

IMPACTO DEL SENTIMIENTO DE LAS NOTICIAS EN EL ANÁLISIS PREDICTIVO DEL MERCADO

VÍCTOR OLIVERA BEGUÉ

GRADO EN ESTADÍSTICA APLICADA. FACULTAD DE CIENCIAS
UNIVERSIDAD AUTÓNOMA DE BARCELONA



14/06/2024

Tutor:

ARIEL DUARTE LÓPEZ

Índice general

Índice	I
Agradecimientos	III
Dedicatoria	IV
Resumen	V
Resum	VI
Abstract	VII
Introducción	1
1. Revisión de la Literatura	4
2. Datasets	10
2.1. Datos financieros y textuales	10
2.2. Análisis descriptivo	11
2.2.1. Preprocesado de datos (imputación, normalización)	11
2.2.2. Correlación	13
2.2.3. Cálculo Lag o Retardos (VAR)	16
2.2.4. Causalidad	18
3. Metodología	21
3.1. Long Short-Term Memory (LSTM)	21
3.2. Gated Recurrent Unit (GRU)	26
3.3. Comparativa	29
4. Configuración del experimento y resultados	31
Conclusiones y Trabajo Futuro	36
Bibliografía	43
A. Amazon	44
B. Apple	53

C. Facebook	62
D. Google	71
E. Netflix	80

Agradecimientos

Quiero agradecer a cada uno de los profesores y compañeros que han formado parte del camino para llegar hasta aquí. En este caso, una especial dedicatoria a Ariel Duarte, quien, aunque no me ha impartido clases en ninguna asignatura, ha sido un gran tutor para el trabajo final de carrera, guiándome y acompañándome para alcanzar un objetivo común. También quiero recalcar la importancia de mis compañeros más cercanos, por su compañía en los momentos difíciles y las horas de estudio compartidas.

Dedicatoria

Quiero aprovechar este espacio para dedicar esta tesis, en primer lugar, a mi madre, Paloma Begué, por su apoyo incondicional desde que nací, y especialmente en estos últimos años. También quiero expresar mi gratitud a mi familia y amigos por hacer el camino más ameno al acompañarme.

RESUMEN

El objetivo principal de este Trabajo Final de Grado (TFG) es evaluar y comparar el rendimiento de los modelos de predicción Long Short-Term Memory (LSTM) y Gated Recurrent Unit (GRU). Además, se investiga la influencia del sentimiento de noticias financieras y su importancia en comparación con el uso exclusivo de los precios de las compañías.

El estudio utiliza datos financieros y de sentimiento de las compañías del grupo FAANG (Facebook, Apple, Amazon, Netflix y Google) para el período comprendido entre el 2 de agosto de 2019 y el 29 de diciembre de 2023. Los datos se preprocesaron mediante técnicas de imputación y normalización. Se implementaron modelos LSTM y GRU, que son mejoras de las Redes Neuronales Recurrentes (RNN), con configuraciones que incluyen dos capas principales alternando con capas de dropout para combatir el sobreajuste. La evaluación de los modelos se realizó utilizando medidas como el Error Cuadrático Medio (MSE), el Coeficiente de Determinación (R^2) y el Error Absoluto Medio Porcentual (MAPE).

Los resultados del análisis indicaron que globalmente los modelos que incluyen la variable de sentimiento muestran una mejor capacidad predictiva en comparación con los que solo usan datos de precios históricos. En específico, el modelo GRU que incorporando datos de sentimiento financiero obtuvo los mejores resultados en términos de precisión y ajuste, superando tanto a los modelos que solo usan precios históricos como a los modelos LSTM con datos de sentimiento.

La investigación concluye que la inclusión de datos de sentimiento mejora significativamente la precisión de los modelos de predicción de precios de acciones. Además, se sugiere la posibilidad de aplicar mejoras en el análisis con la adición de metodologías como el análisis de componentes principales (PCA) y mecanismos de atención, entre otros, para mejorar aún más la capacidad predictiva en futuros trabajos.

RESUM

L'objectiu principal d'aquest Treball de Fi de Grau (TFG) és avaluar i comparar el rendiment dels models de predicció Long Short-Term Memory (LSTM) i Gated Recurrent Unit (GRU). A més, s'investiga la influència del sentiment de notícies financeres i la seva importància en comparació amb l'ús exclusiu dels preus de les companyies.

L'estudi utilitza dades financeres i de sentiment de les companyies del grup FAANG (Facebook, Apple, Amazon, Netflix i Google) per al període comprès entre el 2 d'agost de 2019 i el 29 de desembre de 2023. Les dades es van preprocessar mitjançant tècniques d'imputació i normalització. Es van implementar models LSTM i GRU, que són millores de les Xarxes Neuronals Recurrentes (RNN), amb configuracions que inclouen dues capes principals alternant amb capes de dropout per combatre el sobreajustament. L'avaluació dels models es va realitzar utilitzant mesures com l'Error Quadràtic Mitjà (MSE), el Coeficient de Determinació (R^2) i l'Error Absolut Mitjà Percentual (MAPE).

Els resultats de l'anàlisi van indicar que globalment els models que inclouen la variable de sentiment mostren una millor capacitat predictiva en comparació amb els que només usen dades de preus històrics. En específic, el model GRU que incorporava dades de sentiment financer va obtenir els millors resultats en termes de precisió i ajust, superant tant els models que només usen preus històrics com els models LSTM amb dades de sentiment.

La investigació conclou que la inclusió de dades de sentiment millora significativament la precisió dels models de predicció de preus d'accions. A més, es suggereix la possibilitat d'aplicar millores en l'anàlisi amb l'addició de metodologies com l'anàlisi de components principals (PCA) i mecanismes d'atenció, entre altres, per millorar encara més la capacitat predictiva en futurs treballs.

ABSTRACT

The main objective of this Final Degree Project (TFG) is to evaluate and compare the performance of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) prediction models. Additionally, the influence of financial news sentiment and its importance compared to the exclusive use of company prices is investigated.

The study uses financial and sentiment data from the FAANG group companies (Facebook, Apple, Amazon, Netflix, and Google) for the period between August 2, 2019, and December 29, 2023. The data was preprocessed using imputation and normalization techniques. LSTM and GRU models, which are improvements of Recurrent Neural Networks (RNN), were implemented with configurations that include two main layers alternating with dropout layers to combat overfitting. The evaluation of the models was carried out using measures such as Mean Squared Error (MSE), Coefficient of Determination (R^2), and Mean Absolute Percentage Error (MAPE).

The analysis results indicated that, overall, models that include the sentiment variable show better predictive ability compared to those that only use historical price data. Specifically, the GRU model incorporating financial sentiment data achieved the best results in terms of accuracy and fit, outperforming both models that only use historical prices and LSTM models with sentiment data.

The research concludes that the inclusion of sentiment data significantly improves the accuracy of stock price prediction models. Additionally, it suggests the possibility of applying improvements in the analysis with the addition of methodologies such as principal component analysis (PCA) and attention mechanisms, among others, to further enhance predictive capacity in future work.

Introducción

Históricamente, inversores y analistas han utilizado técnicas de Análisis Fundamental para evaluar indicadores financieros relacionados con las empresas, tales como los ingresos, las ganancias, las deudas y los dividendos, entre otros. Sin embargo, este tipo de análisis puede estar condicionado por la frecuencia de publicación de las variables económicas (muchas de estas variables se publican mensual o trimestralmente).

En la actualidad, el desarrollo tecnológico, y las nuevas técnicas de Inteligencia Artificial han permitido el análisis de conjunto de datos cada vez más extensos. Tomando como referencia los artículos de [Bahoo et al. \[2024\]](#) y [Biju et al. \[2024\]](#) se puede observar el notable aumento de las publicaciones de artículos relacionados con la Inteligencia Artificial y las Finanzas.

Los conjuntos de datos que se utilizan en este trabajo son las acciones de las compañías denominadas FAANG (Facebook, Apple, Amazon, Netflix y Google). Estas empresas representan algunas de las empresas tecnológicas más grandes y cotizadas del mercado actual. Son ampliamente conocidas y frecuentemente discutidas debido a su influencia en el mercado y su constante innovación por eso las convierten en un excelente caso de estudio. El período analizado abarca desde el 02-08-2019 hasta el 29-12-2023. Cabe destacar que dentro de este periodo se encuentra la pandemia de COVID-19, un evento que suele alterar los mercados significativamente. Uno de los objetivos de este proyecto es modelar el precio de estas compañías y analizar la influencia que puede tener una variable de sentimiento en el modelado de los datos. Esta variable representa si las noticias sobre las compañías para un día dado son positivas o negativas.

La utilización de una variable de sentimiento en el análisis se justifica por el impacto significativo que las opiniones y declaraciones de figuras influyentes pueden tener en los mercados. En el entorno actual, ya se han observado correcciones del mercado provocadas

por *tweets* u opiniones de referentes empresariales, como el visionario y polémico Elon Musk.

El 24 de marzo de 2021, Elon Musk anunció que Tesla aceptaría Bitcoin como forma de pago; ese mismo día, el precio de Bitcoin aumentó un 5,2 %. Otro caso notable ocurrió el 5 de enero de 2017, cuando el entonces presidente electo de los Estados Unidos, Donald Trump, tuiteó que impondría un impuesto sustancial a Toyota Motor si construían automóviles en su planta de México para luego comercializarlos en el mercado estadounidense. Ese mismo día, el precio de las acciones de Toyota cayó hasta un 3,1 % y el volumen de transacciones aumentó. Además, en 2019, un *tweet* de Donald Trump anunciando la imposición de nuevos aranceles a las importaciones chinas tuvo una reacción inmediata: los índices estadounidenses S&P 500 y Dow Jones sufrieron pérdidas significativas, mientras que otros activos considerados refugio, como el oro y los bonos del gobierno, experimentaron un aumento en su demanda. Para finalizar, el último caso, y seguramente el más especulativo, ocurrió el 4 de octubre de 2021, cuando Elon Musk, muy seguido por los inversores de criptomonedas, tuiteó una imagen de su perro, raza Shiba Inu, que casualmente compartía nombre con una criptomoneda categorizada como meme sin ninguna funcionalidad. Esta criptomoneda vio un aumento del 238 % en 4 días. Por tanto, esta variable podría enriquecer un modelo puramente matemático con información útil de carácter más público o subjetivo, proporcionando así una perspectiva adicional para la toma de decisiones.

Los objetivos de este Trabajo Final de Grado (TFG) son evaluar y comparar el rendimiento de los modelos de predicción *Long Short-Term Memory* (LSTM) [[Hochreiter and Schmidhuber, 1997](#)] y *Gated Recurrent Unit* (GRU) [[Cho et al., 2014](#)]. Además, investigar la influencia del sentimiento de noticias financieras y determinar su importancia en relación con el uso exclusivo de los precios de las compañías. Por tanto, respondemos a la pregunta ¿bajo condiciones idénticas, qué modelo es el más óptimo para estos casos, el LSTM o el GRU?

En el desarrollo de este TFG se ha tomado como referencia el trabajo de [Dahal et al. \[2023\]](#). Esta publicación analiza la serie de precios de la compañía Amazon y investiga si

existe alguna mejora en la predicción al añadir una variable de sentimiento. Los modelos que utilizan los autores son los mismos que se utilizan en este proyecto, LSTM y GRU. Estos modelos son una mejora de las Redes Neuronales Recurrentes (RNN, *Recurrent Neural Network*) dado que ambas arquitecturas resuelven el principal problema de las RNN, el desvanecimiento del gradiente. Los autores concluyen que los modelos que añaden la variable de sentimiento son significativamente mejores que los que usan únicamente los datos de precios históricos.

Este trabajo está estructurado de la siguiente manera, el Capítulo 1 contiene la revisión de los principales artículos que han guiado el desarrollo de este trabajo. El Capítulo 2 explica el origen, preprocesamiento y análisis descriptivo de los datos. El Capítulo 3 introduce las arquitecturas LSTM y GRU para analizarlas teóricamente y ver su funcionamiento, seguido de la configuración del experimento y sus resultados en el Capítulo 4. Finalmente, se presentan las conclusiones y futuro trabajo del TFG.

Capítulo 1

Revisión de la Literatura

La revisión de la literatura desempeña un papel crucial en el establecimiento del contexto necesario para comprender la relevancia y el estado actual del tema investigado. Este apartado se centra en la revisión y síntesis de artículos y estudios previos que han abordado cuestiones relacionadas, identificando las principales tendencias, debates y vacíos en la investigación. A través de un análisis crítico de estas fuentes, se establece una base sólida que justifica la necesidad del presente estudio y orienta su enfoque metodológico, contribuyendo a un entendimiento más profundo y estructurado del área de interés.

El estudio realizado por [Zhou et al. \[2023\]](#) tiene como objetivo principal integrar factores clave de la cadena de suministro de las empresas en varios modelos de aprendizaje automático, específicamente la red neuronal artificial (ANN, *Artificial Neural Network*), RNN, LSTM y GRU, para la predicción de precios de acciones, y llevar a cabo un análisis comparativo del rendimiento de estos modelos. Las variables que usa para medir la cadena de suministro son: los niveles de inventario, volúmenes de producción, desempeño del proveedor, eficiencia del transporte y pronósticos de demanda. La motivación de esta búsqueda radica en la incertidumbre sobre si las LSTM, al integrarse con datos de la cadena de suministro, superan a otros modelos como las Redes Neuronales Convolucionales (CNN, *Convolutional Neural Network*) o la Media Móvil Integrada Autoregresiva (ARIMA, *AutoRegressive Integrated Moving Average*) en la predicción de precios de acciones.

Los autores utilizaron cinco modelos en este artículo: los cuatro comentados anteriormente y un LSTM sin la variable de cadena de suministro.

En cuanto a la estructura, se menciona que los modelos fueron desarrollados con **TensorFlow** y **Keras**, utilizando la clase `Sequential()` para construirlos capa por capa con *Dense*, *Dropout* y sus correspondientes capas sin especificar su tamaño. Las métricas utilizadas para valorar el rendimiento del modelo fueron el Coeficiente de Determinación (R^2 , *Coefficient of Determination*), Error Absoluto Medio (MAE, *Mean Absolute Error*) y Error Cuadrático Medio (MSE, *Mean Squared Error*). Se compiló con el optimizador Adam, se utilizó MSE como función de pérdida y se hicieron 30 *epochs*.

Los autores concluyen que la inclusión de la variable de factores de la cadena de suministro mejoran significativamente el rendimiento de la predicción de las acciones y que los modelos GRU son los más adecuados para predecir los precios de las acciones utilizando datos de series temporales con factores de la cadena de suministro.

Por otra parte, el trabajo de [Gupta et al. \[2021\]](#) señala que las redes neuronales RNN solo pueden manejar ciertas dependencias a corto plazo y son propensas a la explosión o desaparición del gradiente, por lo tanto, tienen dificultades para capturar dependencias a largo plazo en los datos históricos. Para resolver estos problemas, se propone la estructura LSTM [[Hochreiter and Schmidhuber, 1997](#)], que luego fue mejorada y promovida [Graves \[2012\]](#). Por otro lado, se propone la estructura GRU como una de las variantes de RNN, introducida por [Cho et al. \[2014\]](#). Con estas nuevas estructuras de compuertas, se resuelve el principal problema de las RNN con sus dificultades para manejar información a larga distancia.

Los autores buscan comparar los resultados de predicción entre LSTM y GRU, previamente filtrando las variables menos importantes que solo contribuyen a la multicolinealidad con los métodos: Operador de Selección y Contracción Absoluta Mínima (LASSO, *Least Absolute Shrinkage and Selection Operator*) y Análisis de Componentes Principales (PCA, *Principal Component Analysis*). Los datos usados para el artículo son del índice Shanghai Composite,

desde el 11 de abril de 2007 hasta el 3 de agosto de 2021. Este índice bursátil se utiliza como referencia para medir el rendimiento general de la Bolsa de Valores de Shanghái, en China. Las variables usadas para el modelo son: el precio de apertura, precio más alto, precio más bajo, volumen de negociación y otros indicadores técnicos comunes, como el *On Balance Volume* (OBV), *Relative Strength Index* (RSI), *Commodity Channel Index* (CCI) y *Money Flow Index* (MFI), entre otros. A la vez, se usan otros indicadores como la *Psychological Line* (PSY), que refleja el estado de ánimo psicológico de los inversores.

En lugar de escoger una estructura fija para el modelo, se crea una combinación de parámetros para los cuatro modelos (LASSO-LSTM, LASSO-GRU, PCA-LSTM, PCA-GRU), con las combinaciones para buscar los parámetros ideales de: número de capas (2 o 3), número de neuronas por capa (8, 16 o 32) y valor de lag (10, 20, 30, 40 y 50). Para comparar el rendimiento se usan el MSE, Raíz del Error Cuadrático Medio (RMSE, *Root Mean Squared Error*), MAE y el tiempo de entrenamiento para el train y test. Para todas estas combinaciones se emplea un número de *epochs* de 1000, con una tasa de aprendizaje de 0.001 y una función de activación *tanh*.

En este artículo se sacan dos conclusiones. En primer lugar, los modelos LSTM y GRU pueden ser utilizados para predecir los precios de las acciones de manera efectiva, sin diferencia alguna. En segundo lugar, al comparar diferentes métodos de reducción de dimensiones, los resultados utilizando LASSO son en su mayoría mejores que aquellos que utilizan PCA.

Por otra parte, en la revisión de artículos de [Hu et al. \[2021\]](#) comenta el crecimiento del uso del Deep Learning para predecir el precio del mercado de divisas (*Forex*, *Foreign Exchange*) y acciones (*Stocks*). Esta revisión consta de 86 artículos, los cuales se escogen si superan los siguientes filtros: escritos entre 2015 y 2021, temática principal *Deep Learning*, mínimo de 5 citas, que incluyan áreas poco estudiadas dentro del campo y excluyendo revistas y conferencias publicadas informalmente. De esta recopilación, el 44% de los artículos tiene como temática principal las redes LSTM, lo que nos indica el gran crecimiento que ha tenido durante los últimos años. En el artículo también se muestran y comparan los resul-

tados de las métricas, pero estos no son concluyentes, ya que los datos, objetivos y modelos son distintos. Sin embargo, se sacan conclusiones sobre la metodología usada en los trabajos que emplean estructuras LSTM que nos pueden servir de ayuda: Según los conjuntos de datos utilizados, 3 documentos emplearon una combinación de análisis técnico, análisis de sentimientos y noticias para intentar predecir los movimientos en el mercado. El resto utilizó únicamente análisis técnico, con la excepción de un documento que utilizó recomendaciones de expertos. En cuanto a las variables, el precio de cierre fue la elección de 23 modelos basados en LSTM. Hubo 8 documentos que utilizaron únicamente el precio de cierre, y 12 documentos que incluyeron precio de cierre, precio de apertura, precio máximo y precio mínimo en sus entradas. Se encontró que 15 de los documentos modificaron el modelo LSTM tradicional para buscar un mejor rendimiento en la predicción. El LSTM basado en atención y el LSTM con RNN fueron los modelos más frecuentes, apareciendo en tres documentos diferentes. Dos documentos eligieron el método de LSTM con GRU para mejorar el modelo.

El análisis de [Feng \[2021\]](#) estudia el comportamiento de los modelos LSTM y GRU en la predicción del precio de las acciones del índice S&P 500, un índice mundialmente conocido por agrupar las 500 empresas con más valor de los Estados Unidos. Al mismo tiempo, investiga el rendimiento de los modelos añadiendo indicadores técnicos como el *weighted moving average*, *Bollinger Bands* e indicadores macroeconómicos como el *CBOE Volatility Index* y *Effective Federal Funds Rate*, entre otros.

Los datos de referencia en este artículo son del índice S&P 500, obtenidos de la librería *Yahoo Finance*. Estos fueron divididos en un 70 % para el entrenamiento y un 30 % para la prueba, respectivamente, para los tres tipos de modelos: LSTM, GRU y LSTM-GRU, aplicando los datos con múltiples variables o únicamente con la variable Close. Asimismo, aclaran que 30 *epochs* fue lo ideal para sus modelos. Para la evaluación, se utilizaron las métricas RMSE, MSE, MAE, R^2 y Accuracy.

El artículo concluye que el modelo que tiene mejor rendimiento es el LSTM, como se puede observar en todas las métricas, obteniendo los mejores valores. Además, informa que tener

en cuenta los indicadores técnicos y macroeconómicos no mejora el rendimiento del modelo; de hecho, lo disminuye considerablemente. Por último, expone la alternativa de utilizar PCA para filtrar los indicadores con mayor poder explicativo en próximos experimentos y utilizar un mecanismo de atención.

Otro trabajo en esta misma línea es el de [Dong \[2022\]](#) el cual se centra en comparar los resultados de predicción de dos modelos, ARIMA y LSTM, utilizando como respuesta el precio de las acciones de Google entre 2016 y 2022. Se plantea la hipótesis de que LSTM proporcionará una mayor precisión en las predicciones debido a su capacidad para extraer información valiosa de los datos históricos.

En cuanto la estructuración del modelo utilizado para evaluar el LSTM se realiza lo siguiente: para empezar, rellena los datos faltantes con el valor del día anterior, normaliza y estandariza los datos para evitar la explosión del gradiente y para garantizar un rango uniforme de medidas en las variables. Se asume que el lag sobre la variable objetivo es de 10 días. Además escogen aplicar una capa oculta de 40 neuronas y un learning rate de 0.05 con un número de iteraciones de 400.

Las conclusiones del estudio revelan que el modelo LSTM supera al modelo ARIMA en términos de ajuste y precisión de las predicciones. Así mismo se resalta que el modelo LSTM logra una mayor precisión al extraer de manera más efectiva la información histórica de los precios de las acciones.

El último artículo que se ha revisado es el de [Dahal et al. \[2023\]](#), que, como se ha comentado en la Introducción, usaremos como guía en nuestro trabajo. El objetivo del artículo es evaluar la precisión de los modelos LSTM y GRU, además de determinar si la inclusión de datos de sentimiento extraídos de noticias financieras mejora el rendimiento en la precisión.

Los autores usan los datos del índice estadounidense S&P 500, desde el 06-09-2008 hasta el 11-05-2021, con un mínimo de 25 noticias diarias de las fuentes *Reddit World News Channel* y *Finviz*, utilizando el paquete **VADER** para transformar el texto de las noticias en datos

capaces de capturar el sentimiento. La metodología del modelo explica que los datos son normalizados usando **min-max scaling**, como aparece en la ecuación (1.1), y divididos en tres subgrupos: entrenamiento, validación y prueba. Emplean la misma estructura y los mismos hiperparámetros que se muestran en la Tabla 1.1 para ambas metodologías.

$$Z = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (1.1)$$

Models	No of Neurons		Optimizer	Learning rate	Batch size	No of epochs
	1 hidden layer	2 hidden layer				
LSTM-News & LSTM	100	50	Adam	0.001	8	30
GRU-News & GRU	100	50	Adam	0.001	8	30

Tabla 1.1: *Lista de los mejores hyperparámetros para los modelos.*

Para evaluar los resultados, se utilizan las medidas RMSE, R^2 y el Error Absoluto Medio Porcentual (MAPE, *Mean Absolute Percentage Error*), como se muestra en la tabla 1.2:

Model	LSTM	GRU	LSTM-News	GRU-News
RMSE	39.9628	41.7668	34.8554	34.7262
R	0.9960	0.9961	0.9962	0.9962
MAPE	0.8267	0.8629	0.6703	0.6687

Tabla 1.2: *Métricas de los cuatro modelos.*

Se usan diferentes pruebas para evaluar estadísticamente si existen diferencias entre los resultados. Por un lado, se intenta hacer un estudio paramétrico usando el test ANOVA, pero no se puede confirmar el supuesto de normalidad, calculado con el test de D’Agostino [D’Agostino, 1970]. Por otro lado, dado que no se cumplen los supuestos de normalidad, se realiza el test no paramétrico de Kruskal-Wallis [Kruskal and Wallis, 1952]. En este caso, se cumple esta hipótesis, por lo que se procede al siguiente análisis. Se utiliza el test de Mann & Whitney [Mann and Whitney, 1947] para identificar entre qué residuos de los modelos existen diferencias significativas.

