

CAPÍTULO V

Aprendizaje profundo para series temporales en finanzas: aplicación al factor *momentum*

Iván Blanco*

Sergio J. García

Álvaro Remesal

Este trabajo demuestra la capacidad predictiva de las redes neuronales recurrentes *Long Short-Term Memory (LSTM)* para predecir series temporales financieras. Las arquitecturas *LSTM* son herramientas de predicción de secuencias de datos con escasa implantación hasta ahora en el ámbito académico de las finanzas. Nuestra aplicación de las redes *LSTM* al factor momentum genera predicciones con R^2 fuera de la muestra y ganancias en términos económicos que son superiores al 2 %. Los resultados demuestran el poder predictivo de los modelos *LSTM* y sirven como punto de partida para analizar de manera más exhaustiva series temporales financieras mediante estas arquitecturas.

Palabras clave: machine learning, deep learning, redes neuronales recurrentes, momentum.

* Agradecemos a Santiago Carbó, Juan José Ganuza, Daniel Peña y Pilar Poncela por la invitación para elaborar este capítulo y a los participantes de las Jornadas Funcas sobre Análisis Financiero y Big Data por sus comentarios. Álvaro Remesal ha obtenido financiación por parte del proyecto PID2021-128994NA-I00 durante la elaboración de este trabajo.

1. INTRODUCCIÓN

La tarea de predicción de la rentabilidad de los activos financieros, es decir, determinar cuándo y por qué distintos activos financieros tienen distintas rentabilidades, se remonta a los mismos orígenes del estudio de las finanzas (Bachelier, 1900; Granger y Morgenstern, 1970, Fama *et al.*, 1969). Las diferentes teorías sobre la eficiencia de los mercados financieros sugieren que cualquier tarea de predicción de rentabilidad es un ejercicio fútil. La teoría del Paseo Aleatorio implica que la rentabilidad de los activos financieros no puede predecirse en función de su rentabilidad pasada (Fama, 1970). A su vez, las teorías de equilibrio con arbitraje suponen que el rendimiento esperado de un activo financiero solamente refleja su exposición a fuentes de riesgo sistemáticas (no diversificables) (Ross, 1976).

No obstante, la literatura académica en finanzas, con una atención especial a la rentabilidad en el mercado bursátil, ha generado una multitud de estudios sobre la potencial existencia de variables que permiten predecir la rentabilidad de los activos financieros (*e.g.*, Campbell y Shiller, 1983; Fama y French, 1993; Fama y French, 2015; Harvey, Liu y Zhu, 2016; McLean y Pontiff, 2016; Hou, Xue y Zhang, 2020). En muchos casos, estos resultados sugieren la existencia de oportunidades de inversión rentables al alcance de cualquier inversor, que generan altos niveles de rentabilidad ajustada al riesgo asumido, dando lugar a multitud de hipótesis sobre sesgos conductuales para explicar las anomalías observadas¹.

En este estudio describimos un modelo general y no lineal para series temporales basado en redes neuronales profundas con el objeto de predecir la rentabilidad de una de las estrategias más estudiadas en el mundo de las finanzas: *momentum*. Esta estrategia hace referencia a una anomalía, ampliamente documentada empíricamente, que indica que los activos que han tenido mejores (peores) retornos en el pasado tienden a tener mejores (peores) retornos también en el futuro. El uso de técnicas de aprendizaje automático (*machine learning*) como las redes neuronales profundas permite, de modo natural, asimilar las dinámicas complejas y específicas que subyacen a la rentabilidad de esta estrategia de inversión. Los resultados de la estimación muestran que el modelo basado en redes neuronales profundas tiene una capacidad predictiva fuera de la muestra (*out-of-sample*) significativa en términos estadísticos y económicos. Los resultados ilustran la relevancia de dinámicas no lineales que difícilmente pueden capturarse en modelos econométricos tradicionales.

El modelo predictivo presentado en este estudio está basado en técnicas de *machine learning* aplicadas a datos en los que existe una relación secuencial entre observaciones, en este caso una relación temporal. De manera específica, el modelo predictivo utiliza una arquitectura de redes neuronales *Long Short-Term Memory (LSTM)* que trata de capturar de manera flexible las relaciones no lineales entre las rentabilidades pasadas y futuras. Además de otras ventajas computacionales, la arquitectura *LSTM* es capaz de “guardar” información remota en el tiempo que la propia red identifica como relevante a la hora de predecir la evolución temporal de una serie.

¹ Análisis teóricos en este sentido son los trabajos de Daniel *et al.* (1998), Hong y Stein (1999), Hirshleifer y Luo (2001) o Alti y Titman (2019), entre otros muchos.

Este estudio, por tanto, ofrece dos contribuciones. En primer lugar, presenta un modelo predictivo que puede capturar la predictibilidad de la serie temporal de la rentabilidad diaria del factor *momentum*. Es decir, desarrollamos un modelo lo suficientemente general para poder predecir la persistencia en la rentabilidad de los activos financieros. En segundo lugar, aplica de manera novedosa en la literatura académica de finanzas las arquitecturas *LSTM* de redes neuronales profundas para la predicción de series temporales. Las redes *LSTM*, propuestas inicialmente por Hochreiter y Schmidhuber (1997), están diseñadas para descubrir de modo flexible patrones “ocultos” en las series temporales con interacciones no lineales. Este tipo de arquitectura de predicción de datos se encuentra entre las arquitecturas con mayor éxito a nivel comercial dentro de la inteligencia artificial, siendo aplicadas para el reconocimiento automático de voz (como Siri o Alexa). No obstante, el uso de estas arquitecturas hasta ahora ha sido escaso en el ámbito académico de las finanzas.

A nivel general, los ejercicios de predicción de rentabilidades en la sección cruzada (*cross section*) se basan en aproximar la esperanza condicionada de las rentabilidades en base a una o varias características:

$$\mathbb{E} [r_{t,t+1}|x_t] = F(x_t) \quad [1]$$

donde $F(\cdot)$ es una función que transforma características, x_t , de los activos en un periodo t en una predicción de su rentabilidad en el siguiente periodo. Por su parte, $r_{t,t+1}$, representa la rentabilidad obtenida entre el instante t y el instante $t + 1$. En este ejercicio de predicción queremos entender si los activos con valores altos de ciertas características generan rentabilidades distintas frente activos con valores bajos de esas mismas características.

Las características predictivas, x_t , bien pueden ser variables que reflejan, entre otros, las distintas exposiciones a las fluctuaciones del mercado, los distintos tamaños de las empresas, o diferentes ratios de valoración que combinan precios de mercado y variables contables. A su vez, las características incluidas en x_t bien pueden ser las rentabilidades pasadas del activo, $x_t = r_{t-h,t}$, donde h puede ser un horizonte temporal arbitrario y, por tanto, $r_{t-h,t}$ representa la rentabilidad entre el instante $t-h$ y el instante t . Precisamente, desde Jegadeesh y Titman (1993), una amplia literatura ha analizado la existencia de *momentum*, o inercia, en la rentabilidad de los activos financieros en el medio plazo. Los activos que han generado una rentabilidad relativamente alta en el pasado reciente tienden a generar una rentabilidad relativamente alta en el medio plazo. Y al contrario, activos con una rentabilidad relativamente baja en el pasado reciente tienden a generar una rentabilidad relativamente baja en el medio plazo². Este tipo de predictibilidad supone uno de los mayores desafíos a las teorías de eficiencia de los mercados.

Multitud de análisis empíricos han verificado la generalidad de la anomalía *momentum* debido a su amplia presencia en múltiples activos financieros, múltiples periodos de tiempo y por su robustez en múltiples análisis estadísticos. Con mucha frecuencia, los modelos explica-

² Además, esa relación se revierte en el largo plazo. Los activos “ganadores” pierden parte de su valor en el largo plazo, al contrario que los activos “perdedores” (Jegadeesh y Titman, 1993).

tivos del rendimiento de los activos financieros suelen incorporar un factor *momentum* para tener en cuenta la aparente y mecánica persistencia de la rentabilidad, aunque siga habiendo un amplio debate académico sobre los orígenes e interpretación de esta anomalía (Fama y French, 2015; Cochrane, 2007).

La existencia de *momentum* da lugar a estrategias de inversión, en principio rentables, basadas en la explotación de la persistencia en la rentabilidad. Es decir, supongamos que en cada momento de tiempo construimos una cartera que adquiere el 10 % de las acciones con mayor rentabilidad en un horizonte temporal pasado h (esta cartera la denominamos como $d1$), $r_{t-h,t}^{d1}$, y vende en corto el 10 % de las acciones con menor rentabilidad en ese mismo horizonte (esta cartera la denominamos como $d10$), $r_{t-h,t}^{d10}$, donde h puede ser, por ejemplo, 12 meses. La literatura empírica ha mostrado que la rentabilidad de esta cartera, $r_{t,t+s}^{Mom}$, es sustancialmente positiva para periodos de inversión, s , de entre 6 y 18 meses.

La media obtenida en los estudios empíricos de la rentabilidad de la estrategia *momentum*, representada por la serie temporal $r_{t,t+s}^{Mom}$, es significativamente alta. No obstante, debido a que la cartera *momentum* tiene una composición que varía en el tiempo, las dinámicas de esa persistencia tienen características particulares. En primer lugar, la serie temporal es predecible en función de su pasado (Huang, 2022). En particular, la autocorrelación de $r_{t,t+1}^{Mom}$ es negativa. En segundo lugar, la trayectoria pasada de la serie es importante. Por ejemplo, en la sección cruzada, las series con menos cambios repentinos tienden a mostrar mayor predictibilidad (Da *et al.*, 2014). Además, la persistencia en la rentabilidad es mayor en aquellos activos que acumulan mayor rentabilidad en el periodo más reciente (Blanco *et al.*, 2023).

