El análisis de series temporales es utilizado comunmente para hacer pronósticos de precios, incluyendo el de las criptomonedas. Implementar modelos que hagan predicciones precisas no es una tarea sencilla, especialmente para mercados tan volátiles. Se desarrolla un modelo híbrido ARIMA-LSTM que es evaluado(RECM) contra otros modelos ya utilizados(ARIMA, LSTM) para pronosticar el precio de criptomonedas. Se obtuvieron resultados que demuestran que el modelo híbrido se aproxima mejor que los otros modelos al valor real de los precios.

**Capítulo 1**

**Introducción**

En este trabajo se presenta la investigación e implementación de un modelo híbrido de series temporales utilizado para hacer pronósticos del precio de criptomonedas (Bitcoin y Litecoin). El desarrollo del proyecto implica la implementación de dos modelos (ARIMA y LSTM) que se han utilizado actualmente para hacer los pronósticos y se evalúan los resultados obtenidos por dichos modelos contra un tercer modelo híbrido ARIMA-LSTM que a su vez se implementa.

* 1. **Contexto del Problema**

Las criptomonedas son monedas digitales que operan de forma descentralizada, permitiendo que los usuarios lleven a cabo transacciones sin que éstas dependan de un intermediario. Las criptomonedas, y en especial Bitcoin que es la más popular, atraen cada vez más la atención haciendo que muchas personas opten por invertir en ellas y que ciertos negocios comiencen a aceptarlas como forma de pago [1].

Las criptomonedas se describen como rápidas, convenientes, libres de impuesto, e incluso revolucionarias [1], sin embargo, también son conocidas por su alta volatilidad debido a distintos factores como el tener un mercado relativamente pequeño, baja liquidez, gran cantidad de especulaciones respecto a su futuro, entre otros. Esta naturaleza volátil, hace que predecir los precios de criptomonedas sea una tarea complicada [2].

* 1. **Motivación**

La alta volatilidad de las criptomonedas hace que sean una inversión bastante riesgosa, sin embargo, actualmente se llevan a cabo más de 250, 000 transacciones de criptomonedas al día [3]. El desplome del valor de Bitcoin más reciente ocurrió el 12 de Marzo, reportando pérdidas de prácticamente el 50% para los inversionistas en tan solo un día [4]. A partir de datos como éstos, nace la motivación para realizar este trabajo, que busca explorar la posibilidad de acercarnos a pronósticos más precisos y así reducir el riesgo al invertir en criptomonedas.

* 1. **Objetivo del trabajo**

El objetivo de este trabajo es determinar si los pronósticos del precio de criptomonedas hechos por el modelo híbrido ARIMA-LSTM son más certeros que los pronósticos de los modelos ya hechos anteriormente.

**1.4 Estructura del trabajo**

Este trabajo está conformado por 7 capítulos que detallan la investigación hecha y posteriormente la implementación de los modelos.

En el capítulo 2 se habla del Estado del Arte. Se explican conceptos relevantes para el proyecto en general, así como otras soluciones propuestas para la problemática.

En el capítulo 3 se describe más a detalle la problemática actual y del porqué es importante seguir buscando soluciones alternas.

En el capítulo 4 se habla a profundidad de la propuesta de solución, explicando tanto los aspectos y conceptos relevantes para su desarrollo, como el funcionamiento de los modelos a implementar.

En el capítulo 5 se define la metodología de evaluación que se utiliza, así como la justificación para la elección de la misma.

En el capítulo 6 se muestran los resultados obtenidos al evaluar cada modelo. Se aprecian visualmente los pronósticos de precios hechos por cada modelo y los precios reales para compararlos.

En el capítulo 7 se presentan las conclusiones basadas en los resultados obtenidos, se retoman puntos importantes del trabajo y se plantean posibles trabajos futuros teniendo ya una base.

**Capítulo 2**

**Estado del arte**

En este capítulo se describen conceptos relevantes en el área. Posteriormente se hace mención a los modelos ARIMA y LSTM, que se han implementado para hacer pronósticos del precio de las criptomonedas, así como los resultados que obtuvieron. Analizar estos trabajos permite encontrar un área de oportunidad a la cual se puede contribuir.

