UNIVERSITY OF A CORUÑA

Faculty of Economics and Business

Final Master’s Project

Application of artificial neural networks and quadratic programming in portfolio management

David Díaz Rodríguez

Tutor: Xosé Manuel Martínez Filgueira

Master's Degree in Banking and Finance

Academic year 2022/2023

Final Master's Project presented at the Faculty of
Economics and Business of the University of A Coruña
to obtain the Master´s Degree in Banking and Finance

Thanks

I want to express my sincere gratitude to all the people who contributed significantly to the completion of this final master's thesis. Without your support, guidance and encouragement, this achievement would not have been possible.

I would like to thank the Xunta de Galica for providing me with the resources to conduct master's studies as a beneficiary of the Excellence Juventud Exterior Scholarship.

I would like to thank Universidade da Coruña for the knowledge acquired while studying the master's degree in Banking and Finance. Its commitment to academic excellence has been a constant source of inspiration.

I am deeply grateful to my tutor Xosé Manuel Martínez Filgueira for his expert guidance and valuable suggestions throughout this process. His knowledge and dedication were essential to give direction and quality to this work.

My thanks are extended to my classmates and friends who provided a space for rich discussions and valuable contributions that contributed to the development of this work.

I also want to express my gratitude to my family for their constant emotional support and understanding during the challenging moments of this academic process.

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

Words: 13027

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.

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 carteiras 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 carteiras 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 carteiras, carteiras, redes neuronais artificiais, programación cuadrática, series temporales financeiras, predición de prezos, composición da carteira.

Content

Figures List

Graphs List

Tables List

1. Introduction

In the financial field, efficient portfolio management is a crucial task for investors and asset managers, since it seeks to maximize returns and minimize the risks associated with investments. In recent years, the field of artificial intelligence and machine learning has experienced remarkable progress, which has allowed the application of innovative techniques to improve the financial decision-making process.

This work focuses on the integration of two powerful tools: artificial neural networks and quadratic programming, to address the challenge of portfolio management. The combination of these techniques offers a robust and promising approach to financial time series forecasting and portfolio composition in a highly dynamic and complex financial environment.

The development of the work is structured in several fundamental sections to comprehensively address the topic. First, a detailed characterization of the financial time series is conducted, examining its essential characteristics and properties to better understand the behaviour of asset prices.

Next, the potential of artificial neural networks in time series forecasting is explored. Background on the use of these networks in this context is presented and two widely used architectures are highlighted: convolutional neural networks and Long Short-Term Memory (LSTM) networks, both with the ability to capture complex patterns in financial data.

The section on portfolio composition addresses the problem and presents various techniques applied in asset management. It is here where quadratic programming is introduced as a relevant and efficient tool for the optimal construction of investment portfolios.

Obtaining accurate and relevant data is crucial for any financial analysis and work with Machine Learning algorithms. The methodology applied to obtain data is described, and how some of the most common indicators used in finance were computed to use as descriptive variables of the problem in conjunction with historical data. It also exposes how the vectors that will be used in the modelling and training of neural networks are structured.

In the last sections, the modelling and training process is discussed, which involves the proper configuration of the neural networks and the implementation of quadratic programming to obtain optimal results. Finally, the results obtained are presented, including the predictions generated by the neural networks and the composition of recommended portfolios, thus demonstrating the effectiveness of the proposed methodology in the management of financial portfolios.

Taken together, this work seeks to provide a comprehensive and updated view of the use of artificial neural networks and quadratic programming in portfolio management, highlighting their potential as an option to improve financial decision-making and provide investors with a valuable tool for Optimize your investment strategies in a changing and competitive environment.

2. Work development

The present work delves into the field of machine learning and artificial intelligence, specifically in the combined use of artificial neural networks and quadratic programming. This powerful synergy seeks to offer an effective and sophisticated solution for forecasting financial time series and the optimal composition of investment portfolios. Through the application of these techniques, the aim is to improve financial decision-making and maximize returns, while minimizing the risks associated with investments. In the following sections, the various stages of the process will be explored in detail, from the characterization of time series and the functioning of neural networks to the implementation of quadratic programming in the construction of efficient portfolios.

2.1. Characterization of financial time series

This heading is divided into two sub-headings. The first presents the fundamental concepts related to time series, including their characteristics, components, and classifications, and establishes their relationship with the analysis of time series and forecasting, which is the objective of this report. The second sub-heading delves into the characteristics of share prices, starting with a general description and moving towards more specific aspects, as well as the importance of the structure in which the data related to these are found.

2.1.1. Time series and their characteristics

Time series are a type of stochastic process that is characterized by ordering random variables according to time. This means that each moment is associated with a value of the variable that depends on chance and that can change over time. According to Ruiz (2011), a stochastic process is “a collection or family of random variables, ordered according to a subscript that is usually time” (p.01). The analysis of the time series can have different purposes, such as describing the behaviour of the variables or predicting or forecasting their future values, which is especially relevant for financial series.

Time series analysis is a statistical tool that allows studying the behaviour of a variable over time. However, there is no single consensus on the components that should be considered in this type of analysis. Some authors, such as Kocenda and Cerný (2017) and Anderson et al. (2017), propose that time series can be decomposed into three components: trend, seasonality, and noise. Other authors, such as Dodge (2008) and Espallargas and Solís (2012), suggest that a fourth component should be added: the cycle. Finally, there are authors who suggest that time series can have up to five components, these are the cases of IBM (2021) and Chirinos (2018).

Trend: It is the long-term pattern of change that is observed in a series of data. It can be defined as the general and persistent direction of the variations of the series over time. It can be classified as positive (*Graph 1*), negative (*Graph 2*) or null (*Graph 3*), depending on whether the series increases, decreases or remains constant in the long term. The trend can be identified by graphical analysis or by statistical methods. This component is important to understand the historical behaviour and project the future of a series of data, it is common in the different criteria mentioned.

Seasonality: Also called regular cyclical variation: It refers to the variation corresponding to the movements of the series that occur every certain period of time, *Graph 4*. This component is, like the trend, common in the aforementioned criteria. Differentiating in that those authors who expose four and five components call seasonality the periodic variations corresponding to periods less than or equal to one year (such as daily, weekly, monthly, or annual periodicity), while the periodic variations corresponding to longer periods. They contemplate a component called cyclical variations. Therefore, to determine the seasonality of a time series, it is necessary to analyse them in a period of no less than two years.

One component that cannot be explained by the other elements of the time series is the irregular variation or error. This component is also known as random variation, noise, or residual, and is shown in *Graph 4*. Irregular variation is common in all three criteria mentioned above. Some authors distinguish between irregular variation, which is occasional and random, and atypical variation, which is caused by isolated events that alter the behaviour of the series. Atypical variation can be classified into several types: additive, innovation, level change, transient, additive seasonality, and local trend.

One way to categorize the time series is according to the degree of variability that they present over time. According to what was exposed in Villagarcía (2006), it is possible to distinguish between homoscedastic and heteroscedastic series. The homoscedastic series are those that maintain a constant range of variation, as shown in *Graph 3*. On the contrary, heteroscedastic series are those that change the range of variation, increasing or decreasing its amplitude, as illustrated in *Graph 1* and *Graph 2*.

