



Universidad del
Rosario

**Pronóstico de volatilidad de la TRM mediante un modelo
híbrido LSTM-GARCH**

Trabajo de Grado presentado como requisito para optar al título de Magíster en Finanzas
Cuantitativas

Daniel Enrique Quintero Valencia
Director: Hugo Eduardo Ramirez

Universidad del Rosario
Facultad de Economía
Maestría en Finanzas Cuantitativas
Bogotá, D.C., Colombia
2019

Resumen

En este trabajo se propone un modelo híbrido LSTM-GARCH para el pronóstico de la volatilidad de la tasa representativa del mercado (TRM). Este modelo es una red neuronal recurrente LSTM, en la cual se incluyen como variables explicativas los coeficientes de modelos de series de tiempo GARCH, EGARCH y EWMA para la TRM, de acuerdo a la metodología de [Kim y col. 2018](#). Tanto los modelos GARCH como el modelo LSTM se estiman con datos históricos desde el año 2008 hasta junio de 2018, y los pronósticos generados se contrastan con datos desde julio 2018 a julio 2019. Los resultados se comparan utilizando varias medidas de error lineales y no-lineales. Se evidencia que la inclusión de los coeficientes de los modelos GARCH y EGARCH mejora la precisión de las predicciones del modelo híbrido LSTM comparado con un modelo LSTM estándar.

Clasificación JEL: C40, G15, G17

Palabras clave: Pronóstico de volatilidad, Deep learning, LSTM, GARCH, TRM, Tasa de cambio

Abstract

This work proposes a hybrid LSTM-GARCH model to forecast the volatility of the USD-COP exchange rate, know as tasa representativa del mercado (TRM). This model is a LSTM recurrent neural network which includes coefficients of time series models GARCH, EGARCH and EWMA for the TRM as explanatory variables, according to the framework from [Kim et al. 2018](#). Both GARCH and LSTM models are estimated with historical data from 2008 to June 2018, and the estimated forecasts are contrasted with data from July 2018 to July 2019. The acquired forecasts are compared using various linear and nonlinear error measures. The inclusion of coefficients from GARCH and EGARCH models is shown to improve the accuracy of hybrid LSTM-GARCH model predictions compared to a standard LSTM model.

JEL classification: C40, G15, G17

Keywords: Volatility forecast, Deep learning, LSTM, GARCH, TRM, Foreign Exchange

1. Introducción

La volatilidad es uno de los conceptos más usados e importantes en los análisis de finanzas. Diversos autores como [Andersen y col. 2001](#), [Petneházi y col. 2019](#), [Kim y col. 2018](#) indican su papel fundamental en la valoración de activos financieros y derivados, así como en la estimación de medidas de riesgo como el *VaR* (*Value At Risk*). Puntualmente, realizar pronósticos acertados de la volatilidad permite hacer valoraciones precisas de derivados financieros tales como las opciones, y realizar una gestión adecuada de las posibles pérdidas en las que pueda incurrir un portafolio de inversión por cambios en los activos que lo conforman.

Un enfoque novedoso se ha venido aplicando en los últimos años para el pronóstico de series de tiempo financieras, mediante el uso de técnicas de *Aprendizaje Profundo* (*Deep Learning*), específicamente un modelo de tipo red neuronal artificial recurrente llamado *Long Short Term Memory* (LSTM), alcanzando en la mayoría de los casos resultados más precisos.

Entre los estudios que han aplicado este enfoque se encuentran [Siarni-Namini y col. 2018](#), que realizan un contraste de los pronósticos de algunos índices bursátiles internacionales calculados a partir de un modelo LSTM y un modelo de series de tiempo ARIMA tradicional, encontrando una reducción de la tasa de error por encima del 80 %. [Nelson y col. 2017](#) utilizan un modelo LSTM para el pronóstico de los precios de acciones del índice IBovespa, obteniendo una precisión superior al 50 % en los precios pronosticados. [Fischer y col. 2018](#) utilizan un modelo LSTM para predecir los movimientos de los componentes del índice S&P 500 y crear una estrategia de trading, comparando los resultados con los pronósticos de una regresión logística y una red neuronal, encontrando rendimientos consistentemente superiores para los pronósticos realizados con el modelo LSTM.

Con respecto a la volatilidad, en [Petneházi y col. 2019](#) se realiza un estudio usando un modelo LSTM para el pronóstico de diversas medidas de la volatilidad utilizando los componentes del índice Dow Jones, concluyendo que es posible realizar pronósticos consistentes de la volatilidad con este tipo de técnica. [Bucci 2019](#) utiliza varios modelos de redes neuronales, entre ellos el LSTM, para el pronóstico de la volatilidad realizada, y compara la precisión de dichos pronósticos con respecto a los obtenidos por modelos econométricos autorregresivos de tipo ARFIMA, encontrando que los pronósticos calculados por el modelo LSTM son los más precisos.

En línea con estos estudios, el objetivo de este trabajo es desarrollar un modelo de predicción de tipo LSTM híbrido con modelos de series de tiempo GARCH para la volatilidad de la tasa representativa del mercado (TRM), usando como variable predicativa la volatilidad realizada (en inglés *realized volatility* o *RV*), y su posterior contraste con respecto a los pronósticos generados por modelos GARCH.

La metodología aplicada en este trabajo para la predicción de la volatilidad de la TRM sigue el método presentado en [Kim y col. 2018](#), el cual hace uso de varios modelos de red neuronal recurrente tipo LSTM, potenciados por la inclusión de los coeficientes de modelos de series de tiempo GARCH, EWMA y EGARCH estimados previamente como variables explicativas. La inclusión de dichos parámetros permite obtener pronósticos más precisos en comparación con los modelos GARCH, EWMA y EGARCH.

El contenido de este trabajo está organizado de la siguiente forma, en la sección 2 se exponen los conceptos básicos relacionados con la volatilidad y los modelos de series de tiempo GARCH y de red neuronal recurrente LSTM utilizados en este trabajo; en la sección 3 se realiza una descripción de la metodología mostrada en [Kim y col. 2018](#) para el pronóstico de la volatilidad; la sección 4 describe los planteamientos principales para la estimación de la volatilidad de la tasa representativa del mercado; la sección 5 muestra los resultados obtenidos aplicando la metodología a la volatilidad de la TRM; en la sección 6 se propone un ajuste a los modelos para incrementar la precisión de los pronósticos obtenidos; los resultados de dichos ajustes se muestran en la sección 7. En la sección 8 se hace una revisión de los resultados obtenidos. Finalmente, las conclusiones de este trabajo se presentan en la sección 9.

2. Conceptos Básicos

2.1. Volatilidad

[Engle y col. 2007](#) señala algunos hechos estilizados observables en el análisis de la volatilidad, por ejemplo la existencia de *clusters* o aglomeraciones de volatilidad, es decir la existencia de movimientos amplios o pequeños de los precios, de forma continua y persistente en el tiempo; choques asimétricos de volatilidad, más exactamente, variaciones amplias en los precios cuando estos presentan una tendencia a la baja, y por otro lado variaciones pequeñas cuando estos tienen una tendencia alcista; influencia por

información externa o «noticias», al presentarse o tener disponible nueva información relevante, los mercados responden a esta incrementando o disminuyendo la variación de los precios.

Puesto que no es posible observar directamente la volatilidad de un activo financiero, se debe utilizar una medida que se aproxime a esta, para el caso de este trabajo se considera la volatilidad realizada. Esta medida se utiliza como variable objetivo sobre la cual se realizan las predicciones. Para el cálculo de esta, se usa la medida definida en [Kim y col. 2018](#) para la volatilidad realizada (RV_t), que corresponde a la desviación estándar en un periodo ρ_t :

$$RV_t = \sqrt{\frac{1}{\rho} \sum_{s=1}^{\rho} (r_s - \bar{r}_s)^2}$$

Donde ρ indica el tamaño del intervalo de tiempo con el cual se calcula la desviación estándar, r_s son los retornos del activo analizado y \bar{r}_s es el promedio de los retornos para el periodo ρ . En este trabajo, el intervalo de tiempo definido es de 22 días atrás, es decir $\rho = 22$.

Vale la pena señalar que existen otras medidas y estimadores para la volatilidad realizada. Por ejemplo [Andersen y col. 2001](#) indican que el cálculo de los retornos al cuadrado ofrece una estimación insesgada de la volatilidad, sin embargo esta medida no se tuvo en cuenta buscando obtener resultados comparables con la metodología usada.

2.2. Modelos de volatilidad condicional - GARCH

Para la estimación de la volatilidad de un activo financiero es de interés modelar la varianza de forma condicional o con dependencia del tiempo, es decir, modelar una volatilidad heterocedastica.

Según [Sheppard 2019](#), una característica de los retornos de los activos financieros es la volatilidad cambiante a través del tiempo. Lo anterior debido a que se observan algunos fenómenos como choques causados por nueva información o «noticias», influencia o persistencia de choques de volatilidad anteriores, choques asimétricos por cambios positivos y negativos de los retornos, o la incertidumbre propia de los precios de estos

activos.

Un primer acercamiento al análisis de la volatilidad condicional fue el modelo ARCH (del inglés Autoregressive Conditional Heteroskedasticity) desarrollado por Engle en 1982. En él, la volatilidad condicional se encuentra relacionada, o es dependiente, del valor de las perturbaciones o innovaciones observadas en un número determinado de periodos de tiempo pasados o rezagos.

De acuerdo con [Sheppard 2019](#), un proceso ARCH de orden P está determinado por las siguientes ecuaciones:

$$\begin{aligned}r_t &= \mu_t + \epsilon_t \\ \sigma_t^2 &= \omega + \alpha_1 \epsilon_{t-1}^2 + \alpha_2 \epsilon_{t-2}^2 + \cdots + \alpha_p \epsilon_{t-p}^2 \\ \epsilon_t &= \sigma_t e_t \\ e_t &\sim N(0, 1)\end{aligned}$$

donde r_t es el retorno del activo que se está modelando, ϵ_t es el valor de la perturbación o innovación en tiempo t y σ_t^2 es la varianza condicional en el periodo $t - 1$ para el periodo t , la cual depende de P realizaciones anteriores o rezagos.

2.2.1. Modelo GARCH

En 1986, Bollerslev presentó una mejora al modelo ARCH, introduciendo una relación entre la varianza condicional y realizaciones anteriores o rezagos de la misma. Este modelo se conoce como ARCH generalizado (GARCH). La principal característica de este tipo de modelos es que la varianza condicional depende no solo de valores pasados de las perturbaciones o errores, sino que también depende de un promedio ponderado de las realizaciones anteriores de la volatilidad.

[Sheppard 2019](#) define un proceso GARCH de orden (P, Q) de acuerdo a las siguientes ecuaciones:

$$r_t = \mu_t + \epsilon_t$$

$$\begin{aligned}\sigma_t^2 &= \omega + \sum_{p=1}^P \alpha_p \epsilon_{t-p}^2 + \sum_{q=1}^Q \beta_q \sigma_{t-q}^2 \\ \epsilon_t &= \sigma_t e_t \\ e_t &\sim N(0, 1)\end{aligned}$$

donde ahora la varianza condicional σ_t^2 depende del valor de P realizaciones anteriores o rezagos de las perturbaciones o innovaciones observadas, y de Q rezagos anteriores de la volatilidad.

Un caso especial de modelo GARCH es el modelo EWMA (del inglés Exponentially Weighted Moving Averages), dado a conocer con la metodología RiskMetrics. Un modelo EWMA es un modelo GARCH con $P, Q = 1$, $\omega = 0$ y $\alpha + \beta = 1$. [Sheppard 2019](#) lo define con la siguiente ecuación:

$$\begin{aligned}\sigma_t^2 &= (1 - \lambda) \epsilon_{t-1}^2 + \lambda \sigma_{t-1}^2 \\ \lambda &= \beta, (1 - \lambda) = \alpha\end{aligned}$$

2.2.2. Modelo EGARCH

Nelson expuso en 1991 un caso especial de modelo GARCH llamado GARCH exponencial (EGARCH), en el cual se modela el logaritmo natural de la varianza condicional, e introduce unos términos adicionales para capturar asimetrías de los choques de volatilidad.