Capítulo 2

Datasets

En este capítulo se entra en detalle sobre la obtención, explicación de variables, preprocesado y descripción de los datos. Para la obtención de datos del grupo de acciones FAANG, conformado por Facebook, también conocida recientemente como Meta, Apple, Amazon, Netflix y Google, se utilizó la librería **yfinance**, una biblioteca de Python que ofrece una interfaz fácil de usar para extraer información financiera de Yahoo Finance. Esta librería permite a los usuarios acceder a datos sobre cotizaciones de acciones, índices, divisas, bonos, etc.

2.1. Datos financieros y textuales

En el conjunto de datos relacionado con los precios de las compañías tenemos las siguientes variables:

- **Date**: Esta variable representa la fecha asociada a los datos financieros en el conjunto de datos.
- **Open**: Es el precio al que se negoció el activo financiero al inicio de la jornada bursátil. Es el primer precio registrado durante ese día.
- **High**: Indica el precio más alto al que llegó el activo financiero durante el período de tiempo registrado. Es el valor más elevado observado en el día.
- **Low**: Representa el precio más bajo alcanzado por el activo financiero durante el mismo período de tiempo. Es el valor más bajo observado en el día.

- **Close:** Es el precio al cual se negoció el activo financiero al cierre de la jornada bursátil. Es el último precio registrado durante ese día y nuestra variable objetivo (*target*) para predecir.

- **Volume:** Indica la cantidad total de unidades del activo financiero que se han negociado durante el período de tiempo registrado. El volumen proporciona información sobre la actividad y la liquidez del mercado.

Los datos de sentimientos utilizados en este TFG fueron proporcionados por mi tutor. Estos datos contienen información sobre la positividad o negatividad asociada con una compañía y son extraídos de noticias financieras publicadas en Internet. Los datos de sentimiento tienen una frecuencia diaria y cubren el período desde 02-08-2019 hasta el 29-12-2023. Con esta información se ha construido un indicador de polaridad utilizando la ecuación:

$$\text{Sentiment} = \frac{\text{Positivo} - \text{Negativo}}{\text{Positivo} + \text{Negativo}}.$$

El resultado final de esta fórmula es un ratio que se puede interpretar como la polaridad hacia una compañía en un día particular.

Un resultado positivo indica que hay más críticas positivas que negativas. Por el contrario, un resultado negativo sugiere que hay más críticas negativas que positivas. Si el resultado es cero, esto señala que existe un equilibrio entre críticas positivas y negativas.

2.2. Análisis descriptivo

2.2.1. Preprocesado de datos (imputación, normalización)

Los datos se han agrupado teniendo en cuenta los días en que el mercado estuvo abierto durante el período mencionado. Esto permite evitar la imputación de datos en días festivos. Sin embargo, en algunos casos no se contaba con datos de sentimiento para días en los que el mercado seguía operativo. En tales casos, se decidió imputar los valores faltantes con interpolación lineal:

$$y = y_1 + \frac{x - x_1}{x_2 - x_1} \cdot (y_2 - y_1).$$

En esta fórmula, (x_1, y_1) y (x_2, y_2) representan los puntos conocidos más cercanos al valor que se desea interpolar. El valor x es el punto en el eje x para el cual se desea interpolar el valor y , que es el valor estimado. En este contexto, x_1 y x_2 son los días conocidos más cercanos al día sin datos de sentimiento, y y_1 y y_2 son los valores de sentimiento correspondientes. El valor x es el día para el cual queremos estimar el valor de sentimiento y .

La interpolación lineal se revela como una elección efectiva debido a su simplicidad y capacidad para mantener una relación lineal entre puntos próximos, siendo particularmente útil cuando la variación entre estos puntos es constante, una característica común en series temporales. El código implementado utiliza la función `interpolate()` de la librería **Pandas** para asegurar la continuidad de los datos de la variable de sentimiento. En la Tabla 2.1 se presenta una pequeña descripción de la distribución y cantidad de los datos faltantes.

Acción	Total Faltantes	Primera Fecha Faltante	Última Fecha Faltante	Duración del período
Amazon	8	2019-08-05	2021-12-20	868
Apple	7	2019-08-05	2021-12-20	868
Google	191	2019-08-05	2023-01-10	1254
Facebook	139	2019-08-05	2023-12-22	1600
Netflix	40	2019-08-05	2022-06-06	1036

Tabla 2.1: *Descriptiva datos faltantes*

La normalización de datos desempeña un papel fundamental en la preparación de datos para aplicar modelos de predicción en diversas áreas de la ciencia de datos y el aprendizaje automático. En este trabajo, se optó por utilizar la función `MinMaxScaler()` de la librería **Scikit-learn** para normalizar los datos, la cual escala cada característica individualmente en un rango específico, comúnmente entre 0 y 1. Esta uniformidad en la escala facilita la interpretación y comparación de las características entre sí. También ayuda a los modelos de predicción a converger más rápidamente durante el entrenamiento y puede mejorar su desempeño general.

2.2.2. Correlación

Spearman

Una vez que los datos han sido limpiados y preparados, se enfatiza la exploración de la correlación entre variables para obtener más información sobre ellas. La correlación implica que, a medida que una variable aumenta, la otra también lo hace o disminuye de manera predecible, aunque no necesariamente a un ritmo constante. En el caso de datos financieros, es más común utilizar la prueba de Spearman [Spearman, 1904] en lugar del test de Pearson [Pearson, 1900]. Este enfoque resulta especialmente útil porque las relaciones entre variables no suelen seguir una tendencia lineal y no cumplen los supuestos de normalidad requeridos por el coeficiente de correlación de Pearson.

En contraste con la medición de la fuerza y dirección de una relación lineal, característica del coeficiente de correlación de Pearson, el test de Spearman [Spearman, 1904] evalúa la relación de orden entre las observaciones. Se fundamenta en la clasificación de los datos y la comparación de los rangos asignados a cada observación, lo que lo hace robusto ante datos atípicos y distribuciones no normales. Para calcularlo, hemos usado de la librería **Pandas** la función `corr(method='spearman')`.

El resultado de este test devuelve dos valores: en primer lugar, la correlación de Spearman, como observamos en la Figura 2.1, con los datos de Amazon, que son los que tienen la variable Sentiment más correlacionada con Close; y, en segundo lugar, un p-valor, de la misma manera que en la Figura 2.2, asociado con las siguientes hipótesis:

- H_0 : No hay correlación entre las variables.
- H_1 : Existe una correlación entre las variables.

Distancia

Otra forma de calcular la correlación entre datos no lineales es mediante la correlación de distancia [Szekely et al., 2008]. La correlación de distancia se basa en la noción de distancia

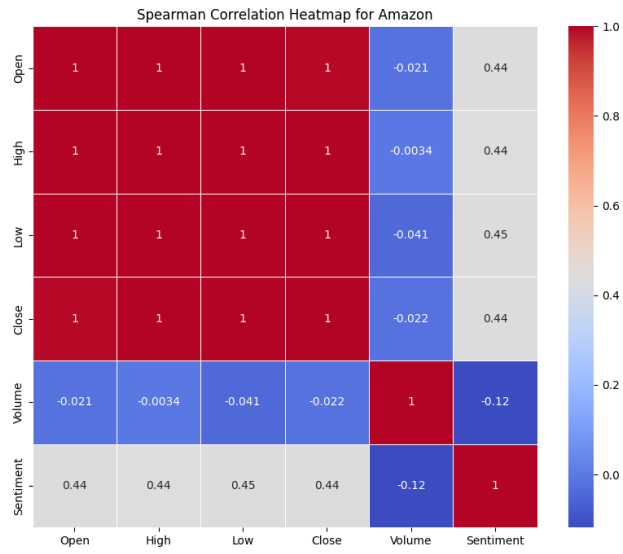


Figura 2.1: Mapa de calor sobre la correlación de Spearman para Amazon-News



Figura 2.2: Mapa de calor sobre P.Valores de Spearman para Amazon-News

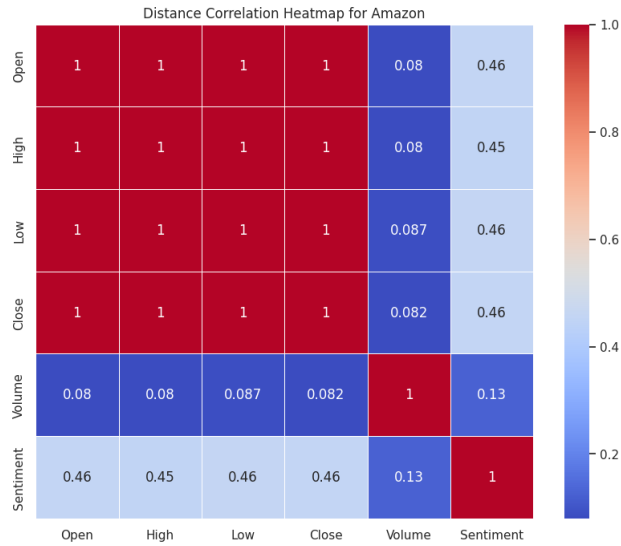


Figura 2.3: Mapa de calor sobre la correlación de distancia para Amazon-News

entre puntos en un espacio multidimensional y evalúa cómo la distancia entre los pares de puntos en dos conjuntos de datos se relaciona con la distancia entre los mismos pares de puntos en una proyección de los datos a un espacio de mayor dimensión. Para calcular esta correlación se usó la función `distance_correlation()` de la librería **statsmodels**. En la Figura 2.3 se añade otro mapa de calor (*heatmap*) con los cálculos sobre Amazon, y se observa un ligero aumento en la correlación entre las variables.

Ventanas

Para enfatizar en períodos más cortos se ha estudiado la correlación de Spearman en períodos de 20 días, que son aproximadamente los días que esta abierto el mercado cada mes, aunque como trabajo futuro se recomienda hacer un análisis más detallado sobre el tamaño de ventana que maximize la correlación entre ambas series. Los motivos principales para ver la correlación por ventanas son: el análisis localizado, la detección de cambios, la reducción de la influencia de la estacionalidad y la flexibilidad en el análisis. En las Figuras 2.2 y 2.3 se observa lo comentado y como periodos cercanos pueden tener una gran varianza.

	Open	High	Low	Close	Volume	Sentiment
Open	1.000000	0.962406	0.938346	0.873684	-0.327820	0.601504
High	0.962406	1.000000	0.969925	0.948872	-0.257143	0.584962
Low	0.938346	0.969925	1.000000	0.966917	-0.398496	0.616541
Close	0.873684	0.948872	0.966917	1.000000	-0.324812	0.554887
Volume	-0.327820	-0.257143	-0.398496	-0.324812	1.000000	-0.419549
Sentiment	0.601504	0.584962	0.616541	0.554887	-0.419549	1.000000

Tabla 2.2: Matriz de correlación de Amazon para el periodo 24/06/2022 al 13/07/2022

	Open	High	Low	Close	Volume	Sentiment
Open	1.000000	0.631579	0.690226	0.384500	-0.196992	-0.096241
High	0.631579	1.000000	0.709774	0.685478	-0.147368	-0.258647
Low	0.690226	0.709774	1.000000	0.848086	-0.323308	-0.102256
Close	0.384500	0.685478	0.848086	1.000000	-0.212942	-0.004515
Volume	-0.196992	-0.147368	-0.323308	-0.212942	1.000000	-0.078195
Sentiment	-0.096241	-0.258647	-0.102256	-0.004515	-0.078195	1.000000

Tabla 2.3: Matriz de correlación de Amazon para el periodo 20/07/2022 al 08/08/2022

2.2.3. Cálculo Lag o Retardos (VAR)

Como comenta [Wilkins \[2018\]](#) en su artículo *To Lag or Not to Lag*, el autor concluye que aplicar retardos mejora los modelos:

Es común el uso de modelos de regresión dinámica, una herramienta crucial en el análisis de series temporales. Estos modelos, que a menudo emplean variables dependientes rezagadas, son fundamentales para comprender procesos donde los valores actuales dependen de los anteriores. La inclusión de estas variables rezagadas es esencial para capturar la dinámica de los datos a lo largo del tiempo, lo que permite modelar de manera más precisa las relaciones complejas entre las variables.

El trabajo también destaca que las variables con retardo de diferencia (LDVs, *Lagged Difference Variables*) son una estrategia comúnmente empleada para eliminar la autocorrelación en los residuos y modelar procesos dinámicos de generación de datos. Aunque no son una solución definitiva, los académicos deberían seguir incluyendo con confianza las LDVs

como parte de una estrategia de estimación robusta. Esto sugiere que, al considerar el *lag* óptimo para cada variable explicativa del modelo, se puede mejorar la precisión de las estimaciones de los parámetros y evitar sesgos en los coeficientes. Para buscar el retardo óptimo hemos usado la función `VAR()`, la cual aplica el modelo *Vector Autoregression* [Sims, 1980]; esta función está dentro de la librería **statsmodels**.

Acción	Mejor Lag AIC	Valor AIC
Amazon	10	-11.5778
Apple	9	-12.6245
Google	3	-11.6637
Facebook	2	-10.9620
Netflix	4	-11.5639

Tabla 2.4: *Mejores lags según AIC*

Se escogió la métrica del criterio de información de Akaike (AIC, *Akaike Information Criterion*) para seleccionar el *lag* óptimo de la variable *sentiment* sobre *Close* proporcionado por el modelo, y esto se refleja en la Tabla 2.4. Para aplicar este retardo en la variable, se compararon las siguientes dos metodologías, escogiendo la que mantenía una correlación mayor:

1. **Media móvil exponencial (EMA, *Exponential Moving Average*)**: La media móvil exponencial es una técnica utilizada para suavizar series de tiempo, otorgando mayor peso a los valores más recientes. Se calcula mediante la fórmula:

$$\text{EMA}(t) = \alpha \cdot X(t) + (1 - \alpha) \cdot \text{EMA}(t - 1),$$

donde $\text{EMA}(t)$ es el valor de la media móvil exponencial en el tiempo t ; α es el factor de suavizado que determina el peso dado al valor más reciente, generalmente se elige un valor entre 0 y 1, donde valores más cercanos a 1 dan mayor peso a los valores más recientes; y $X(t)$ es el valor actual de la serie de tiempo en el tiempo t .

2. **Desplazamiento (*shift*)**: Este método desplaza el sentimiento el número óptimo de días según el lag indicado en la Tabla 2.4, de manera que el cierre del día cuadre con el retardo correspondiente del sentimiento. Esta fórmula es simple, ya que no implica ningún

cálculo adicional, únicamente un cambio de índice:

$$x_{\text{shift}(k),i} = x_{i-k}.$$

Como podemos observar en las Figuras 2.4 y 2.5, claramente el EMA esta más correlacionado; por lo tanto, es el que usaremos para seguir con la metodología.

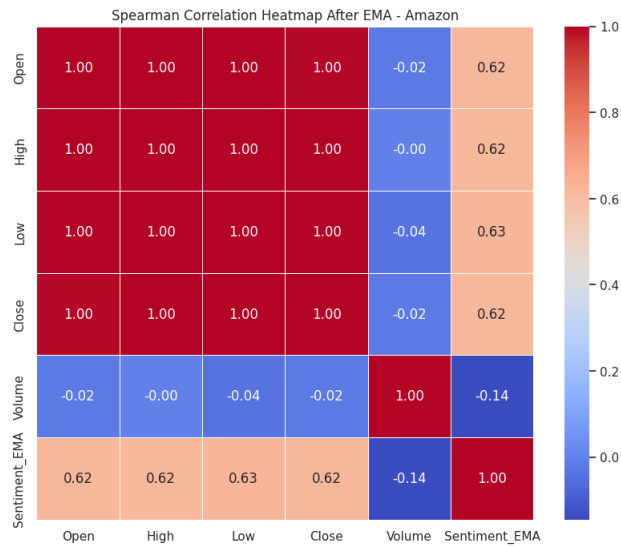


Figura 2.4: Mapa de calor sobre la correlación de Spearman para Amazon-News después de aplicar EMA a Sentiment

2.2.4. Causalidad

Cuando ya se han aplicado los retardos correspondientes a cada variable mediante EMA, se procede a verificar si existe causalidad entre cada variable y nuestra variable objetivo *Close*. Para analizar la supuesta causalidad, se utiliza el Test de Granger [Granger, 1969], una herramienta comúnmente empleada para evaluar la presencia de causalidad entre dos variables.

La hipótesis nula de este test establece que una variable no causa a la otra, es decir, que la información pasada de la primera variable no proporciona información adicional para predecir la segunda variable. En contraposición, la hipótesis alternativa indica lo contrario, sugiriendo que la primera variable sí proporciona información para predecir la segunda.



Figura 2.5: Mapa de calor sobre la correlación de Spearman para Amazon-News después de aplicar desplazamiento a Sentiment

- H_0 : La variable X no causa a la variable Y. En otras palabras, la información pasada de X no proporciona información adicional para predecir Y.
- H_1 : La variable X causa a la variable Y. Esto implica que la información pasada de X sí proporciona información útil para predecir Y.

Es crucial realizar el test en ambas direcciones, ya que la causalidad puede resultar bidireccional o unidireccional. Este enfoque ayuda a determinar si ambas variables se influyen mutuamente o si la influencia es unidireccional en una dirección específica. A continuación, veremos una tabla con los p.valores de la relación *Close* y Sentimiento después de aplicar la Media Móvil Exponencial (EMA), la relación con el resto de variables se puede ver en el apéndice de cada acción:

Los resultados de la tabla 2.5 indican que, aunque *Close* no causa a *Sentiment*, la variable $EMA(Sentiment)$ sí causa, bajo la óptica de Granger, a *Close*. Por lo tanto, esta información tiene valor para ayudar en la predicción futura.

Empresa	P.value Close - Sentiment_ EMA	P.value Sentiment_ EMA - Close
Amazon	0.5222	1.4829e-23
Apple	0.4318	4.3559e-26
Google	0.3335	0.0049
Facebook	0.5054	0.0005
Netflix	0.7112	1.7476e-13

Tabla 2.5: *P.valores test de Granger: Close - Sentiment_ EMA y Sentiment_ EMA - Close*

Capítulo 3

Metodología

En los siguientes capítulos se explorarán dos arquitecturas avanzadas de redes neuronales recurrentes: Long Short-Term Memory (LSTM) y Gated Recurrent Unit (GRU). Estas arquitecturas han sido diseñadas para superar las limitaciones de las Redes Neuronales Recurrentes (RNN) tradicionales, especialmente el problema del desvanecimiento del gradiente.

El capítulo sobre LSTM abordará su estructura y funcionamiento, destacando el uso de células de memoria y mecanismos de puertas (de entrada, de salida y de olvido) que permiten gestionar eficientemente la información a lo largo del tiempo.

El capítulo sobre GRU examinará esta arquitectura más simplificada, que combina las puertas de entrada y olvido en una sola puerta de actualización y utiliza una puerta de reinicio. Aunque más simple, las GRU son efectivas y más eficientes computacionalmente.

Finalmente, se presentará una comparativa entre LSTM y GRU, destacando sus diferencias en arquitectura, manejo de información y eficiencia. Esta comparativa ayudará a entender cuándo es más apropiado utilizar cada una para el análisis de series temporales.

3.1. Long Short-Term Memory (LSTM)

Esta estructura fue presentada por [Hochreiter and Schmidhuber \[1997\]](#) y su principal objetivo era mejorar las estructuras que habían hasta el momento: *Back-Propagation Through Time* (BPTT) y el *Real-Time Recurrent Learning* (RTRL), que como actualmente con las RNN sufrían problemas con la desaparición o explosión del gradiente. Este problema

dificultaba un aprendizaje efectivo para series temporales largas, porque los errores retropropagados, también conocidos como gradientes retropropagados, no se distribuían de manera adecuada.

La manera en que presenta el artículo para abordar el *output weight conflict* o los errores de salida que causan este desajuste con el gradiente, es una teoría creada por [Hochreiter \[1991\]](#) y [Silva et al. \[1996\]](#)) en la cual mediante puertas de entrada y salida se mantiene un flujo constante de error a través de unidades especiales de memoria.

La estructura que se propone utiliza células de memoria para mantener y actualizar información a lo largo del tiempo durante el procesamiento de las series temporales. Estas células son las responsables tanto de retener la información importante a corto plazo como de aprender patrones a largo plazo. Esta arquitectura introduce el concepto de puertas, como la de entrada, salida y olvido; estas controlan el flujo de información dentro de la célula de memoria. A continuación profundizaremos en la descripción y funcionamiento de estas herramientas. Las 4 partes principales de la estructura LSTM son:

- 1. Puerta de Entrada (*Input Gate*)

Como indica su nombre, la puerta de entrada es la responsable de controlar la información nueva que se debe ingresar en la memoria de la célula. Su función principal es decidir qué partes de la entrada actual y de la memoria anterior son consideradas para la actualización de la celda de memoria. La ecuación se refiere a la fórmula matemática para calcular la salida de la puerta de entrada, denotada como i_t , en el tiempo t se define como:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad (3.1)$$

donde el símbolo i_t representa la salida de la puerta de entrada en el tiempo t . La matriz de pesos W_i transforma la concatenación de la salida anterior h_{t-1} y la entrada actual x_t en el espacio de características deseado para la puerta de entrada. La salida de la capa oculta en el paso de tiempo anterior se denota como h_{t-1} , mientras que x_t representa la entrada en el tiempo actual. La función sigmoide, indicada por σ , mapea los valores a un rango entre 0 y 1. Por último, b_i es el sesgo asociado con la puerta de entrada.

- 2. Puerta de Olvido (*Forget Gate*)

La puerta de olvido es la responsable de decidir si la información dentro de la celda de memoria no es útil y debe ser olvidada, a la vez de detectar la necesaria para mantenerla para la siguiente iteración. De manera que su función principal es determinar la influencia que tendrá la información almacenada en la siguiente actualización. La salida de la puerta de olvido en el tiempo t , denotada como f_t , se calcula utilizando la función sigmoide aplicada a una combinación lineal de la entrada actual x_t y la salida anterior de la capa oculta h_{t-1} , más un sesgo (b_f). Matemáticamente, lo explicado sigue la fórmula:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f). \quad (3.2)$$

El símbolo f_t es la salida de la puerta de olvido en el tiempo t . La matriz de pesos W_f transforma la concatenación de h_{t-1} y x_t en el espacio de características deseado para la puerta de olvido. La salida de la capa oculta en el paso de tiempo anterior se denota como h_{t-1} , mientras que x_t es la entrada en el tiempo actual. La función sigmoide, representada por σ , comprime los valores a un rango entre 0 y 1. Finalmente, b_f es el sesgo asociado con la puerta de olvido.

Por lo tanto, la puerta de olvido f_t filtra la información de la celda anterior C_{t-1} que pasará a la memoria C_t . f_t devolverá un valor entre 0 y 1; cuando el valor sea cercano a 1, la celda olvidará la mayoría de la información, mientras que si es cercano a cero, retendrá la mayor parte de ella.

- 3. Candidata a Celda (*Cell Candidate*)

La candidata a celda es la responsable de crear un nuevo vector de estado que será combinado con el estado anterior para actualizar el estado de la celda. Esta candidata se crea usando una función tangente hiperbólica. La fórmula es:

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \quad (3.3)$$

donde \tilde{C}_t es el estado de celda candidato en el tiempo t . La matriz de pesos W_C transforma la concatenación de h_{t-1} y x_t en el espacio de características deseado para la celda candidata.

La función tangente hiperbólica, indicada por \tanh , mapea los valores a un rango entre -1 y 1. Finalmente, b_C es el sesgo asociado con la candidata a celda.

- 4. Estado de la Celda (*Cell State*)

El estado de la celda es el responsable de capturar tanto las tendencias a corto plazo como a largo. Como se ha explicado anteriormente mediante las puertas de entrada y olvido vamos almacenando la información útil a la vez que eliminamos la que no necesitamos en cada paso t , de esta manera el proceso no es inmediato, ya que el filtrado se hace poco a poco y permite retener memoria a largo plazo para capturar patrones. Este proceso se calcula con la fórmula:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t, \quad (3.4)$$

donde C_t es el estado de la celda en el tiempo t . El estado de la celda en el paso de tiempo anterior se denota como C_{t-1} . La salida de la puerta de olvido en el tiempo t , representada por f_t , determina cuánta información anterior debe mantenerse. La salida de la puerta de entrada en el tiempo t , indicada por i_t , determina cuánta información nueva se debe agregar a la memoria. Finalmente, \tilde{C}_t es la candidata a estado de la celda en el tiempo t , que representa la información potencialmente relevante que podría agregarse al estado de la celda.

