 13(4), 2692-2692-2704. <https://doi.org/10.3390/app13042692>
16. Md Zahidul Islam, Md Shahidul Islam, Md Abdullah Al Montaser, Mohammad Abul Basher Rasel, Proshanta Kumar Bhowmik, Hossain Mohammad Dalim, & Laxmi Pant. (2024). Evaluating the effectiveness of machine learning algorithms in predicting cryptocurrency prices under market volatility: a study based on the USA financial market. *The American Journal of Management and Economics Innovations*, 6(12), 15–38. <https://doi.org/10.37547/tajmei/Volume06Issue12-03>
17. Murray, K., Rossi, A., Carraro, D., & Visentin, A. (2023). On Forecasting Cryptocurrency Prices: A Comparison of Machine Learning, Deep Learning, and Ensembles. *Forecasting*, 5(1), 196-209. <https://doi.org/10.3390/forecast5010010>
18. Quiroga Juárez, C. A., & Villalobos Escobedo, A. (2023). Escenarios para el mercado de criptomonedas a partir de un estudio estadístico. *Revista CEA*, 9(20), e2530. <https://doi.org/10.22430/24223182.2530>
19. Rapoport, M., & Brenta, N. (2010). La crisis económica mundial: ¿el desenlace de cuarenta años de inestabilidad? *Revista Problemas Del Desarrollo*, 163(41), 7–30.
20. Regal, A., Morzán, J., Fabbri, C., Herrera, G., Yaulli, G., Palomino, A., & Gil, C. (2019). Proyección del precio de criptomonedas basado en Tweets empleando LSTM. *INGENIARE - Revista Chilena de Ingeniería*, 27(4), 696-706.
21. Sarikaya, A., & Aslan, S. (2022). Deep Learning and Machine Learning-Based Sentiment Analysis on BitCoin (BTC) Price Prediction. *NATURENGS*, 3(2), 1-17.
22. Samson, T. K. (2024). Comparative analysis of machine learning algorithms for daily cryptocurrency price prediction. *Inf. Dyn. Appl.*, 3(1), 64–76. <https://doi.org/10.56578/ida030105>

23. Seabe, P. L., Moutsinga, C. R. B., & Pindza, E. (2023). Forecasting Cryptocurrency Prices Using LSTM, GRU, and Bi-Directional LSTM: A Deep Learning Approach. *Fractal and Fractional*, 7(2), 203. <https://doi.org/10.3390/fractalfract7020203>
24. Sung, S.-H., Kim, J.-M., Park, B.-K., & Kim, S. (2022). A Study on Cryptocurrency Log-Return Price Prediction Using Multivariate Time-Series Model. *Axioms*, 11, 448. <https://doi.org/10.3390/axioms11090448>
25. Valencia, F., Gómez-Espinosa, A., & Valdés-Aguirre, B. (2019). Price Movement Prediction of Cryptocurrencies Using Sentiment Analysis and Machine Learning. *Entropy*, 21(6), 589-589. <https://doi.org/10.3390/e21060589>
26. Yang, Y., Xiong, J., Zhao, L., Wang, X., Hua, L., & Wu, L. (2023). A Novel Method of Blockchain Cryptocurrency Price Prediction Using Fractional Grey Model. *Fractal & Fractional*, 7(7), 547-565. <https://doi.org/10.3390/fractalfract7070547>

ANEXOS

- **Código de Red LSTM**

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error, mean_absolute_error,
mean_absolute_percentage_error, r2_score
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import Callback
from sklearn.impute import KNNImputer
import matplotlib.pyplot as plt