De acuerdo a [Sheppard 2019](#) un proceso EGARCH de orden (P, Q, O) se define con las siguientes ecuaciones:

$$\begin{aligned}r_t &= \mu_t + \epsilon_t \\ \ln(\sigma_t^2) &= \omega + \sum_{p=1}^P \alpha_p \left(\left| \frac{\epsilon_{t-p}}{\sigma_{t-p}} \right| - \sqrt{\frac{2}{\pi}} \right) + \sum_{o=1}^O \gamma_o \frac{\epsilon_{t-o}}{\sigma_{t-o}} + \sum_{q=1}^Q \beta_q \ln(\sigma_{t-q}^2) \\ \epsilon_t &= \sigma_t e_t \\ e_t &\sim N(0, 1)\end{aligned}$$

donde los términos $\left| \frac{\epsilon_{t-p}}{\sigma_{t-p}} \right|$ y $\frac{\epsilon_{t-o}}{\sigma_{t-o}}$ son choques con media cero. El primer término captura choques simétricos de la volatilidad condicional, mientras que el segundo genera choques asimétricos cuya dirección depende del signo del coeficiente γ_o . Comúnmente este signo es negativo para capturar el efecto de incrementos en la volatilidad ante choques negativos en el retorno del activo (o efecto *leverage*).

2.3. Red neuronal artificial

De acuerdo a [Géron 2017](#), las redes neuronales artificiales son modelos matemáticos cuyo funcionamiento se basa en la actividad de las células neuronales y su interacción entre ellas por medio de señales electroquímicas, y constituyen una de las principales técnicas usadas en el campo del *aprendizaje automatizado* (machine learning) para realizar tareas de clasificación y predicción.

Entre sus principales ventajas se encuentran ser modelos no paramétricos y no lineales ([Bucci 2019](#)), es decir que son capaces de capturar e incorporar relaciones no lineales existentes entre los datos. Además no requieren asumir una forma funcional específica para representar dichas relaciones ([Kristjanpoller y col. 2016](#)), lo cual permite aproximar, a partir de su estructura interna una función o proceso generador de los datos que son analizados ([Bucci 2019](#)).

El componente fundamental de este tipo de modelos es la neurona artificial o unidad, la cual es una función que recibe un vector *de entrada* x_i y, a partir de un vector de pesos w , calcula el producto punto entre estos para generar un valor o *señal* de salida a través de una función de *activación*. Adicionalmente es posible incluir una tendencia b a la señal de salida. Lo anterior se representa con la siguiente ecuación:

$$f(x_i) = g(x_i \cdot w) + b$$

Donde x_i es el vector de entrada de datos, w es el vector de pesos, $g(\cdot)$ es la función de activación y b es la tendencia.

Una *red neuronal artificial* es un conjunto de neuronas artificiales o unidades, agrupadas en forma de pilas o *capas*, organizadas de forma secuencial, y conectadas entre sí ([Bucci 2019](#)). Esta red está compuesta por una capa que recibe los datos de entrada y los transmite a la siguiente capa, conocida como *capa de entrada*; una o más *capas*

ocultas, donde se realizan los cálculos de la red; y una *capa de salida*, que recibe las señales de la última capa oculta y genera la señal de salida de la red correspondiente a los datos de entrada ingresados.

La figura 1 representa un esquema de una red neuronal con una capa de entrada compuesta por 4 unidades o neuronas, una capa oculta compuesta por 4 unidades y una capa de salida con una unidad. El número de unidades en la capa de entrada depende del número de variables (*features*) a analizar y el número de unidades en la capa de salida a la variable objetivo (si esta es escalar, vectorial, nominal o categórica).

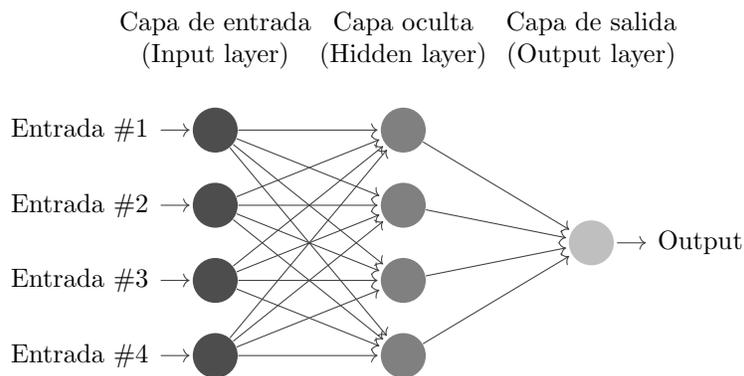


Figura 1: Representación de una red neuronal artificial

La capa de entrada recibe cada uno de los elementos del vector de entrada x_i y los transmite a la primera capa oculta. Las capas ocultas calculan sus valores de salida o señales, y las transmiten como vector de entrada a la siguiente capa.

Este tipo de redes, en las que la información que es procesada proviene únicamente de la capa de entrada y sigue una única dirección o sentido, desde la capa de entrada, pasando por las capas ocultas hacia la capa de salida se conocen en inglés como *feed-forward network* (Géron 2017).

La estimación de un modelo de red neuronal se realiza utilizando un conjunto de datos, denominados de *entrenamiento*, el cual está compuesto por variables explicativas o endógenas, y una variable objetivo o exógena, la cual se busca pronosticar con los valores de las variables explicativas.

A medida que se presenta al modelo las observaciones de los datos de entrenamiento, se calcula la señal de salida de la red que les corresponde. La red se ajusta por medio de las diferencias entre la salida generada y el valor de la variable objetivo, esta diferencia

se denomina error. Este error se calcula a partir de una función de pérdida, que ajusta paulatinamente los vectores de pesos w de cada unidad de la red neuronal, utilizando un algoritmo que propaga los ajustes desde la capa de salida hacia la capa de entrada. Este algoritmo de entrenamiento se conoce en inglés como *backpropagation* (Géron 2017).

2.4. Modelo Long-Short Term Memory (LSTM)

Cuando una red neuronal artificial tiene conexiones en las neuronas o unidades de las capas ocultas hacia ellas mismas, se conoce como *recurrente*. Esta conexión induce un bucle de retroalimentación (en inglés *feed-back*), que envía señales recurrentes como una nueva entrada de datos a las unidades cuando se analiza el conjunto de datos de entrenamiento.

La figura 2 muestra una neurona o unidad A con una conexión recurrente, un vector de entrada x_t y un valor o señal de salida h_t . La conexión recurrente equivale a crear una secuencia o cadena de unidades A , que envían una señal adicional a la unidad a medida que se van analizando las observaciones del conjunto de datos de entrenamiento.

Esta característica de encadenamiento de unidades a través de la conexión recurrente permite tener en consideración no sólo las relaciones existentes entre los datos de una observación, sino además posibles patrones o secuencias existentes entre las observaciones de los datos. Esto facilita a las redes neuronales recurrentes obtener mejores resultados en aplicaciones como generación y traducción de texto, y pronóstico de series de tiempo.

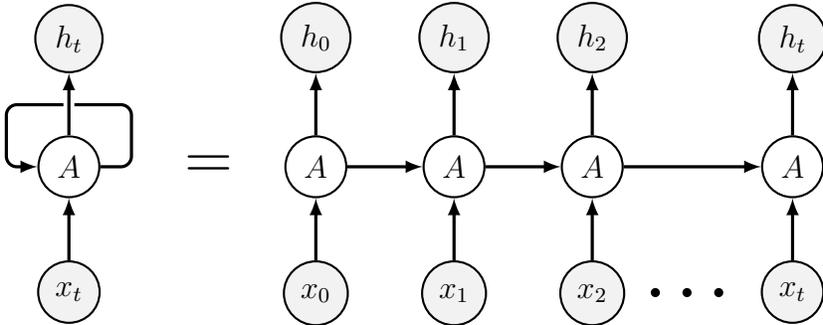


Figura 2: Esquema de una conexión recurrente (Olah 2015).

Una clase de red neuronal recurrente es la llamada Long Short-Term Memory

(LSTM). Una red neuronal LSTM está compuesta por neuronas o unidades más especializadas, llamadas bloques o celdas de memoria (en inglés *memory cell*). Cada bloque o celda de memoria está formado por un vector c_t llamado *estado de la celda* y tres entradas de datos o *puertas*:

- Puerta de olvido (forget gate g_t): Controla la información que se va a mantener del estado de la celda c_t entre iteraciones de entrenamiento.
- Puerta de entrada (input gate i_t): Define que información del estado actual de la celda c_t , se actualiza.
- Puerta de salida (output gate o_t): Genera la salida de la celda

El estado actual de la celda c_t se actualiza a partir del estado pasado (c_{t-1}) y de la nueva información a incorporar en la celda (\tilde{c}_t). Una representación gráfica de una celda LSTM puede verse en la figura 3, donde el estado de la celda c_t es modificado por la información de g_t , i_t y o_t . La salida de la celda una vez actualizado el estado corresponde al vector h_t .

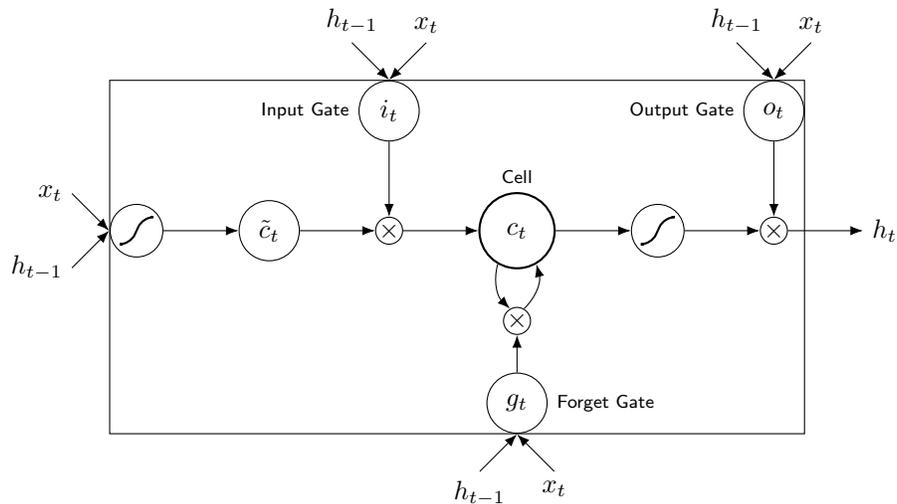


Figura 3: Esquema de un bloque o celda de memoria LSTM

Una celda de memoria LSTM está descrita según [Kim y col. 2018](#) por las siguientes ecuaciones:

$$\begin{aligned}
g_t &= \sigma(W_{gx} \cdot x_t + W_{gh} \cdot h_{t-1} + b_g) \\
i_t &= \sigma(W_{ix} \cdot x_t + W_{ih} \cdot h_{t-1} + b_i) \\
\tilde{c}_t &= \tanh(W_{cx} \cdot x_t + W_{ch} \cdot h_{t-1} + b_c) \\
c_t &= g_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \\
o_t &= \sigma(W_{ox} \cdot x_t + W_{oh} \cdot h_{t-1} + b_o) \\
h_t &= o_t \cdot \tanh(c_t)
\end{aligned}$$

Donde x_t es el vector de entrada con los datos de la observación en el intervalo t , h_{t-1} es el vector de datos de la conexión recurrente, que corresponde a la salida de la celda en el intervalo $t - 1$; W_{gh} , W_{ih} , W_{ch} y W_{oh} son los vectores de pesos asociados a h_{t-1} para las funciones g_t , i_t , \tilde{c}_t y o_t ; de igual forma W_{gx} , W_{ix} , W_{cx} y W_{ox} son los vectores de pesos asociados a x_t para las funciones g_t , i_t , \tilde{c}_t y o_t .

