

Trabajo de Fin de Máster

# Aplicación de redes neuronales artificiales y programación cuadrática en la gestión de carteras

David Díaz Rodríguez

Tutor: Xosé Manuel Martínez Filgueira

**Máster Universitario en Banca y Finanzas**

**Curso académico 2022/2023**

---

Trabajo de Fin de Máster presentado en la Facultad de  
Economía y Empresa de la Universidade da Coruña  
para la obtención del Máster en Banca y Finanzas

# Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas que contribuyeron de manera significativa a la realización de este trabajo final de máster. Sin su apoyo, orientación y aliento, este logro no habría sido posible.

Deseo agradecer a la Xunta de Galicia por proporcionarme los recursos para la realización de estudios de máster como beneficiario de la Becas Excelencia Juventud Exterior.

Deseo agradecer a Universidade da Coruña por los conocimientos adquiridos al cursar el Máster Universitario en Banca y Finanzas. Su compromiso con la excelencia académica ha sido una fuente constante de inspiración.

Agradezco profundamente a mi tutor Xosé Manuel Martínez Filgueira por su orientación experta y valiosas sugerencias a lo largo de este proceso. Sus conocimientos y dedicación fueron fundamentales para dar dirección y calidad a este trabajo.

Mis agradecimientos se extienden a mis compañeros de clase y amigos que brindaron un espacio para discusiones enriquecedoras y aportes valiosos que contribuyeron al desarrollo de este trabajo.

También quiero expresar mi gratitud a mi familia por su constante apoyo emocional y comprensión durante los momentos desafiantes de este proceso académico. Este trabajo sienta las bases para futuras investigaciones, destacando la importancia de utilizar datos actualizados y configurar adecuadamente los modelos para lograr una gestión de carteras más informada y efectiva en un entorno financiero en constante evolución.

# Resumen

En el contexto del mundo financiero en constante cambio y complejidad, este trabajo aborda la aplicación de redes neuronales artificiales y programación cuadrática en la gestión de carteras financieras. Se destaca la importancia de caracterizar adecuadamente las series temporales financieras para realizar pronósticos más precisos y se examina el potencial de la combinación de las redes neuronales convolucionales y LSTM para mejorar la previsión de series de tiempo. En el proceso de composición de carteras, se aplica la programación cuadrática como una técnica eficiente para lograr una distribución óptima de activos financieros. En conclusión, el enfoque de combinar redes neuronales artificiales y programación cuadrática muestra promesa en la gestión de carteras financieras, pero es necesario un estudio más profundo y exhaustivo para determinar su eficiencia óptima. Este trabajo sienta las bases para futuras investigaciones, destacando la importancia de utilizar datos actualizados y configurar adecuadamente los modelos para lograr una gestión de carteras más informada y efectiva en un entorno financiero en constante evolución.

*Palabras clave:* gestión de carteras, carteras, redes neuronales artificiales, programación cuadrática, series temporales financieras, predicción de precios, composición de carteras.

*Número de palabras:* 14001

# Abstract

In the context of the financial world in constant change and complexity, this work deals with the application of artificial neural networks and quadratic programming in the management of financial portfolios. The importance of properly characterizing financial time series for more accurate forecasting is highlighted, and the potential of combining convolutional neural networks and LSTM to improve time series forecasting is examined. In the portfolio composition process, quadratic programming is applied as an efficient technique to achieve an optimal distribution of financial assets. In conclusion, the approach of combining artificial neural networks and quadratic programming shows promise in the management of financial portfolios, but a deeper and more exhaustive study is necessary to determine its optimal efficiency. This paper lays the groundwork for future research, highlighting the importance of using up-to-date data and properly configuring models to achieve more informed and effective portfolio management in an ever-evolving financial environment.

Keywords: portfolio management, portfolios, artificial neural networks, quadratic programming, financial time series, price prediction, portfolio composition

# Resumo

No contexto do mundo financeiro en constante cambio e complexidade, este traballo trata sobre a aplicación das redes neuronais artificiais e da programación cuadrática na xestión de carteras financeiras. Destaca a importancia de caracterizar adecuadamente as series temporales financeiras para unha previsión máis precisa e examínase o potencial de combinar redes neuronais convolucionais e LSTM para mellorar a previsión de series temporais. No proceso de composición da carteira aplícase a programación cuadrática como técnica eficiente para conseguir unha distribución óptima dos activos financeiros. En conclusión, o enfoque de combinar redes neuronais artificiais e programación cuadrática resulta prometedor na xestión de carteras financeiras, pero é necesario un estudo máis profundo e exhaustivo para determinar a súa eficiencia óptima. Este traballo senta as bases para futuras investigacións, destacando a importancia de utilizar datos actualizados e de configurar adecuadamente os modelos para lograr unha xestión de carteira máis informada e eficaz nun entorno financeiro en constante evolución.

Palabras clave: xestión de carteras, carteras, redes neuronais artificiais, programación cuadrática, series temporales financeiras, predición de prezos, composición da carteira.

# Índice

<b>1. Introducción.....</b>	<b>10</b>
<b>2. Desarrollo del trabajo.....</b>	<b>12</b>
2.1. Caracterización de las series temporales financieras.....	12
2.1.1. Series de tiempo y sus características .....	12
2.1.2. Características de los precios.....	15
2.2. Redes neuronales artificiales en la previsión de las series de tiempo .....	18
2.2.1 Antecedentes del uso de redes neuronales artificiales en la previsión de series de tiempo.....	18
2.2.2 Redes neuronales convolucionales .....	21
2.2.3 Long short-term memory .....	24
2.3 Composición de carteras.....	26
2.3.1 Problema y técnicas .....	26
2.3.2 Programación cuadrática .....	29
2.4 Datos .....	33
2.4.1 Obtención de Datos .....	34
2.4.2 Indicadores.....	35
2.4.3 Vectores.....	39
2.5 Modelado y entrenamiento .....	41
2.5.1 Modelado .....	42
2.5.2 Entrenamiento .....	43
2.6 Resultado.....	44

2.6.1 Predicciones.....	45
2.6.2 Composición de carteras.....	48
<b>3. Conclusiones.....</b>	<b>50</b>
<b>Bibliografía.....</b>	<b>52</b>
<b>Anexos.....</b>	<b>60</b>
Anexo.1 Figuras.....	60
Anexo.2 Gráficas.....	68
Anexo.3 Tablas.....	75
Anexo.4 Códigos.....	86

# Índice de figuras

Figura 1: Relación entre IA-ML-DL.....	60
Figura 2: Programación clásica y machine learning .....	60
Figura 3: Estructura básica de una red neuronal artificial .....	60
Figura 4: Como el tamaño del filtro afecta el vector de salida .....	61
Figura 5: Como stride afecta el vector de salida .....	61
Figura 6: Como dilation afecta el vector de salida .....	62
Figura 7: Como padding afecta el vector de salida.....	62
Figura 8:Despliegue del bucle de una red neuronal recurrente estándar.....	63
Figura 9: Información relevante cercana.....	63
Figura 10: Información relevante lejana.....	63
Figura 11: Diferencia entre los módulos de repetición.....	64
Figura 12: LSTM funcionalidad: Representación del paso 1.....	64
Figura 13: LSTM funcionalidad: Representación del paso 2.....	64
Figura 14: LSTM funcionalidad: Representación del paso 3.....	65
Figura 15: LSTM funcionalidad: Representación del paso 4.....	65
Figura 16: Visualización de vectores de entrada y salida.....	66
Figura 17: Distintas estructuras según los distintos tamaños de vectores de entrada	66
Figura 18 Dominio de ReLU y Leaky ReLU.....	67
Figura 19: Diagrama de flujo de la metodología de Walk Forward Validation.....	67



## Índice de gráficas

Gráfica 1: Tendencia alcista y heterocedástica.....	68
Gráfica 2: Tendencia bajista y heterocedástica.....	68
Gráfica 3: Sin tendencia, homocedástica y estacionaria .....	69
Gráfica 4: Descomposición: estacionalidad y error.....	69
Gráfica 5: Correlograma .....	70
Gráfica 6: Tendencia constante en los precios de cierre ajustado de la empresa.....	70
Gráfica 7: Cambio brusco de precios que reflejan calculo erróneo de los precios de cierre ajustado, .....	71
Gráfica 8: Evolución de los indicadores – Entradas con una observación.....	71
Gráfica 9: Evolución de los indicadores – Entradas con dos observaciones.....	72
Gráfica 10: Evolución de los indicadores – Entradas con tres observaciones .....	72
Gráfica 11: Evolución de las carteras y el IBEX – Entradas con una observación .....	73
Gráfica 12: Evolución de las carteras y el IBEX – Entradas con dos observaciones .	73
Gráfica 13: Evolución de las carteras y el IBEX – Entradas con tres observaciones .	74

# Índice de tablas

Tabla 1 Estructura de datos de precios.....	75
Tabla 2 Lista de empresas cotizadas .....	76
Tabla 3 Cantidades de muestras utilizadas para entrenar los modelos.....	85
Tabla 4 Mejores y peores empresas según los resultados obtenidos de los cálculos de los indicadores.....	85

# 1. Introducción

En el ámbito financiero, la gestión eficiente de carteras es una tarea crucial para los inversionistas y administradores de activos, ya que busca maximizar los rendimientos y minimizar los riesgos asociados a las inversiones. En los últimos años, el campo de la inteligencia artificial y el aprendizaje automático ha experimentado un notable avance, lo que ha permitido la aplicación de técnicas innovadoras para mejorar el proceso de toma de decisiones financieras.

Este trabajo se enfoca en la integración de dos poderosas herramientas: las redes neuronales artificiales y la programación cuadrática, para abordar el desafío de la gestión de carteras. La combinación de estas técnicas ofrece un enfoque sólido y prometedor para la previsión de series de tiempo financieras y la composición de carteras en un entorno financiero altamente dinámico y complejo.

El desarrollo del trabajo se estructura en varias secciones fundamentales para abordar de manera integral el tema. En primer lugar, se realiza una caracterización detallada de las series temporales financieras, examinando sus características y propiedades esenciales para entender mejor el comportamiento de los precios de los activos.

A continuación, se explora el potencial de las redes neuronales artificiales en la previsión de las series de tiempo. Se presentan los antecedentes del uso de estas redes en este contexto y se destacan dos arquitecturas ampliamente utilizadas: las redes neuronales convolucionales y las redes *Long Short-Term Memory* (LSTM), ambas con la capacidad de capturar patrones complejos en los datos financieros.

La sección sobre composición de carteras aborda el problema y presenta diversas técnicas aplicadas en la gestión de activos. Es aquí donde se introduce la programación cuadrática como una herramienta relevante y eficiente para la construcción óptima de carteras de inversión.

La obtención de datos precisos y relevantes es crucial para cualquier análisis financiero y el trabajo con algoritmos de *Machine Learning*. Se describe la metodología aplicada para la obtención de datos, y como se computaron algunos de los indicadores más comunes

utilizados en finanzas para usar como variables descriptivas del problema en conjunto con los datos históricos. Se expone además como se estructuran los vectores que se utilizarán en el modelado y entrenamiento de las redes neuronales.

En las últimas secciones, se aborda el proceso de modelado y entrenamiento, que implica la configuración adecuada de las redes neuronales y la implementación de la programación cuadrática para obtener resultados óptimos. Finalmente, se presentan los resultados obtenidos, incluyendo las predicciones generadas por las redes neuronales y la composición de carteras recomendadas, demostrando así la efectividad de la metodología propuesta en la gestión de carteras financieras.

En conjunto, este trabajo busca aportar una visión integral y actualizada sobre el uso de redes neuronales artificiales y programación cuadrática en la gestión de carteras, destacando su potencial como una opción para mejorar la toma de decisiones financieras y brindar a los inversionistas una herramienta valiosa para optimizar sus estrategias de inversión en un entorno cambiante y competitivo.

## 2. Desarrollo del trabajo

El presente trabajo se adentra en el campo del aprendizaje automático y la inteligencia artificial, específicamente en el uso combinado de redes neuronales artificiales y programación cuadrática. Esta poderosa sinergia busca ofrecer una solución efectiva y sofisticada para la previsión de series temporales financieras y la composición óptima de carteras de inversión. A través de la aplicación de estas técnicas, se persigue mejorar la toma de decisiones financieras y maximizar los rendimientos, al tiempo que se minimizan los riesgos asociados a las inversiones. En las siguientes secciones, se explorarán en detalle las distintas etapas del proceso, desde la caracterización de las series temporales y el funcionamiento de las redes neuronales, hasta la implementación de la programación cuadrática en la construcción de carteras eficientes.

### 2.1. Caracterización de las series temporales financieras

Este epígrafe se divide en dos sub-epígrafes. En el primero se presentan los conceptos fundamentales relacionados con las series de tiempo, incluyendo sus características, componentes y clasificaciones, y se establece su relación con el análisis de las series cronológicas y la previsión, que es el objetivo de este informe. En el segundo sub-epígrafe se profundiza en las características de los precios de las acciones, comenzando por una descripción general y avanzando hacia aspectos más específicos, así como en la importancia de la estructura en la que se encuentran los datos relacionados con estos.

#### 2.1.1. Series de tiempo y sus características

Las series cronológicas son un tipo de proceso estocástico que se caracteriza por ordenar las variables aleatorias según el tiempo. Esto significa que cada momento tiene asociado un valor de la variable que depende del azar y que puede cambiar a lo largo del tiempo. Según Ruiz (2011), un proceso estocástico es “una colección o familia de variables aleatorias, ordenadas según un subíndice que en general suele ser el tiempo” (p.01). El análisis de las series de tiempo puede tener distintos fines, como describir el

comportamiento de las variables o predecir o pronosticar sus valores futuros, lo que es especialmente relevante para las series financieras.

El análisis de las series de tiempo es una herramienta estadística que permite estudiar el comportamiento de una variable a lo largo del tiempo. Sin embargo, no existe un consenso único sobre los componentes que se deben considerar en este tipo de análisis. Algunos autores, como Kocenda y Cerný (2017) y Anderson et al. (2017), proponen que las series de tiempo se pueden descomponer en tres componentes: tendencia, estacionalidad y ruido. Otros autores, como Dodge (2008) y Espallargas y Solís (2012), sugieren que se debe añadir un cuarto componente: el ciclo. Finalmente, hay autores que plantean que las series de tiempo pueden tener hasta cinco componentes estos son los casos de IBM (2021) y Chirinos (2018).

**Tendencia:** Es el patrón de cambio a largo plazo que se observa en una serie de datos. Se puede definir como la dirección general y persistente de las variaciones de la serie a lo largo del tiempo. Se puede clasificar en positiva (*Gráfica 1*), negativa (*Gráfica 2*) o nula (*Gráfica 3*), dependiendo de si la serie aumenta, disminuye o se mantiene constante en el largo plazo. La tendencia puede identificarse mediante el análisis gráfico o mediante métodos estadísticos. Este componente es importante para entender el comportamiento histórico y proyectar el futuro de una serie de datos, el mismo es común en los distintos criterios mencionados.

**Estacionalidad:** Llamada también variación cíclica regular: Se refiere a la variación correspondiente a los movimientos de la serie que ocurren cada cierto periodo de tiempo, *Gráfica 4*. Este componente es, al igual que la tendencia, común en los criterios mencionados. Diferenciándose en que aquellos autores que exponen cuatro y cinco componentes llaman estacionalidad a las variaciones periódicas correspondientes a periodos menores o iguales a un año (como son periodicidad diaria, semanal, mensual, o anual), mientras que las variaciones periódicas correspondientes a periodos mayores las contemplan en un componente llamado, variaciones cíclicas. Por lo que para determinar la estacionalidad de una serie de tiempo es necesario analizarlas en un periodo no menor de dos años.

Un componente que no se puede explicar por los otros elementos de la serie de tiempo es la variación irregular o error. Este componente también se conoce como variación

aleatoria, ruido o residuo, y se muestra en la [Gráfica 4](#). La variación irregular es común en los tres criterios mencionados anteriormente. Algunos autores distinguen entre la variación irregular, que es ocasional y aleatoria, y la variación atípica, que es causada por eventos aislados que alteran el comportamiento de la serie. La variación atípica se puede clasificar en varios tipos: aditiva, de innovación, de cambio de nivel, transiente, de estacionalidad aditiva y de tendencia local.

Una forma de categorizar las series de tiempo es según el grado de variabilidad que presentan a lo largo del tiempo, según lo expuesto en Villagarcía (2006) se puede distinguir entre series homocedásticas y heterocedásticas. Las series homocedásticas son aquellas que mantienen un rango constante de variación, como se muestra en la [Gráfica 3](#). Por el contrario, las series heterocedásticas son aquellas que cambian el rango de variación, aumentando o disminuyendo su amplitud, como se ilustra en las [Gráfica 1](#) y [Gráfica 2](#).

Un concepto clave en el análisis de series de tiempo es el de estacionariedad. Una serie de tiempo es estacionaria cuando sus propiedades estadísticas, como la media, la varianza y la covarianza, no cambian con el tiempo. Esto implica que la serie no presenta tendencia, ciclos ni estacionalidad. Como señalan Castillo y Varela (2010), Villavicencio (2010) y Ruiz (2011), la estacionariedad es una condición necesaria para poder predecir el comportamiento futuro de una serie de tiempo usando técnicas estadísticas. En la [Gráfica 3](#) se muestra un ejemplo de una serie de tiempo estacionaria.

Las series de tiempo financieras presentan heterocedasticidad, es decir, varianzas que cambian en el tiempo. Esto implica que no son estacionarias y que su comportamiento depende de factores externos. Para verificar la estacionariedad de una serie de tiempo, se pueden utilizar diferentes métodos, como el correlograma, que muestra las funciones de autocorrelación y autocorrelación parcial de la serie, o las pruebas de raíz unitaria, como la de Dickey Fuller o la de Phillips Perron, que contrastan la hipótesis nula de que la serie tiene una raíz unitaria. Estos métodos se explican con más detalle en Castillo y Varela (2010), Villavicencio (2010) y Ruiz (2011). La [Gráfica 5](#) ilustra un ejemplo de correlograma para una serie de tiempo financiera.

## 2.1.2. Características de los precios

Invertir en acciones o cualquier otro bien que cotice en el mercado de valores es una tarea compleja y desafiante, que requiere una comprensión profunda de las tendencias y fluctuaciones del mercado. En el centro de esta comprensión se encuentra la capacidad de analizar e interpretar los datos de precios del mercado de valores, lo que brinda información clave sobre el comportamiento de los participantes del mercado y los factores que impulsan los movimientos del mercado. El propósito de este sub-epígrafe es proporcionar una descripción general completa del entorno de los precios de las acciones y como son representados comúnmente los mismos, señalando los aspectos más importantes para la aplicación de las técnicas que se exploraran en los siguientes epígrafes.

Como se explica en la CNMV las bolsas de valores son mercados organizados donde se negocian acciones y otros valores, como renta fija, warrants, certificados y fondos cotizados. En BME se expone que, en España, existen cuatro bolsas tradicionales (Madrid, Barcelona, Bilbao y Valencia) que forman parte del holding BME (Bolsas y Mercados Españoles), que también integra otros segmentos y sistemas de negociación, compensación y liquidación de valores. Siendo, como se explica en CNMV, el Sistema de Interconexión Bursátil Español (SIBE) la plataforma que permite la contratación continua y electrónica de todos los valores admitidos a cotización en las cuatro bolsas españolas.

Como expone la CNMV las acciones son valores mobiliarios que representan una parte proporcional del capital social de una sociedad anónima, y sus tenedores son socios propietarios de la misma. Las acciones pueden negociarse en bolsas de valores o en otros mercados secundarios autorizados.

A partir de lo expuesto en Mitchell (2020), Pinset (2021) y C. Team (2023) se puede concluir que, para explicar el precio de las acciones de una empresa, se pueden considerar los siguientes factores:

- La oferta y la demanda de las acciones en el mercado: si hay más compradores que vendedores, el precio subirá y viceversa. Esto depende de las expectativas y la confianza de los inversores en el futuro de la empresa.
- Los cambios en la gestión o la producción de la empresa: si la empresa mejora su eficiencia, su rentabilidad o su innovación, el precio de sus acciones puede



aumentar. Por el contrario, si la empresa tiene problemas internos, pierde competitividad o se ve afectada por crisis externas, el precio puede bajar.

- La reputación de la empresa: si la empresa tiene una buena imagen pública, se asocia con éxitos o logros, o recibe buenas valoraciones de los analistas, el precio de sus acciones puede subir. Por el contrario, si la empresa se ve involucrada en escándalos, demandas o controversias, o recibe malas valoraciones de los analistas, el precio puede bajar.

En los textos Pinset (2021), T. I. Team (2022) y C. Team (2023) también señalan la importancia de diferenciar el precio de una empresa o acción de esta del valor intrínseco de esta. Pudiéndose resumir teniendo en cuenta lo señalado en esos textos y lo expuesto con anterioridad que el precio de una empresa o acción es lo que los compradores y vendedores están dispuestos a pagar por ella en un momento determinado, mientras que el valor intrínseco de una empresa o acción depende en gran medida de la metodología utilizada para valorar las empresas y los objetivos del evaluador.

Una vez contextualizado de manera general el entorno en el que se encuentran los precios de las acciones y explicado algunos de los factores que pueden afectar a los mismos, se explica a continuación la estructura en la que estos datos aparecen habitualmente. Generalmente los precios de las acciones se encuentran registrados de forma periódica (diariamente, semanalmente, mensualmente, anualmente, etc.). registrándose para cada periodo el precio de apertura, el precio más alto, el más bajo, el de cierre, el volumen y el de cierre ajustado, ver [Tabla 1](#).

A partir de lo expuesto en Barone (2022), Chen (2022), Downey (2022), Hayes (2021) y Ganti (2020) se puede entender que:

- El precio de apertura es el primer precio al que se negocia un activo financiero en una sesión bursátil. Este precio puede ser diferente al precio de cierre de la sesión anterior, ya que puede haber cambios en la oferta y la demanda durante el periodo en que el mercado está cerrado. El precio de apertura suele indicar el tono o la tendencia del mercado para ese día.
- El precio más alto es el mayor precio al que se negocia un activo financiero en una sesión bursátil. Este precio refleja el máximo nivel de interés de los compradores

por ese activo en ese día. El precio más alto puede ser un indicador de la fortaleza o la debilidad de un activo, así como de su volatilidad.

- El precio más bajo es el menor precio al que se negocia un activo financiero en una sesión bursátil. Este precio refleja el mínimo nivel de interés de los vendedores por ese activo en ese día. El precio más bajo puede ser un indicador de la presión o la resistencia de un activo, así como de su volatilidad.
- El precio de cierre es el último precio al que se negocia un activo financiero en una sesión bursátil. Este precio es el que se utiliza para calcular el valor de mercado de ese activo al final del día. El precio de cierre suele ser el más importante para los inversores, ya que resume el resultado de las operaciones del día y muestra la dirección del mercado.
- El volumen es la cantidad de unidades de un activo financiero que se negocian en una sesión bursátil. El volumen muestra el nivel de actividad o de liquidez de un mercado o de un activo. El volumen suele acompañar a los movimientos de los precios, ya que indica el grado de consenso o de divergencia entre los participantes del mercado.
- El precio de cierre ajustado es el precio de cierre de un activo financiero que se modifica para tener en cuenta eventos como dividendos, splits, fusiones o adquisiciones que afectan al valor del activo. El precio de cierre ajustado permite comparar el rendimiento histórico de un activo con mayor precisión y consistencia.

A partir de lo expuesto en Hayes (2021) y Ganti (2020) se entiende que la diferencia entre el precio de cierre y el precio de cierre ajustado es de gran importancia, ya que el primero puede dar una imagen distorsionada del rendimiento de una acción a lo largo del tiempo, mientras que el segundo refleja el valor real de la acción después de ajustar los factores que lo alteran.

Por ejemplo, la junta directiva de una empresa puede decidir dividir las acciones de la empresa 3 por 1. Por lo tanto, las acciones en circulación de la empresa aumentan en un múltiplo de tres, mientras que el precio de sus acciones se divide por tres. Supongamos que una acción cerró a \$300 el día anterior a su división de acciones. En este caso, el precio de cierre se ajusta a \$100 (\$300 divididos por 3) por acción para mantener un estándar de comparación consistente. De manera similar, todos los demás precios de cierre anteriores

para esa empresa se dividirían por tres para obtener los precios de cierre ajustados. (Ganti, 2020)

Debido a ello el precio de cierre ajustado es mejor para la aplicación de técnicas de análisis de series de tiempo, ya que permite comparar el comportamiento de una acción a lo largo del tiempo sin las distorsiones causadas por los eventos corporativos. Siendo la serie de tiempo más comúnmente utilizada en los estudios de los análisis de los precios de mercado la conformado por las rentabilidades calculadas a partir del precio de cierre ajustado.

## 2.2. Redes neuronales artificiales en la previsión de las series de tiempo

Este epígrafe está dividido en tres sub-epígrafes. En el primero se abordan los antecedentes del uso de redes neuronales artificiales para el trabajo con series de tiempo, más concretamente en la previsión. En el segundo y tercer sub-epígrafes se exponen el funcionamiento de dos de las estructuras de capas de RNA usadas en el presente trabajo, siendo estas las CNN y las LSTM.

### 2.2.1 Antecedentes del uso de redes neuronales artificiales en la previsión de series de tiempo

En Chollet y Allaire (2018) se plantea que el entorno de las RNA está conformado por la inteligencia artificial (en lo adelante IA), *machine learning* o aprendizaje automatizado (en lo adelante ML) y *deep learning* o aprendizaje profundo (en lo adelante DL), [Figura 1](#). Por lo que es de vital importancia conocer los aspectos de estos campos que se encuentran íntimamente relacionados con las RNA y que se exponen brevemente a continuación.

“Hacer que una máquina se comporte de tal manera que si un humano lo hiciera se le llamaría inteligente” (McCarthy et al. (2006), p.11) es la primera definición que se le dio al problema de IA. Con el objetivo de dar solución a este problema surgieron las primeras IA, las llamadas IA simbólicas.

Como se explica en Haykin (1998), Banda (2014) y Chollet y Allaire (2018), estas primeras IA, involucraban reglas codificadas creadas por los programadores. Con el

objetivo de lograr que estas reglas fueran aprendidas automáticamente por las máquinas al observar los datos surgió una nueva etapa dentro del desarrollo de las IA, la denominada ML. En esta nueva etapa se da pie al surgimiento de una nueva forma de programación, diferenciándose de la clásica, en que, en esta, los programadores introducen los datos y las respuestas esperadas a los mismos, y las computadoras son capaces de generar las reglas, *Figura 2*.