- 5. Puerta de Salida (*Output Gate*)

La puerta de salida es la responsable de decidir qué parte de la celda actual C_t se utilizará para calcular el *output* de la red en el tiempo t . Es decir, su objetivo principal es determinar la manera de seleccionar y utilizar la información de la memoria de largo plazo para cada paso t .

Esta puerta está formada por 2 partes:

- 5.1 Cálculo de la salida

Este cálculo sirve para determinar cuánta información del estado de la celda actual debe ser transferida a la salida en cada momento t . Para la salida de la puerta se necesita

previamente hacer el siguiente cálculo:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o). \quad (3.5)$$

La salida de la puerta de salida en el tiempo t se denota como o_t . Para obtener esta salida, se utiliza una matriz de pesos, W_o , que transforma la concatenación de la salida de la capa oculta del paso de tiempo anterior, h_{t-1} , y la entrada en el tiempo actual, x_t , en el espacio de características deseado para la puerta de salida. La función sigmoide, σ , se aplica a esta transformación para mapear los valores a un rango entre 0 y 1, ajustando así la activación de la puerta. Además, se añade un término de sesgo, b_o , asociado con la puerta de salida para calibrar su respuesta.

- 5.2 Generación de la salida

Una vez que se conoce la salida de la puerta, se calcula la salida de la red. De esta manera, ayuda a regular la cantidad de información que se extrae para potenciar la memoria a largo plazo. El cálculo es el siguiente:

$$h_t = o_t \cdot \tanh(C_t). \quad (3.6)$$

En este contexto, la salida de la red en el tiempo t se denota como h_t . Esta salida representa el resultado de la red neuronal en ese instante temporal. Por otro lado, la función tangente hiperbólica, \tanh , se utiliza para mapear los valores a un rango entre -1 y 1. Esta función es comúnmente empleada en redes neuronales para introducir no linealidad en la salida de las neuronas, permitiendo así que la red capture relaciones más complejas en los datos de entrada.

Para terminar en la Figura 3.1 se ve la estructura completa de una capa LSTM entera de manera esquemática donde se puede ver el flujo de procesos explicados previamente.

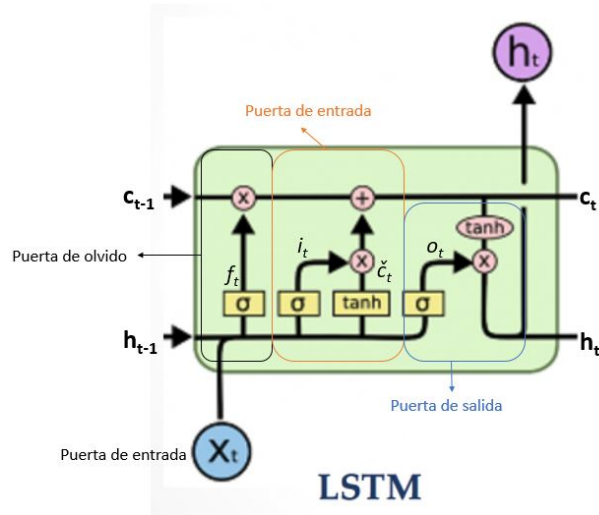


Figura 3.1: Arquitectura LSTM. Fuente: Encontrada en el artículo de [Ferro et al. \[2020\]](#) y con modificaciones propias

3.2. Gated Recurrent Unit (GRU)

Esta estructura fue presentada por [Cho et al. \[2014\]](#) y su principal objetivo era mejorar las estructuras existentes y variantes de RNN, simplificando su arquitectura mientras mantenían un rendimiento comparable en las tareas de procesamiento de secuencias. Las GRU también abordan problemas como la desaparición o explosión del gradiente, que dificultan el aprendizaje efectivo en series temporales largas.

La manera en que el artículo aborda estos problemas es integrando el uso de dos tipos de puertas: la puerta de reinicio (*reset gate*) y la puerta de actualización (*update gate*), que controlan adaptando la medida que se ha de recordar u olvidar en cada unidad oculta en cada secuencia.

A continuación, se entrará en detalle con las 4 partes principales de la estructura que son:

- 1. Puerta de Reinicio (*Reset Gate*):

La puerta de reinicio controla la cantidad de información de la memoria anterior que se debe olvidar. Su función principal es decidir qué partes de la memoria anterior $h^{(t-1)}$ deben

ser olvidadas al procesar la entrada actual x_t .

La fórmula matemática para calcular la salida de la puerta de reinicio, denotada como r_j , para la j -ésima unidad oculta en el tiempo t se define como:

$$r_j = \sigma \left([W_r x]_j + [U_r h^{(t-1)}]_j \right).$$

Considerando este escenario, r_j representa la salida de la puerta de reinicio, mientras que W_r y U_r son las matrices de pesos que transforman la entrada x y la memoria anterior $h^{(t-1)}$, respectivamente. Por otro lado, x es la entrada en el tiempo actual y $h^{(t-1)}$ es la salida de la capa oculta en el paso de tiempo anterior. Finalmente, σ se refiere a la función sigmoide, que mapea los valores a un rango entre 0 y 1.

- 2. Puerta de Actualización (*Update Gate*):

La puerta de actualización decide cuánto de la información anterior $h^{(t-1)}$ se debe mantener y cuánto de la nueva información $\tilde{h}_j^{(t)}$ se debe añadir.

La fórmula para calcular la salida de la puerta de actualización, denotada como z_j se define como:

$$z_j = \sigma \left([W_z x]_j + [U_z h^{(t-1)}]_j \right).$$

En este contexto, z_j representa la salida de la puerta de actualización, mientras que W_z y U_z son las matrices de pesos que transforman la entrada x y la memoria anterior $h^{(t-1)}$, respectivamente. Por otro lado, x es la entrada en el tiempo actual y $h^{(t-1)}$ es la salida de la capa oculta en el paso de tiempo anterior. Finalmente, σ se refiere a la función sigmoide, que comprime los valores a un rango entre 0 y 1.

- 3. Cálculo de la Activación Candidata (*Candidate Activation*):

La activación candidata $\tilde{h}_j^{(t)}$ es la nueva información que se combinará con la memoria anterior según lo que dicte la puerta de actualización.

La fórmula para calcular la activación candidata es:

$$\tilde{h}_j^{(t)} = \phi \left([Wx]_j + [U(r \odot h^{(t-1)})]_j \right).$$

Dentro de este marco, $\tilde{h}_j^{(t)}$ representa la activación candidata, donde ϕ es la función tangente hiperbólica que muestra los valores a un rango entre -1 y 1. Además, W y U son las matrices de pesos que transforman la entrada x y la memoria anterior multiplicada elemento a elemento por la puerta de reinicio r , respectivamente, denotada como $r \odot h^{(t-1)}$.

- 4. Actualización del Estado Oculto (Hidden State):

Finalmente, el estado oculto $h_j^{(t)}$ se actualiza como una interpolación entre la memoria anterior y la activación candidata controlada por la puerta de actualización.

$$h_j^{(t)} = z_j h_j^{(t-1)} + (1 - z_j) \tilde{h}_j^{(t)}.$$

En este contexto, $h_j^{(t)}$ representa el estado oculto, mientras que z_j denota la salida de la puerta de actualización. Por otro lado, $\tilde{h}_j^{(t)}$ se refiere a la activación candidata y $h_j^{(t-1)}$ representa la memoria anterior.

Estas ecuaciones permiten que las GRU recuerden y olviden información de manera adaptativa, lo que las hace efectivas para tareas de modelado de secuencias. Podemos ver en la Figura 3.2 la representación de la estructura de una capa GRU.

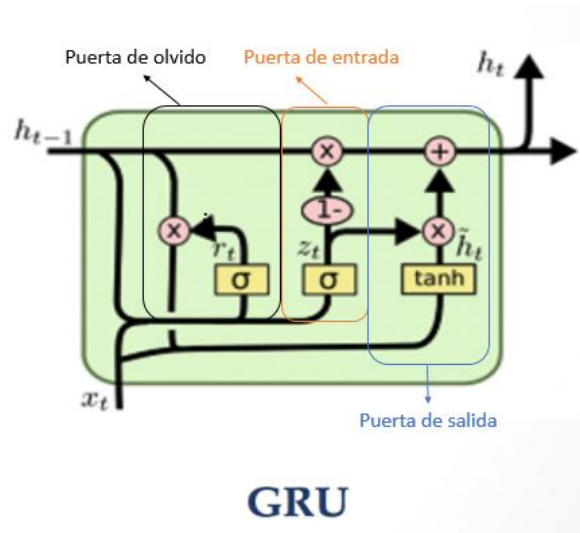


Figura 3.2: Arquitectura GRU. Fuente: Encontrada en el artículo de [Abdulwahab et al. \[2017\]](#) y con modificaciones propias

3.3. Comparativa

Aunque LSTM y GRU pertenecen a la familia de las redes neuronales recurrentes y fueron creadas para manejar datos temporales y abordar el problema del desvanecimiento y explosión del gradiente, tienen ciertas diferencias en estructura, funcionamiento y uso. A continuación, las veremos en detalle, comentando y comparando cada una de ellas.

En cuanto a la estructura, como hemos observado anteriormente, podemos observar ciertas diferencias. LSTM utiliza células de memoria para mantener y actualizar información a lo largo del tiempo mediante las puertas de entrada, olvido y salida, mientras que GRU utiliza el estado oculto para almacenar y transferir información mediante la puerta de actualización, que combina las funciones de puerta de entrada y olvido en una sola, y la puerta de reinicio.

Sobre la capacidad de manejo de información, también podemos ver ciertas diferencias. Mientras que LSTM utiliza células de memoria que se encargan de almacenar información de manera más robusta, GRU simplifica esta estructura juntando dos puertas en una sola.

Además, LSTM permite tener más control sobre la cantidad de información que olvidamos o retenemos, mientras que GRU, al no tener una célula de memoria separada, tiene una actualización de la información más directa, lo que complica la capacidad de aprender tendencias a largo plazo.

Finalmente, en términos de computación y eficiencia, se observan variaciones. Como se mencionó anteriormente, LSTM cuenta con más puertas y una célula de memoria, lo que le otorga la ventaja de captar mejor los patrones a largo plazo. Sin embargo, esto también implica un mayor coste computacional en comparación con GRU, debido a su estructura más compleja. En la siguiente figura, podremos observar las diferencias en las estructuras y el funcionamiento entre RNN, LSTM y GRU

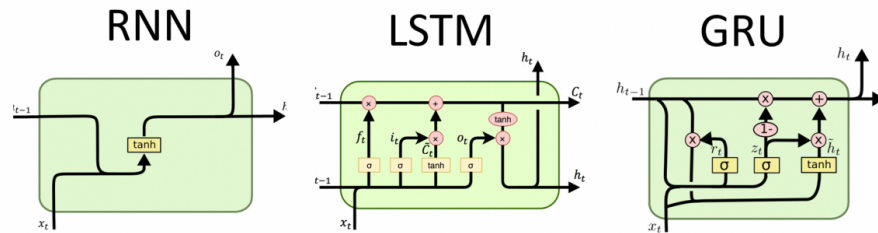


Figura 3.3: Arquitecturas RNN, LSTM y GRU. Fuente: Artículo de [Toharudin et al. \[2021\]](#)

En la Tabla 3.1 se muestran las diferencias de una manera clara y esquemática.

Característica	LSTM	GRU
Puertas	Entrada, Olvido, Salida	Actualización, Reinicio
Estado de la Célula	Sí, separado del estado oculto	No, combina estado de célula y estado oculto
Capacidad de Información	Mejor para dependencias a largo plazo	Menor capacidad para largo plazo, pero más eficiente
Complejidad Computacional	Más complejo y costoso	Menos complejo y más rápido
Rendimiento	Mejor en tareas con dependencias a largo plazo	Mejor en tareas con dependencias a corto/medio plazo

Tabla 3.1: Comparativa entre LSTM y GRU.

Capítulo 4

Configuración del experimento y resultados

La selección de los hiperparámetros es crucial para el rendimiento del modelo. Al optimizarlos, buscamos obtener la mejor configuración de hiperparámetros que permitan mejorar la capacidad predictiva del modelo. Esta metodología implica explorar diferentes combinaciones de configuraciones para encontrar la que se acerque más a los resultados deseados. Además de hacer una buena selección, es fundamental tener una estructura lógica de las capas. En nuestro caso, hemos usado la misma para todos los modelos. Esta contiene dos capas LSTM o GRU alternando con capas de dropout para combatir el sobreajuste. Esta estructuración fue tomada del artículo de referencia escrito por [Dahal et al. \[2023\]](#). Aunque en su propuesta solo se componen de las dos capas principales, nuestra propuesta de mejora fue añadir las capas de dropout. Para finalizar la arquitectura, se añade una capa `Dense(1)`, la cual es responsable de transformar todos los vectores anteriores de alta dimensionalidad a un único número continuo que representa la predicción del precio de la compañía en el instante de tiempo t . En la Figura 4.1, se muestra una representación de la arquitectura en el trabajo.

Utilizamos una estrategia del tipo *Grid search* para encontrar la mejor configuración de hiperparámetros. En este estudio, se investigaron los mejores resultados entre las diferentes combinaciones de parámetros, que se muestran en la Tabla 4.1, y se evaluaron en función

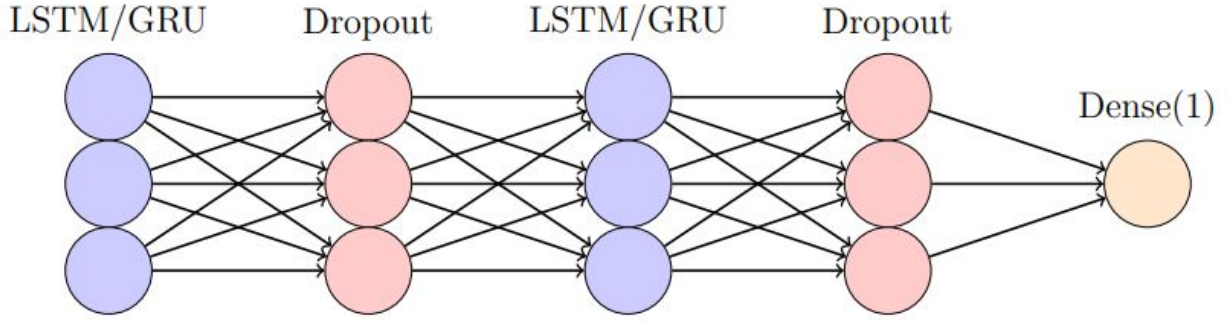


Figura 4.1: Estructura modelo. Fuente: Propia, elaborada con la librería tikzpicture

de su mejor función de pérdida.

Hiperparámetro	Descripción	Intervalos Evaluados
<i>units1</i>	Número de unidades en la primera capa LSTM/GRU	50, 75, 100, 125, 150
<i>dropout1</i>	Tasa de dropout en la primera capa LSTM/GRU	0.1, 0.2, 0.3
<i>units2</i>	Número de unidades en la segunda capa LSTM/GRU	25, 50, 75
<i>dropout2</i>	Tasa de dropout en la segunda capa LSTM/GRU	0.1, 0.2, 0.3
<i>learning_rate</i>	Tasa de aprendizaje del optimizador Adam	1×10^{-2} , 1×10^{-3} , 1×10^{-4}

Tabla 4.1: Intervalos de hiperparámetros evaluados en el experimento

En los Apéndices se muestran las combinaciones de hiperparámetros escogidos para cada modelo. Durante este proceso, utilizamos la combinación de épocas (*epochs*) y tamaño de los lotes (*batch size*) iguales a 64 y 32 respectivamente.

Otra medida a considerar es la curva de aprendizaje del modelo, evaluando el rendimiento del modelo con el conjunto de datos de evaluación. Esta idealmente debería mostrar una caída exponencial. Esto significa que al principio se debería observar una rápida disminución en el error, seguido de una estabilización a lo largo del tiempo.

Por último, se extraen conclusiones sobre los resultados de las predicciones de los modelos seleccionados para los conjuntos de datos de prueba. Primero, observamos Figuras como

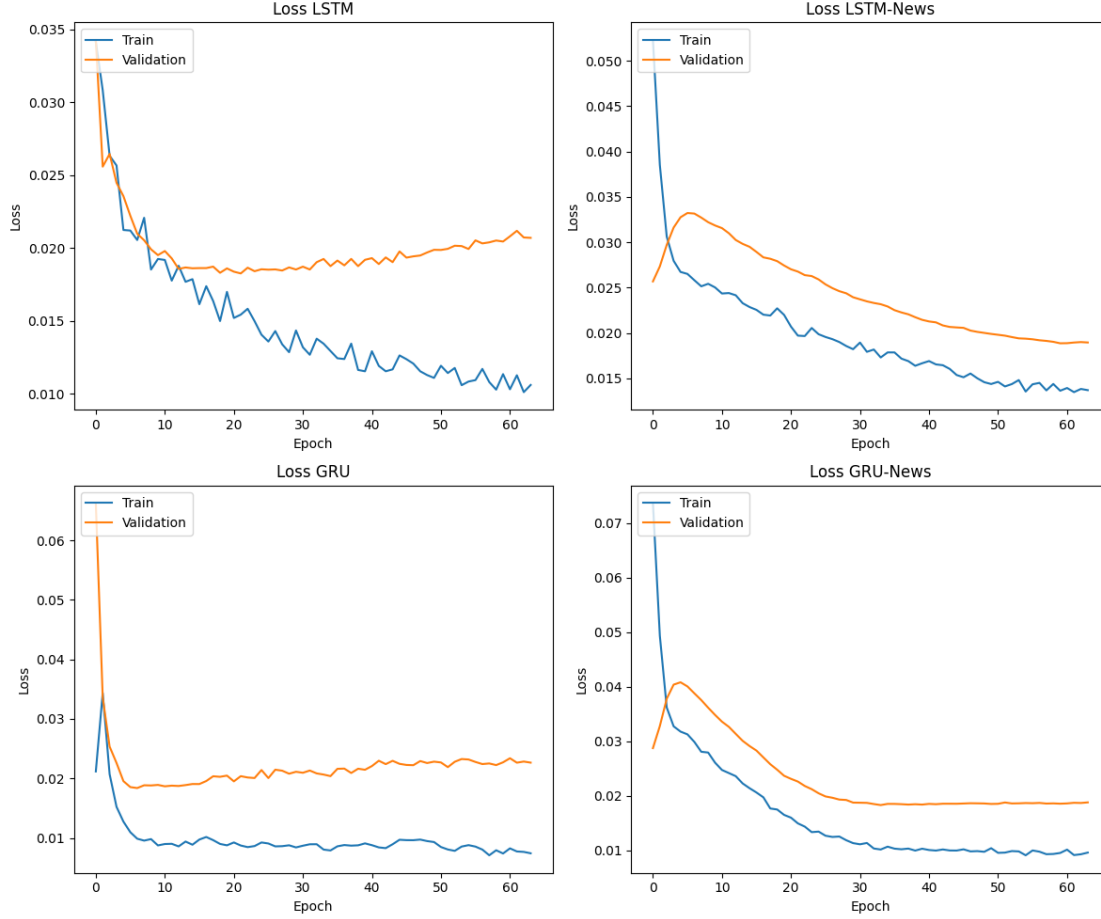


Figura 4.2: *Function Losses de Amazon*

la 4.3, que nos permiten obtener una idea visual de la precisión de las predicciones en comparación con los datos reales. Posteriormente, utilizamos las métricas mostradas en la Tabla 4.4 para establecer cuál es el modelo con mejor capacidad predictiva. En este caso, empleamos las métricas R^2 , $MAPE$ y como referencia MSE , tanto para compilar como para la evaluación final. A diferencia de Dahal et al. [2023] que usaban: R^2 , $MAPE$ y MSE .

Modelo	LSTM	GRU	LSTM-News	GRU-News
MSE	0.000883961	0.00213406	0.000823402	0.000258577
R	0.969909	0.927354	0.97197	0.991198
MAPE	7.26022	11.5721	7.11672	5.44249

Figura 4.4: *Métricas de Amazon para Epochs=64 y Batch Size=32*

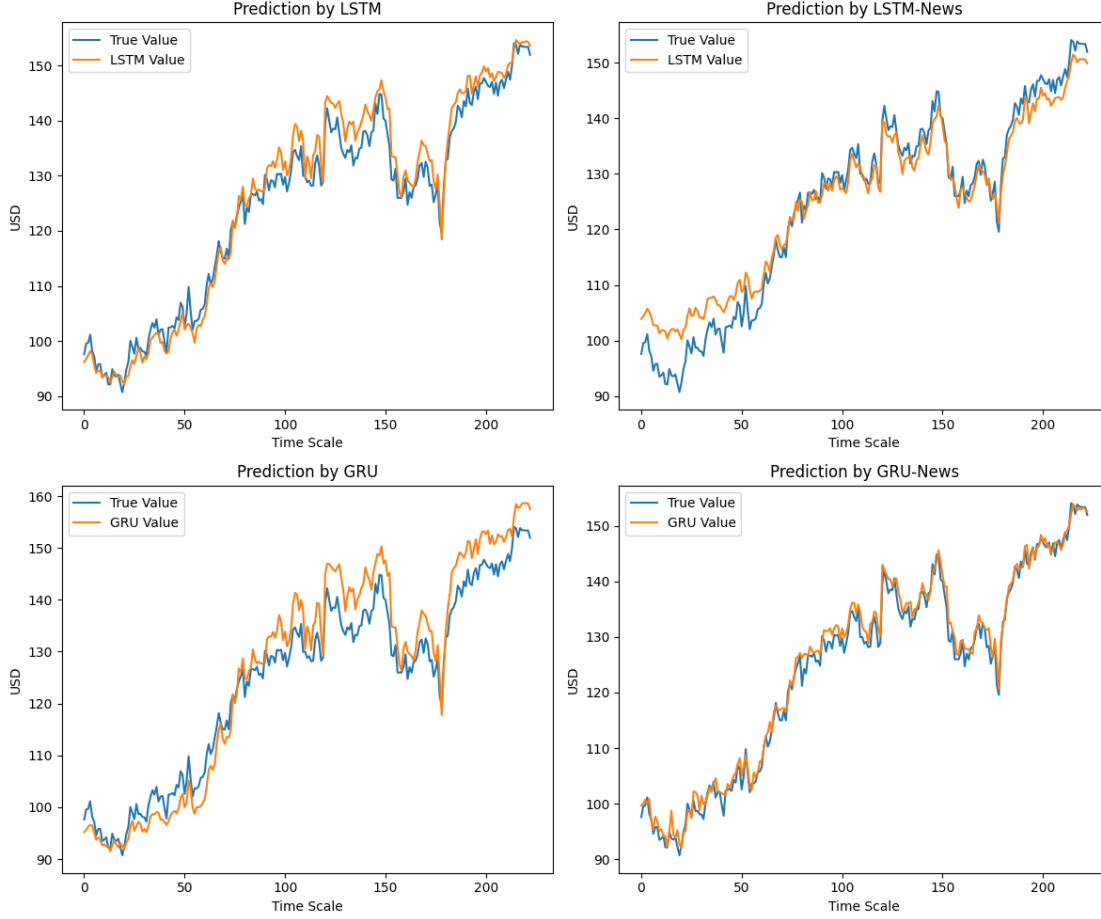


Figura 4.3: *Amazon: Predictions vs real data*

Siguiendo con la metodología de los autores, usaremos las mismas herramientas estadísticas para detectar si hay diferencias entre los modelos. Primero, intentamos comprobar mediante un test paramétrico, como el ANOVA, si no hay diferencias entre los residuos de los modelos. Sin embargo, ningún caso cumple el supuesto de normalidad para los residuos de todos los modelos, como observamos en los Apéndices de los QQ plots y corroboran los resultados del test de D'Agostino [D'Agostino, 1970].

Seguidamente, se examina desde un punto de vista no paramétrico, primero con el test de Kruskal & Wallis [Kruskal and Wallis, 1952], el cual nos indica si los residuos de uno de los modelos tiene diferencias con el resto del grupo en caso de un p-valor menor a la significancia que estemos buscando. En caso de obtener un valor que se diferencie, usaremos el test de

Mann & Whitney [[Mann and Whitney, 1947](#)] para recorrer todas las combinaciones de parejas posibles y confirmar entre cuáles obtenemos diferencias significativas, los resultados están en los apéndices.