Por último, la tercera característica propia de la anomalía *momentum* es que la estrategia está sujeta a rentabilidades negativas extremas (*crashes*) (Barroso y Santa Clara, 2015; Daniel y Moskowitz, 2016). Con relativa frecuencia hay periodos en los que la cartera de activos “ganadores” muestra un rendimiento marcadamente inferior a la cartera de activos “perdedores”. Esto suele ocurrir cuando la cartera de “perdedores” de manera súbita experimenta una reversión en su rentabilidad. La existencia de estos *crashes* reduce el atractivo de la estrategia *momentum* y puede explicar la existencia de una rentabilidad extraordinaria, o “alfa”. No obstante, estrategias *momentum* que ajustan la inversión por el nivel pasado de volatilidad de los componentes de la cartera tienden a salir indemnes de estos periodos con rentabilidades negativas extremas.

Este tipo de características específicas sugieren la existencia de no-linealidades relevantes en la dinámicas de *momentum*. No obstante, cada una de estas características se ha estudiado de manera separada. La cuestión es, por tanto, si podemos obtener un modelo predictivo que, de manera flexible, sea capaz de incorporar todos estos aspectos y permita obtener una predicción de la rentabilidad de la estrategia *momentum*. Concretamente, queremos saber si es viable predecir el grado de persistencia en la rentabilidad de los activos financieros o, en otras palabras, si podemos predecir hasta qué punto la rentabilidad pasada de la cartera *momentum* supone una señal informativa de su rentabilidad futura.

En este estudio, mediante la aplicación de arquitecturas *LSTM*, hacemos frente a la tarea de predecir la serie temporal de rentabilidad diaria de *momentum* capturando su complejo comportamiento. De un modo general, nuestra tarea de predicción se resume en la estimación de una función H tal que:

$$r_{t,t+s}^{Mom} = H(x_{t-1}, \dots, x_{t-K}) + \varepsilon_t \quad [2]$$

Donde cada uno de los argumentos $\{x_{t-j}\}_{j=1}^K$ es un vector con información sobre la rentabilidad de los activos que conforman la cartera *momentum* en diferentes horizontes pasados y ε_t es un término de error.

Los resultados de nuestro análisis muestran una capacidad predictiva de los modelos *LSTM* superiores al 0,5 % en términos de R^2 , que también implican ganancias de utilidad económicamente relevantes del uso del modelo propuesto respecto a una predicción solamente basada en la media histórica (Campbell y Thompson, 2008). De entre todos los modelos explorados, el mayor poder predictivo está generado por el modelo que solamente utiliza información sobre las series de rentabilidades de las carteras *momentum* a un día y a 120 días.

Las ganancias de predicción son significativas en términos estadísticos y también en términos económicos. Un inversor con preferencias media-varianza estaría dispuesto a pagar alrededor de un 2 % de su riqueza, anualmente, por acceder a una estrategia basada en el modelo que mejor predice la serie temporal del factor *momentum*, en comparación con una estrategia que utiliza la media histórica para predecir la rentabilidad. Esto es, la predicción con arquitectura *LSTM* es capaz de capturar de manera notable las no-linealidades intrínsecas al factor *momentum* sin que esto implique mayor riesgo.

Todos los diferentes modelos analizados generan predicciones que, en muchos periodos, no muestran gran variación respecto a la rentabilidad observada del factor *momentum*. No obstante, los modelos tienden a predecir con mucha precisión los periodos en los que se producen rentabilidades negativas extremas en el factor *momentum*, que suelen estar marcados por episodios de alta volatilidad. Por tanto, observamos empíricamente cómo los modelos *LSTM* encuentran estrategias de inversión que son capaces de anticipar las rentabilidades negativas extremas, *crashes*, y generar ratios de Sharpe³ del orden de 0,5.

La investigación en finanzas cada vez hace un mayor uso de técnicas de aprendizaje automático para analizar bases de datos masivas (*big data*). Uno de los desafíos fundamentales de los ejercicios de predicción en series temporales de rentabilidad es la proporción excesiva de ruido³ que contienen estas series temporales en relación a la información relevante sobre la rentabilidad esperada de un activo. Esto es, los precios de los activos financieros incorporan información y se mueven por múltiples motivos que no están relacionados con su verdadera rentabilidad esperada. Por tanto, la tarea fundamental para cualquier modelo de

³ La ratio de Sharpe se define como la rentabilidad en exceso sobre el activo libre de riesgo entre la volatilidad de la rentabilidad de la estrategia.

predicción de rentabilidad, y en el caso particular de este estudio basado en la arquitectura *LSTM*, consiste en aglutinar de manera eficiente la información relevante para predecir la rentabilidad esperada del factor *momentum*.

La literatura académica, hasta el momento, se ha ocupado principalmente de explotar herramientas de aprendizaje automático como método de selección de predictores de la rentabilidad entre una amplia variedad de candidatos. Rapach *et al.* (2013) representa el primer análisis que introduce herramientas específicas de *machine learning*, en este caso una estimación LASSO, a la predicción de rentabilidades financieras. Chincó *et al.* (2019) también utilizan la estimación LASSO para la predicción de rentabilidades minuto a minuto. Gu *et al.* (2018) emplean una lista exhaustiva de herramientas de aprendizaje automático para analizar la predictibilidad de la rentabilidad de las acciones. Kozak *et al.* (2020) también explotan la metodología LASSO para, desde un punto de vista bayesiano, seleccionar entre múltiples características aquellas que determinan el factor estocástico de descuento. Han *et al.* (2019) proponen procedimientos para predecir las rentabilidades en la sección cruzada entre más de cien características⁴.

Los estudios anteriores se centran en la predictibilidad en la sección cruzada, esto es, en las diferencias entre activos que predicen las diferencias en rentabilidades. Otra línea de investigación parte de la literatura que explora la predictibilidad de la rentabilidad del mercado en su conjunto a través de una serie de indicadores económicos y financieros como ratios de valoración, tipos de interés, o inflación⁵. Moskowitz *et al.* (2012) muestran que carteras formadas por activos con rentabilidad pasada positiva generan rentabilidades extraordinarias, aunque la significatividad de estos resultados es disputada por Huang *et al.* (2020). Rapach y Zhou (2019) aplican el método de selección de Han *et al.* (2019) para predecir la rentabilidad del mercado. Fama y French (2020) sugieren utilizar los coeficientes sobre los factores sistemáticos estimados en la sección cruzada como factores en la serie temporal para ajustar los modelos de valoración. Engelberg *et al.* (2021) muestran que las características medias de los factores en la sección cruzada no pueden predecir la rentabilidad de mercado.

El artículo de Dong *et al.* (2021) supone el trabajo más parecido al que se presenta en este estudio. En su trabajo, Dong *et al.* (2021) emplean metodologías de aprendizaje automático para determinar la predictibilidad de la rentabilidad de mercado usando las rentabilidades pasadas de las carteras que explican la rentabilidad de los activos en la sección cruzada (factores). En el caso de nuestro trabajo, también realizamos una tarea de predicción, en este caso para la rentabilidad de una anomalía en concreto, *momentum*, utilizando exclusivamente información pasada sobre esa misma anomalía.

⁴ Otros trabajos que explotan *machine learning* para predecir la rentabilidad en la sección cruzada con múltiples predictores son Chincó, Clark-Joseph y Ye (2019), Freyberger, Neuhierl y Weber (2020), Avramov, Cheng y Metzker (2021), Cong *et al.* (2021), Liu, Zhou y Zhu (2021), Gu *et al.* (2021).

⁵ Trabajos clásicos en este sentido son, por ejemplo, Nelson (1976); Campbell (1987); Fama y French (1988, 1989); Pástor y Stambaugh (2003, 2009).

Las arquitecturas *LSTM* no se han explotado por ahora con frecuencia en la literatura académica. Como excepción, Chen *et al.* (2021), utilizan una arquitectura *LSTM* para extraer información de distintas variables macroeconómicas y para estimar el factor estocástico de descuento. Nuestra aplicación se basa en la predicción de la rentabilidad de *momentum* utilizando información pasada sobre la rentabilidad de sus componentes sobre distintos horizontes. De este modo, nuestro trabajo conecta con la literatura sobre modelos autorregresivos heterogéneos (Hizmeri *et al.*, 2022; Corsi, 2009; Bollerslev *et al.*, 2016). Nuestro modelo tiene en cuenta la relevancia de distintos horizontes predictivos, pero de una manera no lineal y flexible.

El resto de este trabajo se organiza como sigue. En la sección 2 describimos la metodología de investigación. En particular, describimos detalladamente los aspectos más relevantes de las redes neuronales recurrentes y su versión *LSTM*. En la sección 3 presentamos los resultados de predicción para el factor *momentum*. La sección 4 presenta las conclusiones.

2. METODOLOGÍA

2.1. Datos

En este estudio analizamos la serie temporal del factor *momentum* generado por las acciones de empresas cotizadas en Estados Unidos (NYSE, Nasdaq y AMEX). La fuente de datos es la página web del profesor Kenneth French. Para construir ese factor, en primer lugar, cada día se ordenan las acciones de mayor a menor rentabilidad en los doce meses anteriores, excluyendo el mes más reciente⁶. En el modelo de predicción utilizamos la información sobre la series temporales de rentabilidad diaria generadas por el 10 % de acciones con mayor rentabilidad⁷, r_t^{d1} , y por el 10 % de acciones con menor rentabilidad, r_t^{d10} .

Por su parte, la serie de rentabilidad diaria del factor *momentum*, r_t^{mom} , proviene de la diferencia en rentabilidades diarias entre el primer y el tercer tercil de la distribución de rentabilidad en los doce meses anteriores. El retorno de cada tercil se calcula utilizando la media de la rentabilidad entre empresas con capitalización bursátil mayor a la mediana y empresas con capitalización bursátil inferior a la mediana (datos obtenidos de la página web del profesor Kenneth French).