2.4.1. LSTM Stateless y Stateful

De acuerdo con la definición dada de red neuronal recurrente LSTM, las unidades de memoria que la componen mantienen su estado c_t al recorrer completamente la muestra de datos. Sin embargo, esto puede diferir según la implementación computacional usada. Por ejemplo, si los datos de entrenamiento se dividen en muestras o paquetes más pequeños para ser usados en el entrenamiento del modelo, puede suceder que los estados de las unidades no se mantengan cada vez que se finaliza uno de estos paquetes.

Dicho de otra forma, los estados c_t de las unidades que componen el modelo no se mantendrían durante el entrenamiento al cambiar los paquetes de datos. Este tipo de implementación se conoce como en inglés como *LSTM stateless*.

Si los estados c_t de las unidades se mantienen para toda la muestra de datos, sin importar si durante el entrenamiento esta es dividida en trozos, esta implementación se conoce en inglés como *LSTM stateful*.

Una consecuencia de usar un modelo LSTM stateful, es que el modelo es capaz de mantener, y de cierta forma «memorizar» de forma completa el set de datos de entrenamiento. Lo anterior también aplica al momento de realizar pronósticos, el modelo mantiene los estados c_t de sus unidades al realizar pronósticos con datos nuevos.

3. Metodología

Este trabajo sigue el método planteado en [Kim y col. 2018](#) para el pronóstico de la volatilidad de un activo financiero, realizando dos fases; en la primera se realiza una replicación del método con los datos originales para realizar pronósticos de la volatilidad del índice coreano KOSPI 200 (Korea Composite Stock Price Index); y en la segunda fase se utilizan los modelos de series de tiempo GARCH, EWMA, EGARCH, y los modelos de red neuronal recurrente LSTM estimados en la primera fase para el pronóstico de la volatilidad de la TRM.

En el desarrollo de la primera fase los datos usados como variables exógenas son el índice coreano KOSPI 200 (KOSPI), tanto los precios como los log-retornos del índice, la tasa de interés del bono de deuda pública coreana a tres años (KTB), y los precios de cierre del oro y petróleo WTI en dólares estadounidenses. El intervalo de tiempo de estos datos es entre el 1 de enero de 2001 y 1 de enero de 2017.

Los datos descritos anteriormente se dividen en dos muestras, una para estimación de los modelos GARCH, EWMA y EGARCH, y los modelos LSTM con los datos desde el 2 de enero de 2001 hasta el 30 de septiembre de 2011, y otra desde el 1 de octubre de 2011 hasta el 29 de diciembre de 2016, para la evaluación de los modelos realizando pronósticos fuera de muestra (*out-of-sample*).

Como primer paso se realiza un análisis estadístico descriptivo de las variables seleccionadas, así como la prueba de normalidad Jarque-Bera (JB) para verificar la normalidad tanto de los precios del índice KOSPI 200, como de los log-retornos de este; y la prueba de estacionariedad Dickey-Fuller Aumentada (ADF) para la detección de raíz unitaria. De acuerdo con los resultados de [Kim y col. 2018](#), tanto los precios como los log-retornos del índice KOSPI 200 no siguen una distribución normal según la prueba Jarque-Bera, mientras que la prueba Dickey-Fuller Aumentada indica que los precios del índice tienen raíz unitaria, mientras que los log-retornos no.

A continuación se realiza la estimación de los modelos de series de tiempo GARCH, más específicamente se estiman tres modelos: GARCH(1,1), EGARCH(1,1) y EWMA, con los log-retornos del índice KOSPI 200 entre el 2 de enero de 2001 hasta el 30 de septiembre de 2011. Adicionalmente se entrena un modelo de red neuronal recurrente LSTM utilizando como datos el índice coreano KOSPI 200 (KOSPI) en precios y en log-retornos, la tasa de interés del bono de deuda pública coreana a tres años (KTB), y los precios de cierre del oro y petróleo WTI en dólares, utilizando el mismo intervalo

de fechas de los modelos de series de tiempo.

A partir de los coeficientes o parámetros de los modelos GARCH(1,1), EGARCH(1,1) y EWMA estimados con anterioridad, se entrenan versiones del modelo LSTM, adicionando como variables exógenas diferentes combinaciones de los coeficientes de los modelos GARCH, EGARCH y EWMA.

Estas versiones se seleccionan con base en los resultados obtenidos por [Kim y col. 2018](#), buscando los mejores modelos de acuerdo con cuatro medidas de error: error absoluto medio (Mean absolute error; MAE), error cuadrático medio (Mean squared error; MSE), error absoluto medio ajustado por heterocedasticidad (Heteroskedasticity-adjusted MAE) y error cuadrático medio ajustado por heterocedasticidad (Heteroskedasticity-adjusted MSE). Los modelos seleccionados son E-LSTM (modelo LSTM con coeficientes EWMA), GE-LSTM (modelo LSTM con coeficientes GARCH y EWMA) y GEW-LSTM (modelo LSTM con coeficientes GARCH, EGARCH Y EWMA)

Tanto con los modelos de series de tiempo GARCH, EGARCH y EWMA, como con los modelos de red neuronal recurrente LSTM, E-LSTM, GE-LSTM Y GEW-LSTM se calculan pronósticos de un día desde el 1 de octubre de 2011 hasta el 29 de diciembre de 2016, y se compara la precisión de dichos pronósticos contra la medida de la volatilidad realizada observada mediante las cuatro medidas de error MAE, MSE, HMAE y HMSE.

Adicionalmente, para establecer si las predicciones de cada modelo son estadísticamente diferentes entre sí, se realizan pruebas estadísticas Diebold-Mariano (DM) y rangos con signo de Wilcoxon (WS).

La segunda fase, de pronóstico de la volatilidad de la TRM, sigue esta metodología y es explicada en los apartes siguientes de este trabajo.

4. Aplicación a la TRM

La tasa representativa del mercado, abreviada TRM, es el precio de un dólar de los Estados Unidos en pesos colombianos. De acuerdo al [Banco de la República s.f.](#), esta tasa es calculada por la Superintendencia Financiera de Colombia con base en las operaciones de compra y venta realizadas por los intermediarios financieros en el mercado local el día anterior. La figura 4 ilustra el valor que ha tomado la TRM desde

el año 2008 hasta julio de 2019, donde se evidencian dos grandes movimientos, un pico alrededor del año 2009, y una tendencia alcista desde mediados del año 2014.

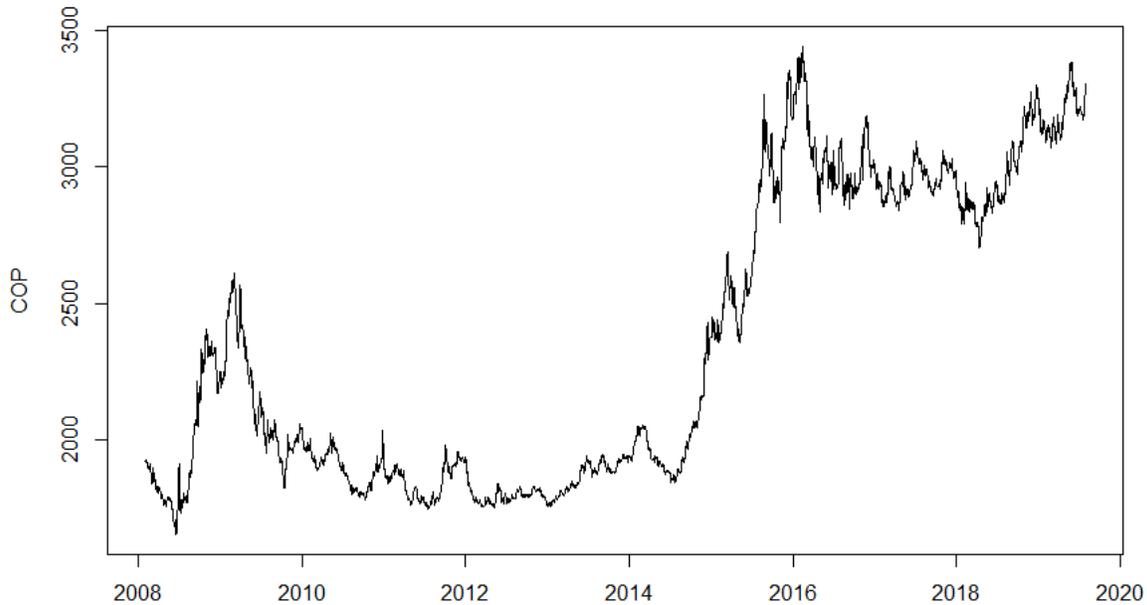


Figura 4: Valor de la TRM entre los años 2008 y 2019

Para el caso particular de este trabajo, se propone aplicar la metodología expuesta en [Kim y col. 2018](#) para el pronóstico de la volatilidad de la TRM, más exactamente sobre los log-retornos. Se hace necesario entonces, determinar cuáles podrían considerarse como factores que podrían influir en los movimientos y variación de la TRM para utilizarse como variables exógenas para los modelos de redes neuronales recurrentes LSTM.

En [Corredor Velandia y col. 2012](#) se hace un estudio de la influencia de algunas variables macroeconómicas de Estados Unidos sobre la economía local. Para el caso particular de la TRM, se encuentra que la tasa de interés de la Reserva Federal (Federal funds rate) es el principal factor de influencia dentro de las variables estudiadas. También se considera la Tasa interbancaria (TIB) del Banco de la República, la tasa DTF y TES cupón 11 % y vencimiento el 24 de julio de 2020 como factores de influencia.

En el análisis de [Parra Barrios 2019](#) también se concluye que la tasa de interés de la Reserva Federal ejerce una influencia importante en los movimientos de la TRM. Por último, en [Suárez y col. 2006](#) se realiza un análisis de la relación entre la tasa de interés de los TES y la TRM.

Luego, siguiendo el análisis de [Kim y col. 2018](#) y las conclusiones de las referencias anteriores, se propone utilizar como variables exógenas para el pronóstico de la volatilidad de los log-retornos la TRM, una combinación de tasas de interés, tanto locales como de Estados Unidos, más los precios de las materias primas (en inglés *commodities*) más importantes para la economía local, en este caso el oro y el petróleo WTI.

4.1. Datos

Según el aparte anterior, las variables seleccionadas para el estudio se describen en la tabla 1, desagregadas en tres grupos principales, tasas de interés nacionales o determinadas de forma local, tasas de interés foráneas de Estados Unidos y precios de materias primas o *commodities*. Se incluye como variable la TRM, tanto en precios como en log-retornos. Para la variable endógena u objetivo, se calcula la volatilidad realizada de los log-retornos de la TRM, a un intervalo de 22 días como se definió anteriormente. La figura 5 muestra el valor calculado de esta volatilidad para el periodo de tiempo en evaluación, desde el año 2008 hasta julio de 2019.

Las series de datos de cada variable se obtuvieron mediante la plataforma Bloomberg entre el 3 de enero de 2008 y el 31 de julio de 2019 en forma diaria, para un total de 3020 observaciones. El conjunto de datos se divide en dos componentes, el primero, del 3 de enero de 2008 al 29 de junio de 2018 (2737 observaciones, o 91 % del total de la muestra) como datos de estimación y entrenamiento de los modelos; y el segundo, del 2 de julio de 2018 al 31 de julio de 2019 (283 observaciones o 9 % de la muestra) para evaluación fuera de muestra (*out-of-sample*) de los pronósticos creados. Por otro lado, Los precios del oro y petróleo WTI se encuentran en dólares, con el fin de aislar el efecto de la tasa de cambio de estos.