A key concept in time series analysis is that of stationarity. A time series is stationary when its statistical properties, such as the mean, variance, and covariance, do not change with time. This implies that the series does not present a trend, cycles, or seasonality. As Castillo and Varela (2010), Villavicencio (2010) and Ruiz (2011) point out, stationarity is a necessary condition to be able to predict the future behaviour of a time series using statistical techniques. An example of a stationary time series is shown in *Graph 3*.

Financial time series present heteroscedasticity, that is, variances that change over time. This implies that they are not stationary and that their behaviour depends on external factors. To verify the stationarity of a time series, different methods can be used, such as the correlogram, which shows the autocorrelation and partial autocorrelation functions of the series, or unit root tests, such as Dickey Fuller’s or Phillips Perron’s. , which evaluates the null hypothesis that the series has a unit root. These methods are explained in more detail in Castillo and Varela (2010), Villavicencio (2010) and Ruiz (2011). The *Graph 5* illustrates an example correlogram for a financial time series.

2.1.2. Pricing Features

Investing in stocks or any other asset listed on the stock market is a complex and challenging task, requiring a thorough understanding of market trends and fluctuations. At the core of this understanding is the ability to analyse and interpret stock market price data, providing key insights into the behaviour of market participants and the factors that drive market movements. The purpose of this sub-section is to provide a comprehensive overview of the stock price environment and how they are commonly represented, pointing out the most important aspects for the application of the techniques that will be explored in the following sections.

As explained in CNMV stock exchanges are organized markets where shares and other securities are traded, such as fixed income, warrants, certificates, and exchange-traded funds. In BME it is stated that, in Spain, there are four traditional stock exchanges (Madrid, Barcelona, Bilbao and Valencia) that are part of the holding Stocks and Spanish Market (hereinafter, BME, for its acronym in Spanish, Bolsas y Mercados Españoles), which also integrates other segments and trading, clearing and settlement systems values. Being, as explained in CNMV, the Spanish Stock Market Interconnection System (hereinafter, SIBE, for its acronym in Spanish, Sistema de Interconexión Bursátil Español) is the platform that allows continuous and electronic trading of all securities admitted to trading on the four Spanish stock.

As CNMV exposes, shares are transferable securities that represent a proportional part of the share capital of a public limited company, and their holders are proprietary partners of the same. Shares may be traded on stock exchanges or other authorized secondary markets.

From what was stated in Mitchell (2020), Pinset (2021) and C. Team (2023) it can be concluded that, to explain the price of a company's shares, the following factors can be considered:

- The supply and demand of shares in the market: if there are more buyers than sellers, the price will rise and vice versa. This depends on the expectations and confidence of investors in the future of the company.

- Changes in the management or production of the company: if the company improves its efficiency, its profitability or its innovation, the price of its shares may increase. On the contrary, if the company has internal problems, loses competitiveness, or is affected by external crises, the price may fall.
- The company's reputation: If the company has a good public image, is associated with successes or achievements, or receives good ratings from analysts, its share price may rise. Conversely, if the company is embroiled in scandals, lawsuits, or controversies, or receives poor ratings from analysts, the price may drop.

In the texts Pinset (2021), T. I. Team (2022) and C. Team (2023) also point out the importance of differentiating the price of a company or its share from its intrinsic value. Being able to summarize considering what is indicated in these texts and what has been previously stated that the price of a company or action is what buyers and sellers are willing to pay for it at a given moment, while the intrinsic value of a company or action depends largely on the methodology used to value the companies and the objectives of the evaluator.

Once the environment in which share prices are found has been contextualized in a general way and some of the factors that may affect them have been explained, the structure in which these data usually appear is explained below. Generally, the prices of the shares are registered periodically (daily, weekly, monthly, annually, etc.). registering for each period the opening price, the highest price, the lowest, the closing price, the volume, and the adjusted closing price, see *Table 1*.

From what was exposed in Barone (2022), Chen (2022), Downey (2022), Hayes (2021) and Ganti (2020) it can be understood that:

- The opening price is the first price at which a financial asset trades in a trading session. This price may be different from the closing price of the previous session, as there may be changes in supply and demand during the period when the market is closed. The opening price usually indicates the tone or trend of the market for that day.
- The highest price is the highest price at which a financial asset trades in a trading session. This price reflects the highest level of buyer interest for that asset on that day. The higher price can be an indicator of an asset's strength or weakness, as well as its volatility.

- The lowest price is the lowest price at which a financial asset trades in a trading session. This price reflects the minimum level of interest of sellers for that asset on that day. The lower price can be an indicator of an asset's pressure or resistance, as well as its volatility.
- The closing price is the last price at which a financial asset is traded in a trading session. This price is the one used to calculate the market value of that asset at the end of the day. The closing price is usually the most important for investors, as it summarizes the result of the day's operations and shows the direction of the market.
- Volume is the number of units of a financial asset traded in a trading session. Volume shows the level of activity or liquidity of a market or an asset. Volume often accompanies price movements, as it indicates the degree of consensus or divergence among market participants.
- Adjusted closing price is the closing price of a financial asset that is changed to consider events such as dividends, splits, mergers, or acquisitions that affect the value of the asset. The adjusted closing price allows you to compare the historical performance of an asset with greater precision and consistency.

Based on what was stated in Hayes (2021) and Ganti (2020), it is understood that the difference between the closing price and the adjusted closing price is of significant importance, since the former can give a distorted image of the performance of a share throughout the year. while the second reflects the actual value of the stock after adjusting for the factors that alter it.

For example, a company's board of directors may decide to divide the company's shares 3 by 1. Thus, the company's outstanding shares increase by a multiple of three, while its share price is divided by three. Let's say a stock closed at $300 the day before your stock split. In this case, the closing price is adjusted to $100 ($300 divided by 3) per share to maintain a consistent standard of comparison. Similarly, all other previous closing prices for that company would be divided by three to get the adjusted closing prices. Ganti (2020)

Due to this, the adjusted closing price is better for the application of time series analysis techniques, since it allows comparing the behaviour of a stock over time without the distortions caused by corporate events. The time series most commonly used in market price analysis studies is that made up of returns calculated from the adjusted closing price.

2.2. Artificial neural networks in the forecast of time series

This epigraph is divided into three sub-headings. The first deals with the background to the use of artificial neural networks for working with time series, more specifically in forecasting. In the second and third sub-headings, the operation of two of the RNA layer structures used in this work are exposed, these being the CNN and the LSTM.

2.2.1 Background on the use of artificial neural networks in time series forecasting

In Chollet and Allaire (2018) it is stated that the ANN environment is made up of artificial intelligence (hereinafter IA), machine learning or automated learning (hereinafter ML) and deep learning or deep learning (hereinafter DL), *Figure 1*. Therefore, it is of vital importance to know the aspects of these fields that are closely related to ANN and that are briefly explained below.

“Making a machine behave in such a way that a human would be called intelligent” (McCarthy et al. (2006), p.11) is the first definition given to the AI problem. With the aim of solving this problem, the first AI emerged, the so-called symbolic AI.