plt.rcParams['font.family'] = 'serif'
# Cargar los datos
file_path = 'Data historica de criptomomedas.xlsx'
sheet_name = 'Datos históricos del Bitcoin'
#sheet_name = 'Datos históricos del Ethereum'
#sheet_name = 'Datos históricos del Polkadot'
#sheet_name = 'Datos históricos del Shiba Inu'
#sheet_name = 'Datos históricos del BNB'
#sheet_name = 'Datos históricos del Avalanche'
#sheet_name = 'Datos históricos TRX'
#sheet_name = 'Datos históricos del Dogecoin'
#sheet_name = 'Datos históricos del Cardano'
#sheet_name = 'Datos históricos del XRP'
#sheet_name = 'Datos históricos del Solana'
#sheet_name = 'Datos históricos del Litecoin'

df = pd.read_excel(file_path, sheet_name=sheet_name)
feature = ['Último']
imputer = KNNImputer(n_neighbors=3)
df[feature] = imputer.fit_transform(df[feature])

#Datos futuros para la comparación predictiva
datos_futuros=np.array([59209,60794,64782,65049,64089, 63980.5,66677.4])

# Función para dividir en conjuntos de entrenamiento y validación
def train_val_split(serie, tr_size=0.8, vl_size=0.2):
    N = serie.shape[0] # Número total de datos
    Ntrain = int(tr_size * N)
    Nval = N - Ntrain

    train = serie[0:Ntrain]
```

```

val = serie[Ntrain:Ntrain + Nval]

return train, val

# Dividir los datos
tr, vl = train_val_split(df['Último'])

# Graficar los subsets
fig, ax = plt.subplots(figsize=(16, 5))
ax.plot(tr, label='Train')
ax.plot(vl, label='Validation')
ax.set_title('Train and Validation Sets')
ax.set_xlabel('Index')
ax.set_ylabel('Value')
plt.legend()
plt.show()

# Función para crear datasets supervisados
def crear_dataset_supervisado(array, input_length, output_length):
    X, Y = [], []
    shape = array.shape
    if len(shape) == 1:
        fils, cols = array.shape[0], 1
        array = array.reshape(fils, cols)
    else:
        fils, cols = array.shape

    for i in range(fils - input_length - output_length): # Cambiar +1 a nada
        X.append(array[i:i + input_length, 0:cols])
        Y.append(array[i + input_length:i + input_length + output_length, -
1].reshape(output_length, 1))

    X = np.array(X)
    Y = np.array(Y)

    return X, Y

# Definir la longitud de entrada y salida
INPUT_LENGTH = 1 # días consecutivos a la entrada de la red
OUTPUT_LENGTH = 1 # Predecir un solo día futuro

# Crear datasets supervisados
x_tr, y_tr = crear_dataset_supervisado(tr.values, INPUT_LENGTH, OUTPUT_LENGTH)
x_vl, y_vl = crear_dataset_supervisado(vl.values, INPUT_LENGTH, OUTPUT_LENGTH)

# Asegurar que las formas sean correctas
y_tr_s = y_tr.reshape(-1, OUTPUT_LENGTH)

```

```

y_vl_s = y_vl.reshape(-1, OUTPUT_LENGTH)

print(f'Set de entrenamiento: x_tr: {x_tr.shape}, y_tr: {y_tr.shape}')
print(f'Set de validación: x_vl: {x_vl.shape}, y_vl: {y_vl.shape}')

# Función para escalar el dataset
def escalar_dataset(data_input):
    scaler_x = MinMaxScaler(feature_range=(-1, 1))
    scaler_y = MinMaxScaler(feature_range=(-1, 1))

    x_tr_s = scaler_x.fit_transform(data_input['x_tr'].reshape(-1,
data_input['x_tr'].shape[2])).reshape(data_input['x_tr'].shape)
    x_vl_s = scaler_x.transform(data_input['x_vl'].reshape(-1,
data_input['x_vl'].shape[2])).reshape(data_input['x_vl'].shape)

    y_tr_s = scaler_y.fit_transform(data_input['y_tr'].reshape(-1,
1)).reshape(data_input['y_tr'].shape)
    y_vl_s = scaler_y.transform(data_input['y_vl'].reshape(-1,
1)).reshape(data_input['y_vl'].shape)

    data_scaled = {'x_tr_s': x_tr_s, 'y_tr_s': y_tr_s.reshape(data_input['y_tr'].shape),
                    'x_vl_s': x_vl_s, 'y_vl_s': y_vl_s.reshape(data_input['y_vl'].shape)}
    return data_scaled, scaler_y