Por lo que se entiende que los modelos de ML tratan de encontrar representaciones apropiadas para sus datos de entrada: transformaciones de los datos que hacen que sea más susceptible a la tarea en cuestión. En DL, que es un subcampo específico de ML, estas representaciones de datos son modeladas a través de arquitecturas compuestas de capas sucesivas, las que son llamadas RNA (Chollet y Allaire, 2018).

Tras el estudio de lo expuesto en Haykin (1998), Larrañaga (2007), Banda (2014) y Chollet y Allaire (2018) sobre las RNA se puede afirmar que están inspiradas en el funcionamiento del cerebro humano, dichos textos confirman y concuerdan en que en una RNA se pueden diferenciar tres tipos de capas: de entrada, de salida y ocultas. Una capa de entrada está compuesta por neuronas que reciben los vectores de entradas. Una capa de salida se compone de neuronas que, durante el entrenamiento reciben los vectores de salidas y que luego generan la respuesta. Una capa oculta se encuentra conectada al entorno a través de las capas de entrada y salida, este tipo de capa oculta procesa la entrada recibida para obtener la salida correspondiente, *Figura 3*.

Una de las aplicaciones de las RNA es la previsión de series temporales. cuyo objetivo es predecir los valores futuros de las variables en función de sus observaciones pasadas. Como se expuso con anterioridad las series de tiempo financieras a menudo son no lineales, ruidosas, caóticas y no estacionarias, lo que las hace difíciles de modelar y pronosticar. Las RNA tienen la ventaja de poder capturar relaciones no lineales complejas y adaptarse a condiciones cambiantes sin requerir suposiciones previas sobre la distribución o estructura de datos.

La historia de las RNA en la previsión de series temporales financieras se remonta a finales de la década de 1980 y principios de la de 1990, cuando los investigadores comenzaron a explorar el potencial de las RNA como una alternativa a los métodos estadísticos tradicionales, como el modelo autorregresivo integrado de media móviles, más

conocido como ARIMA (por sus siglas en inglés Autoregressive Integrated Moving Average) y los modelos autorregresivos generalizados con heterocedasticidad condicional, más conocido como GARCH (por sus siglas en inglés Generalized Autoregressive Conditional Heteroskedasticity). Se demostró que las RNA tienen varias ventajas sobre estos métodos, como la capacidad de capturar relaciones no lineales y dinámicas, manejar datos ruidosos e incompletos y adaptarse a las condiciones cambiantes del mercado (B. Eddy Patuwo & Michael Y. Hu (1998)).

Sin embargo, las RNA también enfrentan algunas limitaciones y desafíos en el pronóstico de series temporales financieras, como la dificultad de elegir una arquitectura de red adecuada, un algoritmo de entrenamiento, una función de activación y variables de entrada; el riesgo de sobreajuste y problemas de generalización; la falta de interpretabilidad y transparencia; y el alto costo computacional y tiempo (Tealab (2018)).

Para superar estas limitaciones y desafíos, los investigadores han propuesto varias mejoras y extensiones de RNA para el pronóstico de series temporales financieras en las últimas décadas. Algunos de los principales desarrollos incluyen:

- El uso de modelos híbridos que combinan RNA con otras técnicas, como lógica difusa, algoritmos genéticos, análisis de ondículas, máquinas de vectores de soporte y aprendizaje profundo para mejorar el rendimiento y la solidez de las RNA (Wong y Guo (2010)).
- El uso de redes neuronales recurrentes (en lo adelante RNR) o bidireccional, que son un tipo especial de RNA que pueden procesar datos secuenciales y capturar dependencias temporales. Se ha demostrado que las RNR superan a las redes neuronales unidireccionales en series temporales complejas y no lineales (Guresen, Kayakutlu, y Daim (2011)).
- El uso de modelos de RNA más complejas mediante la combinación de distintas capas, como son las redes neuronales convolucionales (en lo adelante, CNN), las *long short-term memory* (en lo adelante, LSTM), las gated recurrent units (en lo adelante, GRU) se han aplicado a la previsión de series temporales financieras con resultados prometedores (Sezer, Gudelek, y Ozbayoglu (2020)).

La historia de las RNA en el pronóstico de series temporales financieras muestra que las mismas han ido evolucionando y mejorando con el tiempo para hacer frente a la

complejidad y la incertidumbre de los mercados financieros. Sin embargo, todavía persisten algunos de los desafíos y limitaciones señalados con anterioridad como el sobreajuste, la generalización, la interpretabilidad, la robustez y el costo computacional.

### 2.2.2 Redes neuronales convolucionales

El modelo de RNA que se usó en este trabajo está compuesto por varias capas siendo las más importantes la capa Conv1D, un tipo específico de CNN, y la capa LSTM, ambas mencionadas en el sub-epígrafe anterior cuando se listaron las estructuras de ANN que más se utilicen en la actualidad. Este sub-epígrafe se centra en la Capa Conv1D, por lo que se exploran los conceptos fundamentales para comprender el funcionamiento de esta, explicándose la convolución, las redes neuronales convolucionales y Conv1D y su uso para el análisis de series temporales. Se brinda una descripción general de la convolución y cómo se puede aplicar a los datos de series temporales. Luego, se analizan las CNN y su arquitectura, que les permite aprender características automáticamente a partir de datos de series temporales. Finalmente, se explica Conv1D, un tipo específico de capa de red neuronal convolucional que es particularmente eficaz para procesar datos de series temporales.

Como se expone en Siddiqui (2023) la convolución es una operación matemática que se usa comúnmente en el procesamiento de señales y el análisis de imágenes. Implica tomar dos funciones y producir una tercera función que representa cómo una de las funciones originales modifica a la otra. En el contexto de los datos de series temporales, la convolución se puede utilizar para extraer características de los datos aplicando un filtro a la serie temporal.

Además de extraer características de los datos de series temporales, la convolución también se puede utilizar para otras tareas, como la reducción de ruido, la detección de anomalías y la predicción. Por ejemplo, se puede entrenar una CNN para predecir valores futuros de una serie temporal aprendiendo los patrones subyacentes en los datos. En general, la convolución es una herramienta poderosa para analizar datos de series temporales y sus aplicaciones son numerosas (Siddiqui, 2023).

Las CNN fueron por primera vez introducidas en Lecun et al. (1998) son un tipo de modelo de aprendizaje profundo que se usa comúnmente para el análisis de imágenes. Sin

embargo, como se ha mencionado con anterioridad también se pueden utilizar para el análisis de series temporales, ya que son muy adecuados para aprender características a partir de datos que tienen una estructura espacial o temporal.

La arquitectura de una CNN consta de una o más capas convolucionales, que aplican filtros a los datos de entrada para extraer características. Cada filtro es un conjunto de pesos que se aprenden durante el proceso de entrenamiento. Al deslizar el filtro sobre los datos de entrada, la capa convolucional calcula un producto escalar en cada posición, produciendo un nuevo mapa de características (Lecun et al., 1998).

En un contexto de series de tiempo, una CNN puede aprender a extraer automáticamente características de los datos en diferentes escalas e intervalos de tiempo, lo que la convierte en una herramienta poderosa para el análisis de series de tiempo. Una ventaja clave de usar una CNN para el análisis de series de tiempo es que reduce la necesidad de ingeniería de características manual. En lugar de diseñar filtros a mano, CNN aprende a extraer automáticamente características de los datos, haciéndolo más flexible y adaptable a diferentes tipos de datos de series temporales.

En general, la arquitectura de una CNN le permite aprender características automáticamente a partir de datos de series temporales, lo que la convierte en una herramienta poderosa para el análisis de series temporales, siendo las Conv1D una de las estructuras de CNN más usadas para esta tarea.

Como se explica en Jing (2020) Conv1D es un tipo específico de capa de CNN que está diseñado para procesar datos unidimensionales, como datos de series temporales. Mientras que las CNN tradicionales están diseñadas para procesar datos bidimensionales, Conv1D está optimizado específicamente para datos unidimensionales, lo que lo hace más eficiente y eficaz para el análisis de series temporales.

La arquitectura de una capa Conv1D es similar a la de una CNN tradicional, pero con algunas diferencias clave. En lugar de usar filtros bidimensionales, Conv1D usa filtros unidimensionales, que se aplican a la serie temporal de entrada para extraer características. Las características que se extraen de la serie dependerán de las distintas configuraciones usadas para la configuración del filtro y la cantidad de filtros utilizados, siendo la fórmula

para calcular la cantidad de característica que extrae cada filtro la siguiente [Ecuación 1](#) (Jing, 2020):

$$L_{out} = \frac{L_{in} + 2 * padding - dilation * (kernel\_size - 1) - 1}{stride} + 1 \quad (1)$$

Donde:

*L<sub>out</sub>*: es la longitud de la salida del proceso de filtrado o la cantidad de características.

*L<sub>in</sub>*: la longitud del vector de entrada, correspondiendo en el análisis de series de tiempo a la cantidad de observaciones que contienen las muestras de la serie de tiempo que se pasan al filtro.

*kernel\_size*: es el tamaño del filtro, lo que define cuantas observaciones del vector de entradas se pasan al filtro cada vez. [Figura 4](#) representa como el tamaño del filtro puede afectar la longitud del vector de salida.

*stride*: representa la cantidad de pasos u observaciones en las que se mueve la selección de observaciones que se pasa al filtro. [Figura 5](#) representa como el parámetro stride puede afectar la longitud del vector de salida.

*dilation*: es la distancia de las observaciones que pasan al filtro. [Figura 6](#) representa como el parámetro dilation puede afectar la longitud del vector de salida.

*padding*: representa la cantidad de ceros que se añade a cada extremo del vector. [Figura 7](#) representa como el parámetro padding puede afectar la longitud del vector de salida.

En general, Conv1D es una herramienta poderosa para procesar datos de series temporales y sus ventajas incluyen la eficiencia computacional y la capacidad de capturar dependencias temporales en los datos. Sus casos de uso son numerosos y abarcan diferentes campos, lo que lo convierte en una herramienta valiosa para el análisis de series temporales.



### 2.2.3 Long short-term memory

En el presente sub-epígrafe se explica por qué las LSTM son una de las estructuras más usadas de RNA en la previsión de series de tiempo, partiendo de una breve explicación de las RNR y porque estas son de utilidad en la solución de problemas de previsión de series de tiempo, profundizando en por qué las LSTM se diferencian del resto de las RNN, y el funcionamiento de cada una de las capas que componen la estructura de una capa LSTM.

En Olah (2015) se explica que una RNN puede considerarse como copias múltiples de la misma red, *Figura 8*, expone que este aspecto revela que las RNR están íntimamente relacionadas con secuencias y listas, lo que hace que este tipo de RNA sea el que se use naturalmente para el trabajo con series de tiempo.

Las RNR convencionales presentan un problema en lo relacionado con la capacidad de retener la información, como se explica en Olah (2015), las RNN estándar se desempeñan con gran capacidad solo si, la información relevante para la situación actual es reciente, es decir donde la brecha entre la información relevante y el lugar en que se necesita es pequeña, *Figura 9*; expone además que a medida que crece la brecha, las RNN estándar son incapaces de acceder a la información relevante, *Figura 10*.

Como se ha mencionado con anterioridad las LSTM son un tipo de RNR que puede aprender dependencias a largo plazo en datos secuenciales. Estas fueron propuestas en Hochreiter (1997) y ha sido ampliamente utilizado para diversas tareas como el modelado del lenguaje, el reconocimiento de voz, la traducción automática, la descripción de imágenes y la previsión de series de tiempo.

La idea principal de LSTM es introducir una celda de memoria que pueda almacenar y actualizar información durante largos pasos de tiempo. La celda de memoria está controlada por tres puertas: una puerta de entrada, una puerta de olvido y una puerta de salida. Estas puertas son redes neuronales que aprenden a regular el flujo de información dentro y fuera de la célula *Figura 11*.

La puerta de entrada decide qué cantidad de la nueva entrada agregar al estado de la celda. La puerta de olvido decide qué parte del estado de celda anterior mantener o borrar. La puerta de salida decide qué parte del estado de celda actual se va a enviar a la siguiente

capa. Olah (2015) basado en lo expuesto en Hochreiter (1997), describe la operativa de las puertas en cuatro pasos:

1. Decidir qué información se olvida del estado de la celda a través de la puerta, forget gate layer  $f_t$ . Esta puerta ve a  $h_{t-1}$ , estado oculto del período de tiempo anterior, y  $x_t$ , entrada del instante de tiempo actual, y genera un número entre 0 (deshacerse) y 1 (mantener) para cada número en el estado de la celda  $C_{t-1}$ , [Figura 12](#), [Ecuación 2](#).

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (2)$$

2. Decidir qué nueva información se almacena en el estado de la celda. Para esto primero la puerta llamada input gate layer decide qué valores actualizar y luego, una capa tanh (tangente hiperbólica) crea un vector de nuevos valores candidatos ( $\tilde{C}_t$ ) que podrían agregarse al estado, [Figura 13](#), [Ecuación 3](#) y [Ecuación 4](#).

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (3)$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (4)$$

3. Se actualiza el estado de la celda anterior,  $C_{t-1}$  en el nuevo estado de la celda  $C_t$ . Se multiplica el estado anterior por  $f_t$ , olvidando lo necesario, luego se agrega  $i_t * \tilde{C}_t$ . Estos son los nuevos valores candidatos, escalados según cuánto se necesita actualizar cada valor de estado, [Figura 14](#), [Ecuación 5](#).

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (5)$$

4. Se genera una salida basándose en el estado de celda. Ejecutándose primero una capa sigmoidea que decide qué partes del estado de la celda es la salida; luego el estado de la celda pasa a través de una función tanh (escalando los valores entre -1 y 1) y se multiplican por la salida de la puerta, output gate, [Figura 15](#), [Ecuación 6](#) y [Ecuación 7](#).

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (6)$$

$$h_t = o_t * \tanh(C_t) \quad (7)$$

Las LSTM pueden aprender a capturar dependencias a largo plazo ajustando los valores de la puerta a través de la propagación inversa. Por ejemplo, si una determinada

entrada es relevante para una salida posterior, la puerta de entrada aprenderá a dejarla entrar, y la puerta olvidada aprenderá a conservarla en el estado de celda hasta que sea necesaria. Por el contrario, si una entrada es irrelevante u obsoleta, la puerta de entrada aprenderá a ignorarla, y la puerta olvidada aprenderá a borrarla del estado de la celda.

## 2.3 Composición de carteras

En este epígrafe, se expone el problema de encontrar la mejor composición de cartera posible y explicar de manera general la teoría detrás de los objetivos para solucionarlo. Además, se enumerarán las técnicas más utilizadas, diferenciando entre enfoques clásicos y enfoques inteligentes. Posteriormente, se explicará qué es la programación cuadrática y se mencionarán algunas técnicas dentro de esta disciplina de optimización matemática. Se mostrará cómo el problema de optimización de carteras se puede describir como un problema de programación cuadrática. Además, se proporcionará una breve explicación del Dual Active Set Method, una técnica ampliamente utilizada en esta disciplina y que será utilizada en los capítulos siguientes.

### 2.3.1 Problema y técnicas

Como se explica en Gunjan (2023) la optimización de cartera es el proceso de seleccionar la mejor combinación de activos para mantener en una cartera en función de objetivos predefinidos. Los objetivos pueden ser la maximización del rendimiento o la minimización del riesgo, o ambos. La optimización de la cartera implica encontrar las ponderaciones óptimas para cada activo de la cartera de manera que la cartera general cumpla con los objetivos deseados. Esto puede ser un problema desafiante debido a la gran cantidad de activos para elegir y las complejas relaciones entre ellos.

La optimización de la cartera es un proceso importante para los inversores, ya que les ayuda a minimizar el riesgo y maximizar el rendimiento de sus inversiones. Al seleccionar cuidadosamente los activos que mantendrán en su cartera, los inversores pueden lograr el nivel deseado de riesgo y rendimiento mientras diversifican sus inversiones para reducir el riesgo general. La optimización de la cartera es un mecanismo crucial que se utiliza para reducir el riesgo de la inversión.

Existen diversas técnicas que se pueden utilizar para resolver el problema de optimización de cartera. En Gunjan (2023) estas técnicas se encuentran clasificadas en dos categorías: enfoques clásicos y enfoques inteligentes. A continuación, se explica de manera general algunas de las técnicas pertenecientes a cada enfoque.

#### Enfoques clásicos:

- **Media-varianza:** Esta técnica, propuesta en Markowitz y Markowitz (1967), se basa en la idea de minimizar la varianza para una determinada rentabilidad esperada o maximizar la rentabilidad esperada para una determinada varianza. Es una técnica de programación cuadrática paramétrica (en lo adelante, PQP) que se puede utilizar para resolver problemas de optimización cuadrática que surgen en la optimización de carteras (Aijun Zhang & Chun-hung Li & Agus Sudjianto (2008)). El enfoque de la varianza media supone que los inversores tienen aversión al riesgo y prefieren carteras con una varianza más baja. La técnica consiste en construir una frontera de cartera que representa el conjunto de carteras que ofrecen el rendimiento esperado más alto para un nivel de riesgo dado. A continuación, se selecciona la cartera óptima de esta frontera en función de las preferencias de riesgo del inversor.
- **Varianza con asimetría:** esta técnica amplía el enfoque de media-varianza teniendo en cuenta la asimetría de la distribución. Fue propuesta en Samuelson (1970) y se puede utilizar cuando la función de distribución no es de naturaleza cuadrática. La asimetría mide la asimetría de una distribución y puede proporcionar información adicional sobre los riesgos y rendimientos potenciales de una cartera. Al incorporar la asimetría en el proceso de optimización de la cartera, los inversores pueden comprender mejor los posibles riesgos a la baja y tomar decisiones más informadas.
- **Valor en riesgo (VaR):** este enfoque estadístico mide la pérdida potencial de valor de una cartera durante un período definido para un intervalo de confianza dado. Fue introducido en la primera edición de Jorion (2007) en 1997 y requiere la determinación de tres parámetros: período de tiempo, nivel de confianza y unidad de valor en riesgo. El VaR proporciona una medida de la pérdida potencial máxima que podría ocurrir con una probabilidad dada en un horizonte de tiempo específico. Las instituciones financieras lo utilizan comúnmente para administrar su exposición al riesgo y cumplir con los requisitos reglamentarios.

- Valor en riesgo condicional (CVaR): este enfoque amplía el VaR teniendo en cuenta la pérdida esperada que excede el VaR. Fue introducido en Rockafellar y Uryasev (2002) y puede manejar pérdidas extremas mediante el uso de pesos dinámicos derivados de datos históricos. CVaR proporciona una medida de la pérdida esperada que podría ocurrir más allá del umbral de VaR. También se conoce como Expected Shortfall (ES) o Tail Value-at-Risk (TVaR) y se considera una medida de riesgo más coherente que el VaR.
- Desviación media-absoluta (MAD): esta técnica se puede emplear para problemas de selección de carteras de gran escala y muy diversificados. Fue introducido en Konno y Yamazaki (1991) y penaliza tanto las desviaciones positivas como las negativas. MAD proporciona una medida de la desviación absoluta promedio de los rendimientos de la cartera de su valor medio. Se considera más sólida que las medidas basadas en la varianza, ya que es menos sensible a los valores atípicos.
- Minimax: Esta técnica utiliza la rentabilidad mínima como medida de riesgo. Fue introducido en Cai et al. (2004) y tiene ciertas ventajas cuando los rendimientos no se distribuyen normalmente. Minimax proporciona una medida del peor de los casos para una cartera al minimizar la pérdida potencial máxima que podría ocurrir. Puede ser útil para los inversores que están particularmente preocupados por los riesgos a la baja.

#### Enfoques inteligentes:

- Redes bayesianas: estos modelos gráficos probabilísticos se pueden utilizar para modelar el riesgo y la rentabilidad. Fueron presentados en Shenoy y Shenoy (2000) y se pueden utilizar para visualizar la relación entre diferentes variables en un modelo. Las redes bayesianas proporcionan una forma de representar dependencias complejas entre variables utilizando gráficos acíclicos dirigidos (DAG). Se pueden usar para modelar relaciones inciertas entre variables y para hacer predicciones probabilísticas sobre eventos futuros. En el contexto de la gestión de carteras, las redes bayesianas se pueden utilizar para modelar las relaciones entre diferentes activos y hacer predicciones sobre sus rendimientos futuros en función de datos históricos y otra información relevante.

- Regresión de vectores de soporte (SVR): esta técnica de aprendizaje automático se puede utilizar para determinar la cantidad a comprar y vender. Fue introducido por Drucker et al. (1996) y tiene ciertas ventajas sobre las técnicas basadas en estadísticas, como su capacidad para aprender de datos históricos. SVR implica construir un hiperplano que separa puntos de datos con diferentes etiquetas mientras maximiza el margen entre ellos. Puede usarse para tareas de regresión donde el objetivo es predecir valores continuos en lugar de etiquetas discretas. En el contexto de la gestión de carteras, SVR se puede utilizar para predecir precios de activos futuros en función de datos históricos y otra información relevante.
- Redes neuronales artificiales: como se explicó con anterioridad estos modelos computacionales se pueden utilizar para resolver problemas computacionales y de aprendizaje complejos. En el contexto de la gestión de carteras, las redes neuronales se pueden utilizar para predecir futuros precios o rendimientos de activos en función de datos históricos y otra información relevante, que es para lo que se usan en el presente trabajo.
- Aprendizaje por refuerzo: este tipo de aprendizaje automático involucra a un agente o modelo que interactúa con su entorno para aprender de sus acciones. Fue presentado en Sutton y Barto (2018) y funciona para maximizar la recompensa al agente. El aprendizaje por refuerzo implica aprender a través de interacciones de prueba y error con un entorno. El agente realiza acciones en función de su estado actual y recibe recompensas o sanciones en función de los resultados de esas acciones. Con el tiempo, el agente aprende a realizar acciones que maximicen su recompensa acumulada. En el contexto de la gestión de carteras, el aprendizaje por refuerzo se puede utilizar para desarrollar estrategias comerciales que maximicen los rendimientos mientras se gestiona el riesgo.

### 2.3.2 Programación cuadrática

En este sub-epígrafe se explica que es la programación cuadrática. Cuáles son algunas de las técnicas que existen dentro de esta disciplina de la optimización matemática. Se expone además como el problema de optimización de carteras se puede describir como un problema de programación cuadrática y se expondrá de manera breve cómo funciona

una de las técnicas más usadas en esta disciplina, concretamente la denominada Dual Active Set Method, la cuál es usada en los capítulos posteriores.

La programación cuadrática se puede elegir entre las técnicas enumeradas en el sub-epígrafe anterior por varias razones. En primer lugar, es una técnica bien establecida que se ha utilizado ampliamente en la optimización de carteras. Puede manejar problemas de optimización complejos con múltiples restricciones y puede proporcionar una forma eficiente y efectiva de resolver el problema de optimización de cartera. Esto lo convierte en una herramienta útil para los inversores que buscan minimizar el riesgo mientras logran el nivel de rendimiento deseado. Finalmente, la programación cuadrática tiene una sólida base teórica y ha sido ampliamente estudiada en la literatura. Esto la convierte en una técnica confiable y bien entendida que se puede utilizar con confianza en la optimización de la cartera.

Existen diversas técnicas de programación cuadrática, entre las más utilizadas se encuentran:

- Interior Point: Este es un método de programación lineal o no lineal que logra la optimización al pasar por el centro del sólido definido por el problema en lugar de alrededor de su superficie. Un algoritmo de programación lineal de tiempo polinómico utilizando un método de punto interior fue encontrado por Karmarkar (1984).
- Active Set: Este es un algoritmo utilizado para identificar las restricciones activas en un conjunto de restricciones de desigualdad. Las restricciones activas se expresan entonces como restricciones de igualdad, transformando así un problema restringido por la desigualdad en un subproblema más simple restringido por la igualdad. El método de conjunto activo fue introducido por primera vez en un artículo de Beale (1959) y desarrollado por Fletcher (1971) y Bunch y Kaufman (1977).
- Dual Active Set: El método, como se expone en Goldfarb y Idnani (1982) y Goldfarb y Idnani (1983), es un algoritmo dual eficiente y numéricamente estable para la programación cuadrática definida positiva que aprovecha el hecho de que el mínimo sin restricciones de la función objetivo se puede usar como punto de partida.
- Augmented Lagrangian: Fue introducido independientemente en Magnus R. Hestenes (1969) y Powell (1969). Se utiliza para resolver problemas de optimización

restringidos agregando un término de penalización a la función objetivo que penaliza cualquier violación de las restricciones. El término de penalización suele ser un múltiplo de una medida de infracción de restricción, como la suma de infracciones de restricción al cuadrado.

- Conjugate Gradient: Este es un método iterativo para resolver sistemas de ecuaciones lineales con una matriz definida positiva simétrica. También se puede utilizar para resolver problemas de optimización sin restricciones al encontrar el mínimo de una función cuadrática. El método genera una secuencia de direcciones de búsqueda que se conjugan con respecto a la matriz que define el sistema de ecuaciones o función cuadrática. El método de gradiente conjugado fue introducido originalmente en un artículo de Magnus R. Hestenes y Stiefel (1952).
- Gradient Projection: El método de proyección de gradiente fue introducido en J. B. Rosen (1960) y J. Rosen (1961). Este es un método iterativo para resolver problemas de optimización restringidos proyectando el gradiente en la región factible en cada iteración. El gradiente proyectado se utiliza entonces como dirección de búsqueda, y se realiza una búsqueda de línea a lo largo de esta dirección para encontrar una nueva iteración que satisfaga las restricciones y reduzca la función objetivo.

De las técnicas señaladas con anterioridad se seleccionó el algoritmo Dual Active Set Method (en lo adelante, DASM) que como se mencionó con anterioridad fue introducido en Goldfarb y Idnani (1982) y Goldfarb y Idnani (1983), es un algoritmo de optimización para resolver problemas de programación cuadrática. El algoritmo predice el conjunto activo de restricciones que se satisfacen con igualdad en la solución del problema. Calcula una secuencia de soluciones óptimas de problemas QP que involucran algunas de las restricciones del problema original, denominada secuencia de puntos factibles duales.