Para estimar el modelo de predicción transformaremos las rentabilidades diarias para construir rentabilidades sobre distintos horizontes. De manera formal, definimos la rentabilidad acumulada entre el periodo t y $t + s$ por la cartera p como:

⁶ La exclusión del mes más reciente se debe a la existencia de reversión en el corto plazo debido a factores de microestructura del mercado que pueden contaminar la selección en base a rentabilidades pasadas (Jegadeesh, 1990; Lehman, 1990; Jegadeesh y Titman, 1993).

⁷ La composición de $d1$ y $d10$ se actualiza de manera diaria, ver detalles en la página web del profesor Kenneth French.

$$r_{t,t+s}^p = \prod_{i=0}^s (1 + r_{t+i}^p) - 1$$

2.2. Redes neuronales recurrentes

Los modelos de predicción basados en redes neuronales representan uno de los núcleos fundamentales de los métodos de aprendizaje automático o *machine learning*. En diversos campos, desde la genética, hasta el reconocimiento del lenguaje, las redes neuronales han demostrado ser una herramienta versátil, eficiente y escalable para realizar ejercicios de predicción. Pese a que el concepto de “red neuronal artificial” fuera desarrollado en los años 60, las redes neuronales se han popularizado en las últimas dos décadas⁸. Esto se debe a: 1) la existencia y acumulación de una cantidad ingente de bases de datos masivas en las que las redes neuronales suelen rendir de manera notable respecto a otros métodos de *machine learning* y 2) el desarrollo de las tecnologías de computación que ha permitido manejar cantidades ingentes de datos e implementar algoritmos de optimización con miles (o millones) de parámetros.

Dentro de las redes neuronales, existe un método especializado en el procesamiento de secuencias de datos, ya sean series temporales, audio, notas musicales o texto, para desarrollar modelos predictivos. A este tipo de arquitectura se le denomina red neuronal recurrente (RNN). Las RNN son modelos de aprendizaje profundo que intentan capturar las dinámicas de secuencias a través de conexiones recurrentes. Las RNN consisten en una versión especial de red neuronal que se aplica a cada elemento de la secuencia, manteniendo siempre los mismos parámetros, y que aprende de la información pasada de la serie.

En nuestra aplicación, el objetivo consiste en predecir la rentabilidad del factor *momentum* a 25 días, $r_{t,t+25}^{mom}$ basándonos en información pasada sobre la rentabilidad de las carteras r_t^{d1} y r_t^{d10} en distintos horizontes. Utilizamos estos *inputs* para el modelo debido a que la diferencia entre r_t^{d1} y r_t^{d10} tiene capacidad predictiva sobre la rentabilidad del factor *momentum* (Huang, 2021). Asumamos que en el modelo de predicción incluimos dos horizontes: la rentabilidad en el día actual y la rentabilidad en los 60 días anteriores. Por tanto, en el momento s , el vector de *inputs* que para predecir $r_{t,t+25}^{mom}$ viene dado por:

$$x_s = \left\{ \left(r_s^{d1}, r_{s-60,s}^{d1} \right), \left(r_s^{d10}, r_{s-60,s}^{d10} \right) \right\}$$

donde cada *input* tiene una dimensión 1×4 . La estructura más simple de una red neuronal recurrente aparece en la figura 1. Para generar una predicción $\hat{r}_{t,t+25}^{mom}$, el modelo se nutre de información sobre una secuencia de inputs $\{x_{t-j}\}_{j=1}^K$, donde el parámetro K indica el número

⁸ El nombre de red neuronal se originó bajo la hipótesis de que las neuronas animales funcionan de manera parecida a este tipo de arquitectura de computación. No obstante, la investigación en biología muestra que esta representación es una mera aproximación metafórica del funcionamiento real de las neuronas animales.

de retardos que utilizamos para predecir el *output* en el momento t . Cada *input* x_s se introduce por separado en una “celda de activación” A cuyos parámetros son independientes del momento de tiempo. La celda de activación genera un “estado” h_s combinando el *input* x_s y la variable de “estado” generada en el elemento anterior de la serie:

$$h_s = \phi(x_s w_{xh} + h_{s-1} w_h + b_h)$$

donde $\phi(\cdot)$ es una función no lineal, “función de activación”, y se asume que $h_{\{t-K-1\}} = 0$. w_x , w_h y b_h son parámetros, o “pesos”, a entrenar. En particular, w_x es un vector de pesos con dimensión igual a la dimensión de x_s , mientras que los otros dos elementos, en este modelo más sencillo, son escalares.

El nombre de red neuronal recurrente proviene precisamente del hecho que la activación en cada elemento de la secuencia depende de la información específica de ese periodo e incluye información acumulada en el periodo anterior. El modelo incluye una especie de “memoria” en h_{t-j} , ya que depende de la información aprendida en elementos anteriores de la secuencia. En el modelo RNN más sencillo, la predicción $\hat{r}_{t,t+25}^{mom}$ para una secuencia de *inputs* vendrá dada por otra transformación no lineal del *input* final y el estado en h_{t-1} :

$$\hat{r}_{t,t+25}^{mom} = f(x_t w_{xy} + h_{t-1} w_{hy} + b_y)$$

donde $f(\cdot)$ es una función lineal o no lineal, mientras que w_{xy} , w_{hy} y b_y son parámetros a estimar. Este proceso de propagación en el tiempo finalmente genera un valor de la función de pérdida que compara el rendimiento real con el generado por el modelo⁹. Por ejemplo, si la función de pérdida viene dada por el error cuadrático medio, tendríamos que dentro de un subconjunto de observaciones \mathcal{T} (*mini-batch*):

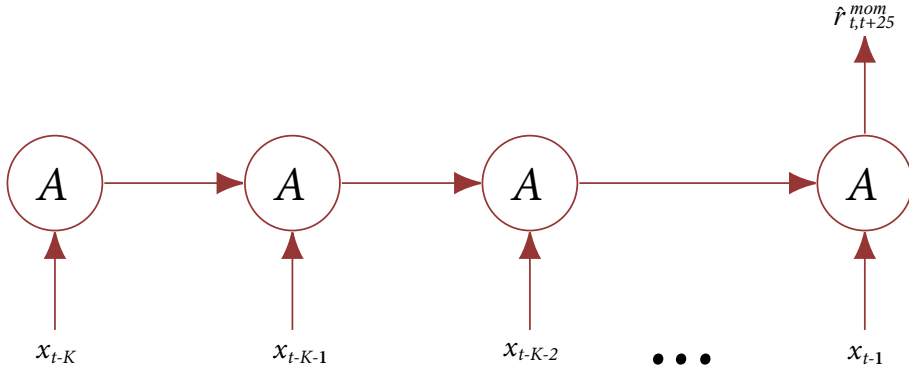
$$\mathcal{L}_{\mathcal{T}} = \frac{1}{|\mathcal{T}|} \sum_{s \in \mathcal{T}} (r_{s,s+25}^{mom} - \hat{r}_{s,s+25}^{mom})^2$$

donde $|\mathcal{T}|$ indica la cardinalidad del subconjunto.

Como analogía, podemos entender una RNN, y la arquitectura *LSTM* que describimos más adelante, como una versión especial de un modelo autorregresivo con heterogeneidad, donde la realización de la serie temporal en el futuro depende de la realización pasada de la serie vista desde diversos horizontes temporales. La capacidad adicional de predicción surge de la no-linealidad de las activaciones entre los distintos elementos de la secuencia.

⁹ Las arquitecturas RNN también permiten una diversidad de diseños de *inputs* y *outputs* que obviamos en la descripción ya que no los aplicamos en nuestro ejercicio de predicción. Las arquitecturas RNN pueden ser, entre otras, secuencia-a-secuencia, donde queremos predecir el *output* en todos los elementos, o vector-a-secuencia, donde utilizamos un único *input* para predecir una secuencia. Ver, por ejemplo, el capítulo 14 de Geron (2019) para una descripción más pormenorizada.

Figura 1.

Estructura de una red neuronal recurrente unicapa

El esquema anterior puede generalizarse incrementando el número de “neuronas” por celda. En tal caso nos encontraríamos con un vector de estados h_s en cada momento del tiempo, en vez de un escalar, e incrementaríamos la dimensión de los pesos a entrenar con el modelo. Lo anterior también puede combinarse con una estructura jerárquica, una “red profunda”, con distintas capas donde el estado de cada capa depende no sólo del *input* en el momento actual, x_s , y de los estados anteriores h_{s-1} , sino también de los estados obtenidos en las capas “inferiores” en el mismo elemento s . Si denominamos como $h_s^{(l)}$ el estado de la secuencia en el elemento s en la capa l la recurrencia de la red neuronal se puede expresar como:

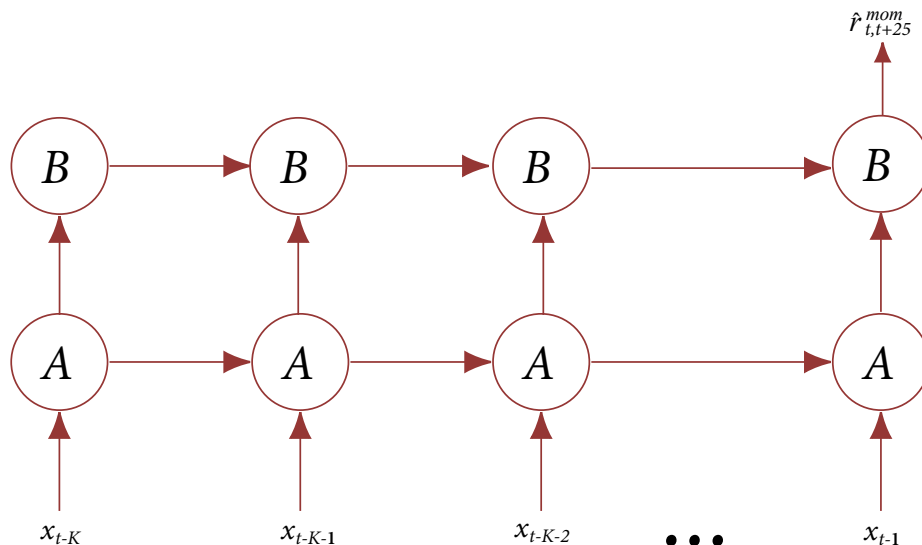
$$h_s^{(l)} = \phi \left(h_s^{(l-1)} w_{h0}^{(l)} + h_{s-1}^{(l)} w_{h1}^{(l)} + b_h^{(l)} \right)$$

con la normalización $x_s = h_s^0$. La **figura 2** ilustra este tipo de arquitectura con dos capas. Como en la arquitectura básica anterior, la celda A se nutre de los *inputs* x_s y de los estados de la celda A en el elemento anterior de la secuencia. Por su parte, la celda B se nutre de los estados generados por la celda A y por los estados de la celda B en el elemento anterior de la secuencia.