# Crear dataset de entrada para escalar
data_in = {'x_tr': x_tr, 'y_tr': y_tr, 'x_vl': x_vl, 'y_vl': y_vl}
data_s, scaler = escalar_dataset(data_in)

# Extraer datos escalados
x_tr_s, y_tr_s = data_s['x_tr_s'], data_s['y_tr_s']
x_vl_s, y_vl_s = data_s['x_vl_s'], data_s['y_vl_s']

#print(len(y_vl_s))

# Definir hiperparámetros
learning_rate = 0.00009
optimizer = Adam(learning_rate=learning_rate)

# Construir el modelo LSTM
model = Sequential([
    LSTM(500, return_sequences=True, input_shape=(INPUT_LENGTH, 1)),
    Dropout(0.3),
    LSTM(500, return_sequences=True),
    Dropout(0.3),
    LSTM(500),
    Dense(OUTPUT_LENGTH)
])

```

```

model.compile(optimizer=optimizer, loss='mean_squared_error')

# Clase para calcular y almacenar las métricas
class MetricsCallback(Callback):
    def on_train_begin(self, logs=None):
        self.rmse = []
        self.mae = []
        self.mape = []
        self.metrics_per_epoch = pd.DataFrame(columns=['Epoch', 'RMSE', 'MAE', 'MAPE'])

    def on_epoch_end(self, epoch, logs=None):
        y_pred = self.model.predict(x_vl_s)
        y_pred_inv = scaler.inverse_transform(y_pred.reshape(-1, OUTPUT_LENGTH))
        y_val_inv = scaler.inverse_transform(y_vl_s.reshape(-1, OUTPUT_LENGTH))
        rmse = np.sqrt(mean_squared_error(y_val_inv, y_pred_inv))
        mae = mean_absolute_error(y_val_inv, y_pred_inv)
        mape = mean_absolute_percentage_error(y_val_inv, y_pred_inv)
        self.rmse.append(rmse)
        self.mae.append(mae)
        self.mape.append(mape)
        new_row = pd.DataFrame({'Epoch': [epoch + 1], 'RMSE': [rmse], 'MAE': [mae], 'MAPE':
[mape]})
        self.metrics_per_epoch = pd.concat([self.metrics_per_epoch, new_row],
ignore_index=True)
metrics_callback = MetricsCallback()

# Entrenar el modelo
history = model.fit(x_tr_s, y_tr_s, epochs=100, batch_size=70, validation_data=(x_vl_s,
y_vl_s), verbose=1, callbacks=[metrics_callback])

# Después de predecir los valores en el conjunto de validación
y_pred_t = model.predict(x_vl_s)
y_pred = y_pred_t[-1]

# Asegurar que el escalado se aplique correctamente al conjunto de validación
y_val_inv = scaler.inverse_transform(y_vl_s.reshape(-1, 1)).reshape(-1)
y_pred_inv = scaler.inverse_transform(y_pred_t.reshape(-1, OUTPUT_LENGTH)).reshape(-
1)
# Recortar y_val_inv para que coincida con y_pred_inv
y_val_trim = y_val_inv[:len(y_pred_inv)]
# Calcular las métricas
rmse = np.sqrt(mean_squared_error(y_val_trim, y_pred_inv))
mae = mean_absolute_error(y_val_trim, y_pred_inv)
mape = mean_absolute_percentage_error(y_val_trim, y_pred_inv)
r2 = r2_score(y_val_trim, y_pred_inv)
print(f'RMSE: {rmse}')
print(f'MAE: {mae}')

```

