



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

ADE

Facultad de Administración
y Dirección de Empresas /UPV

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Facultad de Administración y Dirección de Empresas

Diseño y evaluación de la capacidad predictiva de los
modelos econométricos clásicos frente al modelo basado
en redes neuronales Long-Short Term Memory aplicados al
mercado FOREX

Trabajo Fin de Grado

Grado en Administración y Dirección de Empresas

AUTOR/A: Abad Abad, José Luis

Tutor/a: Oliver Muncharaz, Javier

CURSO ACADÉMICO: 2022/2023

**Diseño y evaluación de la capacidad predictiva de los modelos
econométricos clásicos frente al modelo basado en redes neuronales
Long-Short Term Memory aplicados al mercado FOREX.**

Autor: **JOSÉ LUIS ABAD ABAD**

Tutor: **JAVIER OLIVER MUNCHARAZ**

Trabajo Fin de Grado presentado en la Facultad de Administración y Dirección de Empresas de la Universitat Politècnica de València, para la obtención del Título de Graduado en Administración y Dirección de Empresas.

Curso 2022/2023

Valencia, 6 de junio de 2023.

Agradecimientos

En primer lugar, a mis padres, por su constante apoyo y confianza. A mis hermanos, Miguel y Diego que me hacen mejorar cada día y me dan aliento para continuar. Agradecer a mis amigos que empezaron conmigo desde el primer año de residencia y a mis compañeros de clase, he aprendido de todos ellos un poco. También a mi profesor, Javier Oliver, por aconsejarme cuando lo necesitaba y guiarme durante estos meses.

RESUMEN

En este trabajo se compararán los resultados de dos modelos econométricos clásicos, como son el modelo exponencial simple y el modelo ARIMA con respecto a un modelo de predicción moderno basado en redes neuronales recurrentes Long-Short Term Memory (LSTM). Las series temporales analizadas serán datos históricos de tres tipos de cambio de divisas del mercado FOREX como son el Euro-dólar, Euro-libra y Euro-Dólar Neozelandés. Además, será de interés estudiar la capacidad predictiva de los modelos empleando distintos espacios temporales o ‘time-frames’: diarios, semanales y mensuales. Anticiparse al mercado de divisas permite a empresas y estados realizar operaciones de cobertura con el fin de protegerse frente a los riesgos presentes en los mercados de bienes y servicios internacionales. Asimismo, permite a especuladores realizar operaciones de arbitraje entre mercados y a agentes económicos gubernamentales regular la liquidez de sus propias monedas.

Palabras clave: Predicción; Modelización; Econometría; Forex; Suavizado exponencial; ARIMA; LSTM; Python

ABSTRACT

This project tries to assess the performance in terms of accuracy through the comparison of historical real life Forex Market rentability with the predictions obtained by the implementation in Python of traditional econometrical predicting algorithms as Simple Exponential Smoothing and ARIMA in front of modern Machine Learning algorithms, specifically Long-Short Term Memory Neuronal Network. Also, different time-frames of data (diary, weekly and monthly) are compared in the project so conclusions can be made about the influence of time while predicting Forex Market rentability.

The aim is to analyse if modern algorithms present any sort of gain versus traditional algorithms. Processing the results employing error metrics as Root Mean Square Error and Mean Absolute Error will let us compare the accuracy of the predictions. Reaching models with high accuracy it's key for economical agents such as investors, governments and monetary authorities that trade in international markets or manage the monetary policy regulating the liquidity of their currency.

Key words: Prediction; Modelization; Econometrics; Forex; Exponential smoothing; ARIMA; LSTM; Python

RESUM

En aquest treball es compararan els resultats de dos models econòmics clàssics, com són el model exponencial simple i el model ARIMA respecte a un model de predicció modern basat en xarxes neuronals recurrents Long-Short Term Memory (LSTM). Les sèries temporals analitzades seran dades històriques de tres tipus de canvi de divises del mercat FOREX com són l'Euro-dòlar, Euro-lliura i Euro-Dòlar Neozelandés. A més, serà d'interés estudiar la capacitat predictiva dels models emprant diferents espais temporals o 'time-frames': diaris, setmanals i mensuals. Anticipar-se al mercat de divises permet a empreses i estats realitzar operacions de cobertura amb la finalitat de protegir-se enfront dels riscos presents en els mercats de béns i serveis internacionals. Així mateix, permet a especuladors realitzar operacions d'arbitratge entre mercats i a agents econòmics governamentals regular la liquiditat de les seues pròpies monedes.

Paraules clau: Predicció; Modelització; Econometria; Forex; Suavitzat exponencial; ARIMA; LSTM; Python

Índice general

RESUMEN	3
ABSTRACT	4
RESUM	5
1. INTRODUCCIÓN Y OBJETIVOS	11
1.1 Introducción	11
1.2 Objetivos	12
1.3 Distribución temporal de las actividades	13
2. METODOLOGÍA	15
2.1 Modelo clásico: suavizado exponencial	15
2.2 Modelo clásico: ARIMA	18
2.3 Modelo moderno: redes neuronales recurrentes LSTM	21
2.4 Métricas de error: RMSE y MAE	27
3. DISEÑO DE MODELOS DE PREDICCIÓN EN EL MERCADO FOREX	29
3.1 Origen de los datos	29
3.2 Modelización en Python	33
3.2.1 Importación de datos y cálculo de la rentabilidad logarítmica	33
3.2.2 Implementación suavizado exponencial en Python	36
3.2.3 Implementación ARIMA en Python	38
3.2.4 Implementación red neuronal LSTM en Python	40
4. EVALUACIÓN DE LA CAPACIDAD PREDICTIVA DE LOS MODELOS APLICADOS AL MERCADO FOREX	43
4.1 Predicción de las series temporales con datos diarios	43
4.2 Serie de datos semanales	48
4.3 Serie de datos mensuales	52
5. CONCLUSIONES	56
6. COMPROMISO, ODS Y LÍNEAS FUTURAS	58
6.1 Compromiso y ODS	58
6.2 Líneas futuras	59
7. BIBLIOGRAFÍA	60
8. ANEXO I	63
9. ANEXO 2	65

Índice de figuras

Figura 1. Captura función Autocorrelación muestra EUR/USD diario.	19
Figura 2. Captura función Autocorrelación Parcial muestra EUR/USD diario.	20
Figura 3. Diagrama estructura LSTM (Wenjie Lu et al, 2020).....	22
Figura 4. Función Sigmoide.....	23
Figura 5. Función tangente hiperbólica.....	24
Figura 6. Cálculo del RMSE, figura tomada de Roondiwala et al. 2017.....	27
Figura 7. Precio de cierre de la cotización de la serie EUR/USD diaria.....	31
Figura 8. Precio de cierre de la cotización de la serie EUR/GBP diaria.....	32
Figura 9. Precio de cierre de cotización de la serie EUR/NZD diaria.....	32
Figura 10. Rentabilidad compuesta de la serie EUR/USD diaria.....	34
Figura 11. Rentabilidad compuesta de la serie EUR/GBP semanal.....	34
Figura 12. Rentabilidad compuesta de la serie EUR/NZD mensual.....	34
Figura 13. Bucle diseñado para obtener el vector de observaciones con time-frame diario.	35
Figura 14. Mecanismo de desplazamiento de muestras de una ventana móvil.....	35
Figura 15. Diagrama de flujo modelo suavizado exponencial.....	37
Figura 16. Extracto de la ejecución del modelo ARIMA.....	38
Figura 17. Diagrama de flujo algoritmo modelo ARIMA.....	39
Figura 18. Diagrama de flujo algoritmo Red Neuronal LSTM.....	42
Figura 19. Comparación de los datos reales EUR/USD con la predicción del modelo de suavizado exponencial.	44
Figura 20. Comparación de los datos reales EUR/USD con la predicción del modelo LSTM. ..	45
Figura 21. Resultados RMSE en time-frames diarios.	47
Figura 22. Resultados ARIMA serie temporal EUR/NZD semanal.	48
Figura 23. Resultados LSTM en la serie EUR/NZD semanales.....	49
Figura 24. Resultados LSTM en la serie EUR/GBP semanales.....	49
Figura 25. Gráfica 3D de los resultados obtenidos con data-frames semanales.	51
Figura 26. Resultados EUR/NZD ARIMA mensual.	53
Figura 27. Resultados EUR/NZD LSTM mensual.	53
Figura 28. Gráfica resumen del error RMSE en función del modelo y el time-frame.	55

Índice de tablas

Tabla 1. Diagrama Gantt planificación de tareas.....	14
Tabla 2. Resumen número de muestras de cada serie temporal en función del time-frame.	30
Tabla 3. Resumen estadísticos media y desviación típica de las series temporales.....	31
Tabla 4. Especificaciones e hiperparámetros de la Red Neural LSTM.....	41
Tabla 5. Número de muestras de las series temporales diarias de cada tipo de cambio.	44
Tabla 6. Resumen de los errores cometidos RMSE y MAE al utilizar datos diarios de la serie EUR/USD.	46
Tabla 7. Resumen de los errores cometidos RMSE y MAE al utilizar datos diarios de la serie EUR/GBP.	46
Tabla 8. Resumen de los errores cometidos RMSE y MAE al utilizar datos diarios de la serie EUR/NZD.	47
Tabla 9. Número de muestras de las series temporales semanales de cada tipo de cambio.....	48
Tabla 10. Resumen de los errores RMSE y MAE al utilizar datos semanales de la serie EUR/USD.	50
Tabla 11. Resumen de los errores RMSE y MAE al utilizar datos semanales de la serie EUR/GBP.	50
Tabla 12. Resumen de los errores RMSE y MAE al utilizar datos semanales de la serie EUR/NZD.	50
Tabla 13. Número de muestras de las series temporales mensuales.	52
Tabla 14. Resumen de los errores RMSE y MAE al utilizar datos mensuales de la serie EUR/USD.	53
Tabla 15. Resumen de los errores RMSE y MAE al utilizar datos mensuales de la serie EUR/GBP.	54
Tabla 16. Resumen de los errores RMSE y MAE al utilizar datos mensuales de la serie EUR/NZD.	54

Índice de ecuaciones

Ecuación 1. y 1.2 Formulación matemática del suavizado exponencial simple.....	16
Ecuación 2. Formulación matemática suavizado exponencial doble.	17
Ecuación 3. Formulación matemática modelo ARIMA.....	19
Ecuación 4. Función sigmoide.	23
Ecuación 5. Función tangente hiperbólica.....	24
Ecuación 6. Formulación matemática función ReLU.....	24
Ecuación 7. Formulación Root Mean Square Error	27
Ecuación 8. Formulación matemática Mean Absolute Error.....	28
Ecuación 9. Formulación rentabilidad compuesta	33

Índice de acrónimos

FOREX	Foreign Exchange
OTC	Over the Counter
GMT	Greenwich Mean Time
MAJOR	Pares de divisas más negociados
ML	Machine Learning, aprendizaje automático en su traducción al castellano
EUR	Euro
USD	Dólar estadounidense
GBP	Libra esterlina
NZD	Dólar neozelandés
EUR/USD	Par euro-dólar estadounidense
EUR/GBP	Par euro-libra
EUR/NZD	Par euro-dólar neozelandés
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
ARIMA	Autoregressive Integrated Moving Average
AIC	Akaike Information Criteria
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
NAN	Not a Number
TFG	Trabajo Fin de Grado

1. INTRODUCCIÓN Y OBJETIVOS

1.1 Introducción

El poder adquisitivo de los países está representado principalmente en sus monedas. Éstas son un reflejo del intercambio de bienes y servicios que se realizan tanto a nivel nacional como internacional (Montoya *et al.* 2022). La presencia de dichas monedas en el escenario internacional hace obligatoria la aparición del Mercado internacional de divisas Forex. En dicho mercado se realiza una comparación de la dominancia entre pares de monedas. Las fluctuaciones vienen dadas por cambios en las políticas macroeconómicas, decisiones políticas o incluso el riesgo asociado al país emisor (situación social, terrorismo...etc) (Baillie & McMahon, 1990). El mercado Forex es un mercado no organizado u 'OTC' (Over the Counter) que se caracteriza por ser un mercado descentralizado y con menor regulación debido a la laxitud en su supervisión. El mercado Forex está activo 24 horas al día y 5 días a la semana. La apertura en horario GMT es el domingo a las 22:00 y el cierre, el viernes a las 21:00.

Las monedas que más representación tienen en el mercado (por su demanda) son el dólar estadounidense (USD), el Euro (EUR), la Libra esterlina (GBP) y el Yuan Chino (CNY). Estas monedas se conocen como 'MAJOR' y están controladas por los principales bancos centrales de las economías mundiales, como es el caso de la Reserva Federal, el Banco Central Europeo, el banco de Inglaterra o el Banco Popular de China. Solamente el dólar, está envuelto en las tres cuartas partes de todas las transacciones diarias (Lee & Chinn, 1998) (King, 2012). Por contraposición, existen monedas menos negociadas y asociadas a países en vías de desarrollo que ofrecen menor liquidez, es el caso de divisas menores y exóticas. Ejemplos de monedas exóticas son el dólar neozelandés (NZD), dólar taiwanés (TWD), el riyal de Arabia Saudí (SAR)...etc.

El mercado Forex movió en el pasado año (2022) casi seis trillones de dólares diariamente, lo que hace una media de 250 000 millones por hora. Analizar este mercado es especialmente atractivo para aquellos que utilizan el intercambio de monedas con el fin de obtener ganancias (Frankel & Froot, 1990). Esta práctica se ha vuelto muy frecuente gracias a la facilidad de acceder a este tipo de mercado a través de plataformas online de trading que únicamente precisan de conexión a internet para poder realizar operaciones en tiempo real. Es por esta razón que el pronóstico de los tipos de cambios es especialmente atractivo para aquellos agentes económicos que están involucrados en los mercados internacionales y necesitan anticiparse para proteger sus inversiones. También supone un desafío estadístico, pues son muchos los modelos que se han propuesto para tratar de generar las predicciones que mejor se ajusten a la serie temporal en cuestión.

1.2 Objetivos

El objetivo de este trabajo es comparar los resultados de distintos modelos de predicción clásicos y modernos con el fin de discernir el modelo que menor error comete a la hora de predecir los rendimientos de las divisas. Para ello se calculará la rentabilidad logarítmica de tres tipos de cambio como el euro-dólar cuya abreviatura es EUR/USD, euro-libra esterlina (EUR/GBP) y euro-dólar neozelandés (EUR/NZD).

La rentabilidad compuesta es la forma recomendada de calcular la rentabilidad asociada a activos bursátiles, en este caso las divisas. Será calculada mediante diferencias entre los logaritmos de los precios de cierre de dos muestras consecutivas en el tiempo. Calcular la rentabilidad compuesta o continua, permite al analista obtener un valor de la rentabilidad más próximo a su valor real ya que a diferencia de la rentabilidad simple, ésta se aproxima mejor cuando hay variaciones pequeñas de la cotización entre dos muestras consecutivas en tiempo.

Adicionalmente, será práctico analizar cómo influye el eje de tiempos (diario, semanal y mensual) o también denominado 'time-frame'. Mediante la comparación de los errores cometidos en las predicciones de cada modelo se podrá discernir si existe algún modelo que tienda a ofrecer mejores predicciones cuando se introducen muestras que representan intervalos de tiempo más amplios como pueden ser mensuales y por tanto se ajustará mejor a las necesidades de un inversor largoplacista. También puede ser en sentido inverso y ofrecer mejores predicciones en time-frames diarios, ideales para inversores cortoplacistas.

Se utilizarán dos modelos clásicos de predicción, el suavizado exponencial y el modelo ARIMA. Junto con un modelo moderno, las redes neuronales LSTM. Los tres modelos son modelos univariantes, es decir, se basarán en las propias muestras pasadas para generar las predicciones.

Para cuantificar la bondad de la predicción y poder medir los errores cometidos entre los valores reales y la predicción del modelo, se empleará el RMSE y el MAE. Utilizar dos métodos de cálculo de error nos permitirá alcanzar interpretaciones de los resultados más fiables.

1.3 Distribución temporal de las actividades

La propuesta de realizar el TFG nace del interés en los mercados y los activos financieros más complejos como son los productos derivados: contratos Forward, Opciones exóticas, Productos estructurados, SWAP's...etc. En septiembre de 2022 comienza la documentación con el libro de Xavier Puig: Comprender los productos financieros (2015). Una vez conocidos los mercados OTC, se propone realizar un análisis comparativo entre distintos modelos de predicción, dos modelos econométricos estudiados en la asignatura de econometría (suavizado exponencial y ARIMA) y otro modelo, redes neuronales LSTM. Este último modelo necesitó más documentación pues las redes LSTM tienen unas características distintas al resto de modelos clásicos, por ejemplo, necesita un entrenamiento. Asimismo, fue necesario realizar un proceso de aprendizaje sobre las herramientas Python empleadas en la programación de los modelos.

Las principales líneas de actuación fueron la programación y la documentación bibliográfica. El seguimiento de los avances se realizó a través de informes resumen donde se incluía una actualización del trabajo realizado hasta la fecha. Tras cada informe se concertaba una reunión presencial o por TEAMS para valorar las líneas futuras del trabajo y resolver las dudas que habían surgido.

El primer borrador del TFG salió a la luz en la última semana de marzo, desde entonces se han realizado 4 revisiones hasta llegar a la versión final.

A continuación, se muestra un diagrama de Gantt que incluye la planificación temporal de las tareas durante los meses de realización del TFG.

T1: Documentación sobre el uso de las librerías que implementan los modelos de predicción en Python.

T2: Documentación bibliográfica de los aspectos teóricos de la metodología, así como información útil sobre el análisis de series temporales, mercados OTC...etc.

T3: Programación I y II. En el transcurso del desarrollo de los algoritmos de los modelos de predicción hubo dos grandes fases, ya que el enfoque inicial no fue del todo acertado con la red neuronal LSTM.

T4: Ejecución de códigos y obtención de resultados como gráficas y métricas de error.

T5: Informe de resultados I y II. Engloba todos los resultados obtenidos durante el proceso de documentación previos a la realización del primer borrador.

T6: Elaboración de los sucesivos borradores de la memoria del TFG.

T7: Elaboración de la presentación Power Point para la defensa del trabajo.

T8: Reuniones presenciales entre alumno y profesor.

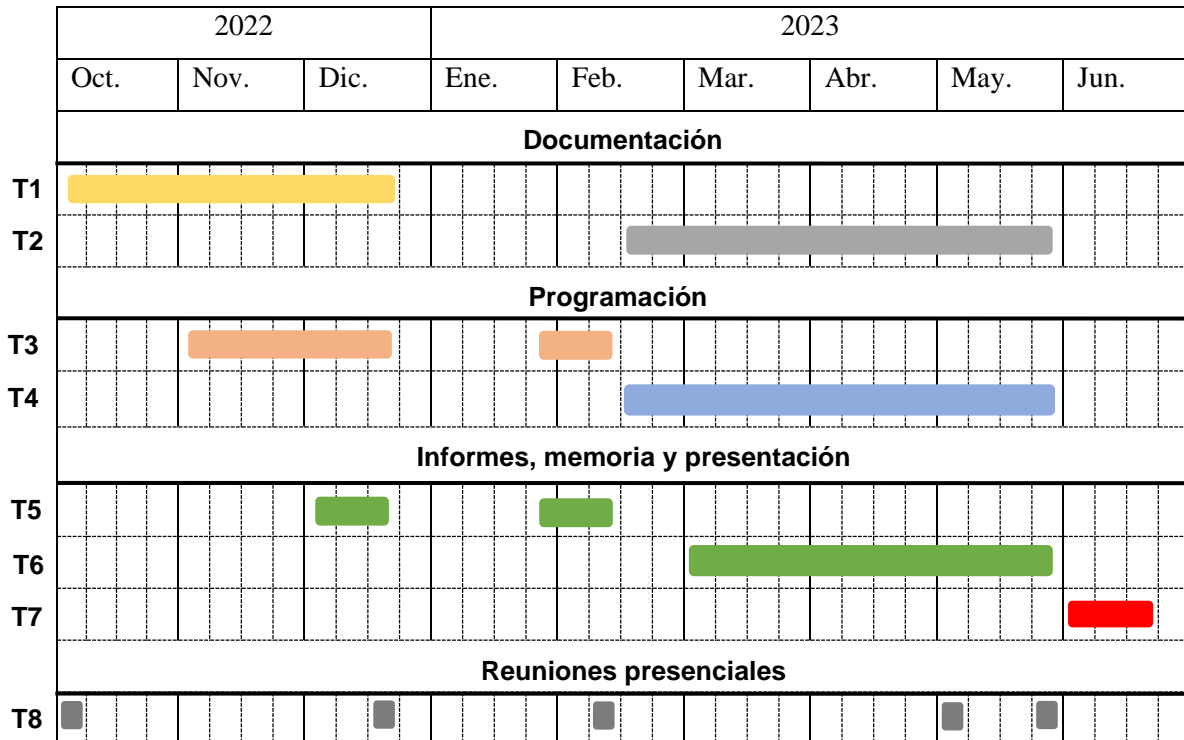


Tabla 1. Diagrama Gantt planificación de tareas.

2. METODOLOGÍA

Este apartado contiene los principios teóricos de los modelos predictivos empleados en el análisis. En concreto, dos modelos clásicos como son el suavizado exponencial y el modelo autorregresivo y de medias móviles ARIMA. Dichos modelos se asocian con la teoría econométrica clásica y tienen su origen en el análisis de series temporales económicas, teniendo como principal referencia el modelo de Holt & Winters y la metodología de Box & Jenkins. En contraposición, se presenta la teoría de un modelo moderno basado en redes neuronales recurrentes Long-Short Term Memory (LSTM) cuyo origen se remonta a 1997 con la publicación de Hochreiter & Schmidhuber.

