



Universidad de
San Andrés

Universidad de San Andrés
Escuela de Negocios
Magister en Finanzas

Evaluación del poder predictivo del modelo LSTM en
el análisis de Series de tiempo financieras

Autor: Lic. Miguel Palou Miceli

DNI: 30.117.750

Director de Trabajo Final de Graduación: Dr. Leonardo Galletti

Ciudad de Buenos Aires - 18 de abril de 2.024

Resumen

El mundo de las finanzas exige soluciones rápidas, seguras y vanguardistas. En un contexto de constante cambio, con la irrupción de la Inteligencia Artificial (AI), los grandes volúmenes de datos almacenables y las mejoras en los procesos computacionales, un mundo de métodos sofisticados emerge como una alternativa fiable a los tradicionales modelos predictivos aplicados a series de precios. A la par, corre el creciente interés por un Mercado Bursátil, que hoy tiene ventajas ciertas desde la accesibilidad, pero que por otro lado se volvió extremadamente complejo y volátil por la diversidad de agentes involucrados. Tratar de inferir el comportamiento vertiginoso de ciertos instrumentos en un contexto holístico, y hacerlo de manera rápida y confiable es el objetivo de los investigadores del área¹.

En el presente estudio se utilizó una red neuronal recurrente, *Long Short Term Memory*² (LSTM, por sus siglas en inglés) para predecir el precio ajustado de cierre al día siguiente de diferentes activos seleccionados. Dado que este modelo goza de muy buena reputación como predictor, el objetivo fue el de evaluar las características y propiedades del mismo, intentar recrear los resultados obtenidos en la bibliografía, y encontrar la combinación óptima de *hiperparámetros* que minimice el error en una ventana de tiempo pasada.

La investigación consistió en una serie de simulaciones de carácter predictivo del Precio de Cierre del día siguiente de distintas acciones, Lantheus Holding Inc., Mercado libre, Well Fargo´s y Marvell Technology Inc. En el proceso se utilizaron distintas arquitecturas, como así también tipos de modelo, periodos, secuencias, hiperparámetros, y características de entrada, a fin de alcanzar el mejor ajuste posible, considerando distintos optimizadores y adoptando como criterio de ajuste la *Minimización del Error Cuadrado* (MSE)³. Los resultados experimentales que se exponen en la presente, advierten sobre el potencial implícito de desarrollar este formato de modelo en el campo de las finanzas, lo que no necesariamente indique carencia de debilidades y limitaciones concretas. Se arribara a la conclusión de que LSTM es un modelo flexible capaz de adaptarse a cualquier entorno o estructura, y que a la luz de los resultados, se presenta como una herramienta muy poderosa de cara al futuro en el análisis de series de tiempo.

¹Simon S, Raoot AD, Lake V. Accuracy driven artificial neural networks in stock market prediction. *International Journal of Soft Computing*. 2012;3:35–44.

²Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>

³El Error Cuadrático Medio (MSE) es una métrica estadística utilizada en el ámbito del aprendizaje automático y la estadística para evaluar la precisión de un modelo predictivo.

Índice

1.INTRODUCCION	4
2.LITERATURA	7
3.REDES NEURONALES ARTIFICIALES	11
4.LSTM	14
5.MODELADO	18
6.ANALISIS DE RESULTADOS	28
7.CONCLUSIONES	45
8.ANEXO	46
9.CODIGO DE PYTHON	56
10.BIBLIOGRAFIA	60



Universidad de
San Andrés

1.-INTRODUCCION

Si de algo podemos estar seguros respecto a lo que puede ocurrirle al precio de una acción cualquiera el día de mañana, es que éste subirá, o bajará. La incógnita, entonces, está asociada a la dirección y a la magnitud del movimiento, y despejarla implica analizar el proceso formador de precios y sus determinantes, así como describir el comportamiento histórico de la serie. A modo de ejemplo: si tomamos los retornos diarios del *SPDR S&P 500 ETF (SPY)*⁴ a lo largo del tiempo, veremos que más del 99,96% de éstos, se encuentran entre valores -1% y 1%, por lo que con un grado de confianza elevado, se podría inferir que el precio del activo mencionado mañana, es el precio de hoy +/- el 1%. La simplicidad del planteo, sugiere que cualquier individuo con mínimo poder deductivo se sienta capacitado de poder abordar con ligereza analítica la tarea de encontrar o identificar patrones o relaciones que expliquen los precios de manera perfecta, o al menos la dirección que estos puedan tomar. Si bien es cierto que la volatilidad varía de activo en activo, volviéndose dramática en algunos o en ciertos períodos, un rango de 2 puntos porcentuales es una invitación de al menos intentarlo. Sin embargo cuando se tratan de volúmenes importantes y períodos largos, cualquier rango, por más insignificante que parezca, puede representar pérdidas catastróficas. Este factor es clave para el análisis, blinda y protege estas técnicas predictivas del uso meramente fortuito.

Los párrafos precedentes son introductorios a la formulación de un simple interrogante, el cual muchos han abordado y han intentado responder: ‘

¿Hasta qué punto se puede utilizar la historia pasada del precio de una acción, para hacer predicciones significativas sobre el precio futuro de la misma?’, documento hecho, así comienza el “ que consagró académicamente a Eugene Fama *mercados eficientes*.

Y cuyo principal aporte fue la introducción del concepto de *Sendero Aleatorio*. Fama, basándose en modelos de 6, descalifica todo intento de análisis chartista, considerándolo solo un hobby, al tiempo que aseguraba que la variación en los precios no tienen memoria, por lo que no hay mucho que el pasado pueda decirnos respecto al futuro. Fama asegura que la evidencia proporcionada por los análisis de *Sendero Aleatorio* es demasiado robusta como para ser refutada. Sin embargo reconoce que esa independencia no es perfecta, puesto que es muy difícil llegar a rechazar la hipótesis nula de dependencia estadística⁷. A su vez, sugiere que los cambios sucesivos en los precios de las acciones son independientes entre sí y siguen una distribución de probabilidad aleatoria, concluyendo respecto a la eficiencia de mercado

⁴El SPY es un ETF que desde 1993 replica al **S&P 500**, un índice compuesto por las 500 empresas más importantes de la mayor economía del mundo, la de EEUU.

⁵

Fama, Eugene: “The Behavior of Stock-Market Prices” *The Journal of Business*, ene, 1965, Vol, 38, No, 1 (ene, 1965), pp, 34-105.

⁶

Esta teoría concluye que: (1) los cambios sucesivos de precios son independientes y (2) los cambios de precios se ajustan a alguna distribución de probabilidad,

⁷

Para todas las acciones analizadas en el siguiente trabajo se testeó estacionariedad por ADF, no pudiendo en ningún caso rechazar la hipótesis nula, indicando que las series pueden contener una estructura sistemática persistente en el tiempo.

y cómo los precios reflejarán toda la información disponible y se ajustarán rápidamente a la nueva información que surgiera.

La aceptación de esta forma *semifuerte* de eficiencia de mercado vía determinación no estacionaria de la serie de precios, nos llevaría a concluir sobre la imposibilidad de obtener ganancias excepcionales. Sin embargo, existen numerosas anomalías en los mercados de capitales que contrastan notablemente con la noción de eficiencia. *Paul Samuelson*⁸ probó que los mercados competitivos lejos están de la perfección, es más, aseguraba que no estaba seguro de que funcionaran '*tan bien*'. Adhiero a *Robert Shiller*⁹, quien desde el *Área de la Finanza Conductual*, asegura que existen factores psicológicos y emocionales que son determinantes en las decisiones financieras, y que llevan a comportamientos irracionales en los mercados. Shiller concluye que suponer que los mercados financieros siempre funcionan bien, es falaz y contradictorio, como también el hecho de que los cambios de precios siempre reflejan información genuina. De hecho reflexiona respecto a las burbujas especulativas, ligándolos directamente a una ineficiencia de mercado con causa conductual, que de hecho se corroboró con el auge mundial de los mercados de valores y su posterior colapso después del año 2000. Shiller no fue el único en desafiar a Fama, dentro del área de *Gestión de Carteras de Inversión y Finanzas Corporativas*, se ha demostrado que cualquier gestor puede generar consistentemente un *alpha positivo*¹⁰, lo que es indicador de que existen oportunidades de inversión subvaloradas o sobrevaloradas en el mercado. La búsqueda de '*alpha*' es fundamental en la gestión activa de inversiones, donde los administradores de cartera intentan seleccionar activos o estrategias que generen retornos superiores al mercado. Si bien es difícil, es un gran indicador de *ineficiencias de mercado*. Otros indicadores de deficiencia están planteados en la *Teoría de la Agencia* o la *Teoría del Ruido*, sin ir tan lejos *Jacobs* (2015) o *Green, Hand y Zhang* (2013) proporcionan encuestas que listan más de cien de estas anomalías del mercado de capitales, que efectivamente se basan en señales predictivas de retorno para superar al mercado.

La presencia de imperfecciones, o la falta de consenso, no implica que e ncontrar patrones o dependencias no lineales en series financieras a fin de inferir a futuro sea sencillo. De hecho se han aplicado muchos modelos econométricos y estadísticos, durante larga data sin obtener resultados relevantes en series de precios bursátiles, tales como los *Modelos de regresión lineal*,

⁸ Samuelson, Paul. (1965), "Proof that properly anticipated prices fluctuate randomly", *Industrial Management Review*, p. 41-49; MIT.

⁹

Shiller, R, J, (2003), "From Efficient Markets Theory to Behavioral Finance," *The Journal of Economic Perspectives*, Vol, 17, No, 1 (Winter, 2003), pp, 83-104, American Economic Association.

¹⁰

El concepto de "alpha" en el contexto de las finanzas y la inversión fue introducido y popularizado por Michael Jensen, un economista y profesor de finanzas de la Escuela de Negocios de Harvard. Jensen es conocido por su trabajo pionero en el campo de la teoría de carteras y la eficiencia del mercado.

Modelos autoregresivos de *media móvil integrada* (ARIMA)¹¹, Modelos de *co-integración*¹², Modelos de *heterocedasticidad condicional autorregresiva generalizada* (GARCH)¹³, Modelos de *vector autorregresivo* (VAR)¹⁴ y Modelos de *corrección de errores* (ECM)¹⁵. Wu and Gao (2018), sostienen: '*los modelos de pronóstico tradicionales han fallado en capturar la no linealidad y complejidad de las series temporales financieras, lo que ha llevado a una baja precisión en las predicciones*'. La inteligencia artificial disrumpe con modelos más complejos y no lineales para el pronóstico de series temporales financieras, como las *Redes Neuronales Artificiales* (ANN), la *Regresión de Vectores de Soporte* (SVR)¹⁶ y *Métodos de deep learning*¹⁷, sin embargo el verdadero avance llega con modelos que lograron eliminar limitaciones de memoria de largo plazo como los CNN y los RNN¹⁸. La presente investigación buscó ser exploratoria y experimental, y consistió en evaluar la capacidad predictiva en series de tiempo de activos bursátiles específicos, utilizando un modelo RNN particular, como lo es el 'Modelo LSTM'. La idea se basó en proveer al modelo de una arquitectura flexible, capaz de adaptarse a una serie de variables selectivas suficientes, para optimizar la predicción del precio ajustado al cierre del siguiente día hábil. En esta entrega, se puso foco principalmente en explorar su potencial predictivo, entendiendo que es factible contribuir a la literatura existente, por medio del análisis comparativo de los resultados obtenidos al combinarse diferentes arquitecturas, enfoques, secuencias e hiperparámetros, de manera tal de encontrar el modelo que minimice la función de pérdida implícita y prediga eficientemente en los periodos seleccionados.

¹¹ Es un modelo estadístico utilizado para analizar y predecir series temporales. Combina componentes autoregresivos (AR), de media móvil (MA) y diferenciación (I) para modelar patrones y tendencias en datos temporales.

¹²

Un modelo de cointegración es un enfoque estadístico que identifica la relación a largo plazo entre dos o más series temporales que pueden mostrar comportamientos de tendencia estocástica individualmente.

¹³

Es un modelo utilizado en el análisis de series temporales financieras para modelar la volatilidad condicional. Captura la heterocedasticidad, es decir, la variabilidad no constante en el tiempo, y se utiliza comúnmente para modelar la volatilidad en mercados financieros.

¹⁴

Es un modelo estadístico que representa múltiples series temporales como un sistema de ecuaciones simultáneas. Cada variable se modela como una función lineal de sus valores pasados y de los valores pasados de otras variables en el sistema.

¹⁵

Es un modelo que se utiliza en el análisis de series temporales cuando se observa una relación de largo plazo entre las variables. Corrige los errores de desequilibrio a corto plazo entre las variables para restaurar el equilibrio a largo plazo.

¹⁶

Es una técnica de regresión basada en máquinas de soporte vectorial. Utiliza vectores de soporte para aproximar una función que mapea las entradas a las salidas, y es especialmente útil en la predicción de valores continuos en problemas de regresión.

¹⁷

El termino aparece por primera vez en 2006, y se le atribuye a Geoffrey Hinton. En la Sección 3 de esta investigación se desarrolla el concepto de manera amplia.

¹⁸

CNN y RNN hacen referencia a las Redes Neuronales Convulcionales y las Recurrentes.

2.-LITERATURA

En los últimos años, la literatura sobre redes neuronales aplicadas a las finanzas se ha multiplicado. Fadlalla y Lin (2001) destacan este fenómeno con información documentada de los últimos 10 años previos a su publicación. El trabajo es relevante desde lo informativo puesto que revela las diversas aplicaciones que tienen estas redes en finanzas¹⁹, las más importantes, anticipar precios de activos, bancarrotas de bancos y rating crediticio; y por otro lado empírico, puesto que cada documento llega a conclusiones y resultados relevantes, los cuales son recopilados y analizados por los autores. Estos concluyen sobre el importante avance de las redes en el campo de investigación, basándose en la mejora en términos predictivos vigentes en la transición de modelos tradicionales de econometría a redes neuronales.

Sin seguir un orden de prelación histórica, se cita preferentemente a Qihang Ma (2020), quien efectúa simulaciones con ARIMA y una red convencional ANN. El autor asegura que por más que se hayan utilizado ampliamente en la predicción de precios de acciones, estos métodos son incapaces de medir la continuidad de las tendencias de precios en evolución, y en contraposición pondera el rendimiento de los modelos LSTM. A su vez destaca la importancia del proceso de ajuste vía retropropagación²⁰ como factor determinante en la mejora introducida por las redes neuronales al campo de investigación. Ma concluye que ‘el modelo LSTM rara vez ha sido utilizado en estudios académicos serios, por ende no se comprueba un preprocesamiento exhaustivo de los datos, por lo que el rendimiento de LSTM no se puede demostrar adecuadamente. Ésta es también una razón importante por la cual no se ha utilizado ampliamente...’. La investigación de Ma se basó en el análisis de la serie de precios de la compañía Dell’s, concluyendo sobre las ventajas claras del modelo LSTM, destacando no solo la posibilidad de encontrar dependencias no lineales en el precio, sino también a que es lo suficientemente flexible a la hora de incorporar los datos de entrada al modelo. Como Ma, Huck (2009, 2010), proporcionó evidencia respecto al avance de los modelos RNN en el análisis de series temporales de datos, identificando patrones y tendencias claras dentro de la secuencia temporal. Huck, como tantos autores, trabajó con una serie de precios del índice NIKKEI, la cual es “no estacionaria”, sin efectuarle ningún tipo de transformación, lo que originó una de las críticas generales más importantes al modelo LSTM.

Resulta muy interesante intentar plasmar diferencias y similitudes entre sistemas tradicionales (como modelos econométricos clásicos) y modelos de Redes Neuronales, y en este sentido el documento de Sendhil Mullainathan and Jann Spiess (2018) es revelador. Los autores

¹⁹ De una muestra de 40 artículos, 22 se correspondían a trabajos sobre predicción de precios de activos financieros.

²⁰

La retropropagación del error es un algoritmo de optimización utilizado en el entrenamiento de redes neuronales. Se basa en la minimización del error entre las predicciones del modelo y los valores reales, ajustando los pesos de la red en sentido contrario al gradiente de la función de error.

enumeran las ventajas en el uso del Machine Learning y el Big Data²¹ por sobre otras técnicas estadísticas convencionales, aunque advierten de algunos riesgos asociados a su uso, sobre todo morales e ideológicos.

De todos los trabajos de la bibliografía, se identificaron tres tipos según el objeto de su análisis.

Primero, aquellos que comparan los resultados obtenidos por LSTM contra otros modelos tradicionales, neuronales o algorítmicos computacionales, evaluando el valor agregado que genera esta tecnología al análisis de las series de tiempo; segundo, aquellos que tratan de incorporar características como entrada al modelo, o bien innovar en la configuración del modelo complementándolo con otros (modelos híbridos), a fin de mejorar la calidad predictiva del mismo. El tercer subgrupo, quizás el más pequeño, está integrado por investigaciones como la de Tannuré & Gerez (2023), la cual consistió en un “barrido de hiperparámetros”, es decir, en la generación de miles de combinaciones distintas de arquitecturas LSTM, con el objetivo de identificar a la que logre el mejor resultado en términos de optimización según el criterio escogido. Si bien el trabajo de Tannuré & Gerez no aventura una combinación óptima, de alguna manera insinúa el potencial del grid search²². Es un punto importante para destacar puesto que si hay algo evidente, y dejando de lado la necesidad imperante de generar backtesting²³ y evaluar sobreajuste²⁴, es que la efectividad de este tipo de modelos, indefectiblemente depende del grid search, por lo que, para poder comparar sus resultados con otros métodos, es importante implementar este proceso.

En la literatura en general se observa evasión de conclusiones fuertes y contundentes, lo que podría ser indicador de lo incipiente del Área de Estudio. Esto se corresponde con una fase de desarrollo inicial caracterizada por un volumen creciente de investigaciones que responden definitivamente a potencialidad. De hecho Krauss (2017), concluye que, si bien las “redes neuronales son notoriamente difíciles de entrenar”, el camino definitivamente es ‘ese’, ya que “puede ser que haya configuraciones en el espacio de parámetros para mejorar aún más el rendimiento del aprendizaje profundo”. Krauss es quizás quien más aportes realizó al ámbito de las LSTM, y junto a Fischer (2018) efectuaron un trabajo insignia en donde utilizaron ‘retornos’ históricos de acciones pertenecientes a S&P 500 con el objetivo de optimizar su estrategia de “posicionamiento”. Los autores emplearon un modelo LSTM con validación cruzada²⁵ como criterio

²¹ Se refiere al manejo y análisis de conjuntos de datos extremadamente grandes y complejos que superan la capacidad de las herramientas de procesamiento de datos tradicionales.

²²

Nombre que se le otorga a la técnica para el barrido de hiperparámetros.

²³

Es una técnica utilizada en finanzas y análisis cuantitativo para evaluar el rendimiento de un modelo o estrategia utilizando datos históricos. Consiste en aplicar el modelo a datos pasados para evaluar cómo se habría comportado en condiciones reales.

²⁴

Ocurre cuando un modelo se ajusta demasiado a los datos de entrenamiento, capturando incluso el ruido o la variabilidad aleatoria. Esto puede resultar en un rendimiento deficiente al enfrentarse a nuevos datos no vistos.

²⁵

Es una técnica utilizada para evaluar el rendimiento de un modelo al dividir el conjunto de datos en varios subconjuntos, entrenar el modelo en algunos de ellos y evaluar en los restantes.

de ajuste, tomando como entrada al modelo, los retornos de todas las empresas componentes del índice. El modelo estaba configurado para indicar la dirección del precio de cada acción al día siguiente, y los autores se posicionaron como comprador o vendedor, según esos rendimientos esperados fueron superiores o inferiores a la mediana de la muestra. Una vez establecida la estrategia óptima, se analizaron los rendimientos de la misma comparándolos a los que se hubiesen obtenido con otros métodos, llámese *Bosque aleatorio*²⁶, *Redes Neuronales Profundas*²⁷ o *Regresión Logística*²⁸. Las carteras construidas por LSTM superaron ampliamente en rendimiento a aquellas construidas por los demás modelos estudiados, aún considerando costos de transacción. A pesar de los buenos resultados, *Krauss* se percibe desalentado, ya que su análisis concluye con la separación en períodos diferentes de datos, y si bien los resultados obtenidos en términos de rendimiento de carteras, es superior a otros métodos, difieren período a período, indicando que el conocimiento de estas técnicas en el mercado y su utilización masiva por los agentes, terminaría comprimiendo su poder predictivo.

Xiao, Feng, Yan y Ma (2022), elaboraron un trabajo donde se comparan rendimientos de un modelo ARIMA con otro LSTM. El trabajo se centra en el análisis de acciones de la industria tecnológica estadounidense. A pesar de que el modelo ARIMA requiere una menor capacidad computacional durante el proceso de entrenamiento, los autores concluyeron que LSTM presenta ventajas en términos de capacidad predictiva.

Choi (2018) desarrolló un modelo LSTM para evaluar la correlación en los precios de dos activos. Si bien su objetivo fue testear y estimar correlación entre variables, éste tipo de arquitecturas bien podría ser empleada para predecir precios. Su aporte es revelador, sobre todo para la optimización de carteras de inversión que suelen contemplar coeficientes de correlación fijos, puesto que contradice a *Francois Chesnay y Eric Jondeau (2001)*. El autor emplea un modelo híbrido, con datos que resultan del residuo generado por el modelo ARIMA, y logra no solo capturar dependencias lineales, sino también no lineales en los precios. En la bibliografía se encuentra basta cantidad de trabajos en donde se comparan modelos, pero pocos en los que los modelos cooperen entre ellos como es este caso.

Yungao Wu y Jianwei Gao (2018), continúan la línea en modelos híbridos e introducen un algoritmo llamado *Impulso Adaptativo (ADABOOST)*²⁹ al final de la secuencia, a fin de dotar de mayor poder predictivo al modelo LSTM. Los resultados para muestras segmentadas del *SPY*, *SHCI* y *USD/EUR* son bastante alentadoras. Los autores comparan resultados con otros

²⁶ Un algoritmo de aprendizaje automático que construye múltiples árboles de decisión durante el entrenamiento y fusiona sus predicciones para mejorar la precisión y evitar el sobreajuste.

²⁷ Modelos de aprendizaje profundo compuestos por múltiples capas de neuronas artificiales, que permiten la representación de patrones complejos y abstracciones en datos.

²⁸ Modela la probabilidad de que una observación pertenezca a una categoría particular y utiliza la función logística para asignar probabilidades y realizar predicciones de clasificación.

²⁹ "Impulso Adaptativo", es un algoritmo de aprendizaje automático que se utiliza para mejorar la precisión de modelos más débiles. Al combinar estos modelos débiles, 'Impulso Adaptativo' crea un modelo fuerte, capaz de realizar predicciones precisas.

métodos predictivos como *ARIMA*, *MLPNN* (Multilayer Perceptron Neural Network)³⁰, *SVR* o *ELM* (Extreme Learning Machine)³¹, e incluso luego ensamblados con el *ADABOOST*, a partir de dos indicadores como son el *MAPE* (Mean Absolute Percentage Error)³². De todas las combinaciones de hiperparámetros y rezagos que se exploraron en el estudio mencionado, el *AdaBoost-LSTM* logra los mejores resultados predictivos por un buen margen, concluyendo respecto de sus bondades en el análisis de series de tiempo financieras. Sin embargo, el documento asume demasiado y no brinda antecedentes de cálculo o inferencias propias, ni siquiera menciona cuestiones ligadas al tratamiento de los datos de entrada, escalado, o período analizado.

Mokhtari, Yen y Liu (2021), pusieron a prueba diferentes modelos de Machine Learning, tales como *Artificial Neural Networks*, *Arboles de decisión*³³, *Bosques Aleatorios*, *XGBoost*,³⁴ *LSTM*, entre otros, para predecir precios de acciones de la compañía Apple. Los autores proyectan los precios al cierre, contemplando la serie histórica, e incorporando algunos indicadores de análisis técnico. Luego de comparar los resultados obtenidos con los diferentes modelos, concluyen que la regresión lineal posee un desempeño superior a LSTM. Sin embargo, la flexibilidad de éste último, les permitió mejorar su rendimiento a través de la incorporación de eventos tales como tweets o artículos lanzados al mercado de manera pública. Los autores concluyen sin embargo, que el mejor modelo es *Support Vector Machine* (SVM)³⁵, aunque destacan que posee un margen considerable de error en la predicción. Finalmente, al intentar prever el precio de cierre de las acciones para el día siguiente, observaron que todos los modelos estudiados, incluida la regresión lineal, perdieron capacidad predictiva. Como resultado, concluyeron sobre la efectividad limitada de los modelos de inteligencia artificial en este contexto, advirtiendo sobre su escasa utilidad en el estado actual de la investigación.