```

print(f'MAPE: {mape*100}%')
print(f'R^2: {r2 * 100}%')

def predecir_futuro(x_actual, model, scaler, n_predicciones):
    predicciones_futuras = []
    x_input = x_actual[-1].reshape(1, INPUT_LENGTH, 1) # Tomar el último valor como inicio

    for _ in range(n_predicciones):
        y_pred_s = model.predict(x_input, verbose=0) # Predecir usando el modelo
        y_pred_inv = scaler.inverse_transform(y_pred_s) # Desescalar la predicción
        predicciones_futuras.append(y_pred_inv.flatten()[0]) # Almacenar predicción

        # Usar la predicción como nuevo input
        x_input = np.roll(x_input, -1) # Desplazar el input
        x_input[0, -1, 0] = y_pred_s[0, 0] # Añadir la predicción en el input

    return np.array(predicciones_futuras)

# Predecir futuros valores
future_predictions_inv = predecir_futuro(x_vl_s, model, scaler, OUTPUT_LENGTH)

# Calcular métricas para predicciones futuras
rmse_future = np.sqrt(mean_squared_error(datos_futuros[:OUTPUT_LENGTH],
future_predictions_inv))
mae_future = mean_absolute_error(datos_futuros[:OUTPUT_LENGTH],
future_predictions_inv)
mape_future = mean_absolute_percentage_error(datos_futuros[:OUTPUT_LENGTH],
future_predictions_inv)
r2_future = r2_score(datos_futuros[:OUTPUT_LENGTH],future_predictions_inv)
# Mostrar las métricas
print(f'RMSE (futuro): {rmse_future}')
print(f'MAE (futuro): {mae_future}')
print(f'MAPE (futuro): {mape_future*100}%')
#print(f'R^2 (futuro): {r2_future * 100}%')# Debido a que solo son 5 datos, esta metrica no
es recomendable

# Guardar comparaciones y métricas futuras en el archivo Excel
comparacion_futuros_df = pd.DataFrame({
    'Real_Futuro': datos_futuros[:OUTPUT_LENGTH],
    'Pred_Futuro': future_predictions_inv
})

metrics_future_df = pd.DataFrame({
    'RMSE_Futuro': [rmse_future],
    'MAE_Futuro': [mae_future],
    'MAPE_Futuro': [mape_future],
    # 'R^2_Futuro': [r2_future]

```

```

})

# Guardar las métricas y predicciones en un archivo Excel
result_df = pd.DataFrame({'True_Ultimo': y_val_inv})
pred_df = pd.DataFrame({'Pred_Ultimo': y_pred_inv})
metrics_df = pd.DataFrame({'RMSE': [rmse], 'MAE': [mae], 'MAPE': [mape], 'R^2': [r2]})
future_df = pd.DataFrame({'Future_Ultimo': future_predictions_inv})

with pd.ExcelWriter('resultados_predicciones_1dia.xlsx') as writer:
    result_df.to_excel(writer, sheet_name='True_Values', index=False)
    pred_df.to_excel(writer, sheet_name='Predictions', index=False)
    metrics_df.to_excel(writer, sheet_name='Metrics', index=False)
    future_df.to_excel(writer, sheet_name='Future_Predictions', index=False)
    metrics_callback.metrics_per_epoch.to_excel(writer, sheet_name='Metrics_Per_Epoch',
index=False)
    metrics_future_df.to_excel(writer, sheet_name='Future_Comparison', index=False)
# Mostrar estadísticas
statistics = df[['Último']].describe()
print(statistics)

# Graficar la pérdida durante el entrenamiento
plt.figure(figsize=(10, 5))
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Loss During Training')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.show()

# Graficar valores reales vs predichos
plt.figure(figsize=(10, 5))
plt.plot(y_val_trim, label='True Values')
plt.plot(y_pred_inv, label='Predicted Values')
plt.xlabel('Data Index')
plt.ylabel('Last Price')
plt.title('True vs Predicted Values for "Last Price"')
plt.legend()
plt.show()

# Graficar RMSE, MAE y MAPE durante el entrenamiento
plt.figure(figsize=(10, 5))
plt.plot(metrics_callback.rmse, label='RMSE')
plt.xlabel('Epoch')
plt.ylabel('RMSE')
plt.title('RMSE Progress During Training')
plt.legend()

```