También se recoge la formulación matemática detrás de las herramientas de medida de error RMSE y MAE empleadas para comparar los resultados de las predicciones con las observaciones reales de las series temporales eurodólar, euro-libra y euro-dólar neozelandés.

2.1 Modelo clásico: suavizado exponencial

El método de suavizado exponencial (Brown & Meyer, 1961) es uno de los modelos de predicción de series temporales más simples utilizados en la práctica (Ostertagová *et al*, 2011). Este método genera la predicción a partir de la suma lineal de la ponderación de la muestra real actual y la predicción anterior por un parámetro alfa. El parámetro alfa o constante de suavización es un valor que podría ser arbitrario, pero de forma más compleja se puede optimizar mediante la minimización del error cuadrático medio de forma que se encuentre un valor óptimo tal que ofrezca los mejores resultados. Conforme las observaciones están más alejadas en el tiempo, la influencia del parámetro alfa en el proceso de predicción es menor (Cipra & Hanzák, 2008). Por otro lado, las muestras más recientes serán las que tengan un mayor peso y relevancia a la hora de generar la predicción.

Este método arroja buenos resultados cuando la predicción es el futuro inmediato y la serie temporal analizada tiene tendencia y estacionalidad nula (Gardner Jr, 1985), lo que es lo mismo, que la serie temporal oscile entorno a una media estable.

La denominación de exponencial viene del desarrollo de la ecuación principal (1):

$$\hat{y}_t = \alpha \cdot y_t + (1 - \alpha) \cdot \hat{y}_{t-1} \quad (1)$$

A continuación, se explica la referencia de cada término:

- y_t corresponde con el valor real actual de la serie temporal.
- \hat{y}_t , representa la predicción de la muestra actual.
- \hat{y}_{t-1} , representa la predicción de la muestra del periodo anterior al actual.
- α , la constante de suavización.

Si sustituimos el término \hat{y}_{t-1} que correspondiente con la predicción realizada en el instante anterior, obtenemos la ecuación (1.2)

$$\hat{y}_t = \alpha \cdot y_{t-1} + \alpha \cdot (1 - \alpha) \cdot y_{t-1} + (1 - \alpha)^2 \hat{y}_{t-2} \quad (1.2)$$

Ecuación 1. y 1.2 Formulación matemática del suavizado exponencial simple.

Entonces es posible observar que la predicción del instante actual se compone de un número cada vez mayor de observaciones reales pasadas y sus componentes de suavización. De forma que los exponentes, que acompañan suavizando las observaciones reales pasadas, están ponderadas por un término que aumenta exponencialmente.

Normalmente las series temporales no suelen oscilar entorno a una media estable, lo cual es una desventaja a la hora de emplear este modelo. No obstante, según la metodología de Holt-Winters se hizo necesario ampliar el modelo anterior introduciendo nuevos elementos característicos de las series temporales como son la tendencia y la estacionalidad (Gardner Jr, 1985).

La tendencia hace referencia a la dirección que toman las observaciones de las series temporales cuando se observan varios periodos de tiempo. Existen diversos tipos de tendencia (Makridakis & Hibon, 1991), aunque hablar de tendencia en un modelo de Holt implica generalmente una tendencia lineal.

La estacionalidad en un periodo de tiempo, por ejemplo, un año, representa una desviación repetitiva con respecto a la media anual que sucede año tras año (Kalekar, 2004). Está estrechamente relacionada con estaciones concretas del año, ya que las series temporales pueden ser vulnerables a cambios de estaciones, efectos económicos provocados por los periodos festivos...etc.

Holt introduce al modelo matemático del suavizado exponencial simple un elemento que permite identificar tendencia en la serie temporal (Gardner Jr, 1985). Este avance hizo denominar al nuevo modelo como 'suavizado exponencial doble' o 'método de Holt'.

El cambio consistió simplemente en añadir un término b dentro de la componente suavizada de la predicción anterior:

$$\hat{y}_t = a \cdot y_{t-1} + (1 - a) \cdot (\hat{y}_{t-1} + b_{t-1}) \quad (2)$$

Ecuación 2. Formulación matemática suavizado exponencial doble.

El modelo de suavizado exponencial triple también denominado modelo de Holt-Winters, introduce la posibilidad de identificar tendencia, igual que con el suavizado exponencial doble, y también la capacidad de identificar estacionalidad en la serie temporal (Winters, 1960; Brown, 1986).

2.2 Modelo clásico: ARIMA

El modelo ARIMA hace referencia a las siglas *Auto-Regressive Integrated Moving Average*. Se trata de un modelo estocástico univariante indexado a la variable tiempo, cuya implementación es más sofisticada que el suavizado exponencial y se basa en la metodología de Box-Jenkins (Box & Jenkins, 1973) Esta metodología sugiere la aplicación de tres pasos: identificación del modelo que mejor se ajusta a los datos, estimación de los parámetros de dicho modelo y un diagnóstico de los residuos que asegure que no hay patrones.

Una de las características necesarias para la ejecución del modelo ARIMA es la condición de estacionariedad que deben seguir los datos de la serie temporal (Hillmer & Tiao, 1982). Esto quiere decir que las observaciones no deben tener tendencia, es decir, su media debe ser constante, al igual que su varianza. En resumen, la variación de las observaciones de la serie temporal no cambia en función del tiempo. Como veremos más adelante, la representación gráfica de las rentabilidades asociadas a los pares de divisas nos indicará que tienen naturaleza estacionaria, también puede emplearse la prueba de Dickey-Fuller.

Por otro lado, la predicción a través de un modelo ARIMA resulta de la construcción de una función lineal entre las observaciones anteriores y los errores cometidos en las predicciones (Zhang, 2003).

El modelo ARIMA (como indica el nombre) está compuesto por la combinación de modelos Auto-Regresivos y de media móvil (Siami-Namini *et al.*, 2018). Los elementos principales son:

- Auto-Regresión (AR): se basa en la dependencia de las observaciones con un número determinado de muestras pasadas (p muestras).
- Media móvil (MA): relaciona la dependencia de las observaciones con el error residual cometido al utilizar el modelo MA con un número (q) de errores cometidos anteriores.
- Integrado: se toman diferencias (d) entre las observaciones para lograr estacionariedad.

La representación del modelo se obtiene de la combinación de los parámetros óptimos de la parte de autoregresión y de medias móviles identificándose de la forma: ARIMA (p,d,q). Los valores p , d y q que se observan en el modelo son números enteros no negativos. Los parámetros p y q indican el orden de la parte AR y MA. En caso ser ' q ' igual a 0, el modelo es autorregresivo puro y si por el contrario ' p ' es igual a 0, el modelo es de media móvil puro (Khan & Alghulaiakh, 2020). Por tanto, resulta fundamental identificar la combinación de parámetros p y q apropiados.

La formulación matemática del modelo ARIMA se representa a continuación en la ecuación (3) (Makridakis, S., & Hibon, M., 1979)

$$y_t = \theta_0 + \phi_1 \cdot y_{t-1} + \phi_2 \cdot y_{t-2} + \dots + \phi_p \cdot y_{t-p} + \varepsilon_t - \theta_1 \cdot \varepsilon_{t-1} - \theta_2 \cdot \varepsilon_{t-2} - \dots - \theta_q \cdot \varepsilon_{t-q}$$

Ecuación 3. Formulación matemática modelo ARIMA.

El modelo matemático se descompone en:

- y_t , predicción actual.
- y_{t-1} , observación real de la serie temporal retrasado un período con respecto a la observación actual.
- y_{t-p} , observación real de la serie temporal retrasada el número óptimo de muestras p según el modelo de medias móviles.
- ε_t , error cometido en la predicción de la observación.
- ε_{t-q} , error cometido en la predicción de la observación correspondiente a un número óptimo q de muestras anteriores a la actual.

Una herramienta que nos permite identificar los parámetros p y q óptimos manualmente es la función de autocorrelación (ACF) y la función de autocorrelación parcial (PACF) (Liu, 1989). La gráfica ACF representa el coeficiente de autocorrelación, calculado a partir de la serie real y la serie real retardada un número entero de periodos. Si la correlación está próxima a 1 indica una correlación directa significativa, en cambio si está próximo a -1, la correlación es significativa pero inversa. Por último, si el coeficiente de correlación es 0 implica una relación de independencia. De la función de autocorrelación ACF se obtiene el orden q correspondiente al modelo de medias móviles MA. En la figura 1 se representa la función ACF aplicada a la serie temporal EUR/USD diario. En la cual se puede identificar el orden de la componente de medias móviles es q=1.

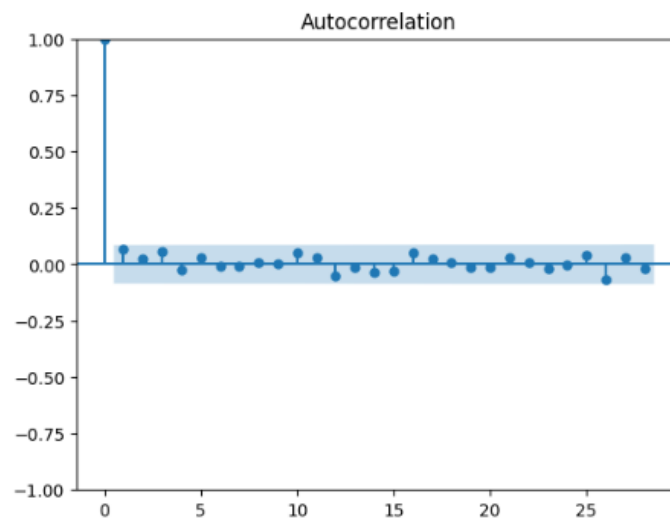


Figura 1. Captura función Autocorrelación muestra EUR/USD diario.

Por otro lado, la función de autocorrelación parcial PACF está formada por la derivada de los intervalos que hay en la función de autocorrelación ACF. La función PACF indica el número de periodos anteriores significativos de la parte de autocorrelación (AR). En la figura 2 se representa la función PACF para la muestra EUR/USD con datos diarios. Puede observarse que el número de periodos significativos es 2 y por tanto el valor p tendrá dicho valor.

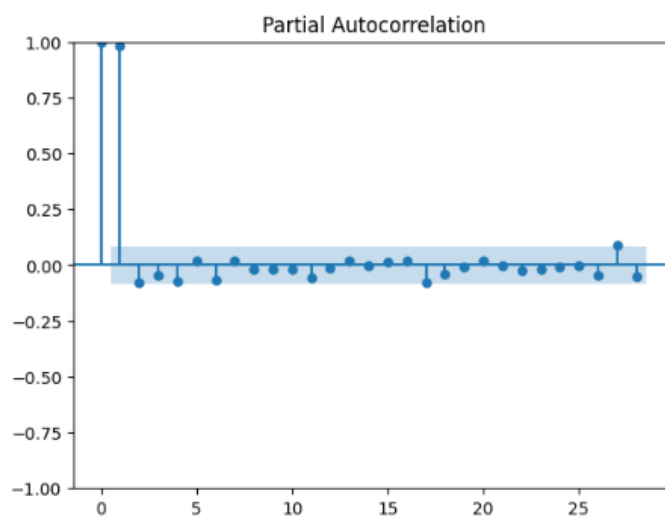


Figura 2. Captura función Autocorrelación Parcial muestra EUR/USD diario.

Otra forma de hallar la combinación adecuada es a partir del cálculo del Akaike Information Criteria (AIC) (Sakamoto *et al.*, 1986). Este criterio es útil para cuantificar la bondad del ajuste del modelo. Según el método de Box-Jenkins (Mondal *et al.*, 2014) tanto el valor de p como el valor de q no deberían tener valores mayores a 2. Así como sumar menos de un valor de 3 entre los dos. De esta forma se puede hallar el modelo óptimo si se tienen en cuenta todas las combinaciones posibles y se selecciona el modelo cuyos parámetros ofrezcan un valor AIC menor.

Cabe destacar que un estudio (Makridakis & Hibon, 1997) puso en duda la practicidad de la metodología empleada, comparando distintos modelos ARIMA. La conclusión fue que la metodología de Box-Jenkins, aplicada en el dominio económico, justificado en gran medida por la aleatoriedad de los datos, puede acarrear que el modelo obtenido de los valores p,d,q extraídos de las funciones ACF y PACF den resultados menos acertados que un modelo ARMA más simple como el AR(1), AR(2) ó ARMA(1,1). En definitiva, puede que un modelo ARMA muy complejo y sofisticado no sea tan preciso como un modelo sencillo.

2.3 Modelo moderno: redes neuronales recurrentes LSTM

LSTM (Long Short-Term Memory) corresponde con una variante de las Redes Neuronales Recurrentes (RNN). Fue propuesta por primera vez en 1997 (Hochreiter & Schmidhuber, 1997) y es una herramienta basada en Machine Learning supervisado especializada en el análisis de series temporales.

Se conoce al aprendizaje automático (Machine Learning) como un conjunto de procesos desarrollados a partir de algoritmos cuya finalidad es alcanzar un aprendizaje efectivo gracias al empleo de gran cantidad de datos. Desde el punto de vista práctico, el aprendizaje supervisado consiste en proporcionar al algoritmo información real cuya respuesta es conocida para generar una respuesta correcta en caso de introducir datos no conocidos o nuevos. Por otro lado, el aprendizaje automático no supervisado se identifica con un algoritmo cuyos procesos internos deben encontrar patrones y ser capaces de generar respuestas correctas sin disponer de información real previa.

Las redes neuronales recurrentes (RNN) tienen la ventaja de permitir que la información obtenida como salida de una capa interna se convierta en entrada de la siguiente capa. A este proceso iterativo se le llama recurrencia, en cierto modo la información dentro de una red neuronal sigue un bucle y en sus capas internas se almacena la información que considera relevante. La capacidad de las redes neuronales para contener información se denomina memoria y permite almacenarla durante largos periodos de tiempo mientras sea considerada relevante (Wenjie Lu *et al*, 2020). Por otro lado, la red neuronal también es capaz de detectar patrones a corto plazo gracias a las funciones de activación.

El funcionamiento de una neurona o celda es bastante intuitivo. Ésta se compone de una puerta de entrada (input gate), otra de olvido (forget gate) y una última puerta de salida (output gate). A su vez las neuronas precisan de tres tipos de datos: la entrada actual (x_t), la salida de la neurona en el paso de tiempo anterior (h_{t-1}) y la celda de memoria anterior (c_{t-1}). En la figura 3 se representa el esquema de una neurona, así como los flujos de entrada y salida de datos:

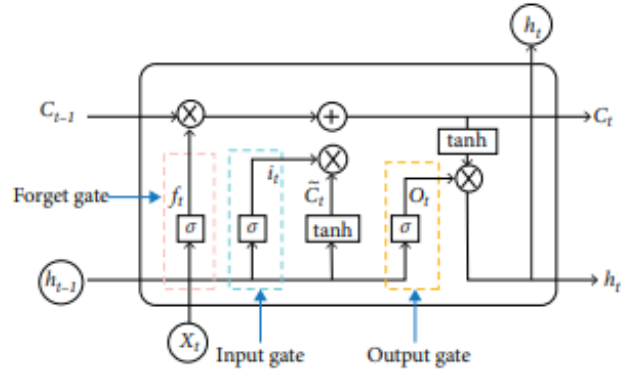


Figura 3. Diagrama estructural LSTM (Wenjie Lu et al, 2020)

Input gate: La puerta de entrada se encarga de decidir qué nueva información se va a agregar a la neurona. Esta información, dada en forma de secuencia de datos como vector de palabras o de números, emplea la función de activación sigmoide para obtener un vector de ponderaciones el cual indicará el peso que tienen los valores a la hora de ser agregados. Los datos de entrada se componen de una combinación de la entrada actual (x_t) con la salida de la neurona anterior (h_{t-1}) normalizados por la función tangente hiperbólica.

Forget gate: La puerta de olvido indica la información antigua ubicada en la memoria que se deberá conservar (c_{t-1}) y la que se deberá descartar. Ésta calculará un vector de ponderaciones de olvido a partir de aplicar la función sigmoide a la concatenación de los datos de la entrada actual (x_t) y la salida de la neurona en el paso de tiempo anterior (h_{t-1}). Dicho vector de ponderaciones tomará valores entre cero y uno. Un valor cercano a cero indicará que se debe olvidar y si es cercano a uno, el valor deberá conservarse. Cuando se multiplica dicho vector de ponderaciones de olvido con la información antigua ubicada en la memoria (h_{t-1}), se obtiene la información que debe restarse de la celda de memoria actual.

Output gate: La puerta de salida indica la información de la memoria actual que se enviará como salida en el siguiente paso de tiempo. De forma análoga a la puerta de entrada, se calculará un vector de ponderaciones de salidas aplicando la función sigmoide a la concatenación de la entrada actual y la salida de la neurona anterior. Por último, se

desnormalizan los datos que se habían normalizado en la puerta de entrada aplicando de nuevo la función de tangente hiperbólica.

La función sigmoide sirve como función de activación para el control del flujo de información en las puertas de entrada y de olvido (Nelson *et al.*, 2017). Toma valores entre cero y uno, lo cual es útil en la toma de decisiones binarias, como por ejemplo decidir las muestras que entran y las que salen. La función sigmoide tiene la siguiente expresión matemática.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Ecuación 4. Función sigmoide.

La figura 4 representa la función sigmoide. Se puede observar que sigue una curva en forma de ‘S’.

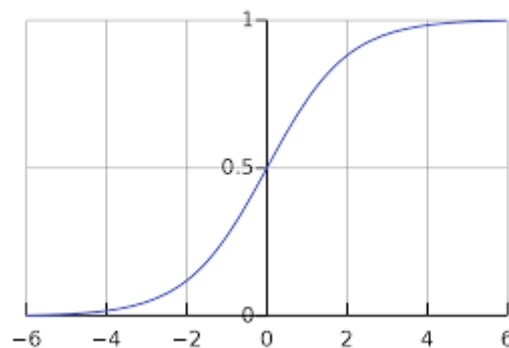


Figura 4. Función Sigmoide.

Por otro lado, la función tangente hiperbólica se utiliza para normalizar los valores de entrada y salida en las puertas de entrada y de salida. Normalizar los datos permite controlar la entrada de datos que sean muy grandes o muy pequeños y por tanto facilite el manejo de la información en las neuronas LSTM (Siami-Namini *et al.*, 2018). La función de activación tangente hiperbólica se define como:

$$\tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$$

Ecuación 5. Función tangente hiperbólica.

A continuación, se grafica la función de activación. Se observa simetría entorno al cero, ya que la salida se encuentra entre los valores -1 y 1.

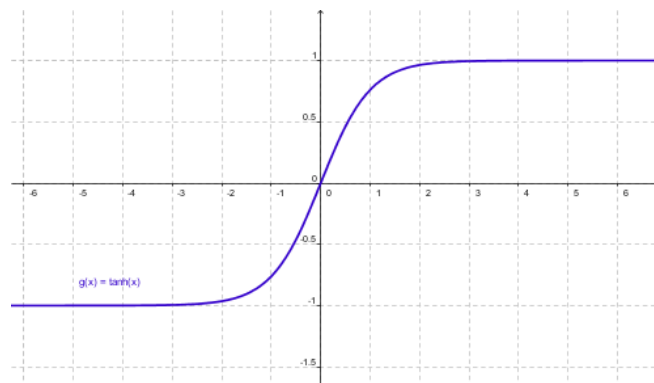


Figura 5. Función tangente hiperbólica.

Otra función de activación muy común es la función ReLU (Rectified Linear Unit). Ésta puede utilizarse como función de activación en la puerta de olvido. Es una función no lineal y especialmente simple que ofrece eficiencia computacional frente a otras funciones de activación como la función Sigmoide y tangente hiperbólica. El funcionamiento básico es que los valores negativos se aproximan a cero y los positivos mantienen su valor normalizado.

$$ReLU(x) = \max(0, x)$$

Ecuación 6. Formulación matemática función ReLU.

El control del flujo de la información que se introduce en la red neuronal juega un papel clave en los resultados de las predicciones. El diseño de la arquitectura de la red neuronal pasa por la elección del número de capas internas que tendrá la red, definiendo así el número de unidades neuronales que almacenarán información. Un número de capas elevado está relacionado con una capacidad de almacenaje de información mayor, aunque si se excede puede resultar en problemas de sobreajuste ocasionando dificultades durante el entrenamiento. Una práctica común es emplear un número de capas menor e ir incrementándolas en vistas a conseguir un rendimiento mejor.

Asimismo, es importante tener en cuenta la limitación de los recursos computacionales, ya que afecta directamente a los tiempos de ejecución y al uso de memoria RAM de las computadoras, haciendo clave la optimización de la configuración y parametrización de la red neuronal.

Las redes neuronales ofrecen buenos resultados hasta en problemas cuya solución no es lineal, pero introduce dos principales inconvenientes. En primer lugar, los tiempos de entrenamiento y de test pueden ser bastante largos y, por otro lado, se puede llegar a sobreajustar (overfitting) la propia red neuronal y terminar obteniendo una solución local, en vez de una solución global (Haiyao Wang *et al*, 2021).

Para evitar obtener una solución local durante el training se utilizan algoritmos cuyo objetivo es minimizar la función de pérdidas de la red neuronal. Un ejemplo de dicho algoritmo es el descenso de gradiente o la función de optimización Adam.

El problema del gradiente proviene del cálculo de la función de pérdidas. Al calcular las derivadas parciales de la función de pérdidas, se comprueba qué tan bien funciona el modelo LSTM validando así los pesos y sesgos asociados a cada neurona de la red. Puede ocurrir que el gradiente explote o se desvanezca. Una explosión implica un crecimiento drástico de los pesos en cada iteración, empujando el proceso de entrenamiento hacia un modelo inestable y de difícil convergencia. Por otro lado, el desvanecimiento implica que los pesos disminuyan, haciendo que la información pasada que contenga cada neurona sea poco relevante en la predicción. Favoreciendo el uso de información más reciente en el eje temporal que información más antigua, complicando la retroalimentación, y la creación de tendencias a largo plazo.