Conclusiones y Trabajo Futuro

Para concluir empezaremos comentando nuestros resultados. Hemos podido observar que aunque presentan diferencias todos los grupos de modelos según Kruskal & Wallis [Kruskal and Wallis, 1952], únicamente Google concluye que todos los modelos son significativamente diferentes, según el test de Mann & Whitney [Mann and Whitney, 1947]. Para el resto, todas son diferentes menos para una combinación de modelos. En el caso de Amazon, no se han podido encontrar diferencias entre LSTM y LSTM-News, para Facebook entre LSTM y GRU, en Google para LSTM-News y GRU-News y para Netflix entre LSTM y GRU.

Acción	Mejor Modelo	Comentarios
Amazon	GRU-News	El mejor modelo es GRU-News.
Apple	LSTM	Las dos estructuras LSTM funcionan mejor que las GRU. En el caso de GRU, con el sentimiento añadido tiene un mejor rendimiento.
Facebook	GRU-News	En ambos casos, los modelos con la variable sentiment predicen mejor.
Google	LSTM-News y GRU-News	En ambos casos, los modelos con la variable sentiment predicen mejor que sin ella.
Netflix	GRU-News	El mejor modelo es GRU-News.

Tabla 4.2: Conclusiones sobre los mejores modelos para distintos conjuntos de datos

La Tabla 4.2 muestra que el modelo GRU-News es el más efectivo en cuatro de los cinco casos, compartiendo el primer lugar con LSTM-News en uno de ellos. Incluso cuando no es el mejor, GRU-News mejora significativamente con la inclusión de la variable *sentiment*. La Tabla 4.3 refuerza estos hallazgos, destacando el desempeño sobresaliente de GRU-News en términos de RMSE y MAPE, con promedios de 0.000482 y 4.189 respectivamente, y un coeficiente de determinación promedio de 0.978, demostrando su capacidad para explicar la variabilidad de los datos de manera más precisa y consistente que otros modelos.

En comparación, los modelos LSTM y GRU sin la variable *sentiment* presentan mayores

errores y menor capacidad predictiva, con RMSE promedios de 0.001066 y 0.001407, y MAPE promedios de 8.268 y 8.330. Los modelos que incluyen *sentiment* (LSTM-News y GRU-News) mejoran en desempeño, siendo más significativa esta mejora en GRU-News. En el caso de LSTM, *sentiment* mejora los modelos para Facebook y Google, empeora para Apple y Google, y no muestra diferencias significativas para Amazon. Esto indica que la variable *sentiment* es especialmente beneficiosa en la estructura GRU, mejorando la precisión en todos los casos.

Los resultados indican una tendencia clara de mejora en los modelos que incluyen la variable *sentiment* extraída de las noticias. El modelo GRU-News, en particular, se destaca por su mejor rendimiento, lo que sugiere que la incorporación de información de sentimiento es especialmente beneficiosa en este tipo de arquitectura de red neuronal. Además, el respaldo del Test de Granger refuerza la relevancia de la variable *sentiment* en la predicción precisa del movimiento de *close*.

Model	MSE	R	MAPE
LSTM	0.001066	0.954949	8.268128
GRU	0.001407	0.938619	8.330290
LSTM-News	0.001430	0.927086	9.789350
GRU-News	0.000482	0.978298	4.189154

Tabla 4.3: Promedios de las métricas de desempeño para diferentes modelos

A continuación se detallan algunas ideas que podrían considerarse como parte del trabajo futuro:

- **Rendimiento de Modelos:** Hemos demostrado que el modelo GRU-News supera consistentemente a los demás modelos en la predicción de precios de acciones.

- **Influencia del Sentimiento de Noticias:** Los resultados indican que la incorporación de la variable *sentiment* mejora significativamente el rendimiento de los modelos, especialmente en el caso del GRU-News. El Test de Granger confirmó que el sentimiento con su lag correspondiente es causal al movimiento de los precios de cierre *Close*, subrayando la importancia de esta variable.

- **Comparación de Modelos LSTM vs GRU:** Bajo condiciones idénticas, el modelo GRU-News es superior al LSTM-News en términos de precisión y consistencia en la predicción de precios de acciones. La estructura GRU se beneficia más de la inclusión de la variable de sentimiento.

- **Lag Óptimo del Sentimiento:** La identificación del *lag* óptimo ha sido crucial para entender la relación temporal entre el sentimiento de las noticias y el precio de las acciones. Esto ha permitido mejorar la precisión de las predicciones.

- **Metodología Detallada:** Hemos desarrollado e implementado una metodología exhaustiva que incluye la normalización de datos y la evaluación de rendimiento mediante métricas estándar. Esta metodología ha sido fundamental para garantizar la calidad de los resultados obtenidos.

- **Calidad y Robustez de los Modelos:** Las conclusiones muestran que los modelos propuestos, particularmente el GRU-News, son robustos y de alta calidad para la predicción de precios de acciones. La inclusión de información de sentimiento ha demostrado ser especialmente beneficiosa, validando la relevancia de combinar datos financieros con análisis de sentimiento.

En cuanto a las mejoras posibles, se identificaron varias estrategias que podrían haber optimizado aún más el estudio:

- **RNN Bidireccionales:** Implementar RNN bidireccionales, como se sugiere en [Schuster and Paliwal \[1997\]](#), podría mejorar significativamente el rendimiento. Estas redes, con dos capas que procesan los datos en direcciones opuestas, capturan información en ambas direcciones de la secuencia.

- **Mecanismo de Atención:** El uso del mecanismo de atención [\[Bahdanau et al., 2014\]](#) podría mejorar el rendimiento de los modelos. Este mecanismo asigna pesos a los valores de entrada según su importancia para la predicción, dando mayor peso a los valores más influyentes y mejorando así la precisión del modelo.

- **Modelo Transformer:** Combinar el modelo Transformer con LSTM y GRU podría

ser beneficioso debido a su capacidad para usar automáticamente el mecanismo de atención, capturando eficientemente las dependencias a largo plazo.

- **Diversificación de Variables y Reducción de Dimensionalidad:** Añadir más tipos de variables, como índices macroeconómicos, y aplicar técnicas de reducción de dimensionalidad como LASSO o PCA, ayudaría a eliminar variables multicolineales, reduciendo el ruido y el sobreajuste del modelo.

- **Optimización de Parámetros:** Incluir la búsqueda de los mejores valores para *Epochs* y *Batch Size* dentro del proceso de *Grid Search* podría haber mejorado los resultados. Aunque se intentó, hubo limitaciones debido a la potencia computacional disponible.

Bibliografía

- S. Abdulwahab, M. Jabreel, and D. Moreno. *Deep Learning Models for Paraphrases Identification*. PhD thesis, 09 2017.
- D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *ArXiv*, 1409, 09 2014.
- S. Bahoo, M. Cucculelli, X. Goga, and J. Mondolo. Artificial intelligence in finance: a comprehensive review through bibliometric and content analysis. *SN Business & Economics*, 4(2):23, Jan. 2024. ISSN 2662-9399. doi: 10.1007/s43546-023-00618-x. URL <https://doi.org/10.1007/s43546-023-00618-x>.
- A. K. V. N. Biju, A. S. Thomas, and J. Thasneem. Examining the research taxonomy of artificial intelligence, deep learning & machine learning in the financial sphere—a bibliometric analysis. *Quality & Quantity*, 58(1):849–878, Feb. 2024. ISSN 1573-7845. doi: 10.1007/s11135-023-01673-0. URL <https://doi.org/10.1007/s11135-023-01673-0>.
- K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- R. D’Agostino. Transformation to normality of the null distribution of g 1. *Biometrika*, 57, 12 1970. doi: 10.2307/2334794.
- K. R. Dahal, N. R. Pokhrel, S. Gaire, S. Mahatara, R. P. Joshi, A. Gupta, H. R. Banjade, and J. Joshi. A comparative study on effect of news sentiment on stock price prediction with deep learning architecture. *PLOS ONE*, 18(4):1–19, 04 2023. doi: 10.1371/journal.pone.0284695. URL <https://doi.org/10.1371/journal.pone.0284695>.

- H. Dong. Stock Price Prediction Using ARIMA and LSTM. *2022 6th Annual International Conference on Data Science and Business Analytics (ICDSBA)*, pages 195–201, 2022. doi: 10.1109/ICDSBA57203.2022.00055. URL http://resolver.scholarsportal.info/resolve/28315944/v2022inone/195_sppuaal.
- Z. Feng. Research on the S&P 500 Index Based on LSTM and GRU. 2021.
- C. Ferro, N. Celis Mayorga, and A. Casallas García. Llenado de series de datos de 2014 a 2019 de pm2.5 por medio de una red neuronal y una regresión lineal. 08 2020. doi: 10.13140/RG.2.2.35092.53126.
- C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. 1969. URL <https://api.semanticscholar.org/CorpusID:41012146>.
- A. Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*, volume 385. 01 2012. ISBN 978-3-642-24796-5. doi: 10.1007/978-3-642-24797-2.
- P. Gupta, Y. Gao, R. Wang, and E. Zhou. Stock prediction based on optimized lstm and gru models. *Scientific Programming*, 2021:4055281, 2021. ISSN 1058-9244. doi: 10.1155/2021/4055281. URL <https://doi.org/10.1155/2021/4055281>.
- S. Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. 04 1991.
- S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780, 11 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL <https://doi.org/10.1162/neco.1997.9.8.1735>.
- Z. Hu, Y. Zhao, and M. Khushi. A survey of forex and stock price prediction using deep learning. *Applied System Innovation*, 4(1), 2021. ISSN 2571-5577. doi: 10.3390/asi4010009. URL <https://www.mdpi.com/2571-5577/4/1/9>.
- W. H. Kruskal and W. A. Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621, 1952. doi: 10.1080/01621459.1952.

10483441. URL <https://www.tandfonline.com/doi/abs/10.1080/01621459.1952.10483441>.
- H. B. Mann and D. R. Whitney. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics*, 18(1): 50 – 60, 1947. doi: 10.1214/aoms/1177730491. URL <https://doi.org/10.1214/aoms/1177730491>.
- K. Pearson. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, 1900. doi: 10.1080/14786440009463897. URL <https://doi.org/10.1080/14786440009463897>.
- M. Schuster and K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997. doi: 10.1109/78.650093.
- G. X. Silva, J. D. Amaral, T. Langlois, and L. B. Almeida. Faster training of recurrent networks. In F. L. Silva, J. C. Principe, and L. B. Almeida, editors, *Spatiotemporal models in biological and artificial systems*, volume 37 of *Frontiers in Artificial Intelligence and Applications*, pages 168–175. IOS Press, Amsterdam, Netherlands, 1996.
- C. A. Sims. Macroeconomics and reality. *Econometrica*, 48(1):1–48, 1980. doi: 10.2307/1912017.
- C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904. ISSN 00029556. URL <http://www.jstor.org/stable/1412159>.
- G. Szekely, M. Rizzo, and N. Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35, 04 2008. doi: 10.1214/009053607000000505.

- T. Toharudin, R. Pontoh, R. Caraka, S. Zahroh, Y. Lee, and R.-C. Chen. Employing long short-term memory and facebook prophet model in air temperature forecasting. *Communication in Statistics- Simulation and Computation*, 01 2021. doi: 10.1080/03610918.2020.1854302.
- A. S. Wilkins. To lag or not to lag?: Re-evaluating the use of lagged dependent variables in regression analysis. *Political Science Research and Methods*, 6(2):393–411, 2018. doi: 10.1017/psrm.2017.4.
- M. Zhou, S. Ghareeb, Z. U. Shamszaman, and J. Mustafina. Enhancing stock price forecasting: Integrating supply chain factors into lstm models and comparative performance analysis. In *2023 16th International Conference on Developments in eSystems Engineering (DeSE)*, pages 253–257, 2023. doi: 10.1109/DeSE60595.2023.10469157.

Apéndice A

Amazon

- Correlaciones:

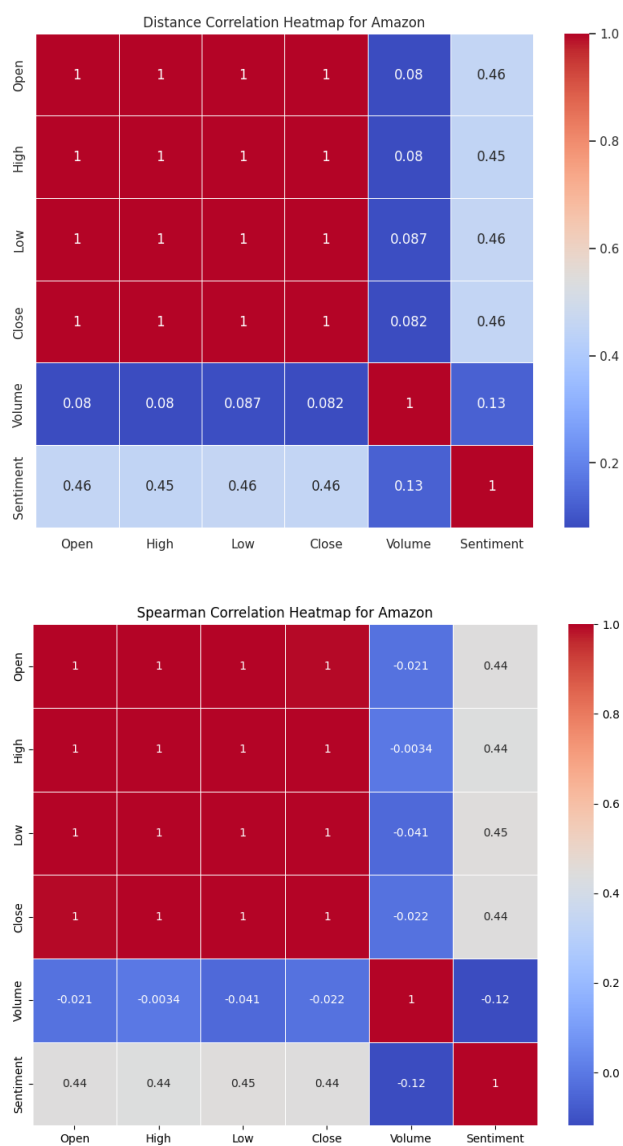


Figura A.1: *Distance correlation, Spearman correlation y Spearman P.Values heatmaps de Amazon*

- Correlaciones después de aplicar LAG:

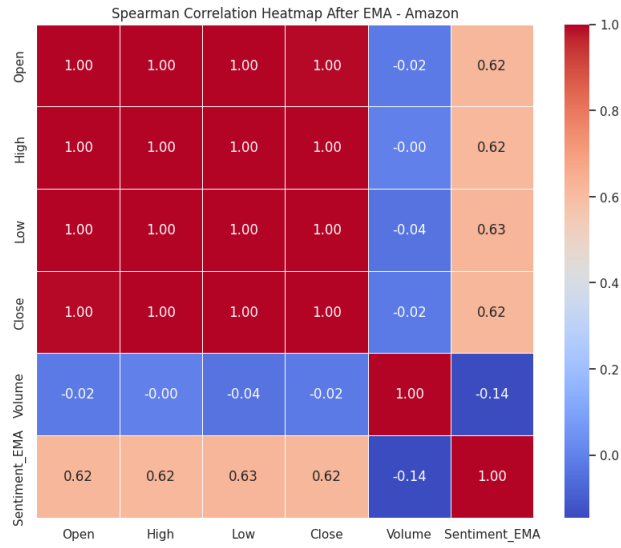


Figura A.2: *Spearman Correlation aplicando EMA con LAG=10*



Figura A.3: *Spearman Correlation aplicando Shift con LAG=10*

- Test de Granger:

Close - Open	Close - High	Close - Volume	Close - Sentiment_EMA
0.1009	0.9393	0.5517	0.5222

Tabla A.1: *P.valores test de Granger: Close - Variable*

Open - Close	High - Close	Volume - Close	Sentiment_EMA - Close
4.8229e-227	8.0499e-65	0.7369	1.4829e-23

Tabla A.2: *P.valores test de Granger: Variable - Close*

- Mejores hiperparametros obtenidos por *Grid Search*:

Modelo	units1	dropout1	units2	dropout2	learning_rate
LSTM	100	0.2	25	0.2	0.001
LSTM_s	100	0.2	25	0.1	0.001
GRU	150	0.1	50	0.1	0.001
GRU_s	150	0.2	75	0.1	0.0001

Tabla A.3: *Mejores hiperparámetros para modelos LSTM y GRU según mejores resultados en el Grid Search.*

- Métricas:

Modelo	LSTM	GRU	LSTM-News	GRU-News
MSE	0.000883961	0.00213406	0.000823402	0.000258577
R	0.969909	0.927354	0.97197	0.991198
MAPE	7.26022	11.5721	7.11672	5.44249

Figura A.4: *Métricas de Amazon para Epochs=64 & Batch Size=32*

- Gráficos función de perdida:

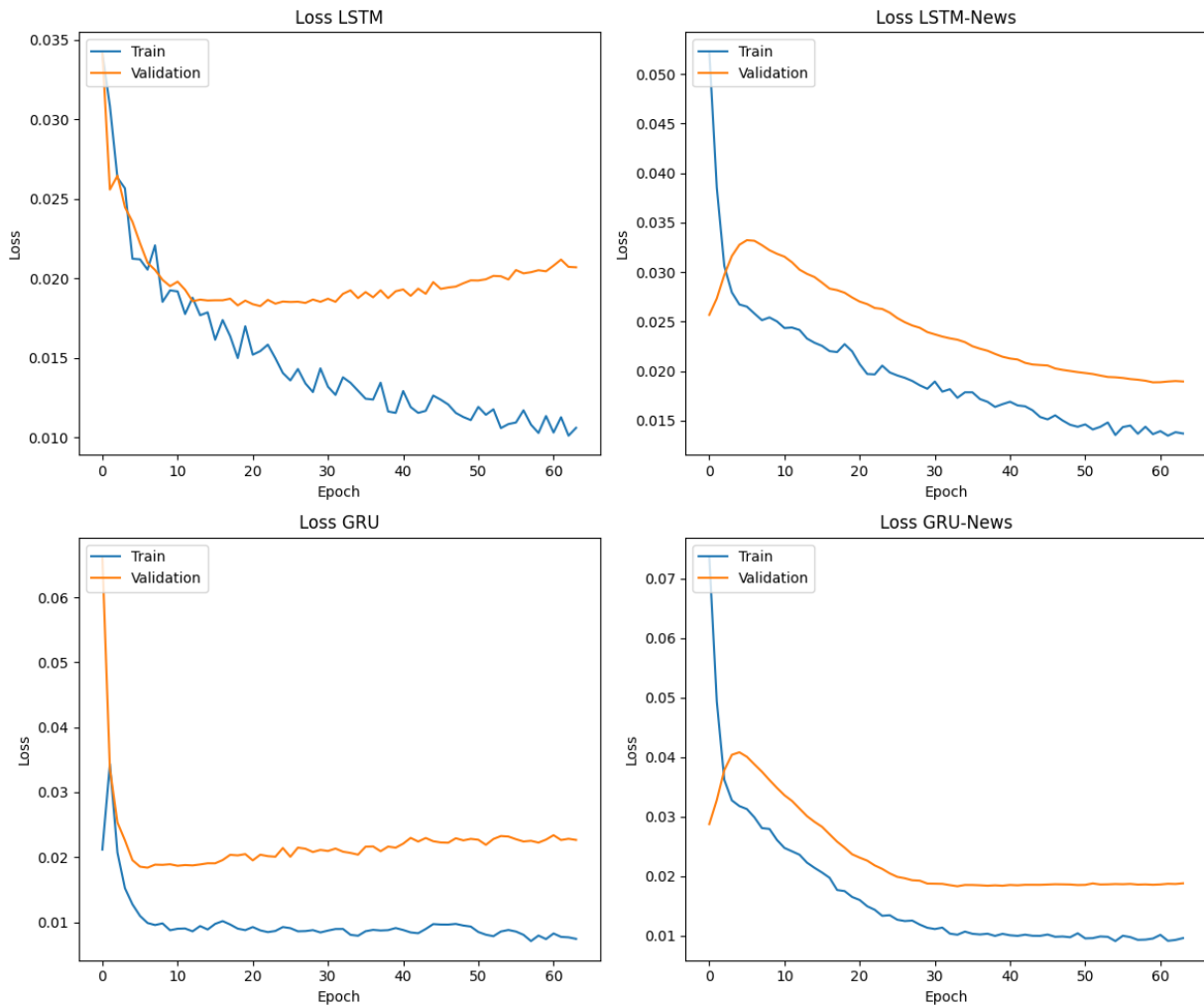


Figura A.5: *Function Losses de Amazon*

- Resultados gráficos:

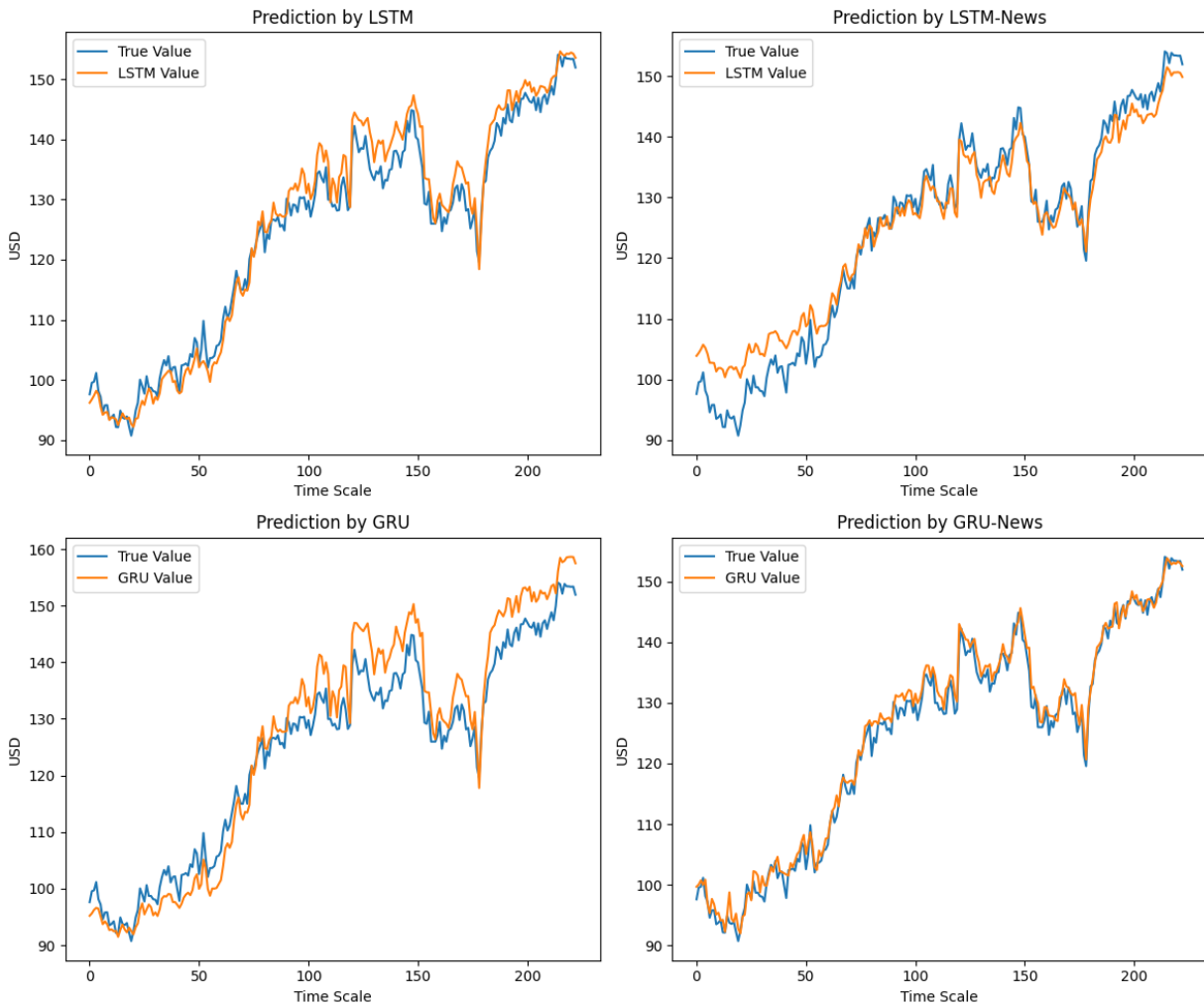


Figura A.6: *Amazon: Predictions vs real data*

- Graficos residuos

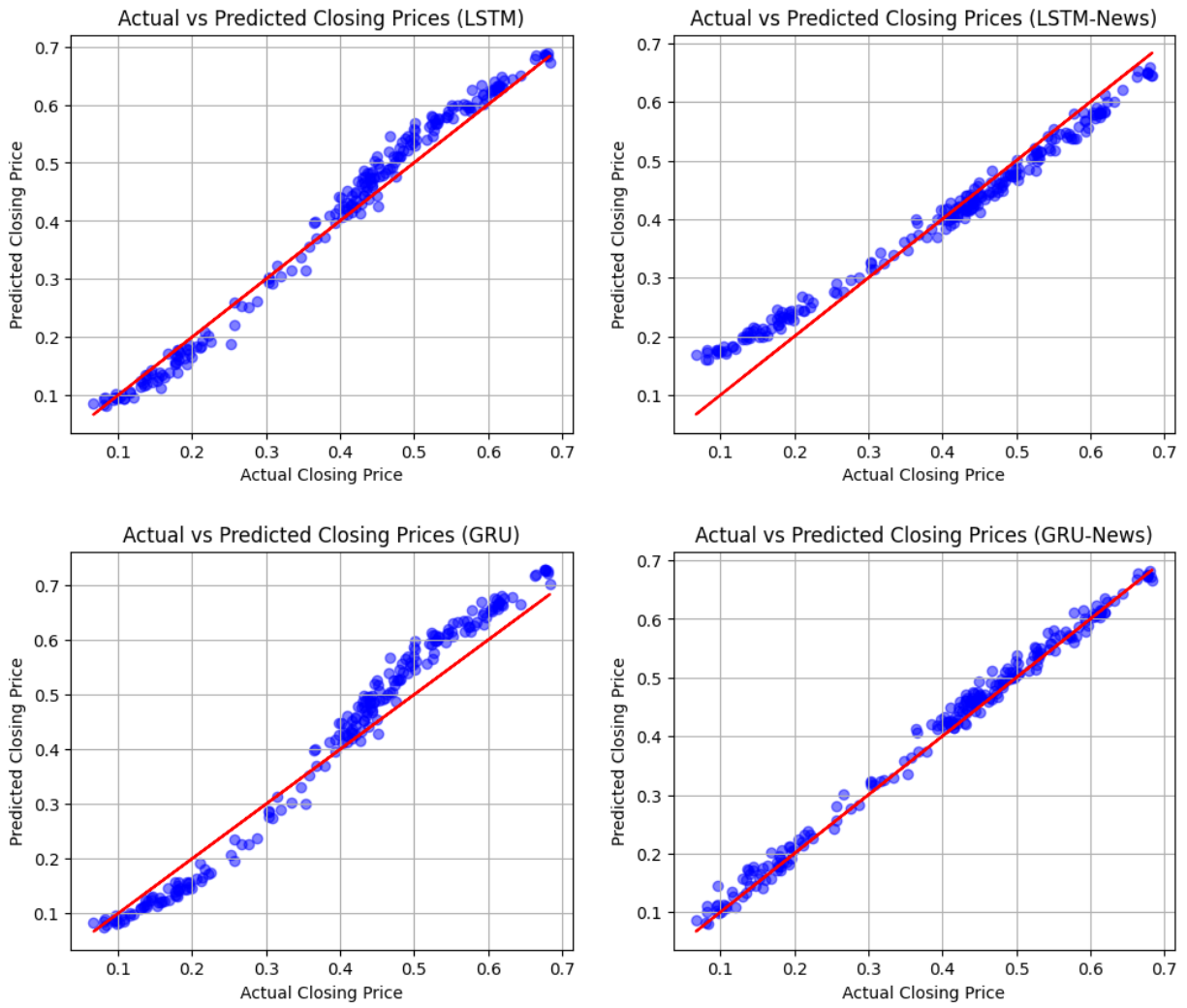


Figura A.7: *Resíduos de Amazon*

- Graficos QQ-Plots

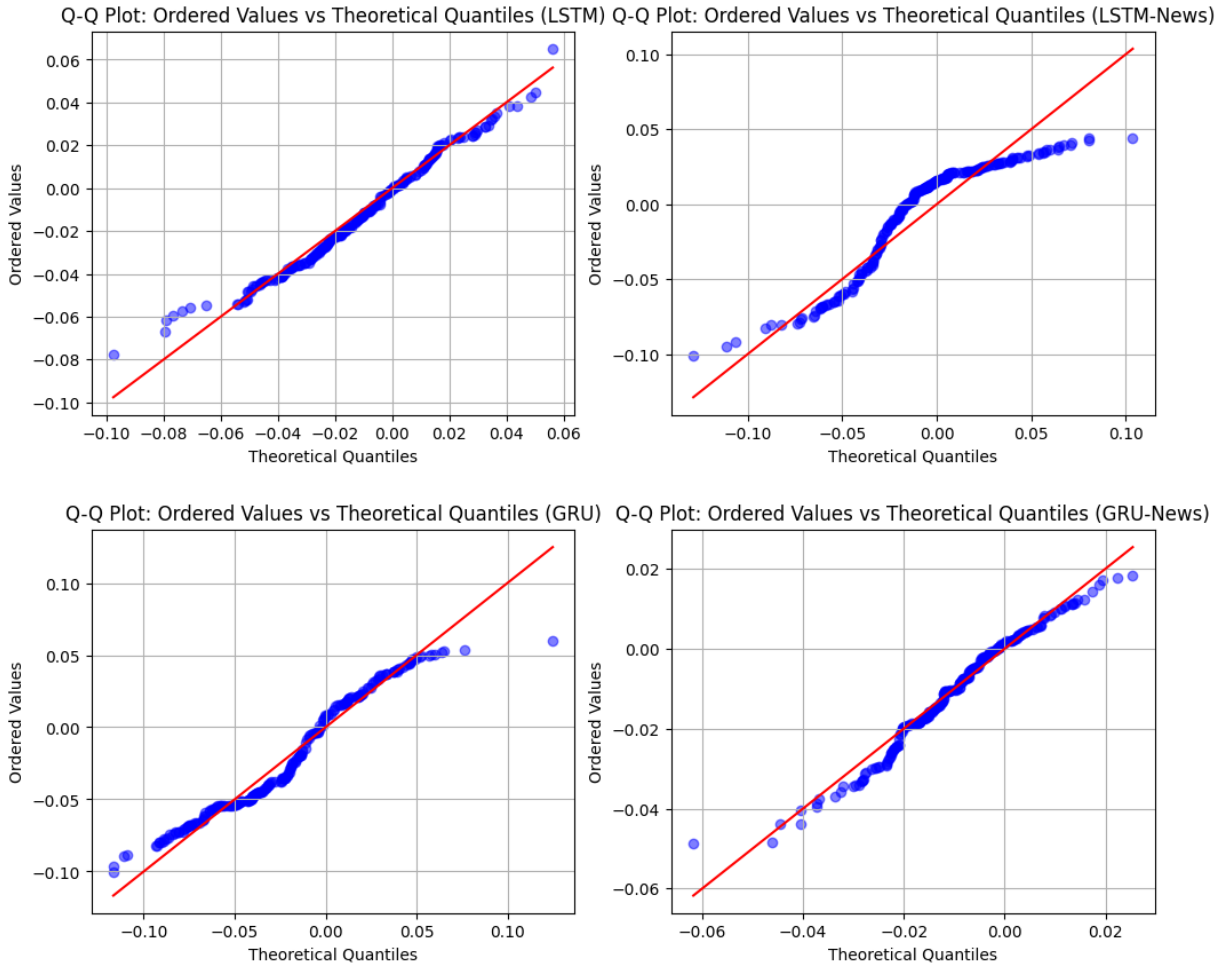


Figura A.8: *QQ-Plots de Amazon*

- Normalidad errores:

Modelo	Estadístico	P.Valor	Normal
LSTM	4.4204	0.1097	True
LSTM_s	2.0890	0.3519	True
GRU	101.2161	1.05e-22	False
GRU_s	4.0988	0.1288	True

Tabla A.4: *Resultados del Test de Normalidad D'Agostino*

- Kruskal Wallis:

Estadístico de Prueba	P.Valor
43.86205482314381	1.6145198793729251e-09

Tabla A.5: *Resultados del test de Kruskal-Wallis para Amazon*

- Mann-Whitney:

Comparación	Estadístico de Prueba	P.Valor	Hipótesis
LSTM vs LSTM_s	25003.0	0.919238	No Rechazada
LSTM vs GRU	27753.0	0.033844	Rechazada
LSTM vs GRU_s	17583.0	8.813852e-08	Rechazada
LSTM_s vs GRU	27713.0	0.036391	Rechazada
LSTM_s vs GRU_s	16802.0	3.152549e-09	Rechazada
GRU vs GRU_s	18726.0	6.483869e-06	Rechazada

Tabla A.6: *Resultados del test de Mann-Whitney para Amazon*

Apéndice B

Apple

- Correlaciones:

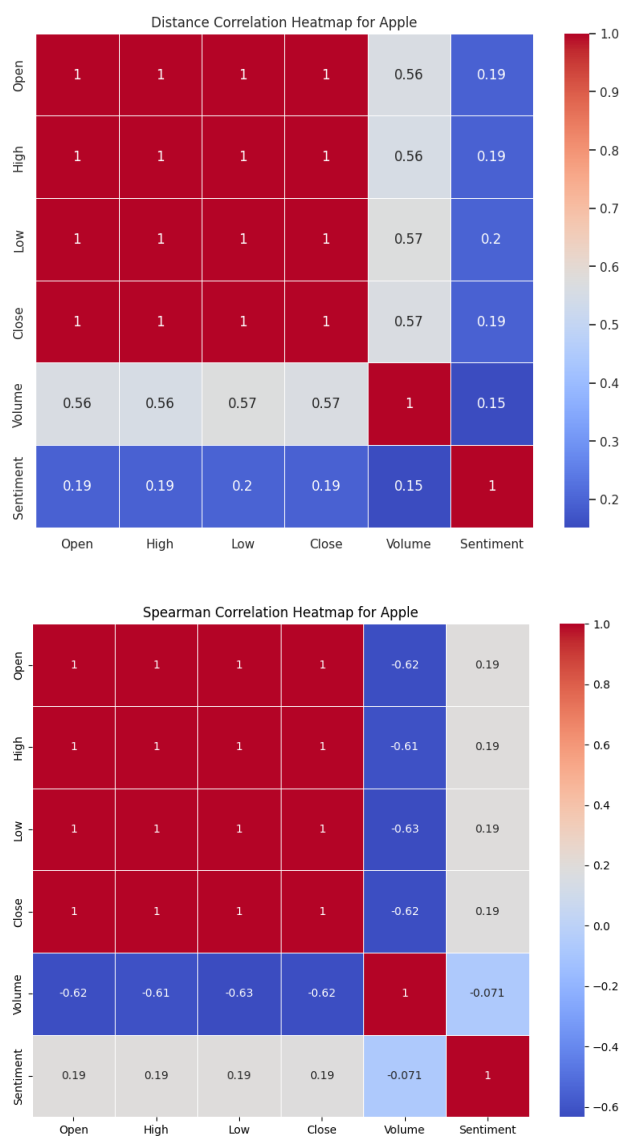


Figura B.1: *Distance correlation y Spearman correlation de Apple*

- Correlaciones después de aplicar LAG:

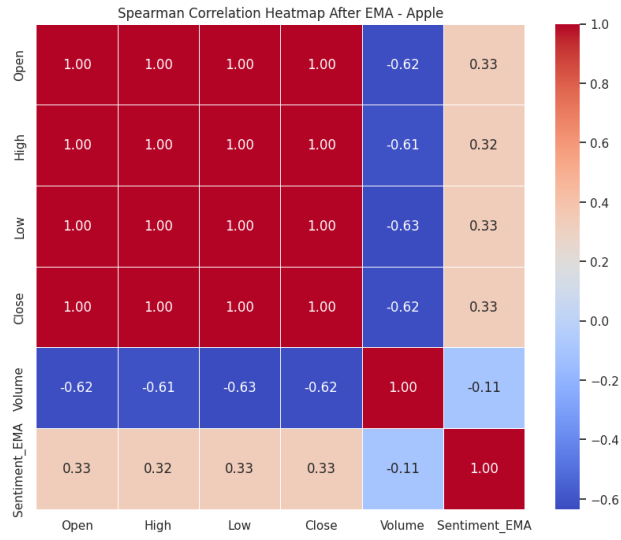


Figura B.2: *Spearman Correlation aplicando EMA con LAG=9*



Figura B.3: *Spearman Correlation aplicando Shift con LAG=9*

- Test de Granger:

Close - Open	Close - High	Close - Volume	Close - Sentiment_ EMA
0.6598	0.4427	0.1907	0.4318

Tabla B.1: *P.valores test de Granger: Close - Variable*

Open - Close	High - Close	Volume - Close	Sentiment_EMA - Close
4.2756e-266	6.6807e-47	0.1244	4.3559e-26

Tabla B.2: *P.valores test de Granger: Variable - Close*

- Mejores hiperparametros obtenidos por *Grid Search*:

Modelo	units1	dropout1	units2	dropout2	learning_rate
LSTM	100	0.1	50	0.1	0.001
LSTM_s	125	0.1	50	0.2	0.001
GRU	100	0.1	50	0.1	0.001
GRU_s	100	0.2	25	0.1	0.001

Tabla B.3: *Mejores hiperparámetros para modelos LSTM y GRU según mejores resultados en el Grid Search.*

- Métricas :

Modelo	LSTM	GRU	LSTM-News	GRU-News
MSE	0.000155104	0.000557639	0.000352629	0.000458686
R	0.98464	0.944776	0.965878	0.954575
MAPE	1.03749	2.2102	1.72798	1.8921

Figura B.4: *Métricas de Apple para Epochs=64 & Batch Size=32*

- Gráficos función de perdida:

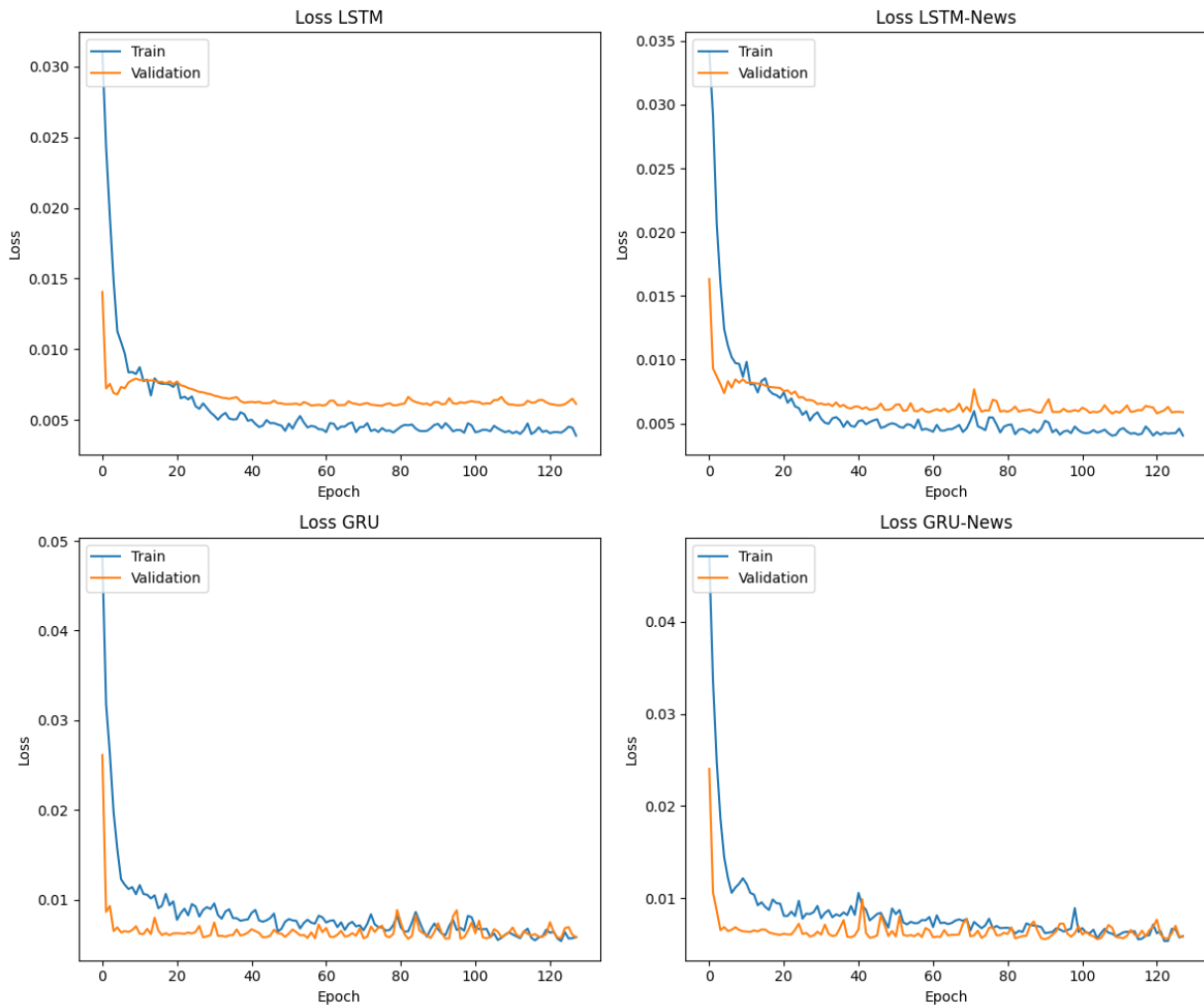


Figura B.5: *Function Losses de Apple*

- Resultados gráficos:

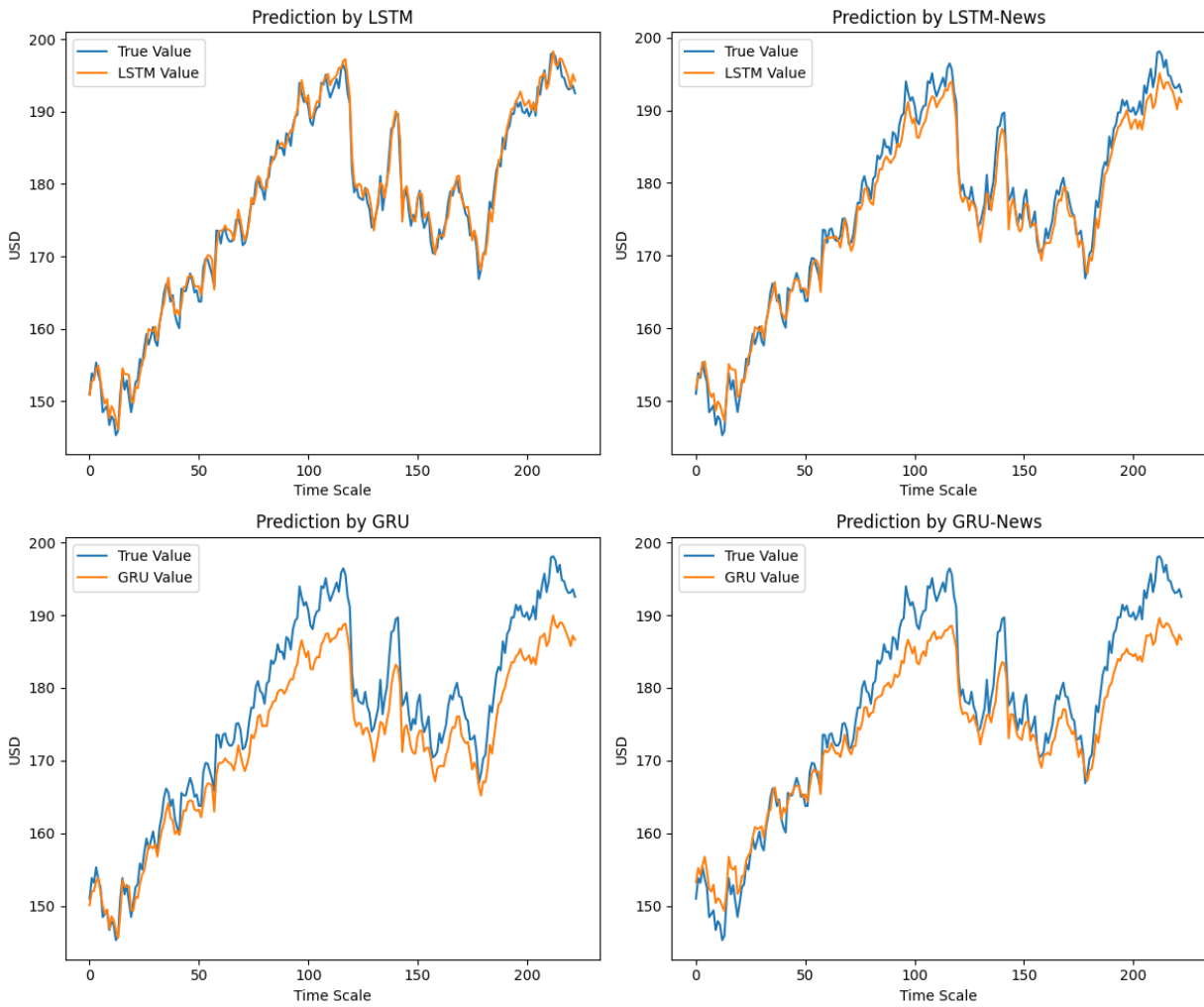


Figura B.6: *Apple: Predictions vs real data*

- Graficos residuos

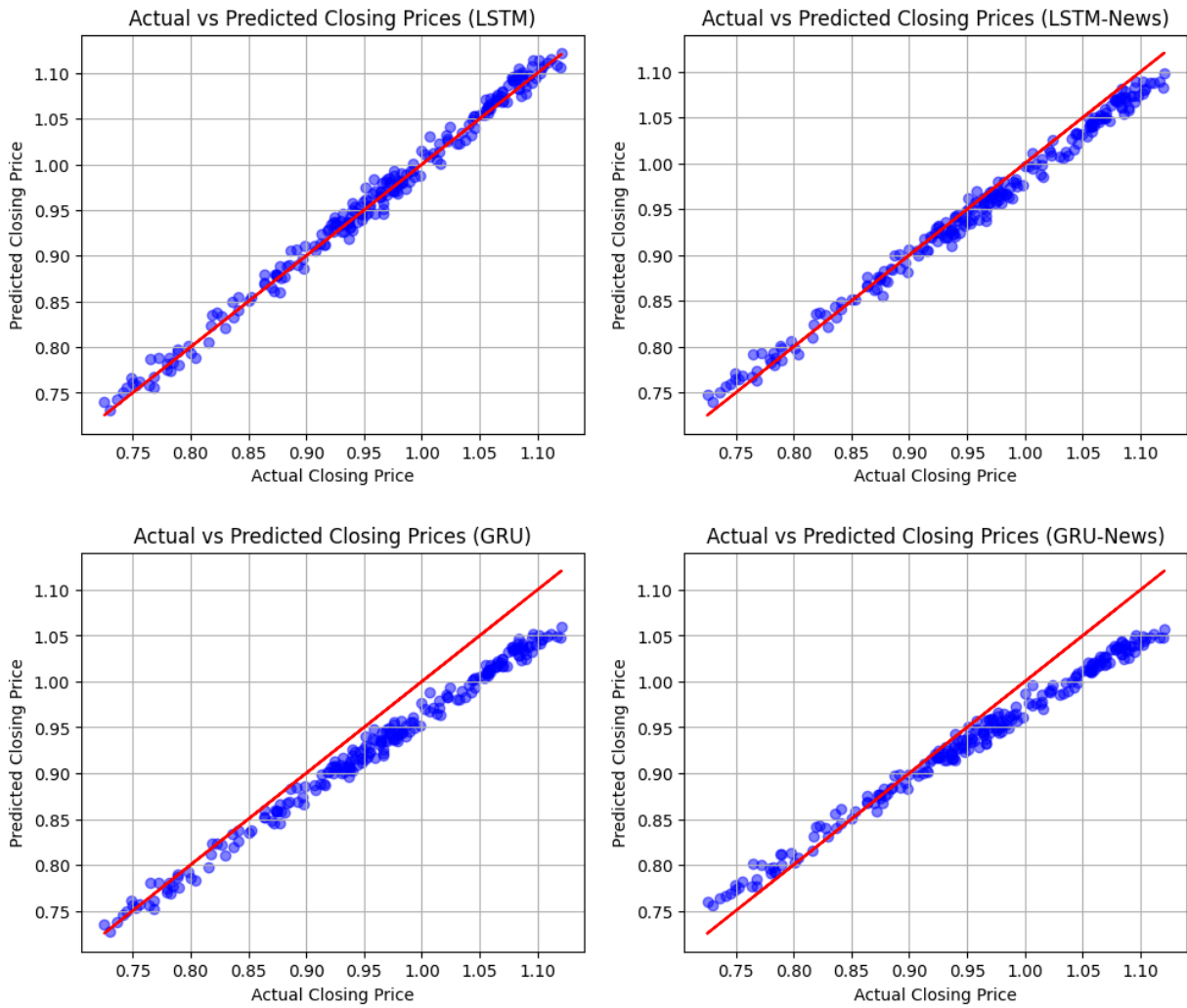


Figura B.7: *Residuos de Apple*

- Graficos QQ-Plots

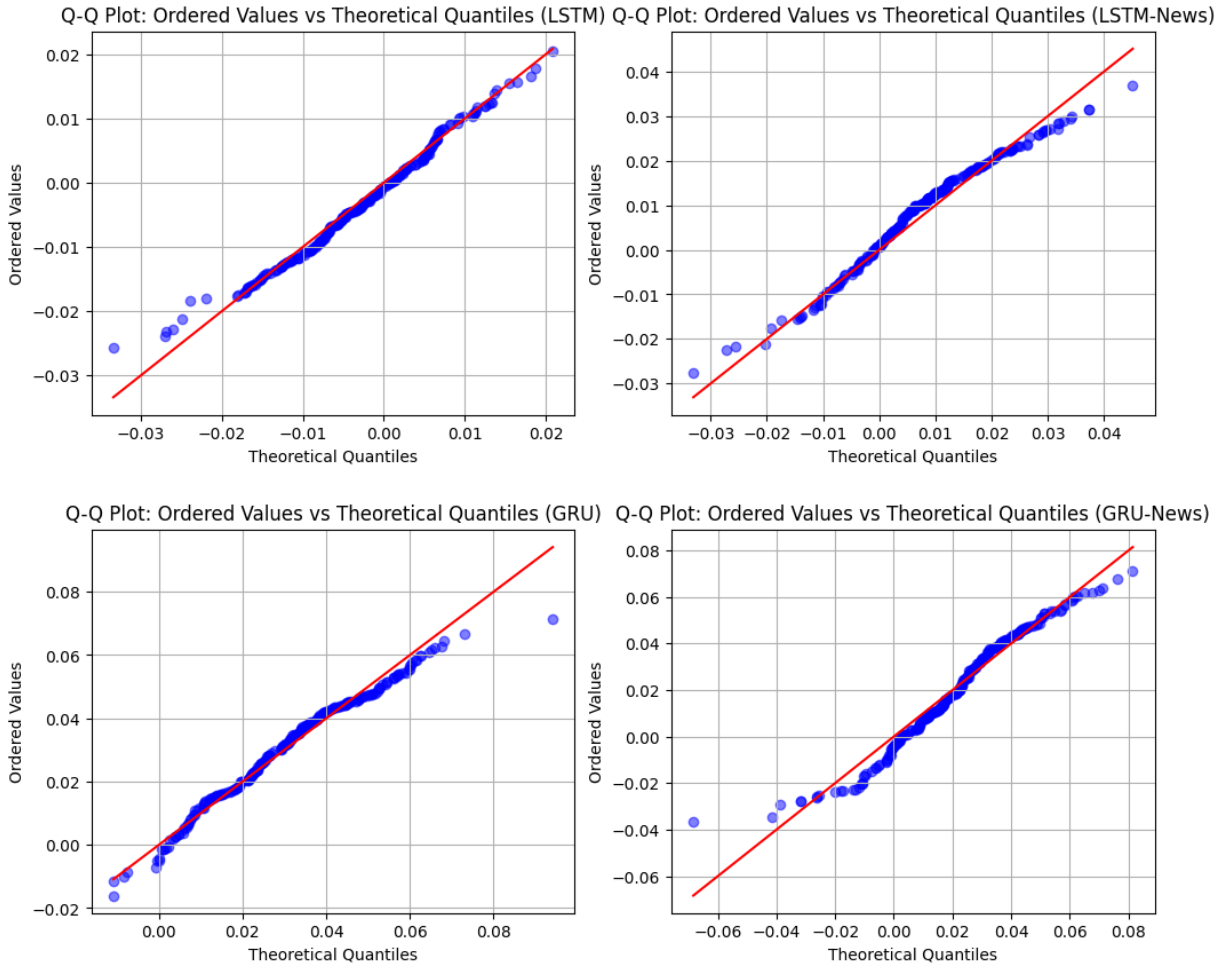


Figura B.8: *QQ-Plots de Apple*

- Normalidad errores:

Modelo	Estadístico	P.Valor	Normal
LSTM	3.5774	0.1671	True
LSTM_s	6.0091	0.0495	False
GRU	3.6143	0.1641	True
GRU_s	3.0449	0.2181	True

Tabla B.4: *Resultados del Test de Normalidad D'Agostino*

- Kruskal Wallis:

Estadístico de Prueba	P.Valor
435.4255771319745	4.6873599448819303e-94

Tabla B.5: *Resultados del test de Kruskal-Wallis para Apple*

- Mann-Whitney:

Comparación	Estadístico de Prueba	P.Valor	Hipótesis
LSTM vs LSTM_s	3379.0	3.895066575702911e-56	Rechazada
LSTM vs GRU	987.0	6.69340412136454e-69	Rechazada
LSTM vs GRU_s	2600.0	3.802762038373065e-60	Rechazada
LSTM_s vs GRU	16823.0	3.4624102772053846e-09	Rechazada
LSTM_s vs GRU_s	21322.0	0.00925632857413608	Rechazada
GRU vs GRU_s	29150.0	0.001641957088121865	Rechazada

Tabla B.6: *Resultados del test de Mann-Whitney para Apple*

Apéndice C

Facebook

- Correlaciones:

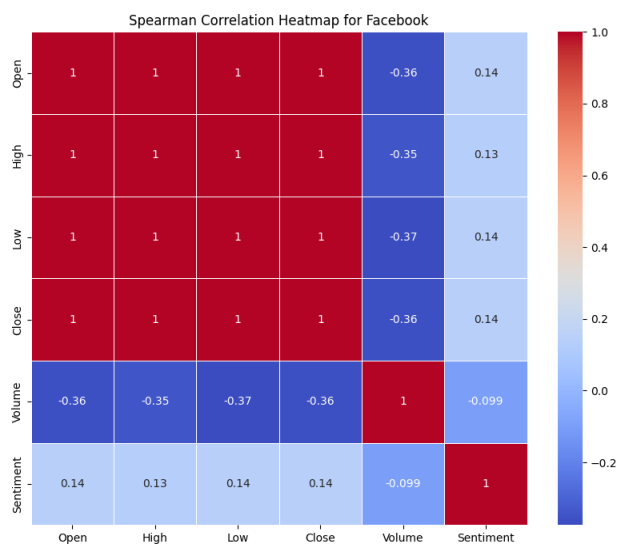
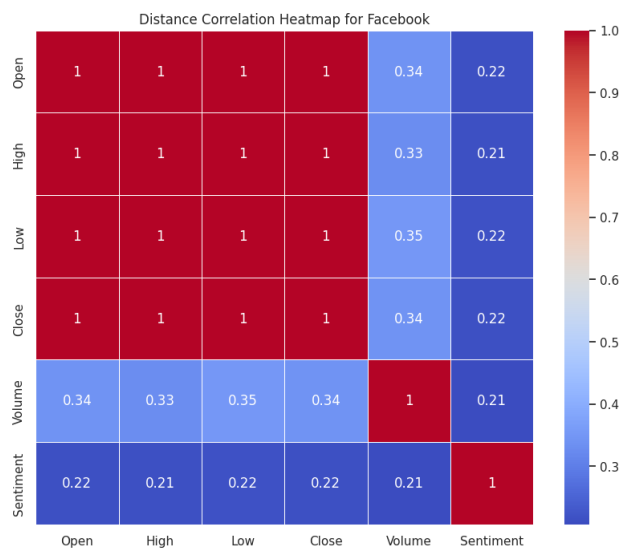


Figura C.1: *Distance correlation y Spearman correlation de Facebook*

- Correlaciones después de aplicar LAG:

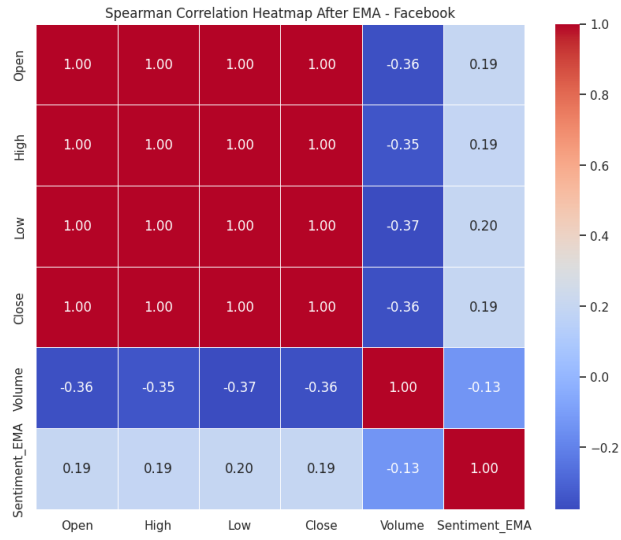


Figura C.2: *Spearman Correlation aplicando EMA con LAG=2*



Figura C.3: *Spearman Correlation aplicando Shift con LAG=2*

- Test de Granger:

Close - Open	Close - High	Close - Volume	Close - Sentiment_ EMA
0.4917	0.8312	0.3183	0.5054

Tabla C.1: *P.valores test de Granger: Close - Variable*

Open - Close	High - Close	Volume - Close	Sentiment_EMA - Close
4.0347e-166	1.8976e-47	0.0007	0.0005

Tabla C.2: *P.valores test de Granger: Variable - Close*

- Mejores hiperparametros obtenidos por *Grid Search*:

Modelo	units1	dropout1	units2	dropout2	learning_rate
LSTM	150	0.2	75	0.1	0.0001
LSTM_s	150	0.2	75	0.1	0.0001
GRU	100	0.1	50	0.2	0.0001
GRU_s	100	0.1	50	0.2	0.0001

Tabla C.3: *Mejores hiperparámetros para modelos LSTM y GRU según mejores resultados en el Grid Search.*

- Métricas:

Modelo	LSTM	GRU	LSTM-News	GRU-News
MSE	0.00193473	0.00121918	0.00169094	0.00104087
R	0.959536	0.974502	0.964635	0.978231
MAPE	12.7091	9.31021	11.8716	6.97954

Figura C.4: *Métricas de Google para Epochs=64 & Batch Size=32*

- Gráficos función de perdida:

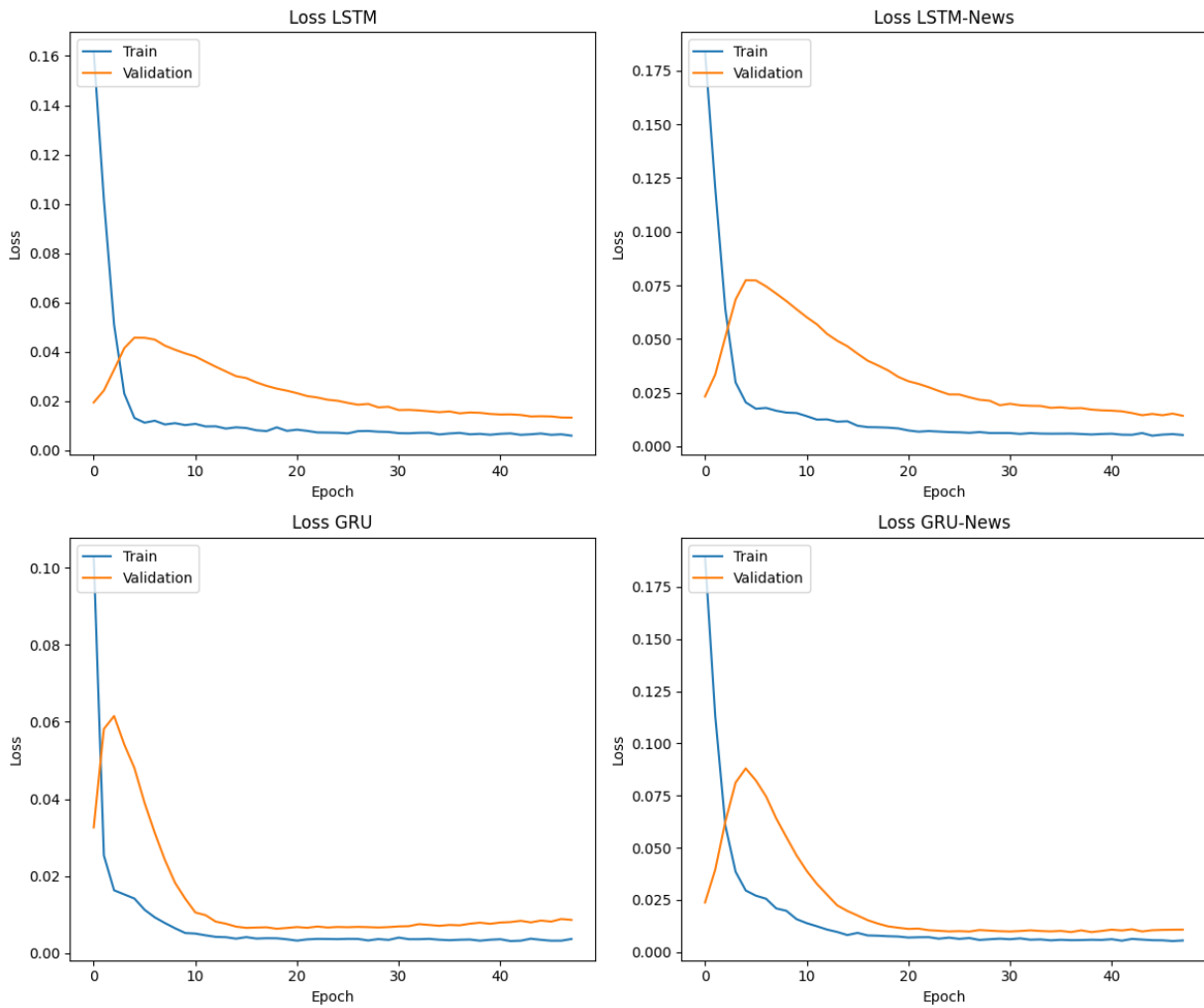


Figura C.5: *Function Losses de Facebook*

- Resultados gráficos:

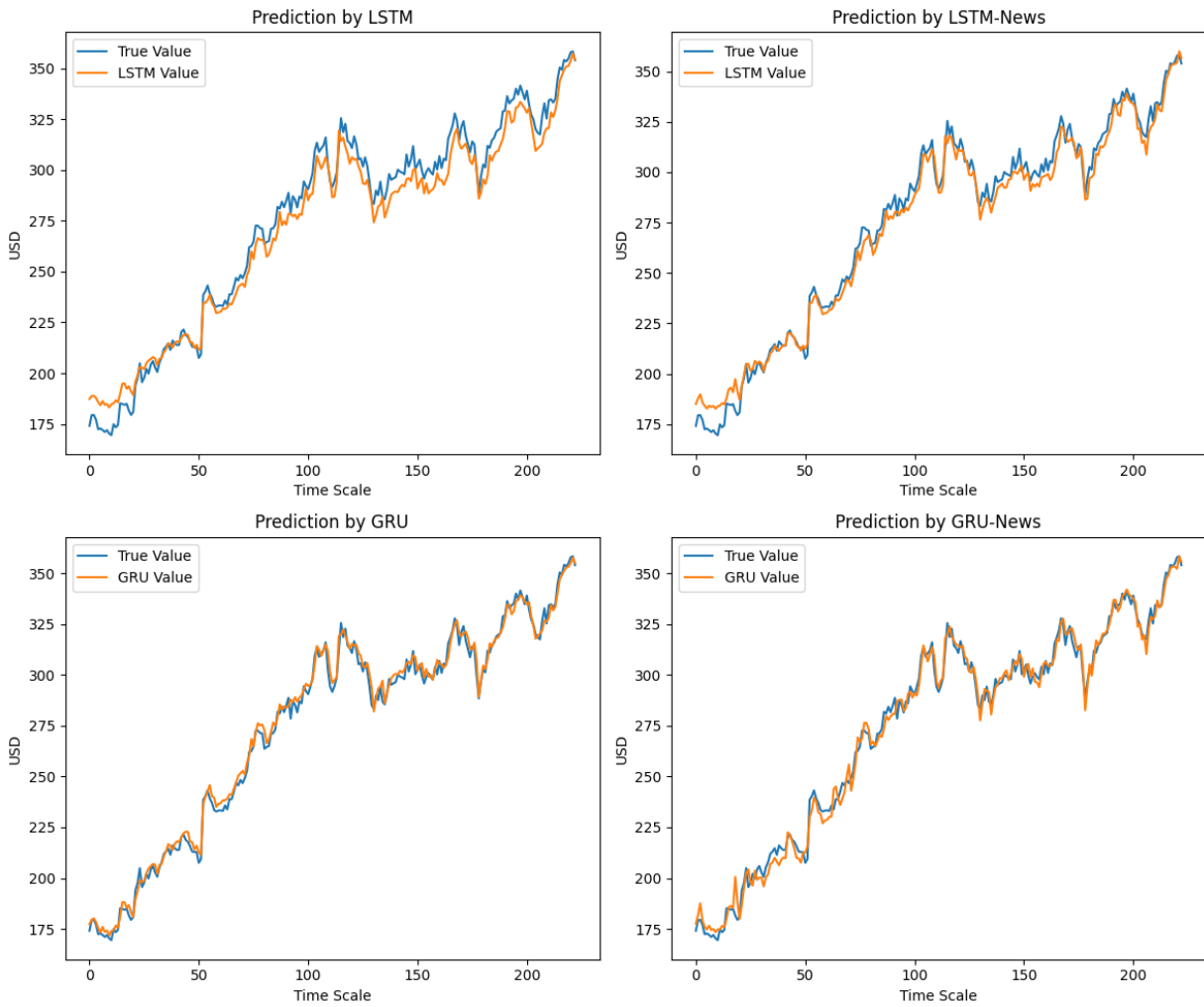


Figura C.6: *Facebook: Predictions vs real data*

- Graficos residuos

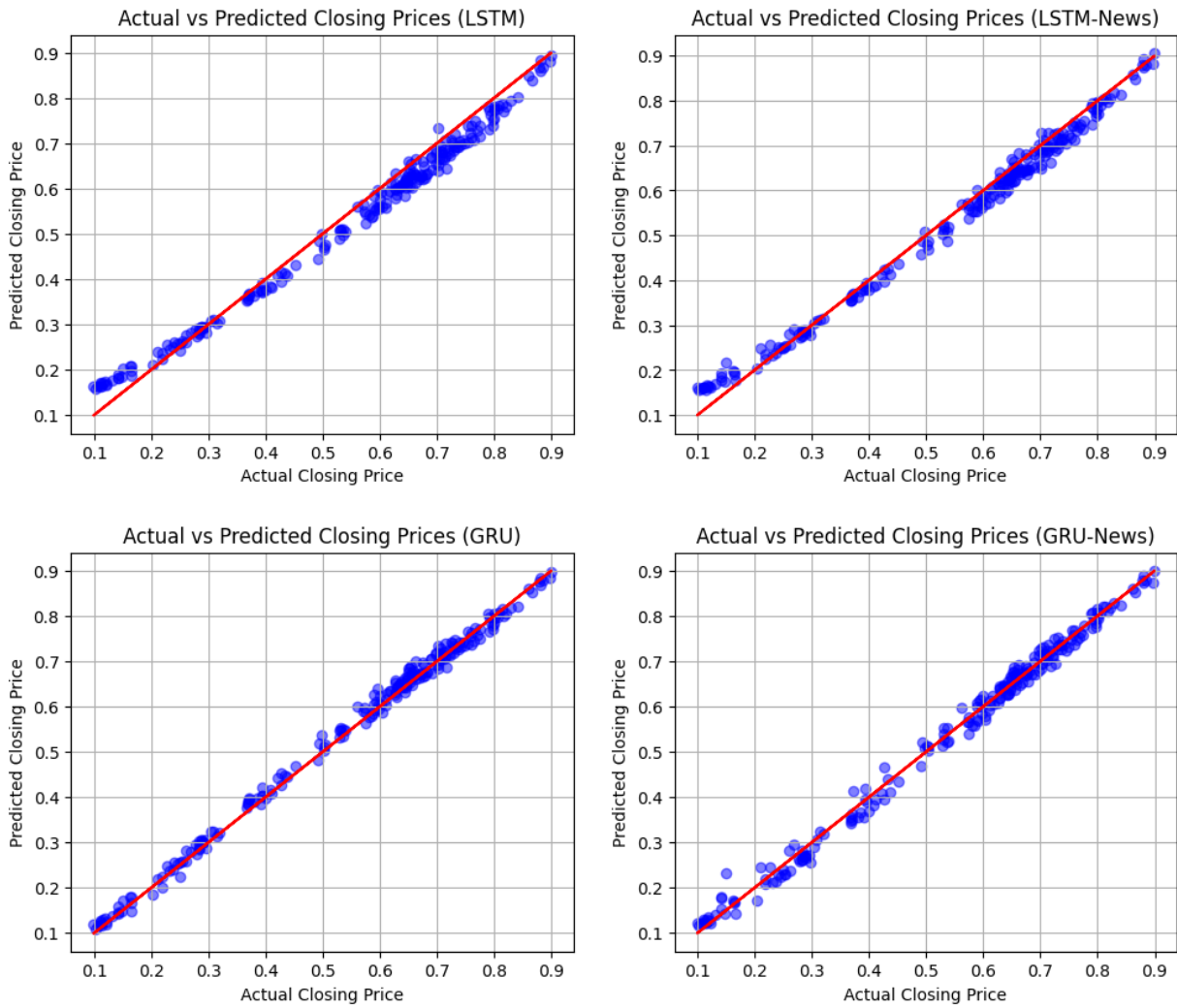


Figura C.7: *Residuos de Facebook*

- Graficos QQ-Plots

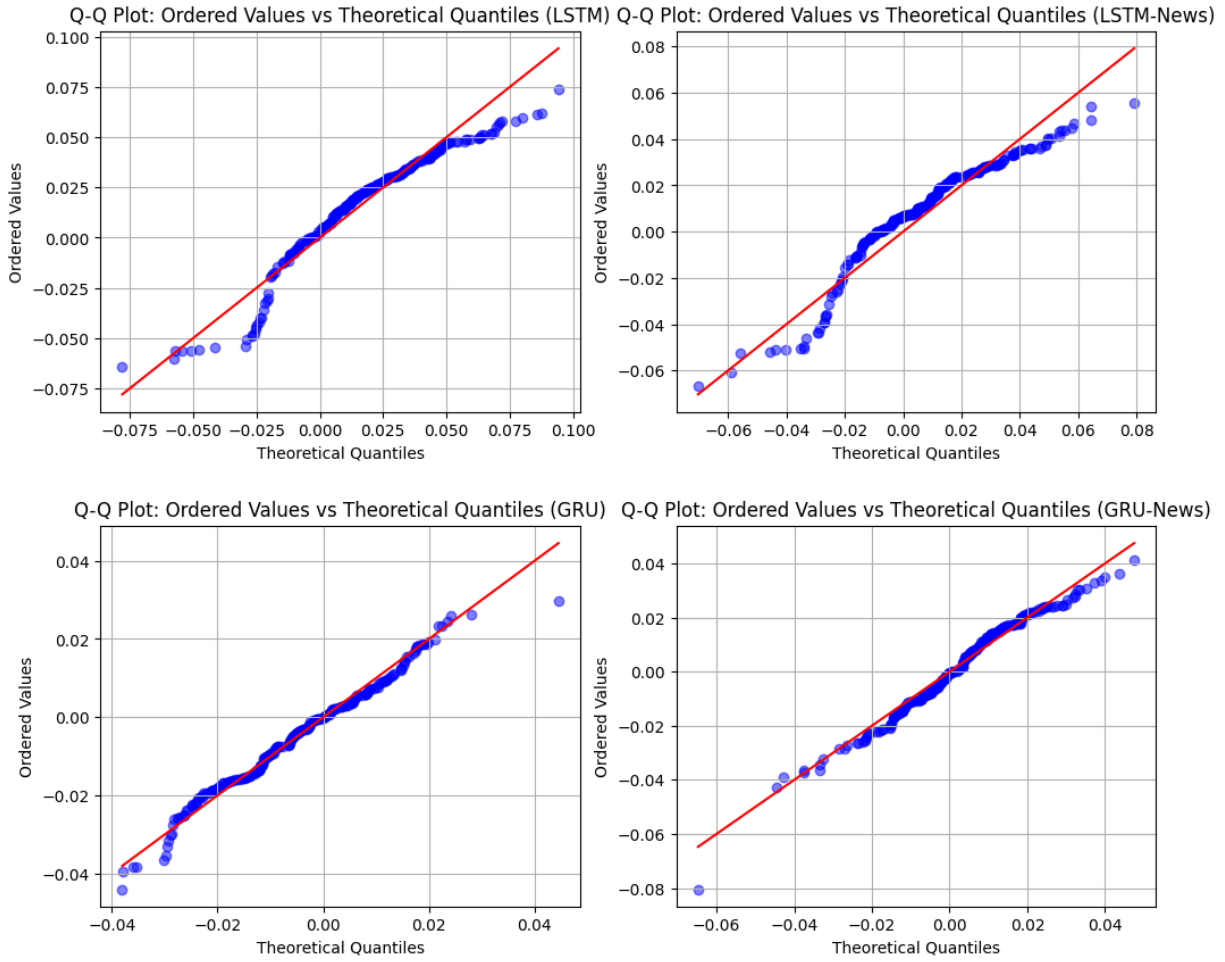


Figura C.8: *QQ-Plots de Facebook*

- Normalidad errores:

Modelo	Estadístico	P.Valor	Normal
LSTM	30.8688	1.981139634060871e-07	False
LSTM_s	16.9845	0.00020505532182131777	False
GRU	18.6757	8.80283350555443e-05	False
GRU_s	44.4349	2.24429804541108e-10	False

Tabla C.4: *Resultados del Test de Normalidad D'Agostino*

- Kruskal Wallis:

Estadístico de Prueba	P.Valor
34.826885414354365	1.3252807999100346e-07

Tabla C.5: *Resultados del test de Kruskal-Wallis para Facebook*

- Mann-Whitney:

Comparación	Estadístico de Prueba	P.Valor	Hipótesis
LSTM vs LSTM_s	28545.0	0.006854478928692411	Rechazada
LSTM vs GRU	24774.0	0.9477272377789048	No Rechazada
LSTM vs GRU_s	21109.0	0.005798953717154883	Rechazada
LSTM_s vs GRU	20826.0	0.00308494982898648	Rechazada
LSTM_s vs GRU_s	16591.0	1.2132678692547545e-09	Rechazada
GRU vs GRU_s	20777.0	0.0026744601112678986	Rechazada

Tabla C.6: *Resultados del test de Mann-Whitney para Facebook*

Apéndice D

Google

- Correlaciones:

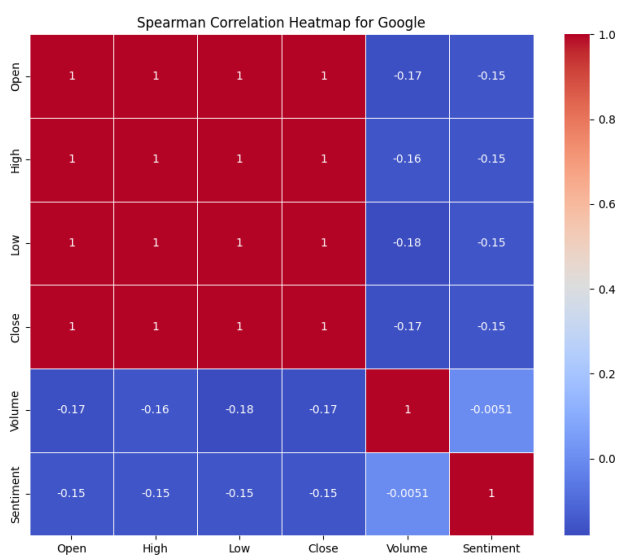
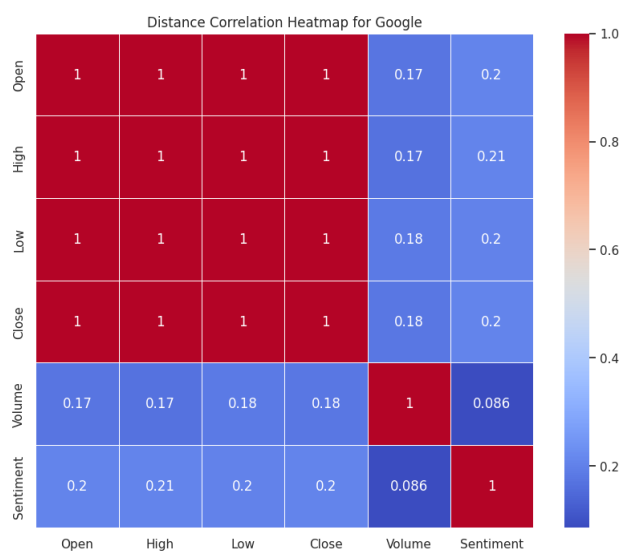


Figura D.1: *Distance correlation y Spearman correlation de Google*

- Correlaciones después de aplicar LAG:

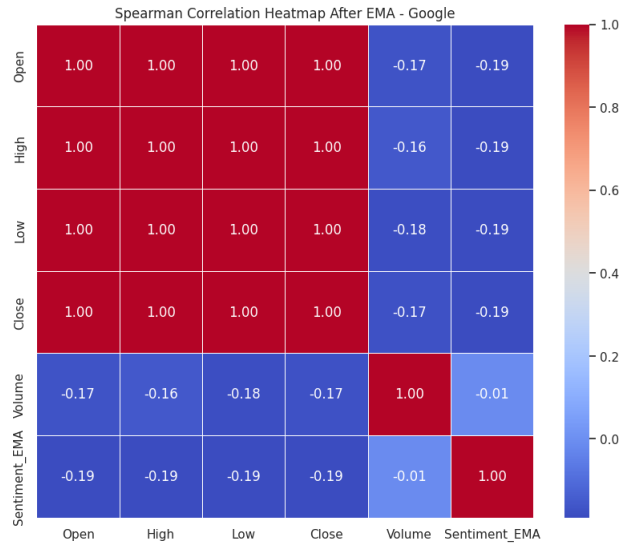


Figura D.2: *Spearman Correlation aplicando EMA con LAG=3*

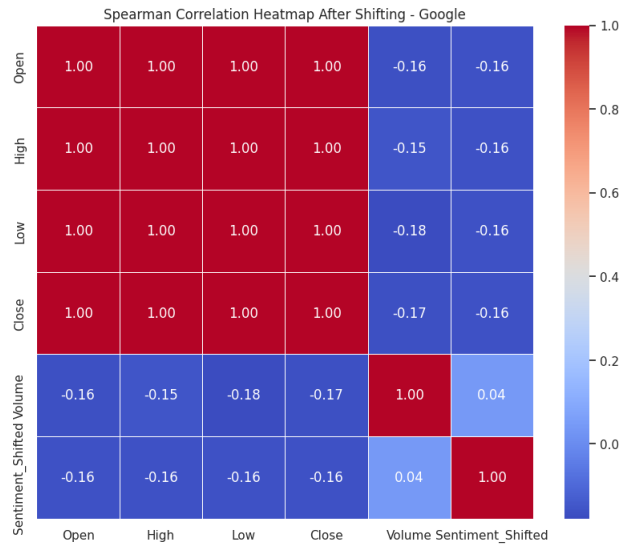


Figura D.3: *Spearman Correlation aplicando Shift con LAG=3*

- Test de Granger:

Close - Open	Close - High	Close - Volume	Close - Sentiment_EMA
0.1144	0.4288	0.6859	0.3335

Tabla D.1: *P.valores test de Granger: Close - Variable*

Open - Close	High - Close	Volume - Close	Sentiment_EMA - Close
2.9454e-197	2.6487e-51	0.0676	0.0049

Tabla D.2: *P.valores test de Granger: Variable - Close*

- Mejores hiperparametros obtenidos por *Grid Search*:

Modelo	units1	dropout1	units2	dropout2	learning_rate
LSTM	75	0.1	75	0.2	0.001
LSTM_s	100	0.1	50	0.1	0.001
GRU	75	0.2	50	0.2	0.001
GRU_s	75	0.2	50	0.2	0.001

Tabla D.3: *Mejores hiperparámetros para modelos LSTM y GRU según mejores resultados en el Grid Search.*

- Métricas:

Modelo	LSTM	GRU	LSTM-News	GRU-News
MSE	0.000567755	0.00233169	0.000383506	0.000466541
R	0.974955	0.897142	0.983082	0.979419
MAPE	3.05433	7.19984	2.36275	2.5581

Figura D.4: *Métricas de Google para Epochs=64 & Batch Size=32*

- Gráficos función de perdida:

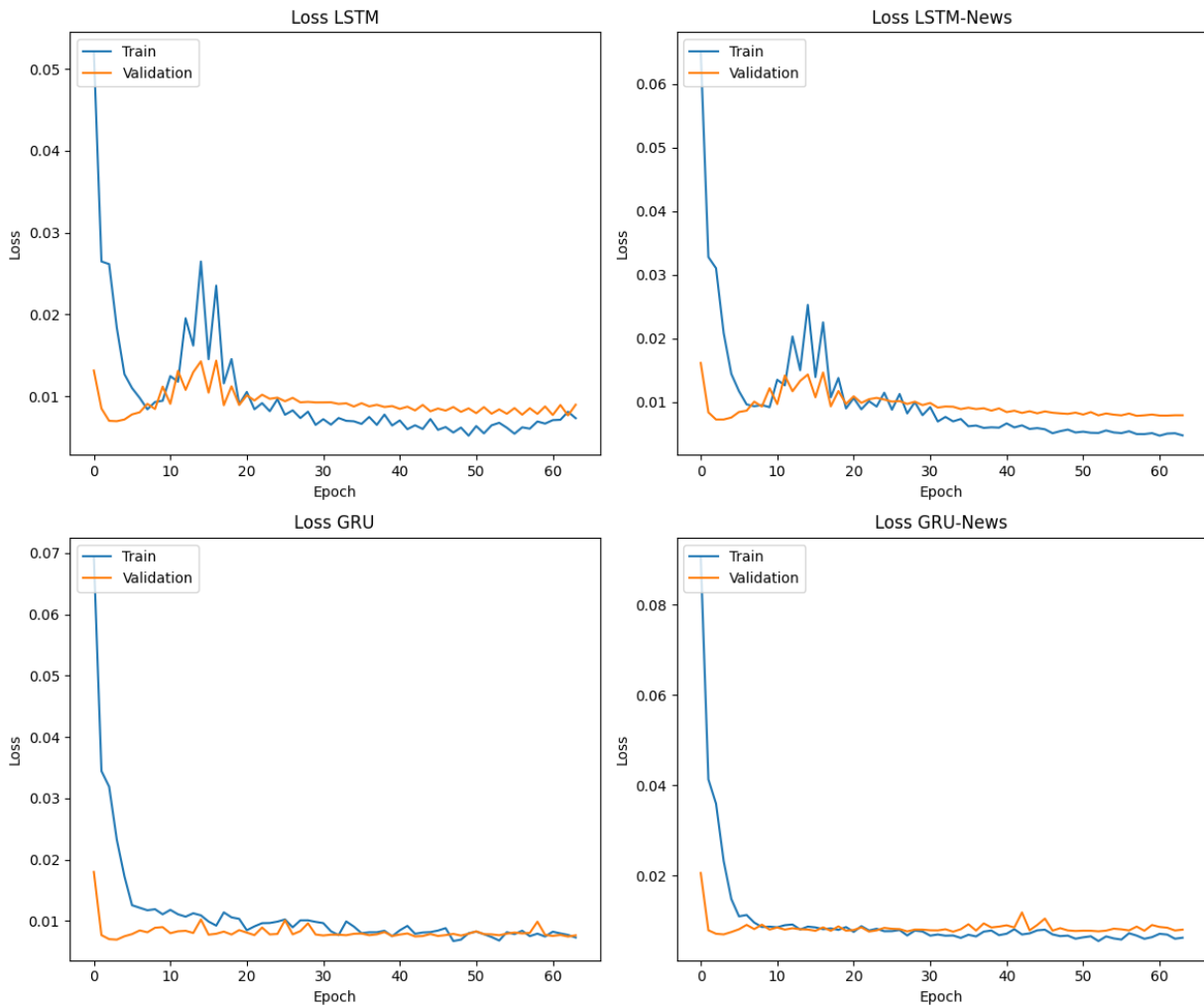


Figura D.5: *Function Losses de Google*

- Resultados gráficos:

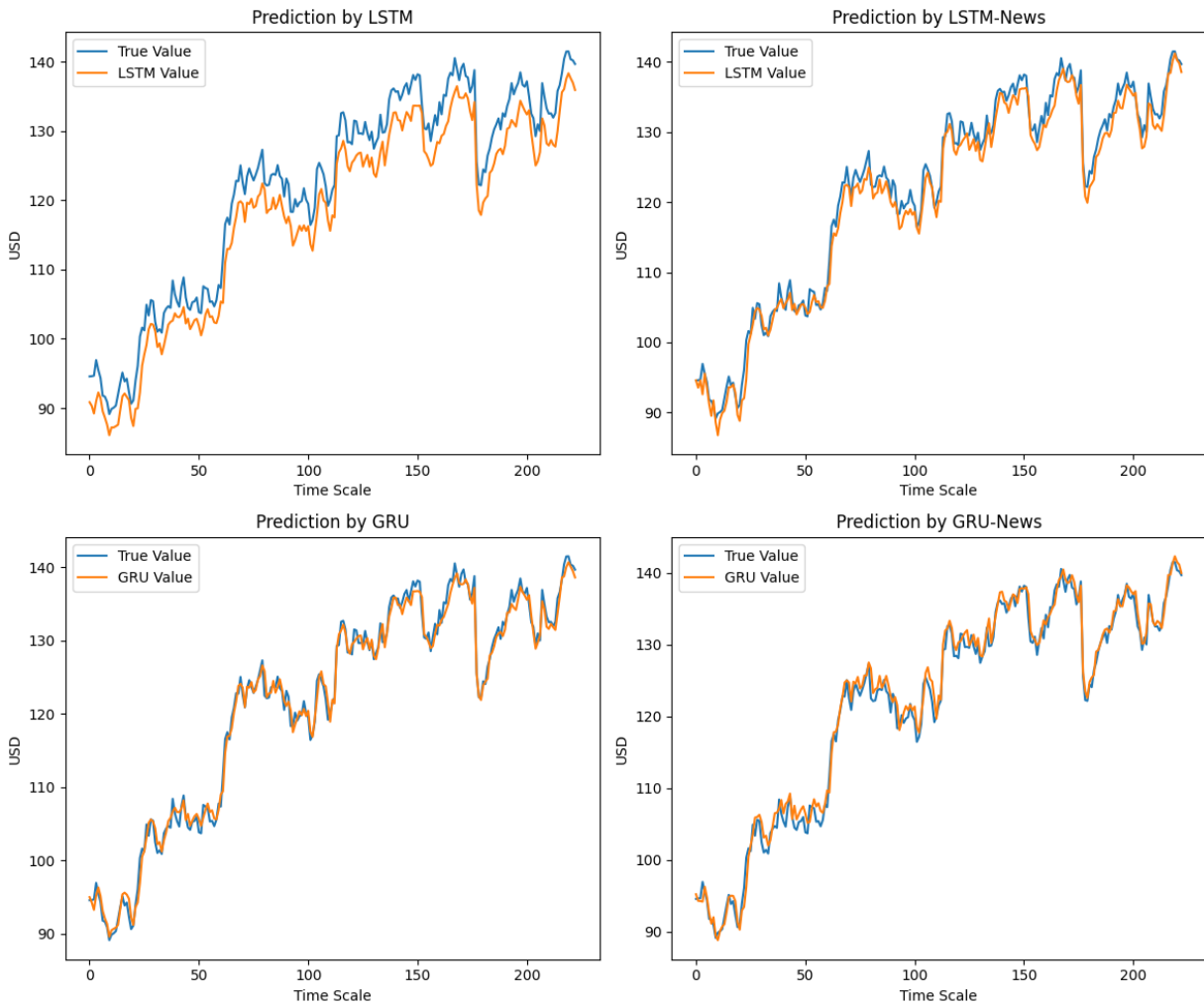


Figura D.6: *Google: Predictions vs real data*

- Graficos residuos

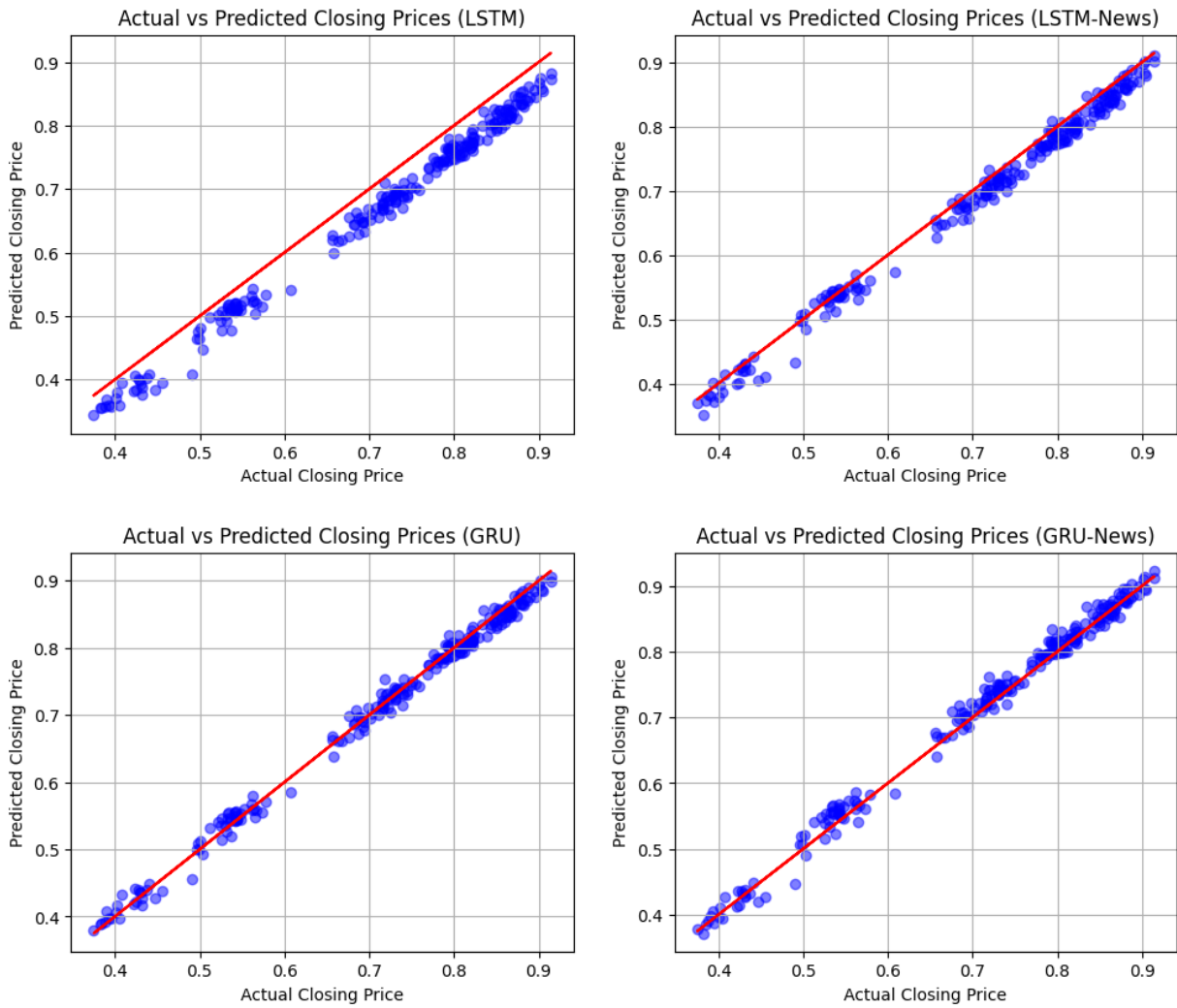


Figura D.7: *Resíduos de Google*

- Graficos QQ-Plots

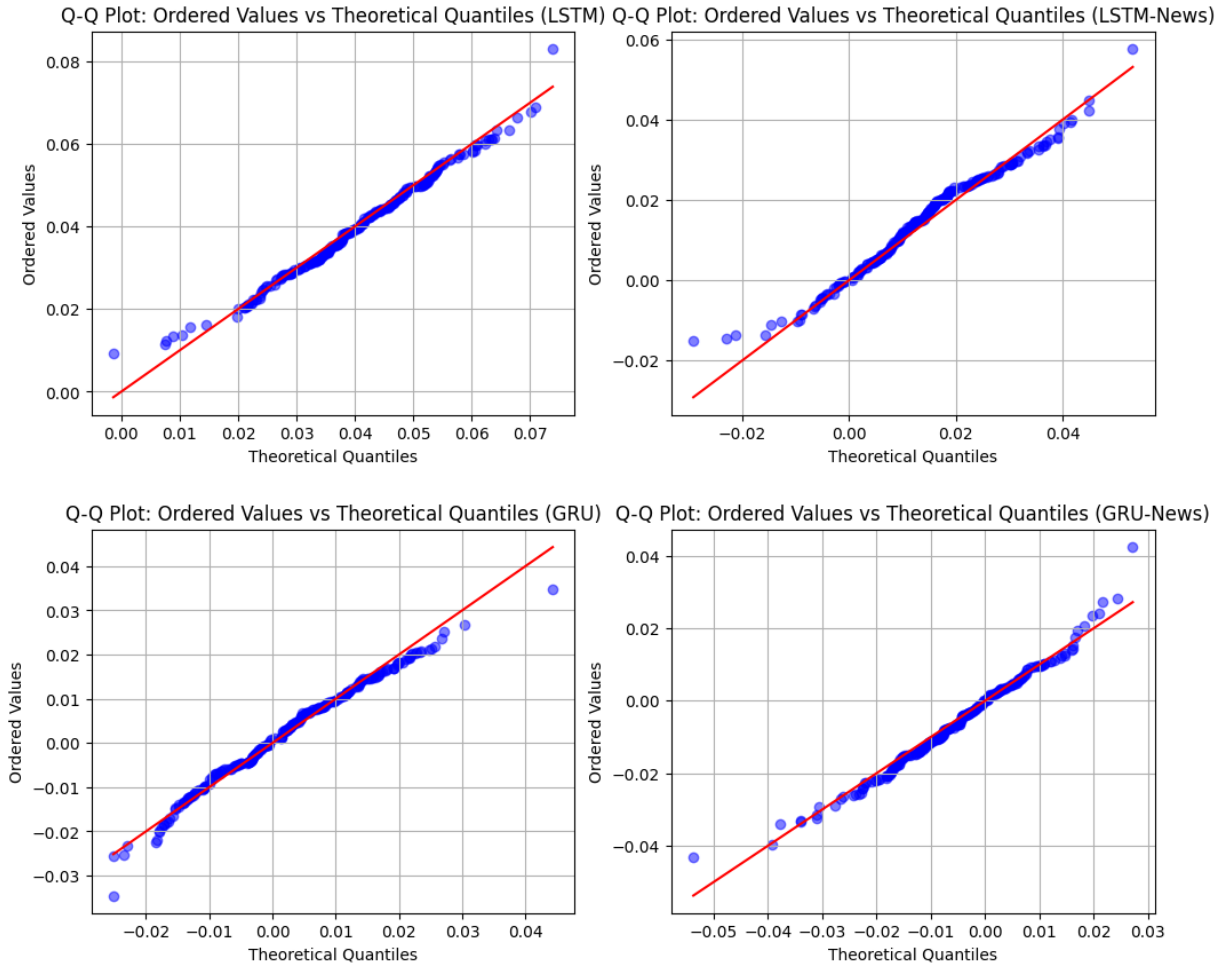


Figura D.8: *QQ-Plots de Google*

- Normalidad errores:

Modelo	Estadístico	P.Valor	Normal
LSTM	1.8747597143738908	0.3916526781047849	True
LSTM_s	3.612563039175625	0.16426381501751186	True
GRU	6.704346750439013	0.03500818543657584	False
GRU_s	1.867965346012248	0.39298545695304443	True

Tabla D.4: *Resultados del Test de Normalidad D'Agostino*

- Kruskal Wallis:

Estadístico de Prueba	P.Valor
513.6574838761585	5.2337683032818695e-111

Tabla D.5: *Resultados del test de Kruskal-Wallis para Google*

- Mann-Whitney:

Comparación	Estadístico de Prueba	P.Valor	Hipótesis
LSTM vs LSTM_s	31274.0	2.4902009635168333e-06	Rechazada
LSTM vs GRU	49729.0	1.4743617669914679e-74	Rechazada
LSTM vs GRU_s	28999.0	0.0023862589729372967	Rechazada
LSTM_s vs GRU	49729.0	1.4743617669914679e-74	Rechazada
LSTM_s vs GRU_s	22985.0	0.1674108737847806	No Rechazada
GRU vs GRU_s	49729.0	1.514596670853207e-74	Rechazada

Tabla D.6: *Resultados del test de Mann-Whitney para Google*

Apéndice E

Netflix

- Correlaciones:

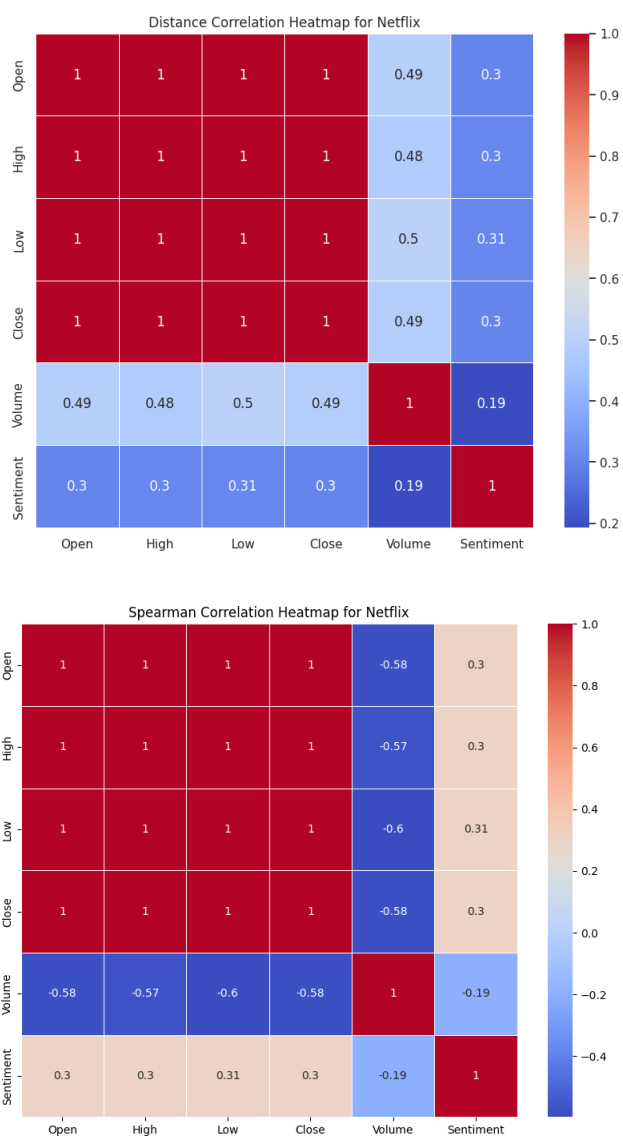


Figura E.1: *Distance correlation y Spearman correlation de Netflix*

- Correlaciones después de aplicar LAG:

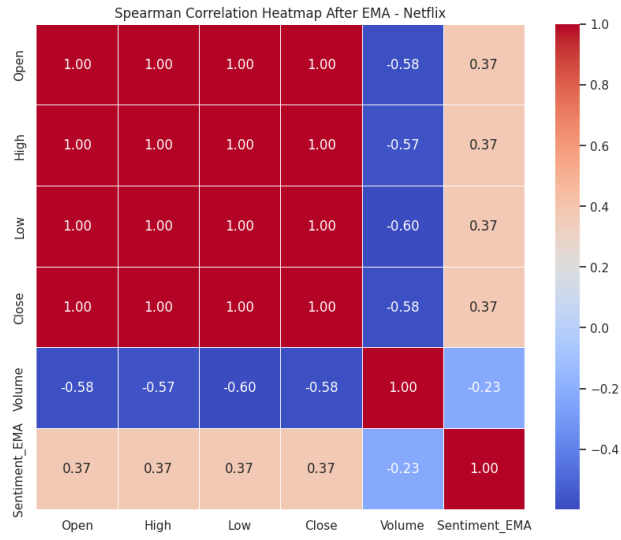


Figura E.2: *Spearman Correlation aplicando EMA con LAG=4*

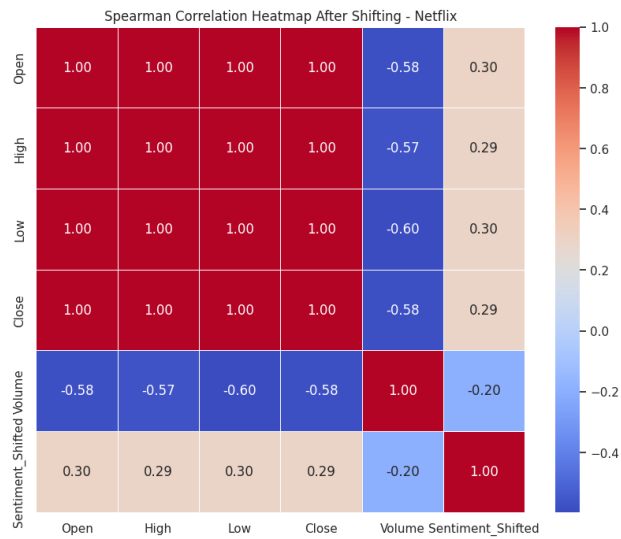


Figura E.3: *Spearman Correlation aplicando Shift con LAG=4*

- Test de Granger:

Close - Open	Close - High	Close - Volume	Close - Sentiment_ EMA
0.2850	0.6949	0.0409	0.7112

Tabla E.1: *P.valores test de Granger: Close - Variable*

	Open - Close	High - Close	Volume - Close	Sentiment_EMA - Close
1.2124e-191	1.2497e-51	4.8388e-05	1.7476e-13	

Tabla E.2: *P.valores test de Granger: Variable - Close*

- Mejores hiperparametros obtenidos por *Grid Search*:

Modelo	units1	dropout1	units2	dropout2	learning_rate
LSTM	125	0.2	75	0.1	0.0001
LSTM_s	100	0.1	50	0.2	0.0001
GRU	75	0.1	25	0.1	0.0001
GRU_s	100	0.1	50	0.2	0.0001

Tabla E.3: *Mejores hiperparámetros para modelos LSTM y GRU según mejores resultados en el Grid Search.*

- Metricas de las diferentes simulaciones:

Modelo	LSTM	GRU	LSTM-News	GRU-News
MSE	0.00178854	0.000793045	0.00390171	0.000186704
R	0.885705	0.949321	0.750666	0.988069
MAPE	17.2795	11.3591	25.8677	4.07354

Figura E.4: *Métricas de Amazon para Epochs=64 y Batch Size=32*

- Gráficos función de perdida:

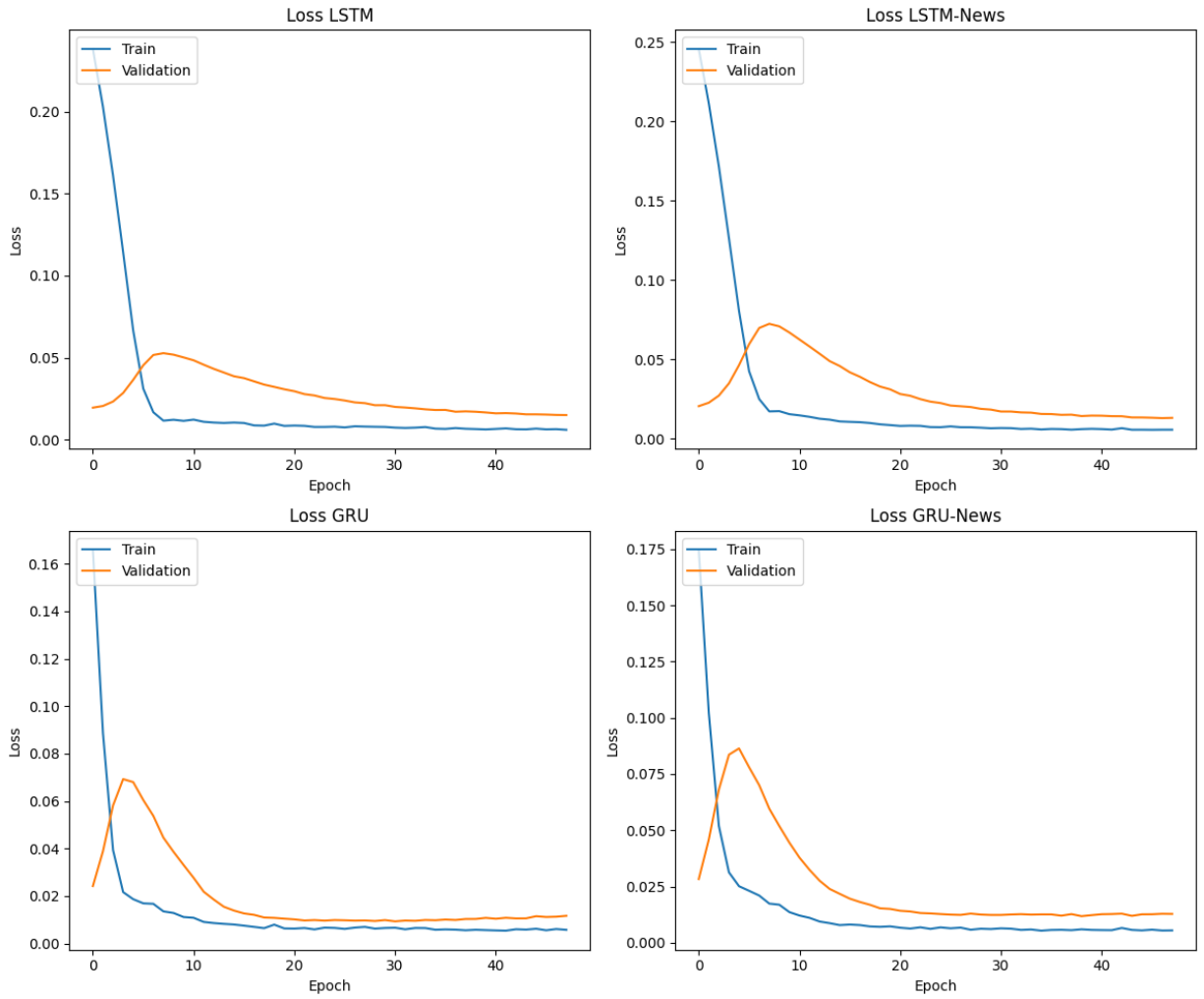


Figura E.5: *Function Losses de Netflix*

- Resultados gráficos:

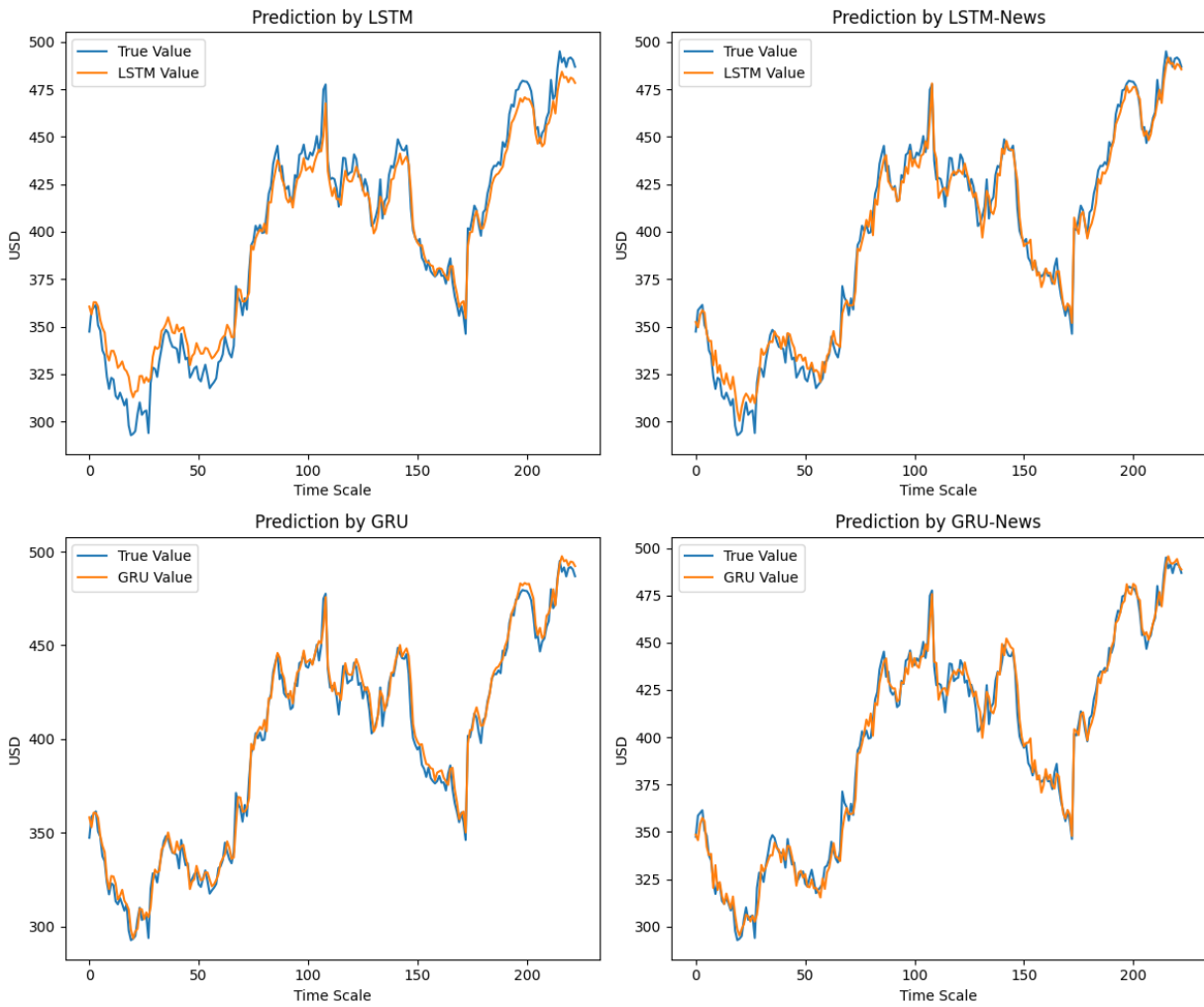


Figura E.6: *Netflix: Predictions vs real data*

- Graficos residuos

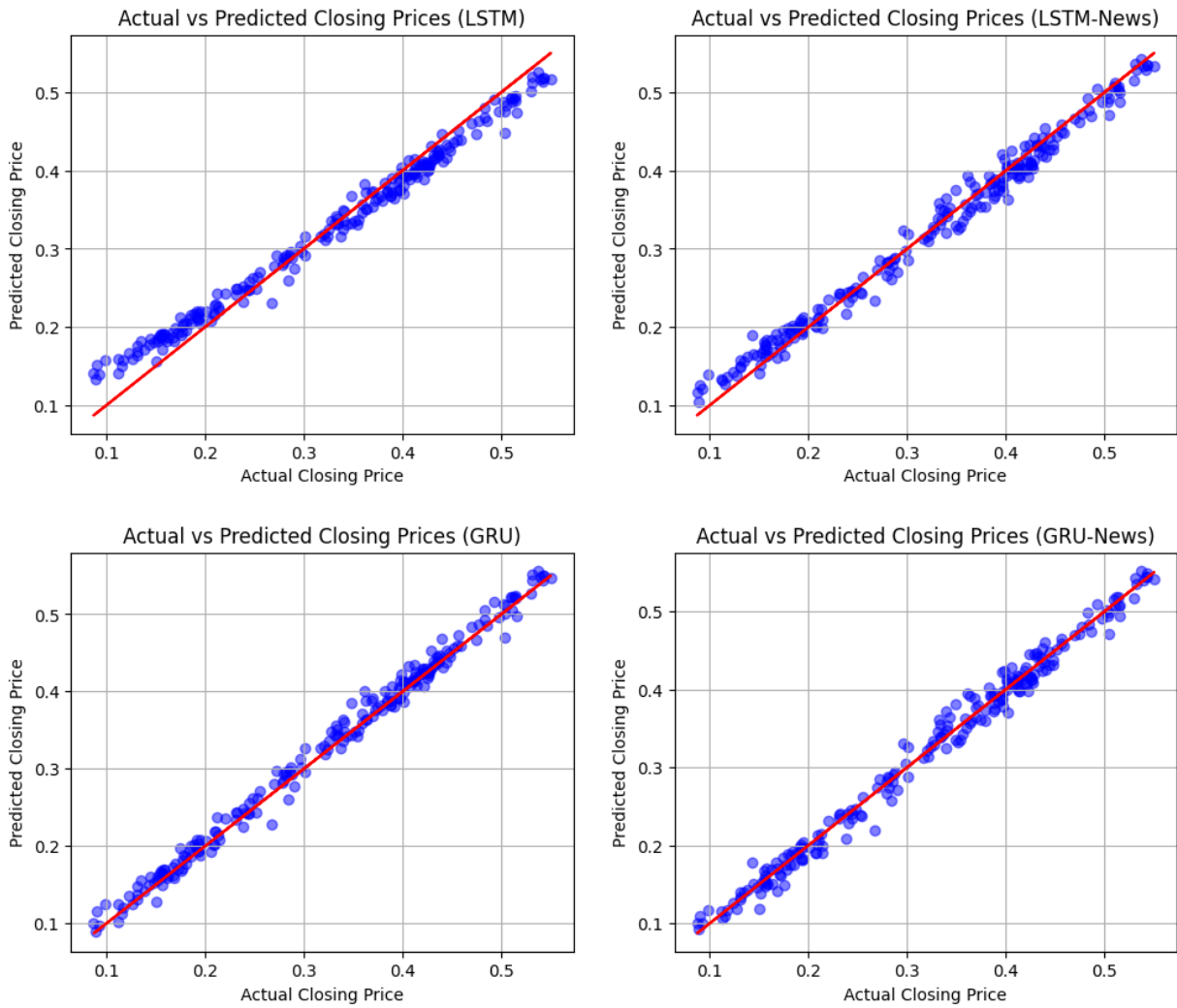


Figura E.7: *Resíduos de Netflix*

- Graficos QQ-Plots

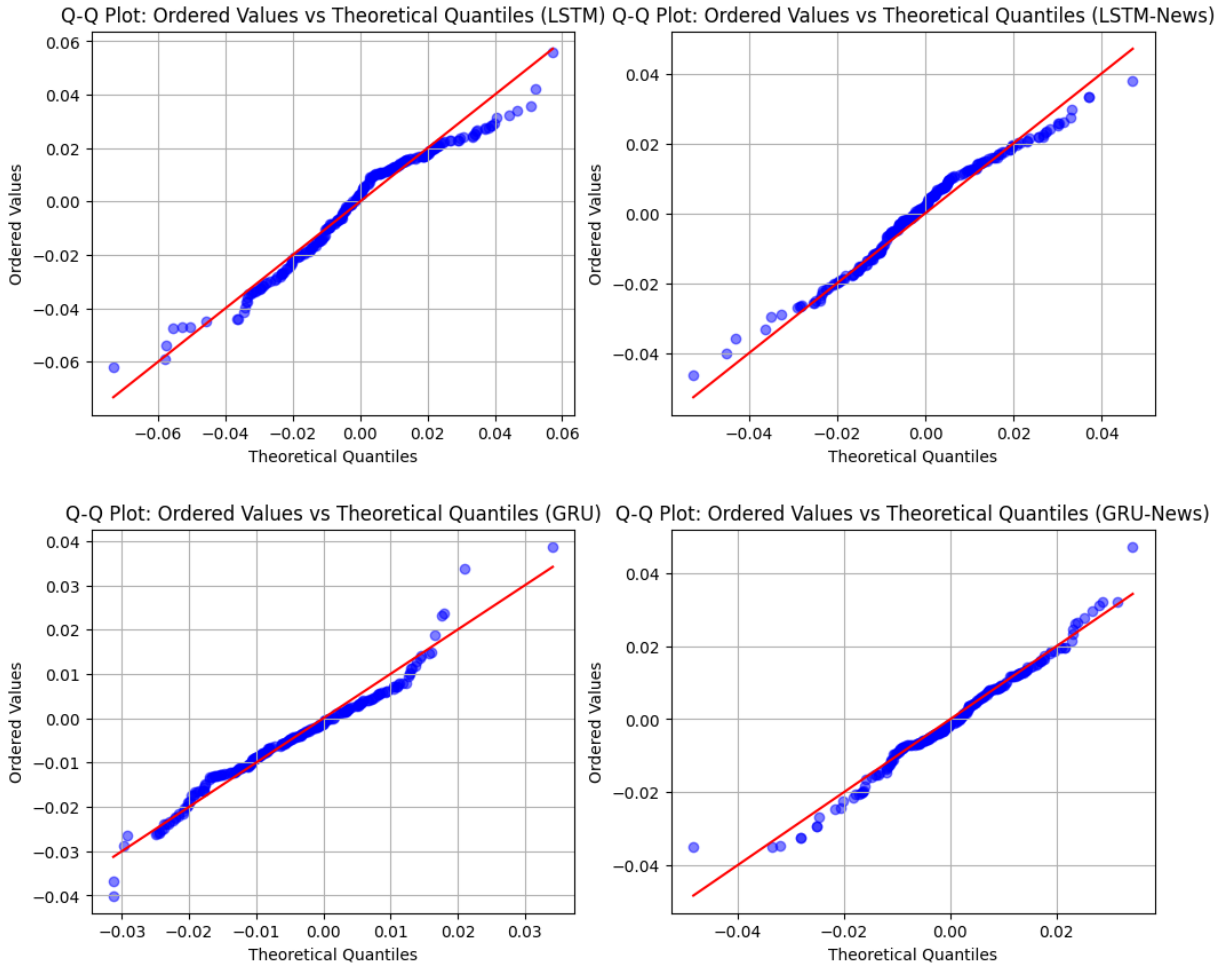


Figura E.8: *QQ-Plots de Netflix*

- Normalidad errores:

Modelo	Estadístico	P.Valor	Normal
LSTM	31.896749534040026	1.1849740598651101e-07	False
LSTM_s	44.13924068773096	2.6018704391977954e-10	False
GRU	1.1894217486312448	0.5517220598279524	True
GRU_s	9.74654195453897	0.007648306889526325	False

Tabla E.4: *Resultados del Test de Normalidad D'Agostino*

- Kruskal Wallis:

Estadístico de Prueba	P.Valor
209.02735430737857	4.7248860177383457e-45

Tabla E.5: *Resultados del test de Kruskal-Wallis para Netflix*

- Mann-Whitney:

Comparación	Estadístico de Prueba	P.Valor	Hipótesis
LSTM vs LSTM_s	30655.0	2.098651493674849e-05	Rechazada
LSTM vs GRU	22830.0	0.1350566412689601	No Rechazada
LSTM vs GRU_s	11656.0	2.8874689241605015e-22	Rechazada
LSTM_s vs GRU	17195.0	1.7534616229617975e-08	Rechazada
LSTM_s vs GRU_s	7956.0	1.9622638137864416e-35	Rechazada
GRU vs GRU_s	9677.0	6.511470484736962e-29	Rechazada

Tabla E.6: *Resultados del test de Mann-Whitney para Facebook*