Adicionalmente se realizó el análisis estadístico descriptivo de las variables, los resultados se presentan en la tabla 2. Puede observarse que ninguna de las variables sigue una distribución normal según la prueba Jarque-Bera. Con la hipótesis nula que la variable se distribuye de forma normal, el p-valor para todas las variables seleccionadas es 0,00 lo cual permite rechazar la hipótesis nula con un nivel de significancia de 0,05.

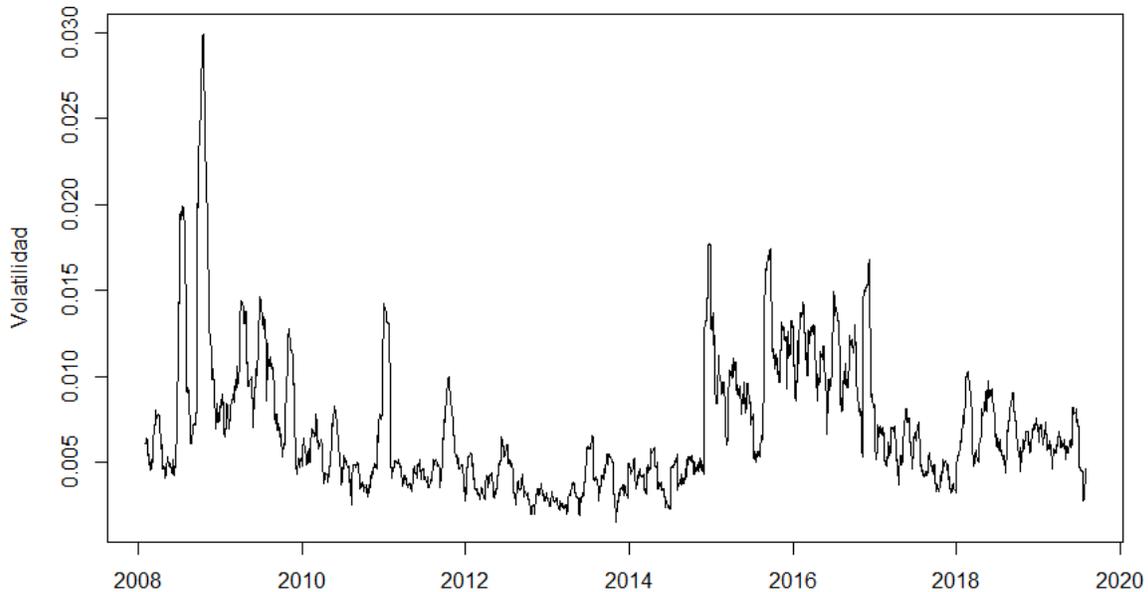


Figura 5: Volatilidad realizada para los log-retornos de la TRM, entre los años 2008 y 2019

También se observa que la mayoría de las variables tienen raíz unitaria, es decir no son estacionarias según la prueba Dickey-Fuller Aumentada. Para la hipótesis nula que la variable tiene raíz unitaria, el p-valor de las variables FFRATE, FFTRATE, así como de los log-retornos de la TRM es menor al nivel de significancia de 0,05 lo que permite rechazar dicha hipótesis.

Se incluyen además los coeficientes de correlación entre las variables explicativas y la TRM, los log-retornos de la TRM y la volatilidad realizada, tal como se muestran en la figura 6.

4.2. Modelos GARCH

Según lo indicado en [Kim y col. 2018](#), se estiman tres modelos de series de tiempo GARCH para el log retorno de la TRM: GARCH, EGARCH y EWMA. Se usó R con

	Variable	Descripción
Tasas de interés locales	IBR	Indicador Bancario de Referencia
	Yield TES 1Y	Curva cero cupon TES a 1 año en pesos
	Yield TES 5Y	Curva cero cupon TES a 5 años en pesos
	Yield TES 10Y	Curva cero cupon TES a 10 años en pesos
	Yield TES 1Y UVR	Curva cero cupon TES a 1 año en pesos UVR
	Yield TES 5Y UVR	Curva cero cupon TES a 5 años en pesos UVR
	Yield TES 10Y UVR	Curva cero cupon TES a 10 años en pesos UVR
Tasas de interés foráneas	US 5Y Yield	Curva cero cupon bono (Treasury) USA a 5 años
	US 10Y Yield	Curva cero cupon bono (Treasury) USA a 10 años
	FF Effective rate	Tasa de interés de la Reserva Federal
	FF Target Rate - Upper Bound	Limite superior de la Tasa de interés Objetivo de la Reserva Federal
Commodities	WTI	Precio petroleo WTI
	GOLD	Precio oro
Tasa de cambio	TRM	Tasa Representativa del Mercado
	LRet_TRM	Log-retornos de la TRM
Variable endógena	RV(LRet_TRM)	Volatilidad realizada de los log-retornos de la TRM

Tabla 1: Variables incluidas en el conjunto de datos para el pronóstico de la volatilidad de la TRM.

la librería `Rugarch` para el cálculo de los parámetros, utilizando el método `ugarchspec`. Los parámetros usados en la especificación de cada modelo son:

- GARCH: `model = 'sGARCH', garchOrder = (1,1), submodel = 'GARCH'`
- EGARCH: `model = 'eGARCH', garchOrder = (1, 1)`
- EWMA: `model = 'sGARCH', garchOrder = (1, 1)), fixed.pars = (omega = 0, beta1 = 0.97)`

	med	de	asim	cur	p-value jb	p-value adf
IBR	4,931	1,812	1,168	0,393	0,00	0,59
TES_1Y	5,508	1,798	1,820	2,433	0,00	0,48
TES_5Y	7,061	1,772	1,563	2,179	0,00	0,40
TES_10Y	7,836	1,668	1,291	1,549	0,00	0,36
TES_1Y_UVR	0,021	0,012	1,756	3,214	0,00	0,12
TES_5Y_UVR	0,033	0,010	1,458	1,768	0,00	0,30
TES_10Y_UVR	0,039	0,009	1,242	1,622	0,00	0,33
US_5Y	1,790	0,675	0,258	-0,620	0,00	0,47
US_10Y	2,598	0,658	0,390	-0,755	0,00	0,32
FFRATE	0,645	0,838	1,543	1,475	0,00	0,01
FFTRATE	0,761	0,814	1,513	1,405	0,00	0,01
WTI	74,492	23,672	0,193	-0,866	0,00	0,48
COLD	1268,914	231,149	0,198	-0,036	0,00	0,63
TRM	2339,815	537,607	0,450	-1,476	0,00	0,61
LRet_TRM	0,000161	0,00814	-0,131	7,367	0,00	0,01

Tabla 2: Estadísticos descriptivos para las variables seleccionadas. *med*: media, *de*: desviación estándar, *asim*: asimetría, *cur*: curtosis, *p-value jb*: p-valor de la prueba Jarque-Vera, *p-value adf*: p-valor de la prueba Dickey-Fuller Aumentada

El código usado para la especificación de los modelos es el siguiente:

```

1 garch.spec <- ugarchspec()
2 egarch.spec <- ugarchspec(variance.model = list(model = "eGARCH", garchOrder =
  c(1, 1)))
3 ewma.spec <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c
  (1, 1)), fixed.pars = list(omega = 0, beta1 = 0.97))

```

Los modelos GARCH, EGARCH y EWMA se estiman con la muestra de entrenamiento desde el 3 de enero de 2008 hasta el 29 de junio de 2018. Una vez estimados los modelos, se extraen los coeficientes obtenidos de cada uno y se actualiza el archivo de datos con los valores de estos coeficientes para el entrenamiento de los modelos LSTM como datos de entrada. Estos coeficientes se muestran en la tabla 3.

Adicionalmente, con los modelos GARCH estimados se pronostica la volatilidad de los log-retornos de la TRM en el intervalo de evaluación fuera de muestra, desde el 2 de julio de 2018 al 31 de julio de 2019. Estos pronósticos se comparan posteriormente con los obtenidos de los modelos LSTM.

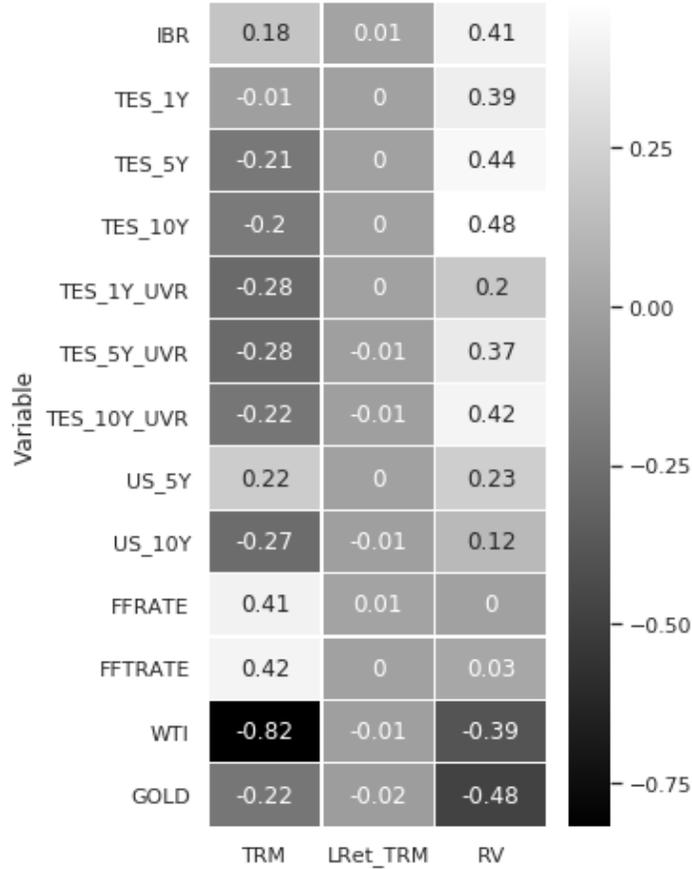


Figura 6: Coeficientes de correlación entre las variables de estudio

4.3. Modelos LSTM

Para la estimación de los modelos LSTM, y con el propósito de obtener resultados consistentes, se siguieron las especificaciones de diseño dadas en [Kim y col. 2018](#), así: se creó una red neuronal artificial de cinco capas, tres capas de tipo LSTM y dos capas *feed-forward* totalmente conectadas (*Fully Connected*). El número de neuronas en cada capa sigue la estructura 10-4-2-5-1, es decir, 10, 4 y 2 neuronas para las capas LSTM, y 5 y 1 neuronas para las capas *feed-forward*.

Para reducir las posibilidades de sobre entrenamiento (en inglés *over-fitting*) de los modelos LSTM, en las capas LSTM se incluye un efecto de *dropout* ([Srivastava y col. 2014](#)), el cual, de forma probabilística activa o desactiva algunas neuronas o unidades,

Modelo	Tend.	ε_{t-1}^2	σ_{t-1}^2	$\frac{\varepsilon_{t-1}^2}{\sigma_{t-1}^2}$	$\left \frac{\varepsilon_{t-1}^2}{\sigma_{t-1}^2} \right $
GARCH	Valor	0,0000007757576	0,07646338	0,9140285	
	Param. ω		α_1	β_1	
EGARCH	Valor	-0,1106628		0,9879426	0,06341836
	Param. ω			β_1	α_1
EWMA	Valor		0.029	0.97	
	Param.		α_1	β_1	

Tabla 3: Parámetros estimados para los modelos tipo GARCH de los log retornos de la TRM

junto con sus conexiones, en la capa sobre la cual es aplicado. Las probabilidades del efecto dropout aplicadas a cada capa son respectivamente 0.3, 0.4 y 0.4.