```
plt.show()

plt.figure(figsize=(10, 5))
plt.plot(metrics_callback.mae, label='MAE')
plt.xlabel('Epoch')
plt.ylabel('MAE')
plt.title('MAE Progress During Training')
plt.legend()
plt.show()
```

```
plt.figure(figsize=(10, 5))
plt.plot(metrics_callback.mape, label='MAPE')
plt.xlabel('Epoch')
plt.ylabel('MAPE')
plt.title('MAPE Progress During Training')
plt.legend()
plt.show()
```

- **Código XGboost**

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error, mean_absolute_error,
mean_absolute_percentage_error, r2_score
from xgboost import XGBRegressor
from sklearn.impute import KNNImputer
import matplotlib.pyplot as plt
plt.rcParams['font.family'] = 'serif'

# Cargar los datos
file_path = 'Data historica de criptomomedas.xlsx'
sheet_name = 'Datos históricos del Bitcoin'
df = pd.read_excel(file_path, sheet_name=sheet_name)

# Imputar valores faltantes
feature = ['Último']
imputer = KNNImputer(n_neighbors=3)
df[feature] = imputer.fit_transform(df[feature])

# Datos futuros para la comparación predictiva
datos_futuros = np.array([59209, 60794, 64782, 65049, 64089, 63980.5, 66677.4])

# Función para dividir en conjuntos de entrenamiento y validación
def train_val_split(serie, tr_size=0.8):
    N = serie.shape[0]
```

```

Ntrain = int(tr_size * N)
train = serie[0:Ntrain]
val = serie[Ntrain:]
return train, val

# Dividir los datos
tr, vl = train_val_split(df['Último'])

# Crear datasets supervisados
def crear_dataset_supervisado(array, input_length, output_length):
    X, Y = [], []
    for i in range(len(array) - input_length - output_length):
        X.append(array[i:i + input_length])
        Y.append(array[i + input_length:i + input_length + output_length])
    return np.array(X), np.array(Y)

# Definir la longitud de entrada y salida
INPUT_LENGTH = 5 # Usamos 5 días previos
OUTPUT_LENGTH = 1 # Predecir un día futuro

# Crear datasets supervisados
x_tr, y_tr = crear_dataset_supervisado(tr.values, INPUT_LENGTH, OUTPUT_LENGTH)
x_vl, y_vl = crear_dataset_supervisado(vl.values, INPUT_LENGTH, OUTPUT_LENGTH)

# Escalar los datos
scaler_x = MinMaxScaler(feature_range=(-1, 1))
scaler_y = MinMaxScaler(feature_range=(-1, 1))

x_tr_s = scaler_x.fit_transform(x_tr)
x_vl_s = scaler_x.transform(x_vl)
y_tr_s = scaler_y.fit_transform(y_tr)
y_vl_s = scaler_y.transform(y_vl)

# Entrenar el modelo XGBoost
model = XGBRegressor(n_estimators=100, learning_rate=0.05, max_depth=5)
model.fit(x_tr_s, y_tr_s.ravel())

# Predecir en el conjunto de validación
y_pred_s = model.predict(x_vl_s)
y_pred_inv = scaler_y.inverse_transform(y_pred_s.reshape(-1, 1)).flatten()
y_vl_inv = scaler_y.inverse_transform(y_vl_s.reshape(-1, 1)).flatten()

# Calcular métricas
rmse = np.sqrt(mean_squared_error(y_vl_inv, y_pred_inv))
mae = mean_absolute_error(y_vl_inv, y_pred_inv)
mape = mean_absolute_percentage_error(y_vl_inv, y_pred_inv)
r2 = r2_score(y_vl_inv, y_pred_inv)