³⁰ Un tipo específico de red neuronal artificial compuesta por múltiples capas de nodos (neuronas), incluyendo una capa de entrada, capas ocultas y una capa de salida.

³¹ Una técnica de aprendizaje automático que se destaca por su entrenamiento rápido y eficiente de una sola capa oculta en redes neuronales. ELM es conocido por su simplicidad y capacidad para manejar grandes volúmenes de datos.

³² Una métrica de evaluación de modelos que calcula el promedio del porcentaje absoluto de error entre las predicciones y los valores reales. Es comúnmente utilizado en problemas de regresión para medir la precisión porcentual promedio del modelo.

³³ Son utilizados para clasificar instancias o predecir valores al seguir el camino desde el nodo raíz hasta una hoja. Se caracterizan por su capacidad para manejar datos categóricos y numéricos, así como por su interpretabilidad.

³⁴ "Extreme Gradient Boosting", es una implementación eficiente y escalable del algoritmo de gradient boosting. Es ampliamente utilizado para problemas de regresión y clasificación. XGBoost utiliza árboles de decisión como modelos débiles y se enfoca en la optimización de funciones de pérdida específicas, lo que mejora la precisión y la velocidad de convergencia. Además, incluye regularización para controlar el sobreajuste y tiene capacidades de manejo de datos faltantes.

³⁵ Busca encontrar un hiperplano óptimo que separe las instancias de diferentes clases en un espacio de características. Puede manejar eficazmente conjuntos de datos de alta dimensionalidad y es especialmente efectivo en espacios no lineales a través del uso de funciones kernel, que mapean los datos a dimensiones superiores.

Maiti y Shetty (2020) llevaron a cabo una comparación entre la arquitectura LSTM y el algoritmo *Red Generativa Adversarial* (GAN)³⁶. Utilizando datos históricos de precios e indicadores técnicos, incorporaron la novedad de la segmentación continua. Al emplear este enfoque de análisis dinámico de los datos, concluyeron que LSTM supera a GAN, aunque con una precisión direccional de predicción no muy elevada.

Por otro lado, Bhandari, Rimal y Pokhrel (2022) llevaron a cabo pruebas con dos tipos de arquitectura LSTM, una multicapa y otra unicapa. Su objetivo fue predecir el precio de cierre para el índice S&P 500, utilizando como variables de entrada precios históricos, indicadores técnicos, índices de volatilidad, así como algunas variables macroeconómicas como la tasa de desempleo y las tasas de interés, entre otras. Al comparar los rendimientos de ambas arquitecturas, observaron que la unicapa supera a su contraparte. Los autores concluyeron su estudio afirmando que el modelo LSTM presenta resultados prometedores para la predicción de precios futuros, basándose en información técnica, fundamental y macroeconómica pasada.

Nguyen, Tran & Nguyen (2019), introdujeron una arquitectura de LSTM dinámico y la compararon con LSTM estático y *Exponential Moving Average* (EMA)³⁷. Utilizaron únicamente datos de precios históricos de Apple y probaron múltiples configuraciones para LSTM. La arquitectura dinámica del modelo implica que cada nuevo dato se incorpora al conjunto de entrenamiento para recalibrar los parámetros del mismo, lo que significa que está en constante evolución y aprendizaje. Los resultados obtenidos revelan que, debido a su proceso de actualización permanente, LSTM dinámico muestra un desempeño superior a los modelos alternativos evaluados.

A partir de la revisión de la bibliografía reseñada, se encuentra valor exploratorio en los modelos LSTM, entendiéndolos que podrían tener potencial en el área del análisis de Series de Tiempo Financieras. En este sentido, se busca evaluar el rendimiento del Modelo LSTM en relación a sus distintas versiones y arquitecturas, sin compararlo con otros métodos predictivos. Por lo tanto, según la calificación referida al comienzo de la Sección, esta investigación se corresponde al tercer subgrupo de estudios.

3.-REDES NEURONALES ARTIFICIALES

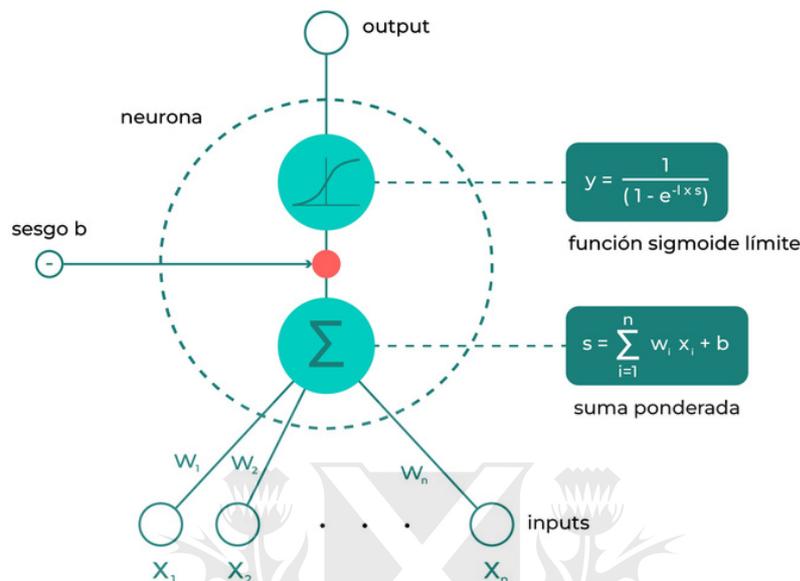
El Deep Learning es una subdisciplina del Machine Learning que se enfoca específicamente en el entrenamiento de Redes Neuronales Artificiales. Inspiradas en la estructura del cerebro humano, estas redes son modelos matemáticos empleados en el aprendizaje automático. A través de unidades simples, denominadas "neuronas", las cuales reciben una serie

³⁶ Consiste en dos redes neuronales, el generador y el discriminador, que se entrenan de manera adversarial. El generador crea datos sintéticos para imitar datos reales, mientras que el discriminador evalúa la autenticidad de los datos.

³⁷ Es una técnica utilizada en análisis de series temporales y estadísticas. Es una versión del promedio móvil que asigna más peso a los datos más recientes.

de entradas con valores reales del entorno y generan una única salida también con valor real. Estas neuronas, a su vez, constituyen el bloque básico de las redes neuronales.

Figura 1: Representación gráfica de una neurona



Fuente: Extraído de Staudemeyer, Morris, et al (2019).

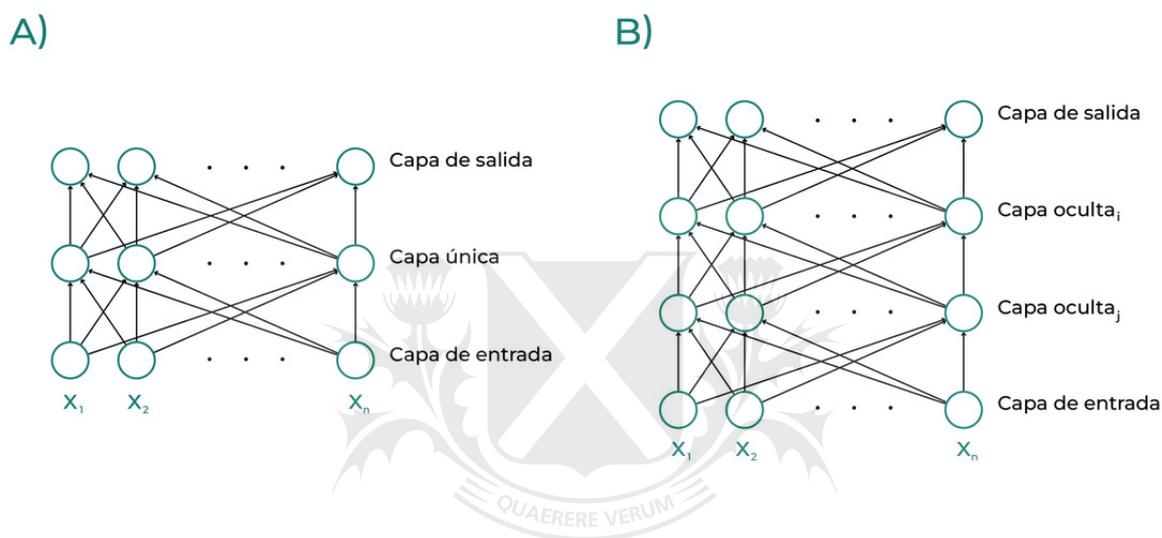
En la *Figura 1* vemos la representación gráfica de una neurona y su funcionamiento. Esta unidad de procesamiento toma múltiples entradas (x_1, \dots, x_n), cada una multiplicada por un peso correspondiente (w_1, \dots, w_n). Los pesos iniciales son aleatorios, generalmente tomando valores muy pequeños, y se ajustan durante el entrenamiento del modelo para mejorar la predicción. Luego, los productos ponderados se suman y se le agrega un sesgo. La salida se determina aplicando una función de activación, originalmente una función escalón (step). Si la suma ponderada más el sesgo supera un umbral, la salida se "activa" (1); de lo contrario, se "desactiva" (0). Durante esta fase, la salida generada se compara con la etiqueta asignada al dato real: si la clasificación no es precisa, se procede al ajuste de los parámetros³⁸.

Para formar una red neuronal, simplemente necesitamos agrupar las neuronas una al lado de la otra. Al hacerlo, obtenemos una red neuronal de una sola capa. Si luego apilamos una red neuronal de una sola capa sobre otra, obtenemos una red neuronal de múltiples capas, como se muestra en la *Figura 2*. Las primeras constan de un conjunto de neuronas de entrada, definidas como la capa de entrada, y un conjunto de neuronas de salida, definidas como la capa de salida. En este tipo de redes, las neuronas de una y otra capa están directamente conectadas, y los pesos se aplican a las conexiones entre la capa de entrada y la capa de salida. Sin embargo, en conjuntos de neuronas organizados en varias capas, las capas de entrada y salida están

³⁸ Este tipo de entrenamiento se denomina *supervisados*, es decir que el modelo aprende por imitación: hay una etiqueta contra la que se compara la predicción del modelo.

conectadas a través de al menos una capa oculta. El término "oculto" se refiere a neuronas que no reciben estímulos directos del entorno ni los devuelven al entorno. A diferencia de las neuronas de entrada o salida, las ocultas reciben información procesada previamente por otras neuronas, mientras que cada neurona de entrada recibe señales del entorno y las neuronas de salida presentan esas señales al entorno.

Figura 2: Redes Neuronales Uniacapa y Bicapa



Fuente: Extraído de Staudemeyer, Morris, et al (2019).

Al aprender representaciones jerárquicas de datos, las RNA son efectivas para tareas de procesamiento de imágenes, procesamiento de lenguaje natural, y reconocimiento de patrones complejos, sin embargo su alcance no se limita a estas áreas. Existió un avance notable en la arquitectura de las redes: al dotarlas de funciones de activación más complejas³⁹, pudieron capturar relaciones no lineales entre los valores de entrada y salida, volviéndose útiles para resolver tareas más desafiantes. De hecho dada su habilidad para manejar datos secuenciales, se convirtieron en una herramienta muy común en el análisis de datos climáticos, en la gestión energética, en biomedicina, y por supuesto en el mercado financiero.

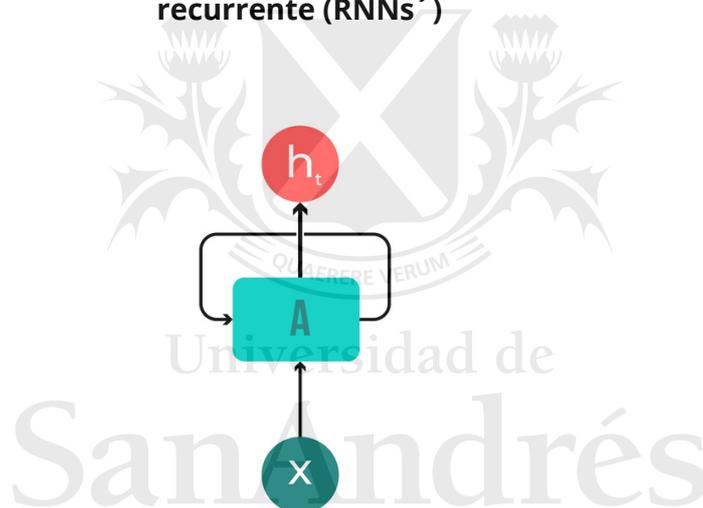
Si bien el uso de redes neuronales para predecir series temporales se remonta a varias décadas, su aplicación en predicciones precisas ha evolucionado significativamente con el avance de técnicas más sofisticadas, como los modelos LSTM. Estos modelos son capaces de recordar patrones a largo plazo en los datos secuenciales, lo que los hace especialmente útiles para problemas de predicción en los que la dependencia temporal es crucial.

³⁹ Las funciones de activación más comunes son la Sigmoide, la Tangente Hiperbólica (TanH), la Unidad Lineal Rectificada (ReLU) y la Unidad Lineal (LU).

4.-LSTM

El modelo Long Short-Term Memory (LSTM) es un tipo de red neuronal recurrente, diseñada para manejar datos secuenciales como las series de tiempo. Las Redes Neuronales Recurrentes (RNN), a las que pertenece el LSTM, representan un avance significativo sobre las redes tradicionales en varias aplicaciones debido a su capacidad para capturar dependencias a lo largo del tiempo. Para comprender mejor el concepto de "capturar dependencias a lo largo del tiempo", podemos trazar una analogía: al igual que los seres humanos, que no comenzamos cada pensamiento desde cero, al leer un artículo, por ejemplo, no analizamos e interpretamos cada palabra de manera independiente. Existe una secuencia de pensamientos continuos que se desarrollan con nuestra capacidad de análisis. Estos pensamientos tienen una persistencia. Las redes neuronales recurrentes abordan este problema, que siempre ha sido un desafío para las redes tradicionales.

Figura 3: Funcionamiento en t de una red neuronal recurrente (RNNs´)



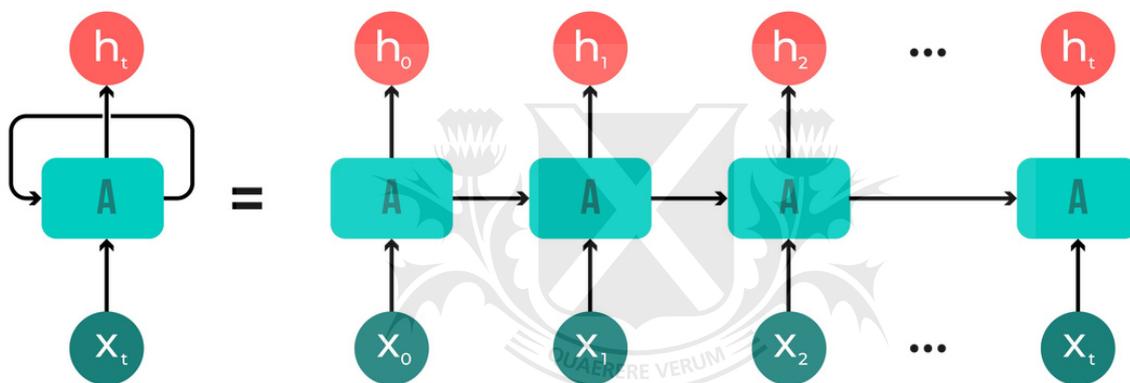
Fuente: Elaborado por el autor basándose en datos recopilados de colah.github.io/posts/2015-08-Understanding-LSTMs/, (2015).

Las RNN (Redes Neuronales Recurrentes) son redes que contienen bucles en su arquitectura, lo que les permite mantener y procesar información a lo largo del tiempo. En la Figura 3 se muestra el funcionamiento de un fragmento de la red neuronal: la neurona 'A' examina una entrada x_t y produce un valor h_{t-1} . Lo innovador es que existe un proceso de retroalimentación que permite que la información se transmita de un paso de la red al siguiente. Si desenrollamos los bucles, podemos pensar en una red neuronal recurrente como múltiples copias de la misma red, cada una enviando un mensaje a su sucesor, tal como se muestra en la Figura 4.

Sin embargo, a pesar de la gran mejora introducida, cuando una RNN entrena en secuencias largas, como texto o series temporales, la red tiende a olvidar información importante a medida que avanza en el tiempo. Esto se debe a que, durante la retropropagación del error (el

proceso de ajustar los pesos de la red), los gradientes tienden a hacerse muy pequeños, lo que resulta en una actualización mínima de los pesos en las capas anteriores de la red. En resumen, las RNN convencionales tienen dificultades para capturar relaciones a largo plazo en los datos, ya que tienden a dar más peso a la información reciente y a perder información relevante en el pasado distante. Este problema se conoce como "desvanecimiento del gradiente". El modelo LSTM, propuesto por Hochreiter y Schmidhuber en 1997, surge precisamente como solución a este dilema.

Figura 4: Desarrollo en cadena de una RNN´s



Fuente: Elaborado por el autor basándose en datos recopilados de colah.github.io/posts/2015-08-Understanding-LSTMs/, [(2015)].

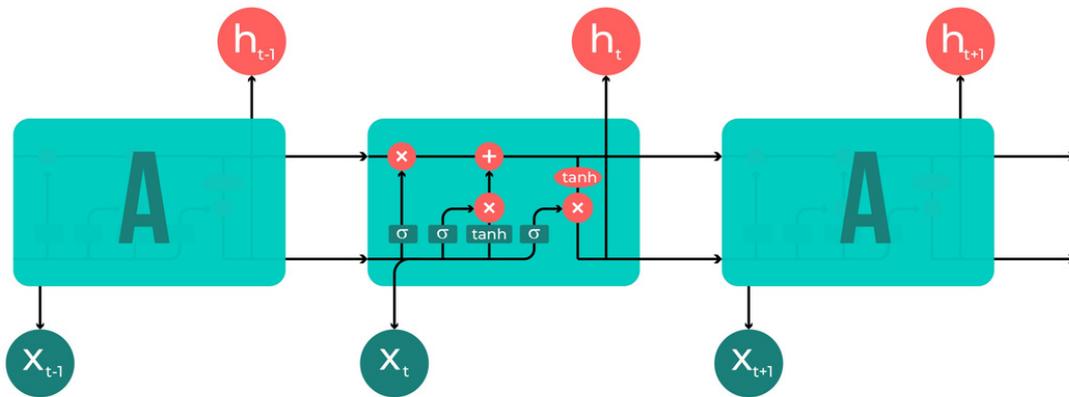
La Figura 5 representa la arquitectura de una celda LSTM, y anticipa porque la calidad predictiva de un modelo LSTM es muy superior al resto de las RNN. La estructura es similar a la de las RNN: es dinámica y se concatena por secuencias. La diferencia está en la incorporación de una *celda de estado*⁴⁰, clave para dotar de capacidad a la red y para mantener y gestionar secuencias de datos a lo largo del tiempo. Graves (2007) es muy didáctico al comparar esta celda con una cinta transportadora la cual se extiende recto a lo largo de toda la cadena de módulos⁴¹. A diferencia de las redes neuronales tradicionales⁴², esto permite que la información fluya a lo largo de toda la secuencia sin modificaciones explosivas. A cada *paso de tiempo*, existe un proceso interno de selección.

⁴⁰ Gráficamente es la línea horizontal que atraviesa la parte superior del diagrama.

⁴¹ Ver figura 5.

⁴² En donde la información fluye linealmente de módulo a módulo en forma incremental.

Figura 5: Las importancia de la Celda de Estado en la Arquitectura LSTM



Fuente: Elaborado por el autor basándose en datos recopilados de colah.github.io/posts/2015-08-Understanding-LSTMs/, [(2015)].

La Figura 6 muestra en detalle los componentes de la celda LSTM, lo cual resulta extremadamente útil para comprender su funcionamiento. A continuación, se detalla cómo opera cada uno de estos componentes.

El primer paso del Modelo LSTM (1) consiste en decidir qué información se descartará de la state cell. Esta decisión la toma una capa sigmoide llamada “puerta de salida”, la cual capta información de x_t y h_{t-1} , generando una salida entre 0 y 1. Un ‘1’ representa “conservar completamente”, mientras que un ‘0’ representa “eliminar completamente”.

$$f_t = \sigma(W_f x_t + W_{fh} h_{t-1} + b_f) \tag{1}$$

El siguiente paso consiste en decidir qué nueva información se almacenará en el estado de la celda. Esto tiene dos partes: en la primer ecuación (2), una capa sigmoide llamada “capa de entrada” decide qué valores se actualizarán, para que luego, una capa tangente hiperbólica (tanh) crea un vector de nuevos valores candidatos, \tilde{C}_t , que podrían sumarse al estado (3).

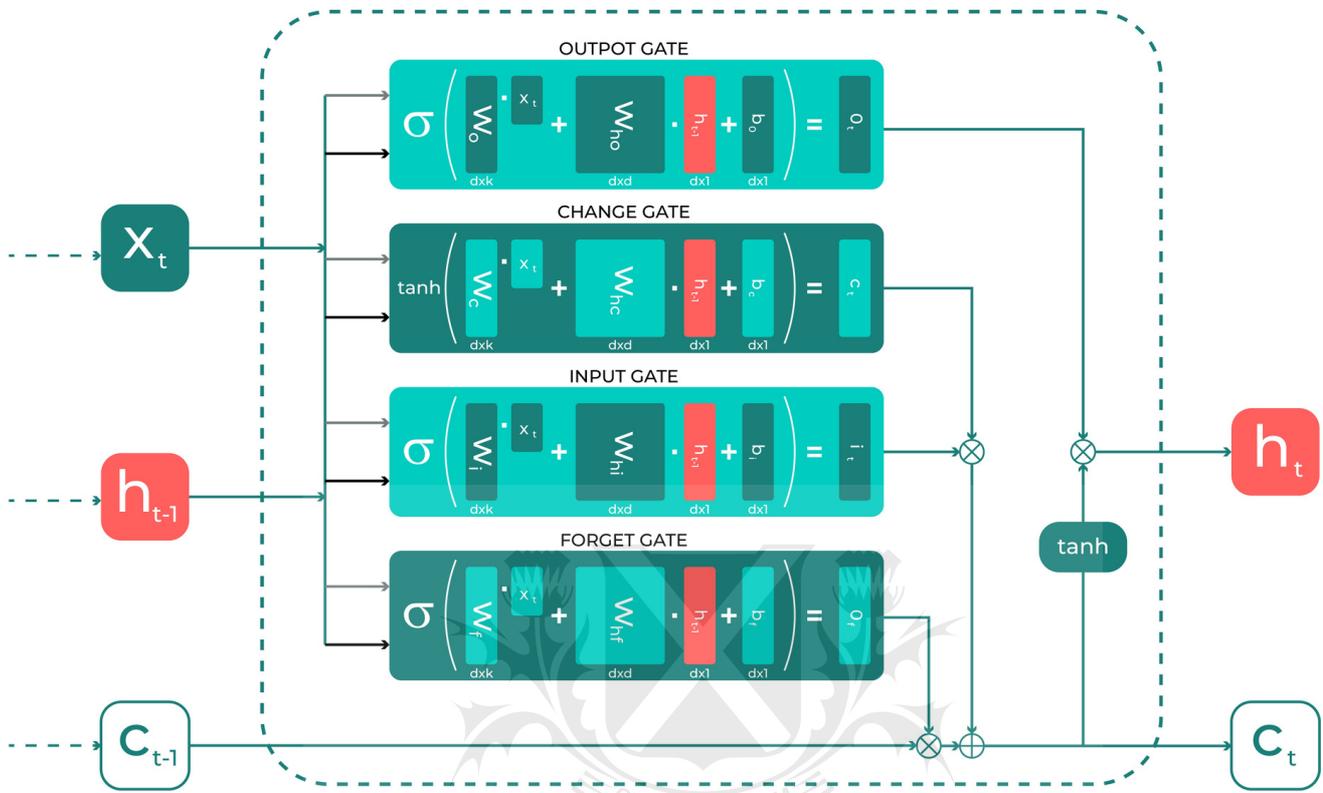
$$i_t = \sigma(W_i x_t + W_{ih} h_{t-1} + b_i) \tag{2}$$

$$\tilde{C}_t = \tanh(W_c x_t + W_{ch} h_{t-1} + b_c) \tag{3}$$

El próximo paso consiste en actualizar el estado de celda, de C_{t-1} multiplicando el estado anterior por f_t , y luego se suma $i_t * \tilde{C}_t$ (4):

$$C_t = f_t \otimes C_{t-1} \oplus i_t \otimes \tilde{C}_t \tag{4}$$

Figura 6: Arquitectura de una Neurona LSTM



Fuente: Extraído y traducido de Bhandari, Rimal, Pokhrel, et al (2022).