Los modelos se entrenan usando el algoritmo de optimización Adam [Kingma y col. 2014](#), el cual es un algoritmo de optimización de gradiente descendiente que suaviza el tamaño de cada gradiente o «descenso» a partir de los gradientes pasados. La tasa de aprendizaje (*learning rate*) usada en la optimización fue de 0.0001.

Los datos para el entrenamiento de los modelos LSTM provienen del archivo creado durante la estimación de los modelos GARCH con R, el cual contiene los coeficientes de los modelos GARCH obtenidos.

La implementación computacional de estos modelos se realizó usando el lenguaje de programación Python junto con la librería de aprendizaje profundo Keras ([Chollet y col. 2015](#)), el código con la especificación de los modelos LSTM es el siguiente:

```

1 model = Sequential()
2 model.add(LSTM(10, input_shape=(X_train.shape[1], X_train.shape[2]), return_
   sequences=True))
3 model.add(Dropout(0.3))
4 model.add(LSTM(4, return_sequences=True))
5 model.add(Dropout(0.4))
6 model.add(LSTM(2, return_sequences=False))
7 model.add(Dropout(0.4))
8 model.add(Dense(5))
9 model.add(Dense(1))

```

Donde `X_train` es el objeto que contiene los datos de entrenamiento para los modelos LSTM, `X_train.shape[1]` indica el número de filas y `X_train.shape[2]` el número de columnas del objeto. La opción `return_sequences` se activa dentro de las capas LSTM

para generar como salida la secuencia completa de los valores pronosticados de la capa.

4.3.1. Ajuste de datos

Para el entrenamiento de los modelos LSTM se requiere aplicar algunos ajustes adicionales a los datos. Como primera medida, los datos se transforman al tipo flotante de Python *float32* y normalizan en una escala $[0, 1]$ para eliminar posibles sesgos por las diferentes escalas en las que se encuentran las variables, de acuerdo con lo indicado por [Brownlee 2017](#).

Con el fin de aplicar el mecanismo de ventana deslizante (*rolling window*) descrito en [Kim y col. 2018](#) durante el entrenamiento de los modelos, para cada variable en la muestra de datos se agregaron 22 valores rezagados como nuevas variables, de tal forma que los 22 rezagos corresponden a una ventada deslizante de 22 días. Es decir: sea t la observación t -ésima de la muestra de datos, $x_{j,t}$ el valor de la variable j -ésima en la observación t , y 22 el número de rezagos a incluir en la muestra de datos, entonces $x_{j,t}$ se expande con sus rezagos de la siguiente forma:

$$[x_{j,t}] \Rightarrow [x_{j,t} \quad x_{j,t-1} \quad x_{j,t-2} \quad \cdots \quad x_{j,t-22}]$$

Donde la matriz x contiene los valores de las variables indicadas en el cuadro 1.

Luego, la observación del conjunto de datos para el periodo t estará compuesta por el valor de las variables que toman en dicho periodo $x_{j,t}$, más sus respectivos rezagos, desde $x_{j,t-1}$ hasta $x_{j,t-22}$.

$$\begin{bmatrix} x_{j,t} & \cdots \\ x_{j,t-1} & \cdots \\ x_{j,t-2} & \cdots \\ \vdots & \\ x_{j,t-n} & \cdots \end{bmatrix} \Rightarrow \begin{bmatrix} x_{j,t} & x_{j,t-1} & x_{j,t-2} & x_{j,t-3} & \cdots & x_{j,t-22} & \cdots \\ x_{j,t-1} & x_{j,t-2} & x_{j,t-3} & x_{j,t-4} & \cdots & x_{j,t-23} & \cdots \\ x_{j,t-2} & x_{j,t-3} & x_{j,t-4} & x_{j,t-5} & \cdots & x_{j,t-24} & \cdots \\ \vdots & & & & & & \vdots \\ x_{j,t-n} & x_{j,t-n-1} & x_{j,t-n-2} & x_{j,t-n-3} & \cdots & x_{j,t-n-22} & \cdots \end{bmatrix}$$

Si se tienen n observaciones de m variables endógenas, y se incluyen p rezagos para

cada una de ellas, la matriz de datos para el entrenamiento de los modelos LSTM será de dimensión $(n - p), (m \times p)$.

La figura 7 ilustra como está conformada una observación durante el entrenamiento de los modelos recurrentes LSTM.

Se conforman paquetes de datos para 22 periodos de tiempo, desde $t - 22$ hasta t . Cada paquete contiene los valores de las variables exógenas observados en dicho periodo, los coeficientes de los modelos GARCH, EGARCH y EWMA, el valor de la volatilidad realizada observado en el mismo periodo.

En la parte final de la observación, se incluye como variable objetivo el valor de la volatilidad realizada de los log-retornos de la TRM correspondiente al periodo $t + 1$, este es el valor que se busca pronosticar durante el entrenamiento.

4.3.2. Especificación de los modelos

De acuerdo con el diseño del modelo LSTM expuesto en el numeral 4.3, se realizó la implementación computacional de varios modelos LSTM. De los modelos realizados por Kim y col. 2018, se seleccionaron los que presentaron las menores medidas de error por cada categoría, como se enuncian a continuación:

- Modelos individuales: LSTM, entrenado únicamente con las variables exógenas seleccionadas, y la volatilidad realizada como variable endógena.
- Modelos LSTM con un parámetro GARCH: E-LSTM, el cual tiene como datos de entrada las variables del modelo LSTM, más los coeficientes obtenidos del modelo EGARCH.
- Modelo LSTM con dos parámetros GARCH: GE-LSTM, que tiene los datos de entrada del modelo E-LSTM, adicionando los coeficientes del modelo GARCH.
- Modelo LSTM con tres parámetros GARCH: GEW-LSTM, cuyos datos de entrada son las variables exógenas de los modelos anteriores, más los coeficientes de los tres modelos (GARCH, EGARCH y EWMA).

Los coeficientes de los modelos GARCH, EGARCH y EWMA provienen de los modelos descritos en el numeral 4.2.

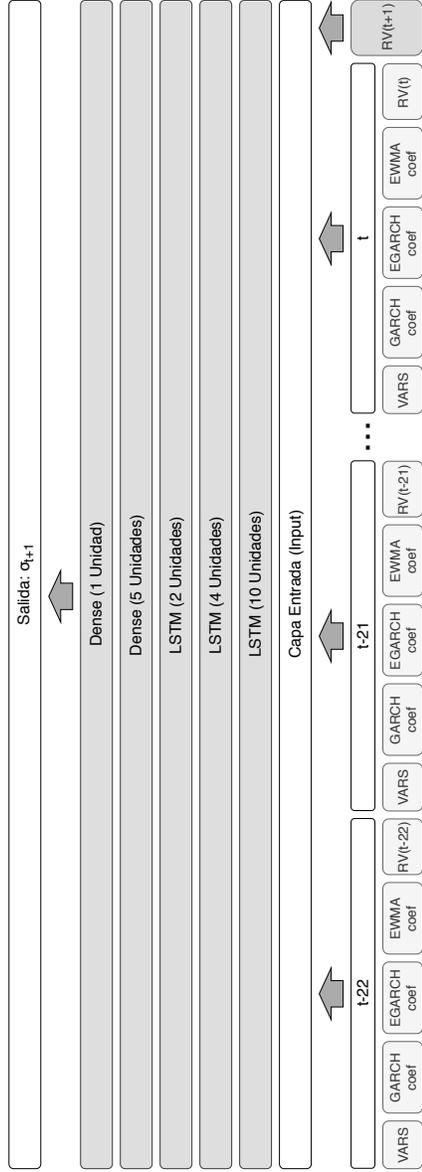


Figura 7: Diagrama de una observación de entrenamiento para los modelos recurrentes LSTM

Adicionalmente, para cada uno de los 4 modelos LSTM descritos, se entrenó una versión stateless y una stateful. En total se crearon 8 modelos LSTM con diferentes configuraciones.

4.3.3. Entrenamiento

Los modelos LSTM implementados en `Keras` requieren para su entrenamiento dividir la muestra de datos en paquetes de observaciones o *batches*. La selección del tamaño del batch, como muestra [Kristjanpoller y col. 2016](#), es uno de los hiperparámetros del modelo que pueden afectar su desempeño. Por lo anterior se realizaron pruebas con diferentes valores arbitrarios para el tamaño del batch, iniciando con un valor de 22, que corresponde al número de rezagos por variable incluidos en la muestra de estimación, según lo mostrado en el numeral [4.3.1](#).

Por otro lado, realizar la implementación en `Keras` de los modelos LSTM stateful exige que el tamaño del batch sea múltiplo del tamaño de la muestra con la cual se hace tanto el entrenamiento como las predicciones, es decir, todos los batches deben tener la misma cantidad de observaciones.

Teniendo en cuenta las consideraciones anteriores, El tamaño seleccionado del batch fue de 23 observaciones, donde cada observación tiene 22 rezagos de cada variable exógena.

Por otro lado, para el entrenamiento de los modelos también se requiere especificar el número de recorridos o iteraciones sobre la muestra de datos que se realizaran durante el entrenamiento. Este número de iteraciones sobre la muestra se conoce en inglés como *epochs*. [Kim y col. 2018](#) definieron 150 iteraciones o *epochs* en el entrenamiento de sus modelos, sin embargo, para el pronóstico de la volatilidad realizada de los log-retornos de la TRM, los modelos se entrenaron con diferentes números de iteraciones, iniciando con 150. Se encontró que los modelos entrenados con 450 epochs tenían los mejores resultados.

Finalmente, de la muestra de entrenamiento se extrajo un 20% de observaciones del tramo final como datos de evaluación de sobre entrenamiento (*overfitting*) de los modelos.

El código para ejecutar el entrenamiento de los modelos LSTM es el siguiente:

```

1 history = model.fit(X_train, Y_train, epochs=450, batch_size=23, validation_
    data=(X_test, Y_test), verbose=0, shuffle=False)

```

5. Resultados preliminares

Una vez entrenados cada uno de los modelos, se calcularon 276 pronósticos para la volatilidad realizada de los log-retornos de la TRM, tanto con los modelos de series de tiempo GARCH, como con los modelos tipo LSTM en sus versiones *stateless* y *stateful*, para el periodo de evaluación *out-of-sample* desde el 2 de julio de 2018 hasta el 31 de julio de 2019. Los pronósticos obtenidos se contrastan contra la volatilidad realizada calculada en el mismo periodo, usando cuatro medidas de error, señaladas a continuación (indicando los nombres en inglés entre paréntesis):

- Error absoluto medio (Mean absolute error): $MAE = \frac{1}{T} \sum |\hat{\sigma}_t - RV_t|$
- Error cuadrático medio (Mean squared error): $MSE = \frac{1}{T} \sum (\hat{\sigma}_t - RV_t)^2$
- Error absoluto medio ajustado por heterocedasticidad (Heteroskedasticity-adjusted MAE): $HMAE = \frac{1}{T} \sum \left| \frac{1 - \hat{\sigma}_t}{RV_t} \right|$
- Error cuadrático medio ajustado por heterocedasticidad (Heteroskedasticity-adjusted MSE): $HMSE = \frac{1}{T} \sum \left(\frac{1 - \hat{\sigma}_t}{RV_t} \right)^2$

Donde RV_t corresponde a la volatilidad realizada calculada para el periodo t usando 22 observaciones (en este caso, días) anteriores, y $\hat{\sigma}_t$ es el pronóstico de la volatilidad realizada para el mismo periodo.