El entrenamiento de las RNN usa la misma lógica de propagación inversa que se usa en las redes neuronales artificiales. No obstante, en el caso de las RNN, en cada elemento de la secuencia se mantienen los mismos pesos de las diferentes capas. La dinámica de entrenamiento de una RNN puede resumirse en tres pasos:

- Dados unos pesos, evaluar el error de predicción de la red en el subconjunto \mathcal{T} mediante la evolución secuencial, en orden cronológico, de los estados e *inputs*.
- Calcular los gradientes de la función de pérdida con respecto a cada uno de los pesos a entrenar sin tener en cuenta su repetición en cada elemento, es decir, como si la red tuviera pesos distintos en cada elemento.

Figura 2.

Estructura de una red neuronal recurrente con 2 capas

- Agregar los gradientes calculados en el paso anterior para cada uno de los pesos que se comparten en cada elemento de la secuencia. Actualizar los pesos usando el criterio de descenso de gradiente y volver al paso inicial hasta cumplir el criterio de convergencia.

Las arquitecturas anteriores tienen una estructura común en la que el estado h_s solamente captura información de los elementos anteriores en la secuencia. Otros modelos de redes recurrentes permiten que el estado también incorpore información de los elementos posteriores. Por ejemplo, en una tarea de reconocimiento de escritura puede ser necesario y más efectivo utilizar información posterior para entender mejor el contexto y reducir ambigüedades. Este principio también es aplicable a las series temporales en finanzas donde queremos extraer las señales más informativas para poder predecir la evolución futura de los precios de activos financieros¹⁰.

¹⁰ El funcionamiento de las redes RNN es análogo a los modelos de extracción de señales en series temporales con procesos de Markov ocultos. El capítulo 22 de Hamilton (1994) presenta una introducción a estos métodos. La estimación de estos modelos con procesos ocultos conlleva la estimación de las probabilidades de los estados no observables. Para ello, es útil explotar información sobre el pasado e información sobre el futuro de la serie. Esto es, conocer el estado oculto en el futuro es información útil para estimar el estado oculto en el momento actual, del mismo modo que es útil conocer el estado oculto en el pasado. La diferencia entre estos métodos es que las redes RNN intentan recoger los factores no observables que determinan la serie temporal teniendo en cuenta interacciones no lineales y sin realizar hipótesis sobre la distribución de las variables aleatorias.

Las redes neuronales recurrentes que permiten el entrenamiento hacia delante y hacia atrás en el tiempo se denominan redes bidireccionales. La **figura 3** ilustra el funcionamiento de este tipo de red. Las redes bidireccionales incorporan celdas como la celda *B* en el gráfico, que genera un estado nuevo utilizando información del futuro. De manera formal, si g_s representa el estado generado en dirección cronológicamente inversa, cada nuevo estado se nutre del *input* en el momento s y del estado g en el momento posterior:

$$g_s = \phi(x_s w_{xg} + g_{s+1} w_{gg} + b_g)$$

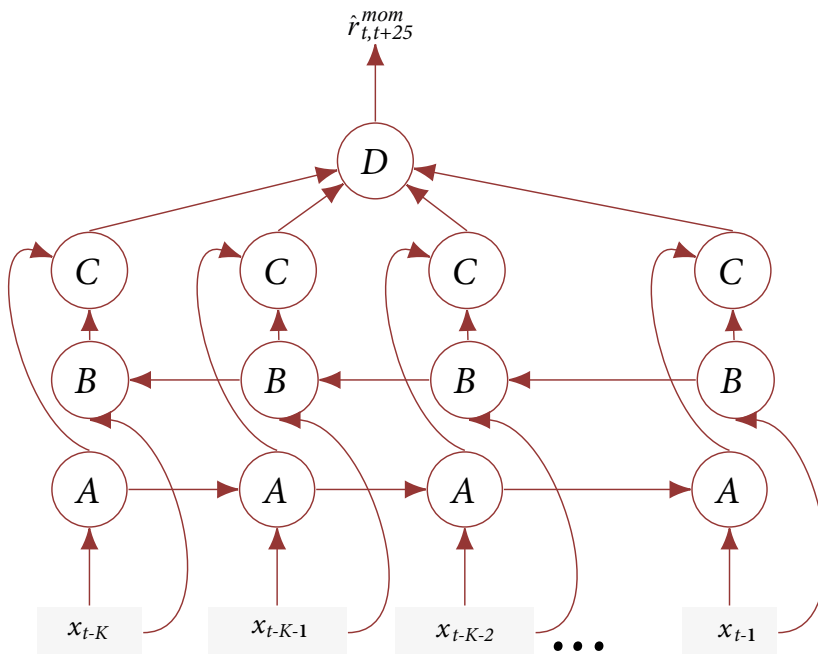
Donde asumimos que $g_{-t} = 0$. Por su parte, la celda *C* permite combinar el aprendizaje generado de manera independiente por las celdas *A* y *B* siguiendo distintas direcciones. Para generar una predicción final, los estados $\{(h_s, g_s)\}_{s=0}^t$ pueden combinarse en una capa adicional, *D*, al estilo de una red neuronal artificial tradicional (perceptrón multicapa o MLP) para generar una predicción final del *output*, $\hat{r}_{t,t+25}^{mom}$:

$$\hat{r}_{t,t+25}^{mom} = f(h w_{hy} + g w_{gy} + b_y)$$

donde h y g son vectores de dimensión $1 \times (h \times K)$ cada uno que resultan de la concatenación de los estados.

Figura 3.

Estructura de una red neuronal bidireccional



La dinámica de entrenamiento de una RNN bidireccional consiste en los siguientes pasos:

1. Dados unos pesos, evaluar los estados h_s y g_s de modo separado en el subconjunto \mathcal{T} .
2. Evaluar el error de predicción de la red dados los estados h_s y g_s junto a los pesos de la capa final de predicción.
3. Calcular los gradientes de la función de pérdida con respecto a los parámetros de la capa final de predicción.
4. Calcular los gradientes de la función de pérdida con respecto a los parámetros de las capas de propagación temporal.
5. Agregar los gradientes calculados en el paso anterior para todos y cada uno de los pesos que se comparten en cada elemento de la secuencia. Actualizar todos los pesos usando el criterio de descenso de gradiente y volver al paso inicial hasta cumplir el criterio de convergencia.

En principio, el diseño de las arquitecturas RNN puede resultar muy adecuado y eficiente para manejar información en forma de secuencias. No obstante, las RNN tienen dos graves problemas en la práctica. En primer lugar, el entrenamiento de la red requiere una propagación inversa, hacia atrás en el tiempo, que debe realizarse a través de los K niveles de cada secuencia que se introduzca en el modelo. Esta propagación hacia atrás en el tiempo genera una cadena de productos de matrices que fácilmente genera inestabilidades numéricas cuando se ejecuta el descenso de gradiente (explosión o desaparición del gradiente). Como consecuencia de lo anterior, los modelos RNN conllevan un entrenamiento con una gran demanda en términos computacionales.

En segundo lugar, los estados generados por los modelos RNN no permiten tener en cuenta que algunos elementos alejados de la secuencia puedan tener una relevancia considerable en el *output* actual. La estructura de las RNN hace que los elementos alejados pierdan relevancia de manera exponencial. También puede haber información reciente que no aporte poder predictivo y que sea mejor “olvidar”. Incluso podemos esperar algún tipo de cambio estructural en la serie que pueda afectar a la capacidad predictiva del modelo, de modo que sea más adecuado “resetear” el modelo. Las arquitecturas que utilizamos para desarrollar nuestro modelo predictivo de *momentum* nos permiten, o al menos aspiramos a ello, recoger este tipo de comportamientos en las series temporales, permitiendo a su vez menores costes computacionales.

2.3. Redes neuronales recurrentes con *LSTM*

El diseño de las arquitecturas *LSTM* afronta el desafío de preservar la información de largo plazo relevante y de ignorar la información de corto plazo irrelevante (Hochreiter y Schmidhuber, 1997). El modelo *LSTM* consiste en el diseño de una celda que absorbe el *input*

actual y el estado anterior y selecciona la información a absorber. Para ello, en el interior de la celda *LSTM* se realizan ciertas operaciones a través de las denominadas “puertas lógicas” (*gates*). El entrenamiento de los parámetros incluidos en esas puertas lógicas permite a la arquitectura filtrar qué información nueva aprender en el estado y qué información ignorar.