El optimizador Adam trata de optimizar la tasa de aprendizaje (learning rate) con la que se ajustan los pesos de la red neuronal en cada iteración. Si la tasa de aprendizaje es muy alta, el entrenamiento puede volverse inestable, no llegando a alcanzar la convergencia. Mientras que, si la tasa es pequeña, el training puede ser bastante lento traduciéndose en el estancamiento de la red neuronal en un mínimo local.

Para que la red neuronal funcione correctamente, se debe reservar una cantidad de información real al margen del entrenamiento. Esta información nos permitirá obtener un resultado aproximado sobre el funcionamiento de la red neuronal cuando se emplea información que no ha sido utilizada en el proceso de entrenamiento. La validación es el paso intermedio que permite al diseñador ajustar parámetros de la red neuronal antes de generar las predicciones.

Para evaluar las pérdidas durante el entrenamiento de la red neuronal, se calcula el MSE entre las predicciones y los datos reales. Recordemos que las redes neuronales LSTM siguen un proceso de aprendizaje supervisado, ya que cada predicción del rendimiento tiene un valor real asociado. Si el error cometido es pequeño, la red neuronal estará bien ajustada, de no ser así, habrá que

indagar los motivos, tanto de ajuste de hiperparámetros como de optimización, que causan un mal aprendizaje debido a su tasa de error elevada.

De forma adicional al número de capas de la red neuronal, el hiperparámetro más importante es el número de 'epochs'. Este hiperparámetro se asocia con el número de pasadas que realiza el algoritmo con los datos de entrenamiento que contiene el vector de observaciones reales con información de un año de cotización del mercado. Estas pasadas permiten a la red neuronal ajustar el peso de las neuronas, clave en el flujo de la información dentro de los procesos internos de las predicciones. Existe la posibilidad de que un mayor número de pasadas perjudique la optimización, dejando la red neuronal sobreentrenada.

2.4 Métricas de error: RMSE y MAE

Para comparar las predicciones obtenidas de los modelos con las muestras reales, se emplearán medidas de error como el RMSE y el MAE (W. Wang *et al.*, 2018). Estas herramientas evalúan la bondad de los modelos de predicción. El valor desprendido del cálculo del RMSE permitirá obtener conclusiones sobre los modelos, los pares de divisas y los time-frame.

El error cuadrático medio (Root Mean Square Error, por sus siglas en inglés) es una de las herramientas que más aplicabilidad práctica tiene a la hora de analizar el funcionamiento de un modelo de predicción (Roondiwala *et al.* 2017). Éste mide la desviación entre los valores reales y las predicciones. El efecto de aplicar la raíz a las diferencias hace que errores mayores tengan un reconocimiento mayor, frente a errores pequeños, penalizando el valor del RMSE. En la ecuación 7 se representa la formulación matemática del error RMSE, donde \hat{y}_n es la predicción e y_n el dato real. Además, en la figura 6 (Roondiwala *et al.* 2017) se muestra de forma visual el cálculo de los residuos o desviaciones entre las predicciones y las observaciones reales.

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^N (\hat{y}_n - y_n)^2}$$

Ecuación 7. Formulación Root Mean Square Error

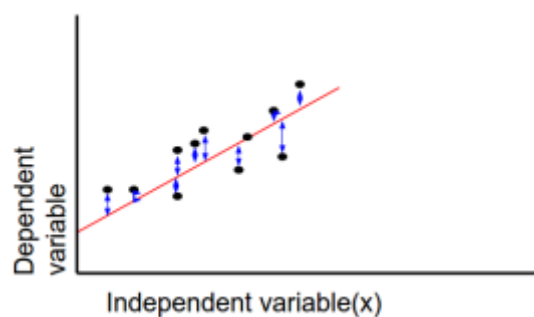


Figura 6. Cálculo del RMSE, figura tomada de Roondiwala *et al.* 2017

Cuanto más pequeño sea el RMSE mejor ajustado estará el modelo. Por el contrario, si el RMSE es grande, puede indicar que el modelo no está lo suficientemente bien implementado y es preciso revisarlo.

Por su parte el MAE (error absoluto medio) también es una herramienta empleada para medir el buen ajuste de los modelos de predicción, pero con la peculiaridad de que al tomar el valor absoluto de la diferencia entre la muestra real con la predicción se está discriminando el signo del error cometido (Chai & Draxler, 2014). El cálculo del error es la media aritmética de aplicar el valor absoluto a todas las diferencias de las predicciones con sus muestras reales.

$$MAE = \frac{1}{N} \sum_{n=1}^N |\hat{y}_n - y_n|$$

Ecuación 8. Formulación matemática Mean Absolute Error.

Una vez obtenidos los valores de RMSE y MAE para cada modelo, divisa y time-frame, se recogerán en tablas resumen que nos permitirán extraer las conclusiones del estudio.

3. DISEÑO DE MODELOS DE PREDICCIÓN EN EL MERCADO FOREX

Como se ha indicado en apartados anteriores y a modo de resumen, se emplearán tres modelos predictivos (suavizado exponencial, ARIMA y redes neuronales LSTM) para obtener la predicción de tres pares de monedas que cotizan en el mercado Forex (Euro-Dólar, Euro -Libra y Euro-Dólar neozelandés). Todo ello desde el punto de vista de distintos time-frames: diarios, semanales y mensuales. Para cuantificar la relación entre las observaciones reales y las predicciones, es decir, la bondad de la capacidad predictiva, se realizará el cálculo del RMSE y el MAE. Dichas tasas de error se plasmarán en tablas resumen que nos permitirán hallar tendencias o anomalías entre los resultados obtenidos.

El histórico de cotización de los pares de divisas se obtendrá a partir de los datos ofrecidos por un software especializado en mercados financieros, VisualChart. Los modelos predictivos, así como la Red Neuronal LSTM se programarán en Python.

3.1 Origen de los datos

Los datos brutos conseguidos para realizar el análisis de los tipos de cambios de las monedas seleccionadas se han obtenido a través de la aplicación informática Visual Chart. Esta herramienta permite analizar gratuitamente las series temporales de los principales mercados bursátiles en tiempo real, así como disponer de los históricos de cotización de todos los valores de los mercados financieros mundiales. En su versión de pago, permite trabajar realizando órdenes y operaciones de trading en tiempo real y de forma fiable. Además, dispone de un simulador para realizar prácticas en mercados sin riesgo real.

En esta ocasión, se ha utilizado la aplicación con la intención de descargar los históricos de los precios de cierre diarios, semanales y mensuales de las series temporales de los pares de divisas EUR/USD, EUR/GBP y EUR/NZD.

Los ficheros generados en la descarga de datos son ficheros de texto con tamaños que varían entre los 31 KB de las series mensuales (con menor número de muestras) y los 670 KB de las series temporales diarias. Dichos ficheros contienen información muy completa de las series temporales. A continuación, se especifica la información que incluyen:

TICKER: Indica el nombre de la serie temporal, en nuestro caso puede ser EURUS, GBPVS ó NZDVS.

PER: Especifica el tipo de time-frame utilizado, diario (D), semanal (W) y mensual (M).

DTYYYYMMDD: fecha de la muestra seleccionada en formato año/mes/día.

OPEN: Precio de apertura en la fecha del time-frame seleccionado.

HIGH: Precio máximo de la cotización durante el time-frame seleccionado.

LOW: Precio mínimo de la cotización durante el time-frame seleccionado.

CLOSE: Precio de cierre de la cotización en la fecha del time-frame seleccionado.

En la tabla 2 se muestra el número total de observaciones en las series temporales de los pares de monedas estudiadas. Los datos se tomaron el 14 de febrero de 2023, por lo que el número de observaciones dependerá de la fecha origen del histórico de datos. Por ejemplo, la serie par eurodólar tiene registros desde el 6 de marzo de 1970. La serie euro-libra desde el 9 de marzo de 1979 y la serie del par euro-dólar neozelandés desde el 28 de julio de 2003. Teniendo en cuenta los periodos de tiempo en los que el mercado ha estado activo se tienen los siguientes números de observaciones en función del time-frame:

	EUR/USD	EUR/GBP	EUR/NZD
Time-frame diario	11 467	11 453	5 206
Time-frame semanal	2293	2298	1026
Time-frame mensual	528	528	236

Tabla 2. Resumen número de muestras de cada serie temporal en función del time-frame.

Además del número de muestras también será imprescindible recoger en una tabla los estadísticos más representativos de las series temporales, como por ejemplo: media, desviación típica, mínimo y máximo de cada par de monedas. Debido a que se utilizan monedas Major y exóticas, estudiar dichos parámetros puede complementar los resultados obtenidos en cada modelo. En la tabla 3 se recoge la información de las series temporales ordenadas en función del time-frame y del tipo de cambio.

Time-frame	EUR/USD		EUR/GBP		EUR/NZD	
	media	Desv. típica	media	Desv. típica	media	Desv. típica
Diario	1.1660	0.1698	0.7326	0.1039	1.7621	0.1849

Semanal	1.1649	0.1706	0.7322	0.1043	1.7596	0.1851
mensual	1.1651	0.1711	0.7319	0.1445	1.7609	0.1855

Tabla 3. Resumen estadísticos media y desviación típica de las series temporales.

Se observa que la desviación típica del par EUR/NZD es mayor que las desviaciones típicas de las series EUR/USD y EUR/GBP. A partir de los datos de las desviaciones típicas se puede afirmar que la moneda exótica tiene más variabilidad que las monedas Mayor.

Continuando el análisis de las series temporales es imprescindible realizar un filtrado de su contenido, además de homogeneizar el número de observaciones. Para ello se asegurará que las series temporales están compuestas por números reales y no negativos evitando también tener en cuenta observaciones que sean NaN.

Por otro lado, para homogeneizar el número de muestras en las series temporales es necesario que para cada tipo de time-frame contengan el mismo número de observaciones.

Para ello, se tomará como referencia la serie temporal EUR/NZD ya que es la serie que menor número de observaciones contiene tanto en el time-frame diario como en el semanal y mensual.

A continuación, en las figuras 7, 8 y 9 se representan las series históricas diarias de los pares de divisas EUR/USD, EUR/GBP y EUR/NZD. Las abreviaturas corresponden con las monedas eurodólar, euro-libra y euro-dólar neozelandés. Del análisis visual de las gráficas se puede extraer que no hay estacionariedad en las series temporales. Esto se debe a que la variación de las observaciones dentro de la serie temporal es independiente del tiempo. Lo cual dificulta enormemente anticiparse al comportamiento futuro de la serie temporal ya que sus propiedades estadísticas son desconocidas.

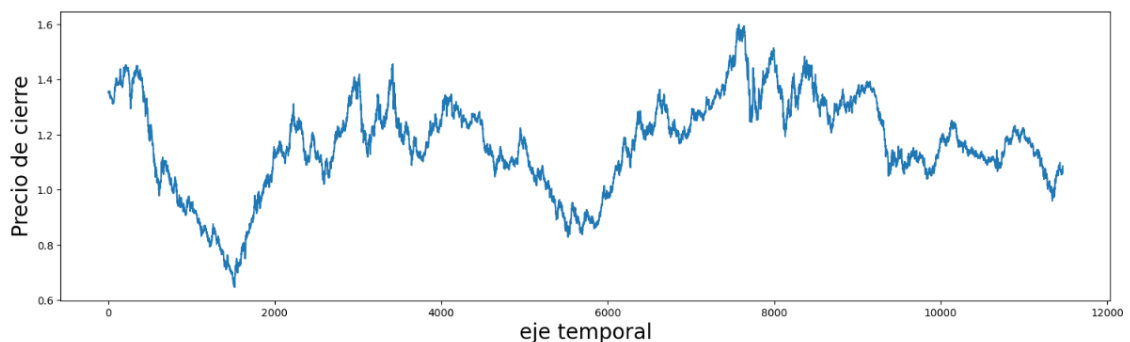


Figura 7. Precio de cierre de la cotización de la serie EUR/USD diaria.

Analizando visualmente las series temporales, se puede distinguir que la dominancia eurodólar ha ido alternándose a lo largo de la serie histórica. En comparación con la serie euro-libra, que indica una dominancia absoluta de la libra frente al euro desde el inicio de los registros.

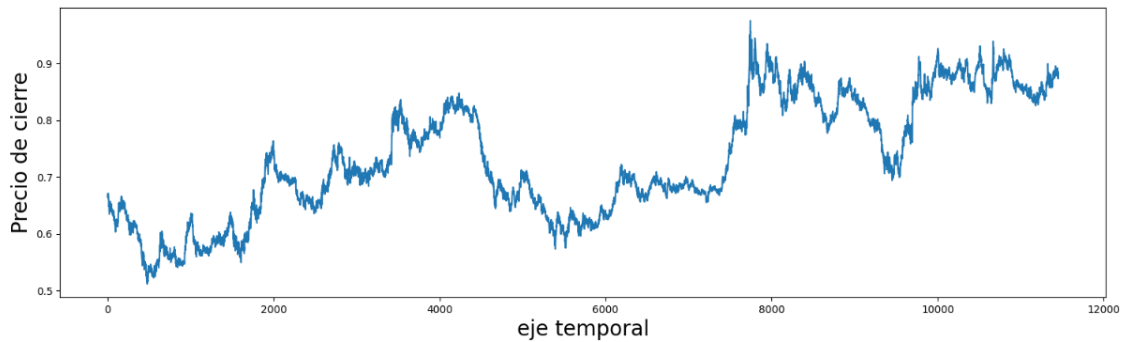


Figura 8. Precio de cierre de la cotización de la serie EUR/GBP diaria.

Con respecto a la serie eurodólar neozelandés cabe decir que el euro ha dominado a lo largo del tiempo. A diferencia de las series temporales anteriores, la serie EUR/NZD presenta cierta estabilidad durante los últimos años, lo cual contrasta con las grandes variaciones de los primeros años de su cotización.

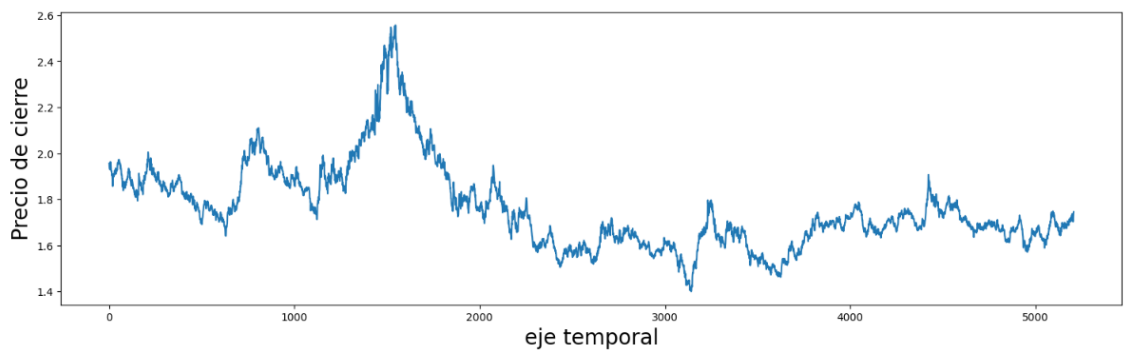


Figura 9. Precio de cierre de cotización de la serie EUR/NZD diaria.

3.2 Modelización en Python

3.2.1 Importación de datos y cálculo de la rentabilidad logarítmica

El entorno de programación en Python empleado para el desarrollo del código de los algoritmos de los modelos de predicción es Google Colab. Se descubrió durante el periodo de recogida de información para la implementación de los modelos. La gran mayoría de fuentes consultadas utilizaban Jupyter, Visual Studio o Google Colab. Existen dos principales motivos por los que se tomó la decisión de emplear el entorno de Google Colab. En primer lugar, por el autoguardado en la nube de Google, ya que te permite acceder desde distintos equipos con tu misma cuenta sin preocuparte de guardar los cambios o los resultados de la ejecución. El segundo motivo es la capacidad de ejecutar código en un servidor de Google, de forma que si la capacidad computacional del ordenador no es buena puedes realizar la ejecución en un servidor externo.

Una vez importados los datos de las series temporales en Google Colab, la información es filtrada y homogeneizada con el fin de asegurar la calidad de las observaciones y un número de muestras común para todas. A continuación, se calcula la rentabilidad compuesta periodo a periodo con la intención de transformar las series temporales originales no estacionarias en series temporales estacionarias. La fórmula matemática de la rentabilidad se muestra a continuación. Las observaciones empleadas en el cálculo de la rentabilidad son los precios de cierre ('CLOSE') obtenidos a partir de los ficheros .txt de VisualChart.

$$r = \ln\left(\frac{P_t}{P_{t-1}}\right) = \ln(P_t) - \ln(P_{t-1})$$
$$r\% = \ln\left(\frac{P_t}{P_{t-1}}\right) \cdot 100 = (\ln(P_t) - \ln(P_{t-1})) \cdot 100$$

Ecuación 9. Formulación rentabilidad compuesta

A continuación, se muestran las gráficas donde se representan las rentabilidades compuestas calculadas de todo el histórico de datos de las series temporales seleccionadas. La figura 10 corresponde con la rentabilidad de la serie eurodólar diaria. La figura 11 muestra las rentabilidades semanales de la serie euro-libra semanal y por último se representa en la figura 12 la rentabilidad mensual de la serie temporal Euro-Dólar Neozelandés con time-frame mensual.

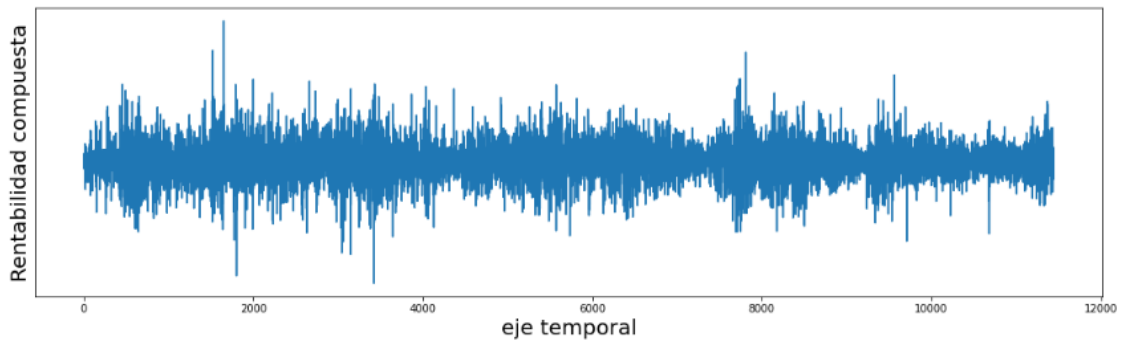


Figura 10. Rentabilidad compuesta de la serie EUR/USD diaria.

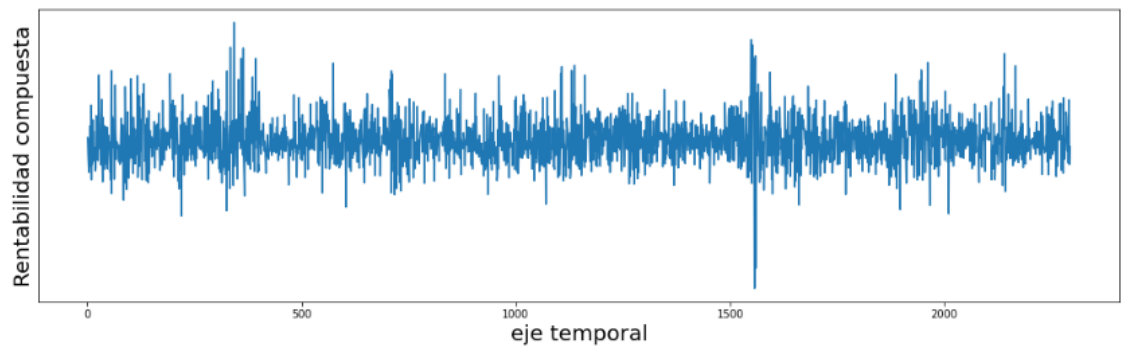


Figura 11. Rentabilidad compuesta de la serie EUR/GBP semanal.

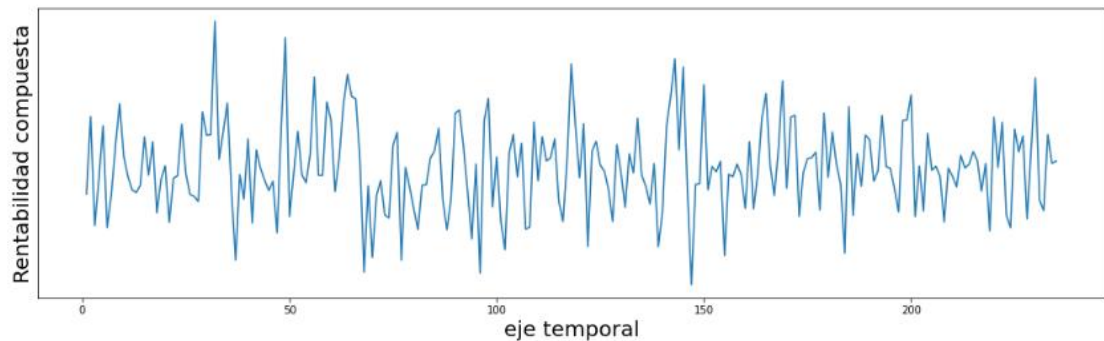


Figura 12. Rentabilidad compuesta de la serie EUR/NZD mensual.