Finalmente, es determinante establecer la salida del modelo, la cual estará implícito en el estado de la celda, pero será una versión filtrada: Primero, se ejecutará una capa sigmoide que decide qué partes del estado de la celda correrán (5), y luego, se evoluciona el estado de la celda a través de la función tangente hiperbólica (\tanh) (para limitar los valores entre -1 y 1) y se lo multiplica por la salida de la compuerta sigmoide (6).

$$o_t = \sigma(W_o \cdot x_t + W_{ho} \cdot h^{t-1} + b_o), \tag{5}$$

$$h_t = o_t \otimes \tanh(c_t) \tag{6}$$

dónde σ y \tanh representan las funciones sigmoide y tangente hiperbólica, respectivamente. El

operador \otimes denota el producto elemento a elemento,

$W \in \mathbb{R}^{d \times k}$, $W_h \in \mathbb{R}^{d \times d}$ son matrices de pesos, y

$b \in \mathbb{R}^{d \times 1}$ son vectores de sesgo. Además, n , k y d son la longitud de la secuencia, el número de características y el tamaño oculto, respectivamente.

5.MODELADO

En el presente trabajo se empleó el modelo LSTM para predecir el precio del día siguiente de cuatro activos financieros considerando dos espacios temporales. Como se anticipó en párrafos anteriores, el modelo LSTM tiene una estructura que le permite calibrar los parámetros según un criterio de ajuste preestablecido para optimizar la predicción. Esta estructura es a la vez flexible, ya que depende de los valores que toman los hiperparámetros. A diferencia de los parámetros, estos son especificaciones cuyos valores no se aprenden ni ajustan automáticamente durante el proceso de entrenamiento, pero afectan significativamente el rendimiento y la capacidad del modelo para capturar patrones en los datos. Es el investigador quien se encarga de asignar valores a los hiperparámetros.

Evaluar el poder predictivo del Modelo LSTM implica analizar la respuesta a distintas combinaciones de hiperparámetros en términos de eficiencia. Esta búsqueda de la combinación óptima se conoce como grid search o ajuste de hiperparámetros.

En la presente investigación, la cantidad de combinaciones no dependerá exclusivamente de los hiperparámetros. Con el fin de ampliar el contenido, se contemplaron más variables dentro del modelo, como el conjunto de datos de entrada, los optimizadores y el tipo de dirección del análisis de secuencias.

A continuación, se desarrollarán las distintas etapas del proceso pertinente con su debida argumentación y detalle bibliográfico, según el esquema sugerido por la Figura 7.

Figura 7: Proceso esquematizado



Fuente: Elaborado por el autor.

Selección de activos

Los precios de los activos financieros responden a distintas variantes según su naturaleza. De la bibliografía empleada, se pudo acceder a un centenar de trabajos que evalúan distintas series financieras. Lo habitual es analizar el índice SP&500, pero también se estudian series de acciones particulares como Apache, Walmart, Amazon, Apple, Google, así como distintas monedas, ETF y diversos índices bursátiles de todo el mundo, como el de Shanghái o Corea. En un proceso de búsqueda y optimización de hiperparámetros, resulta relevante analizar series de activos de distintos sectores de la economía para poder comparar sus resultados.

En la presente investigación se seleccionaron cuatro activos de manera aleatoria, con la premisa de que sus negocios pertenezcan a distintos sectores de la economía:

- Lantheus Holdings, Inc. (LNTH): Una empresa dedicada al desarrollo, producción y comercialización de productos destinados al diagnóstico y tratamiento de enfermedades del corazón, cáncer y otras patologías. También invierte en inteligencia artificial, especialmente para cirugía robotizada, y tiene grandes avances en el área de radiología.
- Wells Fargo (WFC): Una de las instituciones financieras más grandes de Estados Unidos, con una presencia significativa en el sector bancario y financiero. Ofrecen una amplia gama de servicios financieros, que incluyen banca minorista, banca corporativa, banca de inversión, hipotecas y gestión de activos.
- MercadoLibre (MELI): Una de las principales plataformas de comercio electrónico y tecnología financiera en América Latina. Ofrecen una amplia gama de servicios, desde compras en línea hasta soluciones de pago y servicios de entrega.
- Marvell Technology Group (MRVL): Se especializa en la fabricación de semiconductores y soluciones de almacenamiento. Sus productos se utilizan en una variedad de aplicaciones, incluyendo electrónica de consumo y centros de datos.

Definición de variables explicativas, secuencias y preparación del dataset

De la revisión de los trabajos referenciados en la literatura, se identificaron dos clases de investigaciones según su objetivo: los modelos orientados a predecir el precio del día siguiente y aquellos destinados a predecir la dirección del movimiento. Aunque ambas están relacionadas, son fundamentalmente diferentes. La predicción de precios implica una tarea de regresión, mientras que la predicción del movimiento direccional es una tarea de clasificación.

En una tarea de regresión, se busca predecir un valor continuo como salida, mientras que en una tarea de clasificación se pretende predecir una etiqueta discreta o categoría. Esto es crucial en la configuración y programación del modelo, ya que no solo se redefine la capa de salida como una función escalón binaria, sino que toda la estructura cambia, incluyendo optimizadores, objetivos, criterios de optimización y el formato de la variable de entrada al modelo.

En este trabajo, se utilizó el precio de cierre ajustado para la predicción, respaldado por la evidencia empírica y académica. Los trabajos de Ma (2020), Choi (2018), Wo y Gao (2018), Ho, Darman y Musa (2021), Donglin Chen y Dissanayaka M. K. N. Seneviratna (2014), y Tannure y Gerez (2023) emplean precios de cierre para sus predicciones debido al objetivo de la investigación, sin realizar transformaciones, y son referencias comparativas para este estudio.

Cuando el objetivo de la investigación es determinar la dirección futura del precio, la evidencia indica que se emplearán retornos. Por ejemplo, Krauss (2018) utiliza las variaciones diarias del precio para anticipar su reacción, y Khaled A., Salahadyn A., El-Sayed M. (2021) realizan un análisis de multiresolución en la construcción de portafolios eficientes, descomponiendo la serie de retornos con ondas de Haar para mejorar las predicciones. Kamalov, Gurrib y Rajab (2021) proporcionan evidencia sobre la utilización de precios versus retornos, concluyendo que, incluso en modelos destinados a determinar la dirección de la serie, se obtienen mejores resultados utilizando precios en lugar de retornos. Como complemento a esta sugerencia, se realizó una predicción adicional utilizando los retornos diarios de MELI en un total de 2880 combinaciones distintas, sin obtener una aproximación de ajuste y predicción decente. En el Anexo, Figura I, se presentan tres combinaciones en gráficos para ilustrar los resultados.

Respecto a las variables explicativas que se incorporaron como entradas al modelo, se propusieron dos grupos: uno donde solo se contempló el Precio Ajustado de Cierre y otro en el que además se incluyó al⁴³, RSI⁴⁴, MACD⁴⁵, USD⁴⁶ y TNX⁴⁷, MULTIFEATURE de ahora en VIX adelante. Una porción menor de la literatura, argumenta la inconveniencia de introducir características a la entrada más allá del Precio de Cierre Ajustado, no solo por que no suma efectividad, sino que también, aumenta el riesgo de sobreajuste. Estas investigaciones señalan que el precio pasado es suficiente como entrada al modelo, puesto que ya incorpora la coyuntura,

⁴³ VIX (Índice de Volatilidad del CBOE), es un índice que mide la volatilidad implícita en el mercado de valores de EE. UU.

⁴⁴ Relative Strength Index es un indicador de impulso utilizado en análisis técnico para evaluar si un activo financiero está sobrecomprado o sobrevendido. El RSI oscila entre 0 y 100 y se calcula comparando las ganancias y pérdidas de un activo en un período de tiempo específico.

⁴⁵ *Moving Average Convergence Divergence* es un indicador de análisis técnico que se utiliza para identificar la tendencia y los puntos de entrada o salida en el mercado de valores. Se calcula a partir de las medias móviles exponenciales de un activo financiero. El MACD muestra la diferencia entre dos medias móviles exponenciales (generalmente, una a corto plazo y otra a largo plazo) y también incluye una línea de señal. Este indicador puede ayudar a los inversores a identificar cambios en la dirección de la tendencia.

⁴⁶ Variación en el valor del Dolar.

⁴⁷ Rendimientos de Bonos del Tesoro de US a 10 años

Fundamentales y Analisis Técnico, vía expectativas. Si las características propuestas en MULTIFEATURE, a pesar de guardar correlación con el objetivo, no aportan valor agregado a la predicción, podría análogamente establecerse, que son relaciones espurias, tal como lo definió Yule (1925) en procesos regresivos, ya que a pesar de encontrar un buen porcentaje de correlación, no agrega precisión a las predicciones.

La incorporación de estas características a la entrada estuvo sujeta a una evaluación previa de las series mediante mapas de calor, que se presentan en las Figuras A, B, C y D del Anexo. Estas gráficas muestran la correlación entre las características propuestas y la variable objetivo, evidenciando diferencias sustanciales. Esto indica que los precios de los activos no responden de la misma manera a los mismos estímulos. Esta diferencia puede ser un factor clave al buscar patrones generales, lo que sugiere que solo se debería utilizar el precio ajustado de cierre como variable explicativa. Aunque las acciones pertenecen a distintos sectores, sería prudente individualizar las características a emplear como entrada. La generalización sería viable solo con fines exploratorios.

En este trabajo, se consideró que gran parte del impacto de factores fundamentales y coyunturales macroeconómicos sobre los precios ya se había incorporado a través del rendimiento de los bonos a 10 años y el precio del dólar. Por lo tanto, añadir más características podría introducir ruido innecesario. No obstante, no se descarta la posibilidad de evaluar otras características de interés en trabajos futuros.

Por último, se definieron los períodos de análisis. Como se indicó en los párrafos anteriores, es crucial comprender que los precios absolutos no son estacionarios en su comportamiento a lo largo del tiempo, a diferencia de los retornos que sí lo son. Como aseguran Gujarati y Porter (2010), el análisis de series no estacionarias con fines predictivos carece de sentido en la práctica, razón por la cual esta investigación pone énfasis en periodos acotados.

Al analizar cualquier serie financiera, se desprende que el período en estudio sobre el cual se aplique el modelo tendrá un impacto determinante en los resultados de las predicciones. El criterio empleado para la elección de la longitud de los periodos fue empírico y técnico. Algunos trabajos, como el de Krauss (2017), utilizan 3 años de datos para entrenamiento y 1 año para validación, obteniendo buenos resultados. Aunque esto no es una regla fija, ya que depende del objetivo de la investigación, evaluar la respuesta del modelo a distintos niveles de volatilidad, eventos importantes, cambios de tendencias y la presencia de estacionalidad es fundamental.

Para esto, se decidió analizar un período corto sin eventos significativos y uno más largo que evidencie lo mencionado anteriormente, con el fin de contrastar los resultados. Los períodos de estudio van desde enero de 2014 hasta el 10 de octubre de 2023 y desde enero de 2020 hasta la misma fecha. Si bien la selección de diferentes longitudes de período es crucial en el análisis realizado, también lo es la segmentación de los datos en conjuntos de entrenamiento y prueba.

En esta investigación, se utilizó un 80% de los datos para entrenamiento, un 10% para validación y un 10% para prueba, valores ampliamente empleados en la literatura. El objetivo es

determinar si el modelo LSTM realmente tiene una ventaja sobre otros modelos predictivos debido a su capacidad de memoria a largo plazo.

Normalización de datos

En el caso de que la entrada al modelo esté compuesta por más de una característica, es evidente la necesidad de homogenizar los datos para que sean procesables. En este trabajo, se decidió escalar los datos utilizando el criterio de mínimos y máximos, ya que es una sugerencia que se repite en la literatura con buenos resultados. Es importante destacar que, últimamente, algunos autores han desalentado este tipo de estandarización y proponen una normalización z- value debido a las características de la distribución de los retornos. Sin embargo, estas críticas son aisladas y no hay evidencia de que su aplicación conduzca a mejores resultados.

Con el objetivo de centrar el análisis en la combinación de hiperparámetros que haga eficiente la predicción y poder comparar los resultados con otros modelos calificados de terceros, se optó por utilizar el mismo criterio de escalado para no cambiar el contexto en el que se aplica el modelo. Por lo tanto, se escaló según el Método de Mínimos y Máximos Escalados

(MinMaxScaler)⁴⁸:

$$z = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (7)$$

Optimización

Para evaluar la precisión en la capacidad del modelo, se utilizó una función de pérdida, cuyo valor está vinculado al ajuste entre la predicción y el precio real de la acción en cada momento del tiempo. La función de pérdida elegida fue el Error Cuadrático Medio (MSE), fruto de un amplio respaldo en la literatura. Su ecuación está dada por:

$$MSE = 1/n \sum (y_i - \hat{y}_i)^2 \quad (8)$$

Por otro lado, para calcular y modificar los valores de los parámetros en el proceso de optimización, el modelo cuenta con un algoritmo optimizador. Este algoritmo indica la dirección y magnitud de cambio en estos parámetros. En este trabajo experimental se utilizaron tanto el algoritmo "Adam" como "RMSprop". Adam es un método de gradiente descendente estocástico, basado en la estimación adaptativa de los momentos de primer y segundo orden. Este método es

⁴⁸ El Criterio de MinMaxScaler es una técnica común utilizada en el preprocesamiento de datos para normalizar características numéricas en un rango específico, típicamente entre 0 y 1. En el contexto de modelos LSTM, se utiliza para escalar los datos de entrada a la red neuronal.

computacionalmente eficiente, tiene bajos requerimientos de memoria, es invariante al reescalado diagonal de gradientes y está bien preparado para problemas que son grandes en términos de datos/parámetros. Por otro lado, RMSprop (Root Mean Square Propagation) adapta la tasa de aprendizaje para cada parámetro, utilizando una ventana móvil de promedios cuadráticos para normalizar la tasa de aprendizaje.

Construcción del modelo

Los modelos LSTM pueden clasificarse según la dirección temporal que siguen durante el proceso de iteración para la calibración y el análisis de secuencias. Si el modelo avanza en el tiempo, se le llama unidireccional. En cambio, si lo hace en ambas direcciones, se habla de bidireccionalidad. Este último debería ser capaz de capturar mejor las relaciones a largo plazo en una secuencia, ya que tiene acceso a información futura y pasada simultáneamente, lo que le permite ponderar expectativas durante el entrenamiento. Sin embargo, es importante señalar que cualquier modelo supervisado tiene esta capacidad. La literatura no presenta una gran evidencia a favor de algún modelo en particular, por lo que este aspecto fue objeto de análisis en esta investigación. En el presente trabajo, se utilizaron ambos tipos de modelos para evaluar sus rendimientos en un contexto secuencial, utilizando ventanas deslizantes de longitud preestablecida.

Arquitectura

Los modelos LSTM también pueden ser caracterizados según la cantidad de neuronas por capa que se utilicen para procesar información, lo que se conoce como la Arquitectura de la Red. Por un lado, la arquitectura unicapa utiliza una única capa de neuronas para procesar la información y realizar la predicción. En este caso, los estados ocultos finales de cada neurona representan la predicción de dicha neurona para cada momento 't', y estas predicciones se condensan a través de una capa de salida en un único valor que constituye la predicción del modelo. Por otro lado, la arquitectura bicapa presenta una segunda capa de neuronas para procesar los datos. La primera capa de neuronas incorpora los datos de entrada, mientras que la segunda capa utiliza los valores de los estados ocultos de la primera capa como entrada. Es por esto que cada neurona de la segunda capa no se nutre directamente del entorno, sino de la información brindada por cada una de las neuronas de la primera capa. Los estados ocultos finales de la segunda capa representan la predicción de cada conjunto de datos.

Como se mencionó previamente, la evidencia proporcionada por Tannure y Gerez (2023) y Bhandari, Rimal, Pokhrel (2022) sugiere que agregar capas al modelo es ineficiente, ya que implica costos computacionales significativos sin generar una capacidad predictiva que lo justifique. Por lo tanto, este trabajo se centró en el análisis de modelos unicapa para evaluar los rendimientos de esta arquitectura con diferentes cantidades de neuronas por capa. En esta

investigación, se asignaron 16, 32, 64 y 96 neuronas al modelo. La incorporación de más capas no proporciona una mayor capacidad predictiva y puede complicar el proceso y aumentar los recursos de memoria sin generar mejoras significativas. La Figura 8 muestra los resultados de 2300 combinaciones diferentes de modelo LSTM sobre el precio de LNTH para el período 2014-2023, demostrando que en términos de ajuste por MSE, la adición de capas no agrega valor y que, por el contrario, es preferible utilizar una sola capa.

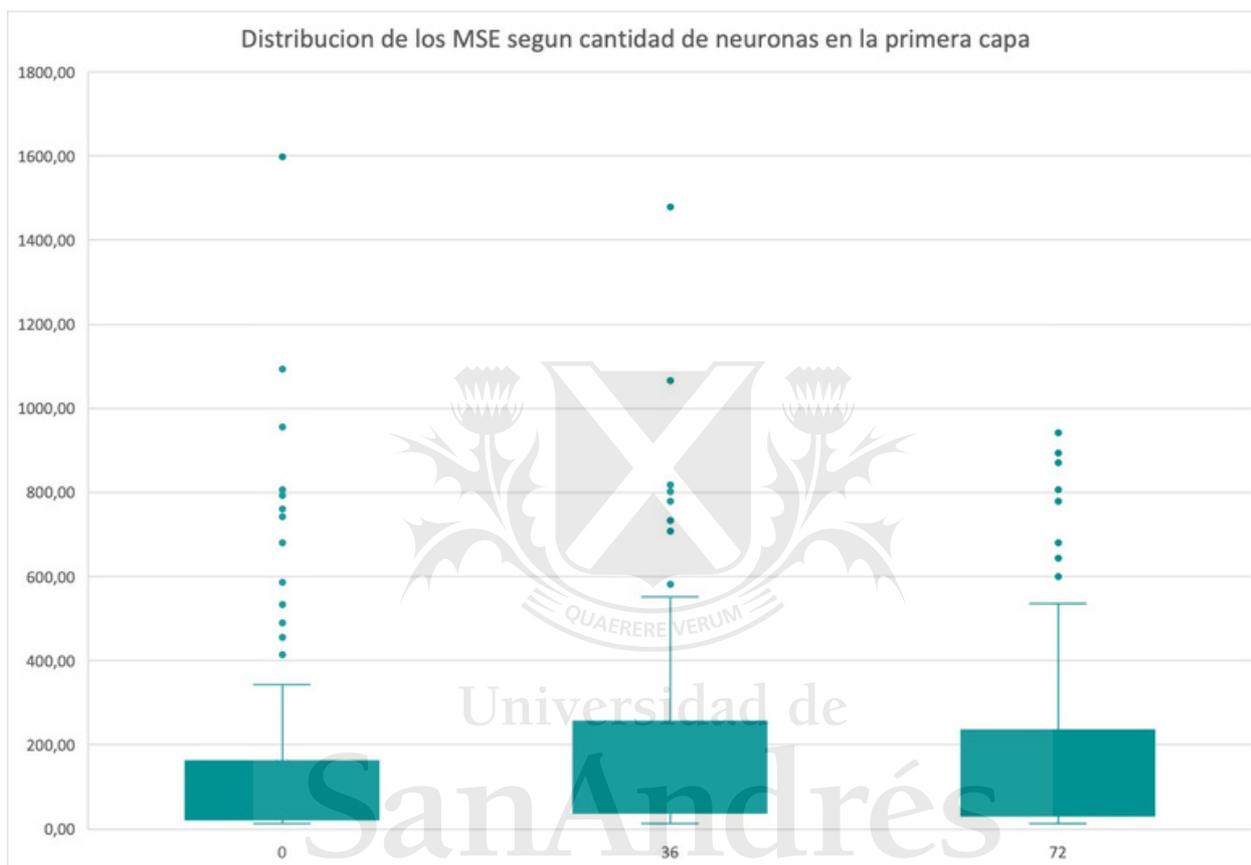


Figura 8: Capacidad de ajuste - Comparativa modelos unicapa y bicapa sobre 2300 simulaciones - SPY

Grid Search

El Cuadro 1 resume el proceso de grid search realizado en esta investigación, que incluye la calibración de hiperparámetros para optimizar el ajuste del modelo LSTM. Se consideraron modelos secuenciales LSTM unidireccionales y bidireccionales, unicapa, con ventanas deslizantes, y se evaluaron varios hiperparámetros clave:

1. Número de Unidades LSTM: Se refiere al número de unidades de memoria o celdas LSTM en la red. Cuantas más unidades, se presupone una mayor capacidad de aprendizaje, pero también implica un mayor uso de recursos computacionales.

2. Tamaño de Lote (Batch Size): Representa la cantidad de muestras de datos procesadas antes de actualizar los pesos del modelo. Un tamaño de lote más grande puede acelerar el entrenamiento, pero a costa de requerir más memoria. Se fijó en 32 debido a resultados satisfactorios en simulaciones previas.

3. Longitud de la Secuencia (Sequence Length o window size): Indica la longitud de las secuencias de datos de entrada que el modelo analiza simultáneamente. Puede influir en cuántos pasos de tiempo se consideran para la predicción.

4. Epochs (épocas): Representa el número de veces que el modelo verá el conjunto de datos completo durante el entrenamiento. Demasiadas épocas pueden conducir a sobreajuste, mientras que muy pocas podrían subutilizar el modelo.

5. Dropout: La técnica de dropout consiste en desactivar de manera aleatoria un cierto porcentaje de neuronas durante el entrenamiento, lo que ayuda a evitar que la red memorice demasiado los datos de entrenamiento específicos y, en cambio, fomente una representación más robusta y generalizable.

Este proceso se realizó para diferentes períodos de tiempo y activos financieros, utilizando características específicas en los grupos de control, distintos tipos de modelos LSTM y optimizadores.

El objetivo fue encontrar la combinación óptima de hiperparámetros que maximizara la capacidad del modelo LSTM para predecir los precios de los activos financieros seleccionados.

COMPAÑIAS						
WELLS FARGO	MELI	MARVELL	LNTH			
SECUENCIAS		DESDE	HASTA			
PERIODO 1		1/01/2	10/10/2			
PERIODO 2		0	3			
GRID SERCH		1/01/1	10/10/2			
MODELO	4 ARQUITECTURA	3 NEURONAS	WINDOWS SIZE	EPOCHS	BATCH SIZE	DROPOUT
BIDIRECCIONAL	UNICAPA	1	10	10	32	0
UNIDIRECCIONAL		6	20	0		10 %
		3	30	20		
		2	40	0		
ENTRADA Y CARACTERISTICAS		6		30		
A- ADJ CLOSE		4		0		
B- MULTIFEATURE	^VIX	^USD	^TNX	RCI	MACD	ADJ CLOSE
OPTIMIZADOR		6				
a) ADAM		B)RMSprop				

Cuadro 1: Esquema del *grid search*

Predicción del retorno y precio del día siguiente

En un modelo con ventanas deslizantes, los datos se ingresan al modelo en secuencias o ventanas de tamaño 'n', donde 'n' representa el número de días consecutivos considerados en cada ventana. Por ejemplo, si tenemos una serie temporal de 2500 días y utilizamos una ventana de tamaño 30, tendremos un total de 2471 muestras para entrenamiento ($2500 - 30 + 1$).

El modelo procesa cada secuencia de datos y realiza una predicción, la cual se compara con el valor real conocido como etiqueta. Si la predicción se ajusta al criterio establecido, el proceso concluye. Sin embargo, si no cumple con el criterio, los parámetros del modelo se ajustan para mejorar la precisión de la predicción.

Para realizar estas predicciones, se empleó un modelo desarrollado en Python 3.12, utilizando entornos de desarrollo en Anaconda y Google Colab. El código completo se encuentra adjunto en el Anexo a este apéndice. Las simulaciones se llevaron a cabo durante un total de 1843 horas continuas, utilizando dos equipos equipados con procesadores Intel Core i7 de 2.07 GHz, de 12ª generación, que operaron simultáneamente.

En términos generales, se exploraron más de 10612 combinaciones de hiperparámetros, y en promedio se ajustaron 75,546 parámetros en cada combinación. La cantidad total de parámetros ajustables en una capa LSTM se calcula sumando los pesos y sesgos asociados con la entrada, el estado anterior y la salida, así como los parámetros de las puertas de la red neuronal.

Número total de parámetros según tipo de modelo unidireccional y bidireccional con window slide:

(NUN) = (cantidad de variables de input x tamaño de la ventana + Número de unidades en la celda LSTM) x 4

(NBID) = 2 x (cantidad de variables de input x tamaño de la ventana + Número de unidades en la celda LSTM) x 4



6.-ANALISIS DE RESULTADOS

MRVL (MARVELL TECHNOLOGY INC.)