En cada elemento de la secuencia la celda *LSTM* se nutre del input x_t , con dimensión $1 \times d$, y del estado en el elemento anterior h_{t-1} , con dimensión $1 \times h$. Esta información es procesada de manera independiente por tres distintas puertas lógicas: *Input*, *Forget* y *Output*. Las tres puertas lógicas tienen valores en $(0, 1)$ al tener asociados una función de activación sigmoidal, $\sigma(z) = (1 + \exp(-z))^{-1}$. Formalmente, las tres puertas lógicas se calculan como:

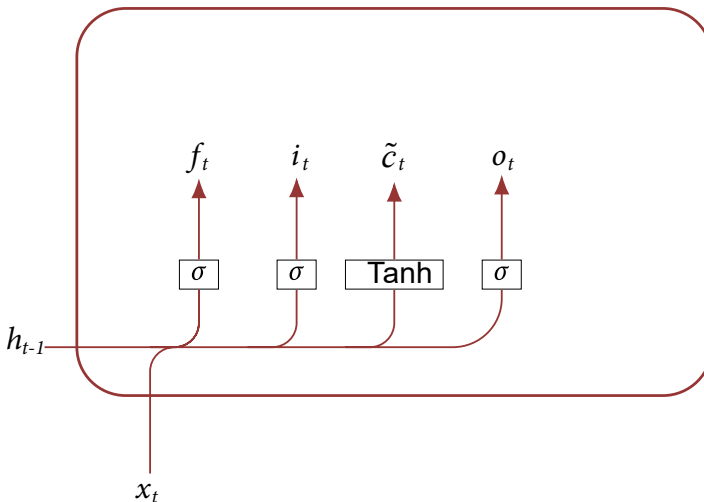
$$\begin{aligned} i_t &= \sigma(x_t W_{xi} + h_{t-1} W_{hi} + b_i), \\ f_t &= \sigma(x_t W_{fi} + h_{t-1} W_{fi} + b_f), \\ o_t &= \sigma(x_t W_{oi} + h_{t-1} W_{ho} + b_o) \end{aligned}$$

donde W_{xi} , W_{fi} , $W_{oi} \in \mathbb{R}^{d \times h}$ y W_{hb} , W_{hb} , $W_{hi} \in \mathbb{R}^{h \times h}$ y b_i , b_f , $b_o \in \mathbb{R}^{1 \times h}$ son parámetros a entrenar. Además, en este paso inicial se determina una “memoria candidata”, \tilde{c}_t , que podrá incorporarse a la memoria a largo plazo según el resultado de la puerta lógica *Input*. \tilde{c}_t , con dimensión $1 \times h$, se calcula a partir de x_t y h_{t-1} y viene dada por una función de activación tangente hiperbólica, $\tanh(z) = (\exp(z) - \exp(-z)) / (\exp(z) + \exp(-z))$, que genera valores restringidos en el rango $(-1, 1)$:

$$\tilde{c}_t = \tanh(x_t W_{xc} + h_{t-1} W_{hc} + b_c)$$

Figura 4.

Estructura de puertas lógicas y memoria candidata



donde $W_{xc} \in \mathbb{R}^{d \times h}$ y $W_{hc} \in \mathbb{R}^{h \times h}$ y $b_c \in \mathbb{R}^{1 \times h}$ son pesos a entrenar. El proceso de cálculo de las tres puertas lógicas y la memoria candidata está ilustrado en la **figura 4**.

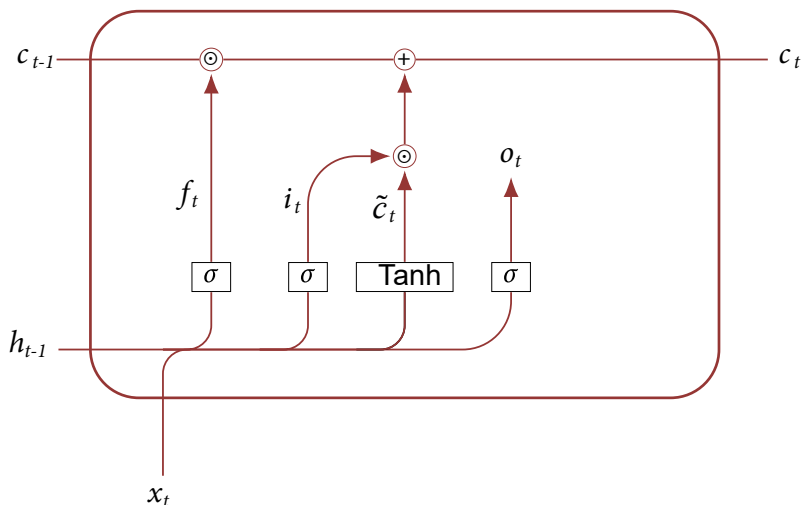
La celda *LSTM* tiene dos puertas lógicas dedicadas a determinar el grado de incorporación y “olvido” de la memoria a largo plazo. La puerta lógica *Input* determina el grado de incorporación de información creada en \tilde{c}_t . La puerta lógica *Forget* determina el grado de memoria heredada del elemento anterior, $c_{t-1} \in \mathbb{R}^h$, que se mantiene. La memoria a largo plazo se actualiza mediante la siguiente ecuación:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

donde el operador \odot representa la multiplicación de matrices elemento por elemento. f_t e i_t se encuentran en el intervalo $(0, 1)$ y determinan, respectivamente, el grado eliminación de memoria pasada y el grado de incorporación de información presente a la memoria actual. El diseño de las funciones de activación permite aliviar el problema de explosión/desaparición del gradiente, mientras que el mantenimiento de la memoria c_t permite capturar la relevancia de las dependencias de largo plazo entre los elementos de la serie.

Figura 5.

Estructura de celda LSTM



El último paso consiste en definir el estado h_t originado en la celda *LSTM*:

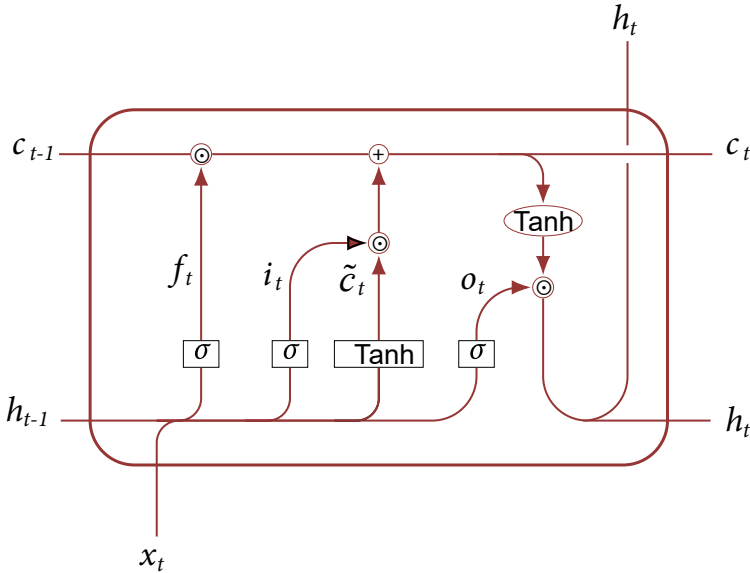
$$h_t = o_t \odot \tanh(c_t)$$

Si la puerta lógica *Open*, o_t , se aproxima a 1 el modelo permite acceder a toda la memoria a largo plazo hacia la predicción del estado. En caso contrario, cuando o_t se aproxima a 0,

se retiene la información dentro de la celda y no se pasa al siguiente elemento. La estructura final de una celda *LSTM* está ilustrada en la figura 5. El estado h_t también puede utilizarse como *input* para la predicción o como *input* para una capa superior de otra celda *LSTM*.

Figura 6.

Estructura de celda *LSTM* y *output* de estado



La arquitectura *LSTM*, pese a su aparente complejidad, permite un entrenamiento “más fluido” con respecto a las RNN básicas. Esto se debe a que en una RNN básica la cantidad de información absorbida en cada periodo de tiempo a través de h_{t-1} es independiente del elemento de la secuencia en el que nos encontremos. Mientras tanto, en una *LSTM* las puertas lógicas f_t y i_t permiten que la información absorbida sea específica en cada elemento de la secuencia. Es más, al ignorarse cierta información, el modelo puede filtrar información de elementos lejanos que sea relevante para la predicción.

2.4. Diseño de la arquitectura de aprendizaje

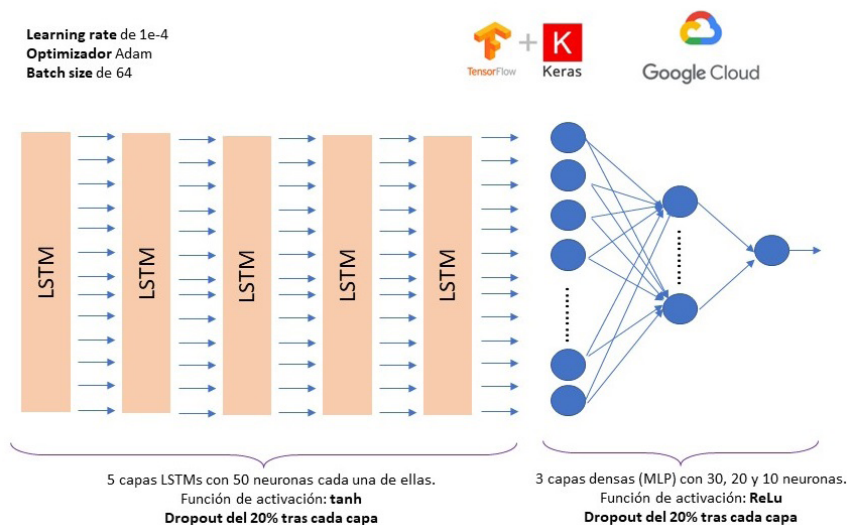
Para la estimación del modelo se ha recurrido a una arquitectura compuesta por 5 capas de redes *LSTM* (bidireccionales) con 50 neuronas cada una de ellas que se completan, tras cada una de ellas, con un capa de dilución (*dropout*) del 20 %. El *dropout* es un técnica de regularización que pretende mitigar posibles problemas de sobreajuste en la arquitectura. Por último, tras la salida de la última capa *LSTM*, la arquitectura incluye tres capas perceptrón multicapa (“MLP” o capas densas) de 30, 20 y 10 neuronas, respectivamente. La función de

activación utilizada es la tangente hiperbólica (“tanh”) para las capas LSTM y la función rectificadora (“ReLU”) para las capas MLP. La figura 7 ilustra la arquitectura y los principales hiperparámetros utilizados.