El siguiente paso es desarrollar el algoritmo de ventana deslizante que irá almacenando las observaciones correspondientes a un año en un vector. Habrá que tener en cuenta que el mercado FOREX no está operativo todos los días del año, en un marco temporal diario, el vector de observaciones equivalentes a un año estará compuesto por 250 datos de rentabilidad compuesta. A modo aclarativo se ilustra en la figura 13 un ejemplo de ventana deslizante que se deberá implementar cuando el time-frame de la serie temporal sea diario, y por tanto el vector de observaciones que predecirá una muestra deberá contener los 250 datos de rentabilidades pasadas. En caso de un time-frame semanal el vector de observaciones deberá contener 52 datos de

rentabilidad y serán 12 si el time-frame es mensual. Conforme pasen las muestras el vector se irá actualizando con nueva información.

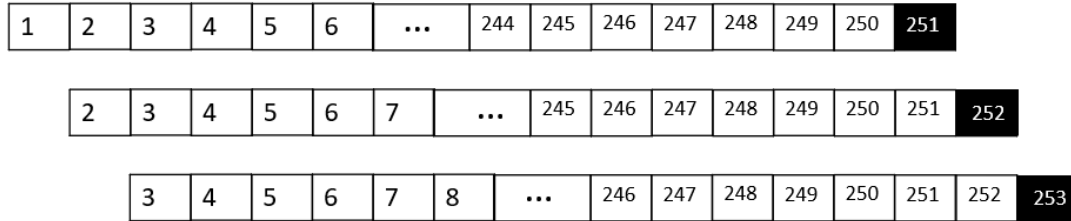


Figura 13. Bucle diseñado para obtener el vector de observaciones con time-frame diario.

La implementación del mecanismo de ventana deslizante se realiza a través de un bucle ‘for’ donde la última observación corresponde con la información más reciente de rentabilidad y las primeras con información más antigua.

Una vez el bucle haya recorrido todas las observaciones hasta llegar a la última, se habrá obtenido una matriz que almacene los vectores con las observaciones equivalentes a un año de cotización. En la figura 14, se toman muestras reales y se ejemplifica el mecanismo de ventana deslizante, suponiendo que el vector de datos sólo precise las 6 muestras anteriores para generar una predicción. Se puede observar el movimiento que siguen las muestras, ya que aquéllas ubicadas más a la derecha se van desplazando posicionalmente hacia la izquierda tras cada iteración, adoptando un carácter más ‘antiguo’ a la hora de generar la predicción futura.

[-0.01520715	-0.00882725	0.0334594	0.01259894	0.00216873	0.03050903
[-0.00882725	0.0334594	0.01259894	0.00216873	0.03050903	-0.03775649
[0.0334594	0.01259894	0.00216873	0.03050903	-0.03775649	0.03803662
[0.01259894	0.00216873	0.03050903	-0.03775649	0.03803662	0.01121065
[0.00216873	0.03050903	-0.03775649	0.03803662	0.01121065	-0.00729702
[0.03050903	-0.03775649	0.03803662	0.01121065	-0.00729702	-0.0153222

Figura 14. Mecanismo de desplazamiento de muestras de una ventana móvil.

3.2.2 Implementación suavizado exponencial en Python

Las librerías utilizadas en la implementación de los algoritmos en Python son Numpy, Pandas, Statsmodels, Sklearn y Keras.

Numpy es una biblioteca de código abierto de Python especialmente útil para la manipulación de matrices numéricas y cálculos científicos. Pandas es otra biblioteca de código abierto especializada en el tratamiento y análisis de gran cantidad de datos almacenados en tablas, así como la gestión, agregación y unión de tablas.

El modelo de suavizado exponencial se ha implementado haciendo uso de las librerías anteriores y la función SimpleExpSmoothig de Statsmodels.

Esta función se utiliza recursivamente dentro de un bucle y cuyos datos de entrada corresponden con el vector de muestras anuales y una constante de suavización (`smoothing_level = 0.6`). El bucle que contiene la función se ejecuta tantas veces como predicciones deben realizarse, es decir, hasta llegar a la última muestra de datos reales.

Una vez obtenido un vector con todas las predicciones, se emplea el algoritmo del RMSE y MAE. Para ello se importa de la librería Sklearn la función `'mean_absolute_error'` y `'mean_squared_error'`. Estas funciones necesitan el vector de predicciones y el vector de datos reales. Hay que tener en cuenta que el número de muestras de ambos vectores deben coincidir, además la posición de la muestra real debe asociarse con su respectiva predicción para poder obtener un resultado correcto.

Por último, se grafican conjuntamente los vectores de las predicciones junto con las observaciones reales de la serie temporal. Para ello se emplea la función plot de la librería Matplotlib.

En la figura 15 se muestra un diagrama de flujo que resume los procesos del algoritmo diseñado y que se ha descrito en este apartado. El primer proceso es la importación de librerías y de las series temporales. En segundo lugar, se filtran las observaciones y se calcula la rentabilidad compuesta. En tercer lugar, se construye una matriz que contenga en cada fila la información necesaria para producir una predicción. El siguiente paso es obtener la propia predicción y por último se calcula el error cometido periodo a periodo.

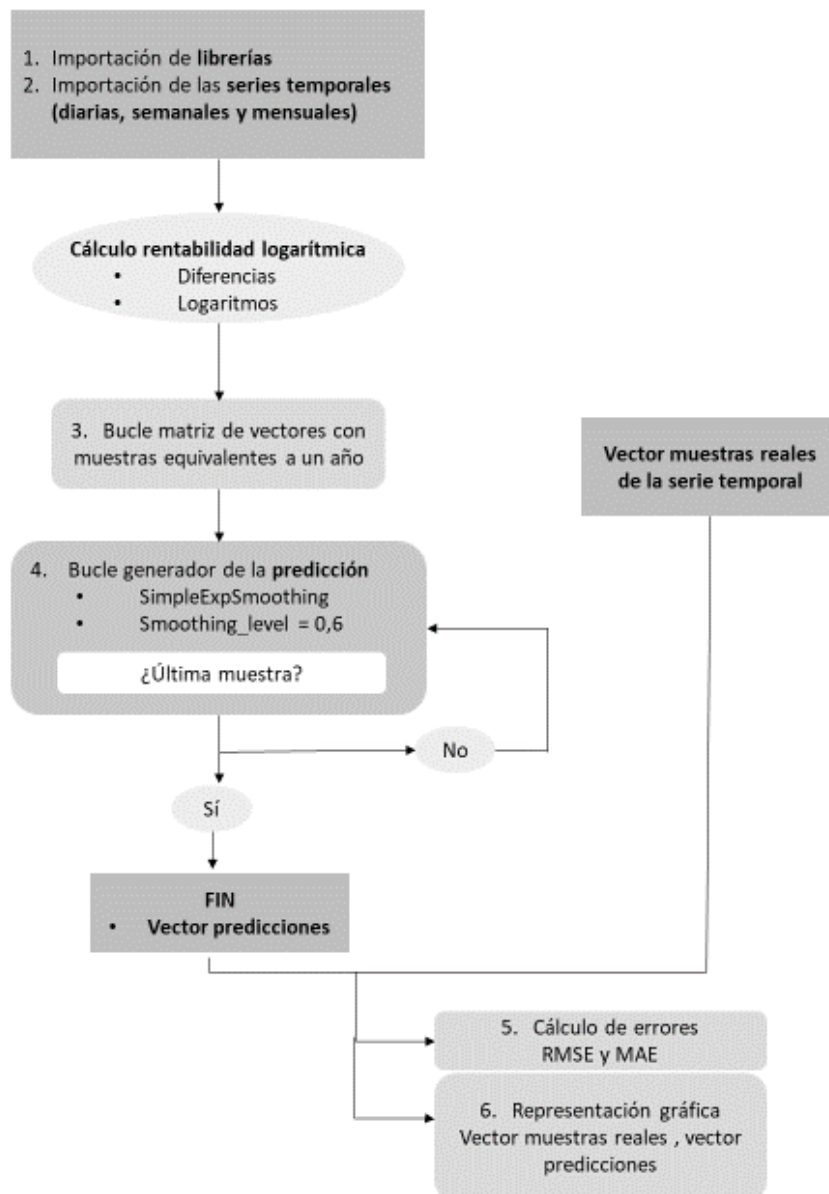


Figura 15. Diagrama de flujo modelo suavizado exponencial.

3.2.3 Implementación ARIMA en Python

Para implementar el modelo ARIMA se utiliza la función SARIMAX de la librería Statsmodels. Esta función implementa un modelo ARIMA estacional con variables exógenas. La metodología empleada en el modelo ARIMA consiste en calcular el parámetro AIC en cada combinación (p, d, q) y quedarnos con aquella que ofrezca menor AIC. Por otro lado, la parte estacional de variables exógenas se fijará a cero (p,d,q x 0,0,0) con la finalidad de que el modelo resultante sea un modelo SARIMA que no tenga en cuenta tendencia ni estacionalidad.

Este procedimiento resulta largo, especialmente en el caso de time-frames diarios, pues la predicción se realiza muestra a muestra, debiéndose generar un vector de 4956¹ predicciones. En definitiva, es clave limitar los tiempos de ejecución del algoritmo y por ende su coste computacional. Por ello se opta en primer lugar por limitar el número de combinaciones (p, d, q). Por ejemplo, se implementa un bucle que recorra todas las combinaciones posibles de (p, d, q) con números enteros comprendidos entre 0 y 2. Otra alternativa que reduce sustancialmente el tiempo de ejecución es imponer una combinación predeterminada. Basándonos en un número elevado de iteraciones se puede identificar la combinación (2,1,2) como la combinación que menor AIC proporciona de forma general.

En la figura 16 se muestra una captura del proceso de cálculo del parámetro AIC del algoritmo ARIMA. Sólo se muestra parte del proceso ya que está programado para realizar todas las combinaciones entre 0 y 2. Por lo que llegaría hasta la combinación (2, 2, 2)x(0, 0, 0). En este caso la mejor combinación es (0, 2, 2)x(0, 0, 0) con un AIC de -20,2067.

```
ARIMA(0, 0, 0)x(0, 0, 0)12 - AIC: -47.559810050265334
ARIMA(0, 0, 1)x(0, 0, 0)12 - AIC: -42.516769271857996
ARIMA(0, 0, 2)x(0, 0, 0)12 - AIC: -35.82955717976
ARIMA(0, 1, 0)x(0, 0, 0)12 - AIC: -36.62627103055044
ARIMA(0, 1, 1)x(0, 0, 0)12 - AIC: -33.65633391916983
ARIMA(0, 1, 2)x(0, 0, 0)12 - AIC: -33.06967347689391
ARIMA(0, 2, 0)x(0, 0, 0)12 - AIC: -22.67315682515282
ARIMA(0, 2, 1)x(0, 0, 0)12 - AIC: -25.619736298312006
ARIMA(0, 2, 2)x(0, 0, 0)12 - AIC: -20.206770169514627
ARIMA(1, 0, 0)x(0, 0, 0)12 - AIC: -45.79357644117553
ARIMA(1, 0, 1)x(0, 0, 0)12 - AIC: -40.24842948527706
ARIMA(1, 0, 2)x(0, 0, 0)12 - AIC: -41.636862054360854
ARIMA(1, 1, 0)x(0, 0, 0)12 - AIC: -37.62945418041955
ARIMA(1, 1, 1)x(0, 0, 0)12 - AIC: -31.88377943922535
```

Figura 16. Extracto de la ejecución del modelo ARIMA.

¹ Las series temporales con time-frames diarios están limitadas a 5206 observaciones, ya que la serie euro-dólar neozelandés es la que menos muestras tiene. Teniendo en cuenta que la ventana de datos anual necesaria para obtener la primera predicción es de 250 observaciones, sólo nos quedarán muestras para realizar 5206-250 = 4956 predicciones.

A continuación, se muestra la figura 17 que contiene el diagrama de flujo del algoritmo diseñado para el modelo de predicción ARIMA. En él se incluyen los procesos y funciones empleadas para obtener las predicciones. A diferencia del suavizado exponencial, debemos obtener primero los órdenes de la parte de autocorrelación y de medias móviles. Para ello se empleará el modelo Akaike y seguido se genera la predicción. El último paso será calcular el error cometido en las predicciones y graficar los resultados junto con las observaciones reales en una misma gráfica.

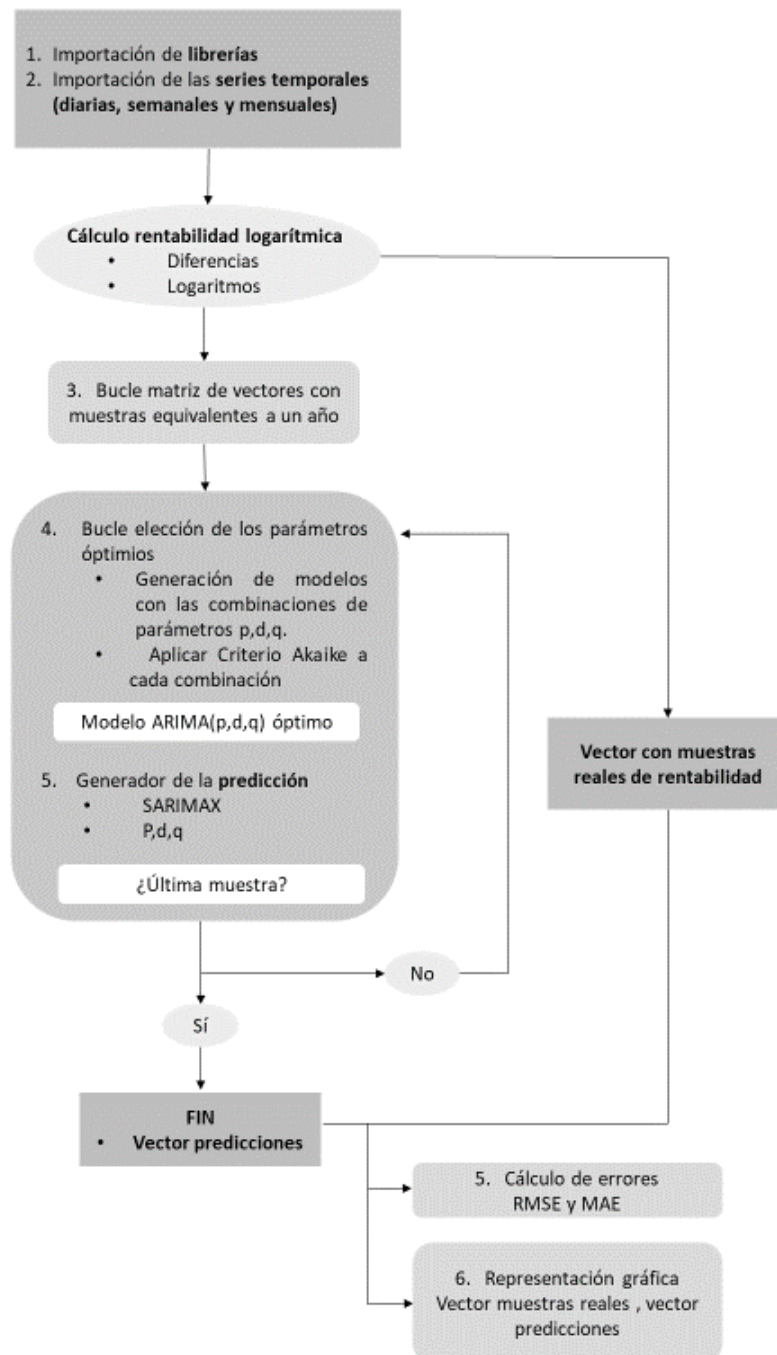


Figura 17. Diagrama de flujo algoritmo modelo ARIMA.

3.2.4 Implementación red neuronal LSTM en Python

La implementación de la Red Neuronal LSTM en Python se hará haciendo uso de la librería de Keras. Keras es una librería de código abierto especializada en funciones de Deep Learning y aprendizaje automático (Machine Learning) útil para construir y entrenar redes neuronales sin necesitar conocimientos profundos sobre la materia ya que los comandos o funciones empleados son bastante intuitivos y accesibles a cualquiera que desee generar su propia Red Neuronal. El procedimiento de implementación del algoritmo requiere ciertos pasos adicionales que los modelos clásicos como el suavizado exponencial y el ARIMA no necesitan.

Antes de manipular los datos es importante filtrar las muestras, y asegurarse que no hay datos de tipo NaN (Not a Number) ya que nos impedirá avanzar en el proceso de entrenamiento de la red neuronal.

En segundo lugar, hay que generar una matriz que contenga los datos que introduciremos a la red neuronal para el entrenamiento. En este caso será una matriz 3D cuyo formato sigue la nomenclatura: [i, j, k], donde i hace referencia al número de predicciones que se pretenden generar, j es la cantidad de observaciones de la ventana de datos equivalentes a un año, depende del data-frame de la serie temporal que se esté analizando y por último k que representa las variables que se tienen en cuenta en la predicción. Como el modelo es univariante, es decir, nos basamos en las propias observaciones de periodos anteriores, k tendrá valor igual a 1. Como ejemplo, la serie temporal Euro-Dólar diario tendrá una matriz asociada de la forma: [4956, 250, 1] y para el vector que almacene las predicciones también habrá que generar una matriz de características [4956, 1], de forma general se representa como [i, k].

Una vez los datos están estructurados, se procederá a su normalización aplicando un escalado que utilizará la función 'MinMaxScaler' y 'fit_transform' de la librería de Keras. Una vez realizada la predicción, se desnormalizarán los datos.

En tercer lugar, se define el modelo. Para ello será imprescindible elegir los parámetros que caracterizarán nuestra red neuronal LSTM. Las rentabilidades se caracterizan por ser datos que no presentan muchas pequeñas variabilidades, por lo que la red neuronal necesitará tener capacidad de almacenar observaciones pasadas para poder generar tendencias que tengan en cuenta el largo plazo. Para ello se necesitará una gran cantidad de unidades de neuronas.

El diseño definitivo de la red neuronal ha empleado 64 capas internas. El optimizador Adams como mecanismo de optimización con una tasa de aprendizaje (learning rate) será de 0.0001. La función de pérdidas, que mide la diferencia entre la predicción y los datos reales, será el Minimum Square Error (MSE) y como función de activación se puede utilizar el sigmoide, la tangente hiperbólica o ReLU.

El proceso de entrenamiento constará de 15 epochs. Es decir, el algoritmo realizará 15 pasadas con los datos de entrenamiento para realizar la predicción, en dicho proceso se ajustan los pesos que parametrizan la red neuronal. Conforme el algoritmo utiliza más los datos de entrenamiento, más se ajustan, pero cabe la posibilidad de que se sobreentrene la red neuronal perjudicando la optimización de los pesos de la red.

Llegados a este punto se pueden modificar otros hiperparámetros del modelo, como por ejemplo el número de epochs, el batch-size o el dropout. Los parámetros del modelo que finalmente se han escogido quedan representados en la siguiente tabla resumen.

Layers (Units)	Epochs	optimizer	Learning rate	Batch size	activation	Función de pérdidas	Dense layer
64	15	Adam	0.0001	-	ReLU	MSE	1

Tabla 4. Especificaciones e hiperparámetros de la Red Neuronal LSTM.

Una vez realizado el entrenamiento, la red neuronal genera la predicción. Dicha predicción será almacenada en un vector de predicciones que se empleará junto con los datos de rentabilidad reales de la serie temporal para calcular el error RMSE y MAE.

El último paso es graficar conjuntamente las muestras reales con las predicciones mediante la función 'plot' de la librería de Python matplotlib.

La figura 18 muestra un resumen de los procesos tenidos en cuenta en la predicción de las rentabilidades futuras mediante la red neuronal LSTM. Esta metodología, a diferencia de los modelos clásicos, necesita normalizar los datos y realizar un entrenamiento. Una vez obtenidas las predicciones, se repite el procedimiento de cálculo del error cometido y se grafica conjuntamente la predicción con las observaciones reales de cada serie temporal.