```

```

print(f'RMSE: {rmse}')
print(f'MAE: {mae}')
print(f'MAPE: {mape * 100}%')
print(f'R^2: {r2 * 100}%')

# Predicción futura
def predecir_futuro(model, x_input, scaler_x, scaler_y, n_predicciones):
    predicciones_futuras = []
    for _ in range(n_predicciones):
        x_scaled = scaler_x.transform(x_input.reshape(1, -1))
        y_pred_s = model.predict(x_scaled)
        y_pred_inv = scaler_y.inverse_transform(y_pred_s.reshape(-1, 1)).flatten()[0]
        predicciones_futuras.append(y_pred_inv)
        x_input = np.roll(x_input, -1)
        x_input[-1] = y_pred_inv
    return np.array(predicciones_futuras)

future_predictions = predecir_futuro(model, x_vl[-1], scaler_x, scaler_y, OUTPUT_LENGTH)

# Calcular métricas para predicción futura
rmse_future = np.sqrt(mean_squared_error(datos_futuros[:OUTPUT_LENGTH],
future_predictions))
mae_future = mean_absolute_error(datos_futuros[:OUTPUT_LENGTH], future_predictions)
mape_future = mean_absolute_percentage_error(datos_futuros[:OUTPUT_LENGTH],
future_predictions)
r2_future = r2_score(datos_futuros[:OUTPUT_LENGTH], future_predictions)

print(f'RMSE Futuro: {rmse_future}')
print(f'MAE Futuro: {mae_future}')
print(f'MAPE Futuro: {mape_future * 100}%')
print(f'R^2 Futuro: {r2_future * 100}%')

```

- **Código Random Forest**

```

import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error, mean_absolute_error,
mean_absolute_percentage_error, r2_score
from sklearn.ensemble import RandomForestRegressor
from sklearn.impute import KNNImputer
import matplotlib.pyplot as plt

# Cargar los datos
file_path = 'Data historica de criptomomedas.xlsx'
sheet_name = 'Datos históricos del Bitcoin'

```

```

# sheet_name = 'Datos históricos del Ethereum'
# sheet_name = 'Datos históricos del Polkadot'
# sheet_name = 'Datos históricos del Shiba Inu'
# sheet_name = 'Datos históricos del BNB'
# sheet_name = 'Datos históricos del Avalanche'
# sheet_name = 'Datos históricos TRX'
# sheet_name = 'Datos históricos del Dogecoin'
# sheet_name = 'Datos históricos del Cardano'
# sheet_name = 'Datos históricos del XRP'
# sheet_name = 'Datos históricos del Solana'
#sheet_name = 'Datos históricos del Litecoin'

df = pd.read_excel(file_path, sheet_name=sheet_name)

# Seleccionar solo la columna 'Último'
feature = ['Último']

# Imputar los valores faltantes
imputer = KNNImputer(n_neighbors=3)
df[feature] = imputer.fit_transform(df[feature])

# Mostrar estadísticas descriptivas de la columna 'Último'
statistics = df[feature].describe()
print(statistics)

# Normalizar los datos
scaler = MinMaxScaler(feature_range=(-1, 1))
scaled_data = scaler.fit_transform(df[feature])

# Definir una función para crear secuencias
def create_sequences(data, seq_length):
    X, y = [], []
    for i in range(len(data) - seq_length):
        X.append(data[i:i + seq_length])
        y.append(data[i + seq_length])
    return np.array(X), np.array(y)

# Definir la longitud de la secuencia
seq_length = 1 # Usar los últimos n meses para predecir el próximo mes

# Crear secuencias
X, y = create_sequences(scaled_data, seq_length)
X = X.reshape(X.shape[0], X.shape[1])

# Dividir los datos en conjuntos de entrenamiento y prueba
train_size = int(len(X) * 0.8)
X_train, X_test = X[:train_size], X[train_size:]