A continuación, se presenta un ejemplo general de cómo podría funcionar el algoritmo DASM usando valores hipotéticos para un problema de optimización de cartera con 2 activos, el ejemplo fue construido a partir de lo expuesto en Goswami, Mondal, y Paruya (2012) y Walker (2014):



Bajo la suposición de que se trata de encontrar la mejor composición de una cartera en la que, por simplicidad, tenemos 2 activos, Se plantearía el problema cuadrático de la siguiente manera *Ecuación 8*:

$$\begin{aligned} \min Q(\vec{w}) &= \vec{w}^T C \vec{w} \\ \text{sujeto a:} & \\ w_1 + w_2 &= 1 & (8) \\ 0 \leq w_i &\leq 1 \\ w_1 \mathbb{E} + w_2 \mathbb{E} &\geq 0.005 \end{aligned}$$

Suponiendo que los cuales tienen unos rendimientos mensuales medios  $r = [0.02 \ 0.03]$  y matriz de covarianza  $C = \begin{bmatrix} 0.001 & 0.0008 \\ 0.0008 & 0.002 \end{bmatrix}$ . Se pueden construir los vectores y matrices necesarios para el algoritmo DASM de la siguiente manera:

- El vector de rentabilidad media mensual sería  $r = [0.02 \ 0.03]$ .
- La matriz de covarianza C se usaría como la matriz D en DASM.
- La restricción  $w_1 + w_2 = 1$  se puede escribir en forma de matriz como  $[1 \ 1] \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = 1$ . Esta sería la primera fila de la matriz A en DASM.
- El requisito de rentabilidad mínima  $w_1 \mathbb{E} + w_2 \mathbb{E} \geq 0.005$  puede escribirse en forma de matriz como  $[0.02 \ 0.03] \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \geq 0.005$ . Esta sería otra fila de la matriz A en DASM.
- Las restricciones  $0 \leq w_i \leq 1$  se pueden escribir en forma de matriz como  $[1 \ 0] \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \geq 0$  y  $[0 \ 1] \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \geq 0$  para límites inferiores y  $[-1 \ 0] \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \geq -1$  y  $[0 \ -1] \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \geq -1$  para límites superiores.

- La matriz A luciría así:  $A = \begin{bmatrix} 1 & 1 \\ 0.02 & 0.03 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}$

El vector  $b$  correspondiente sería  $[1 \ 0.005 \ 0 \ 0 \ -1 \ -1]$ . Luego podemos usar el algoritmo DASM para resolver este problema de programación cuadrática y determinar la asignación óptima de activos en nuestra cartera.

Paso 0: Encuentre el mínimo sin restricciones resolviendo el problema de programación cuadrática sin restricciones. Establecer el número de elementos del conjunto activo A (conjunto vacío) a cero.

Paso 1: Elija una restricción violada, si la hay. En este caso, supongamos que se viola la restricción  $w_1 + w_2 = 1$ .

Paso 2: Calcule las direcciones del paso primario y dual y la longitud del paso  $t = \min(t_1, t_2)$ . Supongamos que  $t = t_2$ .

Paso 3: Da un paso y actualiza el conjunto activo A y la solución (S) par (x, A). Como  $t = t_2$ , agregamos la p-ésima restricción (en este caso  $w_1 + w_2 = 1$ ) a  $\bar{N}$  y actualizamos H y  $N^*$  en [Ecuación 9](#).

$$\begin{aligned} N^* &= (\bar{N}^T Q^{-1} \bar{N}) \bar{N}^T Q^{-1} \\ H &= Q^{-1} (I - \bar{N} N^*) \end{aligned} \quad (9)$$

Donde:

$N^*$  es la pseudo-inversa o la inversa generalizada Moore-Penrose de  $\bar{N}$ .

$\bar{N}$  es la matriz de los vectores normales de las restricciones en el conjunto activo A.

H es el operador hessiano inverso reducido de Q.

Se repiten estos pasos de manera iterativa hasta que se satisfagan todas las restricciones y se haya determinado la asignación óptima de activos.

## 2.4 Datos

Este epígrafe se encuentra dividido en tres sub-epígrafes en los cuales se describe el proceso de obtención de los datos necesarios para la realización del resto del procedimiento. En el primero se detallan los pasos llevados a cabo para obtener los datos de las empresas y seleccionar aquellas con las que se trabajó en el resto del procedimiento. En el segundo sub-epígrafe se presenta una breve explicación de los indicadores computados que se utilizarán como variables de entrada, en conjunto con los valores históricos de rentabilidad de las empresas seleccionadas en el primer sub-epígrafe. El tercer sub-epígrafe expone el procedimiento llevado a cabo para la creación de los vectores de entrada y salida a partir de los datos resultantes del segundo sub-epígrafe.

### 2.4.1 Obtención de Datos

Una explicación más detallada, en lo respectivo a el código utilizado para la realización del procedimiento expuesto en el presente sub-epígrafe se encuentra en el Anexo 4 - *Obtención de Datos*.

Con el objetivo de ejemplificar como las redes neuronales artificiales y la programación cuadrática pueden ser usadas en una estrategia de gestión de cartera, se decidió en el presente trabajo utilizar datos de la bolsa del mercado español. Por lo que se decidió trabajar con la información correspondiente a las empresas que se encuentran en la lista de empresas cotizadas que se expone en «Empresas cotizadas» y puede ver en *Tabla 2*.

En la *Tabla 2* se recogen los datos de 119 empresas. Siendo los datos recogidos el nombre, ticker, sector y subsector, mercado, índice de cada una de las empresas y si fueron seleccionadas o no para realizar el resto del procedimiento después de la realización de los pasos expuestos en el presente sub-epígrafe.

Con el objetivo de obtener los datos de las empresas y analizarlas para seleccionar aquellas con las que se trabajó en el resto del procedimiento se usó como fuente Yahoo! Finance. A continuación, se expone el proceso llevado a cabo para la obtención y selección de los datos.

Se decidió descargar los datos mensuales de cada una de las empresas recogidas en *Tabla 2*. obteniéndose todos los datos comprendidos entre el 31 de enero del 2000 al 28 de febrero del 2023 de cada una de las entidades.

Tras obtener los datos se pasó evaluar la calidad de estos. Se comenzó la evaluación con un análisis exploratorio visual de los precios ajustados ya que, como se explicó con anterioridad estos son los ideales para usar en cualquier metodología de análisis históricos.

Durante el mencionado análisis exploratorio visual, se detectó que existían irregularidades en los precios ajustados de algunas de las series. Las irregularidades detectadas consistían en el incorrecto registro de los precios ajustados, así como errores en el cálculo de estos. Estos errores se detectaron fácilmente al observar en las gráficas de los valores del precio de cierre ajustado tendencia constante en periodos largos de tiempo, como se observa en *Gráfica 6*, lo que indica un registro erróneo de las variaciones de los

precios; así como cambios bruscos de hasta más de un 100% en los mismos en un solo período de tiempo, lo que puede indicar un mal cálculo en el precio ajustado, como se ve en [Gráfica 7](#), en este último caso se verificó con otras fuentes como Investing, para comprobar que de echo estaban mal computados los precios.

Dado el tiempo con el que se contaba para realizar el estudio expuesto en el procedimiento y la extensa cantidad de tiempo que requeriría la investigación a realizar para sustituir los valores erróneos en las series se decidió eliminar estas irregularidades mediante el uso solo de los valores posteriores a enero del 2005, que ya no presentaban inconsistencia en el cálculo del precio ajustado, posteriormente se eliminaron aquellas series que aún contenían valores faltantes y que presentaban irregularidad en el registro de las variaciones, para esto último se eliminaron aquellas series en las que las variaciones de los precios sin registrar sea en más de 10 observaciones.

Quedando tras los ajustes realizados 103 empresas, como se ve en la columna seleccionadas de la [Tabla 2](#), algunas de estas empresas cuentan con distintos números de observaciones, debido a que no todas existían o habían salido al mercado bursátil antes de enero del 2005.

Una vez seleccionadas las empresas con las que se trabajó se computaron las rentabilidades de estas a partir de los precios ajustados. Además de las rentabilidades correspondientes a las empresas seleccionadas se usaron las rentabilidades del precio de cierre ajustado del IBEX 35, además se computaron otras variables que sirven como indicadores del comportamiento de las rentabilidades, y la relación de estas con las del índice, en este caso las del IBEX 35. Entre estas variables se encuentran las volatilidades de las empresas y el índice, la correlación entre los valores de las series y el IBEX, y la beta de las empresas en relación con el IBEX.

#### 2.4.2 Indicadores

En el presente sub-epígrafe se expone una breve explicación de las variables computadas para usar como variables de entradas en conjunto con los valores históricos de las rentabilidades de las empresas. Una explicación más detallada, en lo respecto a el código utilizado para la realización del procedimiento expuesto en el presente sub-epígrafe se encuentra en Anexo 4 - [Indicadores](#).

### 2.4.2.1 Volatilidad

A partir de lo expuesto en Hargrave (2023) y Hayes (2023) la desviación estándar y la volatilidad son dos conceptos relacionados que miden cuánto fluctúa el precio de una acción u otro activo a lo largo del tiempo. La desviación estándar es un término estadístico que cuantifica la dispersión de un conjunto de puntos de datos alrededor de su valor medio. La volatilidad es un término financiero que describe el grado de variación en los rendimientos de un activo durante un período de tiempo determinado.

La desviación estándar y la volatilidad son importantes en el análisis del mercado de valores porque indican el riesgo y la incertidumbre asociados con la inversión en un activo en particular. Una desviación estándar o volatilidad alta significa que el precio del activo puede cambiar significativamente en cualquier dirección, lo que implica un mayor potencial de ganancias o pérdidas. Una desviación estándar o volatilidad baja significa que el precio del activo es relativamente estable y predecible, lo que implica un menor potencial de ganancias o pérdidas (Hayes, 2023).

Para calcular la volatilidad de una acción o índice se calcula la desviación estándar de las rentabilidades. Siendo por tanto los cálculos necesarios los que se muestran a continuación en la *Ecuación 10*:

$$R_i = \frac{P_i - P_{i-1}}{P_{i-1}}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (R_i - \bar{R})^2}{N}} \quad (10)$$

donde:

- $R_i$  es la rentabilidad de la acción en el periodo  $i$
- $P_i$  y  $P_{i-1}$  son los precios de una acción en periodos de tiempo  $i$  e  $i-1$ , respectivamente.
- $\sigma$  es la desviación estandar -  $N$  es el número de observaciones
- $\bar{R}$  es la rentabilidad media de la acción.

La desviación estándar y la volatilidad son herramientas útiles para que los inversores y analistas evalúen el equilibrio riesgo-recompensa de diferentes activos y carteras.

También pueden ayudar a comparar el rendimiento de diferentes activos y carteras a lo largo del tiempo y en diferentes condiciones de mercado.

#### 2.4.2.2 Correlación

Como se explica en Edwards (2022) la correlación es una medida estadística que determina cómo dos variables se mueven entre sí. En el análisis del mercado de valores, la correlación puede ayudar a comprender el comportamiento de diferentes acciones o indicadores del mercado a lo largo del tiempo. Tomando como ejemplo los datos que se usan en este trabajo, si los precios de una de las empresas seleccionadas tienden a subir y bajar junto con el IBEX 35, estos precios tienen una correlación positiva. Si al contrario los precios de la empresa tienden a subir cuando el indicador del IBEX 35 baja, tienen una correlación negativa. Un coeficiente de correlación de cero significa que no existe una relación lineal entre las variables, siendo en este caso los valores del IBEX 35 y los precios de una de las empresas determinadas.

Como se expone en Ross (2022) la correlación entre dos variables se calcula usando la siguiente ecuación, *Ecuación 11*:

$$\rho_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (11)$$

donde:

- $\rho_{xy}$  es el coeficiente de correlación
- $n$  es el número de observaciones
- $x_i$  y  $y_i$  son los valores de las dos variables para la  $i$  observación
- $\bar{x}$  y  $\bar{y}$  son las medias de las dos variables.

Como se explica también en Edwards (2022), el coeficiente de correlación se encuentra en un rango de -1 a 1, donde -1 indica una correlación negativa perfecta, 1 indica una correlación positiva perfecta, y 0 indica que no existe correlación alguna. Pudiéndose entender que, mientras más cercano se encuentre el coeficiente de correlación tanto a -1 como a 1, más fuerte es la relación lineal entre las variables analizadas.

Como ya se explicó con anterioridad el coeficiente de correlación, en el presente trabajo, se puede utilizar para analizar qué tan parecido se mueven las rentabilidades de

una empresa en comparación con las del IBEX 35. La correlación también se puede utilizar para diversificar una cartera eligiendo acciones que tengan una correlación baja o negativa entre sí, como se explica en Boyte-White (2022). Esto puede ayudar a reducir el riesgo general de la cartera, ya que las pérdidas de una acción pueden compensarse con las ganancias de otra. Sin embargo, la correlación no es constante y puede cambiar con el tiempo debido a varios factores, como las condiciones del mercado, los eventos económicos o las noticias de la empresa. Por lo tanto, es importante monitorear la correlación de acciones regularmente y ajustar la cartera en consecuencia (Boyte-White, 2022).

La correlación es una herramienta valiosa en el análisis del mercado de valores, pero no implica causalidad. Tener una correlación alta o baja entre dos variables no implica que una variable cause cambios en la otra. La correlación simplemente mide la fuerza y dirección de la relación lineal entre dos variables, sin considerar otros factores que puedan influir en ellas.

Como se expone también en Edwards (2022), la correlación guarda una íntima relación con la volatilidad del mercado y de las acciones, pudiéndose ver que, durante períodos de mayor volatilidad, como la crisis financiera de 2008, las acciones pueden tener una tendencia a estar más correlacionadas, incluso si se encuentran en diferentes sectores. Los mercados internacionales también pueden volverse altamente correlacionados durante tiempos de inestabilidad. Los inversores pueden querer incluir activos en sus carteras que tengan una baja correlación de mercado con los mercados de valores para ayudar a administrar su riesgo.

#### 2.4.2.3 Beta

Como se explica en Kenton (2022), Beta es una medida de cuán sensibles son los rendimientos de una acción a los cambios en los rendimientos del mercado. Se calcula como la pendiente de la línea de regresión que se ajusta a los rendimientos históricos de la acción y del mercado. Una beta de 1 significa que la acción se mueve en sincronía con el mercado, una beta superior a 1 significa que la acción es más volátil que el mercado y una beta inferior a 1 significa que la acción es menos volátil que el mercado.

Beta es importante en el análisis del mercado de valores porque, como se explica en Kenton (2022), ayuda a los inversores a evaluar el riesgo y el rendimiento de una cartera. Al conocer la versión beta de cada acción en una cartera, los inversores pueden estimar cuánto fluctuará la cartera con los movimientos del mercado y ajustar su asignación de activos en consecuencia. Por ejemplo, si un inversor quiere reducir el riesgo de su cartera, puede elegir acciones con valores beta bajos o negativos que tienden a moverse en dirección opuesta al mercado.

Como se explica en Monaghan (2019), Beta está relacionado con la correlación, pero no son lo mismo. Como se explicó con anterioridad la correlación es una medida de cuán linealmente relacionadas están dos variables, Beta, por otro lado, es una medida de cuán fuertemente relacionadas están dos variables, lo que indica cuánto cambia una variable cuando otra variable cambia en una unidad. Beta se puede calcular a partir de la correlación usando la siguiente fórmula, *Ecuación 12*:

$$\beta = \frac{\rho_{xy}\sigma_x}{\sigma_y} \quad (12)$$

donde:

- $\rho_{xy}$  es el coeficiente de correlación entre  $x$  y  $y$
- $\sigma_x$  es la volatilidad de  $x$
- $\sigma_y$  es la volatilidad de  $y$ .

### 2.4.3 Vectores

En el presente sub-epígrafe se expone el procedimiento llevado a cabo para la creación de los vectores de entrada y salida a partir de los datos resultantes del procedimiento expuesto en el sub-epígrafe anterior. Una explicación más detallada, en lo respectivo a el código utilizado para la realización del procedimiento expuesto en el presente sub-epígrafe se encuentra en Anexo 4 - *Vectores*.

La estructura del conjunto de vectores de entradas y salidas es de vital importancia en el modelado de técnicas de ML teniendo un impacto importante en su eficacia. El conjunto de vectores debe crearse de manera representativa del problema a resolver, por lo que los pasos descritos a continuación explican en detalle los aspectos del problema a dar



respuesta en el presente trabajo y como dar forma al conjunto de vectores de entradas y salidas para ello.

Como se ha mencionado con anterioridad el objetivo del presente trabajo es exponer un procedimiento para la utilización de modelos de RNA y programación cuadrática en una estrategia de inversión. El modelado atiende a la necesidad de obtener unas predicciones lo más acertada posible para posteriormente, basándose en las predicciones y en los datos históricos, hallar la composición de cartera idónea. Por lo que el problema a representar con los conjuntos de vectores de entradas y salidas es como explicar el comportamiento de la rentabilidad de una empresa en un instante de tiempo  $i + 1$  con los valores de varias variables en el instante de tiempo  $i$ .

Para la representación de este problema se crearon vectores tridimensionales, siguiendo lo expuesto en Chollet y Allaire (2018). Las dimensiones de estos vectores se explican de la siguiente manera:

- La primera dimensión está comprendida por el número de muestras obtenidas al seccionar las observaciones de las distintas series en vectores bidimensionales consecutivos.
- La segunda dimensión está comprendida por el número de observaciones, de las distintas series, recogido en cada vector bidimensional.
- La tercera dimensión es el número de series en cada vector bidimensional.

Por lo que para la correcta obtención de estas muestras se deben definir primero que series serán utilizadas para los vectores de entrada y salida. Las series utilizadas en los vectores de entrada fueron definidas en el epígrafe anterior, siendo estas: las rentabilidades históricas de la empresa y el IBEX, las volatilidades históricas de la empresa y el IBEX, la correlación histórica de la empresa y el IBEX, y el Beta histórico de la empresa y el IBEX. La serie utilizada para los vectores de salida es las rentabilidades históricas de la empresa.

Posteriormente se definió el horizonte temporal que se desea prever, este es un aspecto clave en la creación de los conjuntos de entradas y de salidas. El número de observaciones definido como horizonte temporal determina las observaciones los vectores de salida, en el presente trabajo se determinó como horizonte temporal una observación ya

que se desea predecir la rentabilidad del próximo mes de las distintas empresas seleccionadas.

Y el último aspecto a definir es cuantas observaciones debe observar el modelo para inferir la salida deseada. Esto define la cantidad de observaciones que se tomarán de cada serie de tiempo para conformar los vectores de entrada. Para determinar este aspecto se debe realizar un proceso iterativo probando distintas cantidades y evaluar los resultados obtenidos por los modelos que se entrenen con dichos. Para simplificar el proceso, en el presente trabajo se determinó probar distintos tamaños de entrada siendo estos 1, 2 y 3 observaciones. Probándose así de cierta manera como el tamaño de las entradas afecta la predicción obtenida.

Si tenemos una de serie para los vectores de entrada contiene unas 216 observaciones se puede calcular el número de muestras que se obtuvo de esta serie siguiendo la siguiente *Ecuación 13*:

$$m = n - (i - 1 + o) \quad (13)$$

donde:

- $m$  el número de muestras
- $n$  la cantidad de observaciones de las series
- $i$  y  $o$  el número de observaciones en los vectores de entrada y salida respectivamente.

En la *Tabla 3* se expone la cantidad de muestras obtenidas para los distintos tamaños de vectores de entrada planteados, para lo que se tuvo en cuenta los distintos números de observaciones con las que cuentan las 103 empresas seleccionadas. En la *Figura 16* se expone como lucen los vectores de entrada y salida, en el caso en el que el vector de entrada cuenta con 3 observaciones.

## 2.5 Modelado y entrenamiento

Este epígrafe se encuentra dividido en dos sub-epígrafes en los cuales se describe de forma breve los modelos que se construyeron y el procedimiento utilizado para entrenarlos. Explicándose en el primero de los sub-epígrafes las estructuras de los modelos utilizados,

mientras que en el segundo sub-epígrafe se explican las particularidades de la metodología de entrenamiento utilizada.

### 2.5.1 Modelado

Como se explicó con anterioridad los principales elementos de los modelos de redes neuronales artificiales utilizados son una capa CNN y una capa LSTM. Además de esto se utilizó una capa de entrada y una capa de salida, que son las encargadas de suministrar a los modelos la información de los vectores constituidos con anterioridad. Una explicación más detallada, en lo respecto a el código utilizado para la realización del procedimiento expuesto en el presente sub-epígrafe se encuentra en Anexo 4 - *Modelado*.

Dado que se definieron tres tamaños distintos de observaciones para tener en cuenta para realizar una predicción fue necesario construir tres estructuras de modelos distintas que se adaptasen a las dimensiones de los distintos vectores de entrada, las distintas estructuras se pueden observar en la *Figura 17*.

La primera diferencia notable entre las estructuras son las salidas de las capas de entrada, esta diferencia se debe a los tamaños de las muestras si se ha escogido usar 1,2 o 3 observaciones para construir el modelo. Como se observa el tamaño de la salida de la capa de entrada modifica por consiguiente el tamaño de las entradas y las salidas de la capa CNN.

Como se mencionó con anterioridad las variaciones en la segunda dimensión en las salidas de la capa CNN se pueden explicar por los distintos tamaños de los vectores de entrada. Pero como se observa el tamaño de la tercera dimensión de la salida de esta capa es igual en todas las estructuras, 64, lo que señala el número de filtros escogidos a utilizar, uno de los principales parámetros para tener en cuenta durante la configuración de estas capas. Esto último significa que las observaciones correspondientes a las 6 variables utilizadas fueron divididas en 64 variables que permiten al modelo una mejor comprensión de la relación entre las variables.

Otro aspecto que se modificó en la capa CNN de las estructuras fue la función de activación que de manera predetermina es la denominada ReLU (por sus siglas en inglés, Rectified Linear Unit) se cambió a Leaky ReLU debido a que como se explica en OmG (2021), ReLU es una función de activación no lineal que genera cero para entradas

negativas, lo que puede hacer que algunas neuronas dejen de aprender si muchas de sus entradas son negativas, ya que sus gradientes serán cero.

Dado lo explicado con anterioridad y que algunas de las variables que se usan en los valores de entradas poseen un alto número de observaciones negativas, como es el caso de las rentabilidades o la correlación de algunas de las series en ciertos periodos de tiempo el uso de la función de activación ReLU no parecía una buena opción. Por lo que se decidió utilizar como función de activación Leaky Relu que como se explica en OmG (2021), esta es una variante que permite un pequeño gradiente constante, distinto de cero, para entradas negativas. Esto significa que esta función de activación permite que algunas neuronas sigan aprendiendo de las entradas negativas.

En la [Figura 18](#) se observa el dominio de la función ReLU y Leaky ReLU lo que le permitirá una mejor comprensión de lo expuesto con anterioridad.

La capa CNN en todas las estructuras se encuentran enlazada a una capa LSTM, la cual en todos los casos contó con 64 neuronas. La salida de esta capa se encontraba enlazada con la capa de salida la cuál devuelve un solo valor.

Para concluir con la construcción de los modelos se determinó usar el error cuadrático medio (en lo adelante MSE, por sus siglas en inglés, Mean Squared Error) como la función utilizada para evaluar una solución candidata los resultados del modelo y el optimizador SGD (por sus siglas en inglés, Stochastic Gradient Descent) con un Alpha de 0.0005.

## 2.5.2 Entrenamiento

Una explicación más en detalle sobre el código utilizado durante el procedimiento expuesto en este sub-epígrafe se encuentra en Anexo 4 - [Entrenamiento](#).

El entrenamiento de algoritmos de *Machine Learning* en la previsión de series de tiempo tienen sus peculiaridades a como se entrenan modelos con el objetivo de solucionar otro tipo de problemas. Por lo que en este sub-epígrafe se cubre de manera breve la metodología de entrenamiento utilizada, que es la denominada *walk forward validation* o validación de avance.

Como ya se mencionó la validación de avance es un método utilizado para evaluar modelos de aprendizaje automático en datos de series temporales. Esto se debe a que

como se explica en Brownlee, (2019) proporciona la evaluación más realista de modelos de aprendizaje automático en datos de series temporales. Los métodos tradicionales de evaluación de modelos a partir del aprendizaje automático, como la validación cruzada de k-fold o la división en datos de entrenamiento y validación no funcionan en el caso de datos de series temporales porque ignoran los componentes temporales inherentes al problema. La validación Walk-forward tiene en cuenta estos componentes temporales y proporciona una evaluación más realista de cómo funcionará el modelo cuando se use operativamente.

Al evaluar un modelo, nos interesa su rendimiento del modelo en datos que no se usaron para entrenarlo. En el aprendizaje automático, se llaman datos no vistos o fuera de la muestra. Comúnmente para la resolución de otros problemas se dividen los datos en distintos subconjuntos: entrenamiento, prueba y validación, los que tiene como objetivo entrenar y validar el modelo. Con la metodología *walk forward validation* los datos se dividen por periodos de tiempo y se entrena y valida al modelo de forma consecutiva lo que permite evaluar como el modelo entiende la dependencia temporal de los datos.

Al dividir los datos por periodos de tiempo nos permite evaluar el funcionamiento real del modelo si se hubiese aplicado desde el primer periodo, así como analizar su comportamiento a lo largo de todos los periodos, observándose si su desempeño mejora o no.

De lo expuesto en el presente sub-epígrafe se entiende que los modelos fueron entrenados usando los conjuntos de muestras correspondientes, pasándose todas las muestras disponibles en un determinado periodo de tiempo antes de continuar con el siguiente periodo. Obteniéndose como resultado de lo anterior una predicción correspondiente a cada periodo de tiempo contemplado, a excepción de los dos primeros que se usarían para entrenar el modelo por primera vez, como se ve en el siguiente diagrama de la [Figura 19](#).

## 2.6 Resultado

Este epígrafe se encuentra dividido en dos sub-epígrafes en los cuales se describe de forma breve los resultados obtenidos. En el primero de los sub-epígrafe se describen y analizan las predicciones. En el segundo sub-epígrafe se exponen los resultados de aplicar

la programación cuadrática para la composición de cartera a partir de las predicciones obtenidas.

### 2.6.1 Predicciones

Una explicación más en detalle sobre el código utilizado durante el procedimiento expuesto en este sub-epígrafe se encuentra en Anexo 4 - *Predicciones*.