As explained by Haykin (1998), Banda (2014) and Chollet and Allaire (2018), these early AIs involved hardcoded rules created by programmers. With the aim of achieving that these rules were automatically learned by the machines when observing the data, a new stage emerged in the development of AI, the so-called ML. This new stage gives rise to the emergence of a new form of programming, differentiating from the classic, in that, in this, the programmers introduce the data and the expected responses to them, and the computers are capable of generating the rules, *Figure 2*.

So, it is understood that ML models try to find appropriate representations for your input data: transformations of the data that make it more amenable to the task at hand. In DL, which is a specific sub-field of ML, these data representations are modelled through architectures composed of successive layers, which are called RNA Chollet and Allaire (2018).

After studying what was exposed in Haykin (1998), Larrañaga (2007), Banda (2014) and Chollet and Allaire (2018) about ANN, it can be affirmed that they are inspired by the functioning of the human brain, these texts confirm and agree that three types of ANN can be distinguished layers: input, output and hidden. An input layer is composed of neurons that receive the input vectors. An output layer is made up of neurons that, during training, receive the output vectors and then generate the response. A hidden layer is connected to the environment through the input and output layers, this type of hidden layer processes the received input to obtain the corresponding output, *Figure 3*.

One of the applications of ANN is the forecasting of time series. whose objective is to predict the future values of variables based on their past observations. As discussed previously, financial time series are often nonlinear, noisy, chaotic, and nonstationary, making them difficult to model and forecast. ANNs have the advantage of being able to capture complex nonlinear relationships and adapt to changing conditions without requiring prior assumptions about the distribution or structure of the data.

The history of ANNs in financial time series forecasting dates back to the late 1980s and early 1990s, when researchers began to explore the potential of ANNs as an alternative to traditional statistical methods, such as the integrated autoregressive moving average model, better known as ARIMA (Autoregressive Integrated Moving Average) and generalized autoregressive models with conditional heteroskedasticity, better known as GARCH (Generalized Autoregressive Conditional Heteroskedasticity). ANNs were shown to have several advantages over these methods, such as the ability to capture non-linear and dynamic relationships, manage noisy and incomplete data, and adapt to changing market conditions (B. Eddy Patuwo & Michael Y. Hu (1998)).

However, ANNs also face some limitations and challenges in financial time series forecasting, such as the difficulty of choosing a suitable network architecture, training algorithm, activation function, and input variables; the risk of overfitting and generalization problems; the lack of interpretability and transparency; and the high computational cost and time (Tealab (2018)).

To overcome these limitations and challenges, researchers have proposed several enhancements and extensions to ANN for financial time series forecasting in recent decades. Some of the major developments include:

- The use of hybrid models that combine ANN with other techniques such as fuzzy logic, genetic algorithms, wavelet analysis, support vector machines, and deep learning to improve ANN performance and robustness (Wong and Guo (2010)).
- The use of recurrent neural networks (hereinafter RNR) or bidirectional, which are a special type of ANN that can process sequential data and capture temporal dependencies. RNRs have been shown to outperform unidirectional neural networks in complex and non-linear time series (Guresen, Kayakutlu, and Daim (2011)).
- The use of more complex RNA models by combining different layers, such as convolutional neural networks (hereinafter, CNN), long short-term memory (hereinafter, LSTM), gated recurrent units (hereinafter GRU) have been applied to financial time series forecasting with promising results (Sezer, Gudelek, and Ozbayoglu (2020)).

The history of ANNs in financial time series forecasting shows that ANNs have evolved and improved over time to cope with the complexity and uncertainty of financial markets. However, some of the previously mentioned challenges and limitations still persist, such as overfitting, generalization, interpretability, robustness, and computational cost.

2.2.2 Convolutional Neural Networks

The RNA model used in this work is composed of several layers, the most important being the Conv1D layer, a specific type of CNN, and the LSTM layer, both mentioned in the previous subsection when the ANN structures that are most used today. This subsection focuses on the Conv1D Layer, so the fundamental concepts to understand its operation are explored, explaining convolution, convolutional neural networks and Conv1D and their use for time series analysis. An overview of convolution and how it can be applied to time series data is provided. Then, CNNs and their architecture, which allows them to automatically learn features from time series data, are discussed. Finally, Conv1D, a specific type of convolutional neural network layer that is particularly effective for processing time series data, is explained.

As discussed in Siddiqui (2023), convolution is a mathematical operation that is commonly used in signal processing and image analysis. It involves taking two functions and producing a third function that represents how one of the original functions modifies the other. In the context of time series data, convolution can be used to extract features from the data by applying a filter to the time series.

In addition to extracting features from time series data, convolution can also be used for other tasks such as noise reduction, anomaly detection, and prediction. For example, CNN can be trained to predict future values of a time series by learning the underlying patterns in the data. In general, convolution is a powerful tool for analysing time series data and its applications are numerous Siddiqui (2023).

CNNs were first introduced at Lecun et al. (1998) and are a type of deep learning model that is commonly used for image analysis. However, as previously mentioned, they can also be used for time series analysis, as they are well-suited for learning features from data that have a spatial or temporal structure.

The architecture of a CNN consists of one or more convolutional layers, which apply filters to the input data to extract features. Each filter is a set of weights that are learned during the training process. By sliding the filter over the input data, the convolutional layer computes a dot product at each position, producing a new Lecun et al. (1998) feature map.

In time series context, CNN can learn to automatically extract features from data at different scales and time intervals, making it a powerful tool for time series analysis. A key advantage of using CNN for time series analysis is that it reduces the need for manual feature engineering. Instead of designing filters by hand, CNN learns to automatically extract features from the data, making it more flexible and adaptable to diverse types of time series data.

In general, the architecture of a CNN allows it to automatically learn features from time series data, making it a powerful tool for time series analysis, with Conv1D being one of the most widely used CNN structures for this task.

As explained in Jing (2020) Conv1D is a specific type of CNN layer that is designed to process one-dimensional data, such as time series data. While traditional CNNs are designed to process two-dimensional data, Conv1D is specifically optimized for one-dimensional data, making it more efficient and effective for time series analysis.

The architecture of a Conv1D layer is similar to that of a traditional CNN, but with some key differences. Instead of using two-dimensional filters, Conv1D uses one-dimensional filters, which are applied to the input time series to extract features. The features that are extracted from the string will depend on the different configurations used for the filter configuration and the number of filters used, being the following formula to calculate the amount of feature that each filter extracts: *Equation 1* (Jing, 2020):

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

Where:

Lout: is the length of the output of the filtering process or the number of features.

Lin: the length of the input vector, corresponding in time series analysis to the number of observations that contain the samples of the time series that are passed to the filter.

kernel_size: is the size of the filter, which defines how many observations of the input vector are passed to the filter each time. *Figure 4* represents how the size of the filter can affect the length of the output vector.

stride: represents the number of steps or observations by which the selection of observations passed to the filter is moved. *Figure 5* represents how the stride parameter can affect the length of the output vector.

dilation: is the distance of the observations that pass the filter. *Figure 6* represents how the dilation parameter can affect the length of the output vector.

padding: represents the number of zeros to add to each end of the vector. *Figure 7* represents how the padding parameter can affect the length of the output vector.