**2.1 Contexto del área**

**2.1.1 Series de tiempo**

**2.1.1.1 Definición**

Una serie de tiempo es un conjunto secuencial de datos medidos típicamente en tiempos sucesivos. Las medidas que se toman durante cierto periodo se organizan en orden cronológico [5]. Se define matemáticamente una serie de tiempo como:

{X1, X2,… X(t)}, t = 1, 2, 3,…

t: Representa el tiempo transcurrido

X(t): Es el valor del dato medido durante un tiempo en específico.

Las series de tiempo se denominan univariantes si los registros que se tienen a lo largo del tiempo son de una sola variable, pero si los registros son de más de una variable entonces se dice que es multivariante. [5, 6] Un ejemplo de serie de tiempo univariante es el medir la temperatura de alguna máquina de forma periódica cada minuto, por otro lado si se midiera tanto la temperatura como la vibración y las revoluciones, se estaría hablando de una serie de tiempo multivariante.

**2.1.1.2 Visualización**

Para visualizar los datos de una serie de tiempo, generalmente éstos se grafican trazando las observaciones con respecto al tiempo. Esto permite tener una primer aproximación y entendimiento sobre el cómo se comporta la variable a lo largo del tiempo [5].

La figura 2.1 muestra la visualización de una serie de tiempo, donde se aprecian los precios de Bitcoin (en dólares) desde Enero del 2015 hasta Septiembre del 2018. El eje x representa el tiempo, y el eje y el precio de la criptomoneda.



Figura 2.1 Visualización de precios históricos de Bitcoin [11].

**2.1.1.3 Análisis y Pronóstico de series de tiempo**

El pronóstico de series de tiempo consiste en recolectar datos históricos de alguna variable de interés para posteriormente ser analizados y desarrollar un modelo que comprenda las relaciones que existen en los datos y que posiblemente no sean notorias. El modelo después es utilizado para hacer pronósticos del comportamiento que se espera que tenga la variable recolectada [7, 8].

**2.2 Trabajos Relevantes**

**2.2.1 ARIMA**

**2.2.1.1 Descripción del modelo**

De todos los métodos que existen actualmente para hacer pronósticos de series de tiempo, el modelo autorregresivo integrado de promedio móvil (ARIMA) es uno de los más utilizados. Este modelo tiene una característica relativamente sencilla a grandes rasgos, y que a su vez representa una deficiencia: ARIMA asume que los valores futuros y los valores históricos se relacionan linealmente, cuando la realidad es que la mayoría de los datos de series temporales presentan una relación no lineal [9].

**2.2.1.2 Pronósticos del precio de Bitcoin**

El modelo ARIMA es una elección muy popular para realizar pronósticos de criptomoneda, especialmente a corto plazo. Como se describe en [10], se implementó este modelo utilizando los datos históricos de Bitcoin desde Mayo del 2016, hasta finales de Marzo del 2018. Los datos obtenidos contenían información sobre el precio de apertura y cierre, el alto y bajo de cada día, entre otros [10]. Para preprocesar los datos, se removieron todas las columnas a excepción de la fecha y el precio de cierre, se utilizó este dato porque es el que genera más interés para los inversionistas [11].

Para implementar el modelo ARIMA, se necesitan definir 3 valores (p, d, q) utilizados para la construcción del mismo, y ya que seleccionar los valores que encajan mejor en el modelo no es una tarea sencilla, el desarrollo se realizó en el lenguaje de programación R, que cuenta con una función que encuentra los valores óptimos de forma automática [10].

Las pruebas se realizaron pronosticando los precios de cierre desde el 13 de Diciembre del 2017, hasta el 27 de Marzo del 2018, y posteriormente evaluando los precios arrojados por el modelo contra los precios reales de las mismas fechas utilizando la raíz del error cuadrático medio (RECM). La evaluación resultó en un error de 593.80 [10].

**2.2.2 LSTM**

**2.2.2.1 Descripción del modelo**

Las Long Short-Term Memory Networks (LSTM) son un tipo de redes neuronales compuestas por una capa de entrada, una o más capas escondidas, y una capa de salida. Lo que diferencia a las LSTM de las redes neuronales convencionales, es la estructura de sus capas escondidas que les permite recordar información por largos periodos de tiempo [12, 13]. Dentro de la capa escondida de una LSTM se encuentran celdas de memoria, que a su vez cuentan con 3 tipos de compuertas: La compuerta de entrada, la de salida y la de olvido. A grandes rasgos estas compuertas mantienen y ajustan el estado de la celda, es decir, la información que ésta contiene [12].