Como se explicó con anterioridad a la par que se entrenaba el modelo se obtuvieron las predicciones. Al igual que se realizó con los modelos de redes neuronales artificiales, las predicciones se computaron para las distintas observaciones usando la media aritmética de las observaciones. Se usó la media aritmética porque es una de las medidas que se usa con mayor frecuencia como indicador de posible comportamiento futuro en el estudio de las series de tiempo financieras.

Las predicciones se evaluarán mediante el computo del MSE y los valores reales y el  $R^2$  de los resultados obtenidos por los modelos de redes neuronales artificiales y las medias aritméticas.

Como se explica en Glen (2023) el MSE te dice qué tan cerca está una línea de regresión de un conjunto de puntos. Lo hace tomando las distancias desde los puntos hasta la línea de regresión (estas distancias son los "errores") y elevándolas al cuadrado. Elevar al cuadrado es necesario para eliminar cualquier signo negativo. También da más peso a las diferencias más grandes. Se llama el error cuadrático medio ya que estás encontrando el promedio de un conjunto de errores. Cuanto menor sea el MSE, mejor será el pronóstico, la muestra *Ecuación 14* como se calcula.

$$MSE = \frac{1}{n} * \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (14)$$

Donde:

- $n$ =número de observaciones
- $Y_i$ = valor real
- $\hat{Y}_i$ = valor esperado

Como se explica en Nandakumar (2020)  $R^2$  se utiliza comúnmente para explicar que tan bien lo hace un modelo en comparación con la media total de las observaciones, *Ecuación 15*:

$$R^2 = 1 - \frac{SSR}{SST}$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} \quad (15)$$

Donde:

- $\bar{Y}$  = media aritmética de todas las observaciones

Pero este puede ser un indicador injusto del desempeño de un modelo de regresión ya que se asume que se conocen todas las observaciones sobre las que se computa una media, y como ya se mencionó con anterioridad este no es el caso para los modelos de redes neuronales artificiales entrenados usando la metodología de *walk forward validation*. Debido a esto se modificó el cálculo del  $R^2$ , como se ha hecho en otras investigaciones como Shihao et al. (2018), para que el modelo con el que se comparan los resultados obtenidos por las RNA utilizadas sea el comprendido por las medias aritméticas de las observaciones anteriores a la que se predice.

A continuación, se describirán de manera breve los distintos resultados obtenidos por los distintos modelos construidos. Cabe destacar que, aunque se plantearon 3 modelos diversos de cada uno de ellos se construyeron 10, con el objetivo de estandarizar los resultados obtenidos, ya que el proceso de construcción y entrenamiento de redes neuronales contiene un factor aleatorio. Por lo que los resultados descritos a continuación son los resultados medios obtenidos por los diversos modelos construidos.

#### 2.6.1.1 Una observación

Los resultados obtenidos por aquellos modelos que fueron entrenados con vectores de entrada que contaban con una observación de cada serie mostraron, como se ve en la *Gráfica 8*, que en los primeros periodos los modelos presentaron mejores predicciones que las obtenidas por la media aritmética. Se puede observar que la efectividad de los modelos en comparación con las medias decae según el modelo va avanzando en el tiempo y aprendiendo de las nuevas observaciones. Se ve claramente también que en la mayoría de

los periodos el  $R^2$  de este modelo es negativo. Además, se observa un pico en el MSE del modelo a principios del 2020, lo que se entiende como una pérdida de efectividad del modelo, esta pérdida de efectividad del modelo pudiese estar relacionados con los movimientos bruscos de mercado consecuentes de las afectaciones económicas de la Covid-19.

El análisis anterior del comportamiento de los indicadores de estos modelos por periodo nos da una visión general de como se desempeñaron estos modelos, pero dado que para la composición de cartera es fundamental los resultados obtenidos en las empresas, a continuación, analizaremos el comportamiento observado en los resultados obtenidos por las 20 empresas que presentaron los mejores y peores resultados, basándose como criterio el  $R^2$  obtenido.

Observando los resultados de los indicadores expuestos en la [Tabla 4](#) aquellas empresas que presentaron un peor  $R^2$  presentan también un MSE bajo lo que indica que es menos probable que la composición de cartera se vea alterada por los resultados obtenidos por estas empresas. Por otro lado, entre las empresas que obtuvieron un mejor  $R^2$  se encuentran algunas que obtuvieron un MSE alto acompañado de un  $R^2$  mayor de un 5%. Esto indica que se pudieran generar diferencias entre las composiciones de las carteras debido a las diferencias en las predicciones y que además se trata de empresas que no tienen un buen MSE.

Los resultados descritos en el párrafo anterior son similares para los casos de los modelos construidos con dos y tres observaciones respectivamente.

#### 2.6.1.2 Dos observaciones

Los resultados obtenidos por aquellos modelos que fueron entrenados con vectores de entrada que contaban con dos observaciones de cada serie se encontró, como se ve en la [Gráfica 9](#), que en los primeros periodos los modelos presentaron mejores predicciones que las obtenidas por la media aritmética. Se puede observar que la efectividad de los modelos en comparación con las medias decae según el modelo va avanzando en el tiempo, pero decaen a un ritmo más lento que aquellos modelos entrenados con vectores de entrada con una observación. Se ve claramente también que el  $R^2$  de estos modelos tiene una menor variación que el  $R^2$  de los modelos analizados con anterioridad, viéndose que para estos



modelos el  $R^2$  es positivo en la mayoría de los periodos. Además, al igual que en el caso de los modelos analizados con anterioridad también se observa un pico en el MSE del modelo a principios del 2020.

### 2.6.1.3 Tres observaciones

Los resultados obtenidos por aquellos modelos que fueron entrenados con vectores de entrada que contaban con tres observaciones de cada serie se encontró, como se ve en la [Gráfica 10](#), que en los primeros periodos los modelos presentaron mejores predicciones que las obtenidas por la media aritmética. Se puede observar que la efectividad de los modelos en comparación con las medias decae según el modelo va avanzando en el tiempo, pero decaen a un ritmo más lento que aquellos modelos entrenados con vectores de entrada con una observación. Se ve claramente que el  $R^2$  de estos modelos tiene una mayor variación que el  $R^2$  de los modelos analizados con anterioridad, observándose como esta variación disminuye para aquellas predicciones después del 2015. Estos modelos, al igual que los primeros presentaron un  $R^2$  negativo en la mayoría de los periodos. Además, al igual que en los casos anteriores también se observa un pico en el MSE del modelo a principios del 2020.

## 2.6.2 Composición de carteras

Una explicación más en detalle sobre el código utilizado durante el procedimiento expuesto en este sub-epígrafe se encuentra en Anexo 4 - [Composición de carteras](#).

En el presente sub-epígrafe se describen los resultados obtenidos tras aplicar programación cuadrática para determinar la composición de la cartera. Esto al igual que las predicciones se realizó periodo a periodo con el objetivo de emular una situación real en la que se aplicaran las técnicas en su conjunto. Por lo que el presente análisis se enfoca en el comportamiento observado al usar los diversos modelos y la comparación de estos resultados con los obtenidos con el uso de las medias.

Como se observa en la [Gráfica 11](#) las carteras conformadas a partir de las predicciones obtenidas por los modelos de redes neuronales que contaban con una observación obtuvieron en lo general mejores resultados que las carteras compuestas a partir de las predicciones usando la media. Se observa que ambos grupos de cartera presentaron una rentabilidad menor que el índice, IBEX, en el periodo comprendido entre el 2009 y el 2016.

Al realizar el análisis del comportamiento de las rentabilidades obtenidas por los modelos con dos observaciones de entrada, *Gráfica 12*, se observa: el comportamiento de las rentabilidades obtenidas por los distintos modelos varía menos que los analizados con anterioridad; en este caso y contrario al caso anterior las rentabilidades se mantienen similares en el periodo comprendido entre 2009 y 2016; y aunque el resultado final dista del resultado obtenido por las medias, es inferior al obtenido por los modelos anteriores, esto último se debe a que la evaluación de los modelos en este caso comienza en un periodo anterior a las de los modelos analizados con anterioridad.

Al observar los resultados obtenidos por los últimos modelos, *Gráfica 13*, se observa: una distribución de las rentabilidades superior a aquellos entrenados con dos observaciones pero inferior a los que se entrenaron con una observación; se observa que las rentabilidades comienzan a superar las del índice después del 2013 en lugar del 2016 como en años anteriores; y además se observa que las rentabilidades de los modelos de RNA son superiores a los de las medias y constituyen también las rentabilidades máximas obtenidas entre las distintas estructuras de modelos de RNA.

## 3. Conclusiones

Durante el desarrollo de este trabajo, se ha abordado la aplicación de redes neuronales artificiales y programación cuadrática en la gestión de carteras financieras. A través de una cuidadosa caracterización de las series temporales financieras, se pudo comprender la importancia de analizar sus características y patrones para realizar pronósticos más precisos.

Se probaron distintas configuraciones de modelos de redes neuronales artificiales, conformadas por la combinación de capas convolucionales y LSTM, que diferían en la cantidad de observaciones históricas que usarían como entradas antes de realizarse una predicción. Las predicciones obtenidas a partir de los modelos antes mencionados se compararon con predicciones obtenidas mediante el uso de la media aritmética, que es uno de los indicadores que más comúnmente se utiliza. Como resultado de la comparación antes expuestas se obtuvo que los modelos en dependencia de la cantidad de observaciones que usaron como entradas: 1, 2 o 3; obtuvieron un R2 de: -0.00287, 0.0611 y 0.0179 respectivamente.

Las predicciones obtenidas, tanto con los modelos de RNA como con las medias aritméticas en conjunto con los comportamientos históricos fueron usados para, mediante el uso programación cuadrática, buscar la composición de carteras de menor riesgo. Tras la realización de una simulación de gestión de carteras se obtuvo que las carteras conformadas a partir de las predicciones de los modelos de RNA obtuvieron al final del periodo estudiado en comparación con las conformadas por las predicciones usando la media aritmética unas rentabilidades: 5.63% superior, para los modelos que usaron 1 observación como entrada; un 35.67% superior para los que usaron 2; y un 25.51% para los que usaron 3. Además, se observó que las carteras conformadas con los modelos de RNA obtuvieron unas rentabilidades superiores al índice, IBEX, en un 40.86%, 39.78% y 60.54%, para los modelos que usaron 1, 2 y 3 observaciones como entradas respectivamente.

Los resultados antes expuestos demuestran que el uso combinado de estas herramientas, las RNA y la programación cuadrática, pueden ofrecer una ventaja competitiva significativa en la gestión de sus activos financieros a empresas y organizaciones, permitiendo una toma de decisiones más efectiva, optimizando la composición de carteras y maximizando los rendimientos.

No obstante, es importante destacar que los resultados expuestos en el presente trabajo necesitan un estudio más profundo para analizar, entre otros aspectos, el peso que tienen los resultados de las predicciones de las distintas empresas en la composición de carteras. Por esta razón se considera el presente trabajo como el comienzo de una investigación más exhaustiva en la cual: se deben obtener datos de mayor calidad y se contrastará el uso de diversas técnicas, tanto para la obtención de las predicciones como para hallar la composición de cartera adecuada.

# Bibliografía

Yahoo Finance. <https://finance.yahoo.com/>.

Investing. <https://www.investing.com/>.

Aijun Zhang & Chun-hung Li & Agus Sudjianto, Zhi-li Wu &. 2008. «Trace solution paths for SVMs via parametric quadratic programming». Researchgate. 2008. [https://www.researchgate.net/publication/228577955\\_Trace\\_solution\\_paths\\_for\\_SVMs\\_via\\_parametric\\_quadratic\\_programming](https://www.researchgate.net/publication/228577955_Trace_solution_paths_for_SVMs_via_parametric_quadratic_programming).

Anderson, D. R., D. J. Sweeney, T. A. Williams, D. J. Camm, y J. J Cochran. 2017. *Statistics for business & economics*. Boston: Cengage Learning.

B. Eddy Patuwo & Michael Y. Hu, Guoqiang Zhang &. 1998. «Forecasting with artificial neural networks:: The state of the art». *International Journal of Forecasting* 14 (1): 35-62. [https://doi.org/10.1016/S0169-2070\(97\)00044-7](https://doi.org/10.1016/S0169-2070(97)00044-7).

Banda, Hugo. 2014. *Inteligencia Artificial: Principios y Aplicaciones*. Quito, Ecuador: Escuela Politécnica Nacional.

Barone, A. 2022. «Opening Price: Definition, Example, Trading Strategies». 2022. <https://www.investopedia.com/terms/o/openingprice.asp>.

Beale, EML. 1959. «On quadratic programming». *Naval Research Logistics Quarterly* 6 (3): 227-43.

BME. s. f. «¿Qué es BME?» Accedido 24 de abril de 2023. <https://www.bolsasymercados.es/esp/Sobre-BME/Que-es>.

Boyte-White, C. 2022. «How Does Correlation Affect the Stock Market?» 2022. <https://www.investopedia.com/ask/answers/021716/how-does-correlation-affect-stock-market.asp>.

Brownlee, 2019. «How To Backtest Machine Learning Models for Time Series Forecasting». *Machine Learning Mastery| Time series*. Recuperado de: [How To Backtest Machine Learning Models for Time Series Forecasting - MachineLearningMastery.com](https://machinelearningmastery.com/how-to-backtest-machine-learning-models-for-time-series-forecasting/)

- Bunch, James R, y Linda Kaufman. 1977. «Some stable methods for calculating inertia and solving symmetric linear systems». *Mathematics of computation* 31 (137): 163-79.
- Cai, Xiaoqiang, Kok Lay Teo, XQ Yang, y Xun Yu Zhou. 2004. «Minimax portfolio optimization: empirical numerical study». *Journal of the Operational Research Society* 55 (1): 65-72.
- Castillo, R. A., y R. Varela. 2010. *ECONOMETRÍA PRÁCTICA: Fundamentos de Series de Tiempo*. México: Universidad Autónoma de Baja California.
- Chen, J. 2022. «Today's High». 2022. <https://www.investopedia.com/terms/t/todayshigh.asp>.
- Chirinos, S. 2018. «Series cronológicas». <https://www.slideshare.net/SuedimarChirinos/series-cronologicas-119058959>. 2018.
- Chollet, F., y J. J. Allaire. 2018. *Deep Learning with R*. Manning Publications. <https://books.google.es/books?id=xnIRtAEACAAJ>.
- CNMV. «Glosario Financiero: Acción». Accedido 24 de abril de 2023. <https://cnmv.es/Portal/Inversor/Glosario.aspx?id=0&letra=A&idIng=1>.
- CNMV. «Glosario Financiero: Bolsa de valores». Accedido 24 de abril de 2023. <https://cnmv.es/Portal/Inversor/Glosario.aspx?id=0&letra=B&idIng=1>.
- CNMV. «Glosario Financiero: Servicio de Interconexión Bursátil Español, SIBE». Accedido 24 de abril de 2023. <https://cnmv.es/Portal/Inversor/Glosario.aspx?id=0&letra=S&idIng=1>.
- Dodge, Y. 2008. «Time Series». En *The Concise Encyclopedia of Statistics*, 536-39. New York, NY: Springer New York. [https://doi.org/10.1007/978-0-387-32833-1\\_401](https://doi.org/10.1007/978-0-387-32833-1_401).
- Downey, L. 2022. «Today's Low». 2022. <https://www.investopedia.com/terms/t/todayslow.asp>.
- Drucker, Harris, Christopher Burges, Linda Kaufman, Alex Smola, y Vladimir Vapnik. 1996. «Linear support vector regression machines». *Advances in neural information processing systems* 9 (9): 155-61.
- Edwards, J. 2022. «Why Market Correlation Matters?» 2022. <https://www.investopedia.com/articles/financial-advisors/022516/4-reasons-why-market-correlation-matters.asp>.

- «Empresas cotizadas». s. f. BME Exchange. Accedido 21 de mayo de 2023. <https://www.bolsasymercados.es/bme-exchange/es/Mercados-y-Cotizaciones/Acciones/Mercado-Continuo/Empresas-Cotizadas>.
- Espallargas, S. D., y M. V. Solís. 2012. *Econometría y series temporales: aplicaciones*. La Habana: Editorial Félix Varela.
- Fletcher, Roger. 1971. «A general quadratic programming algorithm». *IMA Journal of Applied Mathematics* 7 (1): 76-91.
- Ganti, A. 2020. «Adjusted Closing Price». 2020. [https://www.investopedia.com/terms/a/adjusted\\_closing\\_price.asp](https://www.investopedia.com/terms/a/adjusted_closing_price.asp).
- Glen, S. (2023). «Mean Squared Error: Definition and Example» From StatisticsHowTo.com: Elementary Statistics for the rest of us! <https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/mean-squared-error/>
- Goldfarb, Donald, y Ashok U. Idnani. 1982. «Dual and primal-dual methods for solving strictly convex quadratic programs». En *Numerical Analysis*, editado por J. P. Hennart, 226-39. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Goldfarb, Donald, y Ashok U. Idnani. 1983. «A numerically stable dual method for solving strictly convex quadratic programs». *Mathematical Programming* 27: 1-33.
- Goswami, Nababithi, Supriyo K. Mondal, y Swapan Paruya. 2012. «A Comparative Study of Dual Active-Set and Primal-Dual Interior-Point Method». *IFAC Proceedings Volumes* 45 (15): 620-25. <https://doi.org/https://doi.org/10.3182/20120710-4-SG-2026.00029>.
- Gunjan, Siddhartha, Abhishek & Bhattacharyya. 2023. «A brief review of portfolio optimization techniques». *Artificial Intelligence Review* 56 (5): 3847-86. <https://doi.org/10.1007/s10462-022-10273-7>.
- Guresen, Erkam, Gulgun Kayakutlu, y Tugrul U. Daim. 2011. «Using artificial neural network models in stock market index prediction». *Expert Systems with Applications* 38 (8): 10389-97. <https://doi.org/https://doi.org/10.1016/j.eswa.2011.02.068>.
- Hargrave, M. 2023. «Standard Deviation Formula and Uses vs. Variance». 2023. <https://www.investopedia.com/terms/s/standarddeviation.asp>.

- Hayes, A. 2021. «What Is Closing Price? Definition, How It's Used, and Example». 2021. <https://www.investopedia.com/terms/c/closingprice.asp>.
- Hayes, A. 2023. «Volatility: Meaning In Finance and How it Works with Stocks». 2023. <https://www.investopedia.com/terms/v/volatility.asp>.
- Haykin, Simon. 1998. Neural networks: a comprehensive foundation. Prentice Hall PTR.
- Hestenes, Magnus R. 1969. «Multiplier and gradient methods». *Journal of optimization theory and applications* 4 (5): 303-20.
- Hestenes, Magnus R., y Eduard Stiefel. 1952. «Methods of conjugate gradients for solving linear systems». *Journal of research of the National Bureau of Standards* 49: 409-35.
- Hochreiter, Jürgen, Sepp & Schmidhuber. 1997. «Long Short-Term Memory». *Neural Computation* 9 (8): 1735-80. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- IBM. 2021. «Characteristics of time series». <https://www.ibm.com/docs/en/spss-modeler/saas?topic=data-characteristics-time-series>. 2021.
- Jing, Hong. 2020. «How Convolutional Layers Work in Deep Learning Neural Networks?» Jingles, Github Blog. 2020. <https://jinglescode.github.io/2020/11/01/how-convolutional-layers-work-deep-learning-neural-networks/>.
- Jorion, Philippe. 2007. Value at risk: the new benchmark for managing financial risk. The McGraw-Hill Companies, Inc.
- Karmarkar, Narendra. 1984. «A new polynomial-time algorithm for linear programming». En *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, 302-11.
- Kenton, W. 2022. «Beta: Definition, Calculation, and Explanation for Investors». 2022. <https://www.investopedia.com/terms/b/beta.asp>.
- Kocenda, E., y A. Cerný. 2017. *Elements of Time Series Econometrics: An Applied Approach*. Prague: Karolinum Press.



- Konno, Hiroshi, y Hiroaki Yamazaki. 1991. «Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market». *Management science* 37 (5): 519-31.
- Larrañaga, Iñaki & Moujahid, Pedro & Inza. 2007. «Tema 14. Redes Neuronales». Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad del País Vasco–Euskal Herriko Unibertsitatea. 2007. <http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/t14-neuronales.pdf>.
- Lecun, Y., L. Bottou, Y. Bengio, y P. Haffner. 1998. «Gradient-based learning applied to document recognition». *Proceedings of the IEEE* 86 (11): 2278-2324. <https://doi.org/10.1109/5.726791>.
- Markowitz, Harry M, y Harry M Markowitz. 1967. Portfolio selection: efficient diversification of investments. J. Wiley.
- McCarthy, John, Marvin L. Minsky, Nathaniel Rochester, y Claude E. Shannon. 2006. «A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955». *AI Magazine* 27 (4): 12. <https://doi.org/10.1609/aimag.v27i4.1904>.
- Mitchell, C. 2020. «Market Price: Definition, Meaning, How To Determine, and Example». 2020. <https://www.investopedia.com/terms/m/market-price.asp>.
- Monaghan, B. 2019. «Correlation vs. Beta: What is The Difference and Why Does It Matter?» 2019. [https://www.mackenzieinvestments.com/content/dam/final/corporate/mackenzie/docs/investment-teams/multi-asset-team/en/Correlation%20vs.%20Beta\\_%20What%20is%20The%20Difference%20and%20Why%20Does%20It%20Matter\\_%20\\_%20Mackenzie%20Investments.pdf](https://www.mackenzieinvestments.com/content/dam/final/corporate/mackenzie/docs/investment-teams/multi-asset-team/en/Correlation%20vs.%20Beta_%20What%20is%20The%20Difference%20and%20Why%20Does%20It%20Matter_%20_%20Mackenzie%20Investments.pdf).
- Nandakumar, S. (2020). «How can r-squared be negative when the correlation between prediction and truth is positive?» *Stack|overflow*. Recuperado de: <https://stackoverflow.com/a/63311778/12660035>
- Olah, Christopher. 2015. «Understanding LSTM networks». Colah's blog. 2015. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>.

- OmG, 2021. «Difference between ReLU, ELU and Leaky ReLU. Their pros and cons majorly» *DataScience StackExchange*.  
<https://datascience.stackexchange.com/a/102485>
- Pinset, W. 2021. «Understanding Stock Prices and Values». 2021.  
<https://www.investopedia.com/articles/stocks/08/stock-prices-fool.asp>.
- Powell, Michael JD. 1969. «A method for nonlinear constraints in minimization problems». *Optimization*, 283-98.
- Rallabandi, S. (2023). «Activation functions ReLU vs. Leaky ReLU». Medium | MLearning AI. Recuperado de: <https://medium.com/mlearning-ai/activation-functions-relu-vs-leaky-relu-b8272dc0b1be>
- Rockafellar, R Tyrrell, y Stanislav Uryasev. 2002. «Conditional value-at-risk for general loss distributions». *Journal of banking & finance* 26 (7): 1443-71.
- Rosen, JB. 1961. «The gradient projection method for nonlinear programming. Part II. Nonlinear constraints». *Journal of the Society for Industrial and Applied Mathematics* 9 (4): 514-32.
- Rosen, Jo Bo. 1960. «The gradient projection method for nonlinear programming. Part I. Linear constraints». *Journal of the society for industrial and applied mathematics* 8 (1): 181-217.
- Ross, S. 2022. «How Do I Calculate Correlation Between Market Indicators and Specific Stocks?» 2022. <https://www.investopedia.com/ask/answers/032315/how-do-i-calculate-correlation-between-market-indicators-and-specific-stocks.asp>.
- Ruiz, M. C. 2011. «Tema 5: Procesos Estocásticos». [http://www.dmae.upct.es/~mcruiz/Telem06/Teoria/apuntes\\_procesos.pdf](http://www.dmae.upct.es/~mcruiz/Telem06/Teoria/apuntes_procesos.pdf); Departamento de Matemática y Estadística. Universidad Politécnica de Cartagena. 2011.
- Ryan, Jeffrey A., y Joshua M. Ulrich. 2023. *quantmod: Quantitative Financial Modelling Framework*. <https://CRAN.R-project.org/package=quantmod>.
- Samuelson, Paul A. 1970. «The fundamental approximation theorem of portfolio analysis in terms of means, variances and higher moments». *The Review of Economic Studies* 37 (4): 537-42.

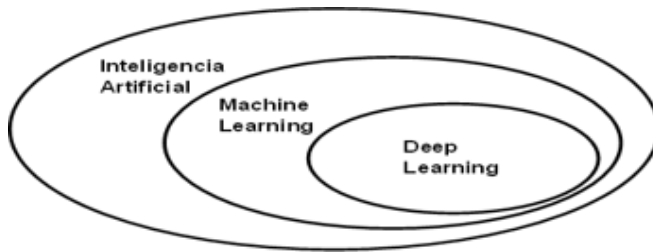
- Sezer, Omer Berat, Mehmet Ugur Gudelek, y Ahmet Murat Ozbayoglu. 2020. «Financial time series forecasting with deep learning : A systematic literature review: 2005–2019». *Applied Soft Computing* 90: 106181. <https://doi.org/https://doi.org/10.1016/j.asoc.2020.106181>.
- Shenoy, Catherine, y Prakash P Shenoy. 2000. «Bayesian network models of portfolio risk and return». En. The MIT Press.
- Shihao, G., Bryan, K., y Dacheng, Xiu. (2018) «Empirical Asset Pricing via Machine Learning». National Bureau of Economic Research. Working Paper | Working Paper Series, num: 25398. Recuperado de: <http://dx.doi.org/10.3386/w25398>
- Siddiqui, J. Rafid. 2023. «Why Convolve? Understanding Convolution and Feature Extraction in Deep Networks». Medium, Towards Data Science. 2023. <https://towardsdatascience.com/why-convolve-understanding-convolution-and-feature-extraction-in-deep-networks-ee45d1fdd17c>.
- Sutton, Richard S, y Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
- Tealab, Ahmed. 2018. «Time series forecasting using artificial neural networks methodologies: A systematic review». *Future Computing and Informatics Journal* 3 (2): 334-40. <https://doi.org/https://doi.org/10.1016/j.fcij.2018.10.003>.
- Team, CFI. 2023. «What is Stock Price?» 2023. <https://corporatefinanceinstitute.com/resources/capital-markets/stock-price/>.
- Team, The Investopedia. 2022. «Intrinsic Value Defined and How It's Determined in Investing and Business». 2022. <https://www.investopedia.com/terms/i/intrinsicvalue.asp>.
- Villagarcía, T. 2006. «Series Temporales». [https://halweb.uc3m.es/fjnm/estind/doc\\_grupo1/archivos/Apuntes%20de%20series.pdf](https://halweb.uc3m.es/fjnm/estind/doc_grupo1/archivos/Apuntes%20de%20series.pdf). 2006.
- Villavicencio, J. 2010. «Introducción a las series de tiempo». <http://www.estadisticas.gobierno.pr/iepr/LinkClick.aspx>; Instituto de estadística de Puerto Rico. 2010.
- Walker, Ryan. 2014. «Solving Quadratic Programs with R's quadprog package». rwalk. 2014. <https://rwalk.xyz/solving-quadratic-programs-with-rs-quadprog-package/>.