Overall, Conv1D is a powerful tool for processing time series data, and its advantages include computational efficiency and the ability to capture time dependencies in the data. Its use cases are numerous and span different fields, making it a valuable tool for time series analysis.

2.2.3 Long short-term memory

This subsection explains why LSTMs are one of the most widely used ANN structures in time series forecasting, based on a brief explanation of RNRs and why they are useful in solving series forecasting problems. of time, delving into why LSTMs differ from the rest of the RNNs, and the operation of each of the layers that make up the structure of an LSTM layer.

Olah (2015) explains that an RNN can be considered as multiple copies of the same network, *Figure 8*, states that this aspect reveals that RNRs are intimately related to sequences and lists, which makes this type of RNA the one that naturally used for work with time series.

Conventional RNRs present a problem in relation to the ability to retain information, as explained by Olah (2015), standard RNNs perform with great capacity only if the information relevant to the current situation is recent, that is, where the gap between the relevant information and where it is needed is small, *Figure 9*; further exposes that as the gap grows, standard RNNs are unable to access the relevant information, *Figure 10*.

As previously mentioned, LSTMs are a type of RNR that can learn long-term dependencies on sequential data. These were proposed in Hochreiter (1997) and have been widely used for various tasks such as language modelling, speech recognition, machine translation, image description, and time series forecasting.

The main idea of LSTM is to introduce a memory cell that can store and update information over long steps of time. The memory cell is controlled by three gates: an entry gate, a forget gate, and an exit gate. These gates are neural networks that learn to regulate the flow of information in and out of the cell *Figure 11*.

The input gate decides how much of the new input to add to the cell state. The forget gate decides which part of the previous cell state to keep or delete. The output gate decides which part of the current cell state is to be sent to the next layer. Olah (2015) based on what was exposed in Hochreiter (1997), describes the operation of the doors in four steps:

1. Deciding which cell state information is forgotten through the gate, forget gate layer f_t. This gate looks at h_{t-1}, hidden state from the previous time period, and x_t, input

from the current time instant, and outputs a number between 0 (undo) and 1 (hold). for each number in cell state C_{t-1}, *Figure 12*, *Equation 2*.

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

2. Decide what latest information is stored in the cell state. For this first the input gate layer decides which values to update and then a tanh (hyperbolic tangent) layer creates a vector of new candidate values ($\tilde{C}_t$) that could be added to the state, *Figure 13*, *Equation 3* y *Equation 4*.

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

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

3. The state of the old cell, C_{t-1} is updated to the new state of cell C_t. Multiply the previous state by f_t, forgetting what is necessary, then add $i_t * \tilde{C}_t$. These are the new candidate values, scaled by how much each status value needs to be updated, *Figure 14*, *Equation 5.*

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

4. An output is generated based on the cell state. Running first a sigmoid layer that decides what parts of the cell state is the output; then the state of the cell is passed through a tanh function (scaling the values between −1 and 1) and multiplied by the output gate, output gate, *Figure 15*, *Equation 6* y *Equation 7*.

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

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

LSTMs can learn to capture long-term dependencies by tuning gate values through back propagation. For example, if a certain input is relevant to a later exit, the input gate will learn to let it in, and the forgotten gate will learn to hold it in the cell state until it is needed. Conversely, if an input is irrelevant or stale, the gateway will learn to ignore it, and the forgotten gate will learn to remove it from the cell state.

2.3 Portfolio composition

In this section, the problem of finding the best possible portfolio composition is exposed and the theory behind the objectives to solve it is explained in a general way. In addition, the most used techniques will be listed, differentiating between classical approaches and intelligent approaches. Subsequently, quadratic programming will be explained and some techniques within this discipline of mathematical optimization will be mentioned. It will be shown how the portfolio optimization problem can be described as a quadratic programming problem. In addition, a brief explanation of the Dual Active Set Method will be provided, a technique widely used in this discipline, and which will be used in the following chapters.

2.3.1 Problem and techniques

As Gunjan (2023) explains, portfolio optimization is the process of selecting the best mix of assets to hold in a portfolio based on pre-defined objectives. The objectives can be the maximization of return or the minimization of risk, or both. Portfolio optimization involves finding the optimal weights for each asset in the portfolio so that the overall portfolio meets the desired objectives. This can be a challenging problem due to the considerable number of assets to choose from and the complex relationships between them.

Portfolio optimization is an important process for investors as it helps them minimize risk and maximize return on their investments. By carefully selecting the assets to hold in their portfolio, investors can achieve their desired level of risk and return while diversifying their investments to reduce overall risk. Portfolio optimization is a crucial mechanism used to reduce investment risk.

There are various techniques that can be used to solve the portfolio optimization problem. In Gunjan (2023) these techniques are classified into two categories: classical approaches and intelligent approaches. Below is a general explanation of some of the techniques belonging to each approach.

Classical approaches:

- Mean-variance: This technique, proposed in Markowitz and Markowitz (1967), is based on the idea of minimizing the variance for a given expected return or maximizing the expected return for a given variance. It is a parametric quadratic programming (hereinafter PQP) technique that can be used to solve quadratic optimization problems that arise in portfolio optimization (Aijun Zhang & Chun-hung

Li & Agus Sudjianto (2008)). The mean variance approach assumes that investors are risk averse and prefer portfolios with lower variance. The technique consists of constructing a portfolio frontier that represents the set of portfolios that offer the highest expected return for a given level of risk. The optimal portfolio from this frontier is then selected based on the investor’s risk preferences.

- Skewed variance: This technique extends the mean-variance approach by accounting for skewed distribution. It was proposed in Samuelson (1970) and can be used when the distribution function is not quadratic in nature. Skewness measures the skewness of a distribution and can provide additional information about the potential risks and returns of a portfolio. By incorporating asymmetry into the portfolio optimization process, investors can better understand potential downside risks and make more informed decisions.
- Value at Risk (VaR): This statistical approach measures the potential loss in value of a portfolio over a defined period for a given confidence interval. It was introduced in the first edition of Jorion (2007) in 1997 and requires the determination of three parameters: time period, confidence level, and value-at-risk unit. VaR provides a measure of the maximum potential loss that could occur with a given probability in a specified time horizon. It is commonly used by financial institutions to manage their risk exposure and comply with regulatory requirements.
- Conditional Value-at-Risk (CVaR): This approach widens the VaR by considering the expected loss that exceeds the VaR. It was introduced in Rockafellar and Uryasev (2002) and can manage extreme losses by using dynamic weights derived from historical data. CVaR provides a measure of the expected loss that could occur beyond the VaR threshold. It is also known as Expected Shortfall (ES) or Tail Value-at-Risk (TVaR) and is considered a more consistent measure of risk than VaR.
- Mean-Absolute Deviation (MAD): This technique can be used for large-scale and highly diversified portfolio selection problems. It was introduced in Konno and Yamazaki (1991) and penalizes both positive and negative deviations. MAD provides a measure of the average absolute deviation of portfolio returns from their mean value. It is considered more robust than variance-based measures, as it is less sensitive to outliers.

- Minimax: This technique uses the minimum return as a measure of risk. It was introduced in Cai et al. (2004) and has certain advantages when the returns are not normally distributed. Minimax provides a worst-case measure for a portfolio by minimizing the maximum potential loss that could occur. It can be useful for investors who are particularly concerned about downside risks.