En la Figura G del Anexo a este documento, se puede observar la descomposición de la Serie para MRVL. Desagregarla resulta crucial para el análisis estadístico de series de tiempo; en el caso de esta importante tecnológica, se evidencia un comportamiento variable. Se observa que hasta enero de 2020, la serie no presenta volatilidad significativa; sin embargo, a partir de allí el ruido se incrementa, la serie se descontrola y se vuelve vertical, exacerbando la tendencia, destacando un sendero explosivo hasta finales de 2020 y contractivo a partir de entonces. La estadística descriptiva proporcionada en el Cuadro B del Anexo refuerza esta premisa de diferencial en la volatilidad. En el periodo 2020-2023, la desviación estándar es considerablemente mayor que en el periodo 2014-2023, al igual que la media. Estos dos momentos estadísticos, tal como revela el Cuadro A del Anexo, son consistentes con los resultados del Test de Estacionariedad proporcionados en el Cuadro B, que resultan negativos para los dos periodos bajo estudio, aunque es considerable que para 2020-2023 el estadístico se acerca bastante a valores de rechazo de hipótesis nula.

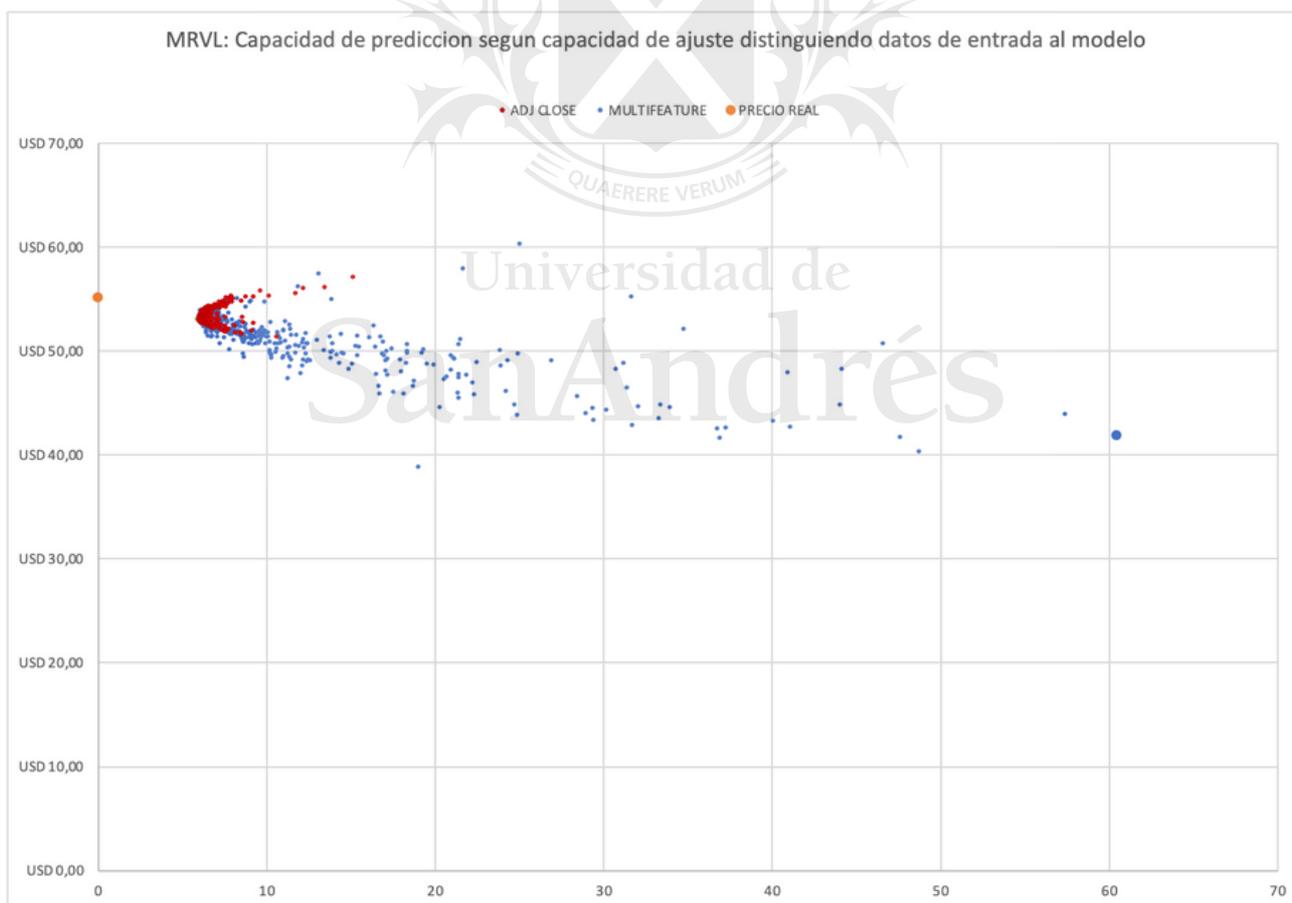


Figura 9: Distribuciones de las predicciones para t+1 según el MSE obtenido

La Figura 9 muestra la dispersión de las predicciones dado el nivel de ajuste del modelo según los datos empleados como entrada para el periodo 2014-2023. De esto se infiere que el

modelo logra ajustar con un MSE en un rango de 5,95 y 10 para el 78,62% de las combinaciones. Sin embargo, existe una mayor dispersión en los resultados al emplear MULTIFEATURE como entrada al modelo. Esta mayor sensibilidad de LSTM a cambios en los datos de entrada conlleva a una pérdida de precisión, puesto que se verifica que más características no implican mejores predicciones. Refuerzan esta inferencia el hecho de que, de todas las arquitecturas disponibles, las nueve con menor MSE utilizan solo el Precio de Cierre Ajustado como dato de entrada. El modelo parece no incorporar el efecto de USD y TNX, que en principio deberían poder explicar la coincidencia entre la volatilidad de la serie y la política monetaria de la FED post pandemia.

En menor medida, se evidencia que las combinaciones con modelos unidireccionales en general obtuvieron menores MSE en comparación con los bidireccionales, como se observa en la Figura 10. Este dato no es menor, puesto que emplear un modelo u otro puede implicar un ahorro en gasto computacional importante.

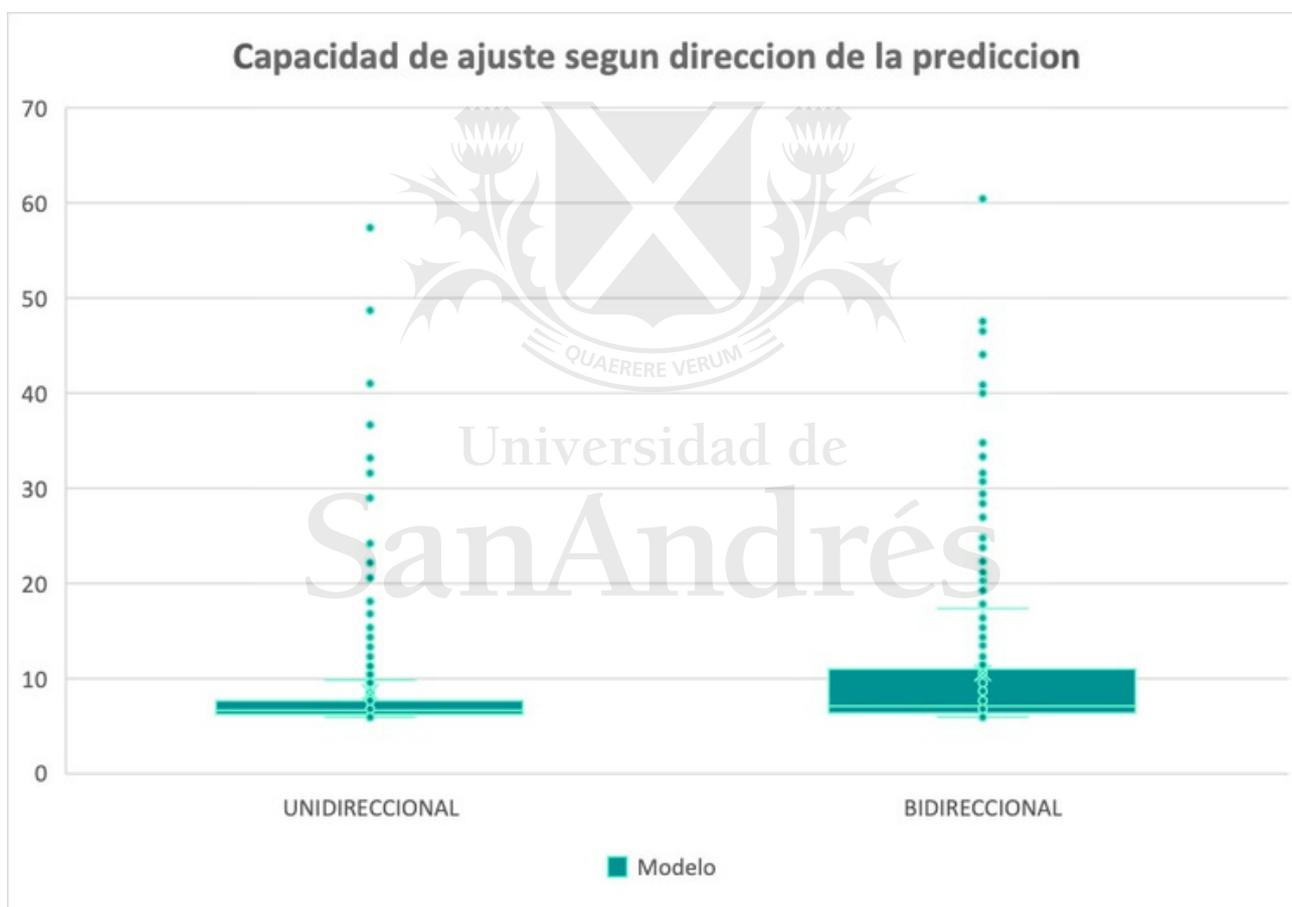


Figura 10: Muestra los MSE resultantes distinguiendo entre modelos Unidireccionales o Bidireccionales

En cuanto a la predicción en $t+1$, LSTM logra un valor en promedio 5% por debajo del dato real, con un desvío relativamente bajo, siendo significativamente menor (3,4%) en toda aquellas arquitecturas que utilizan como entrada solo Precio Ajustado de Cierre.

Entrada	Modelo	Optim.	Neuronas	Window Size	Epochs	Drop out	Batch Size	MSE	Next Day Predictio
ADJ CLOSE	UNIDIRECCIONAL	ADAM	32	10	100	0	32	5,9225	USD 53,09

Cuadro 4: Mejor combinación MRVL 2014-2023

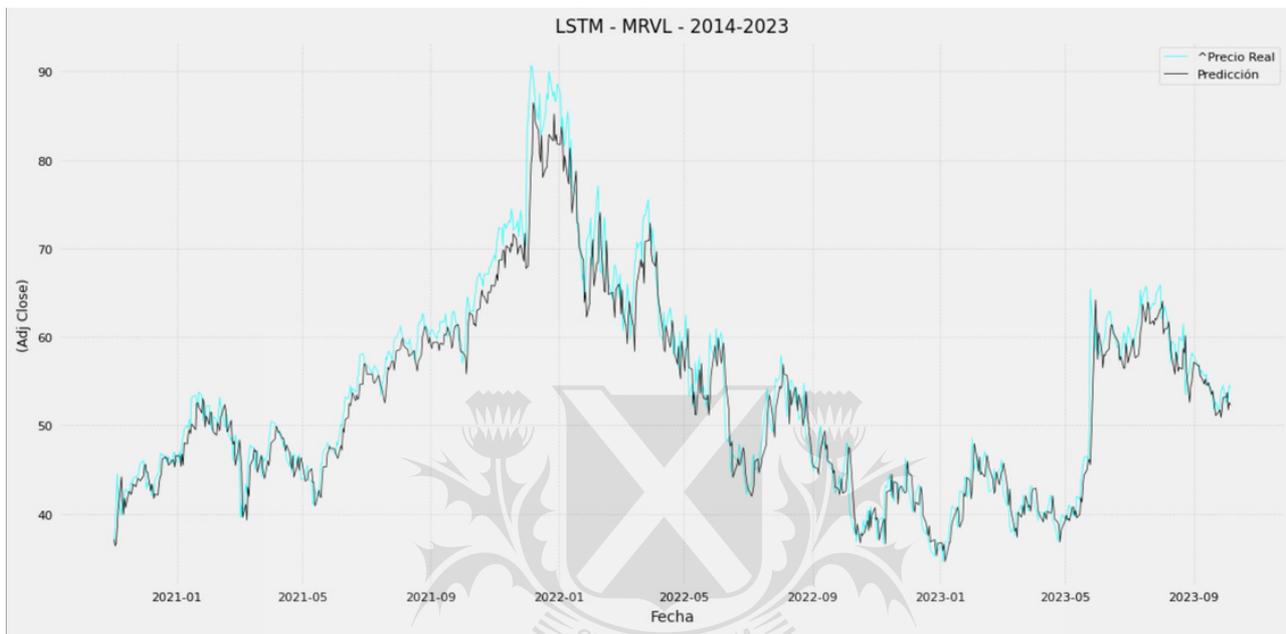


Figura 11: Serie Real y predicciones, combinación que minimiza MSE: MRVL 2014-2023

El mejor ajuste según el criterio preestablecido de MSE es el que se obtiene con un Modelo Unidireccional, como se describe en el Cuadro 4. La serie real y la predicha por el modelo óptimo se representan en la Figura 11. En esta se observa que, a pesar de la volatilidad presente, la respuesta de LSTM es aceptable y, lo que considero importante, muestra homogeneidad a lo largo de toda la serie. Destaco que el sendero resultante de la predicción óptima, y en general del 98% de las combinaciones analizadas, viajan por debajo de la trayectoria de la serie real, lo que podría ser la causa de que la gran mayoría de los precios resultantes del modelo subestimen el precio real en este periodo.

Para el periodo 2020-2023, se observa una mayor desviación estándar en MSE, seguramente en parte debido a la presencia de outliers, como se ilustra en la Figura 12. Este gráfico relaciona la calidad de ajuste del modelo y el nivel de predicción, distinguiendo según el modelo opere con datos MULTIFEATURE o solo con el Precio Ajustado. La similitud en la forma y características de este gráfico con el correspondiente al periodo anterior bien puede ser un indicador de robustez y consistencia en los resultados obtenidos por el modelo LSTM.

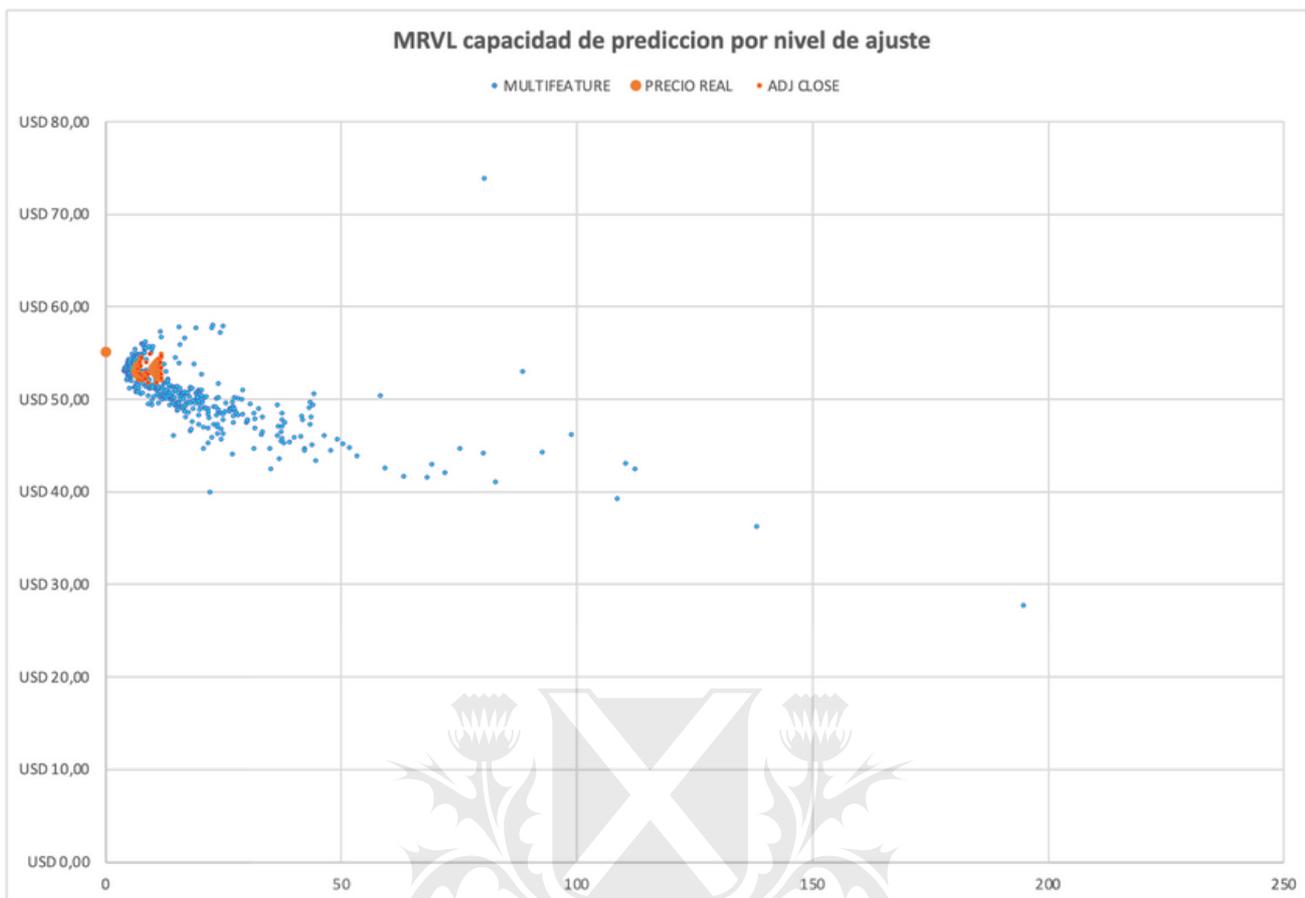


Figura 12: Distribuciones de las predicciones para t+1 según el MSE obtenido

Entrada	Modelo	Optim.	Neuronas	Window Size	Epochs	Drop out	Batch Size	MSE	Next Day Prediction
MULTIFEATURE	UNIDIRECCIONAL	RMSprop	64	40	100	0	32	3,804	USD 53,32

Cuadro 5: Mejor Arquitectura MRVL 2020-2023

En este lapso existen arquitecturas LSTM que obtienen muy buenos ajustes y predicciones, y otras que no logran captar muy bien el movimiento de los precios, lo que resulta en una menor precisión en general. Al comparar las configuraciones que presentan el menor MSE en cada periodo analizado, se observa que para el periodo "corto", la combinación óptima (ver Cuadro 5) tiene un MSE significativamente menor que para el periodo "largo". Sin embargo, a nivel agregado, los resultados en el periodo 2014-2023 muestran menos dispersión y, en promedio, son mejores que en el periodo 2020-2023. Esto podría estar relacionado con el comportamiento de la serie en este trayecto, ya que, aunque se trata de un lapso más corto y homogéneo, presenta más agresividad en sus fluctuaciones, con cambios de tendencia y mayor volatilidad.

Dada las características metodológicas de LSTM, es esperable que su funcionamiento mejore a medida que aumente el volumen de datos de entrada. Sin embargo, esto no se corresponde con los resultados obtenidos hasta ahora. Dada la segmentación de datos y la solapación propuesta, y atentos al comportamiento de la serie, se verifica que al aumentar el volumen de información, se corresponde con un aumento de MSE al incorporar datos de distintos regímenes. En el caso concreto de MRVL, las etiquetas muestran diferencias concretas respecto a los datos de entrenamiento, ya que la serie experimenta cambios abruptos, ganando verticalidad y volatilidad en los últimos años. A la vez, un LSTM aplicado a periodos cortos puede tener ventaja en ese sentido, ya que la calidad de los datos es mejor, sin embargo, tiene menos datos de entrenamiento y de prueba, lo que puede ser negativo.



Figura 13: Predicción/REAL: Mejor Arquitectura según criterio MSE; MRVL 2020-2023

La importancia del grid search queda de manifiesto al comparar las combinaciones óptimas obtenidas en cada periodo, las cuales son diametralmente distintas una de otra. Del análisis agregado se observa que los modelos unidireccionales vuelven a destacarse sobre los bidireccionales, sin embargo, en este periodo se da que algunas combinaciones que usan MULTIFEATURE como entrada ganan precisión en el ajuste contra los que emplean solo el Precio de Cierre Ajustado. Este cambio es bastante notable respecto a los resultados observados en el periodo anterior, especialmente considerando que los modelos que usan MULTIFEATURE, con el optimizador RMSprop y una estructura unidireccional, obtienen los 16 mejores resultados del total de combinaciones posibles.

Sin embargo, es importante realizar un análisis más profundo para concluir al respecto, ya que aunque algunas combinaciones puntuales obtuvieran buenos resultados en comparación, no lo hacen en el espectro general de todas las estructuras posibles. De hecho, en promedio, tanto los MSE como los valores que predice el modelo con MULTIFEATURE son peores que los que obtienen los modelos con una sola característica de entrada. De lo anterior se desprende que la elección de las características no es para nada trivial, puesto que puede aportar valor o bien generar sobreajuste o ruido al entrenamiento del modelo.

La Figura 13 contrasta la serie real contra la que predice el modelo óptimo en el periodo bajo estudio. Aunque el MSE es bajo y la predicción es muy cercana a la etiqueta real, el ajuste en sí no refleja correlación temporal, y esto se evidencia aún más después del salto discreto en junio de 2023; el modelo pierde precisión a partir de ese momento.

WFC (WELL ´S FARGO)

El mapa de calor para los precios de WFC proporcionado en el Anexo es evidentemente distinto al de los demás activos analizados, por lo que resulta importante evaluar el desempeño del modelo según las características empleadas. Igualmente, si comparamos la serie de WFC con las demás, ésta presenta un sendero que, si bien no hay evidencia probatoria que asegure estacionariedad, los estadísticos (Cuadro A del anexo), bajo algún criterio amplio, bien podrían indicarlo, sobre todo para el periodo 2014-2023.

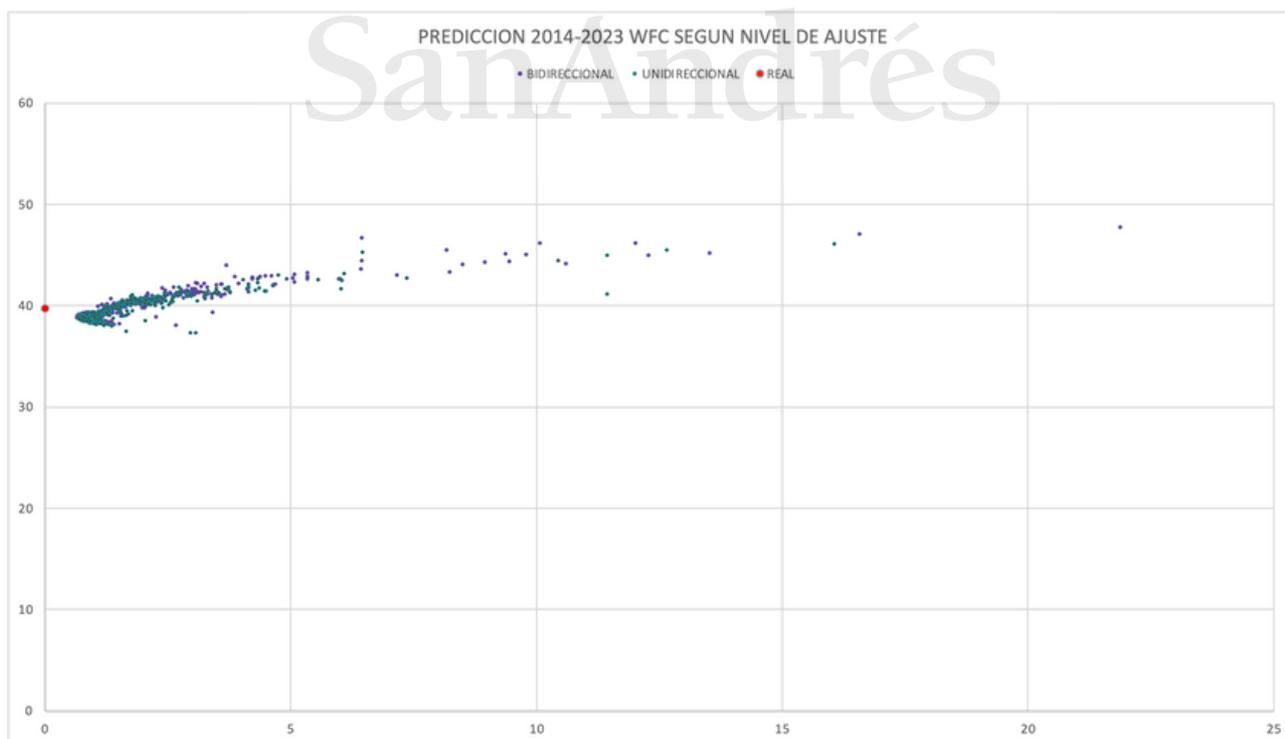


Figura 14: Distribuciones de las predicciones para t+1 según el MSE obtenido

A fin de complementar el análisis previo, se desagregó la serie, (Figura F del Anexo), la cual exhibe homogeneidad en la volatilidad a lo largo de todo el espectro analizado, aunque se destaca un evento disruptivo relacionado a una caída marcada de la cotización, seguida de un aumento de ruido a comienzos de 2020 que luego se estabiliza, y con una reversión permanente a un valor muy cercano a la media de USD 40 (no inmediato).