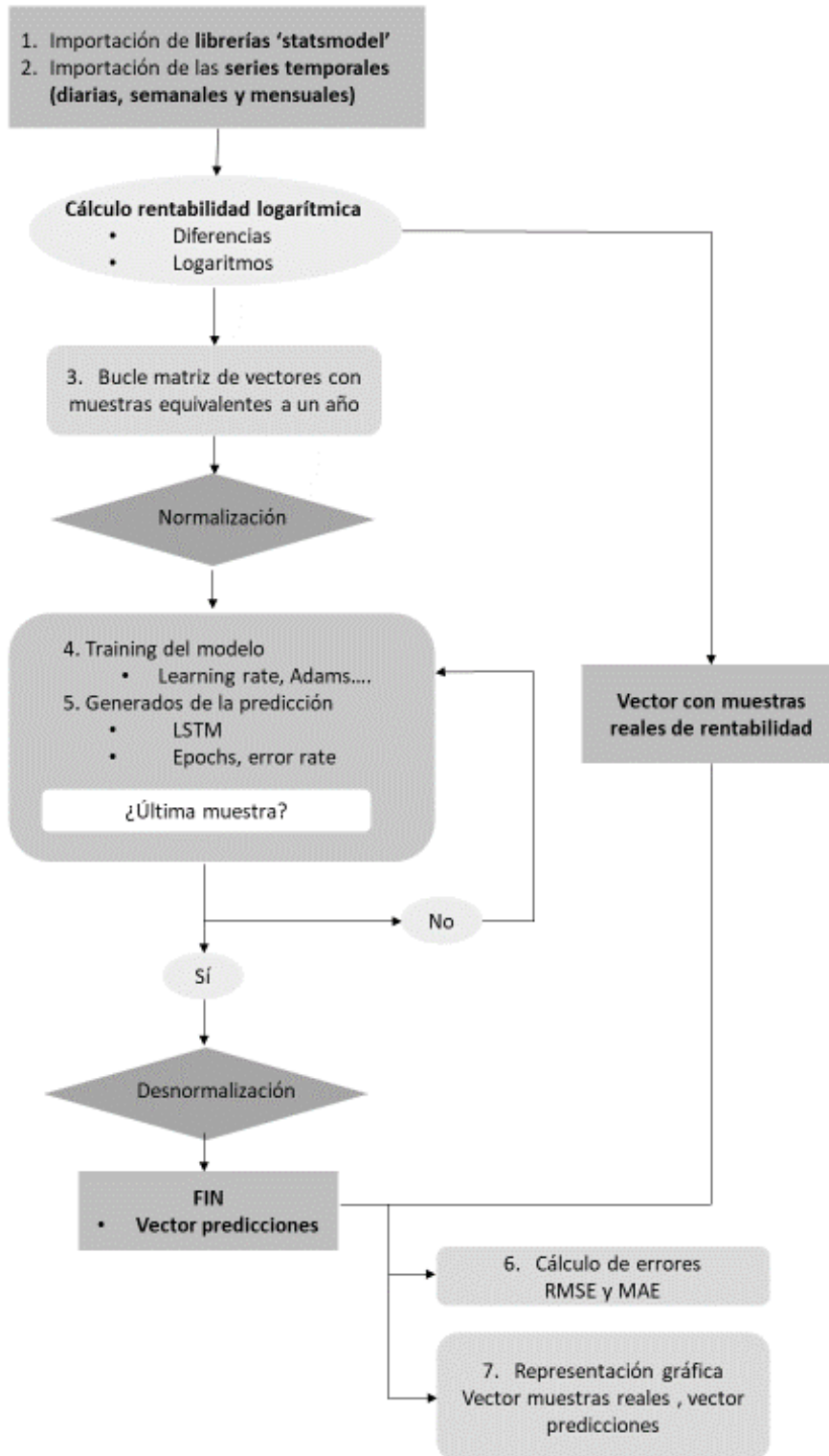


Figura 18. Diagrama de flujo algoritmo Red Neuronal LSTM.

4. EVALUACIÓN DE LA CAPACIDAD PREDICTIVA DE LOS MODELOS APLICADOS AL MERCADO FOREX

En este apartado se compararán los resultados obtenidos tras la ejecución de los códigos implementados en los apartados anteriores con la intención de obtener conclusiones sobre la capacidad predictiva de los modelos clásicos frente al modelo más moderno, basado en redes neuronales LSTM. Utilizar el RMSE y MAE como medida de cuantificación del error cometido en la predicción es de utilidad ya que nos ofrece diversidad de criterios a la hora de comparar las imprecisiones de los modelos predictivos. Una vez calculados los errores, se analizará si existe algún tipo de reducción del RMSE sistemática del modelo basado en redes neuronales LSTM frente al resto de modelos clásicos. También se comparará la influencia del time-frame en la predicción, para ello se tendrán en cuenta los resultados obtenidos de aplicar un modelo concreto a una serie temporal de par de divisas variando sus time-frames. Para finalizar la evaluación, se analizará globalmente cual es la serie temporal que menor error asociado tiene, de forma que podamos extraer conclusiones que distingan entre analizar divisas MAJOR frente a divisas exóticas.

4.1 Predicción de las series temporales con datos diarios

Las series temporales con time-frames diarios tienen la peculiaridad de tener más observaciones que las series con time-frames semanales o mensuales. A partir de los datos diarios se comprobará si los modelos ofrecen errores más pequeños cuando las observaciones contienen información mucho más detallada de la evolución de la serie temporal.

Cuando se tratan time-frames diarios se tiene en cuenta un mayor número de observaciones. Tanto desde el punto de vista del tamaño del vector de datos que nos permiten predecir el periodo siguiente, como desde el número de predicciones totales a realizar, lo cual incrementa el coste computacional.

El análisis diario es ampliamente utilizado por inversores cortoplacistas. Los cuales operan en intervalos de tiempo muy cortos, ya sean diarios o intradía.

El tamaño del vector de datos será el número de observaciones equivalentes a un año de actividad del mercado. Como se ha indicado en el apartado 3.2.1, en time-frames diarios el mercado está activo 250 días y por tanto el vector de observaciones deberá contener la información de la rentabilidad de 250 periodos diarios.

Analizando las series temporales diarias de los tipos de cambio, se observa que el número de observaciones que las componen no son iguales. Para que el cálculo del RMSE sea comparable entre los modelos y series temporales de un mismo time-frame, el tamaño del número de observaciones de las series temporales deberá ser el mismo. De esta forma, atendiendo a la tabla 5, la serie EURNZD indicará el número de observaciones que se tendrán en cuenta en el cálculo de las predicciones. En definitiva, se tendrán en cuenta las 5206 observaciones más recientes de cada serie temporal.

dólar	EUR/USD	EUR/GBP	EUR/NZD
Número de muestras	11 467	11 453	5 206

Tabla 5. Número de muestras de las series temporales diarias de cada tipo de cambio.

Los resultados obtenidos para la serie EUR/USD utilizando el modelo de suavizado exponencial se muestran en la siguiente ilustración. En ella se puede contemplar en azul los datos correspondientes a las observaciones de rentabilidad de la serie EUR/USD, y en naranja las predicciones realizadas por el modelo. Se observa en la ilustración que hay un número de observaciones reales que no tienen una predicción asociada. Esto se debe a que las primeras 250 observaciones se utilizan para obtener la primera predicción. Este hecho habrá que tenerlo en cuenta en el resto de las representaciones gráficas.

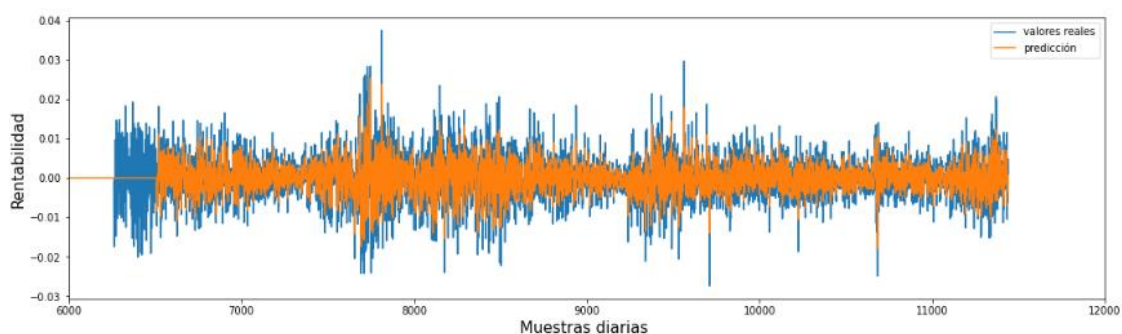


Figura 19. Comparación de los datos reales EUR/USD con la predicción del modelo de suavizado exponencial.

De la gráfica anterior se puede extraer a simple vista que el modelo no está sobredimensionando las predicciones ya que en un primer análisis, las predicciones graficadas en color naranja oscilan dentro del rango de valores de la serie temporal real, el eurodólar, en azul.

Complementando el primer análisis visual, si se observan detalladamente los picos de rentabilidad (tanto alcista como bajista), la predicción se ajusta en amplitud a los datos reales, indicando que el orden cronológico de las predicciones coincide con el orden que le corresponde a su valor real asociado.

Por último, se calcula el error RMSE y MAE. Aplicando el modelo de suavizado exponencial con la serie temporal EUR/USD diaria se obtiene un Root Mean Square Error de 0.00677763 y un Mean Absolute Error de 0.0050533.

Se muestra a continuación, figura 20, los resultados de utilizar la red neuronal LSTM. En la gráfica se representan predicciones y datos reales de la misma moneda EUR/USD diaria utilizada en la figura 19. Pese a ser un modelo más complejo, no se observan a simple vista mejoras sustanciales a partir de su representación gráfica. De forma análoga al suavizado exponencial, las predicciones están dentro del rango de oscilaciones de la serie temporal real, demostrando que el modelo no está sobredimensionado. Las diferencias entre ambos modelos se encontrarán al calcular los errores RMSE y MAE.

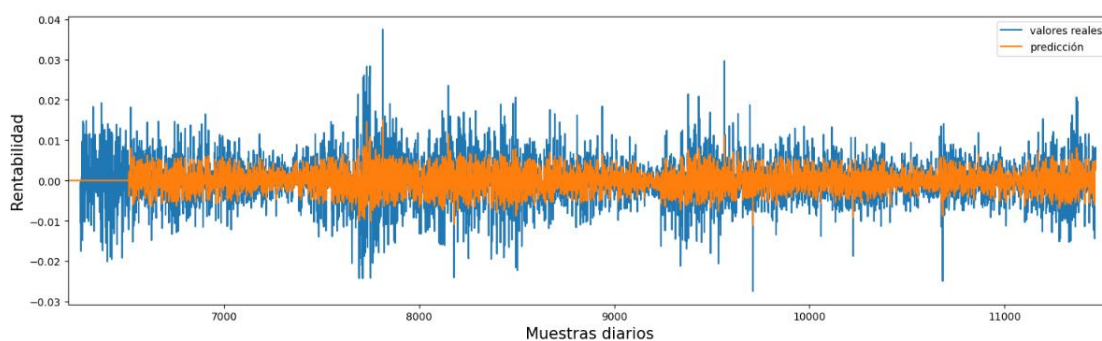


Figura 20. Comparación de los datos reales EUR/USD con la predicción del modelo LSTM.

Para completar el análisis se recogen en tablas todos los errores cometidos para todas las series de datos diarios. Se organizarán en función del modelo empleado y el tipo de moneda: EUR/USD, EUR/GBP y EUR/NZD. Además, se adjuntan a los resultados de RMSE y MAE unas magnitudes que representan en porcentaje la reducción del error cometido con respecto a una referencia. En la siguiente columna a los datos del error, están los porcentajes de reducción del error del modelo ARIMA y LSTM con respecto al modelo de suavizado exponencial. En la columna situada a la

derecha de dichos porcentajes se encuentra el porcentaje de reducción del error del modelo LSTM con respecto al error del modelo ARIMA.

Así se puede comparar la eficacia de cada modelo a partir de sus errores, cuantificando la consistencia de los resultados.

A continuación, se muestran en las tablas 6, 7 y 8 los errores cometidos en las series diarias de los tipos de cambio EUR/USD, EUR/GBP y EUR/NZD utilizando los modelos predictivos de suavizado exponencial, ARIMA y redes neuronales LSTM, junto con las magnitudes de reducción del error.

EUR/USD	RMSE			MAE		
Suavizado Exponencial	0.0067			0.0050		
ARIMA	0.0054	-16%		0.0039	-18%	
LSTM	0.0029	-48%	-45%	0.0021	-50%	-47%

Tabla 6. Resumen de los errores cometidos RMSE y MAE al utilizar datos diarios de la serie EUR/USD.

A partir de la tabla 6 se extrae que el modelo ARIMA reduce un 16% el error cometido con respecto al suavizado exponencial, mientras que el modelo LSTM lo disminuye un 48%. Por último, el modelo LSTM obtiene un error un 45% menor que el modelo ARIMA. A partir de estas conclusiones se extrae que el modelo LSTM reduce drásticamente el error que se comete en las predicciones de los rendimientos de la serie temporal EUR/USD. En las tablas 7 y 8 se muestran los resultados para las series temporales EUR/GBP y EUR/NZD.

EUR/GBP	RMSE			MAE		
Suavizado Exponencial	0.0061			0.0044		
ARIMA	0.0048	-14%		0.0035	-15%	
LSTM	0.0027	-48%	-44%	0.0018	-53%	-46%

Tabla 7. Resumen de los errores cometidos RMSE y MAE al utilizar datos diarios de la serie EUR/GBP.

EUR/NZD	RMSE			MAE		
Suavizado Exponencial	0.0079			0.0058		
ARIMA	0.0065	-23%		0.0047	-21%	
LSTM	0.0035	-51%	-45%	0.0024	-53%	-47%

Tabla 8. Resumen de los errores cometidos RMSE y MAE al utilizar datos diarios de la serie EUR/NZD.

Los resultados recogidos en las tablas anteriores frecen dos conclusiones para los datos diarios, la primera, el modelo de red neuronal LSTM reduce drásticamente el error con respecto a los modelos econométricos clásicos cuando las observaciones son diarias y, en segundo lugar, el par de divisas EUR/NZD tiene asociado un error mayor que las monedas EUR/USD o EUR/GBP.

La figura 21 representa los errores RMSE calculados y recogidos en las tablas anteriores. Se puede observar que el modelo que menos error comete es la red neuronal LSTM. La reducción del error cometido frente a los modelos clásicos es sustancial, por lo que se concluye que las redes neuronales LSTM son la mejor opción a la hora de predecir datos diarios en series temporales del mercado FOREX.

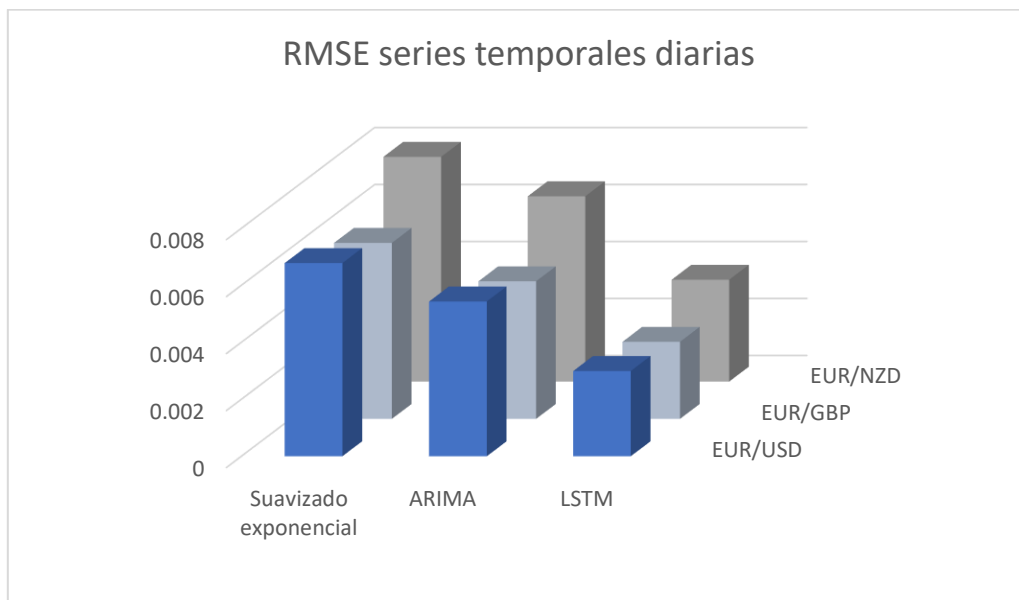


Figura 21. Resultados RMSE en time-frames diarios.

4.2 Serie de datos semanales

Agrupar los datos diarios de cotización en time-frames semanales o mensuales, permite analizar la serie temporal desde un punto de vista más a largo plazo. Los datos semanales pueden compararse con los datos diarios con el fin de observar y contrastar las tendencias del histórico de la serie temporal e incluso las tendencias de órdenes de compra-venta.

Cada observación semanal representa un resumen de todos los datos de interés de las cotizaciones diarias que han ocurrido en ese periodo de tiempo. Por lo que es una forma mucho más rápida de extraer datos agregados. Por ende, realizar la predicción de una muestra en este time-frame implica realizar una predicción de carácter semanal.

Al tener en cuenta periodos semanales en las observaciones, el tamaño del vector de datos empleado para realizar una predicción necesitará 52 observaciones reales con el fin de generar la predicción del periodo próximo. En la tabla 9 se muestra el número de observaciones que componen las series temporales semanales de las tres divisas escogidas.

Divisas	EUR/USD	EUR/GBP	EUR/NZD
Número de muestras	2293	2298	1026

Tabla 9. Número de muestras de las series temporales semanales de cada tipo de cambio.

A continuación, se representan las predicciones del modelo ARIMA y las observaciones reales de la serie temporal semanal del par EUR/NZD en la figura 22. Se puede observar que las predicciones no están sobredimensionadas, además, el modelo identifica las tendencias alcistas y bajistas adaptándose a la evolución de la serie temporal real.

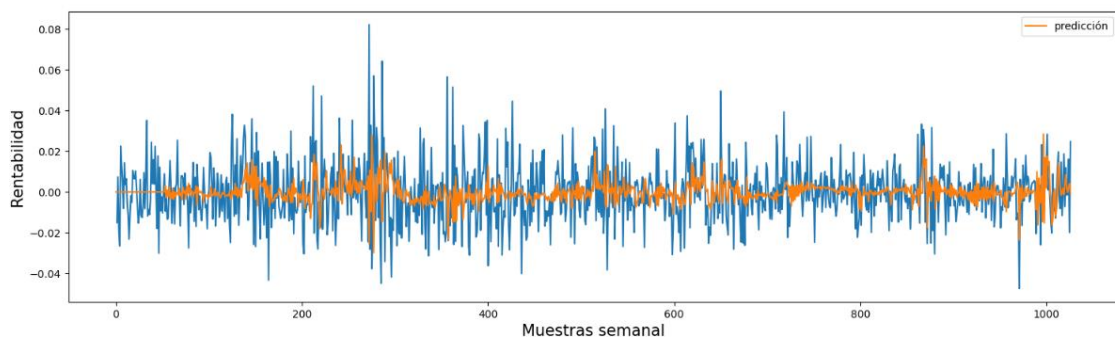


Figura 22. Resultados ARIMA serie temporal EUR/NZD semanal.

Complementando el gráfico anterior, se representa en la figura 23 la predicción de la red neuronal LSTM con respecto a la serie temporal EUR/NZD semanal. De igual forma que el modelo ARIMA, se puede ver a simple vista que la red neuronal también ofrece unos resultados positivos manteniéndose dentro de los rangos reales de la serie temporal estudiada. Pero es visible que el ajuste es sustancialmente mejor, por lo que no solo es capaz de identificar las tendencias alcistas o bajistas en la rentabilidad, sino que también se ajusta a su valor real. Se puede observar que la red neuronal LSTM tiene amplitudes mayores si la comparamos con el modelo ARIMA, ya que muestra mejor adaptación a las oscilaciones de la serie temporal original.

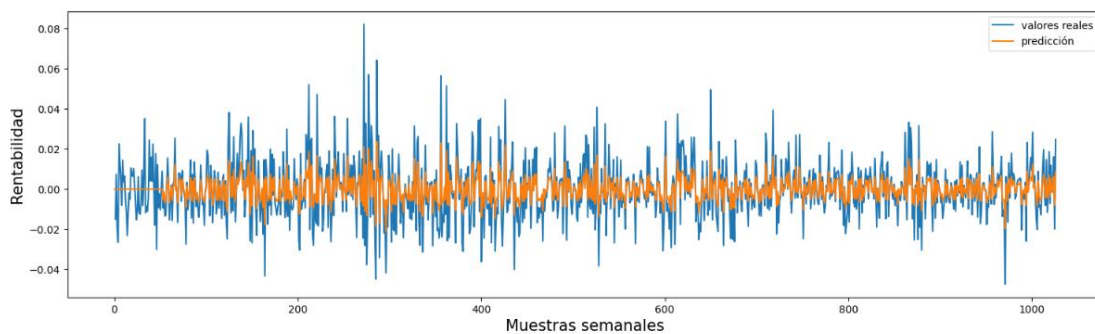


Figura 23. Resultados LSTM en la serie EUR/NZD semanales.

Para complementar los resultados de la red neuronal LSTM con datos semanales, se adjunta en la figura 24 la gráfica con los resultados superpuestos a la serie temporal real de rendimientos EUR/GBP. Al igual que la serie EUR/NZD, los resultados aciertan las tendencias alcistas y bajistas, además de ajustarse positivamente a los valores de amplitud.

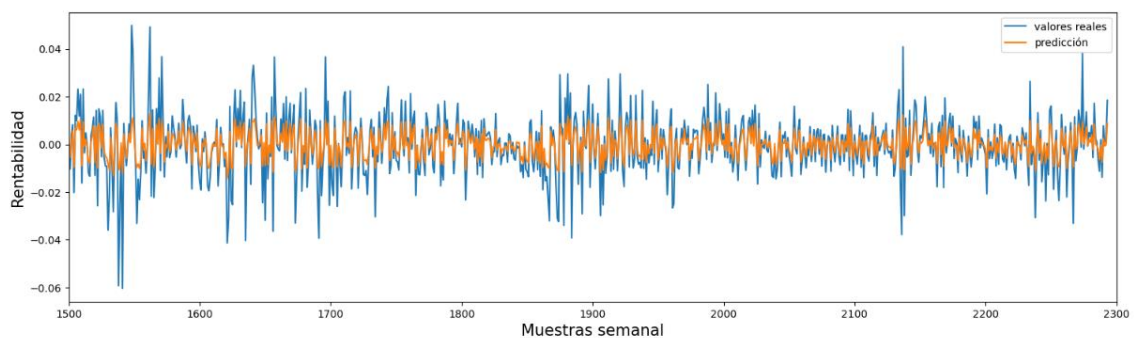


Figura 24. Resultados LSTM en la serie EUR/GBP semanales.

A continuación, se muestran las tablas resumen que recogen los RMSE y MAE calculados a partir de las predicciones de los modelos y las rentabilidades reales. Al igual que las tablas de series

temporales diarias, se comparan en porcentaje la mejora de predicción existente entre los modelos. Para ello se muestra en función del RMSE y MAE porcentajes con respecto al suavizado exponencial y con respecto al ARIMA.