Las medidas HMAE y HMSE se incluyen como medidas de error no lineales compensadas para heterocedasticidad, según lo indicado por [Kim y col. 2018](#). Los valores de cada medida de error para cada uno de los modelos, tipo GARCH y tipo LSTM, se presentan en la [tabla 4](#).

De acuerdo los resultados de la tabla, el modelo con el peor desempeño es el EGARCH (medidas indicadas en cursiva), mientras que el modelo con el mejor desempeño es el GE-LSTM stateful (medidas indicadas en negrita), seguido por el modelo

LSTM sin coeficientes GARCH versión stateful (LSTM SF). Teniendo en cuenta únicamente los modelos GARCH, EGARCH y EWMA, el modelo con el mejor desempeño es el GARCH (medidas subrayadas).

Estos resultados difieren a los obtenidos por [Kim y col. 2018](#), donde el modelo con las menores medidas de error correspondía al modelo LSTM que contiene los coeficientes de los modelos GARCH, EGARCH y EWMA (GEW-LSTM). Teniendo en cuenta que la premisa de la metodología es obtener mejores pronósticos en la medida que se incluyen los coeficientes de todos los modelos GARCH, se esperaría que lo mismo ocurriera para el caso de la volatilidad de los log-retornos de la TRM. En este caso, incluir dentro del modelo GE-LSTM los coeficientes del modelo EWMA no genera pronósticos más precisos.

Modelo	MAE	MSE	HMAE	HMSE
LSTM	0,004535	0,00002109	0,754125	0,61939
<u>LSTM SF</u>	<u>0,000958</u>	<u>0,00000139</u>	<u>0,15905</u>	<u>0,04231</u>
E-LSTM	0,001055	0,00000136	0,187128	0,05363
E-LSTM SF	0,001491	0,00000248	0,260471	0,09084
GE-LSTM	0,000541	0,00000047	0,098624	0,02066
GE-LSTM SF	0,001876	0,00000382	0,315507	0,11780
GEW-LSTM	0,001518	0,00000265	0,254436	0,08064
GEW-LSTM SF	0,001260	0,00000208	0,219890	0,07477
<u>GARCH</u>	<u>0,017377</u>	<u>0,00030491</u>	<u>2,878457</u>	<u>8,75160</u>
<u>EGARCH</u>	<u>0,027603</u>	<u>0,00076646</u>	<u>4,560025</u>	<u>21,76906</u>
EWMA	0,018549	0,00034699	3,070720	9,93765

Tabla 4: Medidas de error para el pronóstico fuera de muestra (out-of-sample) de la volatilidad realizada de los log-retornos de la TRM. Los modelos identificados con SF corresponden a las versiones stateful.

Con respecto a las versiones stateless y stateful, es de esperarse que las últimas generarán pronósticos más precisos y con menores medidas de error. Lo anterior se observa para los modelos LSTM y GEW-LSTM, no así con los modelos GE-LSTM. Luego, los resultados obtenidos no permiten concluir que un modelo de red neuronal recurrente LSTM stateful genera los pronósticos más acertados en comparación con un modelo equivalente stateless.

Por otro lado, tomando como línea base las medidas del modelo GARCH, se calculan las mejoras porcentuales de cada medida para los modelos LSTM, de acuerdo a la

siguiente formula:

$$Mejora_{GARCH,m}^L = \frac{L_m - L_{GARCH}}{L_{GARCH}} * 100$$

Donde L indica una de las 4 medidas de error usadas (MAE, MSE, HMAE, HMSE), m es cada uno de los 8 modelos de red neuronal recurrente LSTM entrenados, L_m es el valor de la medida de error L seleccionada para el modelo m y L_{GARCH} el valor de la medida de error L para el modelo GARCH. Las mejoras porcentuales se muestran a continuación en la tabla 5.

De acuerdo a los resultado de esta tabla, se observa que la reducción de los errores para los modelos LSTM son en su mayoría mayores al 90 %, donde el modelo GE-LSTM versión stateless tienen reducciones superiores al 95 %; mientras que las del modelo LSTM stateful se encuentra sobre el 90 %, luego es posible afirmar de forma general que si se obtienen ganancias substanciales en la precisión al realizar pronósticos de la volatilidad realizada para los log-retornos de la TRM con los modelos recursivos LSTM diseñados según la metodología de [Kim y col. 2018](#).

Modelo	MAE	MSE	HMAE	HMSE
LSTM	-73,90 %	-93,08 %	-73,80 %	-92,92 %
LSTM SF	-94,48 %	-99,54 %	-94,47 %	-99,52 %
E-LSTM	-93,93 %	-99,55 %	-93,50 %	-99,39 %
E-LSTM SF	-91,42 %	-99,19 %	-90,95 %	-98,96 %
GE-LSTM	-96,89 %	-99,85 %	-96,57 %	-99,76 %
GE-LSTM SF	-89,21 %	-98,75 %	-89,04 %	-98,65 %
GEW-LSTM	-91,27 %	-99,13 %	-91,16 %	-99,08 %
GEW-LSTM SF	-92,75 %	-99,32 %	-92,36 %	-99,15 %

Tabla 5: Porcentaje de reducción de las diferentes medidas de error de los modelos LSTM comparados con las medidas del modelo GARCH.

Para la comparación de la precisión predictiva entre los modelos, se realizaron las pruebas Diebold-Mariano (DM) y de rangos con signo de Wilcoxon (WS); donde cada prueba se aplica a los valores pronosticados de los modelos estimados emparejándolos. Tanto para la prueba Diebold-Mariano como para la de rangos con signo de Wilcoxon, la hipótesis nula indica que los pronósticos de dos modelos tienen la misma capacidad de predicción.

	GE-LSTM SF	E-LSTM	E-LSTM SF	LSTM	LSTM SF	GEW-LSTM	GEW-LSTM SF	GARCH	EGARCH	EWMA
GE-LSTM	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GE-LSTM SF	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
E-LSTM					0.07	0.00	0.09	0.00	0.00	0.00
E-LSTM SF				0.00	0.00	0.44	0.00	0.00	0.00	0.00
LSTM					0.00	0.00	0.00	0.00	0.00	0.00
LSTM SF						0.00	0.00	0.00	0.00	0.00
GEW-LSTM						0.00	0.00	0.00	0.00	0.00
GEW-LSTM SF							0.00	0.00	0.00	0.00
GARCH								0.00	0.00	0.00
EGARCH									0.00	0.00

Tabla 6: p-valores para las pruebas Diebold-Mariano (DM) para la comparación de la precisión predictiva

	GE-LSTM SF	E-LSTM	E-LSTM SF	LSTM	LSTM SF	GEW-LSTM	GEW-LSTM SF	GARCH	EGARCH	EWMA
GE-LSTM	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GE-LSTM SF	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
E-LSTM										
E-LSTM SF				0.00	0.00	0.60	0.00	0.00	0.00	0.00
LSTM					0.00	0.00	0.00	0.00	0.00	0.00
LSTM SF						0.00	0.00	0.00	0.00	0.00
GEW-LSTM						0.00	0.00	0.00	0.00	0.00
GEW-LSTM SF							0.00	0.00	0.00	0.00
GARCH								0.00	0.00	0.00
EGARCH									0.00	0.00

Tabla 7: p-valores para las pruebas rangos con signo de Wilcoxon (WS) para la comparación de la precisión predictiva

En la tabla 6 se muestran los p-valores de la prueba Diebold-Mariano (DM), y en la tabla 7 los p-valores correspondientes a la prueba rangos con signo de Wilcoxon (WS).

Con respecto a las pruebas DM y considerando un nivel de significancia de 0,05, en todos los emparejamientos se rechaza la hipótesis nula de precisión predictiva similar, excepto para los modelos LSTM SF y E-LSTM (p-valor 0,07); y GEW-LSTM con E-LSTM SF (p-valor 0,44), es decir, todos los modelos generan pronósticos de diferente precisión, salvo los señalados.

Para la prueba WS y el mismo nivel de significancia de 0.05, igualmente en todos los emparejamientos se rechaza la hipótesis nula de capacidad predictiva equivalente, a excepción de los modelos GEW-LSTM y E-LSTM SF, donde se obtiene un p-valor de 0,60.

6. Mejoras al modelo

De acuerdo con [Kim y col. 2018](#), los pronósticos de los modelos de series de tiempo GARCH para la evaluación fuera de muestra se realizaron estimando dichos modelos únicamente con los datos de entrenamiento, es decir, estos modelos no se actualizaron con nueva información a medida que se creaban pronósticos para el periodo out-of-sample. Esto implica una desventaja, ya que una buena práctica es incluir la nueva información disponible a medida que el horizonte de tiempo va avanzado para que los pronósticos se ajusten a los movimientos de la variación del activo que se está analizando.

De igual forma, al no actualizarse los modelos GARCH, los parámetros de estos se mantienen fijos en el periodo de evaluación y no es posible incluir nueva información vía estos parámetros a los modelos de red neuronal recurrente LSTM.

Por otro lado, el modelo EWMA se estimó con un parámetro $\lambda = 0,97$, esto es adecuado para generar pronósticos con este modelo, pero este coeficiente no brinda información respecto a la dinámica de la volatilidad del activo.

Para corregir los inconvenientes anteriores, se propone como mejora a la metodología de [Kim y col. 2018](#), generar coeficientes dinámicos para los modelos GARCH, usando la técnica de ventanas deslizantes (*rolling windows*). De esta forma, los modelos se

reestimarían según avanza la ventana de estimación en el periodo de evaluación, y se tendrían coeficientes dinámicos, que se van actualizando a la nueva la información contenida en esta ventana de datos, así mismo se tendría un valor pronosticado para cada ventana.

Lo anterior implica que el modelo EWMA no se estimaría con un coeficiente fijo de 0,97, sino que este parámetro estaría sin restricción, pasando a ser un modelo GARCH(1,1) sin tendencia.

El tamaño de la ventana deslizante para los modelos de series de tiempo GARCH se estableció inicialmente en 90 días, sin embargo, durante la estimación se obtuvieron errores de convergencia. Luego se hizo necesario realizar varias estimaciones incrementando paulatinamente este tamaño. Se llegó a un valor final de 521 días, que corresponden a dos años. De esta forma, la primera ventana de estimación se estableció desde el 3 de enero de 2008 hasta el 30 de diciembre de 2009, para generar el primer pronóstico el 2 de enero de 2010, después la ventana de estimación se desplazaría un día para realizar el siguiente pronóstico.

De forma simultánea, se van apilando los coeficientes de los modelos GARCH a medida que estos se actualizan al avanzar la ventana de datos. Luego ya no se tendrían valores estáticos para estos coeficientes, sino una serie temporal con los distintos valores que toman los coeficientes según avanza la ventana de estimación.

Estas series de tiempo de los coeficientes de los modelos GARCH reemplazan a los coeficientes fijos en los datos de entrenamiento de los modelos de red neuronal recurrente LSTM usados anteriormente.

Los parámetros de la implementación computacional de los modelos de series de tiempo GARCH y LSTM no se modificaron en esta mejora, únicamente se cambió el método de estimación de los modelos GARCH, EGARCH y EWMA a *rolling window*.

Finalmente, el intervalo de evaluación de los pronósticos fuera de muestra se definió desde el 11 de julio de 2018 hasta el 31 de julio de 2019. Es decir, tanto en los modelos GARCH como en los de tipo LSTM se usó esta muestra para generar los pronósticos *out-of-sample*, teniendo en cuenta el avance de la ventana de datos ya descrito.