El entrenamiento y desarrollos de la arquitectura ha sido realizado con Tensorflow-Keras y la arquitectura ha sido entrenada en Google Cloud mediante el uso de GPU (“Graphics Processing Unit”) por sus ventajas computacionales.

Figura 7

Estructura de la arquitectura y principales hiperparámetros



Como describimos anteriormente, para entrenar y evaluar el modelo utilizamos datos diarios del factor *momentum* desde el año 1926 hasta la actualidad, año 2022. Para obtener las predicciones $\hat{r}_{t,t+25}^{mom}$ utilizamos una secuencia de *inputs* $\{x_{t-j}\}_{j=1}^K$ con $K = 60$. Del total de observaciones, se han utilizado para el conjunto de entrenamiento $\mathcal{T}_1 = 12.410$ observaciones, para el conjunto de validación¹¹ $\mathcal{T}_2 = 3.013$ observaciones, y se han reservado $\mathcal{T}_3 = 9.284$ observaciones para el conjunto de test. De esta forma, considerando la secuencia temporal, el periodo de entrenamiento y validación iría desde el año 1926 hasta el año 1983, reservando desde 1984 hasta el año 2022 para el test.

2.5. Medición de la capacidad predictiva y valor económico

Para medir la capacidad predictiva del modelo desde un punto de vista estadístico, utilizamos como referencia un modelo de predicción que utiliza la media histórica de la rentabi-

¹¹ El conjunto de validación se ha utilizado exclusivamente para la detención temprana, *early stopping*, del entrenamiento.

lidad $r_{t,t+25}^{mom}$ en el conjunto de las muestras de entrenamiento y validación, \bar{r}^{mom} . Los errores de predicción del modelo de referencia y del modelo LSTM vendrán dados por:

$$\hat{e}_{M,t} = r_{t,t+25}^{mom} - \bar{r}^{mom}$$

$$\hat{e}_{LS,t} = r_{t,t+25}^{mom} - \hat{r}_{t,t+25}^{mom}$$

Por tanto, el error cuadrático medio de las predicciones de cada modelo se puede expresar como:

$$\widehat{ECM}_j = \frac{1}{|T_3|} \sum_{t \in T_3} \hat{e}_{j,t}^2 \text{ for } j \in \{M, LS\}$$

donde $|T_3|$ indica el número de observaciones en la muestra test. Una vez calculados los errores de predicción de ambos modelos podemos construir el estadístico R_{OOS}^2 de Campbell y Thompson (2008):

$$R_{OOS}^2 = 1 - \frac{\widehat{ECM}_{LS}}{\widehat{ECM}_M}$$

El componente predecible de las rentabilidades es muy escaso, pero Campbell y Thompson (2008) sugieren que un R_{OOS}^2 superior a un 0,5 % implican ganancias de utilidad económicamente relevantes. Además, para comprobar la significatividad estadística de la estimación del R_{OOS}^2 , realizamos el contraste de hipótesis de Clark y West (2007) que puede implementarse a través del siguiente modelo de regresión:

$$\hat{e}_{M,t}^2 - \hat{e}_{LS,t}^2 + (\bar{r}^{mom} - \hat{r}_{t,t+25}^{mom})^2 = \mu + \varepsilon_t$$

El estadístico t asociado al estimador MCO del coeficiente μ en la regresión anterior sirve para contrastar las siguientes hipótesis:

$$H_0 : R_{OOS}^2 \leq 0; H_a : R_{OOS}^2 > 0$$

Es decir, rechazamos la hipótesis de que la predicción LSTM es igual o peor que la media histórica cuando el p-valor del contraste unilateral de significatividad de la estimación de μ es suficientemente pequeño.

Además de comparar la capacidad predictiva de los modelos en términos estadísticos, también comparamos el valor económico generado para un inversor representativo. Consideramos un inversor con preferencias media-varianza que distribuye su riqueza entre bonos sin riesgo y una estrategia *momentum*. Dependiendo del método de predicción usado por el inversor, los pesos de la estrategia *momentum* en la cartera vendrán dados por:

$$w_{M,t,t+25} = \frac{1}{\gamma} \left(\frac{\bar{r}^{mom} - r_{t,t+25}^f}{\hat{\sigma}_t^2} \right)$$

$$w_{LS,t,t+25} = \frac{1}{\gamma} \left(\frac{\hat{r}_{t,t+25}^{mom} - r_{t,t+25}^f}{\hat{\sigma}_t^2} \right)$$

donde $r_{t,t+25}^f$ es la rentabilidad libre de riesgo a 25 días y γ captura la aversión al riesgo del inversor¹². Siguiendo los procedimientos de la literatura (Dong *et al.*, 2021), asumimos que γ toma un valor de 3 y que el inversor estima la volatilidad de la rentabilidad de *momentum* $\hat{\sigma}_t$ usando la varianza de las rentabilidades en los anteriores doce meses. La utilidad de cada estrategia viene dada por:

$$\bar{U}_j = \bar{r}_j - \frac{\gamma}{2} \hat{\sigma}_j^2 \text{ para } j \in \{M, LS\}$$

donde \bar{r}_j es la rentabilidad media de la estrategia j respecto a la rentabilidad libre de riesgo y $\hat{\sigma}_j$ es la desviación típica de la rentabilidad. Para determinar el valor económico creado con el modelo de predicción *LSTM* respecto a la media histórica, calculamos la diferencia de utilidades $\Delta = \bar{U}_{LS} - \bar{U}_M$.

Ya que utilizamos datos diarios para estimar el modelo de predicción a 25 días, para cada modelo y cada día obtenemos 25 carteras activas, de las cuales solamente una se rebalanea. Por tanto, calculamos por separado la utilidad generada por cada cartera entre sus fechas de rebalanceo a lo largo de la muestra de test y posteriormente calculamos la media entre las 25 carteras.

Un aspecto relevante para señalar es que la estrategia que utiliza la media histórica para determinar el peso de la cartera *momentum* puede parecer una opción inferior. Sin embargo, Welch y Goyal (2008) muestran que la media histórica es una opción competitiva en relación a los modelos lineales tradicionales para predecir la rentabilidad. Además, la estrategia de inversión de la cartera media-varianza está ajustada mediante una estimación de la volatilidad usando la volatilidad pasada. Este tipo de ajuste suele generar rentabilidades superiores a carteras que no están ajustadas por volatilidad (Moreira y Muir, 2017). Además, en el caso de *momentum*, ese ajuste es suficiente para reducir la exposición a los episodios de rentabilidad negativa extrema o *crashes* (Barroso y Santa Clara, 2015; Daniel y Moskowitz, 2016).

3. RESULTADOS

El cuadro 1 presenta los resultados de la predicción *out-of-sample* de cada uno de los modelos *LSTM* entrenados, donde la referencia en cada caso es la predicción de rentabilidad que utiliza la media histórica en las muestras de entrenamiento y validación. La columna 1 muestra que el estimador R_{OOS}^2 es superior al 0,5 % en todos los modelos, lo que implica ganancias de predicción relevantes, desde el punto de vista estadístico y económico, del uso del modelo no lineal respecto a una predicción solamente basada en la media histórica. El mayor poder predictivo está generado por el modelo que solamente utiliza información sobre las rentabilidades de las carteras *momentum* a un día y a 120 días, R_{OOS}^2 del 2,49 %. Cuando incluimos más información en el modelo añadiendo la rentabilidad a 60 días y a 240 días

¹² La información sobre la rentabilidad libre de riesgo también se ha extraído de la web de Kenneth French y se corresponde con la rentabilidad diaria de las letras del Tesoro de EE. UU. con vencimiento a un mes.

como elementos del vector x_s , el poder predictivo se reduce. Este resultado evidencia cierto exceso de ajuste (*overfitting*) en las muestras de entrenamiento y validación cuando incluimos más información en el modelo *LSTM*. En términos estadísticos, en tres de los cuatro modelos podemos rechazar la hipótesis nula de que el modelo *LSTM* sea igual o inferior en términos de errores de predicción comparados con la predicción basada en la media histórica.

En términos de valor económico para un inversor con preferencias media-varianza, la columna 2 muestra las ganancias de utilidad generadas por los modelos *LSTM*. En consonancia con los estadísticos R_{OOS}^2 de la columna 1, los modelos *LSTM* generan una ganancia de utilidad con respecto al modelo de predicción de referencia. De nuevo el modelo que genera una mayor ganancia de utilidad es el que combina información de las rentabilidades de las carteras *momentum* a un día y a 120 días.

Pese a lo anterior, un aspecto a destacar es que el *ranking* de estrategias basados en valor económico no sigue el mismo orden que en términos de poder predictivo. Paradójicamente, la peor estrategia en valor económico es la que utiliza un conjunto más amplio de información. Esto se debe a que el R_{OOS}^2 es una medida que captura puramente la capacidad predictiva, sin distinguir entre compensación por el riesgo (“beta”) y compensación por errores de valoración (“alpha”). Además el ECM de la estrategia puede estar muy influenciado por eventos extremos en los que el modelo predice extremadamente bien. El modelo que utiliza más información puede ser más preciso en media pero generar una variación excesiva en la rentabilidad, como de hecho mostraremos a continuación.