EUR/USD	RMSE			MAE		
Suavizado Exponencial	0.0152			0.0117		
ARIMA	0.0132	-13%		0.0102	-13%	
LSTM	0.0075	-50%	-43%	0.0054	-53%	-46%

Tabla 10. Resumen de los errores RMSE y MAE al utilizar datos semanales de la serie EUR/USD.

EUR/GBP	RMSE			MAE		
Suavizado Exponencial	0.0139			0.0101		
ARIMA	0.0118	-14%		0.0089	-11%	
LSTM	0.0076	-45%	-35%	0.0054	-47%	-40%

Tabla 11. Resumen de los errores RMSE y MAE al utilizar datos semanales de la serie EUR/GBP.

EUR/NZD	RMSE			MAE		
Suavizado Exponencial	0.0178			0.0134		
ARIMA	0.0162	-8%		0.0120	-10%	
LSTM	0.0091	-48%	-43%	0.0066	-50%	-45%

Tabla 12. Resumen de los errores RMSE y MAE al utilizar datos semanales de la serie EUR/NZD.

Las conclusiones de los resultados de los modelos predictivos mantienen que la red neuronal obtiene mejor resultado que los modelos clásicos, ya que el error cometido es menor tanto desde el punto de vista del RMSE como del MAE. Además, analizando los modelos clásicos, el modelo ARIMA tiende a ser más preciso que el suavizado exponencial. Por último, la serie EUR/NZD mantiene sistemáticamente un error mayor que el resto de pares de monedas, corroborando las conclusiones extraídas del apartado anterior al utilizar observaciones diarias.

En la figura 25 se muestran los resultados de los RMSE para las series temporales semanales. A partir de la gráfica se puede comprobar que la red neuronal LSTM reduce su error drásticamente frente a los errores cometidos por modelos clásicos.

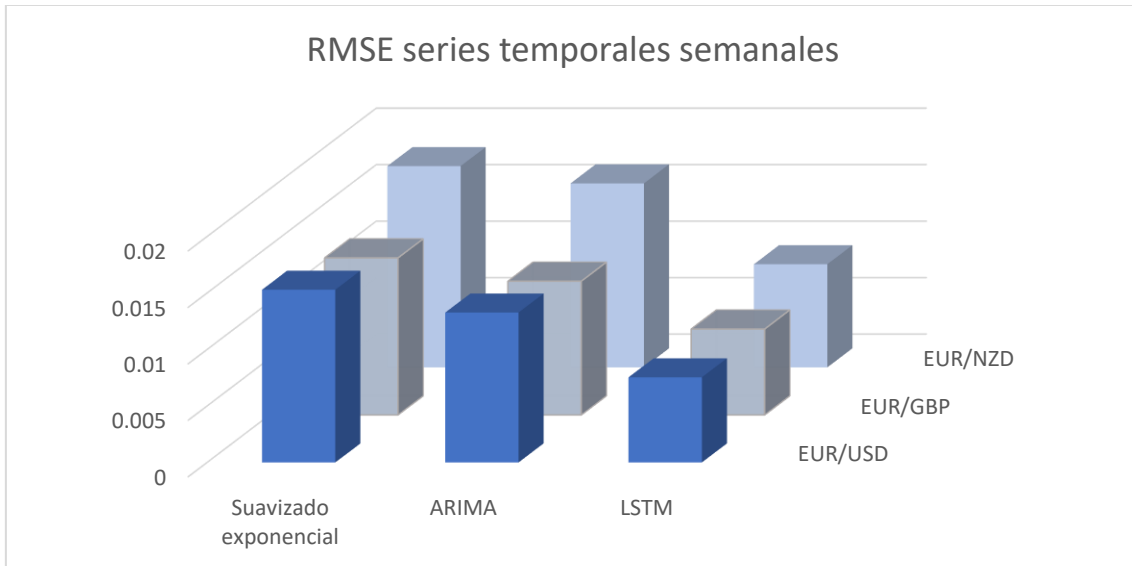


Figura 25. Gráfica 3D de los resultados obtenidos con data-frames semanales.

4.3 Serie de datos mensuales

Realizar predicciones mensuales es más atractivo que una predicción diaria desde el punto de vista del inversor a largo plazo. Cada observación representada en la serie temporal resume la información clave de la cotización durante el periodo de un mes de cotización. Esta información se interpreta como más estable y conservadora en inversiones a largo plazo, ya que permite identificar tendencias de la serie temporal apoyando la toma de decisiones de aquellos que realizan operaciones de cobertura. Este time-frame puede ser especialmente útil para las autoridades monetarias que pretenden modificar la liquidez de sus monedas analizando la cotización a largo plazo y ajustando su política a la economía en función de las predicciones obtenidas.

Los modelos predictivos tendrán la peculiaridad de utilizar información más generalista ya que las observaciones mensuales son una representación de todo lo sucedido a lo largo de un mes de cotización.

El tamaño del vector de observaciones que debe recoger la información equivalente a un periodo anual sólo contendrá 12 observaciones. Recordemos que los modelos que realizaban predicciones diarias utilizaban vectores de 250 datos diarios. La diferencia en el tamaño del vector es significativa, por tanto, a partir del cálculo del RMSE y MAE veremos si el error cometido se compensa con la fiabilidad que dan los datos mensuales al inversor a largo plazo.

A continuación, se muestra una tabla que recoge el número de muestras mensuales de las divisas eurodólar, euro-libra y euro-dólar neozelandés.

Divisa	EUR/USD	EUR/GBP	EUR/NZD
Número de muestras	528	528	236

Tabla 13. Número de muestras de las series temporales mensuales.

La figura 26 representa las predicciones del modelo ARIMA con respecto a los datos reales de la serie temporal del euro-dólar neozelandés. En este ejemplo el modelo ARIMA sobredimensiona claramente sus predicciones, ya que se observa a simple vista que las observaciones calculadas distan en términos de amplitud de los valores reales.

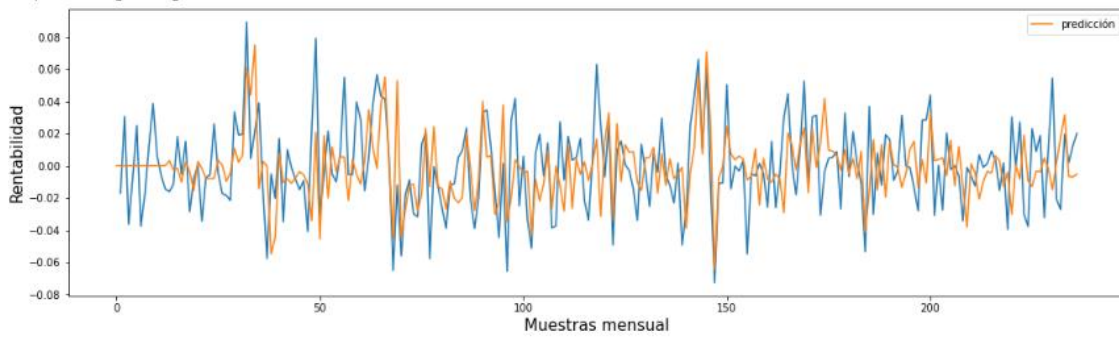


Figura 26. Resultados EUR/NZD ARIMA mensual.

Por contraposición, la figura 27 representa los resultados de las predicciones del modelo de redes neuronales LSTM. Donde las predicciones no están sobredimensionadas y el ajuste a las tendencias alcistas y bajistas se identifica correctamente.

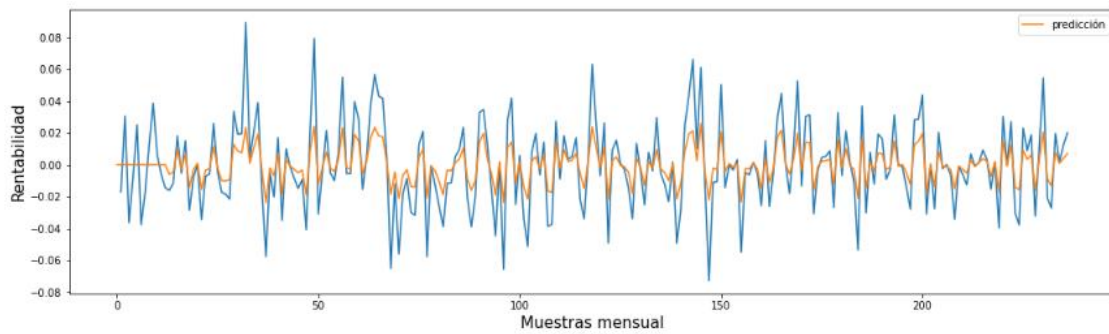


Figura 27. Resultados EUR/NZD LSTM mensual.

Las tablas 14, 15 y 16 recogen los resultados del cálculo del error RMSE y MAE entre las predicciones y las observaciones reales en función del par de divisas utilizado y el modelo de predicción.

EUR/USD	RMSE			MAE		
Suavizado Exponencial	0.0322			0.0241		
ARIMA	0.0271	-19%		0.0196	-21%	
LSTM	0.0166	-55%	-38%	0.0119	-58%	-39%

Tabla 14. Resumen de los errores RMSE y MAE al utilizar datos mensuales de la serie EUR/USD.

EUR/GBP	RMSE			MAE		
Suavizado Exponencial	0.0275			0.0198		
ARIMA	0.0236	-21%		0.0168	-22%	
LSTM	0.0142	-56%	-39%	0.0092	-58%	-45%

Tabla 15. Resumen de los errores RMSE y MAE al utilizar datos mensuales de la serie EUR/GBP.

EUR/NZD	RMSE			MAE		
Suavizado Exponencial	0.0339			0.0264		
ARIMA	0.0259	-17%		0.0208	-19%	
LSTM	0.0165	-54%	-36%	0.0124	-57%	-40%

Tabla 16. Resumen de los errores RMSE y MAE al utilizar datos mensuales de la serie EUR/NZD.

Atendiendo a los resultados de las series temporales mensuales, se puede observar que el modelo ARIMA reduce entre un 17% y un 21% el error cometido por el suavizado exponencial. Por otro lado, el modelo LSTM mejora entre un 54% y un 56% el error del suavizado y entre un 36% y un 39% el error del modelo ARIMA. Por ende, cabe recalcar que el modelo basado en redes neuronales ofrece mejores resultados que cualquiera de los modelos clásicos, reforzando las conclusiones alcanzadas hasta este punto.

Por otro lado, sigue siendo el par EUR/NZD la moneda que mayor error asociado tiene, desde el punto de vista del RMSE como del MAE. Es inevitable interpretar que la moneda exótica, que posee una mayor desviación típica en sus valores reales, empuja a todos los modelos predictivos a cometer un error mayor.

La figura 28 representa todos los errores cometidos entre los modelos clásicos: suavizado exponencial y ARIMA frente al modelo basado en redes neuronales LSTM. Se observa gráficamente una reducción drástica del error si se emplea el modelo de redes neuronales. Esta conclusión sigue la línea de las conclusiones alcanzadas en los apartados 4.3.2 y 4.3.1 donde el

modelo LSTM ofrece los mejores resultados reduciendo el error cometido en la predicción tanto para datos diarios como semanales.

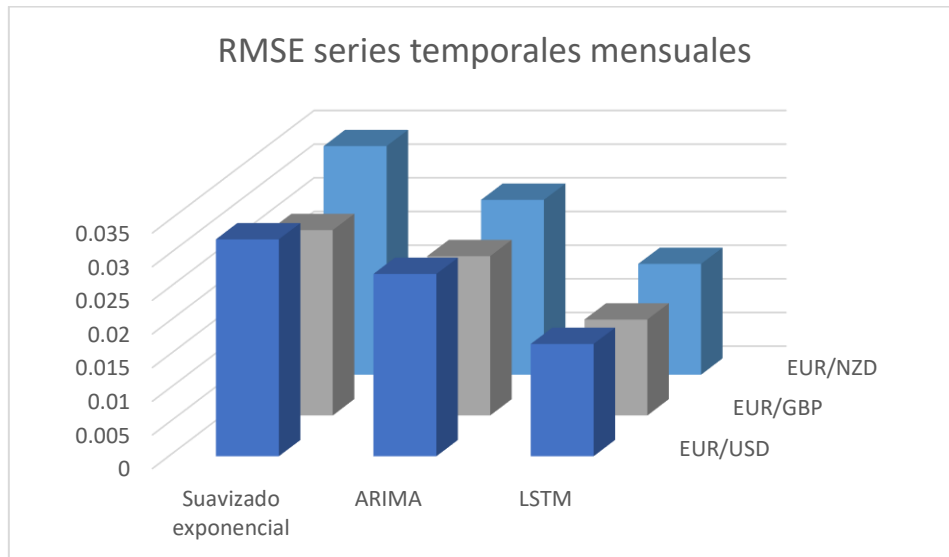


Figura 28. Gráfica resumen del error RMSE en función del modelo y el time-frame.

5. CONCLUSIONES

En primer lugar, las métricas de error RMSE y MAE conducen a las mismas conclusiones sobre los modelos, las divisas y los time-frames estudiados.

Atendiendo a los errores cometidos por los modelos, se puede afirmar que la red neuronal LSTM supera, reduciendo drásticamente el error cometido, a los modelos clásicos: ARIMA y suavizado exponencial. Por tanto, la capacidad predictiva de un modelo moderno es mejor que la de los modelos clásicos ofreciendo mejores resultados desde el punto de vista de aquellos interesados en el análisis de series temporales del mercado FOREX. Partiendo de los gráficos y analizando todas las tablas que recogen los errores RMSE y MAE, se comprueba que el modelo basado en redes neuronales LSTM tiene una mejor capacidad predictiva independientemente del tipo de moneda y el time-frame empleado. Según los resultados de las tablas, la red neuronal es capaz de reducir alrededor de un 50% el error cometido por el modelo de suavizado exponencial y un 45% de media con respecto al modelo ARIMA. Por su parte, el modelo ARIMA es capaz de mejorar un 20% la capacidad predictiva del modelo de suavizado exponencial. En definitiva, la red neuronal LSTM reduce en gran medida los errores cometidos por los modelos econométricos clásicos, siendo una herramienta más atractiva a la hora de generar predicciones a partir de datos de series temporales.

En segundo lugar, comparando los distintos marcos temporales de las muestras se extrae que al emplear time-frames mayores el error de predicción aumenta. Por ejemplo, teniendo en cuenta los resultados del modelo LSTM para la serie EUR/USD con datos diarios, se obtiene un RMSE de 0,0029 (tabla 8). En cambio, el modelo LSTM con la misma serie temporal pero empleando el time-frame mensual, genera un error de 0,0166 (tabla 16). Si se analiza esta desviación del error, se tiene que el uso del time-frame mensual introduce un error 4 veces mayor que el error usando time-frames diarios. Si se extrapola al conjunto de todos los errores, se puede concluir que utilizar datos mensuales incrementa entre 4 y 5 veces el error que se comete con time-frames diarios. Complementariamente, utilizar datos semanales incrementa el error entre 2 y 3 veces con respecto a datos que emplean time-frames diarios. Así se demuestra que para obtener un error más pequeño es recomendable emplear el time-frame más pequeño, en este caso, periodos diarios.

El hecho de que el error se incremente cuando se emplean datos semanales y mensuales, puede ser debido a un entrenamiento desigual de la red neuronal o una escasez de observaciones a la hora de realizar la predicción. Es decir, el vector de muestras que almacena los datos de un año de cotización para obtener la predicción estará compuesto por 250 observaciones si los datos son

diarios, 53 si los datos son semanales y 12 si los datos son mensuales. El hecho de utilizar 12 datos en la predicción frente al empleo de 250 datos de cotización hace pensar que el resultado tiene peor calidad, incrementando el error en la predicción. Teniendo en cuenta esta apreciación, cabe deducir que los modelos mejoran sus rendimientos cuando disponen de más información. Por otro lado, y en vista a los buenos resultados obtenidos con time-frames diarios, se puede advertir que los más favorecidos son aquellos inversores que necesiten predicciones diarias o a corto plazo, en contraposición con aquellos inversores que hagan inversiones a 1 mes.

Por otro lado, atendiendo a los resultados de la serie temporal EUR/NZD, se observa un incremento sistemático del error si se comparara con los resultados de las monedas EUR/USD y EUR/GBP, independientemente del modelo y el time-frame empleado. Partiendo del análisis de la tabla 5 del apartado 3.1 donde la serie temporal EUR/NZD posee la mayor desviación típica de las tres monedas, se concluye que la volatilidad asociada a las series temporales exóticas influye en la predicción de los rendimientos. Por tanto, una mayor desviación típica ocasionada por la alta volatilidad de la cotización de la moneda dará lugar a un error de predicción mayor.

Cabe resaltar que, aunque los resultados de la red neuronal son en su conjunto muy buenos, permitiéndonos concluir que domina completamente a los modelos clásicos, el coste computacional asociado es sustancialmente más elevado que el de los modelos clásicos. Esto se debe en gran medida a que el modelo LSTM precisa un entrenamiento de la red neuronal, mientras que los otros modelos no, de forma que los modelos clásicos requieren capacidades computacionales menores.

En vistas al futuro, y conocedores de los esfuerzos realizados en mejorar el rendimiento y la velocidad del procesado de los datos, se puede afirmar que el coste computacional será un problema cada vez menos importante, ya que se irán consiguiendo capacidades de computación mayores y, por tanto, lo que a priori es la mayor desventaja al emplear modelos basados en redes neuronales, se irá convirtiendo en un problema secundario. La precisión en los resultados que es capaz de alcanzar un modelo de inteligencia artificial bien entrenado justifica en el corto y largo plazo los costes computacionales.

6. COMPROMISO, ODS Y LÍNEAS FUTURAS

6.1 Compromiso y ODS

Este trabajo tiene un compromiso claro con la innovación y la mejora de las técnicas actuales en el campo de la econometría y el análisis de las series temporales. Además de una fuerte motivación por los problemas de la economía actual reflejados en el estudio de las divisas internacionales, reflejo de la situación macroeconómica inédita que estamos presenciando, junto con los estímulos y las políticas que las autoridades monetarias vienen realizando durante los últimos años. En un escenario tan confuso, se hace imprescindible el desarrollo de herramientas modernas que faciliten a la sociedad y en concreto, a aquellos agentes económicos interesados, la toma de decisiones.

En resumen, este trabajo persigue el objetivo de desarrollo sostenible 9: industria, innovación e infraestructuras. Principalmente por la innovación que supone emplear modelos de predicción modernos de inteligencia artificial en el análisis de las series temporales, campo dominado generalmente por los modelos econométricos clásicos. Por otro lado, las conclusiones alcanzadas invitan a reflexionar sobre la influencia de las políticas monetarias de los grandes Bancos Centrales, que controlan las monedas más importantes del mundo en el mercado internacional FOREX, en las monedas exóticas cuyo curso y liquidez se ve limitado en el mercado internacional en detrimento de la confianza que los inversores puedan depositar en sus economías. Claramente relacionado con el ODS número 16: Paz, justicia e instituciones sólidas.



Ilustración 1. Objetivos de desarrollo sostenible

6.2 Líneas futuras

El análisis realizado permite extraer conclusiones a partir de la idoneidad de utilizar modelos modernos basados en Machine Learning frente a modelos clásicos. Teniendo en cuenta que los modelos modernos ofrecen mejores resultados, es posible continuar el estudio introduciendo modelos que combinen cualidad modernas y clásicas, es el caso del neurocomputing (Zhang, 2003).

También será práctico profundizar en la modelización de otras redes neuronales, por ejemplo, Modified LSTM (Melis *et al.*, 2019) que introduce operaciones no lineales o Transformers (Vaswani *et al.*, 2017). El objetivo principal es encontrar un modelo que permita obtener errores de predicción menores.

Otra línea de trabajo futuro es diseñar modelos que ofrezcan resultados semejantes, pero con costes computacionales menores. Es decir, tratar de reducir el tiempo de entrenamiento de la red neuronal. Para lo cual se podría profundizar en la optimización de los hiperparámetros de la red neuronal LSTM implementada mediante un proceso de tuning.

Finalmente, sería de interés introducir otro tipo de activo financiero, por ejemplo, otra moneda presente en el mercado FOREX o incluso el oro, ejemplo de activo refugio. Con el fin de obtener resultados que nos permitan comprender el presente y futuro del entorno macroeconómico que nos rodea.