WFC 2014-2023		
	PRECIO	MSE
Media	39,93	1,98
Error típico	0,06	0,09
Mediana	39,30	1,19
Moda	38,83	0,69
Desviación estándar	1,56	2,59
Varianza de la muestra	2,44	6,73
Curtosis	12,24	83,75
Coefficiente de asimetría	2,59	7,62
Rango	16,11	37,27
Mínimo	37,30	0,65
Máximo	53,41	37,93
Suma	31942,61	1586,80
Cuenta	800,00	800,00
Nivel de confianza(95,0%)	0,11	0,18

Cuadro 6: ESTADISTICA DESCRIPTIVA WFC 2014-2023: PRECIO Y MSE

En lo que respecta a los resultados obtenidos por el modelo LSTM en el periodo 2014-2023, la Figura 14 evidencia que la mayoría de las combinaciones obtienen un nivel de predicción homogéneo, donde la desviación respecto a la etiqueta de $t+1$ en promedio es de USD 0,22, muy inferior a las obtenidas en otros activos. En la Serie de WFC, no hay ni sobre ni subestimación; las predicciones se distribuyen normalmente alrededor de la etiqueta real. En este gráfico se distinguen performances según la dirección de análisis secuencial empleado. Se observa que los modelos Bidireccionales presentan ventajas sobre los Unidireccionales, ya que, independientemente de la arquitectura que adopten, los primeros obtienen similares niveles de ajuste y predicción con menor desvío.

	<i>neuronas</i>	<i>window</i>	<i>epochs</i>
Coefficiente de correlación múltiple	0,0235	0,0676	0,0504
Coefficiente de determinación R ²	0,0006	0,0046	0,0025
R ² ajustado	-0,0007	0,0033	0,0013
Error típico	2,5953	2,5901	2,5928
Observaciones	800,0000	800,0000	800,0000

Cuadro 7: REGRESION ENTRE HIPERPARAMETROS WFC 2014-2023

Resulta interesante que el modelo, al iterar en 300 épocas, logra los mejores ajustes, aunque se incurre en un gasto computacional muy grande. Es importante destacar que, a pesar de esto, el índice de correlación de Pearson indica positividad en la regresión tanto en épocas, neuronas y tamaño de la ventana, lo que es indicador de que la "optimalidad" de los hiperparámetros debería encontrarse en valores inferiores a los presentados en este ejemplo. Sin embargo, estos resultados no son significativos, como se detalla en el Cuadro 7, y no permiten inferir al respecto.

El mejor ajuste, de acuerdo al criterio preestablecido, se obtiene con un modelo Bidireccional que itera en secuencias de 10 observaciones (Ver Cuadro 8). Los datos de entrada con características múltiples obtienen muy buenos resultados, y la medida de dispersión no es significativa respecto al conjunto de control. El resultado de la simulación del modelo óptimo, contrastando con la serie real, se grafica en la Figura 15, en la que se aprecia un ajuste aceptable y homogéneo a pesar de la volatilidad presente.

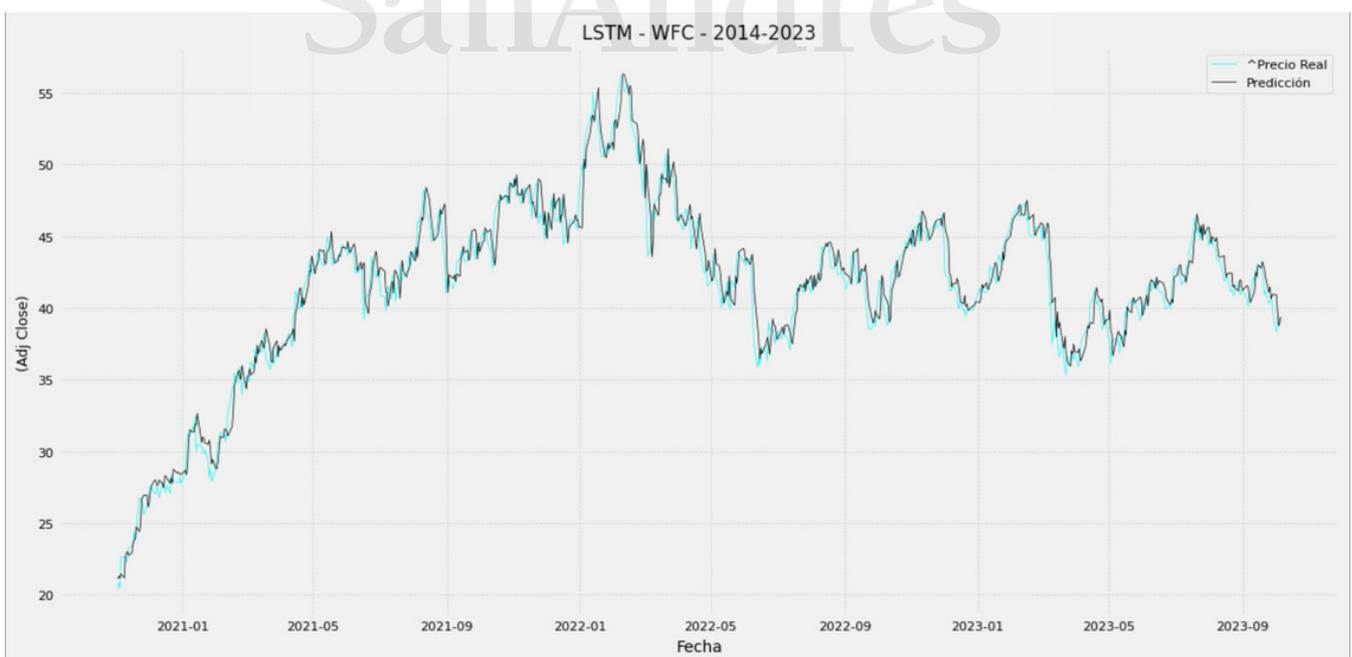


Figura 15: Predicción/REAL: Mejor Arquitectura según criterio MSE; WFC 2014-2023

Entrada	Modelo	Optim.	Neuronas	Window Size	Epochs	Drop out	Batch Size	MSE	Next Day Prediction
MULTIFEATURE	BIDIRECCIONAL	RMSPROP	64	10	300	0	32	0,654	\$ 39,00

Cuadro 8: Mejor combinación WFC 2014-2023

El modelo LSTM muestra mejores resultados en el periodo 2020-2023, al menos en términos generales y en lo que se refiere solo al ajuste. La media del MSE se ubica en 1,81, por debajo de la obtenida en el periodo 2014-2023, aunque con un desvío mayor (ver Cuadro 9). Sin embargo, las predicciones obtenidas en promedio ubican el precio en torno a USD 39,19, sobre USD 39,79 de la etiqueta real, lo que representa una caída importante en el poder predictivo respecto al anterior periodo de análisis.

WFC 2020-2023		
	MSE	PRECIO
Media	1,819	39,163
Error típico	0,088	0,038
Mediana	0,975	38,982
Moda	0,650	38,689
Desviación estándar	2,441	1,062
Varianza de la muestra	5,959	1,128
Curtosis	39,684	2,671
Coficiente de asimetría	5,276	0,643
Rango	29,407	9,461
Mínimo	0,630	34,975
Máximo	30,038	44,436
Suma	1396,685	30077,028
Cuenta	768,000	768,000
Nivel de confianza(95,0%)	0,173	0,075

Cuadro 9: ESTADISTICA DESCRIPTIVA WFC 2020-2023

La Figura 16 muestra la dispersión de las predicciones según el MSE, distinguiendo entre modelos según sea la dirección del análisis. La distribución, forma y el volumen de datos concentrados alrededor de la media, conservan similitud para los periodos bajo estudio. Se

observa en este caso que el modelo Unidireccional obtiene una mejor respuesta, ya que el 85% de los 40 mejores resultados se logra con LSTM de este tipo.

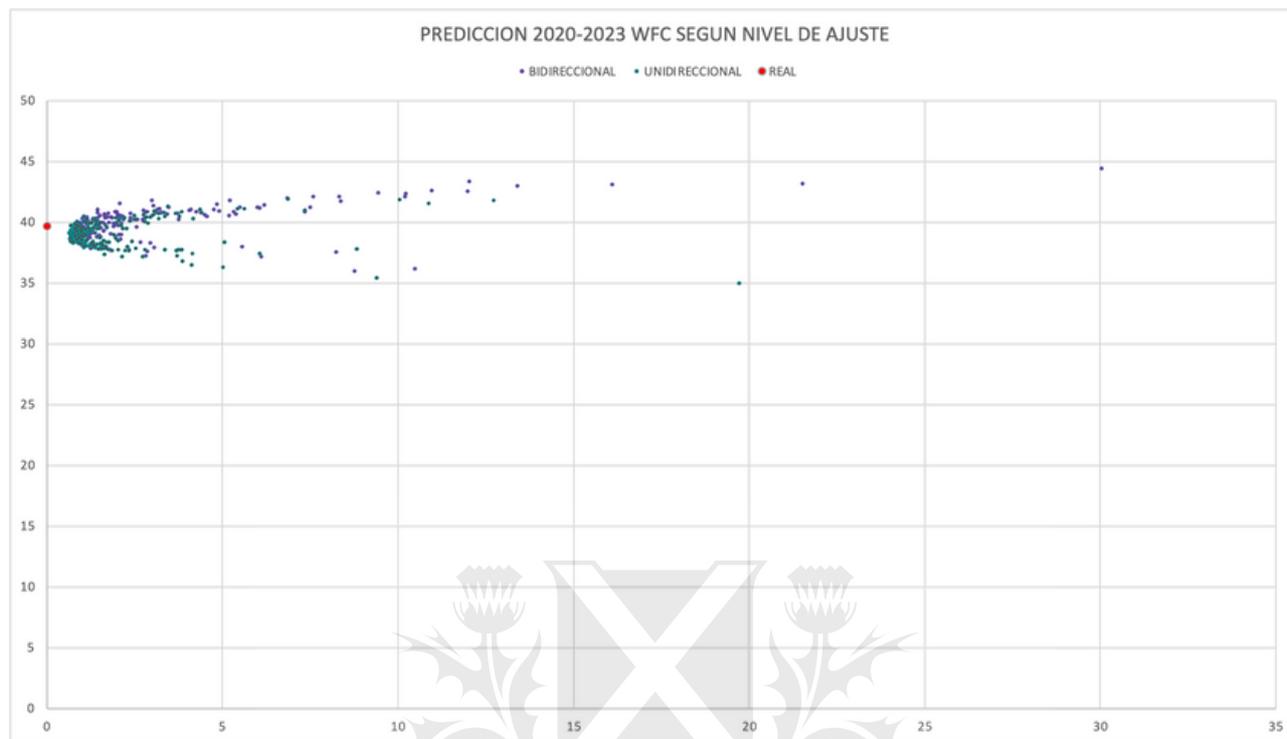


Figura 16: Distribuciones de las predicciones para t+1 según el MSE obtenido

Sin embargo, lo más relevante es que utilizando MULTIFEATURE se obtiene nuevamente un mejor nivel de ajuste, y no solo como promedio general, sino que las estructuras óptimas comparten esta condición. En este sentido, y para este activo en particular, independientemente del periodo bajo análisis, la incorporación de características aporta valor en términos predictivos y se constituye como posibles catalizadores de la dinámica exhibida.

Entrada	Modelo	Optim.	Neuronas	Window Size	Epochs	Drop out	Batch Size	MSE	Next Day Prediction
MULTIFEATURE	UNIDIRECCIONAL	RMSprop	64	40	200	0	32	0,63	\$ 39,09

Cuadro 10: combinación que destaca por se la mejor predicadora de MRVL 2014-2023

La arquitectura óptima se exhibe en el Cuadro 10, donde se destacan diferencias respecto al periodo anterior, no solo en la dirección del análisis secuencial, sino también en épocas y tamaño de las ventanas. Tal como en el activo anterior, MRVL, se observa una relación inversa entre la longitud del periodo analizado y el tamaño de las secuencias empleadas, a mayor longitud, menor tamaño de la ventana deslizante.

La Figura 17 muestra el proceso de ajuste del modelo a las etiquetas reales. Si bien la validación en este caso se hace con un volumen considerablemente menor de datos, dado el tamaño de la muestra, hay homogeneidad en todo el segmento analizado. Es evidente que la predicción sobrevuela la etiqueta real en todo el periodo, no terminando de adherir al sendero, sobreestimando de esta manera el precio en $t+1$.

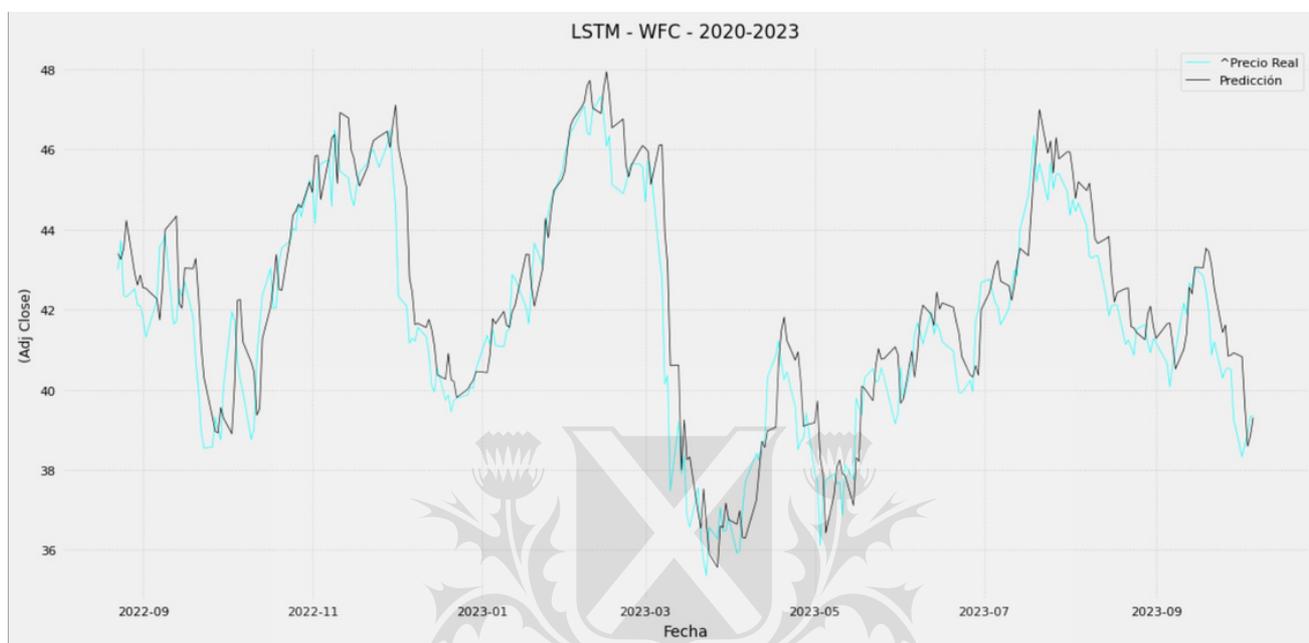


Figura 17: Predicción/REAL: Mejor Arquitectura según criterio MSE, WFC 2020-2023

Dentro de los hiperparámetros, se observa una correlación negativa entre el número de neuronas y el MSE. En general, el modelo con 96 neuronas obtiene un menor error. En la misma línea, aunque el óptimo se alcanza con una ventana deslizante de 40 observaciones, los resultados promedio indican que la ventana de 30 es más efectiva. El hecho de que esta información no sea relevante en un barrido de hiperparámetros podría ser una señal de que el modelo está sobreajustando.

MELI (MERCADO LIBRE)

La gigante tecnológica argentina creada por Marcos Galperin a finales del siglo XX experimentó un crecimiento muy significativo durante el periodo bajo estudio, especialmente hasta finales de 2021, cuando alcanzó una cotización de USD 2.000. Sin embargo, como resultado del ajuste de balances de la FED en 2020, el precio se deprimió hasta mediados de 2022, a partir de donde comenzó a mostrar signos de recuperación. La descomposición realizada en la Figura E destaca un aumento en el ruido, coincidente con el inicio de la pandemia, que se ha mantenido hasta la actualidad, sin mostrar un patrón estacional claro. Es evidente que la serie no es estacionaria para ninguno de los periodos de análisis, como se muestra en el Cuadro A.

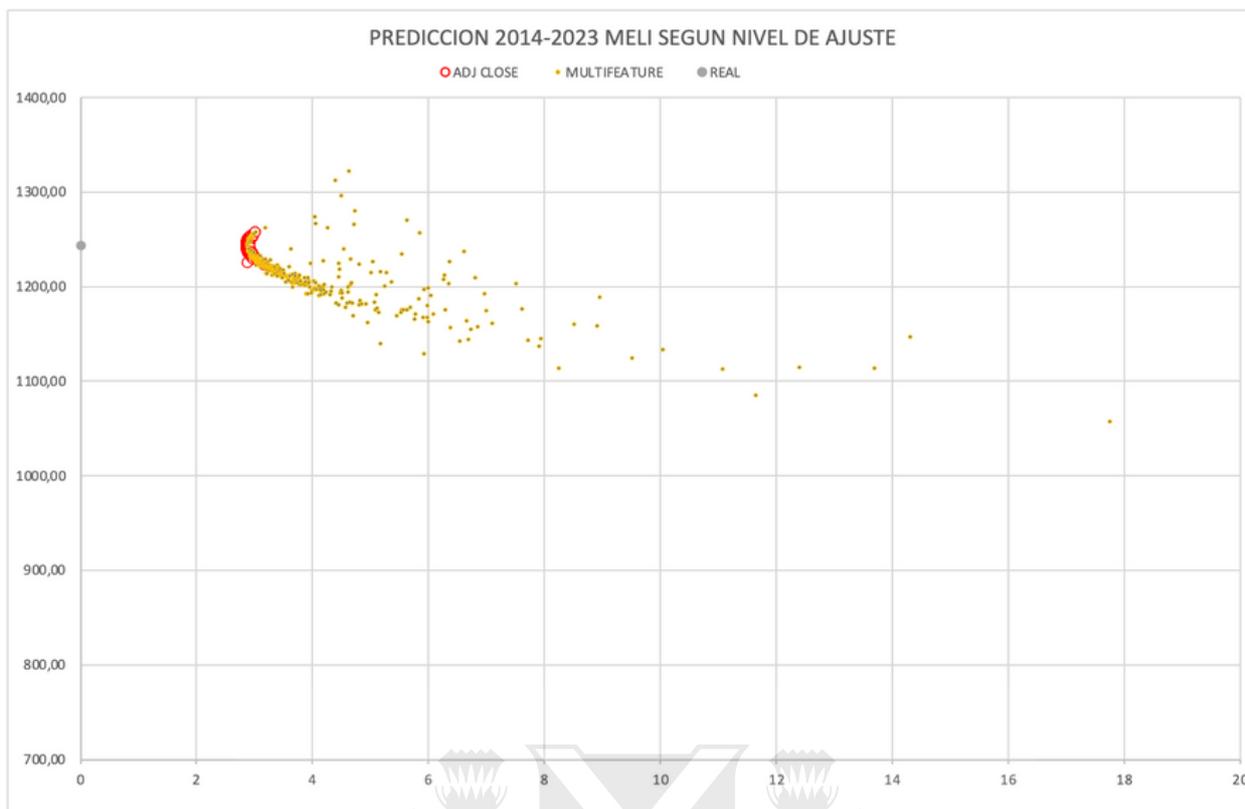


Figura 17: Distribuciones de las predicciones para $t+1$ según el MSE obtenido

Para el periodo 2014-2023, la Figura 17 proporciona evidencia gráfica de la mayor dispersión de los errores cuadráticos medios (MSE) al utilizar características múltiples como entrada al modelo. Este resultado es similar al observado en MRVL, con la diferencia de que ni siquiera existen estructuras particulares que muestren un mejor rendimiento, como sucedió en el caso de la empresa estadounidense en la serie 2020-2023. Aquí hay una evidencia muy fuerte de que cuando tomamos como entrada al modelo características múltiples, el error típico de la muestra es elevado en comparación con los resultados obtenidos al emplear solo el precio ajustado. Esto claramente podría ser indicador de sobreajuste.

En términos de predicción, el modelo tiende a subestimar la etiqueta real, ya que el 90% de las combinaciones arrojan un valor por debajo del valor objetivo en $t+1$. Esto es fácilmente apreciable en la Figura 17, donde la mayoría de las observaciones están por debajo de la línea horizontal que representa este valor.

En promedio, las distintas arquitecturas introducidas en el análisis predicen con un grado de acierto aceptable, mostrando un desvío promedio de solo el 1,2% respecto de la etiqueta real. En el caso específico de MELI, si consideramos que el precio el 11 de octubre fue de USD 1243,95, el modelo pronosticó en promedio USD 1228,55. Sin embargo, si solo consideramos las arquitecturas que emplean como dato de entrada el precio ajustado de cierre, la capacidad predictiva del modelo aumenta, prediciendo un valor de USD 1243,84.

La mejor combinación de ajuste, como se detalla en el Cuadro 11, coincide con lo estipulado en párrafos anteriores. Aunque no es posible establecer una regla general, se observa que del total de arquitecturas, las 30 que obtienen menor MSE optimizan con RMSprop. Además,

en promedio, una ventana de 20 observaciones parece tener mejores resultados, manteniendo constante el resto de los parámetros y con iteraciones de 300 épocas. La correlación de estos hiperparámetros con el MSE es negativa en ambos casos. Es notable también que los modelos Bidireccionales muestran un mejor ajuste en general.

Entrada	Modelo	Optim.	Neuronas	Window Size	Epochs	Drop out	Batch Size	MSE	Next Day Predictio
ADJ CLOSE	BIDIRECCIONAL	RMSprop	32	20	300	0	32	2,85	n \$ 1244,46

Cuadro 11: combinación que destaca por se la mejor predicadora de MELI 2014-2023

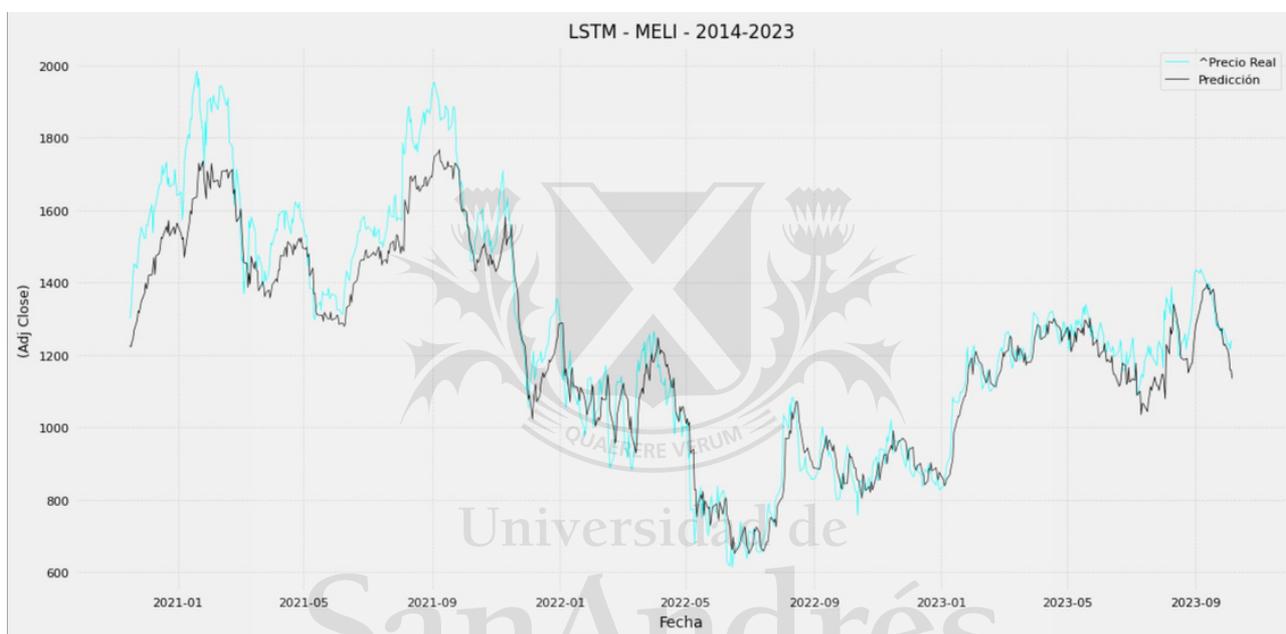


Figura18: Predicción/REAL: Mejor Arquitectura según criterio MSE; MELI 2014-2023

La Figura 18 presenta los senderos de precios obtenidos por el modelo óptimo y los contrasta con la serie real durante el periodo de validación, que corresponde al 10% de la secuencia total. Se observa que la predicción inicial no está alineada con la serie original, pero luego comienza a converger. Esto podría ser consecuencia directa de que los datos de entrenamiento no incluyen la volatilidad experimentada por los datos de prueba y validación. Se podría hacer una analogía con un deportista que se entrena a nivel del mar para correr en la altura.