Este modelo a diferencia del anterior, es capaz de ajustarse a relaciones no lineales en las secuencia de datos, lo que permite generar pronósticos más confiables cuando son a plazos más largos [13].

**2.2.2.2 Pronósticos del precio de Bitcoin**

Wu, C.-H., et al. [12] implementaron una LSTM utilizando los precios de Bitcoin desde el inicio del 2018, hasta el 28 de Julio del mismo año. El preprocesamiento de los datos fue muy similar al mencionado previamente, quedándose únicamente con la fecha y el precio de cierre para el entrenamiento de los modelos y los pronósticos a generar.

Para realizar las pruebas se utilizaron los últimos 71 registros de las series de tiempo, teniendo un total de 138 muestras para entrenar al modelo. La raíz del error cuadrático medio fue una de las metodologías de evaluación usadas, resultando en un error total de 247.33 [12].

**2.2.3 ARIMA-LSTM**

**2.2.3.1 Descripción del modelo**

El modelo híbrido ARIMA-LSTM combina ambos modelos aprovechando las ventajas que cada uno ofrece, buscando lograr pronósticos más precisos. Primero se utiliza el modelo ARIMA para ajustar las relaciones lineales en los datos y posteriormente el modelo LSTM se encarga de las relaciones no lineales [9].

La secuencia de pasos para hacer pronósticos de series de tiempo con el modelo ARIMA-LSTM se representa en la figura 2.2. A grandes rasgos, los pasos son los siguientes:

1: Entrenar el modelo ARIMA con los datos históricos de las series de tiempo.

2: Obtener la secuencia residual resultante (errores en los pronósticos de entrenamiento).

3: Ajustar el modelo LSTM utilizando los residuos de ARIMA.

4: Hacer pronóstico con el modelo ARIMA del paso 1.

5: Hacer pronóstico con el modelo LSTM del paso 3.

6: Sumar ambos resultados para obtener el pronóstico final [14].

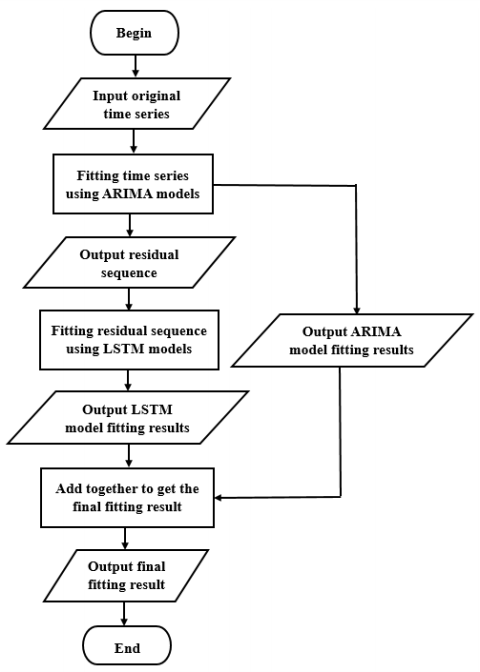


Figura 2.2 Pasos ARIMA-LSTM [14]

**2.2.3.2 Pronósticos hechos con ARIMA-LSTM**

Wang, Z., & Lou, Y. propusieron un modelo de series temporales ARIMA-LSTM para pronosticar el nivel de agua promedio de una estación hidrológica ubicada en el río Chuhe. Para ponerlo a prueba, desarrollaron los modelos de forma independiente y de forma híbrida para ver si en conjunto conseguían mejores resultados. Sus experimentos mostraron que las ventajas de ambos modelos se complementan, y logran pronósticos más precisos [9].

Liu, J., Tan, X., & Wang, Y. desarrollaron el modelo ARIMA-LSTM para pronosticar el envejecimiento del software para servicios en la nube. Su propuesta (muy similar a la anterior) buscaba comprobar si se podían obtener mejores resultados usando el modelo híbrido. Para comprobarlo, obtuvieron registros diarios del Centro de Datos de Google Cloud, teniendo acceso al uso del CPU y la memoria de más de 1600 máquinas. Después de evaluar los tres modelos, la raíz del error cuadrático medio de ARIMA-LSTM fue la menor de todas, concluyendo que se puede reducir el error y hacer mejores pronósticos con el modelo híbrido [14].