Wong, W. K., y Z. X. Guo. 2010. «A hybrid intelligent model for medium-term sales forecasting in fashion retail supply chains using extreme learning machine and harmony search algorithm». *International Journal of Production Economics* 128 (2): 614-24. <https://ideas.repec.org/a/eee/proeco/v128y2010i2p614-624.html>.

# Anexos

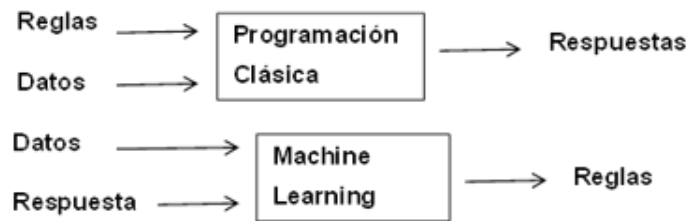
## Anexo.1 Figuras

Figura 1: Relación entre IA-ML-DL



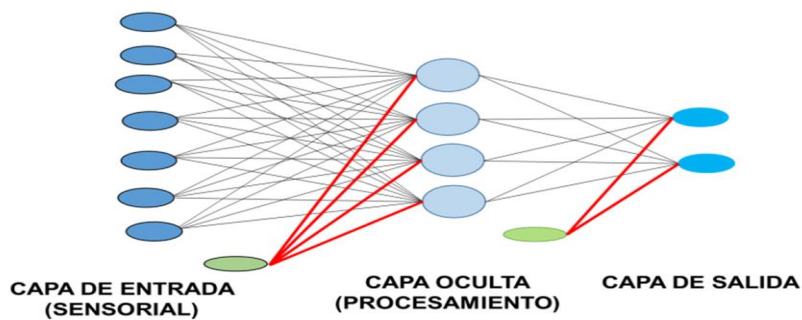
Tomada de: Deep learning with R de Chollet y Allaire (2018).

Figura 2: Programación clásica y machine learning



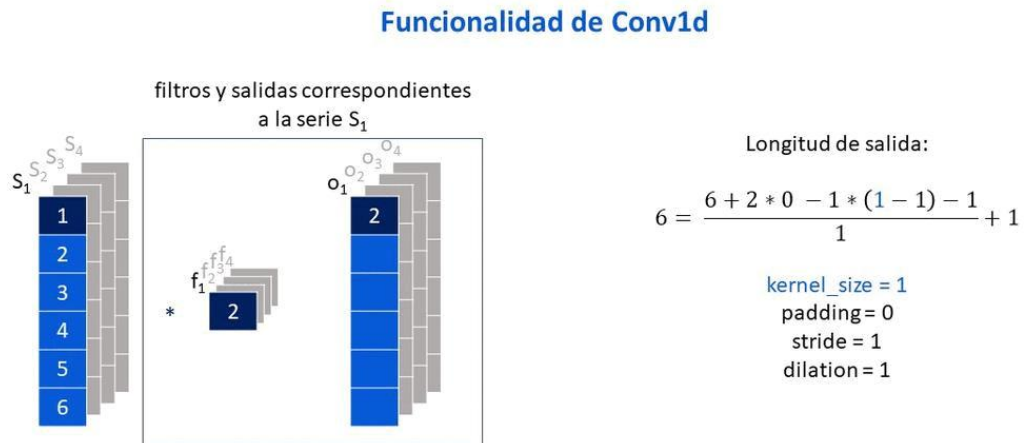
Tomada de: Deep learning with R de Chollet y Allaire (2018).

Figura 3: Estructura básica de una red neuronal artificial



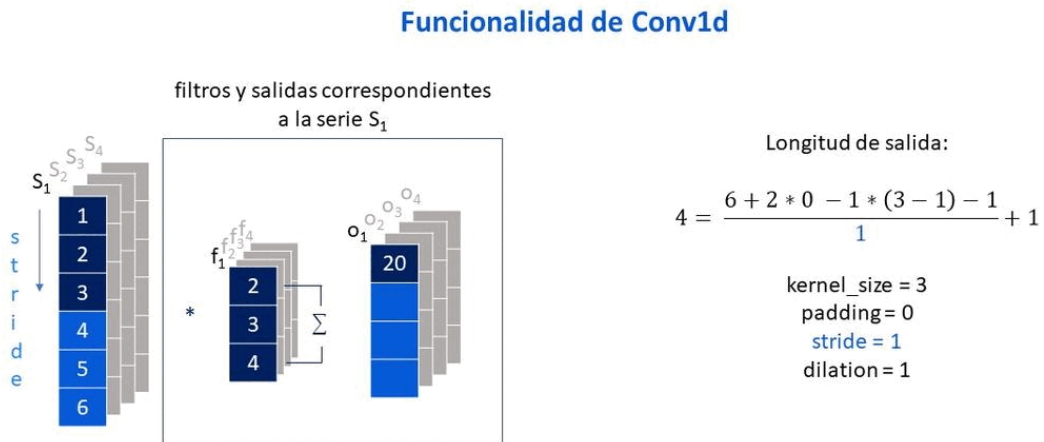
Tomada de: Tema 14: redes neuronales de Larrañaga (2007).

Figura 4: Como el tamaño del filtro afecta el vector de salida



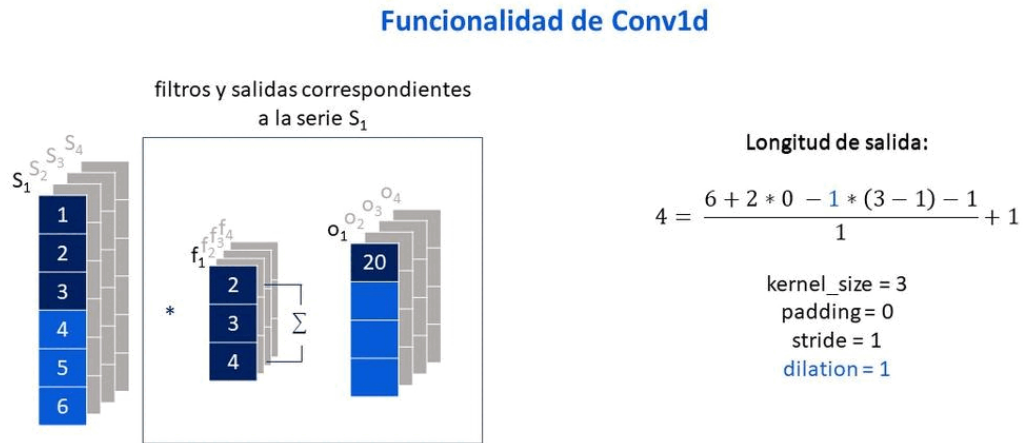
Elaboración propia: Elaborada a partir de Jing (2020). Muestra como el tamaño del vector de salida cambia según el tamaño de filtro que se usa.

Figura 5: Como stride afecta el vector de salida



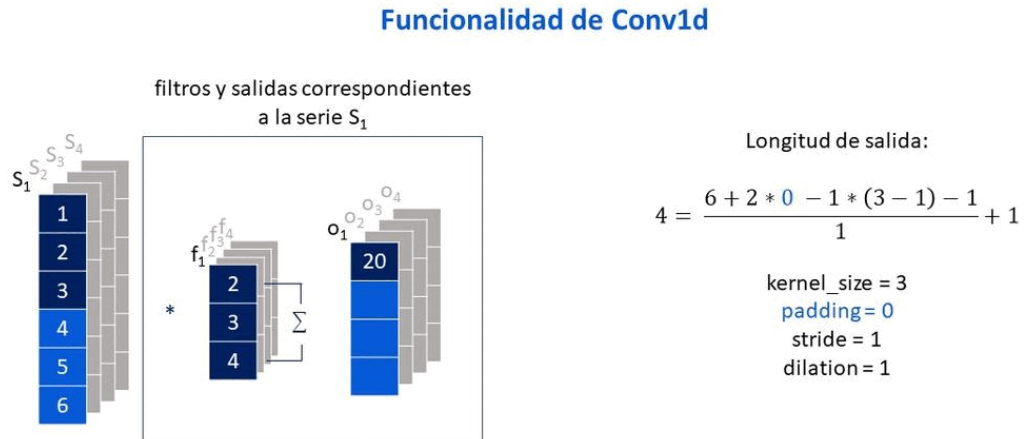
Elaboración propia: Elaborada a partir de Jing (2020). Muestra como el parámetro stride afecta el tamaño del vector de salida.

Figura 6: Como dilation afecta el vector de salida



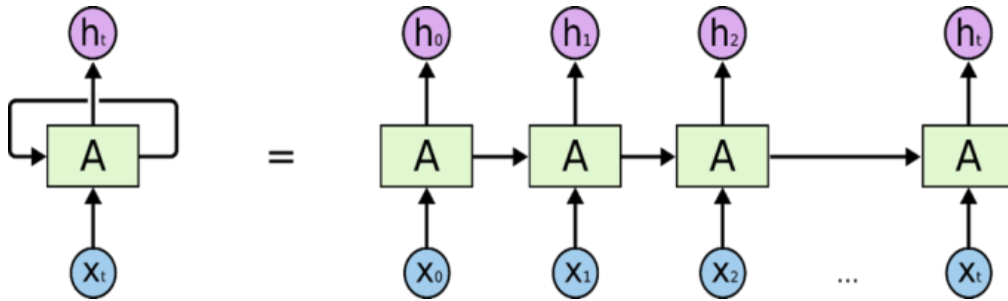
Elaboración propia: Elaborada a partir de Jing (2020). Muestra como el parámetro dilation afecta el tamaño del vector de salida.

Figura 7: Como padding afecta el vector de salida



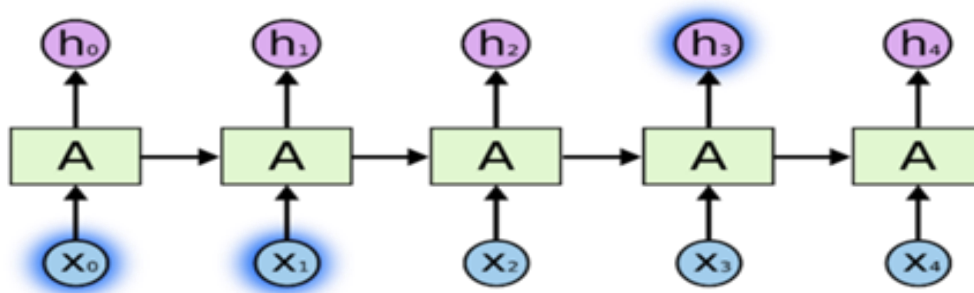
Elaboración propia: Elaborada a partir de Jing (2020). Muestra como el parámetro padding afecta el tamaño del vector de salida.

Figura 8: Despliegue del bucle de una red neuronal recurrente estándar



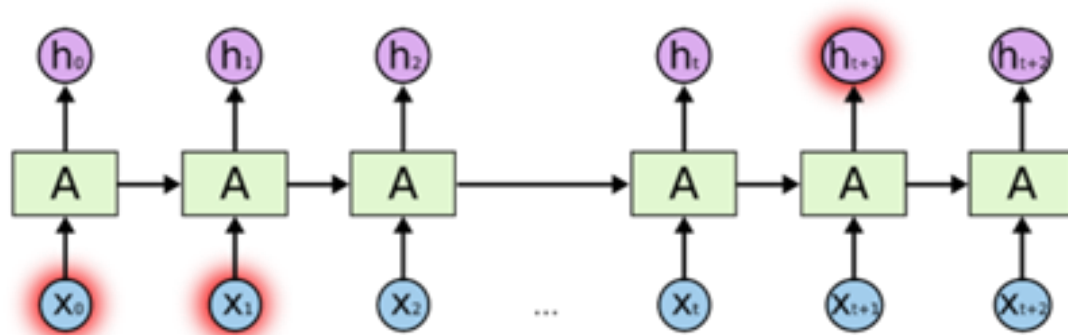
Tomada de: Understanding LSTM networks, Olah (2015).

Figura 9: Información relevante cercana



Tomada de: Understanding LSTM networks, Olah (2015).

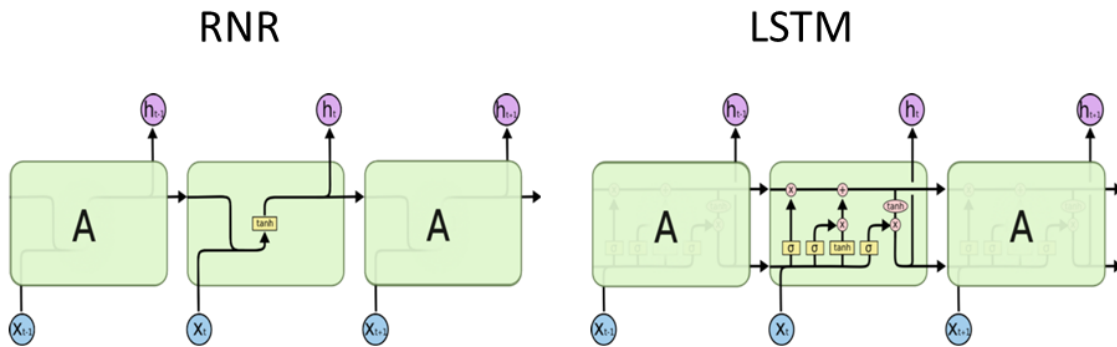
Figura 10: Información relevante lejana



Tomada de: Understanding LSTM networks, Olah (2015).

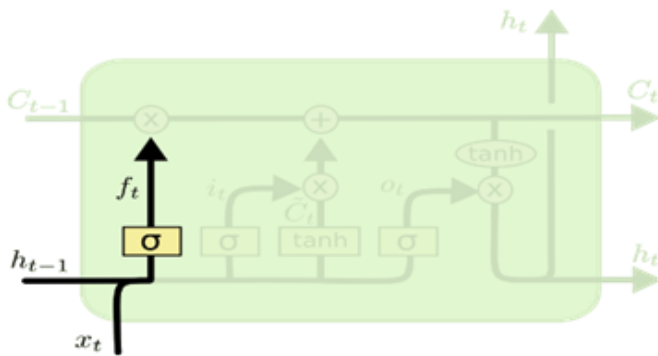


Figura 11: Diferencia entre los módulos de repetición



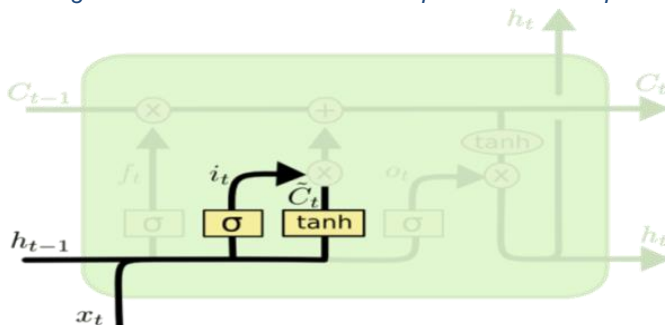
Tomada de: Understanding LSTM networks, Olah (2015).

Figura 12: LSTM funcionalidad: Representación del paso 1



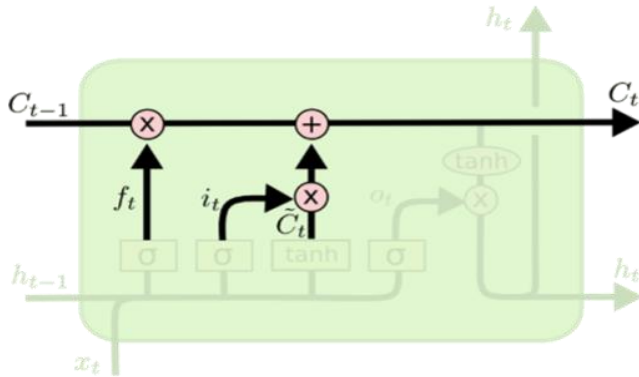
Tomada de: Understanding LSTM networks, Olah (2015).

Figura 13: LSTM funcionalidad: Representación del paso 2



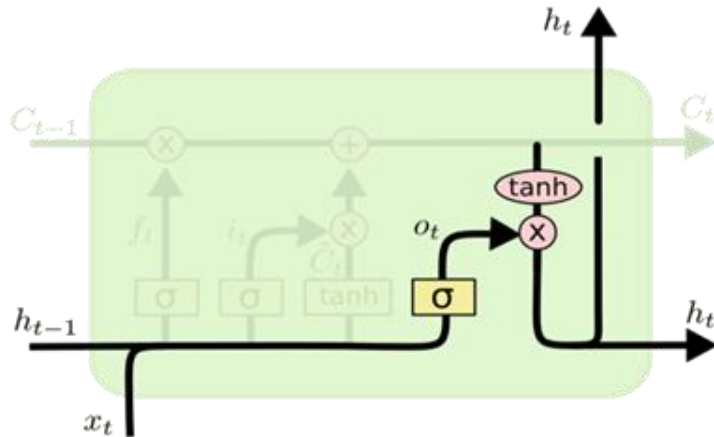
Tomada de: Understanding LSTM networks, Olah (2015).

Figura 14: LSTM funcionalidad: Representación del paso 3



Tomada de: Understanding LSTM networks, Olah (2015).

Figura 15: LSTM funcionalidad: Representación del paso 4



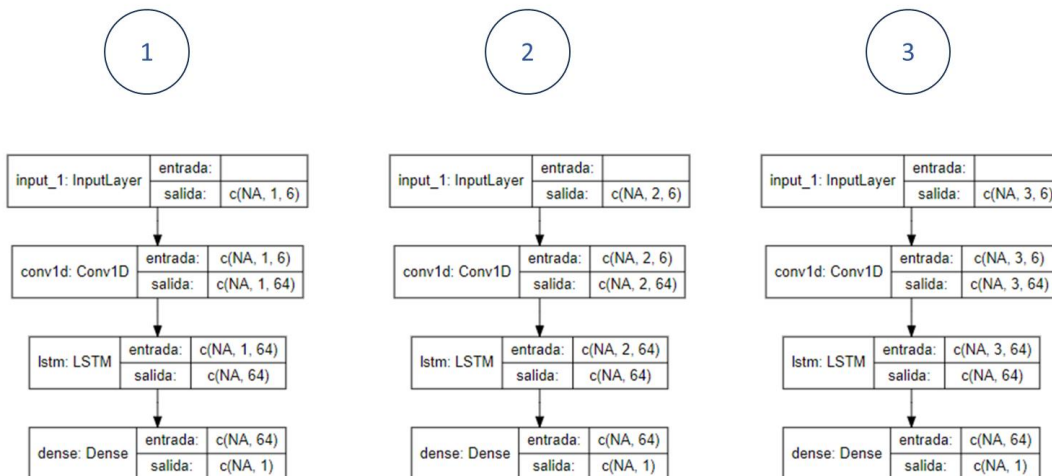
Tomada de: Understanding LSTM networks, Olah (2015)

Figura 16: Visualización de vectores de entrada y salida



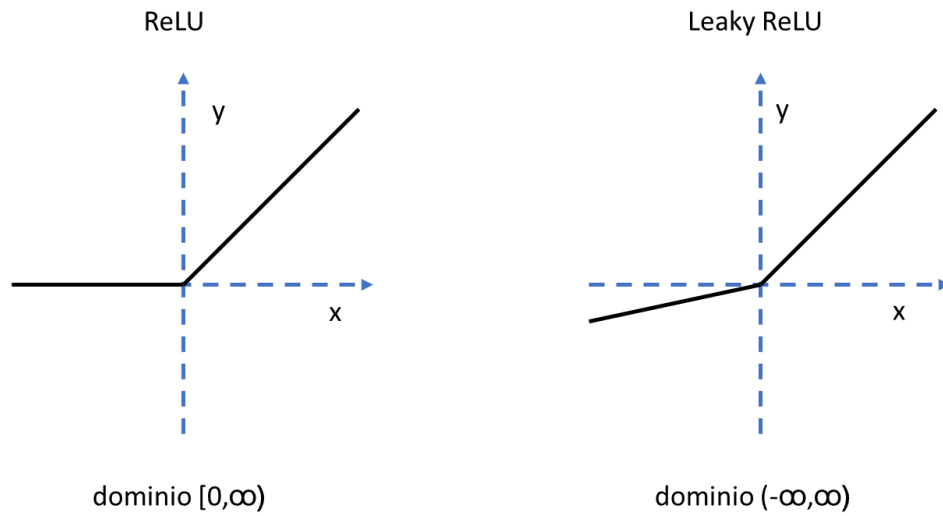
Elaboración propia: Elaborada a partir de imagen en Chollet y Allaire (2018). Muestra como lucen los vectores tridimensionales de entradas y salidas correspondientes a los datos de una empresa, en el caso de que se usen tres observaciones para crear el vector de entrada.

Figura 17: Distintas estructuras según los distintos tamaños de vectores de entrada



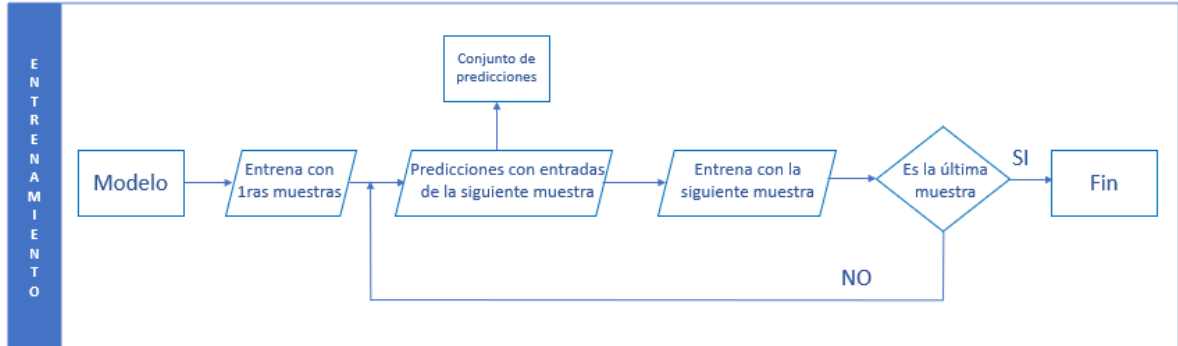
Elaboración propia: Elaborada a partir de los distintos modelos construidos usando los paquetes keras y tensorflow en R, y fueron gráficas mediante el uso del paquete Diagrammer.

Figura 18: Dominio de ReLU y Leaky ReLU



Elaboración propia: Elaborada a partir de las imágenes que se observan en Rallabandi (2023).

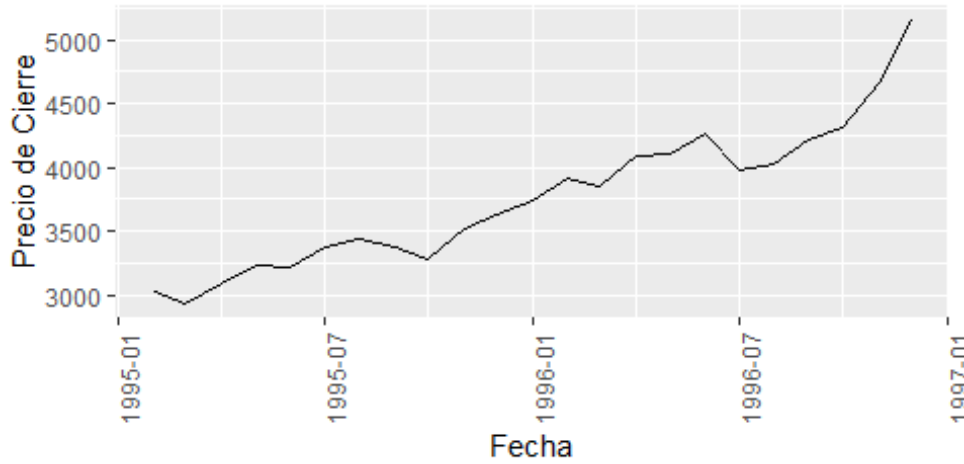
Figura 19: Diagrama de flujo de la metodología de Walk Forward Validation



Elaboración propia

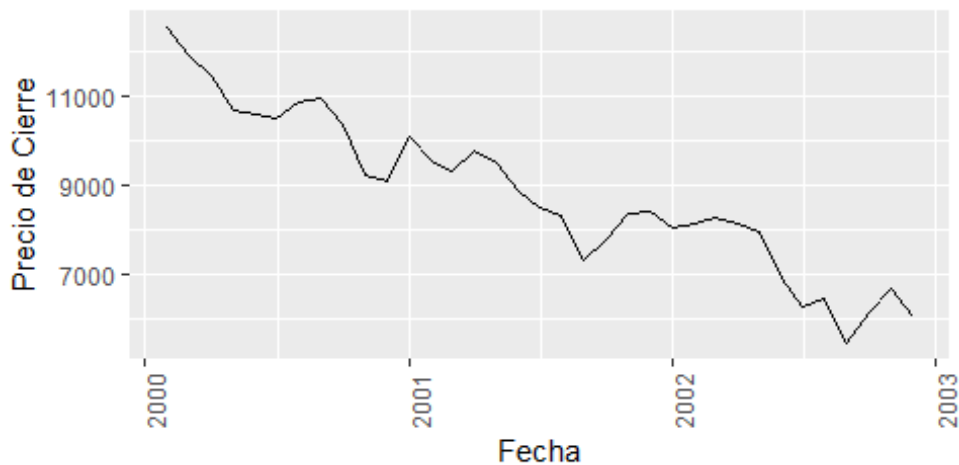
## Anexo.2 Gráficas

*Gráfica 1: Tendencia alcista y heterocedástica*



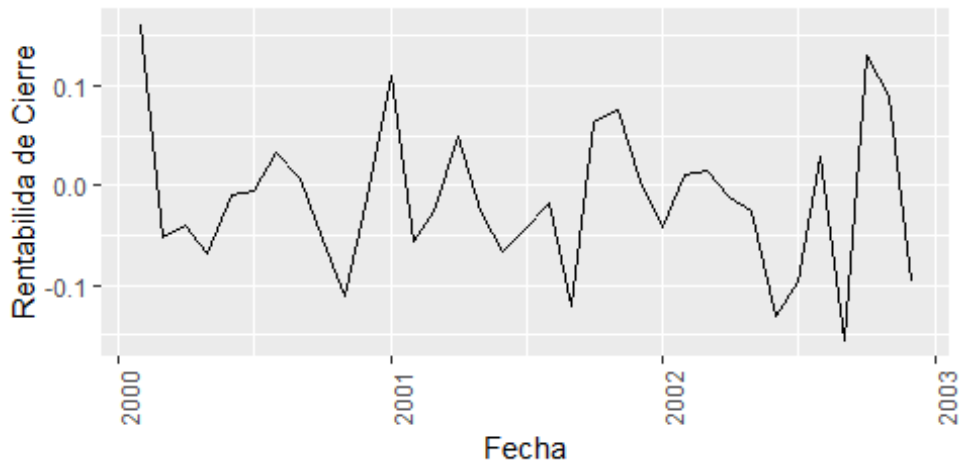
Elaboración propia: Mediante el uso de RStudio con la base de datos histórico del IBEX, obtenida de <https://finance.yahoo.com/>, en el periodo comprendido entre el 01-1995 al 01-1997.