```

```

y_train, y_test = y[:train_size], y[train_size:]

# Construir el modelo Random Forest
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train.ravel())

# Hacer predicciones
y_pred = model.predict(X_test)

# Invertir la normalización de los datos escalados
y_test_inv = scaler.inverse_transform(y_test.reshape(-1, 1))
y_pred_inv = scaler.inverse_transform(y_pred.reshape(-1, 1))

# Calcular RMSE, MAE, MAPE y R^2
rmse = np.sqrt(mean_squared_error(y_test_inv, y_pred_inv))
mae = mean_absolute_error(y_test_inv, y_pred_inv)
mape = mean_absolute_percentage_error(y_test_inv, y_pred_inv)
r2 = r2_score(y_test_inv, y_pred_inv) * 100
print(f'RMSE: {rmse}')
print(f'MAE: {mae}')
print(f'MAPE: {mape}')
print(f'R^2: {r2} %')

# Predecir valores futuros
def predict_future(model, data, n_future, seq_length):
    predictions = []
    current_sequence = data[-seq_length:] # Obtener la última secuencia del conjunto de
    datos

    for _ in range(n_future):
        prediction = model.predict(current_sequence.reshape(1, -1))[0]
        predictions.append(prediction)
        current_sequence = np.append(current_sequence[1:], [[prediction]], axis=0)

    return np.array(predictions)

# Número de meses futuros a predecir
n_future = 1 # Pasos a predecir

# Predecir futuros valores
future_predictions = predict_future(model, scaled_data, n_future, seq_length)

# Invertir la normalización de las predicciones futuras
future_predictions_inv = scaler.inverse_transform(future_predictions.reshape(-1, 1))

# Guardar las métricas y predicciones en un archivo Excel
result_df = pd.DataFrame(y_test_inv, columns=['True_Ultimo'])

```

```

pred_df = pd.DataFrame(y_pred_inv, columns=['Pred_Último'])
metrics_df = pd.DataFrame({'RMSE': [rmse], 'MAE': [mae], 'MAPE': [mape], 'R^2': [r2]})
future_df = pd.DataFrame(future_predictions_inv, columns=['Future_Último'])

```

```

with pd.ExcelWriter('resultados_predicciones_random_forest.xlsx') as writer:
    result_df.to_excel(writer, sheet_name='True_Values', index=False)
    pred_df.to_excel(writer, sheet_name='Predictions', index=False)
    metrics_df.to_excel(writer, sheet_name='Metrics', index=False)
    future_df.to_excel(writer, sheet_name='Future_Predictions', index=False)

```

```

# Graficar todos los datos reales vs. todos los predichos
plt.figure(figsize=(10, 5))
plt.plot(y_test_inv, label='Valores Reales')
plt.plot(y_pred_inv, label='Valores Predichos')
plt.xlabel('Índice de datos')
plt.ylabel('Último')
plt.title('Valores Reales vs. Valores Predichos para "Último"')
plt.legend()
plt.show()

```

- **Codigo SVM**

```

import random
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error, mean_absolute_error,
mean_absolute_percentage_error, r2_score
from sklearn.svm import SVR
from sklearn.impute import KNNImputer
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
# Cargar los datos
file_path = 'Data historica de criptomonedas.xlsx'
sheet_name = 'Datos históricos del Bitcoin'
df = pd.read_excel(file_path, sheet_name=sheet_name)

# Seleccionar solo la columna 'Último'
feature = ['Último']

# Imputar los valores faltantes
imputer = KNNImputer(n_neighbors=3)
df[feature] = imputer.fit_transform(df[feature])

# Mostrar estadísticas descriptivas de la columna 'Último'
statistics = df[feature].describe()
print(statistics)