7. Resultados finales

Los valores de las medidas de error para 276 pronósticos, desde el 11 de julio 2018 hasta el 31 de julio 2019, con los modelos de la mejora propuesta se presentan en la tabla 8:

Modelo	MAE	MSE	HMAE	HMSE
LSTM	0,00198	0,0000043	0,3070	0,0985
LSTM SF	0,00128	0,0000022	0,1942	0,0465
E-LSTM	0,00287	0,0000092	0,4421	0,2034
E-LSTM SF	0,00081	0,0000010	0,1276	0,0243
GE-LSTM	0,00118	0,0000018	0,2126	0,0759
<i>GE-LSTM SF</i>	<i>0,00742</i>	<i>0,0000562</i>	<i>1,2621</i>	<i>1,8246</i>
GEW-LSTM	0,00327	0,0000114	0,5579	0,3759
GEW-LSTM SF	0,00188	0,0000041	0,2900	0,0909
GARCH	0,00067	0,0000008	0,1186	0,0314
EGARCH	0,00111	0,0000020	0,1896	0,0645
EWMA	0,00058	0,0000006	0,1069	0,0286

Tabla 8: Medidas de error para el pronóstico fuera de muestra (out-of-sample) obtenidos por los modelos mejorados. Los modelos identificados con SF corresponden a las versiones stateful.

Según los resultados anteriores, el modelo con las menores medidas de error es el EWMA (medidas resaltadas en negrita), mientras que el modelo con las mayores medidas de error es el GE-LSTM en versión stateful (resaltado en cursiva). Considerando únicamente los modelos recursivos LSTM, el que obtiene menores medidas de error es el E-LSTM en versión stateful (con resultados subrayados).

Comparando los valores de las medidas de error para los modelos LSTM, se puede observar que, salvo para el caso de los modelos GE-LSTM, las versiones stateful de los modelos tienen menores medidas que las versiones stateless. También puede observarse que, exceptuando el modelo GE-LSTM, cuando se van incorporando los coeficientes de los modelos GARCH en los modelos recursivos LSTM, se obtienen pronósticos menos precisos, lo contrario a lo observado con los modelos sin mejora.

Teniendo en cuenta los resultados del modelo E-LSTM, no es posible afirmar que incorporar la información de los modelos GARCH dentro de los modelos LSTM mejorados se obtengan pronósticos más precisos para el caso de la volatilidad realizada de

los log-retornos de la TRM.

Con base a las medidas del modelo EWMA, se calculan los incrementos porcentuales de los demás para cada medida, mostradas en la tabla 9.

Modelo	MAE	MSE	HMAE	HMSE
LSTM	239 %	650 %	187 %	244 %
LSTM SF	119 %	283 %	82 %	63 %
E-LSTM	392 %	1490 %	314 %	611 %
E-LSTM SF	38 %	70 %	19 %	-15 %
GE-LSTM	102 %	210 %	99 %	165 %
<i>GE-LSTM SF</i>	<i>1173 %</i>	<i>9660 %</i>	<i>1081 %</i>	<i>6282 %</i>
GEW-LSTM	461 %	1888 %	422 %	1215 %
GEW-LSTM SF	223 %	618 %	171 %	218 %
GARCH	14 %	36 %	11 %	10 %
EGARCH	91 %	242 %	77 %	125 %

Tabla 9: Porcentaje de reducción de las diferentes medidas de error de los modelos mejorados comparados con las medidas del modelo EWMA.

Según los datos de la tabla, el modelo E-LSTM versión stateful muestra una disminución porcentual del 15 % en la medida HMSE, mientras que para las demás presenta aumentos porcentuales, siendo el mayor en la medida MSE, con 70 %. Comparando los cambios porcentuales por medidas de error, puede observarse que las medidas HMAE y HMSE presentan menores incrementos con respecto a las correspondientes MAE y MSE. Así mismo, para los modelos LSTM los incrementos porcentuales mayores se observan en la medida MSE.

Como complemento a esta descripción, se presentan los cambios porcentuales de los modelos LSTM con respecto al modelo GARCH en la tabla 10, y al modelo EGARCH en la tabla 11. Para el caso del modelo GARCH, los resultados observados anteriormente se mantienen, donde el modelo E-LSTM tiene una disminución porcentual de 23 % en la medida HMSE, mientras que el mayor cambio porcentual se observa en la medida MSE, con 25 %. De igual forma el contraste entre las medidas HMAE y HMSE con las respectivas MAE y MSE, donde las primeras presentan menores incrementos.

En contraste con los anteriores comparativos, con el modelo EGARCH, el modelo E-LSTM presenta disminuciones porcentuales en todas sus medidas, con la mayor disminución en la medida HMSE (62 %) y la menor en la medida MAE (27 %). Com-

Modelo	MAE	MSE	HMAE	HMSE
LSTM	197 %	452 %	159 %	214 %
LSTM SF	92 %	182 %	64 %	48 %
E-LSTM	331 %	1070 %	273 %	548 %
E-LSTM SF	21 %	25 %	8 %	-23 %
GE-LSTM	77 %	128 %	79 %	142 %
<i>GE-LSTM SF</i>	<i>1014 %</i>	<i>7080 %</i>	<i>964 %</i>	<i>5714 %</i>
GEW-LSTM	391 %	1363 %	371 %	1098 %
GEW-LSTM SF	183 %	428 %	145 %	190 %

Tabla 10: Porcentaje de reducción de las diferentes medidas de error de los modelos mejorados comparados con las medidas del modelo GARCH.

parando las medidas del modelo E-LSTM, se observa que en este caso la medida MSE es la segunda con mayor disminución porcentual, mientras que en el caso del contraste con el modelo EWMA, esta era la que presentaba peor desempeño con los mayores incrementos porcentuales.

Modelo	MAE	MSE	HMAE	HMSE
LSTM	78 %	119 %	62 %	53 %
LSTM SF	15 %	12 %	2 %	-28 %
E-LSTM	158 %	364 %	133 %	216 %
E-LSTM SF	-27 %	-50 %	-33 %	-62 %
GE-LSTM	6 %	-9 %	12 %	18 %
<i>GE-LSTM SF</i>	<i>568 %</i>	<i>2750 %</i>	<i>566 %</i>	<i>2731 %</i>
GEW-LSTM	194 %	481 %	194 %	483 %
GEW-LSTM SF	69 %	110 %	53 %	41 %

Tabla 11: Porcentaje de reducción de las diferentes medidas de error de los modelos mejorados comparados con las medidas del modelo EGARCH.

La figura 8 muestra la volatilidad realizada calculada y los pronósticos estimados a partir de los modelos EWMA y E-LSTM SF, en el intervalo de evaluación fuera de muestra.

Finalmente, se muestran los resultados de las pruebas de precisión predictiva Diebold-Mariano (DM) y de rangos con signo de Wilcoxon (WS) en las tablas 12 y 13 respectivamente. Para las pruebas DM, se rechaza la hipótesis nula de igual precisión predictiva de los modelos con un nivel de significancia de 0,05, salvo para el caso de LSTM SF y

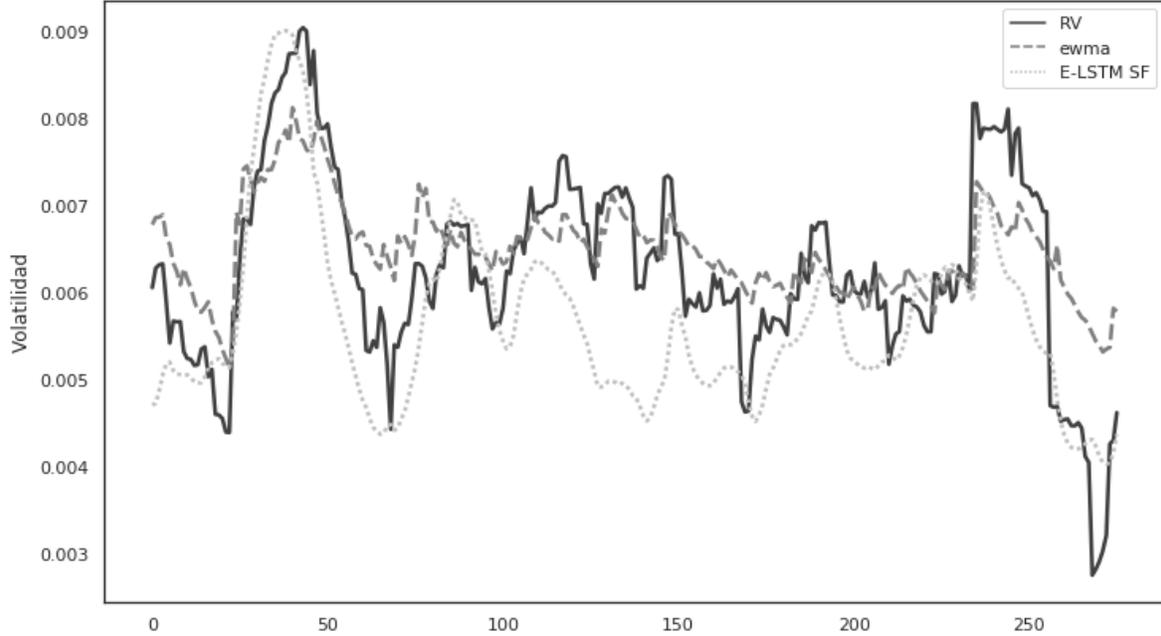


Figura 8: Volatilidad realizada calculada y pronósticos estimados con los modelos mejorados

GE-LSTM (p-valor 0,18), y EGARCH con GE-LSTM (p-valor 0,34). Con respecto a la prueba WS y con el mismo nivel de significancia, todos los modelos evaluados tienen diferente capacidad predictiva a excepción de EGARCH y GARCH (p-valor 0,08) y EWMA con GARCH (p-valor 0,37).

	GE-LSTM SF	E-LSTM	E-LSTM SF	LSTM	LSTM SF	GEW-LSTM	GEW-LSTM SF	GARCH	EGARCH	EWMA
GE-LSTM	0.00	0.00	0.00	0.00	0.18	0.00	0.00	0.00	0.34	0.00
GE-LSTM SF		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
E-LSTM			0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
E-LSTM SF				0.00	0.00	0.00	0.00	0.01	0.00	0.00
LSTM					0.00	0.00	0.00	0.00	0.00	0.00
LSTM SF						0.00	0.00	0.00	0.02	0.00
GEW-LSTM							0.00	0.00	0.00	0.00
GEW-LSTM SF								0.00	0.00	0.00
GARCH									0.00	0.00
EGARCH										0.00

Tabla 12: p-valores para las pruebas Diebold-Mariano (DM) para la comparación de la precisión predictiva de los modelos mejorados

	GE-LSTM SF	E-LSTM	E-LSTM SF	LSTM	LSTM SF	GEW-LSTM	GEW-LSTM SF	GARCH	EGARCH	EWMA
GE-LSTM	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GE-LSTM SF		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
E-LSTM			0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
E-LSTM SF				0.00	0.00	0.00	0.00	0.00	0.00	0.00
LSTM					0.00	0.00	0.00	0.00	0.00	0.00
LSTM SF						0.00	0.00	0.00	0.00	0.00
GEW-LSTM							0.00	0.00	0.00	0.00
GEW-LSTM SF								0.00	0.00	0.00
GARCH									0.08	0.37
EGARCH										0.01

Tabla 13: p-valores para las pruebas rangos con signo de Wilcoxon (WS) para la comparación de la precisión predictiva de los modelos mejorados

8. Discusión de los resultados

Los ajustes propuestos en la mejora para la metodología de [Kim y col. 2018](#) buscan incrementar la precisión de las estimaciones generadas por los modelos de red neuronal recurrente LSTM al incluir nueva información relevante por medio de la actualización de los coeficientes de los modelos de series de tiempo GARCH.

Más precisamente, al realizar la estimación de los modelos GARCH por medio de ventanas deslizantes, se espera que sus parámetros se actualicen de acuerdo a los datos contenidos en la ventana de estimación. Esto permitiría capturar más eficientemente los efectos observados de la volatilidad de los log-retornos de la TRM, y transferir esta información vía dichos coeficientes a los modelos recurrentes LSTM.