Para el periodo 2020-2023, al igual que en otros activos, nos enfrentamos a un proceso en el que el modelo se entrena con menos información pero con una volatilidad persistente y homogénea. A pesar de esto, el modelo responde bien en términos de optimización mediante MSE, mostrando una capacidad de ajuste aceptable en la mayoría de las arquitecturas planteadas. Sin embargo, al igual que en el periodo anterior, se observa una caída significativa en el ajuste cuando se utiliza MULTIFEATURE como entrada. El uso del Precio Ajustado de Cierre

logra reducir a la mitad el MSE promedio y acercar la predicción a la etiqueta real, aproximándola de USD 1231 a USD 1243,37. La Figura 19 resume gráficamente estos resultados, mostrando claramente la diferencia en términos de desviación estándar al utilizar uno u otro conjunto de datos de entrada. Estos resultados confirman la importancia crucial de la elección de las características, que requiere un estudio detallado y paciente del activo y los posibles impulsores de las cotizaciones.

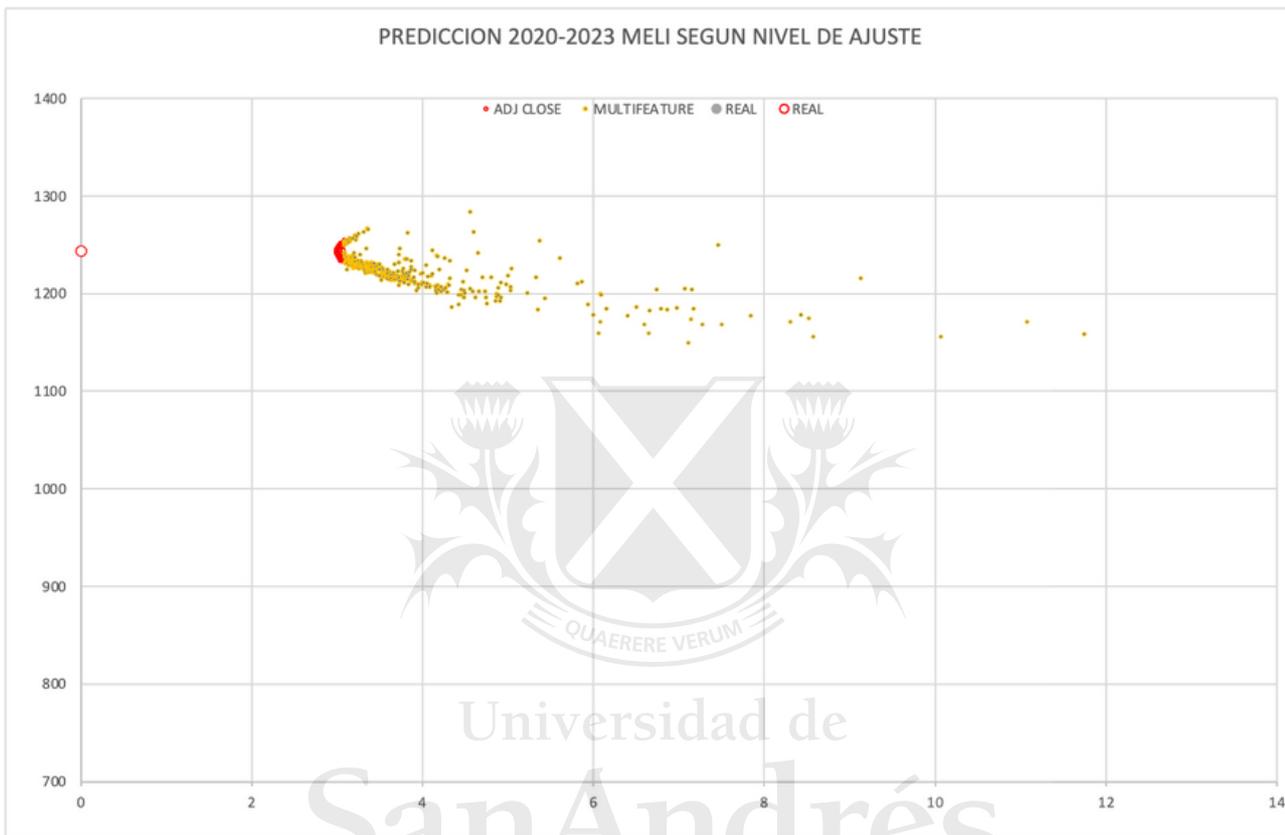


Figura 19: Distribuciones de las predicciones para t+1 según el MSE obtenido

El grid search en este periodo devuelve una estructura óptima (ver Cuadro 12) muy similar a la obtenida en el periodo anterior, excepto por la dirección en la que corre la secuencia. La gráfica de los resultados de esta arquitectura LSTM, que se muestra en la Figura 20, revela una mayor homogeneidad en el ajuste, siendo aceptable incluso en los tramos de mayor volatilidad. Sin embargo, se observa un salto discreto importante a comienzos de 2023 que podría haber provocado un desacople mayor entre la predicción y la serie real.

Entrada	Modelo	Optim.	Neuronas	Window Size	Epochs	Drop out	Batch Size	MSE	Next Day Prediction
ADJ CLOSE	UNIDIRECCIONAL	RMSprop	32	10	300	0	32	2,99	\$ 1243,30

Cuadro 12: combinación que destaca por se la mejor predicadora de MELI 2020-2023



Figura 20: Predicción/REAL: Mejor Arquitectura según criterio MSE; MELI 2020-2023

LNTH (LANTHEUS HOLDINGS INC.)

Para el caso de LNTH, se planteó una perspectiva analítica distinta respecto a los demás activos analizados previamente. En cuanto a los datos de entrada, se emplearon tres grupos de control, sumando al anterior un conjunto integrado por los precios ajustados y el índice \wedge VIX. La decisión en este sentido estuvo orientada a tratar de aportar luz a la divergencia de resultados ligados a la variable "características de entrada" y su impacto en el funcionamiento del modelo. Otra modificación introducida fue respecto a los periodos bajo análisis: LNTH cotiza en bolsa desde 2015, por lo que obligadamente se empleó el periodo 2015-2023.

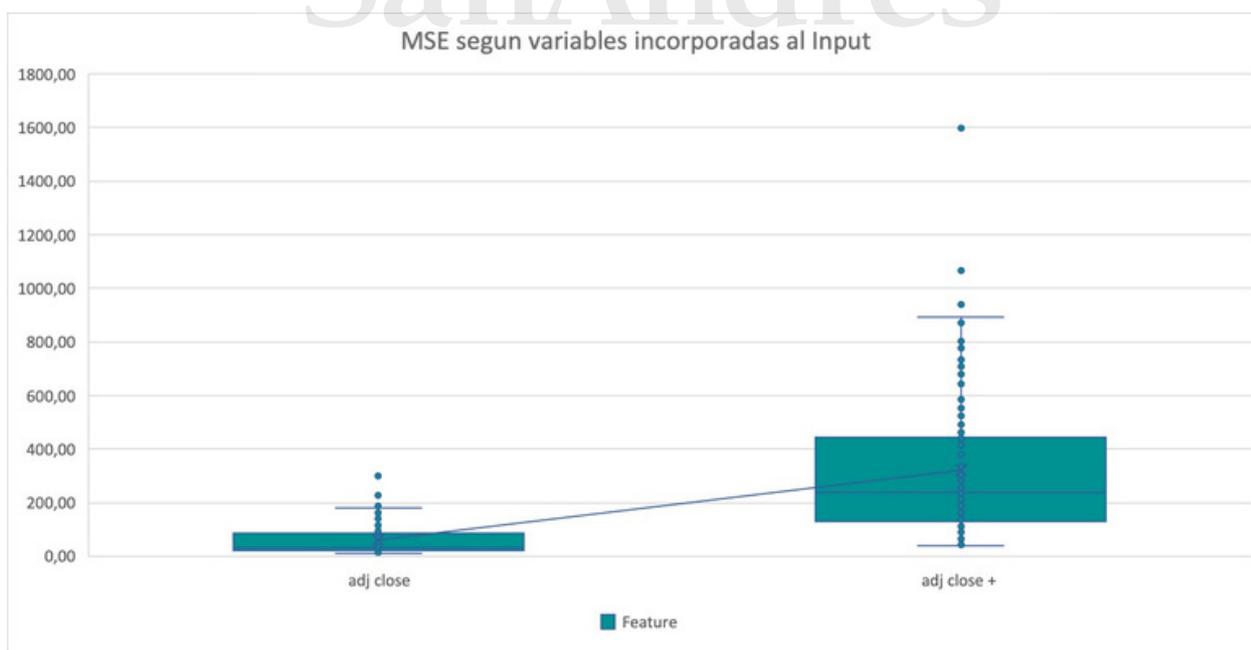


Figura 21: Gráfica de caja : Niveles de MSE según datos de entrada

La incorporación de un tercer grupo de control a los datos de entrada se realizó de manera probatoria ante los resultados que se iban obteniendo en las simulaciones en tiempo real. La Figura 21 es una prueba contundente de este hecho, ya que se constata que cuando el modelo es provisto solo con datos de Precio Ajustado, los MSE son considerablemente más bajos en comparación con los obtenidos con MULTIFEATURE. Existe un dato aún más revelador en este sentido: del total de arquitecturas posibles, las 156 más eficientes en términos de MSE usan solo el Precio ajustado de cierre como entrada al modelo. Hasta el momento, en el análisis de los activos incluidos en este trabajo, hubo casos en donde se pudo destacar situaciones similares, sin embargo, no tan contundentes como para elaborar una premisa. En el análisis de LNTH, hay argumentos sólidos para rechazar por completo la incorporación de "características" a los datos de entrada. Al incorporar el tercer grupo de control, los resultados de MSE se acercan drásticamente a los obtenidos solo con los Precios de cierre Ajustados; sin embargo, la desviación de las observaciones es más elevada. Esto último nos podría llevar a inferir que, si bien el agregado de características no mejora los errores de predicción, podría hacer que el modelo entrenado no sea tan sensible a los parámetros de ajuste. Claramente, la elección de estas características, nuevamente, es la cuestión determinante.

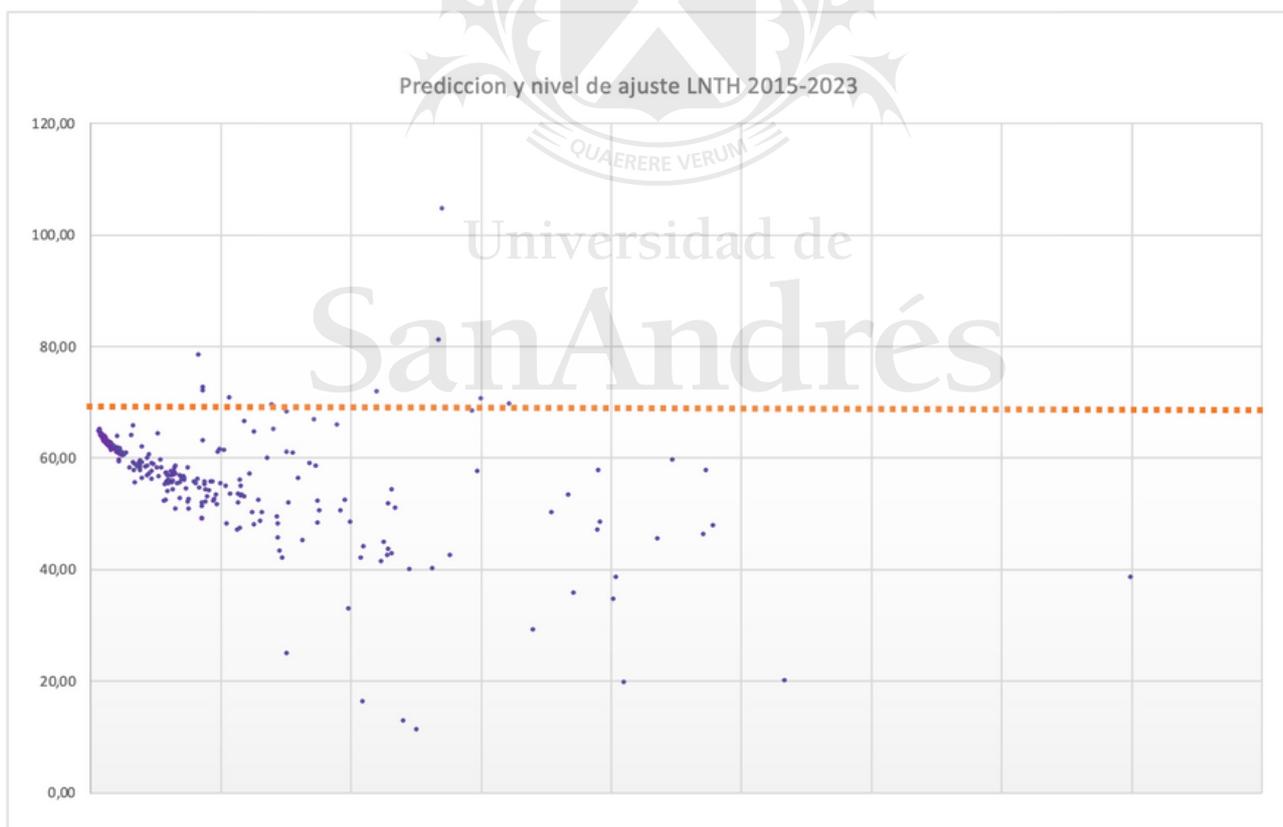


Figura 28: El modelo tiende a subestimar las predicciones, independientemente del nivel de ajuste.

En la FIGURA H del Anexo, descomponemos la serie según tendencia, estacionalidad y ruido. Se observa que a partir de febrero de 2022, la serie gana volatilidad, lo que puede representar un problema en términos de eficiencia. El modelo está programado de manera que se entrene con el 80% de los datos de la serie, lo que significa que la mayor parte del entrenamiento se realiza con datos que no guardan relación con los valores actuales, al menos en términos nominales y de volatilidad. A pesar de esta limitación, el nivel de ajuste que se expone es muy aceptable, con la salvedad de que luego del evento disruptivo de febrero de 2022, la predicción pierde homogeneidad y se desacopla de la serie real.

Entrada	Modelo	Optim.	Neuronas	Window Size	Epochs	Drop out	Batch Size	MSE	Next Day Predictio
ADJ CLOSE	BIDIRECCIONAL	Adam	64	20	200	0	32	12,16	\$ 64,99

Cuadro 13: combinación que destaca por se la mejor predicadora de LNTH 2015-2023

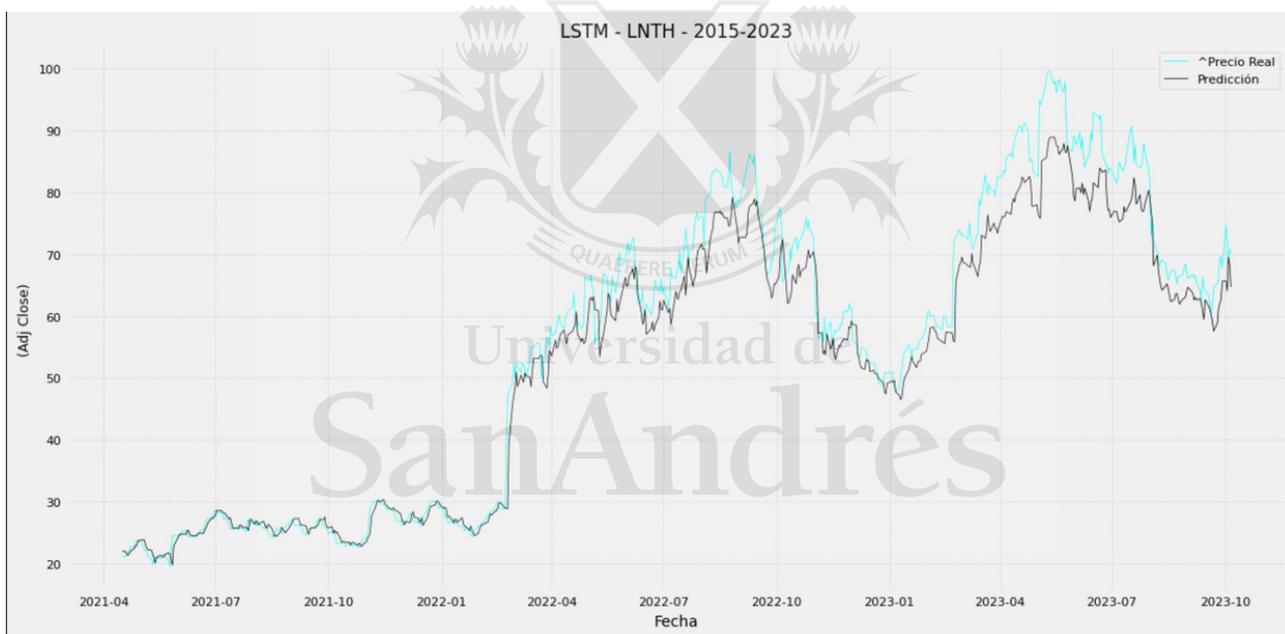


Figura 20: Predicción/REAL: Mejor Arquitectura según criterio MSE; LNTH 2015-2023

Del análisis del total de resultados proporcionados por el grid search, se observa que los modelos Bidireccionales optimizados por Adam obtienen mejores resultados. No hay nada relevante en términos de los demás hiperparámetros que se pueda destacar en este caso, ya que no guardan correlación con el MSE. Las predicciones del precio en t+1, como muestra la Figura 23, en su gran mayoría (98%) tienden a subestimar la etiqueta real. Al realizar un análisis gráfico, similar al planteado en la Figura 20, y hacerlo extensivo a la totalidad de las arquitecturas analizadas, se observa que uno de los factores de la subestimación del modelo puede deberse a que este no logra captar el salto discreto de febrero de 2022. En todos los gráficos se observa que

si bien la predicción ajusta, siempre queda por debajo de la etiqueta real y no logra acoplarse al trayecto de las cotizaciones.

7.-CONCLUSIONES

Como resultado de un intenso análisis, que incluye la evaluación de diferentes configuraciones del modelo LSTM, aplicado a diferentes activos y periodos, se puede concluir que los modelos LSTM pueden llegar a ser un herramienta compleja, que puede bajo una serie de especificaciones, captar dependencias no lineales en series de tiempo financieras.

De los propósitos planteados inicialmente para esta investigación, se destaca la de evaluar la calidad predictiva de los modelos LSTM, y en este sentido, los resultados obtenidos para las distintas arquitecturas propuestas, pueden convalidar el enorme potencial de este tipo de RNN's en el análisis de series de tiempo. El nivel de ajuste en algunos casos es mas que satisfactorio en concordancia con la evidencia bibliográfica, la celda de estado y el proceso de optimización de LSTM muestran fortaleza, sin embargo el costo de modelado y de simulación en términos computacionales y de tiempo, no logran compensar su uso por sobre modelos tradicionales por el momento.

Luego del análisis de los resultados, donde se pudieron verificar las limitaciones que tiene LSTM concordantes al consenso académico, como los enormes costos de memoria, dificultad para plasmar eventos exógenos en los datos de entrada, encuentro que su mayor problema esta ligado a la poca capacidad para manejar situaciones de volatilidad extrema.

El rezago que introduce el modelo a los valores predichos, podría ser consistente con el supuesto que señalan Tannuré & Gerez (2023), donde concluyen que LSTM tiende a eliminar la información de largo plazo, para predecir solo con el dato del día previo. La información y los resultados obtenidos en esta investigación, sin embargo, no son suficiente evidencia para afirmarlo. Los hallazgos presentados en este estudio resaltan una mayor confianza en cuanto a la capacidad de LSTM.

Como conclusiones importantes de este estudio, se ha logrado identificar un conjunto de hiperparámetros en los que el error se “clusteriza”, es decir que se observan resultados agrupados en las distintas distribuciones diagramadas, lo que evidencia la baja sensibilidad de ajuste del modelo en este conjunto específico. Se observa que hay un número significativo de resultados para los cuales el modelo exhibe estabilidad. Todas las distribuciones que se presentaron en este trabajo guardan extrema similitud, describen una parábola, un símil boomerang achatado que presenta una densidad de puntos considerable alrededor del precio objetivo, y que son indicador del potencial del modelo LSTM, puesto que demuestra que la distribución de errores no depende del tiempo y el activo bajo análisis.

Es crucial destacar que la amplia divergencia de resultados obtenidos al incorporar diferentes características al modelo, subraya la complejidad en la elección de estas variables.

Este hecho resalta la importancia de identificar catalizadores o drivers de riesgo, lo que representa verdaderamente el desafío central en este campo de estudio.

Por último, este trabajo no solo proporciona conclusiones significativas, sino que también plantea nuevos retos para la investigación en el uso de modelos LSTM para análisis de series temporales financieras.