7. BIBLIOGRAFÍA

- Baillie, R. T., & McMahon, P. C. (1990). *The foreign exchange market: Theory and econometric evidence*. Cambridge University Press. DOI.: 10.2307/2554948
- Box, G. E., & Jenkins, G. M. (1973). Some comments on a paper by Chatfield and Prothero and on a review by Kendall. *Journal of the Royal Statistical Society. Series A (General)*, 136(3), págs: 337-352. DOI.: 10.2307/2344995
- Brown, L. D. (1986). Fundamentals of statistical exponential families: with applications in statistical decision theory. *Ims*. DOI.: 10.1214/Inms/1215466762
- Cipra, T., & Hanzák, T. (2008). Exponential smoothing for irregular time series. *Kybernetika*, 44(3), 385-399. DOI.: 10.14736/kyb-2015-6-0973
- Frankel, J. A., & Froot, K. A. (1990). Chartists, fundamentalists, and trading in the foreign exchange market. *The American Economic Review*, 80(2), 181-185. DOI.: 10.3386/w3470
- Gardner Jr, E. S. (1985). Exponential smoothing: The state of the art. *Journal of forecasting*, 4(1), 1-28. DOI.: 10.1002/for.3980040103
- Hillmer, S. C., & Tiao, G. C. (1982). An ARIMA-model-based approach to seasonal adjustment. *Journal of the American Statistical Association*, 77(377), 63-70. DOI.: 10.1080/01621459.1982.10477767
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780. 10.1162/neco.1997.9.8.1735
- Kalekar, P. S. (2004). Time series forecasting using holt-winters exponential smoothing. *Kanwal Rekhi school of information Technology*, 4329008(13), 1-13. DOI.: 10.22630/prs.2022.22.2.6
- Khan, S., & Alghulaiakh, H. (2020). ARIMA model for accurate time series stocks forecasting. *International Journal of Advanced Computer Science and Applications*, 11(7). DOI.: 10.14569/ijacsa.2020.0110765
- King, M. R., Osler, C., & Rime, D. (2012). Foreign exchange market structure, players, and evolution. *Handbook of exchange rates*, 1-44. DOI.: 10.1002/9781118445785.ch1
- Lee, J., & Chinn, M. (1998). The Current Account and the Real Exchange Rate: A Structural VAR Analysis of Major Currencies. DOI.: 10.3386/w6495
- Liu, L. M. (1989). Identification of seasonal ARIMA models using a filtering method. *Communications in Statistics-Theory and Methods*, 18(6), 2279-2288. DOI.: 10.1080/03610928908830035

- Lu, Wenjie., Li, J., Li, Y., Sun, A., & Wang, J. (2020). A CNN-LSTM-based model to forecast stock prices. *Complexity*, 2020, 1-10. DOI.: 10.1155/2020/6622927
- Makridakis, S., & Hibon, M. (1979). Accuracy of forecasting: An empirical investigation. *Journal of the Royal Statistical Society: Series A (General)*, 142(2), 97-125. DOI.: 10.2307/2345077
- Makridakis, S., & Hibon, M. (1991). Exponential smoothing: The effect of initial values and loss functions on post-sample forecasting accuracy. *International Journal of Forecasting*, 7(3), 317-330. DOI.: 10.1016/0169-2070(91)90005-g
- Makridakis, S., & Hibon, M. (1997). ARMA models and the Box–Jenkins methodology. *Journal of forecasting*, 16(3), 147-163. DOI.: 10.1002/(SICI)1099-131X
- Melis, G., Kočíský, T., & Blunsom, P. (2019). Mogrifier lstm. *arXiv preprint arXiv:1909.01792*. DOI.: 10.48550/arXiv.1909.01792
- Mondal, P., Shit, L., & Goswami, S. (2014). Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices. *International Journal of Computer Science, Engineering and Applications*, 4(2), 13. DOI.: 10.5121/ijcsea.2014.4202
- Montoya Zapata, C., & Corredor Roa, C. A. (2022). Aplicación de modelos de Deep Learning y Machine Learning para el pronóstico de la serie de tiempo del par de divisas Euro-Dólar.
- Nelson, D. M., Pereira, A. C., & De Oliveira, R. A. (2017, May). Stock market's price movement prediction with LSTM neural networks. In *2017 International joint conference on neural networks (IJCNN)* (pp. 1419-1426). Ieee. DOI.: 10.1109/ijcnn.2017.7966019
- Ostertagová, E., & Ostertag, O. (2011, September). The simple exponential smoothing model. In *The 4th International Conference on modelling of mechanical and mechatronic systems, Technical University of Košice, Slovak Republic, Proceedings of Conference* (pp. 380-384). DOI.: 10.2478/v10198-012-0034-2
- Robert G. Brown, Richard F. Meyer, (1961) The Fundamental Theorem of Exponential Smoothing. *Operations Research* 9(5):673-685. DOI.: 10.1287/opre.9.5.673
- Roondiwala, M., Patel, H., & Varma, S. (2017). Predicting stock prices using LSTM. *International Journal of Science and Research (IJSR)*, 6(4), 1754-1756.
- Sakamoto, Y., Ishiguro, M., & Kitagawa, G. (1986). Akaike information criterion statistics. *Dordrecht, The Netherlands: D. Reidel*, 81(10.5555), 26853. DOI.: 10.2307/2348776
- Siami-Namini, S., & Namin, A. S. (2018). Forecasting economics and financial time series: ARIMA vs. LSTM. *arXiv preprint arXiv:1803.06386*. DOI.: 10.1109/icmla.2018.00227

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, H., Wang, J., Cao, L., Li, Y., Sun, Q., & Wang, J. (2021). A stock closing price prediction model based on CNN-BiSLSTM. *Complexity*, 2021, 1-12. DOI: 10.1155/2021/5360828

Wang, W., & Lu, Y. (2018, March). Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model. In *IOP conference series: materials science and engineering* (Vol. 324, No. 1, p. 012049). IOP Publishing. DOI: 10.1088/1757-899x/324/1/012049

Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management science*, 6(3), 324-342. DOI: 10.1287/mnsc.6.3.324

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50, 159-175. DOI: 10.1016/s0925-2312(01)00702-0

8. ANEXO I



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ANEXO I. RELACIÓN DEL TRABAJO CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE DE LA AGENDA 2030

Anexo al Trabajo de Fin de Grado y Trabajo de Fin de Máster: Relación del trabajo con los Objetivos de Desarrollo Sostenible de la agenda 2030.

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.		x		
ODS 2. Hambre cero.			x	
ODS 3. Salud y bienestar.				x
ODS 4. Educación de calidad.		x		
ODS 5. Igualdad de género.			x	
ODS 6. Agua limpia y saneamiento.				x
ODS 7. Energía asequible y no contaminante.			x	
ODS 8. Trabajo decente y crecimiento económico.		x		
ODS 9. Industria, innovación e infraestructuras.	x			
ODS 10. Reducción de las desigualdades.		x		
ODS 11. Ciudades y comunidades sostenibles.		x		
ODS 12. Producción y consumo responsables.			x	
ODS 13. Acción por el clima.			x	
ODS 14. Vida submarina.				x
ODS 15. Vida de ecosistemas terrestres.				x
ODS 16. Paz, justicia e instituciones sólidas.	x			
ODS 17. Alianzas para lograr objetivos.		x		

Descripción de la alineación del TFG/TFM con los ODS con un grado de relación más alto.

***Utilice tantas páginas como sea necesario.

Capítulo 6, Compromiso y ODS.
Página 58.

ODS número 9: Industria, innovación e infraestructuras.
ODS número 16: Paz, justicia e instituciones sólidas.



Anexo al Trabajo de Fin de Grado y Trabajo de Fin de Máster: Relación del trabajo con los Objetivos de Desarrollo Sostenible de la agenda 2030. (Numere la pàgina)

Este trabajo tiene un compromiso claro con la innovación y la mejora de las técnicas actuales en el campo de la econometría y el análisis de las series temporales. Además de una fuerte motivación por los problemas de la economía actual reflejados en el estudio de las divisas internacionales, reflejo de la situación macroeconómica inédita que estamos presenciando, junto con los estímulos y las políticas que las autoridades monetarias vienen realizando durante los últimos años. En un escenario tan confuso, se hace imprescindible el desarrollo de herramientas modernas que faciliten a la sociedad y en concreto, a aquellos agentes económicos interesados, la toma de decisiones.

En resumen, este trabajo persigue el objetivo de desarrollo sostenible 9: industria, innovación e infraestructuras. Principalmente por la innovación que supone emplear modelos de predicción modernos de inteligencia artificial en el análisis de las series temporales, campo dominado generalmente por los modelos econométricos clásicos.

Por otro lado, las conclusiones alcanzadas invitan a reflexionar sobre la influencia de las políticas monetarias de los grandes Bancos Centrales, que controlan las monedas más importantes del mundo en el mercado internacional FOREX, en las monedas exóticas cuyo curso y liquidez se ve limitado en el mercado internacional en detrimento de la confianza que los inversores puedan depositar en sus economías. Claramente relacionado con el ODS número 16: Paz, justicia e instituciones sólidas.

9. ANEXO 2

El anexo está compuesto por todas las gráficas que han sido extraídas para el análisis de los modelos, las series temporales y los time-frame. Se organizan según el orden de aparición en el trabajo, primero los time-frames diarios, después semanales y por último diarios. Dentro de cada time-frame se utilizan los tres modelos de predicción aplicados a los 3 pares de monedas.

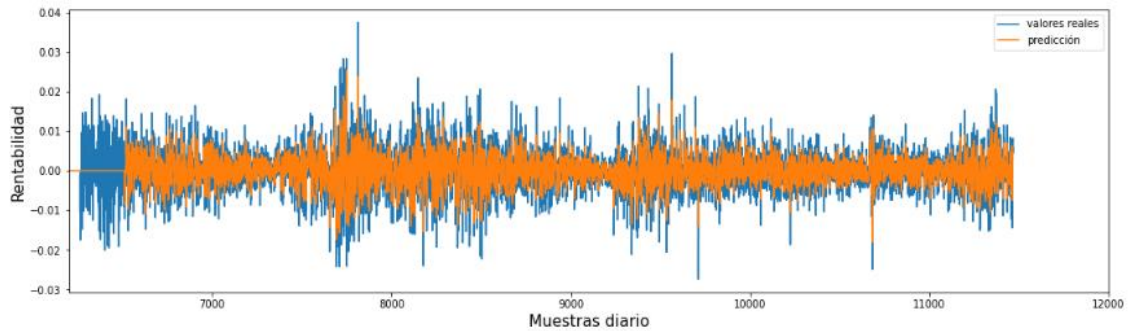


Figura Anexo 1. EUR/USD time-frame diario con suavizado exponencial.

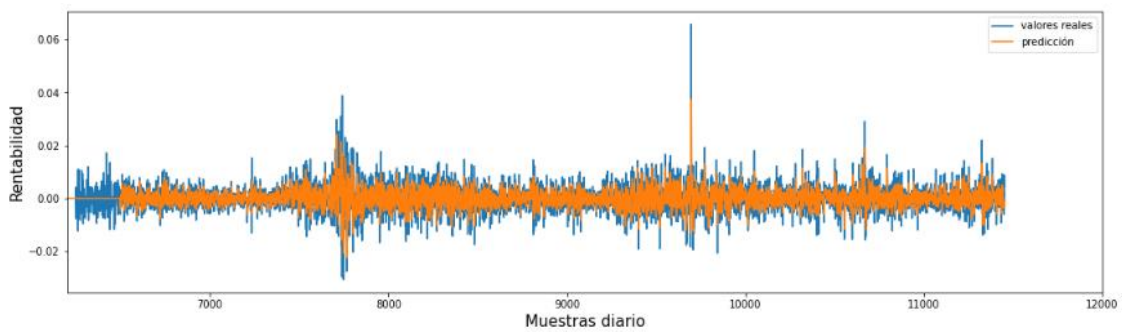


Figura Anexo 2. EUR/GBP time-frame diario con suavizado exponencial.

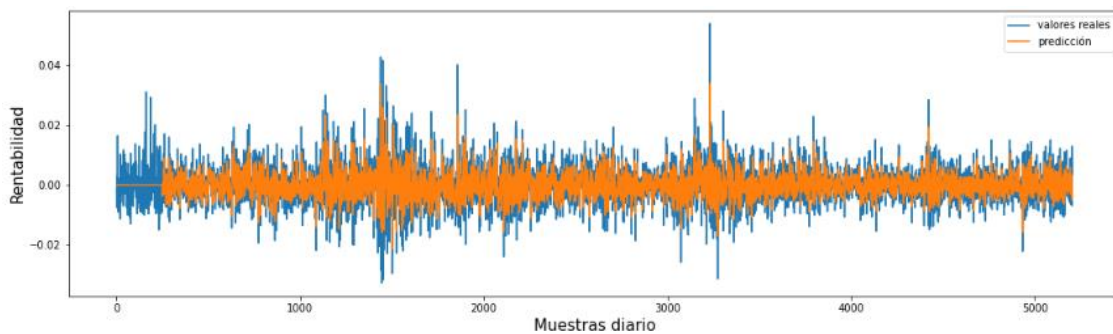


Figura Anexo 3. EUR/NZD time-frame diario con suavizado exponencial.

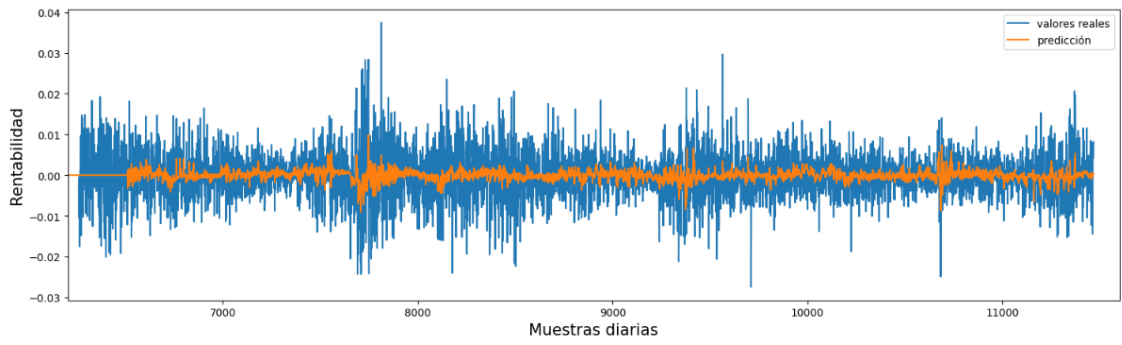


Figura Anexo 4. EUR/USD time-frame diario con ARIMA

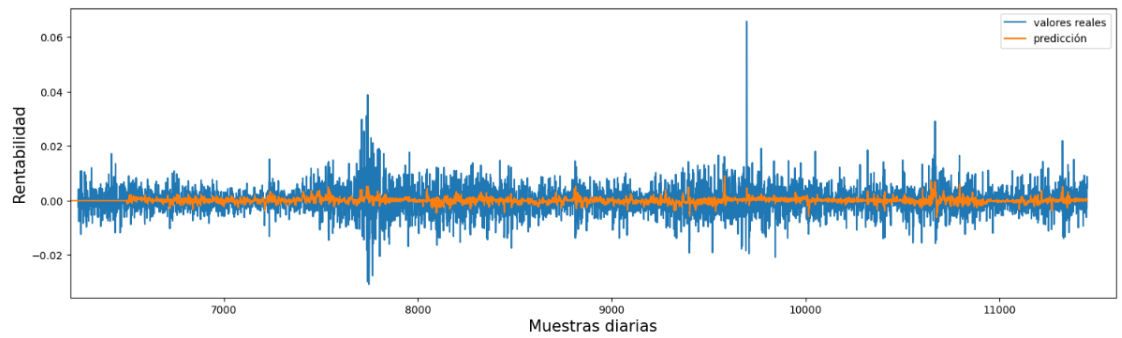


Figura Anexo 5. EUR/GBP time-frame diario con ARIMA.

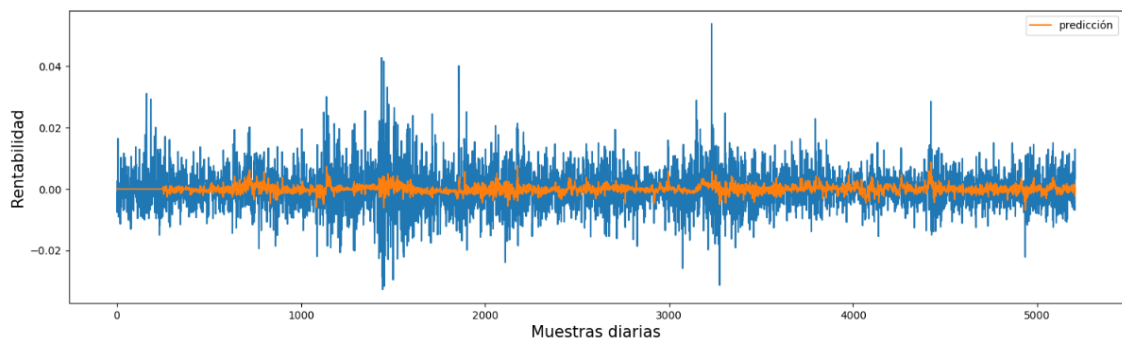


Figura Anexo 6. EUR/NZD time-frame diario con ARIMA.

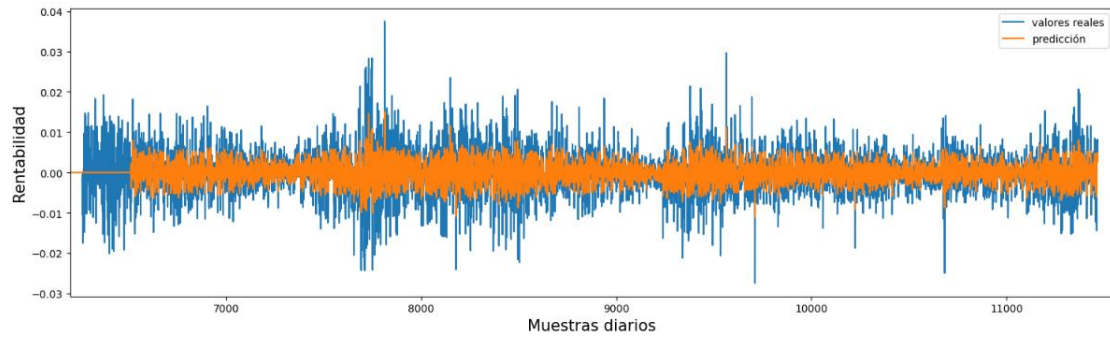


Figura Anexo 7. EUR/USD time-frame diario con LSTM.

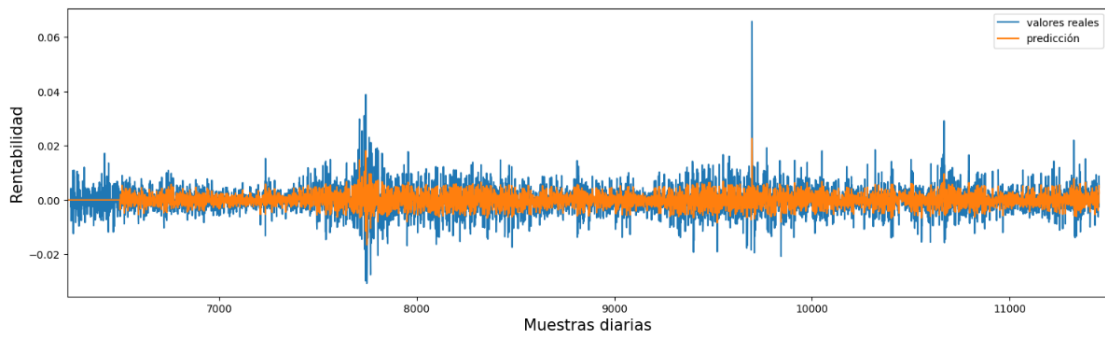


Figura Anexo 8. EUR/GBP time-frame diario con LSTM.

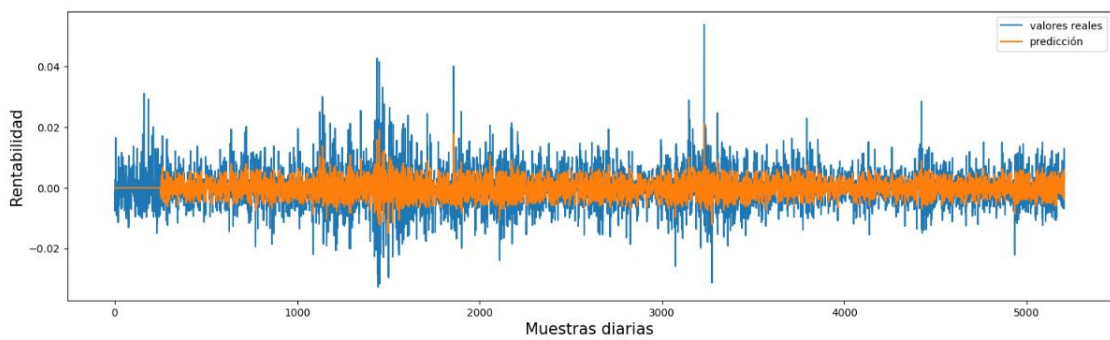


Figura Anexo 9. EUR/NZD time-frame diario con LSTM.

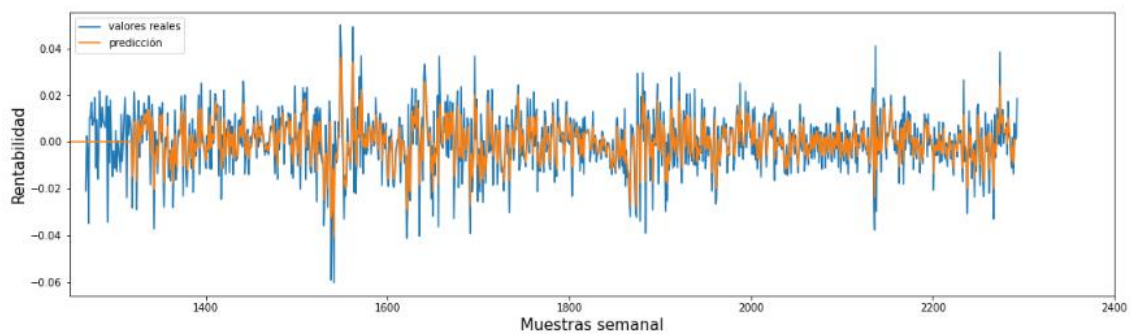


Figura Anexo 10. EUR/USD time-frame semanal con suavizado exponencial.