Con respecto a los resultados obtenidos con los modelos aplicando esta mejora, según las medidas de error obtenidas se concluye que el modelo recurrente LSTM con los pronósticos más precisos es el E-LSTM versión stateful. Sin embargo, este tiene medidas de error mayores al modelo EWMA, el cual es el que presenta los pronósticos más precisos. Los demás modelos de red recurrente LSTM generan pronósticos con errores superiores a los del modelo EWMA, contrario a los resultados obtenidos aplicando la metodología original.

Una posible influencia en el aumento de los errores de pronóstico calculados con los modelos recurrentes LSTM mejorados puede ser la inestabilidad de los datos, detectada por los errores de convergencia obtenidos al realizar la estimación por ventanas deslizantes de los modelos GARCH. Puesto que en la metodología de [Kim y col. 2018](#) los modelos GARCH se estimaron con toda la muestra de entrenamiento, estos errores de convergencia no se detectaron hasta utilizar la mejora por ventanas deslizantes.

Adicionalmente, la estimación de los modelos GARCH con ventanas deslizantes permite eliminar posibles sesgos por información pasada que ya no es relevante, facilitando obtener unos pronósticos que reaccionan mucho más rápido a la nueva información, particularmente el modelo EWMA, lo cual posibilita ajustar los pronósticos de forma ágil.

En lo referente a la metodología de [Kim y col. 2018](#) para el pronóstico de la volatilidad realizada de los log-retornos de la TRM, si bien estos concuerdan con los resultados de la referencia, no lo hacen con el modelo de mejor desempeño, teniendo en cuenta que fue el modelo GEW-LSTM el que obtenía mejores resultados en el trabajo original.

Sin embargo, para el caso de la TRM el modelo con mejor desempeño fue el GE-LSTM.

Lo anterior es posible verificar contrastando las medidas de error calculadas para el modelo GE-LSTM y GEW-LSTM. Las medidas correspondientes al modelo GE-LSTM son en promedio un 29 % del valor de las medidas observadas para el modelo GEW-LSTM. Esto puede ser consecuencia de ruido inducido al incluir los coeficientes EWMA en el modelo GE-LSTM, o que el modelo GEW-LSTM requiere ajustar su especificación, alterando el número de unidades y capas ocultas.

Para las versiones stateful de los modelos, el resultado es el opuesto. Los errores del modelo GEW-LSTM SF son en promedio un 36 % menores a las medidas de error para el modelo GE-LSTM SF. Este escenario podría explicarse por el efecto de los modelos stateful, es decir, que mantienen su «memoria» a través de toda la muestra de datos. Luego, en la versión stateful, es posible que el modelo GEW-LSTM sea capaz de usar más eficientemente la información disponible para generar pronósticos más precisos.

Sin embargo, las medidas de error del modelo GEW-LSTM SF contrastadas con las del modelo GE-LSTM son en promedio un 315 % mayores, luego el efecto del modelo stateful no compensa la posible especificación errónea del modelo GEW-LSTM SF.

Por otro lado, considerando que las medidas de error corregidas para heterocedasticidad HMAE y HMSE fueron en las que el modelo E-LSTM tuvo mejor desempeño, especialmente HMSE, estas pueden dar un indicio de las capacidades de los modelos recurrentes LSTM para el pronóstico de la volatilidad. Tal como indican [Kim y col. 2018](#), y sobre todo [Kristjanpoller y col. 2016](#), que utilizan únicamente estas medidas de error para evaluar sus pronósticos, tales medidas son más adecuadas para medir el desempeño de modelos no lineales, como en el caso de la estimación de la volatilidad.

Los resultados dispares de los modelos recurrentes LSTM comparados con los resultados de la metodología ajustada con ventanas deslizantes muestran indicaciones acerca de la aplicabilidad de un mismo diseño para el análisis de diferentes activos. Tal como es mostrado por [Kristjanpoller y col. 2016](#), un mismo modelo de red neuronal no se desempeña de la misma forma cuando es aplicado a varios activos. Luego, para optimizar los pronósticos que generan este tipo de modelos se requiere el afinamiento tanto en su diseño, seleccionando el número de capas usadas, unidades en cada capa; como en los hiperparámetros, tales como el algoritmo de optimización usado durante el entrenamiento, la tasa de aprendizaje, el número de iteraciones o *epochs*; así como en la selección de las variables.

Una limitación que implica utilizar variables exógenas para el pronóstico de una serie temporal en una aplicación práctica, es que son relevantes únicamente para pronósticos de un día. Realizar pronósticos para un intervalo mayor (por ejemplo 7 o 15 días adelante) acarrea imponer una forma funcional o proceso generador de datos para estas variables exógenas, lo cual es una fuente adicional de incertidumbre. Esto crea una dicotomía entre crear un modelo de pronóstico que considere únicamente la serie temporal que se está analizando y limitar sus resultados, o enriquecer las estimaciones utilizando otras variables relevantes y estar sujeto a la incertidumbre adicional que implica estimar valores futuros para estas.

9. Conclusiones

Siguiendo la metodología establecida en [Kim y col. 2018](#) para el pronóstico de la volatilidad realizada de los log-retornos de la TRM, se crean varios modelos de red neuronal recurrente LSTM, aumentados con los coeficientes de modelos de series de tiempo GARCH, EGARCH y EWMA. Estos coeficientes son incluidos en los modelos recurrentes por medio de los datos de entrenamiento.

Utilizar modelos de series de tiempo GARCH permite incluir hechos relevantes del comportamiento de la volatilidad, tales como los clústeres y las asimétricas causados por la dinámica de los precios del activo analizado. Por otro lado, los modelos recurrentes LSTM capturan los patrones y relaciones Inter temporales de la volatilidad. Luego, combinar estos dos tipos de modelos permite obtener estimaciones más precisas de las variaciones de la TRM.

Una vez entrenados tanto los modelos de series de tiempo GARCH como los modelos de red neuronal recurrente LSTM, se calculan estimaciones de la volatilidad realizada para 276 días. Estos pronósticos se evalúan utilizando 4 medidas de error. Los pronósticos calculados con los modelos de redes neuronales recurrentes LSTM son los que presentan mejor desempeño de acuerdo a las medidas de error calculadas, siendo el modelo GE-LSTM el más preciso.

Se propone una mejora a la metodología actualizando los coeficientes de los modelos GARCH de forma dinámica, estimándolos por medio de la técnica de ventanas deslizantes, e incorporándolos a la muestra de datos out-of-sample, para luego incorporar dichos coeficientes como datos de entrada a los modelos de tipo LSTM.

Para los pronósticos calculados con esta mejora, los modelos recurrentes se ven relegados por los modelos de series de tiempo GARCH, siendo el modelo E-LSTM versión stateful el que genera los pronósticos más precisos, considerando únicamente los modelos recurrentes LSTM. Sin embargo, las estimaciones de este modelo generan medidas de error mayores en comparación con el modelo EWMA.

Finalmente, con los resultados observados para el pronóstico de la volatilidad realizada de los log-retornos de la TRM, es posible concluir que existe información relevante que es capturada por los modelos GARCH y EGARCH, la cual mejora los pronósticos estimados por los modelos de red neuronal recurrente LSTM cuando estos incluyen entre sus variables explicativas los coeficientes de los modelos GARCH y EGARCH.

Una posible opción para aprovechar las ventajas de un modelo de red neuronal recurrente LSTM potenciado con modelos GARCH es realizar el pronóstico de la volatilidad realizada utilizando la serie temporal, y complementarla con pronósticos calculados a partir de modelos de series de tiempo GARCH, bien sea de la misma serie temporal o de otras variables relevantes, de tal forma que no se utilizarían los coeficientes de estos modelos, sino sus valores pronosticados como datos de entrenamiento. Lo anterior también permitiría simplificar la estructura de la red neuronal recurrente LSTM, reduciendo el número de unidades en las capas, y posiblemente el número de capas que componen esta red.

Con respecto al cálculo de pronósticos para un horizonte mayor a un día, se deben considerar varios escenarios; realizar un pronóstico día a día, utilizando el valor del primer pronóstico obtenido para estimar el siguiente; o aprovechar las capacidades de predicción de secuencias de los modelos de red neuronal recurrente LSTM para predecir en una iteración, una sucesión de días para el horizonte de tiempo que se busca estimar.

Referencias

- Andersen, Torben G y col. (2001). «The distribution of realized stock return volatility». En: *Journal of financial economics* 61.1, págs. 43-76.
- Banco de la República (s.f.). *Tasa Representativa del Mercado*. <https://www.banrep.gov.co/es/estadisticas/trm>. Accedido 08-11-2019.
- Brownlee, Jason (2017). *Multivariate Time Series Forecasting with LSTMs in Keras*. <https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/>. Accedido 02-07-2019.
- Bucci, Andrea (2019). *Realized Volatility Forecasting with Neural Networks*. MPRA Paper. URL: <https://ideas.repec.org/p/pramprapa/95443.html>.
- Chollet, François y col. (2015). *Keras*. <https://keras.io>.
- Corredor Velandia, César Augusto y Stefano Vega Mazzeo (2012). «Análisis de corto plazo del contagio de variables y noticias financieras en estados unidos y Colombia». En: *Revista de Economía del Caribe* 9.
- Engle, Robert F y Andrew J Patton (2007). «What good is a volatility model?» En: *Forecasting volatility in the financial markets*. Elsevier, págs. 47-63.
- Fischer, Thomas y Christopher Krauss (2018). «Deep learning with long short-term memory networks for financial market predictions». En: *European Journal of Operational Research* 270.2, págs. 654-669.
- Géron, Aurélien (2017). *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Inc.
- Kim, Ha Young y Chang Hyun Won (2018). «Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models». En: *Expert Systems with Applications* 103, págs. 25-37.
- Kingma, Diederik P. y Jimmy Ba (2014). *Adam: A Method for Stochastic Optimization*. arXiv: [1412.6980 \[cs.LG\]](https://arxiv.org/abs/1412.6980).
- Kristjanpoller, Werner y Marcel C Minutolo (2016). «Forecasting volatility of oil price using an artificial neural network-GARCH model». En: *Expert Systems with Applications* 65, págs. 233-241.
- Nelson, David MQ, Adriano CM Pereira y Renato A de Oliveira (2017). «Stock market's price movement prediction with LSTM neural networks». En: *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, págs. 1419-1426.
- Olah, Christopher (2015). *Understanding LSTM networks*. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Accedido 06-11-2019.

- Parra Barrios, Alberto (2019). «Impacto de las decisiones de política monetaria de la FED en indicadores de la economía colombiana durante el periodo 2007-2015». En: *Revista Finanzas y Política Económica*, Vol. 11, no. 1 (ene.-jun.) p. 149-182.
- Petneházi, Gábor y József Gáll (2019). «Exploring the predictability of range-based volatility estimators using recurrent neural networks». En: *Intelligent Systems in Accounting, Finance and Management*.
- Sheppard, Kevin (2019). *Financial Econometrics Notes*. <https://www.kevinsheppard.com/teaching/mfe/notes/>.
- Siami-Namini, Sima, Neda Tavakoli y Akbar Siami Namin (2018). «A Comparison of ARIMA and LSTM in Forecasting Time Series». En: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, págs. 1394-1401.
- Srivastava, Nitish y col. (2014). «Dropout: a simple way to prevent neural networks from overfitting». En: *The journal of machine learning research* 15.1, págs. 1929-1958.
- Suárez, Álvaro Andrés Cámara y col. (2006). «Una aproximación empírica en la relación entre las tasas de interés de los TES TF y el tipo de cambio en Colombia». En: *INNOVAR. Revista de Ciencias Administrativas y Sociales* 16.27, págs. 47-55.