8.-ANEXO

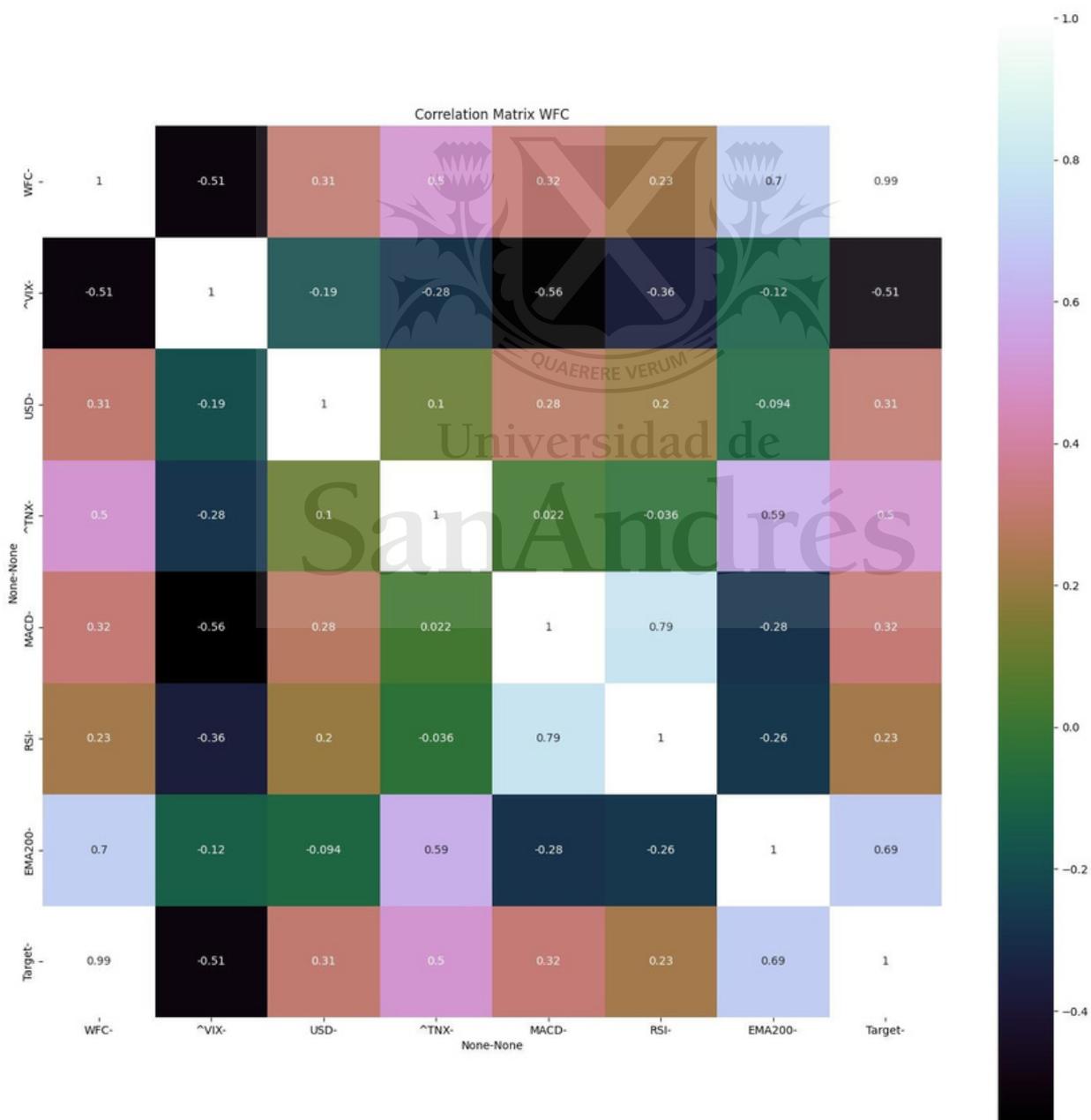


Figura A: Matriz correlación de WFC serie 2014-2023

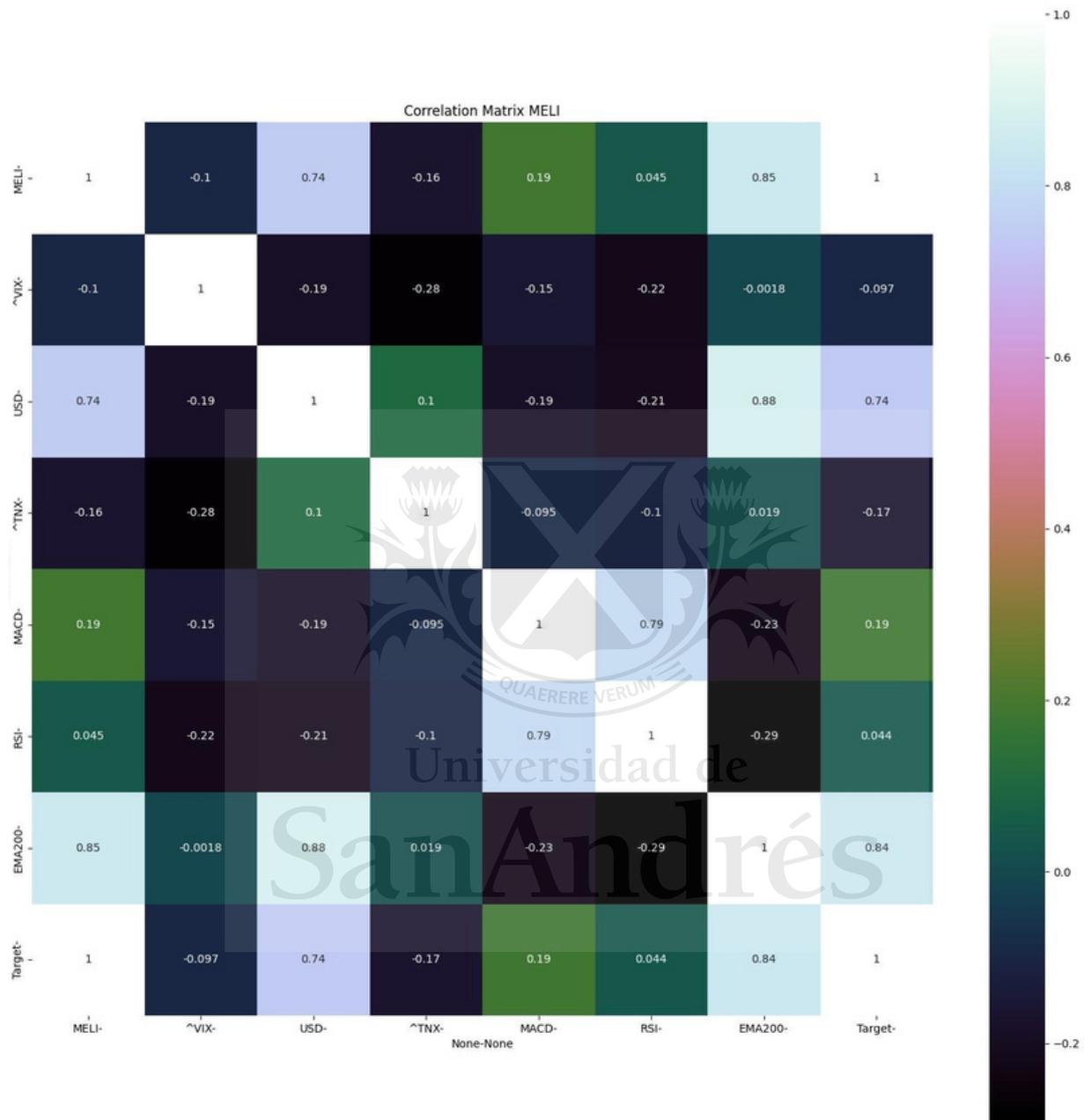


Figura B: Matriz correlación de MELI serie 2014-2023

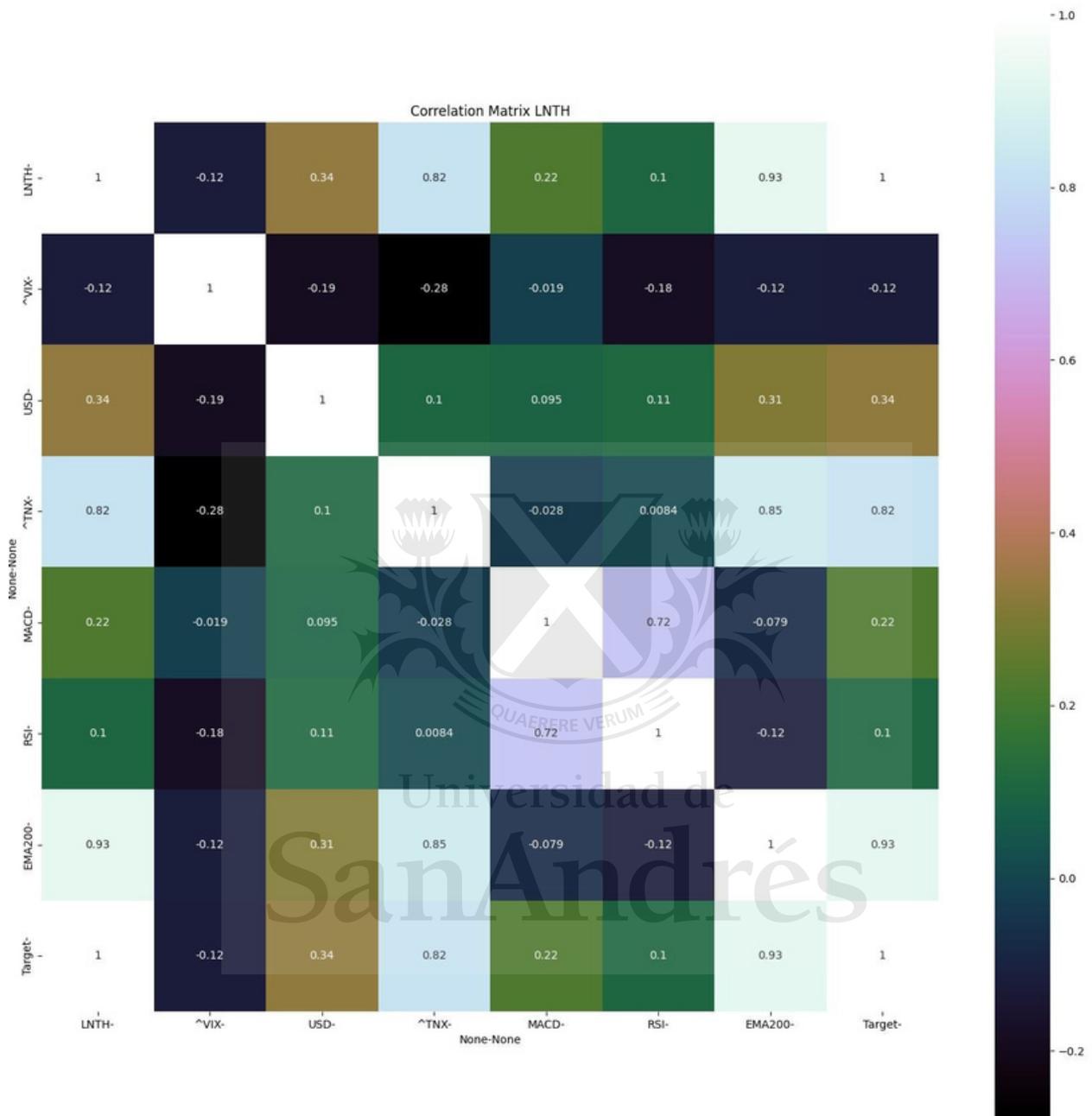


Figura C: Matriz correlación de LNTH serie 2014-2023

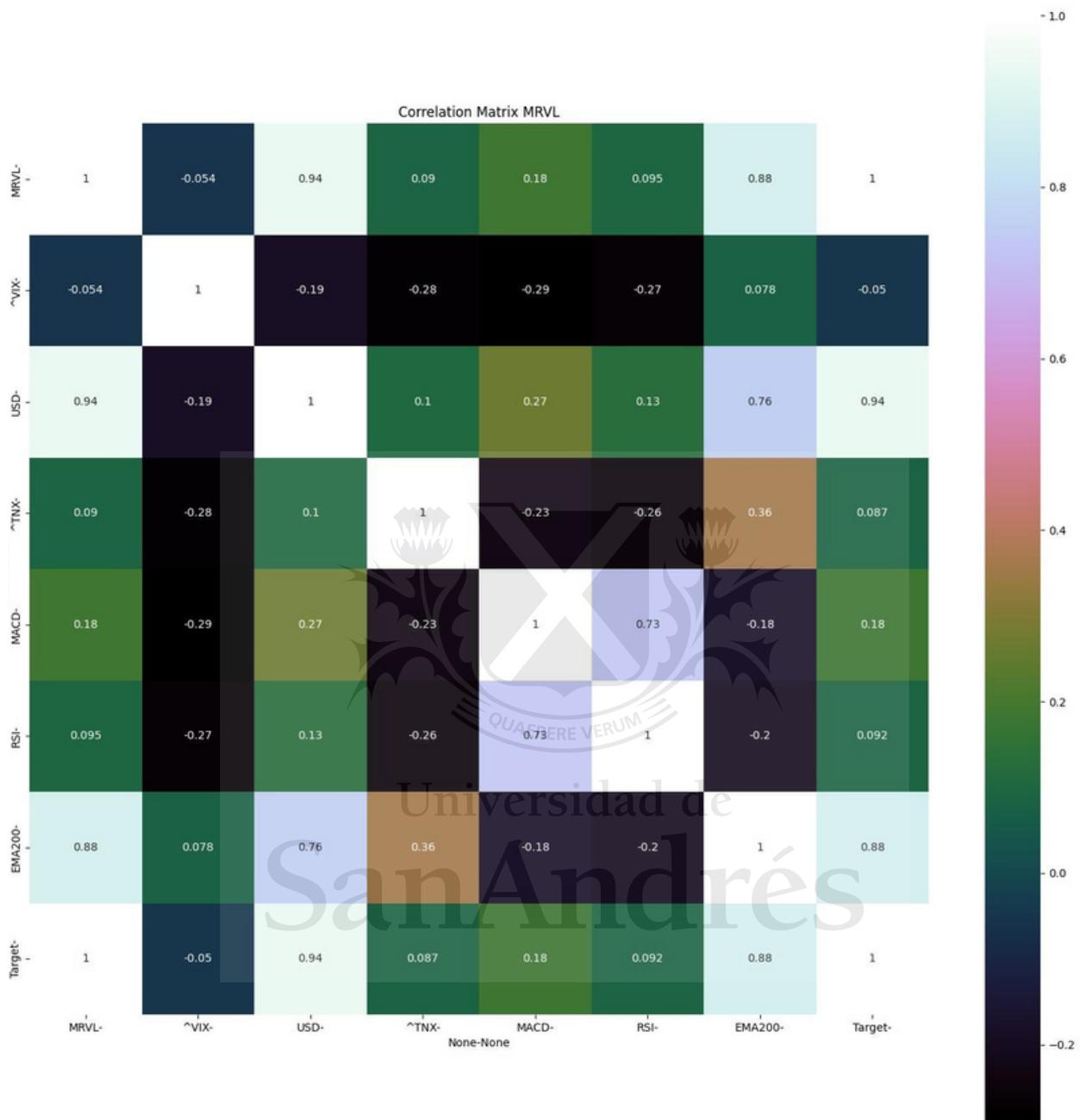


Figura D: Matriz correlación de MRVL serie 2014-2023

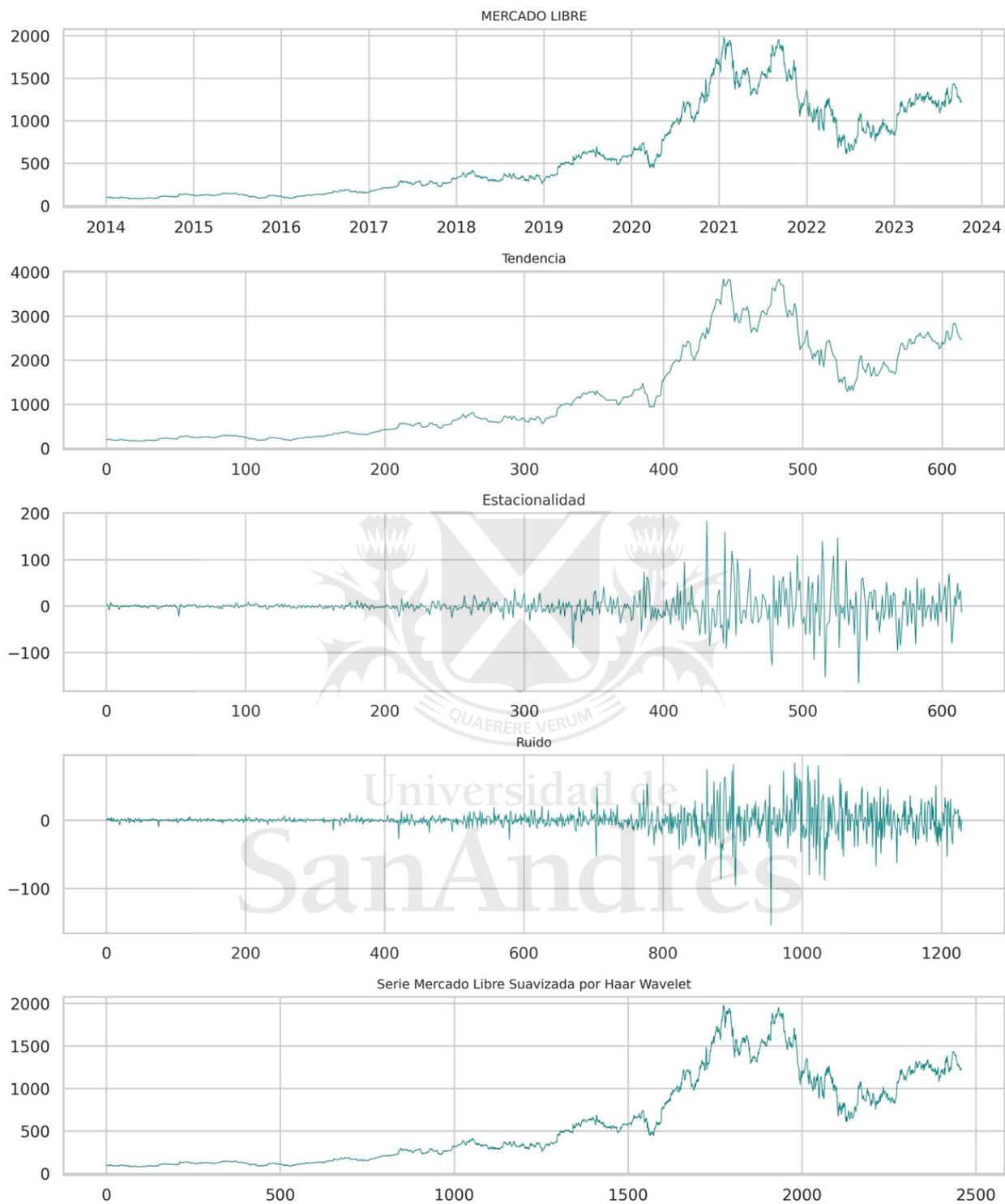


Figura E: descomposición de la serie de Mercado Libre

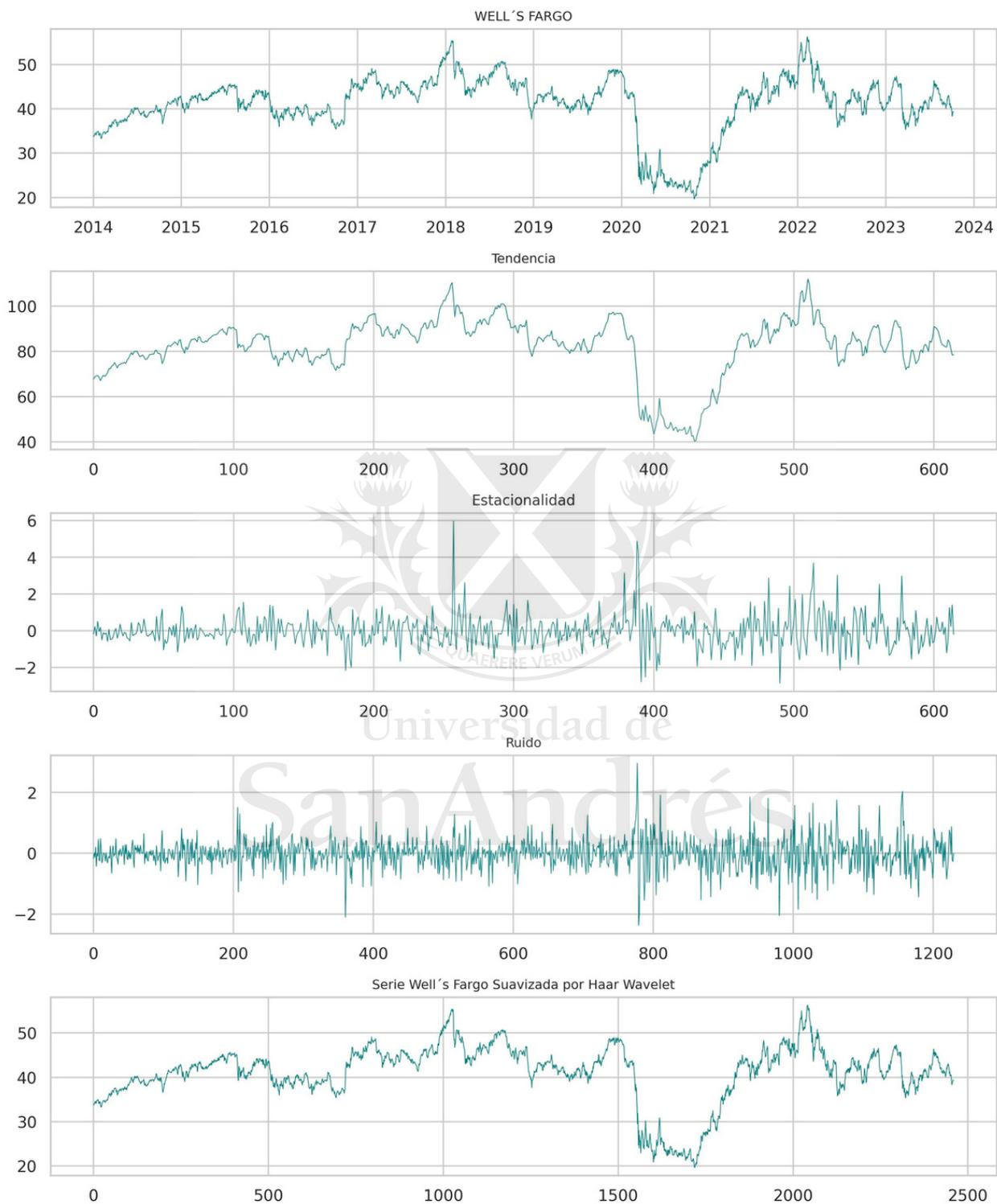


Figura F: descomposición de la serie de WELL'S FARGO

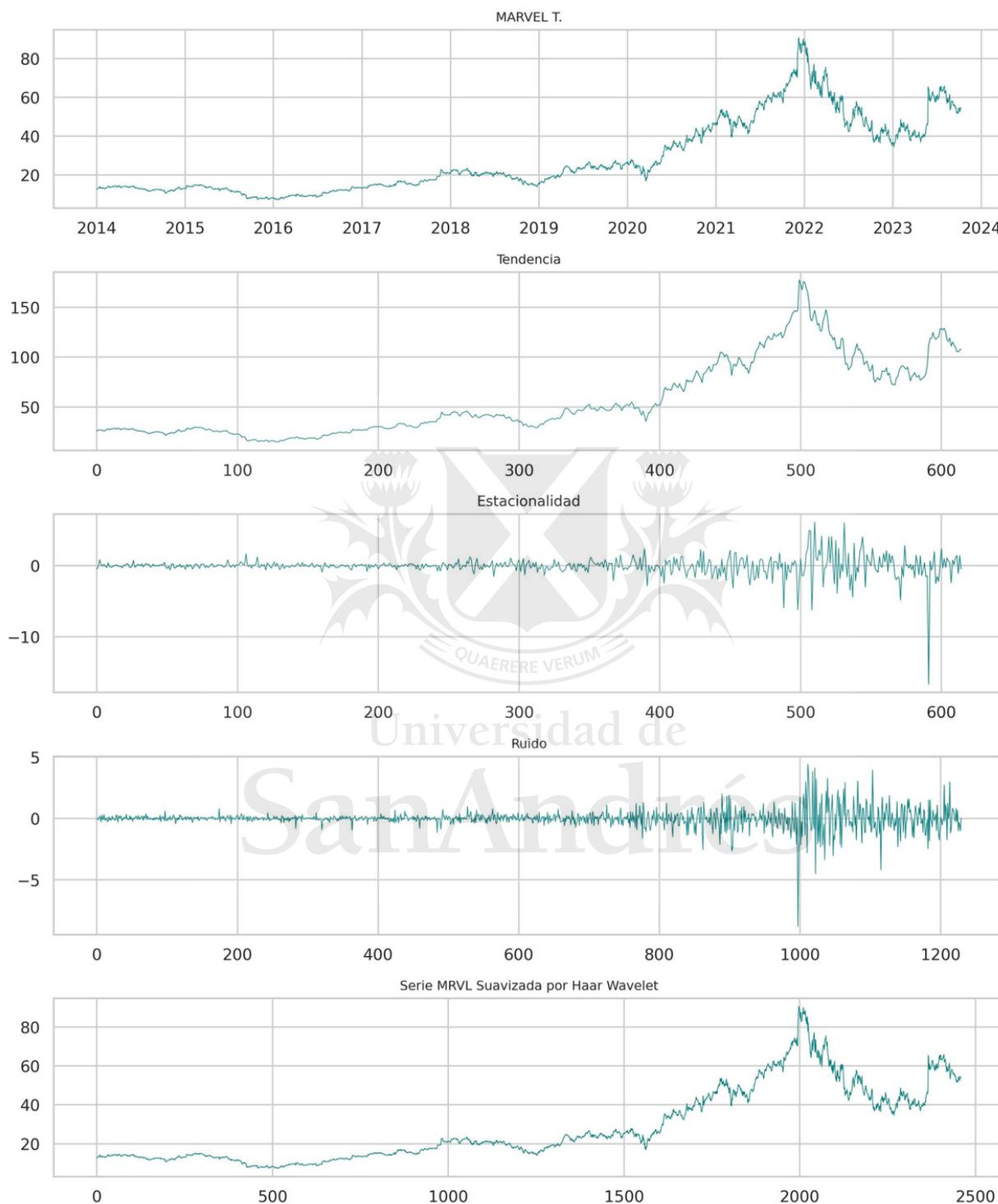


Figura G: descomposición de la serie de MARVELL TECHNO.