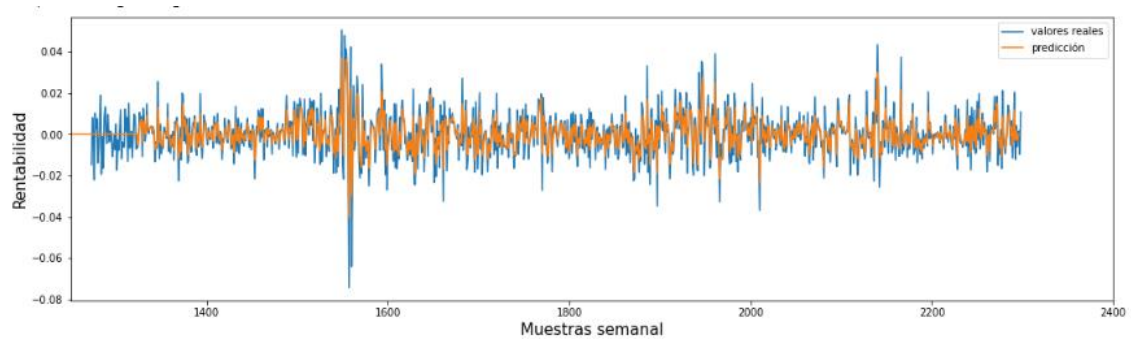


Figura Anexo 11. EUR/GBP time-frame semanal con suavizado exponencial.

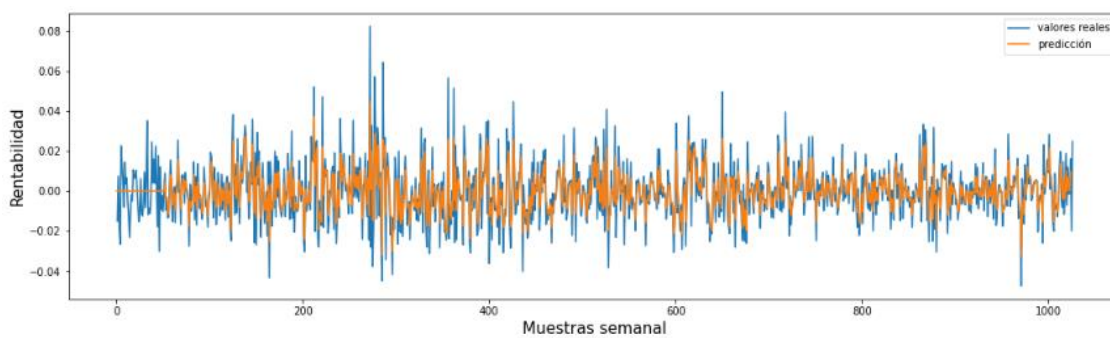


Figura Anexo 12. EUR/NZD time-frame semanal con suavizado exponencial.

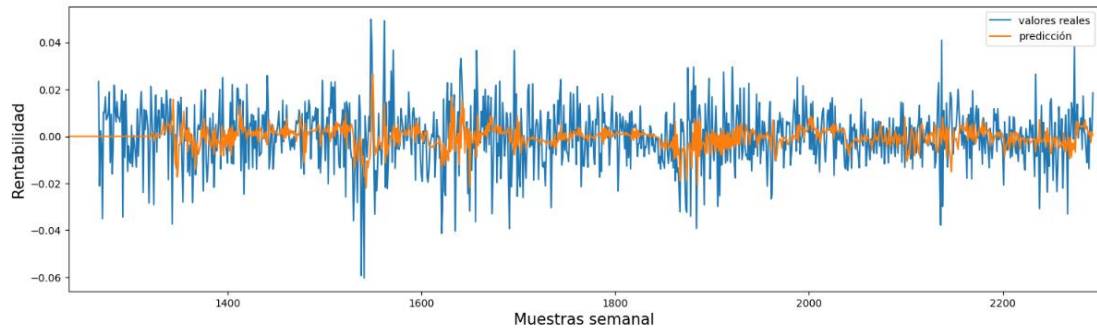


Figura Anexo 13. EUR/USD time-frame semanal con ARIMA.

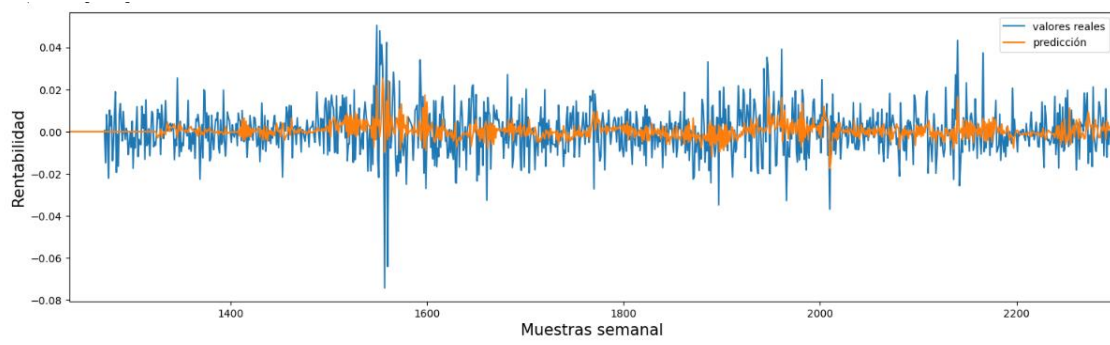


Figura Anexo 14. EUR/GBP time-frame semanal con ARIMA.

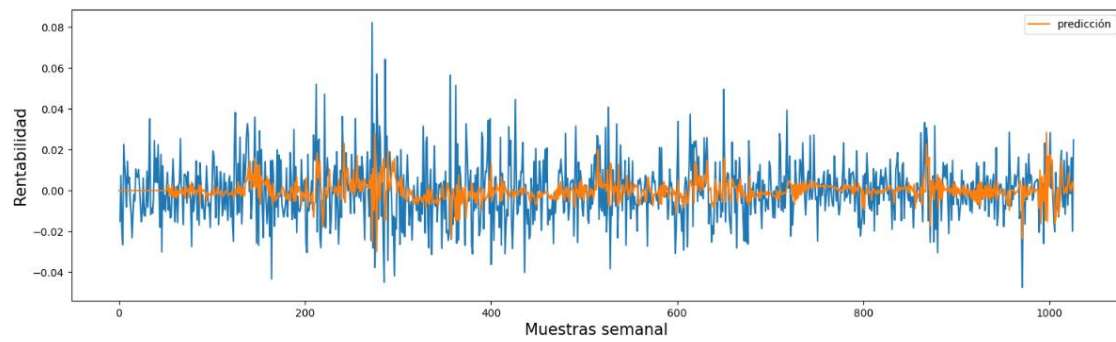


Figura Anexo 15. EUR/NZD time-frame semanal con ARIMA.

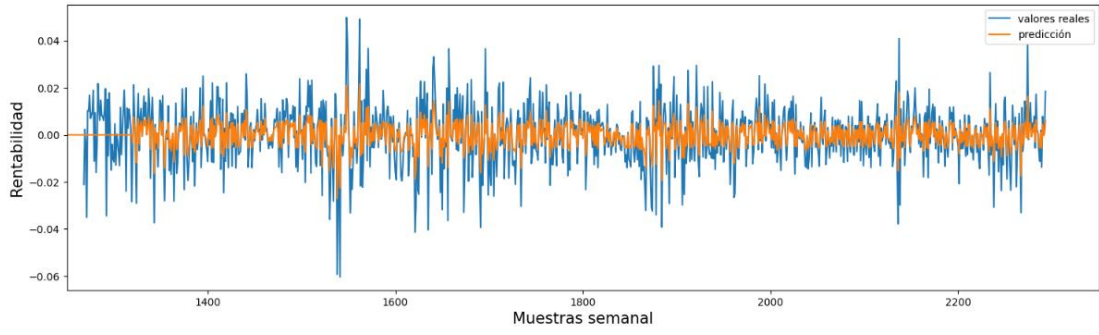


Figura Anexo 16. EUR/USD time-frame semanal con LSTM.

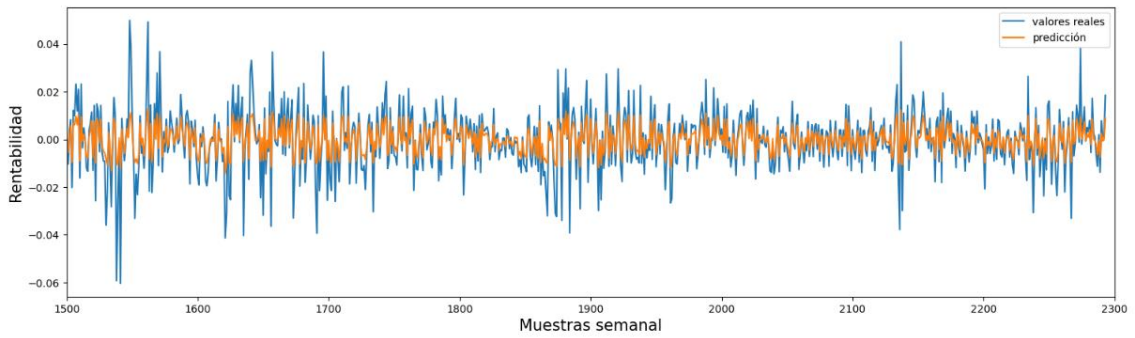


Figura Anexo 17. EUR/GBP time-frame semanal con LSTM.

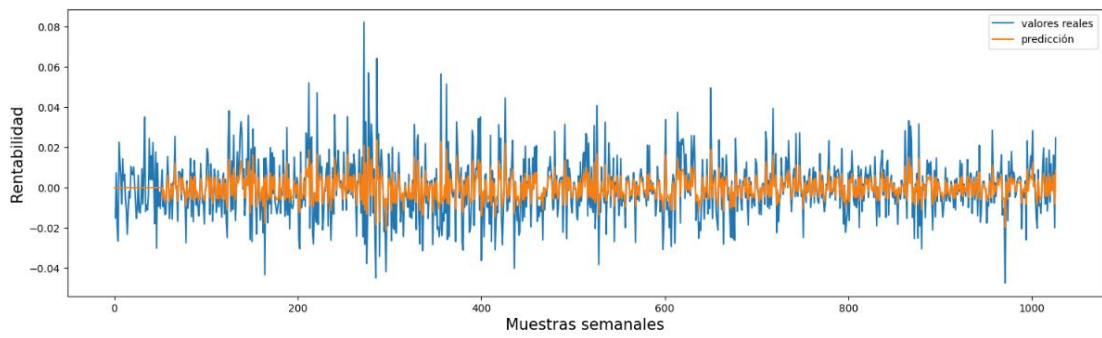


Figura Anexo 18. EUR/NZD time-frame semanal con LSTM.

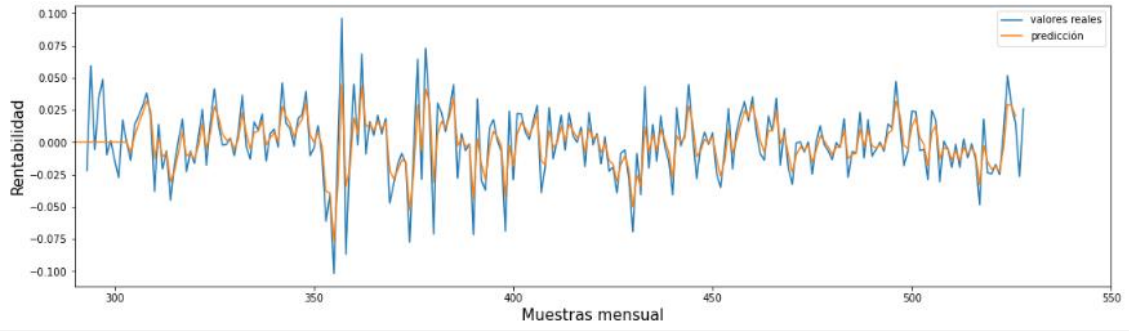


Figura Anexo 19. EUR/USD time-frame mensual con suavizado exponencial.

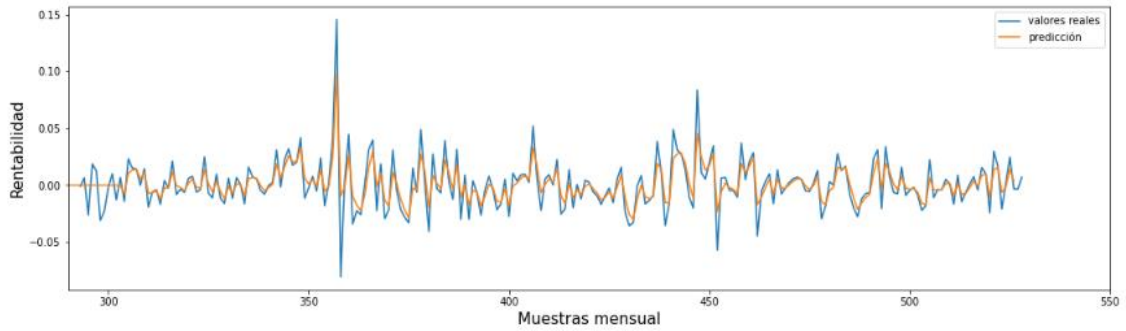


Figura Anexo 20. EUR/GBP time-frame mensual con suavizado exponencial.

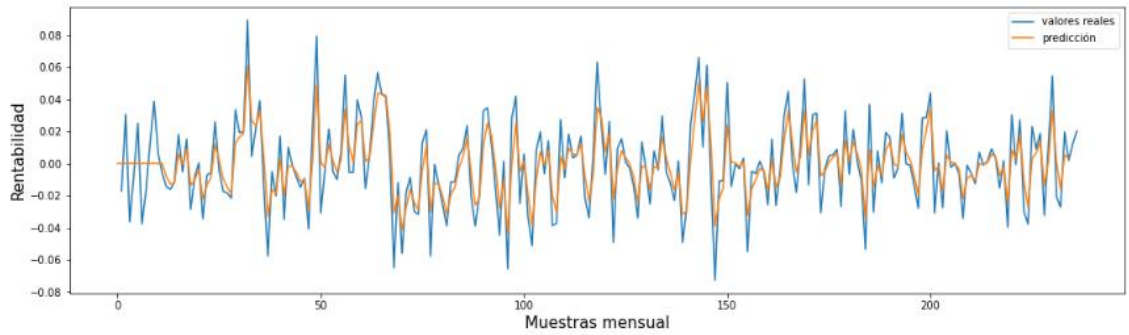


Figura Anexo 21. EUR/NZD time-frame mensual con suavizado exponencial.

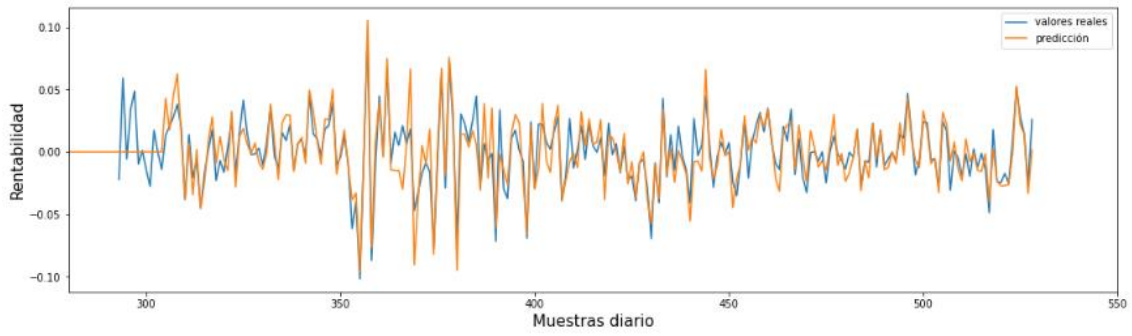


Figura Anexo 22. EUR/USD time-frame mensual con ARIMA.

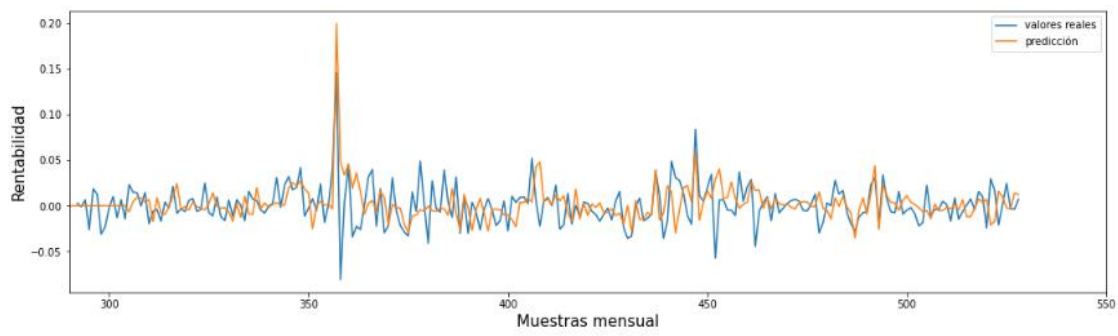


Figura Anexo 23. EUR/GBP time-frame mensual con ARIMA.

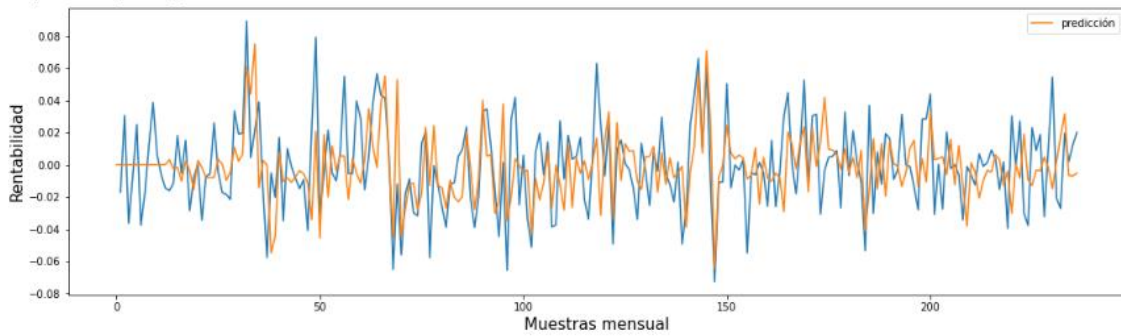


Figura Anexo 24. EUR/NZD time-frame mensual con ARIMA.

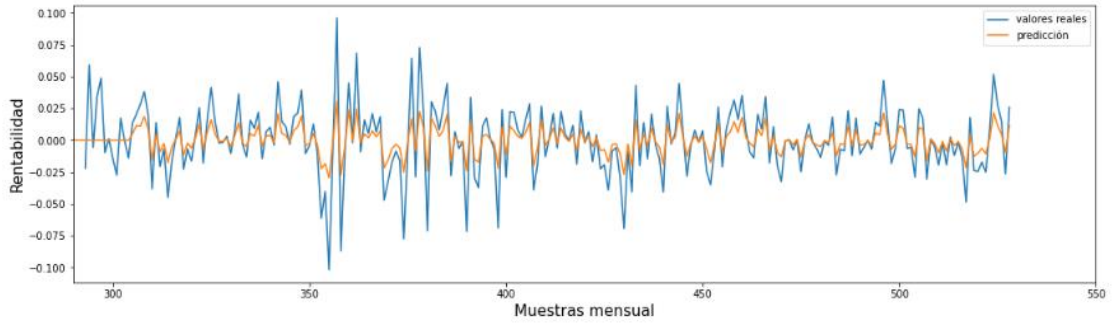


Figura Anexo 25. EUR/USD time-frame mensual con LSTM.

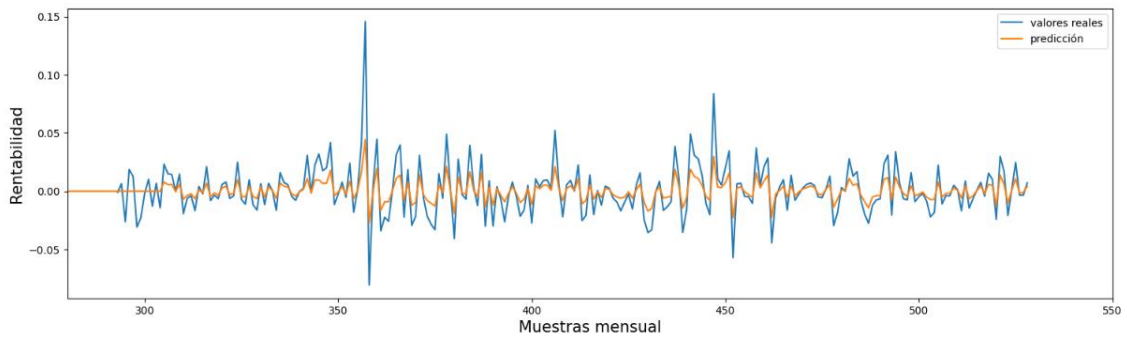


Figura Anexo 26. EUR/GBP time-frame mensual con LSTM.

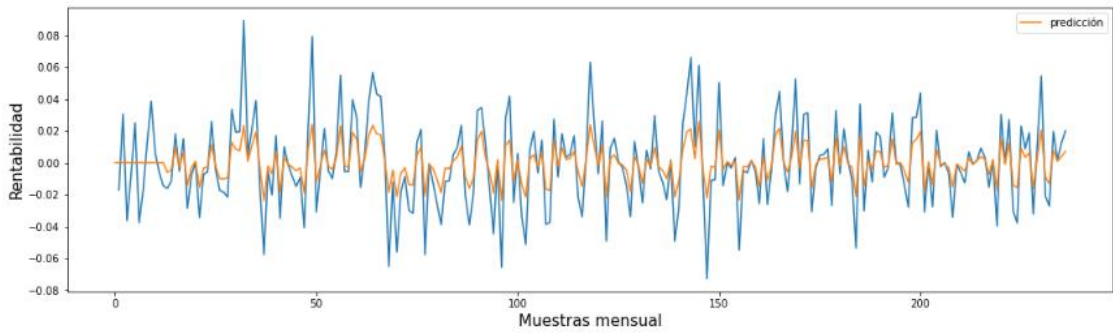


Figura Anexo 27. EUR/NZD time-frame mensual con LSTM.