```

```

# Normalizar los datos
scaler = MinMaxScaler(feature_range=(-1, 1))
scaled_data = scaler.fit_transform(df[feature])

# Definir una función para crear secuencias
def create_sequences(data, seq_length):
    X, y = [], []
    for i in range(len(data) - seq_length):
        X.append(data[i:i + seq_length])
        y.append(data[i + seq_length])
    return np.array(X), np.array(y)

# Definir la longitud de la secuencia
seq_length = 1 # Usar los últimos n meses para predecir el próximo mes

# Crear secuencias
X, y = create_sequences(scaled_data, seq_length)
X = X.reshape(X.shape[0], X.shape[1])

# Dividir los datos en conjuntos de entrenamiento y prueba
#X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=True)
train_size = int(len(X) * 0.8)
X_train, X_test = X[:train_size], X[train_size:]
y_train, y_test = y[:train_size], y[train_size:]

# Configurar el modelo SVR con hiperparámetros específicos
svr = SVR(kernel='linear', C=60, epsilon=0.10, gamma=0.1) #sigmoid, linear, poly, rbf

# Entrenar el modelo
svr.fit(X_train, y_train.ravel())

# Hacer predicciones con el modelo
y_pred = svr.predict(X_test)

# Invertir la normalización de los datos escalados
y_test_inv = scaler.inverse_transform(y_test.reshape(-1, 1))
y_pred_inv = scaler.inverse_transform(y_pred.reshape(-1, 1))

# Calcular RMSE, MAE, MAPE y R^2
rmse = np.sqrt(mean_squared_error(y_test_inv, y_pred_inv))
mae = mean_absolute_error(y_test_inv, y_pred_inv)
mape = mean_absolute_percentage_error(y_test_inv, y_pred_inv)
r2 = r2_score(y_test_inv, y_pred_inv) * 100
print(f'RMSE: {rmse}')
print(f'MAE: {mae}')
print(f'MAPE: {mape}')

```

```

print(f'R^2: {r2} %')

# Predecir valores futuros
def predict_future(model, data, n_future, seq_length):
    predictions = []
    current_sequence = data[-seq_length:] # Obtener la última secuencia del conjunto de
datos

    for _ in range(n_future):
        prediction = model.predict(current_sequence.reshape(1, -1))[0]
        predictions.append(prediction)
        current_sequence = np.append(current_sequence[1:], [[prediction]], axis=0)

    return np.array(predictions)

# Número de meses futuros a predecir
n_future = 7 # Pasos a predecir

# Predecir futuros valores
future_predictions = predict_future(svr, scaled_data, n_future, seq_length)

# Invertir la normalización de las predicciones futuras
future_predictions_inv = scaler.inverse_transform(future_predictions.reshape(-1, 1))

# Guardar las métricas y predicciones en un archivo Excel
result_df = pd.DataFrame(y_test_inv, columns=['True_Ultimo'])
pred_df = pd.DataFrame(y_pred_inv, columns=['Pred_Ultimo'])
metrics_df = pd.DataFrame({'RMSE': [rmse], 'MAE': [mae], 'MAPE': [mape], 'R^2': [r2]})
future_df = pd.DataFrame(future_predictions_inv, columns=['Future_Ultimo'])

with pd.ExcelWriter('resultados_predicciones_svr_manual.xlsx') as writer:
    result_df.to_excel(writer, sheet_name='True_Values', index=False)
    pred_df.to_excel(writer, sheet_name='Predictions', index=False)
    metrics_df.to_excel(writer, sheet_name='Metrics', index=False)
    future_df.to_excel(writer, sheet_name='Future_Predictions', index=False)

# Graficar los valores reales y las predicciones
plt.figure(figsize=(10, 6))
plt.plot(y_test_inv, label='Valor Real')
plt.plot(y_pred_inv, label='Predicción') #, linestyle='--') #mjc
plt.xlabel('Índice de Muestra')
plt.ylabel('Último')
plt.title('Validación del modelo SVR: Valores Reales vs Predicciones')
plt.legend()
plt.show()

# Graficar todos los datos reales vs. todos los predichos

```

```
#plt.figure(figsize=(10, 5))
#plt.plot(y_test_inv, label='Valores Reales')
#plt.plot(y_pred_inv, label='Valores Predichos')
#plt.xlabel('Índice de datos')
#plt.ylabel('Último')
#plt.title('Valores Reales vs. Valores Predichos para "Último"')
#plt.legend()
#plt.show()
```