*Gráfica 2: Tendencia bajista y heterocedástica*



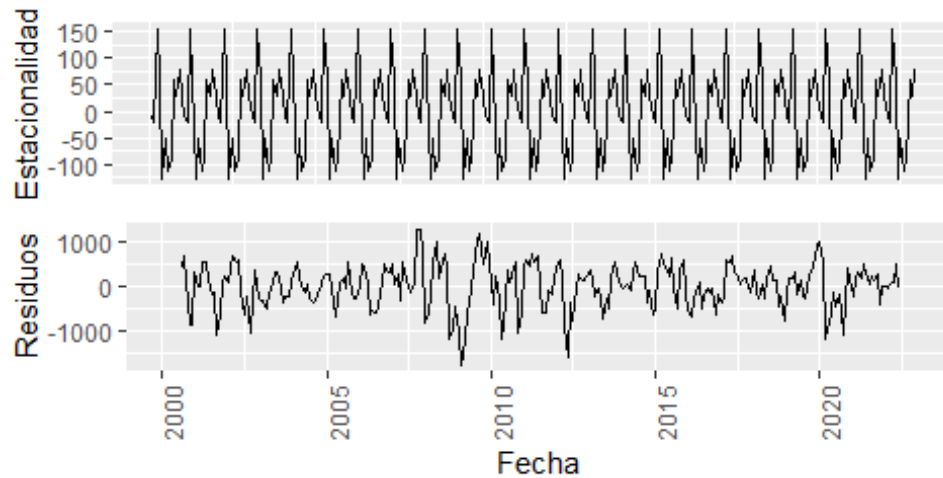
Elaboración propia: Mediante el uso de RStudio con la base de datos histórico del IBEX, obtenida de <https://finance.yahoo.com/>, en el periodo comprendido entre el 01-2000 al 01-2003.

Gráfica 3: Sin tendencia, homocedástica y estacionaria



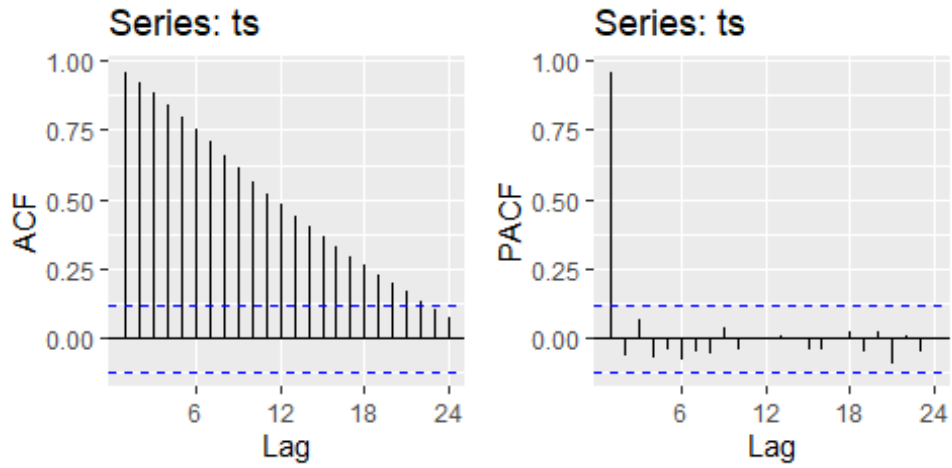
Elaboración propia: Mediante el uso de RStudio con la base de datos histórico del IBEX obtenida de <https://finance.yahoo.com/>, en el periodo comprendido entre el 01-2000 al 01-2003, usando las rentabilidades calculadas a partir del precio de cierre.

Gráfica 4: Descomposición: estacionalidad y error



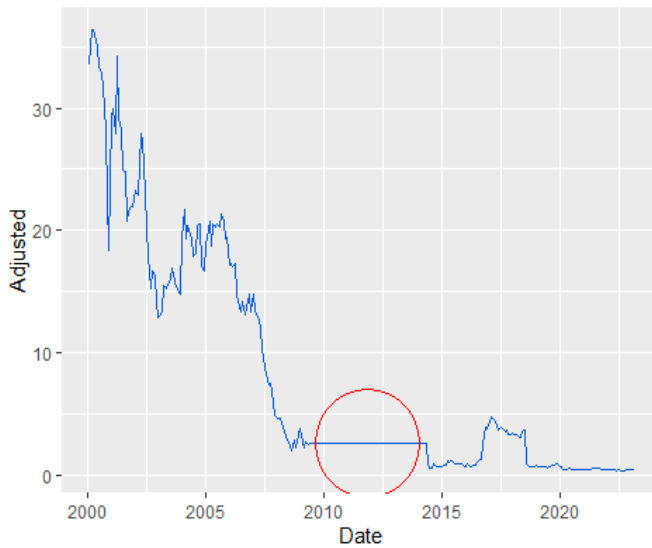
Elaboración propia: Mediante el uso de RStudio con la base de datos histórico del IBEX obtenida de <https://finance.yahoo.com/>, descomponiendo la serie de tiempo conformada por las observaciones que abarcan el periodo del 01-2000 al 01-2023.

Gráfica 5: Correlograma



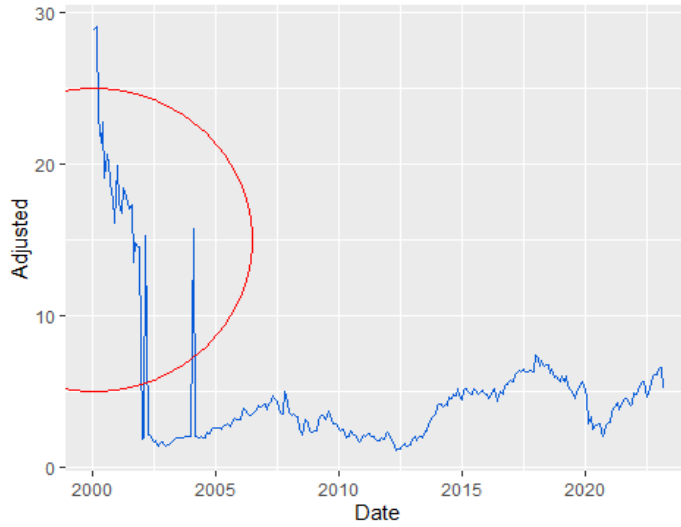
Elaboración propia: Mediante el uso de RStudio.

Gráfica 6: Tendencia constante en los precios de cierre ajustado de la empresa "Nueva Expresión Textil S.A"



Elaboración propia: A partir de los datos obtenidos de Yahoo Finance correspondientes a la empresa "Nueva Expresión Textil S.A" en el periodo del 31 de enero del 2000 al 28 de febrero del 2023.

Gráfica 7: Cambio brusco de precios que reflejan calculo erróneo de los precios de cierre ajustado, "BANKINTER,S.A."



Elaboración propia: A partir de los datos obtenidos de Yahoo Finance correspondientes a la empresa "Bankinter, S.A" en el periodo del 31 de enero del 2000 al 28 de febrero del 2023.

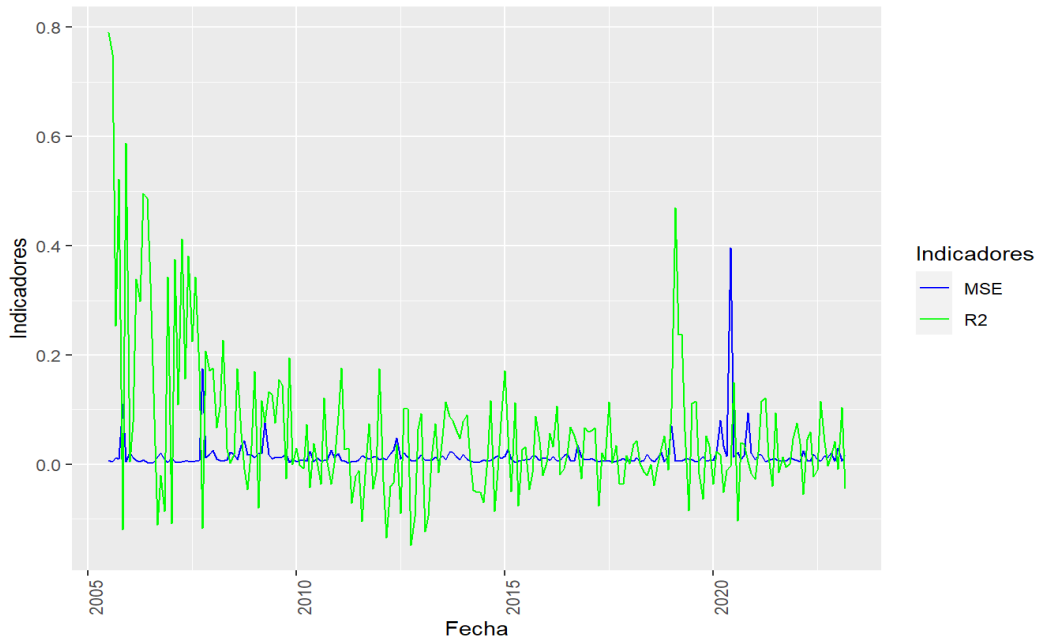
Gráfica 8: Evolución de los indicadores – Entradas con una observación



Elaboración propia: Mediante el uso de R y Rstudio.

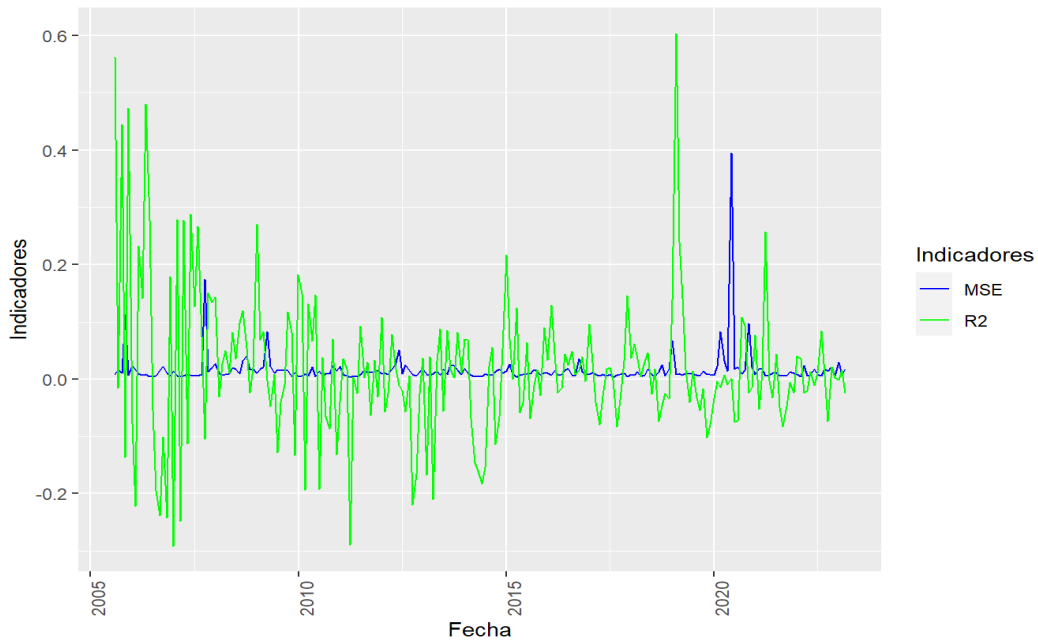


Gráfica 9: Evolución de los indicadores – Entradas con dos observaciones



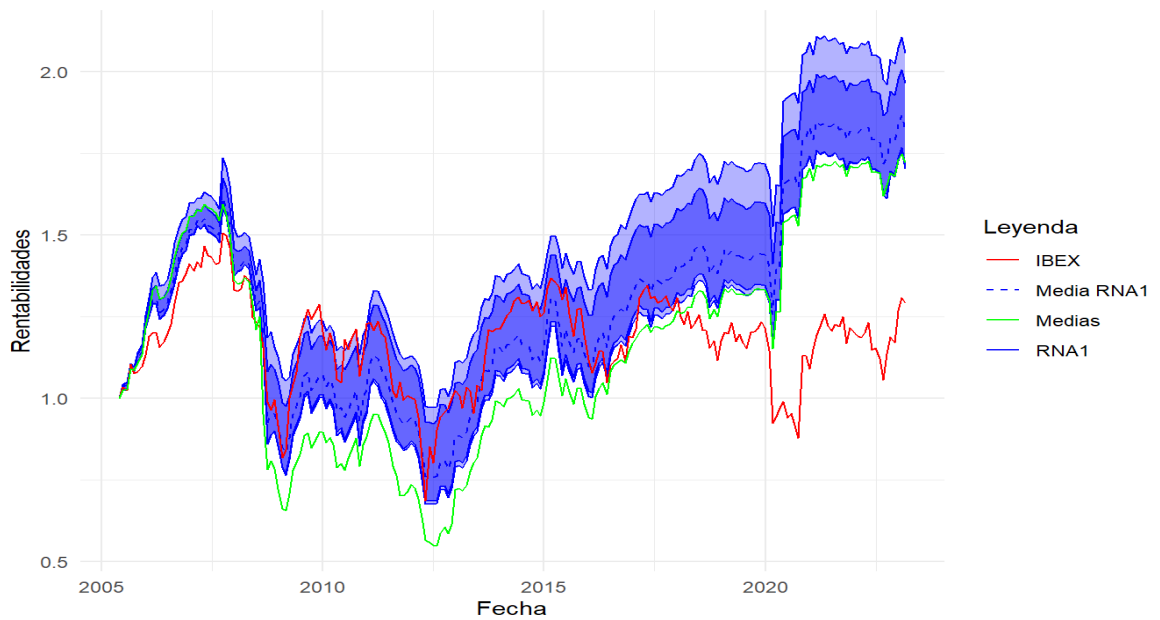
Elaboración propia: Mediante el uso de R y Rstudio.

Gráfica 10: Evolución de los indicadores – Entradas con tres observaciones



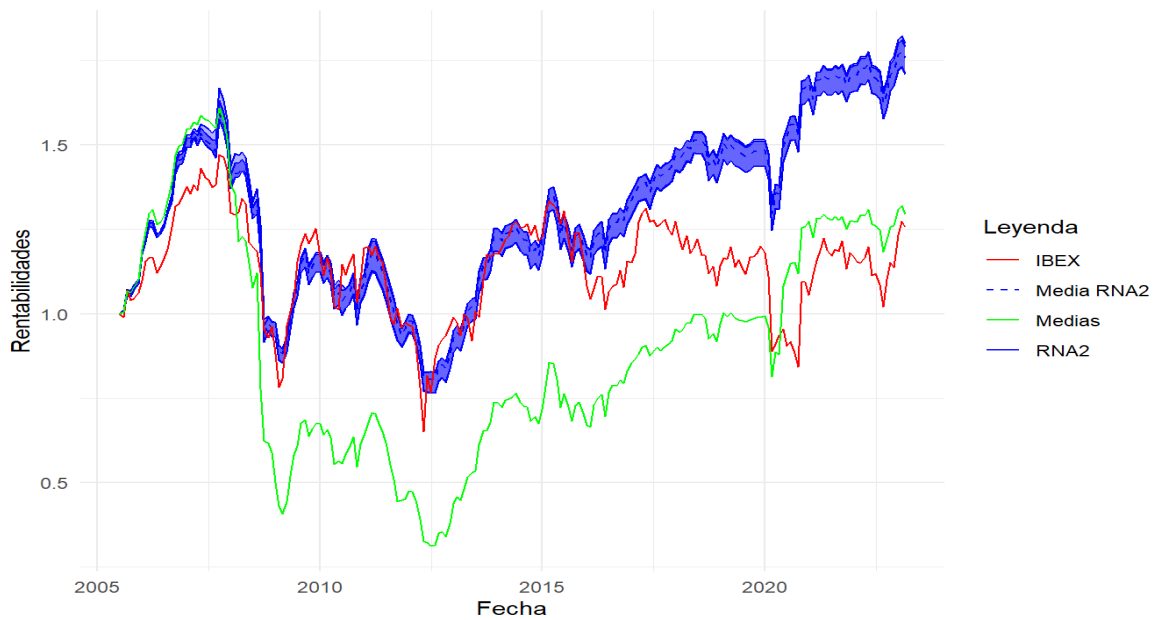
Elaboración propia: Mediante el uso de R y Rstudio.

Gráfica 11: Evolución de las carteras y el IBEX – Entradas con una observación



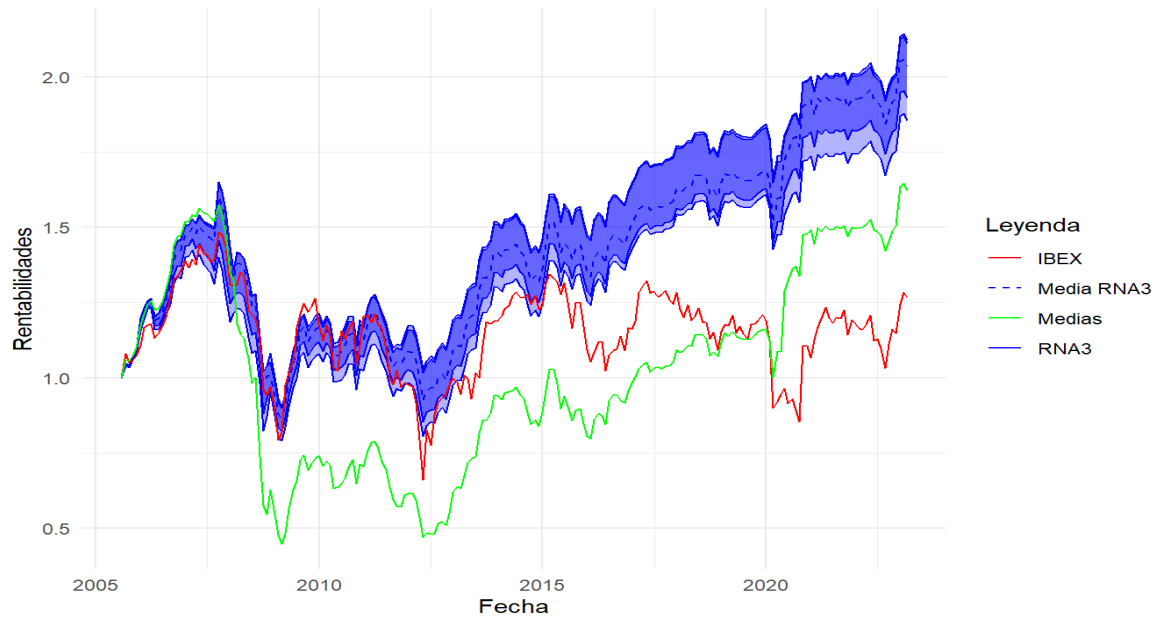
Elaboración propia: Mediante el uso de R y Rstudio.

Gráfica 12: Evolución de las carteras y el IBEX – Entradas con dos observaciones



Elaboración propia: Mediante el uso de R y Rstudio.

Gráfica 13: Evolución las carteras y el IBEX – Entradas con tres observaciones



Elaboración propia: Mediante el uso de R y Rstudio.

## Anexo.3 Tablas

*Tabla 1: Estructura de datos de precios*

Date	Open	High	Low	Close	Volume	Adjusted
2001-05-24	3.600	3.620	3.510	3.608	216270100	-0.1317839
2001-05-25	3.600	3.676	3.580	3.602	50448300	-0.1315648
2001-05-28	3.560	3.604	3.544	3.580	26118945	-0.1307612
2001-05-29	3.562	3.626	3.562	3.614	26910070	-0.1320031
2001-05-30	3.606	3.648	3.602	3.620	48229995	-0.1322222
2001-05-31	3.620	3.676	3.610	3.670	24806710	-0.1340484

Elaboración propia: Mediante el uso de RStudio con la base de datos histórico de "INDITEX", obtenida de <https://finance.yahoo.com/>, en el periodo comprendido entre el 24-05-2001 al 31-05-2001.

Tabla 2: Lista de empresas cotizadas

NOMBRE	TICKERS	SECTOR-SUBSECTOR	MERCADO	INDICE	Seleccionadas
ACCIONA,S.A.	ANA.MC	Mat.Basicos, Industria y Construcción - Construcción	Mercado Continuo	IBEX 35®, IGBM	X
ACERINOX, S.A.	ACX.MC	Mat.Basicos, Industria y Construcción - Mineral, Metales y Transformación	Mercado Continuo	IBEX 35®, IGBM, IBEXTD®	X
ACS,ACTIVIDADES DE CONST.Y SERVICIOS S.A	ACS.MC	Mat.Basicos, Industria y Construcción - Construcción	Mercado Continuo	IBEX 35®, IGBM, IBEXTD®	X
ADOLFO DOMINGUEZ, S.A.	ADZ.MC	Bienes de Consumo - Textil, Vestido y Calzado	Mercado Continuo	IGBM	X
AEDAS HOMES, S.A.	AEDAS.MC	Servicios Inmobiliarios - Inmobiliarias y Otros	Mercado Continuo	IGBM	X
AENA, S.M.E., S.A.	AENA.MC	Servicios de Consumo - Transporte y Distribución	Mercado Continuo	IBEX 35®, IGBM	X
AIRBUS SE	AIR.MC	Mat.Basicos, Industria y Construcción - Aeroespacial	Mercado Continuo	IGBM	X
AIRTIFICIAL INTELLIGENCE STRUCTURES S.A.	AI.MC	Mat.Basicos, Industria y Construcción - Ingeniería y Otros	Mercado Continuo	IGBM	
ALANTRA PARTNERS, S.A.	ALNT.MC	Servicios Financieros - Cartera y Holding	Mercado Continuo	IGBM	
ALMIRALL, S.A.	ALM.MC	Bienes de Consumo - Productos farmacéuticos y Biotecnología	Mercado Continuo	IGBM	X
AMADEUS IT GROUP, S.A.	AMS.MC	Tecnología y Telecomunicaciones - Electrónica y Software	Mercado Continuo	IBEX 35®, IGBM	X
AMPER, S.A.	AMP.MC	Tecnología y Telecomunicaciones - Electrónica y Software	Mercado Continuo	IGBM	X
AMREST HOLDINGS, S.E.	EAT.MC	Servicios de Consumo - Ocio, Turismo y Hostelería	Mercado Continuo	IGBM	X
APERAM, SOCIETE ANONYME	APAM.MC	Mat.Basicos, Industria y Construcción - Mineral, Metales y Transformación	Mercado Continuo		X

Aplicación de redes neuronales artificiales y programación cuadrática en la gestión de carteras

NOMBRE	TICKERS	SECTOR-SUBSECTOR	MERCADO	INDICE	Seleccionadas
APPLUS SERVICES, S.A.	APPS.MC	Mat.Basicos, Industria y Construcción - Ingeniería y Otros	Mercado Continuo	IGBM	X
ARCELORMITTAL, S.A.	MTS.MC	Mat.Basicos, Industria y Construcción - Mineral, Metales y Transformación	Mercado Continuo	IBEX 35®, IGBM	X
ÁRIMA REAL ESTATE SOCIMI, S.A.	ARM.MC	Servicios Inmobiliarios - SOCIMI	Mercado Continuo	IGBM	X
ATRESMEDIA CORP. DE MEDIOS DE COM. S.A.	A3M.MC	Servicios de Consumo - Medios de Comunicación y Publicidad	Mercado Continuo	IGBM	X
ATRY'S HEALTH, S.A.	ATRY.MC	Bienes de Consumo - Productos farmacéuticos y Biotecnología	Mercado Continuo	IGBM	X
AUDAX RENOVABLES, S.A.	ADX.MC	Petróleo y Energía - Energías Renovables	Mercado Continuo	IGBM	
AZKOYEN S.A.	AZK.MC	Mat.Basicos, Industria y Construcción - Fabric. y Montaje Bienes de Equipo	Mercado Continuo	IGBM	X
BANCO BILBAO VIZCAYA ARGENTARIA, S.A.	BBVA.MC	Servicios Financieros - Bancos y Cajas de Ahorro	Mercado Continuo	IBEX 35®, IGBM	X
BANCO DE SABADELL, S.A.	SAB.MC	Servicios Financieros - Bancos y Cajas de Ahorro	Mercado Continuo	IBEX 35®, IGBM	X
BANCO SANTANDER, S.A.	SAN.MC	Servicios Financieros - Bancos y Cajas de Ahorro	Mercado Continuo	IBEX 35®, IGBM	X
BANKINTER,S.A.	BKT.MC	Servicios Financieros - Bancos y Cajas de Ahorro	Mercado Continuo	IBEX 35®, IGBM, IBEXTD®	X
BERKELEY ENERGIA LIMITED	BKY.MC	Mat.Basicos, Industria y Construcción - Mineral, Metales y Transformación	Mercado Continuo	IGBM	X
BODEGAS RIOJANAS, S.A.	RIO.MC	Bienes de Consumo - Alimentación y Bebidas	Mercado Continuo	IGBM	X
BORGES AGRICULTURAL & INDUST. NUTS, S.A.	BAIN.MC	Bienes de Consumo - Alimentación y Bebidas	Mercado Continuo		