Smart approaches:

- Bayesian Networks: These probabilistic graphical models can be used to model risk and return. They were featured on Shenoy and Shenoy (2000) and can be used to visualize the relationship between different variables in a model. Bayesian networks provide a way to represent complex dependencies between variables using directed acyclic graphs (DAGs). They can be used to model uncertain relationships between variables and to make probabilistic predictions about future events. In the context of portfolio management, Bayesian networks can be used to model relationships between different assets and make predictions about their future returns based on historical data and other relevant information.
- Support Vector Regression (SVR): This machine learning technique can be used to determine the amount to buy and sell. It was introduced by Drucker et al. (1996) and has certain advantages over statistical-based techniques, such as its ability to learn from historical data. SVR involves building a hyperplane that separates data points with different labels while maximizing the margin between them. It can be used for regression tasks where the goal is to predict continuous values instead of discrete labels. In the context of portfolio management, SVR can be used to predict future asset prices based on historical data and other relevant information.
- Artificial neural networks: As explained previously, these computational models can be used to solve complex computational and learning problems. In the context of portfolio management, neural networks can be used to predict future asset prices or returns based on historical data and other relevant information, which is what they are used for in this paper.
- Reinforcement Learning: This type of machine learning involves an agent or model interacting with its environment to learn from its actions. It was featured in Sutton and Barto (2018) and works to maximize agent reward. Reinforcement learning

involves learning through trial and error interactions with an environment. The agent takes actions based on its current state and receives rewards or penalties based on the results of those actions. Over time, the agent learns to take actions that maximize his accumulated reward. In the context of portfolio management, reinforcement learning can be used to develop trading strategies that maximize returns while managing risk.

2.3.2 Quadratic programming

This sub-heading explains what quadratic programming is. What are some of the techniques that exist within this discipline of mathematical optimization. It also exposes how the portfolio optimization problem can be described as a quadratic programming problem and briefly explains how one of the most used techniques in this discipline works, specifically the so-called Dual Active Set Method, which is used in the later chapters.

Quadratic programming can be chosen from among the techniques listed in the previous subheading for several reasons. First, it is a well-established technique that has been widely used in portfolio optimization. It can manage complex optimization problems with multiple constraints and can provide an efficient and effective way to solve the portfolio optimization problem. This makes it a useful tool for investors looking to minimize risk while achieving the desired level of return. Finally, quadratic programming has a solid theoretical foundation and has been widely studied in literature. This makes it a reliable and well-understood technique that can be used with confidence in portfolio optimization.

There are several quadratic programming techniques, among the most used are:

- Interior Point: This is a linear or nonlinear programming method that achieves optimization by going through the centre of the solid defined by the problem instead of around its surface. A polynomial time linear programming algorithm using an interior point method was found by Karmarkar (1984).
- Active Set: This is an algorithm used to identify the active constraints in a set of inequality constraints. The active constraints are then expressed as equality constraints, thus transforming an inequality constrained problem into a simpler equality constrained subproblem. The active set method was first introduced in an

article by Beale (1959) and developed by Fletcher (1971) and Bunch and Kaufman (1977).

- Dual Active Set: The method, as exposed by Goldfarb and Idnani (1982) and Goldfarb and Idnani (1983), is an efficient and numerically stable dual algorithm for positive definite quadratic programming that takes advantage of the fact that the unrestricted minimum of the objective function can be used as a point of departure.
- Augmented Lagrangian: It was independently introduced in Magnus R. Hestenes (1969) and Powell (1969). It is used to solve constrained optimization problems by adding a penalty term to the objective function that penalizes any violation of the constraints. The penalty term is typically a multiple of a constraint violation measure, such as the sum of squared constraint violations.
- Conjugate Gradient: This is an iterative method for solving systems of linear equations with a symmetric positive definite matrix. It can also be used to solve unrestricted optimization problems by finding the minimum of a quadratic function. The method generates a sequence of search addresses that are conjugated with respect to the matrix that defines the system of equations or quadratic function. The conjugate gradient method was originally introduced in an article by Magnus R. Hestenes and Stiefel (1952).
- Gradient Projection: The gradient projection method was introduced in J. B. Rosen (1960) and J. Rosen (1961). This is an iterative method for solving constrained optimization problems by projecting the gradient into the feasible region at each iteration. The projected gradient is then used as the search direction, and a line search is performed along this direction to find a new iteration that satisfies the constraints and reduces the objective function.

From the previously mentioned techniques, the Dual Active Set Method algorithm (hereinafter, DASM) was selected, which, as previously mentioned, was introduced in Goldfarb and Idnani (1982) and Goldfarb and Idnani (1983), it is an optimization algorithm to solve quadratic programming problems. The algorithm predicts the active set of constraints that are equally satisfied in the solution of the problem. Computes a sequence of optimal solutions of QP problems involving some of the constraints of the original problem, called a sequence of dual feasible points.

Below is a general example of how the DASM algorithm could work using what-if values for a 2-asset portfolio optimization problem, the example was built from Goswami, Mondal, and Paruya (2012) and Walker (2014):

Under the assumption that it is about finding the best composition of a portfolio in which, for simplicity, we have 2 assets, the quadratic problem would be posed as follows *Equation 8*:

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

Assuming that they have average monthly returns $r = [0.02 \quad 0.03]$ and covariance matrix $C = \begin{bmatrix} 0.001 & 0.0008 \\ 0.0008 & 0.002 \end{bmatrix}$. The vectors and matrices needed for the DASM algorithm can be constructed as follows:

- The average monthly return vector would be $r = [0.02 \quad 0.03]$.
- The covariance matrix C would be used as the matrix D in DASM.
- The constraint $w_1 + w_2 = 1$ can be written in matrix form as $[1 \quad 1]\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = 1$. This would be the first row of the A array in DASM.
- The minimum return requirement $w_1 \mathbb{E} + w_2 \mathbb{E} \geq 0.005$ can be written in matrix form as $[0.02 \quad 0.03]\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \geq 0.005$. This would be another row of the A array in DASM.
- The constraints $0 \leq w_i \leq 1$ can be written in matrix form as $[1 \quad 0]\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \geq 0$ y $[0 \quad 1]\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \geq 0$ for lower bounds and $[-1 \quad 0]\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \geq -1$ and $[0 \quad -1]\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \geq -1$ for upper bounds.
- The matrix A would look like this: $A = \begin{bmatrix} 1 & 1 \\ 0.02 & 0.03 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}$

The corresponding vector b would be $[1 \quad 0.005 \quad 0 \quad 0 \quad -1 \quad -1]$. We can then use the DASM algorithm to solve this quadratic programming problem and determine the optimal asset allocation in our portfolio.

Step 0: Find the unrestricted minimum by solving the unrestricted quadratic programming problem. Set the number of elements in the active set A (empty set) to zero.

Step 1: Choose a violated constraint, if any. In this case, suppose that the constraint $w_1 + w_2 = 1$ is violated.

Step 2: Calculate the primary and dual step directions and the step length $t = min(t_1, t_2)$. Suppose $t = t_2$.