Cuadro 1.

Poder de predicción *out-of-sample* y valor económico

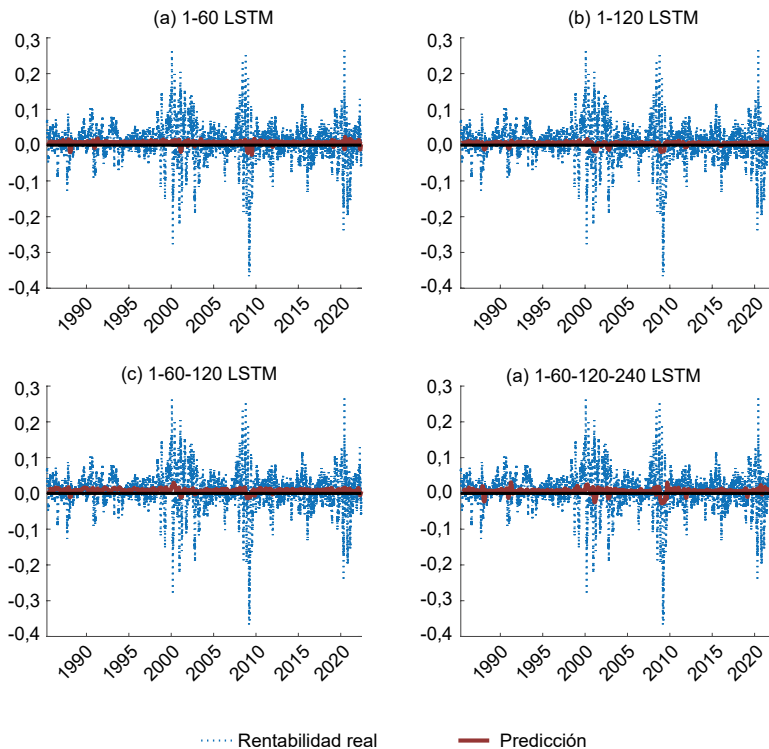
Modelo	(1) R_{OOS}^2 (%)	(2) Δ (%)
LSTM 1-60	0,536 (0,150)	0,990
LSTM 1-120	2,486* (0,071)	2,585
LSTM 1-60-120	1,896* (0,058)	1,787
LSTM 1-60-120-240	2,100* (0,099)	0,568

La figura 8 ilustra la capacidad predictiva del modelo como proporción de la variación global de la rentabilidad asociada al factor *momentum*. Los paneles representan, en líneas sólidas, las predicciones de rentabilidad realizadas por cada modelo *LSTM*. Las líneas discontinuas representan la rentabilidad real del factor *momentum*. Los modelos generan predicciones en general de rentabilidad positiva con poca variación en relación con la variación real de la rentabilidad. No obstante, los modelos tienden a predecir una rentabilidad negativa en la serie temporal en ciertos periodos señalados, que suelen estar marcados por episodios de gran volatilidad. Por tanto, los modelos *LSTM* consiguen poder predictivo por encima

de una simple media histórica al aprender la relación entre volatilidad de la rentabilidad de *momentum* y la realización de rentabilidades negativas.

Figura 8.

Rentabilidades modelo LSTM vs. factor *momentum*



La figura 9 muestra la rentabilidad acumulada por las distintas estrategias por encima de la rentabilidad libre de riesgo en el periodo *out-of-sample*, expresada en logaritmos. En todos los paneles, a modo de referencia y en una serie temporal con puntos, se muestra la rentabilidad acumulada de una estrategia “Buy&Hold” que invierte todos los fondos del inversor en la cartera *momentum*¹³. Esta estrategia genera una rentabilidad positiva a lo largo del periodo, pero muy influenciada por el *crash* sufrido por la estrategia en 2009.

A su vez, la serie temporal con líneas discontinuas muestra la rentabilidad acumulada por la estrategia que determina el peso sobre la cartera *momentum* utilizando la media histórica, denominada estrategia “Base”. Como muestra la figura, esta estrategia bate la rentabilidad de la estrategia “Buy&Hold”. Esta mayor rentabilidad se explica fundamentalmente por

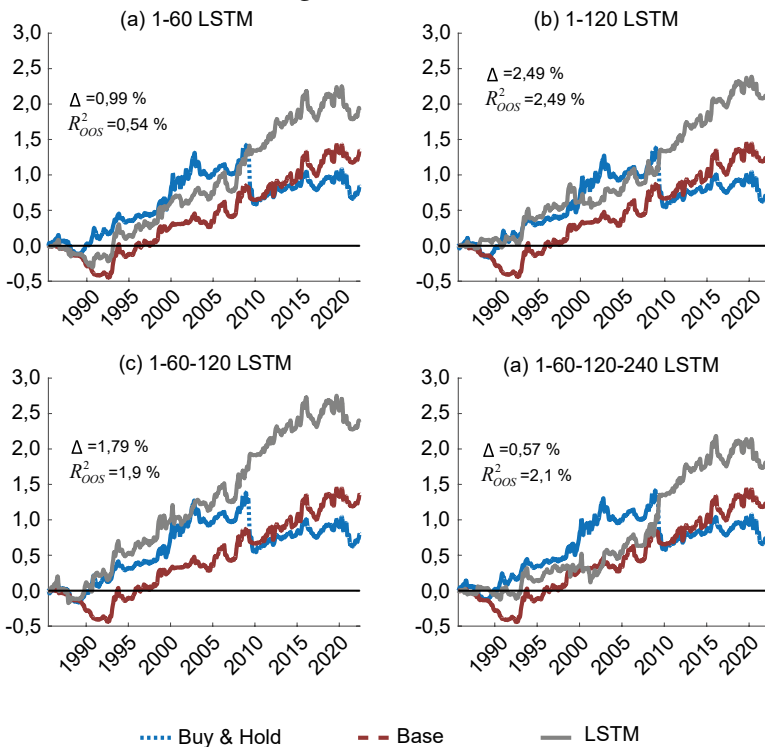
¹³ Ya que en cada momento hay 25 carteras activas, cada punto en el gráfico muestra una media con ventana móvil de 25 observaciones pasadas.

su mínima exposición al *crash* de 2009 y corrobora la efectividad de la ponderación por la volatilidad pasada (Barroso y Santa Clara, 2015; Daniel y Moskowitz, 2016).

Por su parte, cada uno de los paneles de la **figura 9** representa en una línea continua la rentabilidad acumulada por las cuatro estrategias basadas en los modelos *LSTM*. Los cuatro modelos de predicción son capaces de batir a las otras dos estrategias de referencia en términos de rentabilidad acumulada. La mayor rentabilidad acumulada está generada por la estrategia que utiliza las rentabilidades de las carteras *momentum* a un día, a 60 días y a 120 días. Las ganancias de las estrategias *LSTM* se basan en una predicción notable del *crash* de 2009 por parte de todos los modelos.

Figura 9.

Rentabilidad acumulada estrategias *LSTM*



Para completar el análisis de los resultados, el **cuadro 2** muestra algunos estadísticos descriptivos adicionales de las estrategias. Los resultados muestran cómo las estrategias basadas en *LSTM* generan una rentabilidad media a 25 días considerablemente mayor que las estrategias “Base” y “Buy&Hold”. La volatilidad de las estrategias *LSTM* son relativamente superiores a la estrategia “Base”, pero sus ratios de Sharpe muestran que la compensación en rentabilidad por unidad de riesgo es sustancialmente mayor.

La comparación entre los distintos modelos *LSTM* ofrece una visión análoga a los resultados presentados anteriormente. La estrategia que solamente utiliza información sobre las rentabilidades de las carteras *momentum* a un día y a 120 días supera en términos de relación rentabilidad-riesgo a las otras estrategias, pese a no conseguir la rentabilidad media mayor. Además, el signo de la asimetría de esas rentabilidades se invierte con respecto al resto de estrategias (excepto para el modelo que utiliza el mayor nivel de información) sugiriendo una menor exposición a rentabilidades extremas negativas, como también indica el valor de curtosis (7,4 respecto a 9,3 de la estrategia “Buy&Hold”).

Cuadro 2.

Estadísticos de las distintas estrategias

Modelo	Rentabilidad media (%)	Volatilidad (%)	Ratio de Sharpe	Asimetría	Curtosis
Buy&Hold	0,649	5,068	0.24	-0.996	9.298
Base	0,744	4,526	0.344	-0.038	7.002
LSTM 1-60	0,919	5,169	0.419	-0.085	6.696
LSTM 1-120	0,984	4,701	0.508	0.267	7.492
LSTM 1-60-120	1,096	5,832	0.476	-0.244	6.427
LSTM 1-60-120-240	0,92	5,392	0.404	0.219	8.298

4. CONCLUSIONES

En este trabajo hemos analizado la capacidad de las arquitecturas de redes neuronales *Long Short-Term Memory (LSTM)* para predecir la serie temporal del factor *momentum*. La arquitectura *LSTM* supone una de las herramientas más potentes en el campo de *machine learning* para realizar predicciones en datos con forma de secuencia, pero ha sido aplicado de manera muy limitada en el área académica de las finanzas. Por su parte, la anomalía *momentum* supone uno de los fenómenos más analizados en el ámbito de las finanzas por su prevalencia en el tiempo y su marcado carácter de “anomalía” que desafía a las teorías de eficiencia.

Los resultados sugieren que los modelos *LSTM* pueden predecir de manera notable la serie temporal del factor *momentum* y anticipar episodios de rentabilidad extrema. Las estrategias de inversión que resultan de los modelos *LSTM* dominan considerablemente a una estrategia basada en la media histórica, incluso ponderando las estrategias por la volatilidad pasada del factor *momentum*. Las ganancias en la predicción se traducen en medidas de R^2 fuera de la muestra (*out-of-sample*) y ganancias económicas de más del 2 %.