Aplicación de redes neuronales artificiales y programación cuadrática en la gestión de carteras

NOMBRE	TICKERS	SECTOR-SUBSECTOR	MERCADO	INDICE	Seleccionadas
CAIXABANK, S.A.	CABK.MC	Servicios Financieros - Bancos y Cajas de Ahorro	Mercado Continuo	IBEX 35®, IGBM	X
CAJA DE AHORROS DEL MEDITERRANEO	CAM.MC	Servicios Financieros - Bancos y Cajas de Ahorro	Mercado Continuo		
CASH, S.A.	CASH.MC	Servicios de Consumo - Otros Servicios	Mercado Continuo	IGBM, IBEXTD®	X
CELLNEX TELECOM, S.A.	CLNX.MC	Tecnología y Telecomunicaciones - Telecomunicaciones y Otros	Mercado Continuo	IBEX 35®, IGBM	X
CIA. DE DIST. INTEG. LOGISTA HOLDINGS	LOG.MC	Servicios de Consumo - Transporte y Distribución	Mercado Continuo	IBEX 35®, IGBM, IBEXTD®	X
CIA. ESPAÑOLA VIVIENDAS EN ALQUILER, S.A	CEV.MC	Servicios Inmobiliarios - Inmobiliarias y Otros	Mercado Continuo		
CIA.LEVANTINA, EDIFICACION DE O.PUBLICAS	CLEO.MC	Mat.Basicos, Industria y Construcción - Construcción	Mercado Continuo		
CIE AUTOMOTIVE, S.A.	CIE.MC	Mat.Basicos, Industria y Construcción - Mineral, Metales y Transformación	Mercado Continuo	IGBM	X
CLINICA BAVIERA, S.A.	CBAV.MC	Servicios de Consumo - Otros Servicios	Mercado Continuo	IGBM	X
COCA-COLA EUROPACIFIC PARTNERS PLC	CCEP.MC	Bienes de Consumo - Alimentación y Bebidas	Mercado Continuo	IGBM	X
CONSTRUCC. Y AUX. DE FERROCARRILES, S.A.	CAF.MC	Mat.Basicos, Industria y Construcción - Fabric. y Montaje Bienes de Equipo	Mercado Continuo	IGBM	X
CORP. ACCIONA ENERGÍAS RENOVABLES, S.A.	ANE.MC	Petróleo y Energía - Energías Renovables	Mercado Continuo	IBEX 35®, IGBM	X

Aplicación de redes neuronales artificiales y programación cuadrática en la gestión de carteras

NOMBRE	TICKERS	SECTOR-SUBSECTOR	MERCADO	INDICE	Seleccionadas
CORPORACION FINANCIERA ALBA, S.A.	ALB.MC	Servicios Financieros - Cartera y Holding	Mercado Continuo	IGBM	X
DEOLEO, S.A.	OLE.MC	Bienes de Consumo - Alimentación y Bebidas	Mercado Continuo	IGBM	X
DESA	DESA.MC	Mat.Basicos, Industria y Construcción - Mineral, Metales y Transformación	Mercado Continuo		
DIA-DISTRIBUIDORA INT. DE ALIMENT. S.A.	DIA.MC	Servicios de Consumo - Comercio	Mercado Continuo	IGBM	X
DURO FELGUERA, S.A.	MDF.MC	Mat.Basicos, Industria y Construcción - Ingeniería y Otros	Mercado Continuo	IGBM	X
EBRO FOODS, S.A.	EBRO.MC	Bienes de Consumo - Alimentación y Bebidas	Mercado Continuo	IGBM, IBEXTD®	X
EDREAMS ODIGEO, S.A.	EDR.MC	Servicios de Consumo - Ocio, Turismo y Hostelería	Mercado Continuo	IGBM	X
ELECNOR S. A.	ENO.MC	Mat.Basicos, Industria y Construcción - Fabric. y Montaje Bienes de Equipo	Mercado Continuo	IGBM, IBEXTD®	X
ENAGAS, S.A.	ENG.MC	Petróleo y Energía - Electricidad y Gas	Mercado Continuo	IBEX 35®, IGBM, IBEXTD®	X
ENCE ENERGIA Y CELULOSA, S.A.	ENC.MC	Bienes de Consumo - Papel y Artes Gráficas	Mercado Continuo	IGBM	X
ENDESA, SOCIEDAD ANONIMA	ELE.MC	Petróleo y Energía - Electricidad y Gas	Mercado Continuo	IBEX 35®, IGBM, IBEXTD®	X
ERCROS S.A.	ECR.MC	Mat.Basicos, Industria y Construcción - Industria Química	Mercado Continuo	IGBM	X
FAES FARMA, S.A.	FAE.MC	Bienes de Consumo - Productos farmacéuticos y Biotecnología	Mercado Continuo	IGBM, IBEXTD®	X
FERROVIAL, S.A.	FER.MC	Mat.Basicos, Industria y Construcción - Construcción	Mercado Continuo	IBEX 35®, IGBM	X



Aplicación de redes neuronales artificiales y programación cuadrática en la gestión de carteras

NOMBRE	TICKERS	SECTOR-SUBSECTOR	MERCADO	INDICE	Seleccionadas
FLUIDRA, S.A.	FDR.MC	Mat.Basicos, Industria y Construcción - Ingeniería y Otros	Mercado Continuo	IBEX 35®, IGBM	X
FOMENTO DE CONSTR. Y CONTRATAS S.A.	FCC.MC	Mat.Basicos, Industria y Construcción - Construcción	Mercado Continuo	IGBM, IBEX TOP Dividendo®	X
GENERAL DE ALQUILER DE MAQUINARIA, S.A.	GAM.MC	Mat.Basicos, Industria y Construcción - Ingeniería y Otros	Mercado Continuo	IGBM	
GESTAMP AUTOMOCION, S.A.	GEST.MC	Mat.Basicos, Industria y Construcción - Fabric. y Montaje Bienes de Equipo	Mercado Continuo	IGBM	X
GLOBAL DOMINION ACCESS, S.A.	DOM.MC	Tecnología y Telecomunicaciones - Telecomunicaciones y Otros	Mercado Continuo	IGBM	X
GREENERGY RENOVABLES, S.A.	GRE.MC	Petróleo y Energía - Energías Renovables	Mercado Continuo	IGBM	X
GRIFOLS, S.A.	GRF.MC	Bienes de Consumo - Productos farmacéuticos y Biotecnología	Mercado Continuo	IBEX 35®, IGBM	X
GRUPO CATALANA OCCIDENTE, S.A.	GCO.MC	Servicios Financieros - Seguros	Mercado Continuo	IGBM	X
GRUPO ECOENER, S.A.	ENER.MC	Petróleo y Energía - Energías Renovables	Mercado Continuo	IGBM	X
GRUPO EMPRESARIAL SAN JOSE, S.A.	GSJ.MC	Mat.Basicos, Industria y Construcción - Construcción	Mercado Continuo	IGBM	X
GRUPO EZENTIS, S.A.	EZE.MC	Tecnología y Telecomunicaciones - Telecomunicaciones y Otros	Mercado Continuo		X
IBERDROLA, S.A.	IBE.MC	Petróleo y Energía - Electricidad y Gas	Mercado Continuo	IBEX 35®, IGBM, IBEXTD®	X
IBERPAPEL GESTION, S.A.	IBG.MC	Bienes de Consumo - Papel y Artes Gráficas	Mercado Continuo	IGBM	X

Aplicación de redes neuronales artificiales y programación cuadrática en la gestión de carteras

NOMBRE	TICKERS	SECTOR-SUBSECTOR	MERCADO	INDICE	Seleccionadas
INDRA SISTEMAS, S.A., SERIE A	IDR.MC	Tecnología y Telecomunicaciones - Electrónica y Software	Mercado Continuo	IBEX 35®, IGBM	X
INDUSTRIA DE DISEÑO TEXTIL, SA "INDITEX"	ITX.MC	Bienes de Consumo - Textil, Vestido y Calzado	Mercado Continuo	IBEX 35®, IGBM	X
INMOBILIARIA COLONIAL SOCIMI, S.A.	COL.MC	Servicios Inmobiliarios - SOCIMI	Mercado Continuo	IBEX 35®, IGBM	X
INMOBILIARIA DEL SUR, S.A.	ISUR.MC	Servicios Inmobiliarios - Inmobiliarias y Otros	Mercado Continuo	IGBM	X
INNOVATIVE SOLUTIONS ECOSYSTEM, S.A.	ISE.MC	Servicios de Consumo - Comercio	Mercado Continuo		
INTERNATIONAL CONSOLIDAT. AIRLINES GROUP	IAG.MC	Servicios de Consumo - Transporte y Distribución	Mercado Continuo	IBEX 35®, IGBM	X
LABORATORIO REIG JOFRE, S.A.	RJF.MC	Bienes de Consumo - Productos farmacéuticos y Biotecnología	Mercado Continuo	IGBM	X
LABORATORIOS FARMACEUTICOS ROVI, S.A.	ROVI.MC	Bienes de Consumo - Productos farmacéuticos y Biotecnología	Mercado Continuo	IBEX 35®, IGBM	X
LAR ESPAÑA REAL ESTATE, SOCIMI, S.A.	LRE.MC	Servicios Inmobiliarios - SOCIMI	Mercado Continuo	IGBM, IBEXTD®	X
LIBERTAS 7, S.A.	LIB.MC	Servicios Inmobiliarios - Inmobiliarias y Otros	Mercado Continuo		
LINEA DIRECTA ASEGURADORA, S.A.	LDA.MC	Servicios Financieros - Seguros	Mercado Continuo	IGBM	X
LINGOTES ESPECIALES, S.A.	LGT.MC	Mat.Basicos, Industria y Construcción - Mineral, Metales y Transformación	Mercado Continuo		X

Aplicación de redes neuronales artificiales y programación cuadrática en la gestión de carteras

NOMBRE	TICKERS	SECTOR-SUBSECTOR	MERCADO	INDICE	Seleccionadas
MAPFRE, S.A.	MAP.MC	Servicios Financieros - Seguros	Mercado Continuo	IBEX 35®, IGBM, IBEXTD®	X
MELIA HOTELS INTERNATIONAL, S.A.	MEL.MC	Servicios de Consumo - Ocio, Turismo y Hostelería	Mercado Continuo	IBEX 35®, IGBM	X
MERLIN PROPERTIES, SOCIMI, S.A.	MRL.MC	Servicios Inmobiliarios - SOCIMI	Mercado Continuo	IBEX 35®, IGBM, IBEXTD®	X
METROVACESA, S.A.	MVC.MC	Servicios Inmobiliarios - Inmobiliarias y Otros	Mercado Continuo	IGBM	X
MIQUEL Y COSTAS & MIQUEL, S.A.	MCM.MC	Bienes de Consumo - Papel y Artes Gráficas	Mercado Continuo	IGBM, IBEXTD®	X
MONTEBALITO, S.A.	MTB.MC	Servicios Inmobiliarios - Inmobiliarias y Otros	Mercado Continuo		X
NATURGY ENERGY GROUP, S.A.	NTGY.MC	Petróleo y Energía - Electricidad y Gas	Mercado Continuo	IBEX 35®, IGBM, IBEXTD®	X
NATURHOUSE HEALTH, S.A.	NTH.MC	Bienes de Consumo - Alimentación y Bebidas	Mercado Continuo	IGBM	X
NEINOR HOMES, S.A.	HOME.MC	Servicios Inmobiliarios - Inmobiliarias y Otros	Mercado Continuo	IGBM	X
NH HOTEL GROUP, S.A.	NHH.MC	Servicios de Consumo - Ocio, Turismo y Hostelería	Mercado Continuo	IGBM	X
NICOLAS CORREA S.A.	NEA.MC	Mat.Basicos, Industria y Construcción - Fabric. y Montaje Bienes de Equipo	Mercado Continuo	IGBM, IBEXTD®	X
NUEVA EXPRESION TEXTIL, S.A.	NXT.MC	Bienes de Consumo - Textil, Vestido y Calzado	Mercado Continuo	IGBM	
NYESA VALORES CORPORACION, S.A.	NYE.MC	Servicios Inmobiliarios - Inmobiliarias y Otros	Mercado Continuo	IGBM	
OBRASCON HUARTE LAIN, S.A.	OHLA.MC	Mat.Basicos, Industria y Construcción - Construcción	Mercado Continuo	IGBM	X

NOMBRE	TICKERS	SECTOR-SUBSECTOR	MERCADO	INDICE	Seleccionadas
OPDENERGY HOLDING, S.A.	OPDE.MC	Petróleo y Energía - Energías Renovables	Mercado Continuo	IGBM	X
ORYZON GENOMICS, S.A.	ORY.MC	Bienes de Consumo - Productos farmacéuticos y Biotecnología	Mercado Continuo	IGBM	X
PESCANOVA, S.A.	PVA.MC	Bienes de Consumo - Alimentación y Bebidas	Mercado Continuo	IGBM	
PHARMA MAR, S.A.	PHM.MC	Bienes de Consumo - Productos farmacéuticos y Biotecnología	Mercado Continuo	IGBM	X
PRIM, S.A.	PRM.MC	Bienes de Consumo - Productos farmacéuticos y Biotecnología	Mercado Continuo	IGBM	X
PROMOTORA DE INFORMACIONES, S.A.	PRS.MC	Servicios de Consumo - Medios de Comunicación y Publicidad	Mercado Continuo	IGBM	X
PROSEGUR, CIA. DE SEGURIDAD, S.A.	PSG.MC	Servicios de Consumo - Otros Servicios	Mercado Continuo	IGBM, IBEXTD®	X
REALIA BUSINESS, S.A.	RLIA.MC	Servicios Inmobiliarios - Inmobiliarias y Otros	Mercado Continuo	IGBM	X
RED ELECTRICA CORPORACION, S.A.	RED.MC	Petróleo y Energía - Electricidad y Gas	Mercado Continuo	IBEX 35®, IGBM, IBEXTD®	X
RENTA 4 BANCO, S.A.	R4.MC	Servicios Financieros - Servicios de Inversión	Mercado Continuo	IGBM	X
RENTA CORPORACION REAL ESTATE, S.A.	REN.MC	Servicios Inmobiliarios - Inmobiliarias y Otros	Mercado Continuo	IGBM	
REPSOL, S.A.	REP.MC	Petróleo y Energía - Petróleo	Mercado Continuo	IBEX 35®, IGBM, IBEXTD®	X
SACYR, S.A.	SCYR.MC	Mat.Basicos, Industria y Construcción - Construcción	Mercado Continuo	IBEX 35®, IGBM, IBEXTD®	X

Aplicación de redes neuronales artificiales y programación cuadrática en la gestión de carteras

NOMBRE	TICKERS	SECTOR-SUBSECTOR	MERCADO	INDICE	Seleccionadas
SOLARIA ENERGIA Y MEDIO AMBIENTE, S.A.	SLR.MC	Petróleo y Energía - Energías Renovables	Mercado Continuo	IBEX 35®, IGBM	X
SOLTEC POWER HOLDINGS, S.A.	SOL.MC	Petróleo y Energía - Energías Renovables	Mercado Continuo	IGBM	X
SQUIRREL MEDIA, S.A	SQRL.MC	Servicios de Consumo - Medios de Comunicación y Publicidad	Mercado Continuo		
TALGO, S.A.	TLGO.MC	Mat.Basicos, Industria y Construcción - Fabric. y Montaje Bienes de Equipo	Mercado Continuo	IGBM	X
TECNICAS REUNIDAS, S.A.	TRE.MC	Mat.Basicos, Industria y Construcción - Ingeniería y Otros	Mercado Continuo	IGBM	X
TELEFONICA, S.A.	TEF.MC	Tecnología y Telecomunicaciones - Telecomunicaciones y Otros	Mercado Continuo	IBEX 35®, IGBM, IBEXTD®	X
TUBACEX, S.A.	TUB.MC	Mat.Basicos, Industria y Construcción - Mineral, Metales y Transformación	Mercado Continuo	IGBM	X
TUBOS REUNIDOS,S.A.	TRG.MC	Mat.Basicos, Industria y Construcción - Mineral, Metales y Transformación	Mercado Continuo	IGBM	X
UNICAJA BANCO, S.A.	UNI.MC	Servicios Financieros - Bancos y Cajas de Ahorro	Mercado Continuo	IBEX 35®, IGBM	X
URBAS GRUPO FINANCIERO, S.A.	UBS.MC	Servicios Inmobiliarios - Inmobiliarias y Otros	Mercado Continuo	IGBM	
VIDRALA S.A.	VID.MC	Bienes de Consumo - Otros Bienes de Consumo	Mercado Continuo	IGBM	X
VISCOFAN, S.A.	VIS.MC	Bienes de Consumo - Alimentación y Bebidas	Mercado Continuo	IGBM, IBEXTD®	X
VOCENTO, S.A.	VOC.MC	Servicios de Consumo - Medios de Comunicación y Publicidad	Mercado Continuo	IGBM	X

Obtenido de: La información expuesta en el sitio oficial de Bolsas y Mercados españoles, «Empresas cotizadas».

Tabla 3: Cantidades de muestras utilizadas para entrenar los modelos

Entradas	Muestras.totales
1	17347
2	17244
3	17141

Obtenido de: Elaboración propia.

Tabla 4: 10 Mejores y peores empresas según los resultados obtenidos de los cálculos de los indicadores

1			2			3		
TICKER	R <sup>2</sup>	MSE	TICKER	R <sup>2</sup>	MSE	TICKER	R <sup>2</sup>	MSE
OPDE.MC	0.4980	0.01400	OPDE.MC	0.3890	0.01190	OPDE.MC	0.4890	0.01220
OLE.MC	0.1410	0.20900	SOL.MC	0.3040	0.02260	ANE.MC	0.3490	0.00635
ANE.MC	0.1070	0.00662	ANE.MC	0.1970	0.00649	SOL.MC	0.3160	0.02080
DOM.MC	0.1030	0.00555	BKY.MC	0.1690	0.18700	BKY.MC	0.1980	0.20200
MVC.MC	0.1010	0.00959	TLGO.MC	0.1390	0.00876	LOG.MC	0.1930	0.00451
BKY.MC	0.0612	0.20700	OLE.MC	0.1080	0.20400	LDA.MC	0.1500	0.00351
TLGO.MC	0.0586	0.00916	AEDAS.MC	0.1070	0.00693	TLGO.MC	0.1170	0.00813
MRL.MC	0.0559	0.00623	MVC.MC	0.0990	0.00980	OLE.MC	0.0677	0.20700
AEDAS.MC	0.0505	0.00721	ENER.MC	0.0977	0.00984	CBAV.MC	0.0590	0.01160
NTH.MC	0.0497	0.01080	CCEP.MC	0.0865	0.00412	ENER.MC	0.0524	0.01120
EBRO.MC	-0.1130	0.00265	AENA.MC	-0.0116	0.00595	BBVA.MC	-0.1020	0.01060
VID.MC	-0.1180	0.00547	PRS.MC	-0.0147	0.02740	SAN.MC	-0.1060	0.01010
AMS.MC	-0.1200	0.00618	DIA.MC	-0.0251	0.02180	EBRO.MC	-0.1100	0.00262
ATRY.MC	-0.1370	0.01230	RIO.MC	-0.0251	0.00296	RED.MC	-0.1310	0.00389
ENER.MC	-0.1390	0.01250	VIS.MC	-0.0289	0.00345	MVC.MC	-0.1330	0.01250
PRM.MC	-0.1690	0.00731	TEF.MC	-0.0294	0.00542	RIO.MC	-0.1360	0.00329
APPS.MC	-0.2300	0.00809	ITX.MC	-0.0311	0.00778	ALB.MC	-0.1490	0.00521
R4.MC	-0.2790	0.00227	GRE.MC	-0.0457	0.02860	IBE.MC	-0.1520	0.00563
VIS.MC	-0.2860	0.00429	ATRY.MC	-0.1520	0.01250	HOME.MC	-0.1530	0.00740
LDA.MC	-0.3180	0.00543	LDA.MC	-0.1580	0.00430	VIS.MC	-0.1830	0.00398

Elaboración propia.

## Anexo.4 Códigos

A continuación, se presenta el código utilizado para la realización del procedimiento descrito en el desarrollo del trabajo.

### Datos

#### Obtención de Datos

Lo primero que se realizó fue cargar la tabla de las empresas.

```
R
empresas <- read_excel("data/000_empresas.xlsx")
```

Luego se extrajeron los ticks de las empresas.

```
R
ticks <- empresas |>
  select(TICKERS) |>
  pull()
```

Una vez almacenados los ticks de las empresas en la variable ticks se procedió a descargar los datos correspondientes a dichas empresas desde Yahoo Finance usando el paquete quantmod de Ryan y Ulrich (2023).

```
R
nombres_colum <-
c("Date", "Open", "High", "Low", "Close", "Volume", "Adjusted")
qmd_data <- list()
for (i in 1:length(ticks)) {
  tick <- ticks[i]
  value <- getSymbols(
    tick,
    from = "2000-01-02",
    to = "2023-03-01",
    auto.assign = F,
    periodicity = "monthly") |>
    as.data.frame()
  dates <- row.names(value)
  row.names(value) <- NULL
  value <- cbind(dates, value)
  names(value) <- nombres_colum
  qmd_data[[tick]] <- value
}
```

Con el objetivo de realizar un análisis exploratorio de los datos, se decidió realizar una evaluación visual de los datos históricos del precio ajustado para lo que se ejecutó:

```
R
lapply(qmd_data, function(x){
  x |>
  ggplot(aes(x=as.Date(Date), y=Adjusted))+
    geom_line(color="#065AD8")
})
```

Tras el análisis visual ejecutado con el fragmento de código anterior se percibió la existencia de precios constantes, así como cálculos erróneos en el precio ajustado correspondiente a los primeros años de algunas series. Con el objetivo de eliminar estas irregularidades se seleccionaron solo aquellas observaciones posteriores a enero del 2005.

```
R
returns_emps <- qmd_data |>
  lapply(function(x){
    emps <- x |>
      filter(Date >= "2005-01-31")
  })
```

Con el objetivo de determinar si los datos que habían sido importados contaban con valores faltantes se ejecutó el siguiente código:

```
R
na_values <- returns_emps |>
  sapply(function(x){
    na <- length(which(is.na(x)))
  })
emp_con_na <- which(na_values > 0)
```

Con el objetivo de solucionar el problema con respecto al incorrecto registro de los datos se decidió eliminar aquellas que no presentaran variaciones en los precios en más de 10 observaciones. Para lo que primero se computaron las rentabilidades ejecutando el siguiente código, mediante el cual además se eliminaron las series con valores faltantes.



```
R
returns_emps2 <- returns_emps[-emp_con_na] |>
  lapply(function(x){
    returns <- x |>
      select(Date, Adjusted) |>
      mutate(Return_Ad = Delt(Adjusted)[,1]) |>
      na.omit() |>
      select(Date, Return_Ad)
  })
```

Una vez computadas las rentabilidades se eliminaron aquellas series que presentaban en más de 10 observaciones rentabilidad 0, para lo que se ejecutó el siguiente código.

```
R
zero_values <- returns_emps2 |>
  sapply(function(x){
    zeros <- length(which(x[,2]==0))
  })
returns_emps3 <- returns_emps2[zero_values<10]
```

### Indicadores

A continuación, se expone el código utilizado durante el proceso expuesto en el sub-epígrafe *indicadores* del capítulo 2.

Primero se descargaron los datos del IBEX, se computaron las rentabilidades del precio ajustado del mismo y se seleccionaron los valores posteriores a enero del 2005.

```
R
#Importando IBEX
IBEXsel <- getSymbols(
  "^IBEX",
  from = "1990-01-01",
  to = "2023-03-01",
  auto.assign = F,
  periodicity = "monthly") |>
  as.data.frame()
dates <- row.names(IBEXsel)
row.names(IBEXsel) <- NULL
IBEXsel <- cbind(dates,IBEXsel)
names(IBEXsel) <- nombres_colum
```

```
R
# Calculando rentabilidad y seleccionando observaciones posteriores a
# enero del 2005.
IBEXsel <- IBEXsel |>
  mutate(Return_I = Delt(Adjusted)[,1]) |>
  na.omit() |>
  filter(Date >= "2005-01-31") |>
  select(Date, Return_I)
```

Luego se agregaron los valores de las rentabilidades del IBEX a las tablas de las rentabilidades de las acciones de las empresas seleccionadas, y se computaron y agregaron las variables listadas a continuación a cada una de las tablas:

- Volatilidad de la empresa
- Volatilidad del índice
- Correlación entre las rentabilidades de la empresa y el índice
- La Beta entre la empresa y el índice

```
R
returns_indc <- returns_emps3 |>
  lapply(function(x, ind = IBEXsel){
    emp <- x |>
    left_join(ind) |>
    mutate(
      VE = sqrt(cumsum((Return_Ad -
cummean(Return_Ad))^2)/1:length(Return_Ad)),
      VI = sqrt(cumsum((Return_I -
cummean(Return_I))^2)/1:length(Return_I)),
      Cor = cumsum((Return_Ad-cummean(Return_Ad))*(Return_I-
cummean(Return_I)))/(sqrt(cumsum((Return_Ad-
cummean(Return_Ad))^2))*sqrt(cumsum((Return_I-cummean(Return_I))^2)))
    )|>
    na.omit() |>
    mutate(
      Beta = (Cor*VE)/VI
    )
  })
```

### Vectores

A continuación, se expone el código utilizado durante el proceso expuesto en el sub-epígrafe *vectores* del epígrafe modelado del Capítulo 2.