Step 3: Step up and update the active set A and the solution (S) for pair (x, A). Since $t = t_2$, we add the pth constraint (in this case $w_1 + w_2 = 1$) to $\bar{N}$ and update H and N^* in *Equation 9*.

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

Where:

N^* is the pseudo inverse or generalized Moore-Penrose inverse of $\bar{N}$.

$\bar{N}$ is the matrix of the normal vectors of the constraints in the active set A.

H is the reduced inverse Hessian operator of Q.

These steps are repeated iteratively until all constraints are satisfied and the optimal asset allocation has been determined.

2.4 Data

This section is divided into three sub-sections in which the process of obtaining the data necessary to conduct the rest of the procedure is described. The first details the steps conducted to obtain the data from the companies and select those with which they worked in the rest of the procedure. The second sub-heading presents a brief explanation of the computed indicators that will be used as input variables, together with the historical profitability values of the companies selected in the first sub-heading. The third sub-heading exposes the procedure conducted for the creation of the input and output vectors from the data resulting from the second sub-heading.

2.4.1 Data Collection

A more detailed explanation regarding the code used to conduct the procedure described in this sub-heading can be found in Annex. 4 - *Data Collection*.

In order to exemplify how artificial neural networks and quadratic programming can be used in a portfolio management strategy, it was decided in this paper to use data from the Spanish market. Therefore, it was decided to work with the information corresponding to the companies that are in the list of listed companies that is exposed in “Empresas Cotizadas” and can be seen in *Table 2*.

The *Table 2* collects the data of 119 companies. The data collected being the name, ticker, sector and subsector, market, index of each of the companies and whether or not they were selected to conduct the rest of the procedure after conducting the steps set out in this sub-heading.

In order to obtain the data from the companies and analyse them to select those with which we worked in the rest of the procedure, was used as a source. Next, the process conducted to obtain and select the data is explained.

It was decided to download the monthly data of each of the companies collected in *Table 2*. Obtaining all the data between January 31, 2000, to February 28, 2023, of each of the entities.

After obtaining the data, the quality of the data was evaluated. The evaluation began with a visual exploratory analysis of adjusted prices since, as explained in the previous chapter, these are the ideal ones to use in any historical analysis methodology.

During the aforementioned visual exploratory analysis, it was detected that there were irregularities in the adjusted prices of some of the series. The irregularities detected consisted of the incorrect registration of the adjusted prices, as well as errors in their calculation. These errors were easily detected by looking at the graphs of the constant trend adjusted closing price values over long periods of time, as seen in *Graph 6*, which indicates mis recording of price changes; as well as sudden changes of up to more than 100% in them in a single period of time, which may indicate a miscalculation in the adjusted price, as seen in *Graph 7*, in this last case it was verified with other sources like, to verify that the prices were indeed miscalculated.

Given the time available to carry out the study described in the procedure and the extensive amount of time that the investigation would require to be carried out to replace the erroneous values in the series, it was decided to eliminate these irregularities by using only the values after January 2005, which no longer presented inconsistencies in the calculation of the adjusted price, subsequently those series that still contained missing values and that presented irregularities in the registration of variations were eliminated. For the latter, those series in which the variations of unrecorded prices are in more than 10 observations.

Remaining after the adjustments made 103 companies, as seen in the selected column of the *Table 2*, some of these companies have different numbers of observations, because not all of them existed or had gone on the market stock market before January 2005.

Once the companies with which we worked were selected, their returns were computed from the adjusted prices. In addition to the returns corresponding to the selected companies, the returns of the adjusted closing price of the IBEX 35 were used, as well as other variables that serve as indicators of the behaviour of the returns, and their relationship with those of the index, in this case those of the IBEX 35. These variables include the volatilities of the companies and the index, the correlation between the values of the series and the IBEX, and the beta of the companies in relation to the IBEX.

2.4.2 Indicators

This sub-heading presents a brief explanation of the computed variables to be used as input variables in conjunction with the historical values of the companies' returns. A more detailed explanation regarding the code used to conduct the procedure described in this sub-heading can be found in Annex. 4 - *Indicators*.

2.4.2.1 Volatility

Building on Hargrave (2023) and Hayes (2023), standard deviation and volatility are two related concepts that measure how much the price of a stock or other asset fluctuates over time. Standard deviation is a statistical term that quantifies the spread of a set of data points around its mean value. Volatility is a financial term that describes the degree of variation in the returns of an asset over a given period of time.

Standard deviation and volatility are important in stock market analysis because they indicate the risk and uncertainty associated with investing in a particular asset. A high standard deviation or volatility means that the price of the asset can change significantly in either direction, implying greater potential for profit or loss. A low standard deviation or volatility means that the asset's price is relatively stable and predictable, which means less potential for profit or loss Hayes (2023).

To calculate the volatility of a stock or index, the standard deviation of returns is calculated. Therefore, the necessary calculations are those shown below in the *Equation 10*:

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

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

Where:

- R_i is the return of the stock in the period i
- P_i and P_{i-1} are the prices of a share in time periods i and $i-1$, respectively.
- σ is the standard deviation - N is the number of observations.
- $\bar{R}$ is the average return on the stock.

Standard deviation and volatility are useful tools for investors and analysts to assess the risk-reward balance of different assets and portfolios. They can also help compare the performance of different assets and portfolios over time and under different market conditions.

2.4.2.2 Correlation

As Edwards (2022) explains, correlation is a statistical measure that determines how two variables move relative to each other. In stock market analysis, correlation can help understand the behaviour of different stocks or market indicators over time. Taking the data used in this paper as an example, if the prices of one of the selected companies tend to go up and down together with the IBEX 35, these prices have a positive correlation. On the other hand, the company's prices tend to rise when the IBEX 35 indicator falls, they have a negative correlation. A correlation coefficient of zero means that there is no linear relationship between the variables, being in this case the values of the IBEX 35 and the prices of one of the determined companies.

As exposed by Ross (2022) the correlation between two variables is calculated using the following equation, *Equation 11*:

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

where:

- ρ_{xy} is the correlation coefficient.
- n is the Number of observations.
- x_i and y_i are the values of the two variables for the i observation.
- $\bar{x}$ and $\bar{y}$ are the means of the two variables.

As also explained by Edwards (2022), the correlation coefficient ranges from -1 to 1, where -1 indicates perfect negative correlation, 1 indicates perfect positive correlation, and 0 indicates no correlation at all. Being able to understand that the closer the correlation coefficient is to both -1 and 1, the stronger the linear relationship between the variables analysed.

As previously explained, the correlation coefficient, in this paper, can be used to analyse how similarly the returns of a company move compared to those of the IBEX 35. The correlation can also be used to diversify a portfolio by choosing stocks that have a low or negative correlation with each other, as explained by Boyte-White (2022). This can help reduce overall portfolio risk, as losses from one stock can be offset by gains from another. However, the correlation is not constant and can change over time due to several factors, such as market conditions, economic events, or company news. Therefore, it is important to monitor the stock correlation regularly and adjust the portfolio accordingly Boyte-White (2022).

Correlation is a valuable tool in stock market analysis, but it does not imply causation. Having a high or low correlation between two variables does not imply that one variable causes change in the other. Correlation simply measures the strength and direction of the linear relationship between two variables, without considering other factors that may influence them.