**2.3 Conclusión**

Actualmente existen varios métodos utilizados para pronosticar el precio de criptomonedas. Los trabajos previamente mencionados muestran el uso de distintos modelos de series temporales, así como las evaluaciones de los mismos. El punto más interesante a notar es el resultado que se obtiene al comparar el modelo híbrido ARIMA-LSTM contra ARIMA y LSTM de forma individual, ya que se lograron pronósticos más acertados en ambos casos.

A pesar de los resultados favorables, el modelo ARIMA-LSTM no se ha utilizado comúnmente para hacer pronósticos de series de tiempo, y en particular, no se ha utilizado para pronosticar el precio de criptomonedas, por lo que no se puede afirmar aún si es una mejor opción o no que los modelos ya propuestos.

**Capítulo 3**

En este capítulo se habla acerca de la gran volatilidad de las criptomonedas y del cómo ha afectado a inversionistas. Posteriormente se habla acerca de un área de oportunidad que se presenta como una alternativa para posiblemente acercarse a pronósticos más precisos del precio de las criptomonedas.

**Objetivo de Investigación**

**3.1 Volatilidad de las criptomonedas**

La alta volatilidad de las criptomonedas es un hecho que hace que la mayoría de las personas tengan cuidado al invertir en ellas, pero esto no siempre ha sido así. Uno de los principales momentos en los que Bitcoin fue el centro de atención ocurrió en el 2017, año en el que se registró un aumento desmesurado en los precios de dicha criptomoneda. Esto atrajo a un gran número de inversionistas con poca experiencia aprovechando la “oportunidad de su vida”, sin saber que en poco tiempo tendrían pérdidas masivas de hasta un 96% con respecto a su inversión inicial en algunos casos [15].

Las fluctuaciones drásticas en el precio de Bitcoin siguen siendo algo común al día de hoy; el desplome del valor de Bitcoin más reciente ocurrió el 12 de Marzo, reportando pérdidas de prácticamente el 50% para los inversionistas en tan solo un día [4]. Esto no ha impedido que las criptomonedas sigan siendo una opción popular de inversión, actualmente se llevan a cabo más de 250, 000 transacciones de criptomonedas al día [3].

**3.2 Trabajos anteriores**

Previamente se habló de ARIMA y de LSTM, dos modelos utilizados comúnmente para hacer pronósticos de criptomonedas. En cuanto a la implementación de dichos modelos no existe algún problema en particular, desde la forma en que se desarrollaron hasta la metodología de evaluación, sin embargo, es importante hacer énfasis en el modelo híbrido ARIMA-LSTM, ya que los resultados obtenidos por este modelo fueron positivos al probarse contra ARIMA y contra LSTM como se menciona en el capítulo anterior, y esta comparación de modelos aún no se ha llevado a cabo para pronósticos del precio de criptomonedas.

A partir de esto se detecta un área de oportunidad, ya que existe la posibilidad de que el modelo híbrido obtenga mejores resultados que ARIMA y que LSTM, misma que no se ha puesto a prueba aún.

**3.3 Objetivo de Investigación**

Es importante seguir buscando alternativas que permitan acercarse a pronósticos más precisos y así reducir el riesgo al invertir en criptomonedas.

El área de oportunidad detectada, permite definir un objetivo de investigación claro: Determinar si los pronósticos del precio de criptomonedas hechos por el modelo híbrido ARIMA-LSTM son más precisos que los pronósticos de los modelos ARIMA y LSTM.

**Capítulo 4**

**Propuesta de Solución**

Como solución para cumplir con el objetivo de investigación planteado, primero se obtuvieron datos históricos de las criptomonedas Bitcoin y Litecoin, posteriormente se desarrollaron los tres modelos a comparar (ARIMA, LSTM, ARIMA-LSTM), entrenándolos con una parte de los datos obtenidos, y finalmente se hicieron pronósticos que después podrán ser evaluados contra los datos no utilizados aún. Todo el proceso se llevó a cabo usando el lenguaje de programación Python.