Diversas líneas de trabajo pueden continuar el análisis realizado en este estudio. En primer lugar, las arquitecturas entrenadas son relativamente sencillas y están destinadas a ilustrar el poder predictivo de este tipo de modelos. Determinar la capacidad predictiva de estos modelos necesita un análisis más exhaustivo de las arquitecturas, los hiperparámetros y

la información usada en el modelo. Segundo, resulta de interés aplicar las arquitecturas *LSTM* para comprobar su poder predictivo respecto a otros factores y anomalías analizados en la literatura. Finalmente, queda la tarea pendiente de refinar la interpretabilidad de la información absorbida por las redes *LSTM*. Para ello, someter el entrenamiento de la arquitectura a la intuición económica nos puede servir para detectar los patrones relevantes de predictibilidad de las series temporales en finanzas.

Referencias

- ALTI, A. y TITMAN, S. (2019). A dynamic model of characteristic-based return predictability. *The Journal of Finance*, 74(6), 3187-3216.
- AVRAMOV, D., CHORDIA, T., JOSTOVA, G. y PHILIPPOV, A. (2013). Anomalies and financial distress, *Journal of Financial Economics*, 108, 1391-159.
- BACHELIER, L. (1900). Théorie de la spéculation. *Annales Scientifiques de l'École Normale Supérieure*, vol. 3, no. 17, pp. 218-6.
- BARROSO, P. y SANTA-CLARA, P. (2015). Momentum has its moments. *Journal of Financial Economics*, 116(1), pp. 111-120.
- BLANCO, I., DE JESUS, M. y REMESAL, A. (2023). Overlapping Momentum Portfolios. *Journal of Empirical Finance*, 72, pp. 1-22.
- BOLLERSLEV, T., PATTON, A. J. y QUAEDVLIEG, R. (2016). Exploiting the errors: A simple approach for improved volatility forecasting. *Journal of Econometrics*, 192(1), pp. 1-18.
- CAMPBELL, J. Y. (1987). Stock returns and the term structure. *Journal of Financial Economics*, 18, 373-399.
- CAMPBELL, J. Y. y SHILLER, R. J. (1988). The dividend-price ratio and expectations of future dividends and discount factors. *The Review of Financial Studies*, 1(3), pp. 195-228.
- CHEN, L., PELGER, M. y ZHU, J. (2023). Deep learning in asset pricing. *Management Science*.
- CHINCO, A., CLARK-JOSEPH, A. D. y YE, M. (2019). Sparse signals in the cross-section of returns. *Journal of Finance*, 74, 4494-92.
- CLARK, T. E. y WEST, K. D. (2007). Approximately normal tests for equal predictive accuracy in nested models. *Journal of Econometrics*, 138(1), 291-311.
- COCHRANE, J. H. (2007). The dog that did not bark: A defense of return predictability. *The Review of Financial Studies*, 21, 1533-75.
- CONG, L., KE TANG, J. y ZHANG, Y. (2021). Alphaportfolio: Direct construction through deep reinforcement learning and interpretable AI. *Working paper*. Cornell University, Tsinghua University y Beihang University.
- CORSI, F. (2009). A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics*, 7(2), pp. 174-196.
- DA, Z., GURUN, U. G. y WARACHKA, M. (2014). Frog in the pan: Continuous information and momentum. *Review of Financial Studies*, 27(7), 2171-2218.
- DANIEL, K., HIRSHLEIFER, D. y SUBRAHMANYAM, A. (1998). Investor psychology and security market under- and overreactions. *Journal of Finance*, 53(6), 1839-1885.
- DANIEL, K. y MOSKOWITZ, T. J. (2016). Momentum crashes. *Journal of Financial Economics*, 122(2), pp. 221-247.

- DONG, X., LI, Y., RAPACH, D. E. y ZHOU, G. (2022). Anomalies and the Expected Market Return. *The Journal of Finance*, 77, pp. 639-681.
- ENGELBERG, J., DAVID MCLEAN, R. y PONTIFE, J. (2018). Anomalies and news. *Journal of Finance*, 73, 19712001.
- FAMA, E. F., FISHER, L., JENSEN, M. C. y ROLL, R. (1969). The Adjustment of Stock Prices to New Information. *International Economic Review*, 10(1), p. 121.
- FAMA, E. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*, 25(2), 383417.
- FAMA, E. F. y FRENCH, K. R. (1988). Dividend yields and expected stock returns. *Journal of Financial Economics*, 21, p. 325.
- FAMA, E. F. y FRENCH, K. R. (1989). Business conditions and the expected returns on stocks and bonds. *Journal of Financial Economics*, 25, p. 2349.
- FAMA, E. F. y FRENCH, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33, p. 356.
- FAMA, E. F. y FRENCH, K. R. (2015). A five-factor asset pricing model. *Journal of Financial Economics*, 116, p. 122.
- FAMA, E. F. y FRENCH, K. R. (2020). Comparing cross-section and time-series factor models. *Review of Financial Studies*, 33, 18911926.
- FREYBERGER, J., NEUHIERL, A. y WEBER, M. (2020). Dissecting characteristics nonparametrically. *Review of Financial Studies*, 33, 23262377.
- GERON, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (2nd ed.). O'Reilly.
- GRANGER, C. W. J. y MORGENSTERN, O. (1970). *Predictability of Stock Market Prices*. Heath Lexington Books.
- GU, S., KELLY, B. y XIU, D. (2020). Empirical asset pricing via machine learning. *Review of Financial Studies*, 33, 22232273.
- GU, S., KELLY, B. y XIU, D. (2021). Autoencoder asset pricing models. *Journal of Econometrics*, 222(1), pp. 429-450.
- HAMILTON, J. D. (1994). *Time Series Analysis*. Princeton University Press.
- HAN, Y., HE, A., RAPACH, D. E. y ZHOU, G. (2021). Expected stock returns and firm characteristics: E-LASSO, assessment, and implications. *Working paper*. University of North Carolina, Charlotte, University of South Carolina, Saint Louis University y Washington University in St. Louis.
- HARVEY, C. R., LIU, Y. y ZHU, H. (2016). And the cross-section of expected returns. *Review of Financial Studies*, 29, p. 568.
- HIRSHLEIFER, D. y LUO, G. Y. (2001). On the survival of overconfident traders in a competitive securities market. *Journal of Financial Markets*, 4(1), 7384.
- HIZMERI, R., IZZELDIN, M., NOLTE, I. y PAPPAS, V. (2022). A generalized heterogeneous auto-regressive model using market information. *Quantitative Finance*, pp. 1-22.
- HOCHREITER, S. y SCHMIDHUBER, J. (1997). Long short-term memory. *Neural computation*, 9(8), pp. 1735-1780.
- HONG, H. y STEIN, J. C. (1999). A unified theory of underreaction, momentum trading, and overreaction in asset markets. *Journal of Finance* 54(6), 21432184.
- HOU, K., XUE., C. y ZHANG, L. (2020). Replicating anomalies. *Review of Financial Studies*, 33, 20192133.
- HUANG, D., LI, J., WANG, L. y ZHOU, G. (2020). Time series momentum: Is it there? *Journal of Financial Economics*, 135(3), pp. 774-794.

- HUANG, S. (2022). The Momentum Gap and Return Predictability. *The Review of Financial Studies*, Volume 35, Issue 7, July 2022, 33033336.
- JEGADEESH, N. (1990). Evidence of predictable behavior of security returns. *The Journal of Finance*, 45(3), pp. 881-898.
- JEGADEESH, N. y TITMAN, S. (1993). Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of Finance*, 48(1), pp. 65-91.
- KOZAK, SERHIY, S. N. y SANTOSH, S. (2020). Shrinking the cross-section. *Journal of Financial Economics*, 135, 271292.
- LEHMANN, B. N. (1990). Fads, martingales, and market efficiency. *The Quarterly Journal of Economics*, 105(1), pp. 1-28.
- LIU, Y., ZHOU, G. y ZHU, Y. (2021). Maximizing the Sharpe ratio: A genetic programming approach. *Working paper*. Tsinghua University and Washington University in St. Louis.
- MCLEAN, R. D. y PONTIFF, J. (2016). Does academic research destroy stock return predictability? *The Journal of Finance*, 71(1), pp. 5-32.
- MOREIRA, A. y MUIR, T. (2017). Volatility-managed portfolios. *The Journal of Finance*, 72(4), pp. 1611-1644.
- MOSKOWITZ, T. J., OOI, Y. H. y PEDERSEN, L. H. (2012). Time series momentum. *Journal of Financial Economic*, 104(2), pp. 228-250. Special Issue on Investor Sentiment.
- NELSON, C. R. (1976). Inflation and rates of return on common stocks. *The Journal of Finance*, 31, 471483.
- PÁSTOR, L. y STAMBAUGH, R. F. (2003). Liquidity risk and expected stock returns. *Journal of Political Economy*, 111, 642685.
- PÁSTOR, L. y STAMBAUGH, R. F. (2009). Predictive systems: Living with imperfect predictors. *The Journal of Finance*, 64, 15831628.
- RAPACH, D. E., STRAUSS, J. K. y ZHOU, G. (2013). International stock return predictability: What is the role of the United States? *Journal of Finance*, 68, 16331662.
- RAPACH, D. E. y ZHOU, G. (2020). Timeseries and crosssectional stock return forecasting: New machine learning methods. *Machine learning for asset management*. *New developments and financial applications*, 1-33.
- ROSS, S. A. (1976). The arbitrage theory of capital asset pricing. *Journal of Economic Theory*, 13(3), pp. 341-360.
- WELCH, I. y GOYAL, A. (2008). A comprehensive look at the empirical performance of equity premium prediction. *The Review of Financial Studies*, 21(4), pp. 1455-1508.