As also exposed in Edwards (2022), the correlation is closely related to the volatility of the market and of the shares, being able to see that, during periods of greater volatility, such as the financial crisis of 2008, the shares can tend to be more correlated, even if they are in different sectors. International markets can also become highly correlated during times of instability. Investors may want to include assets in their portfolios that have low market correlation to equity markets to help manage their risk.

2.4.2.3 Beta

As explained by Kenton (2022) Beta is a measure of how sensitive a stock's returns are to changes in market returns. It is calculated as the slope of the regression line that fits historical stock and market returns. A beta of 1 means the stock is moving coordinated with the market, a beta greater than 1 means the stock is more volatile than the market, and a beta less than 1 means the stock is less volatile than the market.

Beta is important in stock market analysis because, as Kenton (2022) explains, it helps investors assess the risk and return of a portfolio. By knowing the beta of each stock in a portfolio, investors can estimate how much the portfolio will fluctuate with market movements and adjust their asset allocation accordingly. For example, if an investor wants to reduce the risk in their portfolio, they might choose stocks with low or negative beta values that tend to move in the opposite direction of the market.

As explained by Monaghan (2019) Beta is related to mapping, but they are not the same. As explained above, correlation is a measure of how linearly related two variables are, Beta, on the other hand, is a measure of how strongly related two variables are, indicating how much one variable change when another variable changes by one unit. Beta can be calculated from the correlation using the following equation, *Equation 12*:

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

where:

- ρ_{xy} is the correlation coefficient between x and y.

- σ_x is the volatility of x.
- σ_y is the volatility of y.

2.4.3 Vectors

In this sub-section, the procedure conducted to create the input and output vectors from the data resulting from the procedure described in the previous sub-section is explained. A more detailed explanation regarding the code used to conduct the procedure described in this sub-heading can be found in Annex. 4 - *Vectors*.

The structure of the set of input and output vectors is of vital importance in the modelling of ML techniques, having a significant impact on its effectiveness. The set of vectors must be created in a representative way for the problem to be solved, so the steps described below explain in detail the aspects of the problem to be answered in this work and how to shape the set of input and output vectors to it.

As previously mentioned, the objective of this paper is to present a procedure for the use of RNA models and quadratic programming in an investment strategy. The modelling addresses the need to obtain the most accurate predictions possible so that later, based on the predictions and historical data, find the ideal portfolio composition. Therefore, the problem to be represented with the sets of input and output vectors is how to explain the behaviour of the profitability of a company at an instant of time $i + 1$ with the values of several variables at the instant of time i.

To represent this problem, three-dimensional vectors were created, following what was exposed in Chollet and Allaire (2018). The dimensions of these vectors are explained as follows:

- The first dimension is comprised of the number of samples obtained by sectioning the observations of the different series into consecutive two-dimensional vectors.
- The second dimension is comprised of the number of observations of the different series, collected in each two-dimensional vector.
- The third dimension is the amount of series in each two-dimensional vector.

Therefore, in order to correctly obtain these samples, it is necessary to first define which series will be used for the input and output vectors. The series used in the input vectors were defined in the previous section, being these: the historical returns of the company and the IBEX, the historical volatilities of the company and the IBEX, the historical correlation of the company and the IBEX, and the Historical beta of the company and the IBEX. The series used for the output vectors is the historical profitability of the company.

Subsequently, the time horizon to be foreseen was defined, this is a key aspect in the creation of the sets of inputs and outputs. The number of observations defined as the time horizon determines the observations of the output vectors, in the present work an observation was determined as the time horizon since it is desired to predict the profitability of the next month of the different selected companies.

And the last aspect to define is how many observations the model must observe to infer the desired output. This defines the number of observations that will be taken from each time series to form the input vectors. To determine this aspect, an iterative process must be conducted, testing different quantities, and evaluating the results obtained by the models that are trained with them. To simplify the process, in the present work it was determined to evaluate different input sizes, these being 1, 2 and 3 observations. Thus, testing in a certain way how the size of the inputs affects the prediction obtained.

If we have an array for the input vectors contains some 216 observations, we can calculate the number of samples obtained from this array following the following equation, *Equation 13*:

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

where:

- m the Number of samples
- n the Number of observations in the series
- i and o the Number of observations in the input and output vectors, respectively.

The *Table 3* shows the number of samples obtained for the varied sizes of input vectors proposed, for which the different numbers of observations of the selected 103 were considered. The *Figure 16* shows what the input and output vectors look like, in the case in which the input vector has 3 observations.

2.5 Modelling and training

This section is divided into two sub-sections in which the models that were built and the procedure used to train them are briefly described. The first subheading explains the structures of the models used, while the second subheading explains the particularities of the training methodology used.

2.5.1 Modelling

As previously explained, the main elements of the artificial neural network models used are a CNN layer and an LSTM layer. In addition to this, an input layer and an output layer were used, which are in charge of supplying the models with the information of the vectors previously constituted. A more detailed explanation regarding the code used to conduct the procedure described in this sub-heading can be found in Annex. 4- *Modelling*.

Since three different sizes of observations were defined to consider making a prediction, it was necessary to build three different model structures that would adapt to the dimensions of the different input vectors, the different structures can be observed in the *Figure 17*.

The first notable difference between the structures is the outputs of the input layers. This difference is due to the sample sizes if you have chosen to use 1, 2 or 3 observations to build the model. As can be seen, the size of the input layer output therefore modifies the size of the inputs and outputs of the CNN layer.

As previously mentioned, the variations in the second dimension in the outputs of the CNN layer can be explained by the varied sizes of the input vectors. But as can be seen, the size of the third dimension of the output of this layer is the same in all the structures, 64, which indicates the number of filters chosen to use, one of the main parameters to consider when configuring these layers. The latter means that the observations corresponding to the 6 variables used were divided into 64 variables that allow the model a better understanding of the relationship between the variables.

Another aspect that was modified in the CNN layer of the structures was the activation function that by default is called ReLU (for its acronym in English, Rectified Linear Unit) was changed to Leaky ReLU because as explained in OmG (2021) , ReLU is a nonlinear activation function that generates zero for negative inputs, which can cause some neurons to stop learning if many of their inputs are negative, since their gradients will be zero.

Given what was previously explained and that some of the variables used in the input values have a high number of negative observations, as is the case of returns or the correlation of some of the series in certain periods of time, the use of the ReLU activation function did not seem like a good option. Therefore, it was decided to use Leaky Relu as activation function, which as explained in OmG (2021), this is a variant that allows a small constant gradient, non-zero, for negative inputs. This means that this activation function allows some neurons to continue learning from negative inputs.

In the *Figure 18* the domain of the ReLU and Leaky ReLU function is observed, which will allow you a better understanding of what was previously exposed.

The CNN layer in all the structures is linked to an LSTM layer, which in all cases had 64 neurons. The output of this layer was linked to the output layer which returns a single value.

To conclude with the construction of the models, it was determined to use the mean square error (hereinafter MSE) as the function used to evaluate a candidate solution, the results of the model and the SGD optimizer (for its acronym in English, Stochastic Gradient Descent) with an Alpha of 0.0005..