**4.1 Datos**

**4.1.1 Obtención de datos históricos**

Los datos históricos a utilizar tanto para entrenar como probar los modelos fueron obtenidos de CoinMarketCap, que cuenta con datos relevantes de las criptomonedas día a día. Antes el sitio web ofrecía la opción de descargar estos datos pero actualmente ya no está disponible, por lo que se tuvo que acceder a la página directamente desde Python para obtener los datos necesarios. [16, 17]

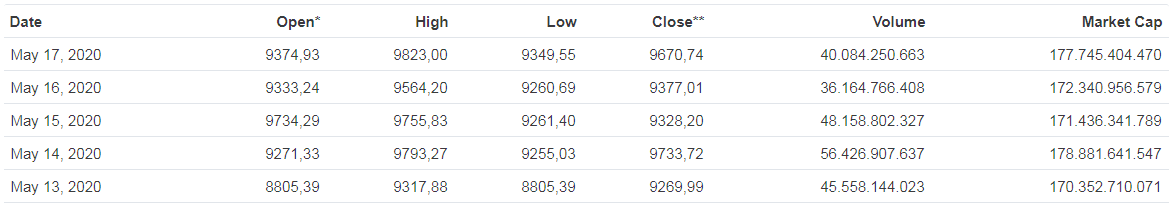


Figura 4.1 Muestra de datos históricos de Bitcoin en dólares [16].

**4.1.2 Preprocesamiento general de datos**

El preprocesamiento de datos se realizó como en el estado del arte, desechando todas las columnas a excepción de la fecha y el precio de cierre. El resultado del preprocesamiento es una serie de tiempo univariante que se define como:

{X1, X2,… X(t)}, t = 1, 2, 3,…

t: Fecha dada por Día, Mes y Año

X(t): Precio de cierre de la criptomoneda en una fecha en particular

**4.2 Implementación de modelos**

Es prácticamente imposible encontrar modelos que se hayan entrenado y evaluado utilizando las mismas fechas, y aunque se diera la casualidad de encontrarlos, éstos no estarían evaluados con los datos históricos más recientes. Por estas razones la propuesta de solución implica implementar los tres modelos utilizando los mismos datos para que los resultados obtenidos sean más acertados.

Las fechas utilizadas para entrenar los modelos fueron desde 01/01/2017 hasta 31/12/2019, dejando todos los precios de cierre del 2020 para hacer las pruebas y evaluar cada modelo. Estas fechas aplicaron para hacer los pronósticos de ambas criptomonedas, Bitcoin y Litecoin.

**4.2.1 ARIMA**

La implementación de este modelo es relativamente rápida, ya que las series de tiempo se pueden usar de forma directa para entrenar al modelo y que éste se ajuste a los datos.

Así como se mencionó en el estado del arte, no es una tarea sencilla elegir los tres valores que encajan mejor para construir el modelo ARIMA, y como Python actualmente ya cuenta con la opción de encontrar los parámetros más óptimos de ARIMA haciendo uso de su propia función auto\_arima, el desarrollo de este modelo se facilitó drásticamente.

El modelo seleccionado fue ARIMA(1, 1, 3) para Bitcoin y ARIMA(1, 1, 2) para Litecoin. Estos modelos fueron utilizados para hacer los pronósticos y posteriormente compararse contra los modelos LSTM y ARIMA-LSTM.

**4.2.1.1 Pronósticos**

Para hacer los pronósticos con ARIMA, se tomó ventaja de la velocidad con la que se entrena el modelo. Para cada una de las fechas de prueba, se siguieron estos pasos:

1: Pronosticar el precio de cierre para la fecha correspondiente.

2: Agregar el precio real a los precios usados para el entrenamiento.

3: Reentrenar el modelo [19].

**4.2.2 LSTM**

El primer paso para implementar este modelo fue normalizar los datos después de hacer el preprocesamiento general, para lograr esto se utilizó la librería scikit-learn, específicamente haciendo uso de su MinMaxScaler que ofrece para normalizar los datos y que éstos se encuentren en una escala de 0 a 1 [13].

A diferencia del modelo ARIMA, las LSTM no se pueden entrenar de forma directa con tan solo tener la serie de tiempo.