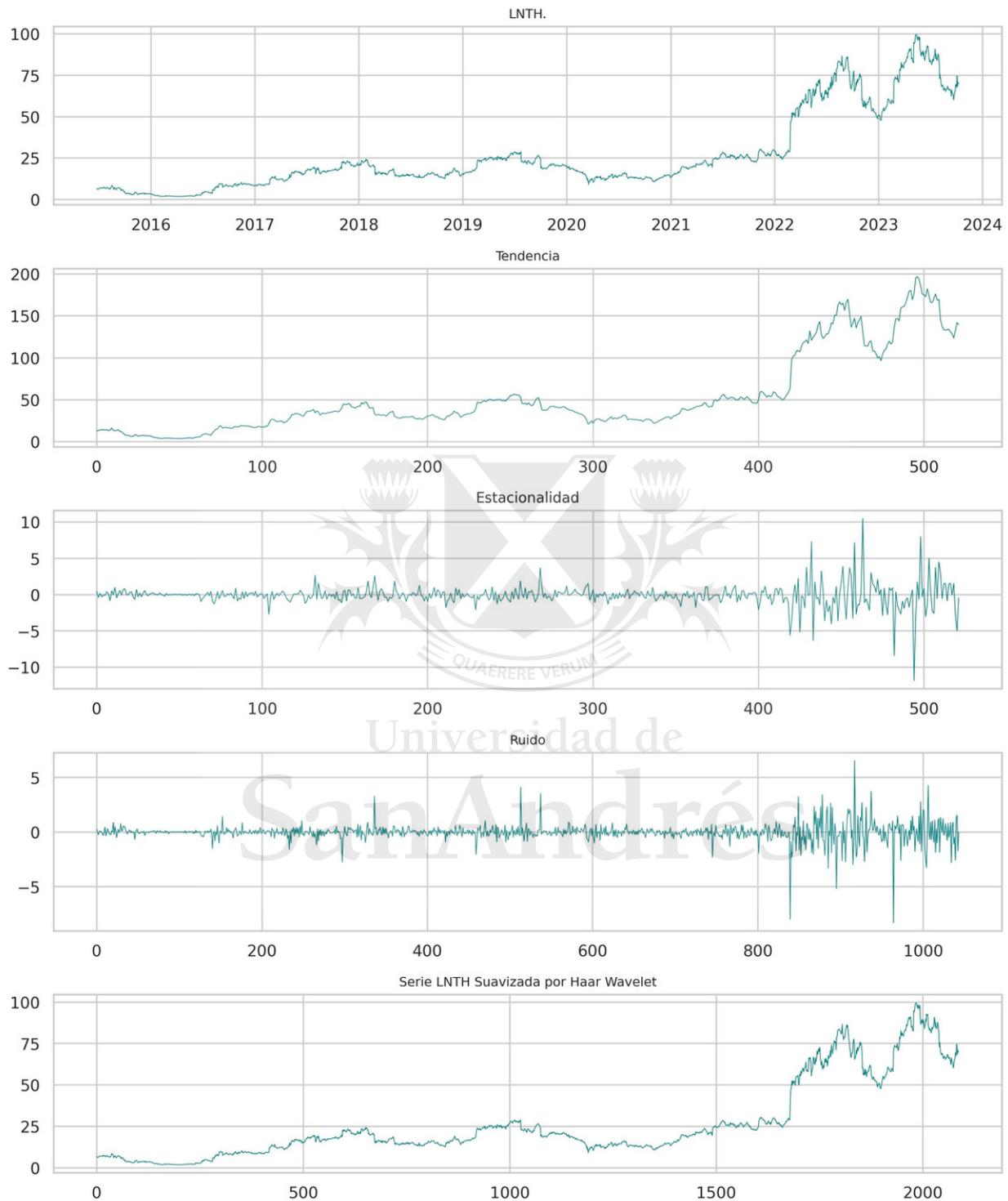


Figura H: descomposición de la serie de LANTHEUS HOLDINGS



FIGURA I: predicción contra serie real usando rendimientos como input MELI

Activo	Periodo	Estadístico	Valor_p	Estacionaria
MRVL	2014-2023	-0,92	0,78	False
MRVL	2020-2023	-2,12	0,24	False
WFC	2014-2023	-2,27	0,18	False
WFC	2020-2023	-2,02	0,28	False
LNTH	2014-2023	-0,59	0,87	False
LNTH	2020-2023	-0,74	0,84	False
MELI	2014-2023	-1,09	0,72	False
MELI	2020-2023	-2,13	0,23	False

CUADRO A: TEST DF ESTACIONARIEDAD PRECIOS Y RETORNOS

Activ	Periodo	Media	Std	Min	25 %	50 %	75 %	Max
	2014-2023	29,04	18,63	7,77	14,89	21,74	42,83	91,51
MRVL	2020-2023	49,03	14,53	17,25	39,48	47,03	59,15	91,51
MRVL	2014-2023	47,67	8,54	21,14	44,94	49,02	53,75	65,93
WFC	2020-2023	40,47	8,86	21,14	35,08	42,74	46,68	59,06
WFC	2014-2023	26,21	23,33	1,83	12,80	17,85	26,42	99,65
LNTH	2020-2023	41,13	26,90	9,11	16,57	26,97	66,16	99,65
LNTH	2014-2023	603,31	513,76	80,53	146,2	358,20	1024,28	1984,34
MELI	2020-2023	1170,26	348,44	447,34	0	1181,69	1392,75	1984,34

MELI

CUADRO B: Estadística Descriptiva de las Series Originales por pedido

897,6
2

9.-CODIGO EMPLEADO PYTHON

```

pip install yfinance
pip install plotly
pip install mplfinance
pip install ta

import math
import numpy as np
import pandas_datareader as web
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from keras.models import Sequential
from keras.layers import LSTM, Dense
import mplfinance as mpf
import ta
from keras.layers import Dropout
from keras.layers import Bidirectional
from keras.optimizers import RMSprop, Adam

plt.style.use('fivethirtyeight')

import yfinance as yf

data = yf.download(['MELI', 'USD', '^TNX', '^VIX'], start= '2014-01-01', end = '2023-10-10')

# Calcular los retornos diarios
data['MELI'] = data['Adj Close']['MELI']#.pct_change()

# Calcular el RSI
rsi = ta.momentum.RSIIndicator(data['Adj Close']['MELI'])
data['RSI'] = rsi.rsi()

# Calcular el MACD
macd = ta.trend.MACD(data['Adj Close']['MELI'])
data['MACD'] = macd.macd()

# Calcular el objetivo
data['Target'] = data['MELI'].shift(-1)

#demás columnas
data['^VIX'] = data['Adj Close']['^VIX']
data['USD'] = data['Adj Close']['USD']
data['^TNX'] = data['Adj Close']['^TNX']

features = ["MELI"]#, '^VIX', 'USD', '^TNX', 'MACD', 'RSI'
target = "Target"
data = data[features + [target]].dropna()
data

# Escalar los datos
scaler = MinMaxScaler()
data[features] = scaler.fit_transform(data[features])
data[target] = scaler.fit_transform(data[[target]])

import itertools
import matplotlib.pyplot as plt
import matplotlib.dates as mdates

# Definir listas de valores para hiperparámetros
data_splits = [0.8]
validation_split = [0.1]

```

```

lstm_units = [0]
second_lstm_units = [16, 32, 64, 96]
window_sizes = [10, 20, 30, 40]
epochs_values = [100, 200, 300]
Drop_out = [0, 0.1]
batch_sizes = [32]

# Crear una lista de todas las combinaciones posibles de hiperparámetros
param_combinations = list(itertools.product(data_splits, validation_split, lstm_units, second_lstm_units, window_sizes
, Drop_out, epochs_values, batch_sizes))

# Crear una lista para almacenar los resultados
results = []

# Iterar sobre todas las combinaciones de hiperparámetros
for params in param_combinations:
    data_splits, validation_split, lstm_unit, second_lstm_units, window_size, Drop_out, epochs, batch_size = params

# Crear el sliding window
X, y = [], []

for i in range(window_size, len(data)):
    X.append(data.iloc[i - window_size:i][features].values)
    y.append(data.iloc[i]['Target'])

X, y = np.array(X), np.array(y)

#split = int(data_split * len(X))
#X_train, X_test = X[:split], X[split:]
#y_train, y_test = y[:split], y[split:]

# Dividir los datos en conjuntos de entrenamiento, validación y prueba
split_train = int(len(X) * data_splits) # 80% para entrenamiento
split_valid = int(len(X) * validation_split) # 10% para validación
split_test = len(X) - split_train - split_valid # El resto para prueba

X_train, y_train = X[:split_train], y[:split_train]
X_valid, y_valid = X[split_train:split_train + split_valid], y[split_train:split_train + split_valid]
X_test, y_test = X[split_train + split_valid:], y[split_train + split_valid:]

# Crear y compilar el modelo LSTM
model = Sequential()
#model.add(LSTM(lstm_unit, return_sequences=True, input_shape=(window_size, len(features))))
#model.add(Bidirectional(LSTM(lstm_unit, return_sequences=True), input_shape=(window_size, len(features))))
model.add(Dropout(Drop_out))
model.add(Bidirectional(LSTM(second_lstm_units, return_sequences=False)))
#model.add(LSTM(second_lstm_units, return_sequences=False))
model.add(Dropout(Drop_out))

model.add(Dense(1))
model.compile(optimizer='adam', loss='mean_squared_error')

# Define una función para calcular RMSE
#def rmse(y_true, y_pred):
#return np.sqrt(mean_squared_error(y_true, y_pred))

model.fit(X_train, y_train, epochs=epochs, batch_size=batch_size, verbose=0)

# Hacer predicciones en el conjunto de prueba
y_pred = model.predict(X_test)

# Predicciones en el conjunto de entrenamiento
#y_train_pred = model.predict(X_train)

# Predicciones en el conjunto de validación
#y_valid_pred = model.predict(X_valid)

# Inversión de la normalización

```

```

#y_valid_pred = scaler.inverse_transform(y_valid_pred.reshape(-1, 1))
#y_train_pred = scaler.inverse_transform(y_train_pred.reshape(-1, 1))
y_pred = scaler.inverse_transform(y_pred.reshape(-1, 1))
y_test = scaler.inverse_transform(y_test.reshape(-1, 1))

# Calcular el MSE en el conjunto de prueba, entrenamiento y validacion
mse = mean_squared_error(y_test, y_pred)
#mse_train = mean_squared_error(y_train, y_train_pred)
#mse_valid = mean_squared_error(y_valid, y_valid_pred)

#print(f'MSE en conjunto de entrenamiento: {mse_train}')
#print(f'MSE en conjunto de validación: {mse_valid}')
print(f'MSE en conjunto de prueba: {mse}')

#cross validation
#kf = KFold(n_splits=5)
#rmse_scorer = make_scorer(rmse, greater_is_better=False)
#predictions = cross_val_predict(model, X, y, cv=kf)

# Calcula el promedio y la desviación estándar del RMSE en las divisiones
#average_rmse = predictions.mean()
#std_rmse = predictions.std()

# Muestra los resultados
#print(f'Promedio RMSE: {average_rmse}')
#print(f'Desviación estándar del RMSE: {std_rmse}')

# Calcular el precio del día siguiente
last_window = X[-1].reshape(1, window_size, len(features))
next_day_prediction = model.predict(last_window)
next_day_prediction = scaler.inverse_transform(next_day_prediction)[0][0]

# Crear un gráfico de los valores de entrenamiento y las predicciones
plt.figure(figsize=(12, 6))
plt.plot(data.index[-len(y_test):], y_test, label='^Precio Real', color='aqua', linewidth=0.5)
plt.plot(data.index[-len(y_pred):], y_pred, label='Predicción', color='BLACK', linewidth=0.5)
plt.xlabel('Fecha', fontsize=10)
plt.xticks(fontsize=8)
plt.ylabel('Adj Close', fontsize=9)
plt.yticks(fontsize=8)
plt.legend(fontsize=8)
plt.title('LNTH ADJ CLOSE con Modelo LSTM', fontsize=12)
plt.grid(True, linestyle='--', alpha=0.7, linewidth=0.5)
plt.tight_layout()

# Guardar el gráfico como una imagen (descomentar la línea siguiente si deseas guardar)
plt.savefig(f'grafico_{params}.png')

# Guardar el gráfico en formato jpg
plt.savefig('WFC_Predictions.jpg', dpi=300, bbox_inches='tight')

plt.show()
# Agregar los resultados y el gráfico a la lista
results.append({
'Data Split': data_splits,
'Validation_split': validation_split,
'LSTM Units': lstm_unit,
'Second LSTM Units': second_lstm_units,
'Window Size': window_size,
'Epochs': epochs,
'Drop_out': Drop_out,
'Batch Size': batch_size,
#'mse_train': mse_train,
#'mse valid': mse_valid,
'MSE': mse,
'Next Day Prediction': next_day_prediction,
#'Graph': plt # Agregar el gráfico al resultado
})

```

```

results_df = pd.DataFrame(results)

results_df.to_csv('MELI_2_U_R_1.CSV', index=False)

model.summary()

import yfinance as yf
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import LSTM, Dense, Dropout
import matplotlib.pyplot as plt
from keras.optimizers import RMSprop

# Descargar datos de MRVL desde Yahoo Finance
mrvl_data = yf.download('MRVL', start='2014-01-01', end='2023-10-10')

# Seleccionar la columna Adj Close
mrvl_prices = mrvl_data['Adj Close'].to_frame()

# Crear columna con el precio del día siguiente (target)
mrvl_prices['Next Day Price'] = mrvl_prices['Adj Close'].shift(-1)

# Eliminar la última fila que contendrá NaN en 'Next Day Price'
mrvl_prices = mrvl_prices[:-1]

# Escalar los datos
scaler = MinMaxScaler()
mrvl_prices_scaled = scaler.fit_transform(mrvl_prices)

# Crear conjuntos X y Y
X, y = [], []
for i in range(len(mrvl_prices_scaled) - 10): # Usar una ventana de 10 días
    X.append(mrvl_prices_scaled[i:i+10, 0]) # Tomar solo la columna de precio
    y.append(mrvl_prices_scaled[i+10, 0]) # Tomar el precio del día siguiente

X, y = np.array(X), np.array(y)

# Dividir datos en conjuntos de entrenamiento y prueba (80-20)
split = int(0.8 * len(X))
X_train, X_val, y_train, y_val = X[:split], X[split:], y[:split], y[split:]

# Definir configuraciones a probar
neuronas = [32, 64]
dropouts = [0.1]
window_sizes = [10, 20]
epochs_list = [100]

# Almacenar resultados en un DataFrame
results_df = pd.DataFrame(columns=['Neurons', 'Dropout', 'Window Size', 'Epochs', 'MSE'])

# Iterar sobre las configuraciones
for n_neurons in neuronas:
    for dropout_rate in dropouts:
        for window_size in window_sizes:
            for n_epochs in epochs_list:
                # Redefinir y entrenar el modelo LSTM
                model = Sequential()
                model.add(LSTM(units=n_neurons, input_shape=(X_train.shape[1], 1)))
                model.add(Dropout(dropout_rate))
                model.add(Dense(units=1))
                model.compile(optimizer=RMSprop(), loss='mse')

# Entrenar el modelo con número de epochs especificado
model.fit(X_train, y_train, epochs=n_epochs, batch_size=32, validation_data=(X_val, y_val), verbose=0)

```

```

# Obtener predicciones en el conjunto de validación
val_predictions = model.predict(X_val)

# Desescalar las predicciones
val_predictions = scaler.inverse_transform(val_predictions.reshape(-1, 1))

# Calcular el error cuadrático medio (MSE)
mse = np.mean((val_predictions.flatten() - mrvl_prices['Next Day Price'].values[split:]**2)

# Almacenar los resultados en el DataFrame
results_df = results_df.append({
'Neurons': n_neurons,
'Dropout': dropout_rate,
'Window Size': window_size,
'Epochs': n_epochs,
'MSE': mse
}, ignore_index=True)

# Graficar la serie real y las predicciones (solo en los datos de validación)
plt.figure(figsize=(10, 6))
plt.plot(mrvl_prices.index[split:], mrvl_prices['Next Day Price'].values[split:], label='Real Prices', linewidth=0.5,
color='seagreen')
plt.plot(mrvl_prices.index[split:], val_predictions.flatten(), label=f'LSTM Predictions (Neurons={n_neurons},
Dropout={dropout_rate}, Window={window_size}, Epochs={n_epochs})', linestyle='dashed', linewidth=0.5, color='red')
plt.title('MRVL Next Day Price Predictions')
plt.xlabel('Date')
plt.ylabel('Next Day Price')
plt.legend()

# Guardar el gráfico en formato jpg
plt.savefig(f'MRVL_Predictions_{n_neurons}_{dropout_rate}_{window_size}_{n_epochs}.jpg', dpi=300,
bbox_inches='tight')

# Guardar los resultados en un archivo CSV
results_df.to_csv('LSTM_Results.csv', index=False)

```

Universidad de
San Andrés

10.-Bibliografía, ordenada alfabéticamente por apellido del autor

Bao, W., Yue, J., & Rao, Y, (2017), A deep learning framework for financial time series using stacked autoencoders and long-short term memory, PLoS One, 12(7), 1–24,

Benhabib, J., Wang, P., & Wen, Y, (2015), Sentiments and aggregate demand fluctuations, Econometrica, 83(2), 549–585,

Bernanke, B., & Kuttner, K, (2005), What explains the stock market’s reaction to federal reserve policy? The Journal of Finance, 60(3), 1221–1257,

Bhandari, H, N., Rimal, B., Pokhrel, N, R., Rimal, R., Dahal, K, R., & Khatri, R, K, (2022),

Predicting stock market index using LSTM, *Machine Learning with Applications*, 9, 100320,

Chaovalit, P., Gangopadhyay, A., Karabatis, G., & Chen, Z, (2011), Discrete wavelet transform- based time series analysis and mining, *ACM Computing Surveys*, 43(2), Chen, K., Zhou, Y., & Dai, F, (2015), A LSTM-based method for stock returns prediction:

Fischer, T., & Krauss, C, (2018), Deep learning with long short-term memory networks for financial market predictions, *European journal of operational research*, 270(2), 654669,

Fradkov, A, (2018), Early history of Machine Learning, *IFAC Papersonline* 53-2, 1385-1390,

Gao, P., Zhang, R., & Yang, X, (2020), The application of stock index price prediction with neural network, *Mathematical and Computational Applications*, 25(3), 53,

Gers, F, A., Schmidhuber, J., & Cummins, F, (2000), Learning to forget: Continual prediction with LSTM, *Neural Computation*, 12(10), 2451–2471,

Gers, F, A., Schraudolph, N, N., & Schmidhuber, J, (2003), Learning precise timing with lstm recurrent networks, *Journal of Machine Learning Research*, 3(null), 115–143, Greff, K., Srivastava, R, K., Koutník, J., Steunebrink, B, R., & Schmidhuber, J, (2017).

Graves, Alex & Fernandez, Santiago & Schmidhuber, Juergen. (2007). Multidimensional Recurrent Neural Networks. *Computing Research Repository*.

LSTM: A search space odyssey, *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), 2222–2232.

Hansen, J, V., McDonald, J, B., & Nelson, R, D, (1999), Time series prediction with genetic-algorithm designed neural networks: An empirical comparison with modern statistical models, *Computational Intelligence*, 15(3), 171–184,

Hochreiter, S, (1998), The vanishing gradient problem during learning recurrent neural nets and problem solutions, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02), 107–116,

Hochreiter, S., & Schmidhuber, J, (1997), Long short-term memory, *Neural Computation*, 9(8), 1735–1780,

Karmiani, D., Kazi, R., Nambisan, A., Shah, A., & Kamble, V, (2019), Comparison of predictive algorithms: Backpropagation, SVM, LSTM and Kalman filter for stock market, In 2019 amity international conference on artificial intelligence (AICAI) (pp, 228–234),

Maiti and P, Shetty D, (2020), "Indian Stock Market Prediction using Deep Learning", 2020 IEEE REGION 10 CONFERENCE (TENCON), Osaka, Japan, 2020, pp, 1215-1220,

Mokhtari, S., Yen, K, K., & Liu, J, (2021), Effectiveness of artificial intelligence in stock market prediction based on machine learning, *International Journal of Computer Applications* (0975 - 8887)

Murphy, J, J, (1999), *Technical analysis of the financial markets: a comprehensive guide to trading methods and applications*, Penguin,

Nelson, D, M., Pereira, A, C., & de Oliveira, R, A, (2017), Stock market's price movement prediction with LSTM neural networks, In *2017 international joint conference on neural networks (IJCNN)* (pp, 1419–1426), IEEE,

Nguyen, D, H, D., Tran, L, P., & Nguyen, V, (2019), Predicting stock prices using dynamic LSTM models, In *Applied Informatics: Second International Conference, ICAI 2019, Madrid, Spain, November 7–9, 2019, Proceedings 2* (pp, 199-212), Springer International Publishing,

Novianti, M, (2016), *Analysis on the influence of selected macroeconomic indicators (consumer price index, trade balance, non-farm payroll, housing starts, and S&P 500) towards US index (period 2010–2015)* (Ph,D, thesis), President University,

Ortega, L., & Khashanah, K, (2014), A neuro-wavelet model for the short-term forecasting of high-frequency time series of stock returns, *Journal of Forecasting*, 33(2), 134–146,

Otoo, M, W, (1999), *Consumer sentiment and the stock market*, Available at SSRN 205028,

Pan, W,-F, (2018), Does the stock market really cause unemployment? A cross-country analysis, *The North American Journal of Economics and Finance*, 44, 34–43,

Patel, J., Shah, S., Thakkar, P., & Kotecha, K, (2015), Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques, *Expert Systems with Applications*, 42(1), 259–268,

Qiu, M., & Song, Y, (2016), Predicting the direction of stock market index movement using an optimized artificial neural network model, *PLoS One*, 11(5), 1–11,

Qiu, J., Wang, B., & Zhou, C, (2020), Forecasting stock prices with long-short term memory neural network based on attention mechanism, *PLoS One*, 15(1), 1–15,

Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio, *How to construct deep recurrent neural networks*, arXiv preprint arXiv:1312.6026, 2013a,

Rodríguez-González, A., García-Crespo, A., Colomo-Palacios, R., Iglesias, F, G., & Gómez-Berbís, J, M, (2011), CAST: Using neural networks to improve trading systems based on technical analysis by means of the RSI financial indicator, *Expert Systems with Applications*, 38(9), 11489–11500,

Roondiwala, M., Patel, H., & Varma, S, (2017), Predicting stock prices using LSTM,

Sarno, L., & Thornton, D, L, (2003), The dynamic relationship between the federal funds rate and the treasury bill rate: An empirical investigation, *Journal of Banking & Finance*, 27(6), 1079–1110,

Staudemeyer, R, C., Morris, E,(2019), Understanding LSTM a tutorial into Long Short-Term Memory Recurrent Neural Networks.

Tatsat, H,, Puri, S,, & Lookabaugh, B, (2020), Machine Learning and Data Science Blueprints for Finance, O'Reilly Media,

Trinh, H, D,, Giupponi, L,, & Dini, P, (2018), Mobile traffic prediction from raw data using LSTM networks, In 2018 IEEE 29th annual international symposium on personal, indoor and mobile radio communications (PIMRC) (pp, 1827–1832), IEEE,

Wang, H,, & Raj, B, (2018), Predicting stock price trend using MACD optimized by historical volatility, Mathematical Problems in Engineering, 2018,

Wang, J,, & Kim, J, (2017), On the origin of Deep Learning, Carnegie Me2018,

Tannure, F. , Gerez, L. A., (2023), Uso de Deep Learning en la predicción de precios de las acciones, UNT.

Xiao, R,, Feng, Y,, Yan, L,, & Ma, Y, (2022), Predict stock prices with ARIMA and LSTM

Yadav, A,, Jha, C, K,, & Sharan, A, (2020), Optimizing LSTM for time series prediction in Indian stock market, Procedia Computer Science, 167, 2091–2100, International Conference on Computational Intelligence and Data Science,

Yu, P,, & Yan, X, (2019), Stock price prediction based on deep neural networks, Neural Computing and Applications, 32, 1609–1628,

Yule,G. U, “Why Do We Sometimes Get Nonsense Correlations Between Time Series? A Study in Sampling and the Nature of Time Series”, en Journal of the Royal Statistical Society, vol. 89, 1926, pp. 1-64.

Para amplias simulaciones Monte Carlo sobre regresiones espurias, véase C.W.J. Granger y P. Newbold, “Spurious Regressions in Econometrics”, en Journal of Econometrics, vol. 2, 1974, pp. 111-120.

Yungao, W,, Jianwei G,, (2018) AdaBoost-based long short-term memory ensemble learning approach for financial time series forecasting, Current Science , Vol, 115, No, 1, pp, 159-165