El primer paso llevado a cabo para la ejecución del proceso explicado en el sub-epígrafe en cuestión fue crear una función que permitió obtener las muestras consecutivas para cada serie utilizada. La función expuesta a continuación, como ya se mencionó, permite obtener las muestras consecutivas de una serie, para lo que se utilizan los parámetros mencionados en el sub-epígrafe, número de observaciones de entradas y número de observaciones de salida, así como un parámetro condicional con el que se indica si el vector a crear es de entrada o de salida.

```
R
vector2dmaker <- function(vec, ent, sal, eos=T){
  if(eos==T){
    emp <- 1
    term <- (length(vec) - (ent+sal-1))
    ob <- ent
  }else{
    emp <- ent + 1
    term <- (length(vec)-sal+1)
    ob <- sal
  }

  vec2d <- sapply(emp:term,
                 function(x) vec[x:(x + ob-1)]) |>
    matrix(nrow = ob) |>
    t()

  return(vec2d)
}
```

A continuación, se muestra el código utilizado para la creación de los vectores de entrada correspondiente a cada una de las series. Para lo que primero se crearon dos funciones una para las entradas y otra para las salidas.

```

R
# Función que se utilizará para crear las entradas tridimensionales
input3dmaker <- function(x,inp,out){
  empre <- x
  series <- 2:dim(x)[2]
  for (i in series) {
    if(i==series[1]){
      vec3d <- vector2dmaker(empre[[i]],ent=inp,sal=out)
    }else{
      vec3d <- abind(vec3d,vector2dmaker(empre[[i]],ent=inp,sal=out),
along = 3)
    }
  }
  return(vec3d)
}

# Función que se utilizará para crear las salidas tridimensionales
output3dmaker <- function(x,inp,out){
  empre <- x[["Return_Ad"]]
  vec3d <- vector2dmaker(empre,ent=inp,sal=out,F)
  dim(vec3d) <- c(dim(vec3d),1)
  return(vec3d)
}

```

Luego se crearon las listas de vectores tridimensionales de entradas y salidas por empresa, ejecutándose el siguiente código otras dos veces con el objetivo de crear las listas `vecs3d2e` y `vecs3d3e` que corresponden a aquellos casos en los que se seleccionaron 2 y 3 entradas.

```

R
#Se define el horizonte temporal
ht <- 1

#Se definen las observaciones de entrada
oe <- 1

```

```
R
# Se crean los vectores de entrada 3d y el vector 2d de salida
# para tamaño de entrada 1
vecs3d1e <- list()
for(i in 1:length(returns_indc)){
  emp <- returns_indc[[i]]
  inps <- input3dmaker(emp, oe, ht)
  outs <- output3dmaker(emp, oe, ht)
  dates <- emp[(oe + ht):dim(emp)[1],1]
  id <- rep(names(returns_indc)[i],length(dates))
  tibblem <- tibble(
    Date = dates,
    ID = id,
    inputs = inps,
    outputs = outs
  )
  vecs3d1e[[names(returns_indc)[i]]] <- tibblem
}

```

### Modelado y entrenamiento

A continuación, se presenta el código utilizado durante el proceso descrito en los distintos sub-epígrafes del epígrafe *Modelado y entrenamiento*.

### Modelado

Para la creación de los modelos el primer paso a ejecutar es obtener la información de los vectores para los que se va a construir el modelo, lo que se hizo ejecutando el siguiente código:

```
R
data <- bind_rows(vecs3d1e)
data <- data |>
  arrange(Date)
inputsinfo <- data|>
  select(inputs) |>
  pull() |>
  dim()
outputsinfo <- data|>
  select(outputs) |>
  pull() |>
  dim()

```

```
R
# Definir parámetros
n_ob_pas <- inputsinfo[2]
n_variables <- inputsinfo[3]
n_ob_fut <- outputsinfo[2]
```

Luego se constituyó la estructura de los modelos con los aspectos descritos en [2.5.1 Modelado](#).

```
R
# Capa de entrada
inp <- layer_input(
  shape = c(NULL,n_ob_pas,n_variables))

# Capas ocultas
# - CNN
cnn <- inp |>
  layer_conv_1d(
    filters = 64,
    kernel_size = 1,
    activation = layer_activation_leaky_relu())
# - LSTM
lstm <- cnn |>
  layer_lstm(64)

# Capa de Salida
out <- lstm |>
  layer_dense(
    n_ob_fut*1)

# Juntar las capas para constituir el modelo
modelo <- keras_model(inp, out)
# Estableciendo parámetros de aprendizaje
modelo |>
  compile(loss = "mse", optimizer = optimizer_sgd(0.0005))
```

#### 📌 Nota

Puede encontrar modelos sin entrenar en la carpeta `data` del repositorio en el que se encuentra el presente trabajo. Los modelos fueron guardados usando la extensión `hdf5` y bajo los nombres `model1e`, `model2e` y `model3e`.

## Entrenamiento

El primer paso es definir la función a utilizar para el entrenamiento de los modelos. Esta función fue construida con el objetivo de emplear el método de entrenamiento descrito en [2.5.2 Entrenamiento](#). Como resultado esta función devuelve una lista que contiene las predicciones obtenidas y el modelo después de haber sido entrenado y tomará como entradas principales el tibble llamado `data`, constituido en el primer paso que se expone en este Anexo en la sección Modelado y el modelo además de otros argumentos que permite la utilización de la función con unos inputs principales que no sean los utilizados en el presente trabajo.

```
R
wfv_train <- function(x, modelo, seq_var_name, inp_var_name = "inputs",
  out_var_name = "outputs", progress_bar=T){

  predictions <- c()
  seq_val <- unique(x[[seq_var_name]])

  if(progress_bar){
    pb <- txtProgressBar(min = 0, max = length(seq_val), initial = 0,
  style = 3)
  }

  # Iteración que se ejecutará para cada valor único en la variable que
  # define la secuencia de los datos. Por ello es de vital importancia
  # que los datos en el tibble x se encuentren ordenados por la
  # variable de secuencia cuyo nombre se pasa a seq_var_name

  for (i in 1:length(seq_val)) {
    val_seq <- seq_val[i]
    # Extraer entradas y salidas correspondiente al periodo en la
    # variable de secuencia actual
    inputs <- x |>
      filter(!sym(seq_var_name) == val_seq) |>
      select(!sym(inp_var_name)) |>
      pull()
    outputs <- x |>
      filter(!sym(seq_var_name) == val_seq) |>
      select(!sym(out_var_name)) |>
      pull()
    outputs <- outputs[, ,1]
```

```
R
# Usar entradas para obtener predicciones para los periodos en la
# variable secuencia a excepción del primero
if(i > 1){
  pred <- modelo |>
    predict(inputs, verbose = 3)
  predictions <- rbind(predictions, pred)
}

# Entrenar el modelo
modelo |>
  fit(
    inputs,
    outputs,
    epochs = 1,
    batch_size = 10,
    shuffle = F,
    verbose = 0)

if(progress_bar){
  setTxtProgressBar(pb,i)
}

}

if(progress_bar){
  close(pb)
}

results <- list()
results[['predicciones']] <- predictions
results[['modelo']] <- modelo
return(results)
}
```

Una vez creada la función se obtuvieron las predicciones utilizando el siguiente código:

```
R
resultados <- wfv_train(data,model,'Date')
predicciones1e <- resultados$predicciones
```



**Nota**

Puede encontrar modelos entrenados en la carpeta `data` del repositorio en el que se encuentra el presente trabajo. Los modelos fueron guardados usando la extensión `hdf5` y bajo los nombres `model1etd`, `model2etd` y `model3etd`.

Como se explica en [2.6.1 Predicciones](#) además de las predicciones obtenidas por los modelos se computaron predicciones obtenidas a partir del uso de la media aritmética, para comparar con las obtenidas con los modelos. Para el computo de estas predicciones se creó la siguiente función:

```
R
wfv_means <- function(x, seq_var_name, inp_var_name = "inputs",
out_var_name = "outputs", id_var_name, progress_bar=T){

  means <- c()
  seq_val <- unique(x[[seq_var_name]])

  if(progress_bar){
    pb <- txtProgressBar(min = 0, max = length(seq_val), initial = 0,
style = 3)
  }

  for (i in 1:length(seq_val)) {
    val_seq <- seq_val[i]
    inputs <- x |>
      filter(!sym(seq_var_name) == val_seq) |>
      select(!sym(inp_var_name)) |>
      pull()
    inputspred <- x |>
      filter(!sym(seq_var_name) == val_seq) |>
      select(!sym(inp_var_name)) |>
      pull()
    outputs <- x |>
      filter(!sym(seq_var_name) == val_seq) |>
      select(!sym(out_var_name)) |>
      pull()
    outputs <- outputs[, ,1]

    ids <- x |>
      filter(!sym(seq_var_name) == val_seq) |>
      select(!sym(id_var_name)) |>
```

```

pull()

if(i==1){
  dfmeans <- inputs[,1] |>
  as.data.frame() |>
  cbind(ID = ids)
}else{
  dfmeansupd <- inputs[,dim(inputs)[2],1] |>
  as.data.frame() |>
  cbind(ID = ids)
  names(dfmeansupd)[1] <- paste0("V",dim(dfmeans)[2])
  idsdf <- unique(c(ids, dfmeans[[id_var_name]]))
  idsdf <- data.frame(ID = idsdf)
  dfmeansupd <- dplyr::left_join(idsdf, dfmeansupd, by = "ID")
  ifelse(
    dim(dfmeansupd)[1] > dim(dfmeans)[1],
    dfmeans <- dplyr::left_join(dfmeansupd, dfmeans, by = "ID"),
    dfmeans <- dplyr::left_join(dfmeans, dfmeansupd, by = "ID")
  )
}

if(i > 1){
  MEANS <- dfmeans |>
  rowwise() |>
  mutate(
    means = mean(c_across(-!!sym(id_var_name)), na.rm = T) |>
    slice(match(ids,!!sym(id_var_name))) |>
    pull(means) |>
    as.matrix()
  )
  means <- rbind(means, MEANS)
}

if(progress_bar){
  setTxtProgressBar(pb,i)
}

}

if(progress_bar){
  close(pb)
}

return(means)
}

```

Una vez creada la función se obtuvieron las predicciones utilizando el siguiente código:

R

```
meanse1 <- wfv_means(data, 'Date', id_var_name = "ID")
```

### Nota

En adición a lo expuesto con anterioridad se crearon dos funciones `getconfig` y `plot_modelk`, en el archivo `.Rprofile` del repositorio en el que se encuentra este trabajo, que permiten graficar la estructura de los modelos mediante el uso del paquete `lannone` (2023), como se ve en la [Figura 17](#). El código a utilizar sería:

R

```
# Las funciones están creadas para graficar las estructuras utilizadas en
# el presente trabajo.
model |>
  getconfig() |>
  plot_modelk() |>
  grViz()
```

El procedimiento expuesto en las secciones Modelado y Entrenamiento del presente anexo fue repetido para construir los 10 modelos realizados a partir de cada grupo de vectores tridimensionales, sustituyendo en el primer código expuesto la llamada a `vecs3d1e` por `vecs3d2e` y `vecs3d3e`, según el grupo de vectores tridimensionales utilizado.

### Resultado

A continuación, se presenta el código utilizado durante el proceso descrito en los distintos sub-epígrafes del epígrafe [Resultado](#).

### Predicciones

El análisis expuesto en [2.6.1 Predicciones](#) fue realizado a partir de gráficas (vea [Gráfica 8](#), [Gráfica 9](#) y [Gráfica 10](#)), en las que se recogen los valores de los indicadores  $MSE$  y  $R^2$  para cada una de las estructuras probadas.

El primer paso para la obtención de estas gráficas fue el de computar los indicadores, para cada periodo de tiempo, para cada una de las predicciones obtenidas a partir de los distintos modelos construidos con cada estructura. Esto se realizó mediante el siguiente código.

```

R
#Extraer salidas reales
salidas <- data |>
  filter(
    Date > data$Date[1]
  ) |>
  select(outputs) |>
  pull()
salidas <- salidas[,1]

#Computar indicadores MSE y R2
indicadores <- data |>
  filter(Date > data$Date[1]) |>
  cbind(predicciones = predicciones1e[,1]) |>
  cbind(means = meanse1) |>
  mutate(salidas = salidas) |>
  select(Date, predicciones, means, salidas) |>
  group_by(Date) |>
  summarise(
    r2 = 1 - (sum((salidas - predicciones)^2)/sum((salidas -
means)^2)),
    mse = mse(predicciones, salidas),
  )

```

Los diferentes indicadores computados para cada uno de las 10 modelos entrenados con cada una de las estructuras fueron guardados en una lista llamada `list_indicadores`. Esto se realizó utilizando el siguiente código:

```

R
list_indicadores[["indicadores1"]] <- indicadores

```

Una vez realizado esto se obtiene una lista que contiene 10 data frames (`indicadores1, ..., indicadores10`), los cuales a su vez contienen los valores de los de  $MSE$  y  $R^2$  de las predicciones obtenidas por los modelos de RNA para cada una de las empresas agrupados por fecha. Por lo que luego se construyó la gráfica mediante el uso del siguiente código.

```

R
# Agrupar la información de las distintas construcciones en un solo
data frame
indi_graf_data <- do.call(cbind,list_indicadores)

# Obtener los resultados medios, para cada periodo de tiempo,
# usando las distintas construcciones
indi_graf_data |>
  rowwise() |>
  mutate(
    Date = `indicadores1.Date`,
    meanmse = mean(c_across(contains("mse"))),
    meanr2 = mean(c_across(contains("r2")))
  ) |>
  select(
    Date, meanmse, meanr2
  ) |>
  # Graficar
  mutate(
    Date = as.Date(Date) |>
  ggplot(aes(x = Date, group = 1)) +
  geom_line(aes(y = meanmse, color = "MSE")) +
  geom_line(aes(y = meanr2, color = "R2")) +
  scale_color_manual(values = c("blue", "green")) +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(x = "Fecha", y = "Indicadores", color = "Indicadores")

```

Además de las gráficas se utilizó también en el análisis de los resultados la [Tabla 4](#), en la que se encuentran las empresas que obtuvieron mejores y peores indicadores para cada estructura, para la obtención de estos datos se usó el siguiente código:

```

R
indicadores_X_emp <- data |>
  filter(Date > data$Date[1]) |>
  cbind(predicciones = predicciones1e[,1]) |>
  cbind(means = meanse1) |>
  mutate(salidas = salidas) |>
  select(Date, predicciones, means, salidas, ID) |>
  group_by(ID) |>
  summarise(
    r2 = 1 - (sum((salidas - predicciones)^2)/sum((salidas -
means)^2)),
    mse = mse(predicciones, salidas)
  ) |>
  select(ID, r2, mse)

```

Al igual que los indicadores computados por fecha para guardar los indicadores computados por empresa se creó una lista denominada `list_indic_emp`. Luego de haber almacenado los 10 data frames de indicadores por empresa en la lista se extrajeron las empresas con los mejores y peores resultados mediante el siguiente código:

```
R
# Agrupar la información de las distintas construcciones en un
# solo data frame
ind_emp_t <- do.call(rbind, list_indic_emp)

# Computar los R2 y MSE medios por empresa
ind_emp_t <- ind_emp_t |>
  group_by(ID) |>
  summarize(
    r2 = mean(r2),
    mse= mean(mse)) |>
  ungroup() |>
  arrange(desc(r2))

# Obtener las 10 empresas con mejores y peores indicadores
mejores10 <- head(ind_emp_t,10)
peores10 <- tail(ind_emp_t,10)
```

Y mediante la utilización de las variables anteriores y el uso de las funciones `rbind()` y `cbind` fue como se creó la [Tabla 4](#).

### Composición de carteras

En esta sección se explica como se realizó el análisis del comportamiento de los resultados obtenidos por las distintas carteras (ver [Gráfica 11](#), [Gráfica 12](#) y [Gráfica 13](#)). Para ello es necesario primero obtener la composición de las carteras, por fecha, a partir de las predicciones obtenidas mediante el uso de las medias aritméticas y los modelos de RNA.

Para el cálculo de la composición de las carteras se usó el paquete de R llamado `quadprog` ( Berwin A. Turlach R port by Andreas Weingessel <Andreas.Weingessel@ci.tuwien.ac.at> Fortran contributions from Cleve Moler dpodi/LINPACK) (2019)), a continuación, se muestra el código utilizado para hallar la composición de carteras a partir de las predicciones de la media:

```
R
# Se creo un data frame en el que se guardó toda la información:
# - Valores del IBEX, como índice de referencia
# - Valores de las predicciones, tanto las obtenidas por el modelo de
#   RNA como por las medias aritméticas

DATA <- data |>
  left_join(IBEXsel, by = "Date") |>
  mutate(IBEX = Return_I) |>
  arrange(Date) |>
  filter(
    Date > data$Date[1]
  ) |>
  mutate(predicciones = predicciones1e[,1]) |>
  mutate(
    Real = salidas,
    RNA = predicciones,
    Means = meanse1
  ) |>
  select(Date, Real, IBEX, RNA, Means, ID)

# A partir del data frame DATA se crearon:
# - Un data frame cuyas columnas son los datos reales de cada una
#   de las empresas para cada uno de los periodos de tiempo para
#   los que se obtuvieron predicciones.
# - Un data frame cuyas columnas son los datos obtenidos mediante
#   el uso de las medias aritméticas de cada una de las empresas
#   para cada uno de los periodos de tiempo para los que se
#   obtuvieron predicciones.

pvtReal <- DATA |>
  select(Date, Real, ID) |>
  pivot_wider(
    names_from = ID,
    values_from = Real
  )

pvtMeans <- DATA |>
  select(Date, Means, ID) |>
  pivot_wider(
    names_from = ID,
    values_from = Means
  )
```

```

R
# Se creo el data frame en el que se guardará la composición de las
# carteras para cada uno de los periodos para los que se obtuvo
# predicción
weightsm <- data.frame()

# Iteración mediante la cual se halla la composición de las carteras

pb <- txtProgressBar(min = 0, max = length(unique(data$Date)[-1]),
initial = 0, style = 3)

for (i in 1:length(unique(data$Date)[-1])) {
  if(i>1){

    # Se crea el data frame que comprende los datos a utilizar
    # para hallar la composición de la cartera, este está creado
    # por los datos reales hasta la fecha y la previsión del
    # siguiente periodo
    datamQP <- pvtReal |>
      filter(Date < unique(data$Date)[-1][i]) |>
      rbind(pvtMeans |>
        filter(Date == unique(data$Date)[-1][i])
      )

    # Elimina aquellas empresas que no tengan ni datos reales o
    # de previsión
    nare <- which(is.na(datamQP[dim(datamQP)[1],]))
    naremo <- which(is.na(datamQP[(dim(datamQP)[1]-1),]))
    nare <- c(nare,naremo)
    nare <- unique(nare)
    if(length(nare) != 0){
      carteram <- datamQP[, - nare]
    }else{
      carteram <- datamQP
    }

    # Extrae las previsiones
    returnm <- carteram[dim(carteram)[1], -1] |>
      as.matrix() |>
      t()

    # Calcula la matriz de covarianza
    covmm <- cov(carteram[, -1], use = "complete.obs")
    npcovmm <- nearPD(covmm)$mat |>
      as.matrix()
    # Extrae el número de empresas
    n <- ncol(npcovmm)
  }
}

```



```

R
# Halla la composición de la cartera
qp_outm <- solve.QP(
  Dmat = 2*npcovmm,
  dvec = rep(0,n),
  Amat = cbind(-1, diag(n)),
  bvec = c(-1, rep(0,n)),
  meq = 1)
qp_outm <- qp_outm$solution
qp_outm <- floor(qp_outm*100)/100
for(j in 1:length(qp_outm)){
  if(qp_outm[j] < 0.001){
    qp_outm[j] <- 0
  }else{}
}

# Guarda la composición de la cartera
names(qp_outm) <- names(carteram[, -1])
weightsm <- bind_rows(weightsm, qp_outm)
}

setTxtProgressBar(pb,i)
}

close(pb)

# Sustituir los pesos y observaciones reales con valores faltantes
# con cero
pvtReal[is.na(pvtReal)] <- 0
weightsm[is.na(weightsm)] <- 0

```

Luego para hallar la rentabilidad de la cartera se multiplicaron las composiciones por las rentabilidades reales, se asumió que se invertía uno en el primer periodo y se realizó una sumatoria acumulativa a lo largo de los valores para obtener el comportamiento de la rentabilidad a lo largo del tiempo.

```
R
# Hallando las rentabilidades de las carteras conformadas a partir
# de las predicciones de la media aritmética

return_CM <- weightsm * pvtReal[-1,-1]
return_CM <- rowSums(return_CM)
return_CM <- c(1,return_CM)
return_CM <- data.frame(
  Date = pvtReal[,1],
  Mean = return_CM
)
```

Los mismos pasos que se realizaron para hallar el comportamiento de la rentabilidad de las carteras a partir de las medias aritméticas se realizaron para hallar el comportamiento a partir de las predicciones obtenidas por el modelo de RNA como se ve en el código a continuación.

```
R
# A partir del data frame DATA se creó un data frame cuyas columnas
# son los datos obtenidos mediante el uso del model de RNA de cada
# una de las empresas para cada uno de los periodos de tiempo para
# los que se obtuvieron predicciones.

pvtRNA <- DATA |>
  select(Date, RNA, ID) |>
  pivot_wider(
    names_from = ID,
    values_from = RNA
  )

# Se creo el data frame en el que se guardara la composición de
# las carteras para cada uno de los periodos para los que se
# obtuvo predicción
weightse <- data.frame()

# Iteración mediante la cual se halla la composición de las carteras

pb <- txtProgressBar(min = 0, max = length(unique(data$Date)[-1]),
  initial = 0, style = 3)

for (i in 1:length(unique(data$Date)[-1])) {
  if(i>1){
```

R

```

# Se crea el data frame que comprende los datos a utilizar
# para hallar la composición de la cartera, este está creado
# por los datos reales hasta la fecha y la previsión del
# siguiente periodo
dataeQP <- pvtReal |>
  filter(Date < unique(data$Date)[-1][i]) |>
  rbind(pvtRNA |>
    filter(Date == unique(data$Date)[-1][-1][i])
  )
# Elimina aquellas empresas que no tengan ni datos reales o
# de previsión
nare <- which(is.na(dataeQP[dim(dataeQP)[1],]))
naremo <- which(is.na(dataeQP[(dim(dataeQP)[1]-1),]))
nare <- c(nare,naremo)
nare <- unique(nare)
if(length(nare) != 0){
  carterae <- dataeQP[, - nare]
}else{
  carterae <- dataeQP
}

# Extrae las previsiones
returne <- carterae[dim(carterae)[1], -1] |>
  as.matrix() |>
  t()

# Calcula la matriz de covarianza
covme <- cov(carterae[, -1], use = "complete.obs")
npcovme <- nearPD(covme)$mat |>
  as.matrix()
# Extrae el número de empresas
n <- ncol(npcovme)

# Halla la composición de la cartera
qp_oute <- solve.QP(
  Dmat = 2*npcovme,
  dvec = rep(0,n),
  Amat = cbind(-1, diag(n)),
  bvec = c(-1, rep(0,n)),
  meq = 1)
qp_oute <- qp_oute$solution
qp_oute <- floor(qp_oute*100)/100
for(j in 1:length(qp_oute)){
  if(qp_oute[j] < 0.001){
    qp_oute[j] <- 0
  }else{}
}

```

```
}  
R  
  # Guarda la composición de la cartera  
  names(qp_oute) <- names(carterae[, -1])  
  weightse <- bind_rows(weightse, qp_oute)  
}  
  
  setTxtProgressBar(pb,i)  
}  
  
close(pb)  
  
# Sustituir los pesos con valores faltantes con cero  
weightse[is.na(weightse)] <- 0
```

Luego para hallar la rentabilidad de la cartera se multiplicaron las composiciones por las rentabilidades reales, se asumió que se invertía uno en el primer periodo y se realizó una sumatoria acumulativa a lo largo de los valores para obtener el comportamiento de la rentabilidad a lo largo del tiempo.

```
R  
# Hallando las rentabilidades de las carteras conformadas a partir  
# de las predicciones del modelo RNA  
  
return_CRNA <- weightse * pvtReal[-1,-1]  
return_CRNA <- rowSums(return_CRNA)  
return_CRNA <- c(1,return_CRNA)  
return_CRNA <- data.frame(  
  Date = pvtReal[,1],  
  RNA = return_CRNA  
)
```

Luego, al igual que con los indicadores se creó una lista `list_ret_RNA` en la que se guardaron los data frames de los distintos modelos construidos con cada una de las estructuras. Después se ejecutó el siguiente código para obtener la gráfica.

```

R
# Hallando el comportamiento de las rentabilidades del IBEX para el
# periodo

IBEXvals <- IBEXsel |>
  filter(Date > unique(data$Date)[2]) |>
  select(2) |>
  pull()
IBEXvals <- c(1, IBEXvals)

data_rent_RNA <- do.call(cbind,list_ret_RNA)
data_rent_RNA <- data_rent_RNA |>
  mutate(
    Date = RNA1.Date,
    IBEX = IBEXvals,
    Means = return_CM$Mean) |>
  mutate_at(vars(contains(".RNA")), ~ cumsum(.)) |>
  mutate(
    IBEX = cumsum(IBEX),
    Means = cumsum(Means)) |>
  group_by(Date) |>
  summarize(
    meanRNA = mean(c_across(contains(".RNA"))),
    max_y = max(c_across(contains(".RNA"))),
    min_y = min(c_across(contains(".RNA"))),
    min_5 = unname(quantile(c_across(contains(".RNA")),0.05)),
    max_95 = unname(quantile(c_across(contains(".RNA")),0.95)),
    IBEX = IBEX,
    Means = Means)
data_rent_RNA |>
  mutate(
    Date = as.Date(Date)) |>
  ggplot(aes(x = Date)) +
    geom_ribbon(aes(ymin = min_y, ymax = min_5), fill = "blue", alpha =
0.3) +
    geom_ribbon(aes(ymin = max_y, ymax = max_95), fill = "blue", alpha =
0.3) +
    geom_ribbon(aes(ymin = min_5, ymax = max_95), fill = "blue", alpha =
0.6) +
    geom_line(
      aes(y = meanRNA, color = "Media RNA1"),
      linetype = "dashed") +
    geom_line(aes(y = max_y, color = "blue") +
    geom_line(aes(y = min_y, color = "blue") +
    geom_line(aes(y = max_95, color = "blue") +
    geom_line(aes(y = min_5, color = "RNA1")) +
    geom_line(aes(y = IBEX, color = "IBEX")) +

```

```
geom_line(aes(y = Means, color = "Medias")) +
scale_color_manual(
  values = c(
    "Media RNA1"="blue",
    "RNA1" = "blue",
    "IBEX" = "red",
    "Medias" = "green")) +
guides(
  color = guide_legend(
    override.aes = list(
      linetype = c("solid","dashed","solid","solid")))))+
labs(x = "Fecha",
     y = "Rentabilidades",
     color = "Leyenda")+
theme_minimal()
```