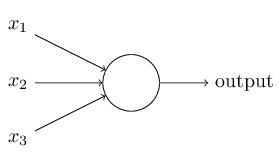


Figura 4.2: Estructura de un Perceptron [18].

La figura 4.2 muestra la estructura más básica de una red neuronal, pero es suficiente para entender el problema que presentan los datos actualmente. Las redes neuronales reciben un cierto de número de datos de entrada para al final tener uno o más datos de salida (en este caso uno, ya que los pronósticos serán únicamente del precio de cierre) [18], pero actualmente la serie de tiempo sólo tiene un valor, el precio de cierre, registrado para cada día, entonces… ¿Cómo se determina qué valores se usan como entrada y cuál es la salida esperada para entrenar al modelo?

**4.2.2.1 Ventana deslizante**

Dada una secuencia de números representados por series de tiempo, es posible reestructurarlos para hacer que tengan el formato que espera la red neuronal (cierto número de datos de entrada y un dato de salida). Esto se logra utilizando los valores de periodos de tiempo anteriores como datos de entrada, y el periodo siguiente como dato de salida. El número de periodos anteriores a tomar por cada fecha depende de un ancho de ventana que se define previo a la reestructuración de los datos [19]. Las siguientes tablas muestran un ejemplo de este proceso aplicado a los precios de Bitcoin.

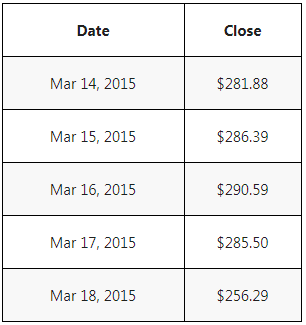


Figura 4.3 Muestra de serie de tiempo del precio de Bitcoin



Tabla 4.2 Ventana deslizante aplicada con ancho de ventana = 2

Como se puede apreciar, las primeras fechas que se encuentran dentro del ancho de la ventana, tienen al menos un valor anterior desconocido, por lo que estas fechas fueron descartadas por completo. Es importante mencionar que la implementación de la ventana deslizante se efectuó una vez que los precios ya estaban normalizados, las figuras anteriores son únicamente un ejemplo para mostrar cómo se reestructura una serie de tiempo.

Para esta implementación se utilizó un ancho de ventana de 4.

**4.2.2.2 Pronósticos**

Para hacer pronósticos, lo primero que se hizo fue construir el modelo haciendo uso de la librería Keras, que permite crear redes neuronales de forma fácil y rápida [13]. En cuanto a la arquitectura de la red neuronal, se agregó una capa escondida LSTM con 50 nodos, que recibe como entrada los precios correspondientes a los valores anteriores. Después se agregó una capa Dropout con un valor de 0.2, y finalmente una capa Dense con un nodo que se utilizó para ser el valor de salida, es decir, el pronóstico del precio de cierre. El optimizador a utilizar fue adam [13].

Para cada una de las fechas de prueba, se utilizaron los 4 precios de cierre anteriores para hacer el pronóstico del siguiente día. El valor de salida de la LSTM se encontraba dentro de la escala de 0 a 1, por lo que se tuvo que desnormalizar para obtener el pronóstico final del día.

**4.2.3 ARIMA-LSTM**

La implementación de este modelo se basó en el algoritmo presentado en el capítulo 2.

Después del preprocesamiento general se implementó el modelo ARIMA con las mismas configuraciones que ARIMA implementado individualmente, la diferencia ahora fue que después de entrenar el modelo se obtuvieron los residuales para entrenar la LSTM.

Un residual es la diferencia entre una observación y un pronóstico para una fecha, es decir, el error. Si se juntan todos los residuales se obtiene una secuencia de datos que contiene las relaciones que posiblemente el modelo ARIMA no pudo capturar por ser un modelo lineal.

Una vez se obtuvo la secuencia residual, se realizó exactamente el mismo procedimiento para entrenar la LSTM, la diferencia es que ahora la serie de tiempo a utilizar, a normalizar, y a reestructurar usando la ventana deslizante, se construyó a partir de los residuos de ARIMA, esperando que fuera posible ajustarse a ellos con ayuda de un modelo no lineal [9].

**4.2.2.1 Pronósticos**

Los pronósticos de ARIMA-LSTM se efectuaron haciendo pronósticos con ambos modelos de la misma forma en la que se hizo con cada uno y combinándolos. Los pasos para hacer los pronósticos para cada fecha fueron los siguientes:

1: Pronosticar el precio de cierre usando ARIMA.

2: Pronosticar el residuo usando LSTM.

3: Desnormalizar el residuo pronosticado.

4: Sumar los pronósticos de ambos modelos.

5: Calcular el residuo del día usando el precio real y el pronóstico de ARIMA.

6: Normalizar el residual.

7: Agregar el nuevo residual a los datos de entrada para la LSTM

8: Agregar el precio real a los precios usados para el entrenamiento de ARIMA.

9: Reentrenar el modelo ARIMA [14].

**Capítulo 7**

**Conclusiones**

TRABAJOS FUTUROS:

Yo use el closing Price porque es como el estado del arte

Maybe un EDA para checar si es chido usar algo más que el precio de cierre

**Bibliografía**

[1] M. Rahouti, K. Xiong and N. Ghani, "Bitcoin Concepts, Threats, and Machine-Learning Security Solutions," in IEEE Access, vol. 6, pp. 67189-67205, 2018.

[2] Asante, S. Are Bitcoins price predictable? Evidence from machine learning techniques using technical indicators. 2019. pp. 2.

[3] Beck et al. Sensing Social Media Signals for Cryptocurrency News . 2019. pp. 1-2.

[4] Bovaird, C. (2020). *Bitcoin Lost Roughly 50% Of Its Value In A Day*. Mayo 4, 2020, de Forbes Sitio web: <https://www.forbes.com/sites/cbovaird/2020/03/12/bitcoin-lost-roughly-50-of-its-value-in-a-day>

[5] Adhikari, R; Agrawal R. An Introductory Study on Time Series Modeling and Forecasting. pp. 12-13.

[6] John H. Cochrane, “Time Series for Macroeconomics and Finance”, Graduate School of Business, University of Chicago, spring 1997.

[7] G.P. Zhang, “A neural network ensemble method with jittered training data for time series forecasting”, Information Sciences 177 (2007), pp: 5329–5346.

[8] G.P. Zhang, “Time series forecasting using a hybrid ARIMA and neural network model”, Neurocomputing 50 (2003), pp: 159–175.

[9] Wang, Z., & Lou, Y. (2019). Hydrological time series forecast model based on wavelet de-noising and ARIMA-LSTM. 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC).

[10] Yenidogan, I., Cayir, A., Kozan, O., Dag, T., & Arslan, C. (2018). Bitcoin Forecasting Using ARIMA and PROPHET. 2018 3rd International Conference on Computer Science and Engineering (UBMK).

[11] Anupriya, & Garg, S. (2018). Autoregressive Integrated Moving Average Model based Prediction of Bitcoin Close Price. 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT).

[12] Wu, C.-H., Lu, C.-C., Ma, Y.-F., & Lu, R.-S. (2018). A New Forecasting Framework for Bitcoin Price with LSTM. 2018 IEEE International Conference on Data Mining Workshops (ICDMW).

[13] S. Tandon, S. Tripathi, P. Saraswat and C. Dabas, "Bitcoin Price Forecasting using LSTM and 10-Fold Cross validation," 2019 International Conference on Signal Processing and Communication (ICSC), NOIDA, India, 2019, pp. 323-328.

[14] Liu, J., Tan, X., & Wang, Y. (2019). CSSAP: Software Aging Prediction for Cloud Services Based on ARIMA-LSTM Hybrid Model. 2019 IEEE International Conference on Web Services (ICWS).

[15] Kaplan, M. (2018). Bitcoin crash: This man lost his savings when cryptocurrencies plunged. Mayo 16, 2020, de CNN Sitio web: <https://money.cnn.com/2018/09/11/investing/bitcoin-crash-victim/index.html>

[16] <https://coinmarketcap.com/currencies/bitcoin/historical-data/>

[17] https://coinmarketcap.com/currencies/litecoin/historical-data/

[18] Nielsen, M. A. (2018). Neural Networks and Deep Learning [misc]. Determination Press

[19] Brownlee, J. (2017). Introduction to Time Series Forecasting with Python: How to Prepare Data and Develop Models to Predict the Future. Machine Learning Mastery.