2.5.2 Training

A more detailed explanation of the code used during the procedure described in this subsection can be found in Annex. 4 - *Training*.

The training of Machine Learning algorithms in the forecast of time series has its peculiarities to how models are trained with the aim of solving other types of problems. Therefore, in this sub-section the training methodology used is briefly covered, which is the so-called walk forward validation or advance validation.

As already mentioned, feedforward validation is a method used to evaluate machine learning models on time series data. This is because as explained by Brownlee (2019) it provides the most realistic evaluation of machine learning models on time series data. Traditional model evaluation methods from machine learning, such as k-fold cross-validation or splitting into training and validation data, do not work for time series data because they ignore the time components inherent in the problem. Walk-forward validation takes these temporal components into account and provides a more realistic assessment of how the model will perform when used operationally.

When evaluating a model, we are interested in how the model performs on data that was not used to train it. In machine learning, this is called unseen or out-of-sample data. Commonly, for the resolution of other problems, the data is divided into different subsets: training, testing and validation, whose objective is to train and validate the model. With the walk forward validation methodology, the data is divided by time periods and the model is trained and validated consecutively, which allows evaluating how the model understands the temporal dependence of the data.

By dividing the data by time periods, it allows us to evaluate the actual functioning of the model if it had been applied from the first period, as well as to analyse its behaviour throughout all the periods, observing whether its performance improves or not.

From what is stated in this sub-section, it is understood that the models were trained using the corresponding sample sets, passing all the available samples in a certain period of time before continuing with the next period. Obtaining as a result of the above a prediction corresponding to each period of time contemplated, with the exception of the first two that would be used to train the model for the first time, as seen in the following diagram of the *Figure 19*.

2.6 Result

This epigraph is divided into two sub-headings in which the results obtained are briefly described. In the first of the sub-headings the predictions are described and analysed. The second sub-heading presents the results of applying quadratic programming for portfolio composition based on the predictions obtained.

2.6.1 Predictions

A more detailed explanation of the code used during the procedure described in this subsection can be found in Annex. 4 - *Predictions*.

As explained previously, while the model was being trained, predictions were obtained. As was done with the artificial neural network models, the predictions were computed for the different observations using the arithmetic mean of the observations. The arithmetic mean was used because it is one of the most frequently used measures as an indicator of probable future behaviour in the study of financial time series.

The predictions will be evaluated by computing the MSE and the real values and the R^2 of the results obtained by the artificial neural network models and the arithmetic means.

As explained by Glen (2023) the MSE tells you how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line (these distances are the “errors”) and squaring them. Squaring is necessary to remove any negative signs. It also gives more weight to larger differences. It is called the root mean square error since you are finding the average of a set of errors. The lower the MSE, the better the forecast, shown by *Equation 14* as calculated.

$$MSE = \frac{1}{n} * \sum_{i=1}^{n} \left(Y_i - \widehat{Y_\imath}\right)^2 \qquad (14)$$

Where:

- n: Number of observations.
- Y_i: real value.
- $\widehat{Y_\imath}$: expected value.

As explained by Nandakumar (2020) R^2 is commonly used to explain how well a model does compared to the total mean of the observations, *Equation 15*:

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

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(Y_i - \widehat{Y_\imath}\right)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2} \qquad (15)$$

Where:

- $\bar{Y}$: arithmetic mean of all observations

But this can be an unfair indicator of the performance of a regression model since it is assumed that all observations over which a mean is computed are known, and as mentioned above this is not the case for neural network models. artificial animals trained using the walk forward validation methodology. Due to this, the calculation of R^2 was modified, as has been done in other investigations such as Gu, Kelly, and Xiu (2018), so that the model with which the results obtained by the ANNs used are compared is that comprised by the arithmetic means of the observations earlier than the one predicted.

Next, the different results obtained by the different models built will be briefly described. It should be noted that, although 3 different models of each of them were proposed, 10 were built, with the aim of standardizing the results obtained, since the process of building and training neural networks contains a random factor. Therefore, the results described below are the average results obtained by the various models built.

2.6.1.1 One observation

The results obtained by those models that were trained with input vectors that had one observation from each series showed, as seen in *Graph 8*, that in the first periods the models presented better predictions than those obtained by the arithmetic mean. . It can be observed that the effectiveness of the models in comparison with the means decays as the model progresses in time and learns from the new observations. It is also clearly seen that in most periods the R^2 of this model is negative. In addition, a peak is observed in the MSE of the model at the beginning of 2020, which is understood as a loss of effectiveness of the model. This loss of effectiveness of the model could be related to sudden market movements resulting from the economic effects of Covid-19.

The previous analysis of the behaviour of the indicators of these models by period gives us an overview of how these models performed but given that the results obtained in the companies are essential for the composition of the portfolio, we will now analyse the behaviour observed in the results obtained by the 20 companies that presented the best and worst results, based on the R^2 obtained as a criterion.

Observing the results of the indicators exposed in the *Table 4* , those companies that presented a worse R^2 also present a low , which indicates that it is less likely that the composition of the portfolio will be altered by the results obtained by these companies.

companies. On the other hand, among the companies that obtained a better R^2 there are some that obtained a high MSE accompanied by a R^2 greater than 5%. This indicates that differences could be generated between the compositions of the portfolios due to the differences in the predictions and that these are companies that do not have a good MSE.

The results described in the previous paragraph are similar for the cases of the models built with two and three observations, respectively.

2.6.1.2 Two observations

The results obtained by those models that were trained with input vectors that had two observations from each series, it was found, as seen in *Graph 9*, that in the first periods the models presented better predictions than those obtained by the mean. arithmetic. It can be seen that the effectiveness of the models compared to the means decays as the model progresses in time, but they decay at a slower rate than those models trained with input vectors with one observation. It is also clearly seen that the R^2 of these models has less variation than the R^2 of the previously analysed models, seeing that for these models the R^2 is positive in most of the periods. In addition, as in the case of the models analysed previously, a peak is also observed in the MSE of the model at the beginning of 2020.

2.6.1.3 Three observations

The results obtained by those models that were trained with input vectors that had three observations of each series, it was found, as seen in *Graph 10*, that in the first periods the models presented better predictions than those obtained by the mean. arithmetic. It can be seen that the effectiveness of the models compared to the means decays as the model progresses in time, but they decay at a slower rate than those models trained with input vectors with one observation. It is clearly seen that the R2 of these models has a greater variation than the R2 of the previously analysed models, observing how this variation decreases for those predictions after 2015. These models, like the first ones, presented a negative R2 in most of them. of the periods. In addition, as in the previous cases, a peak is also observed in the MSE of the model at the beginning of 2020.

2.6.2 Portfolios composition

A more detailed explanation of the code used during the procedure described in this subsection can be found in Annex. 4 – *Portfolios composition*.

This sub-heading describes the results obtained after applying quadratic programming to determine the composition of the portfolio. This, as well as the predictions, was made period by period with the aim of emulating a real situation in which the techniques were applied as a whole. Therefore, the present analysis focuses on the behaviour observed when using the various models and the comparison of these results with those obtained with the use of means.

As can be seen in *Graph 11*, the portfolios made from the predictions obtained by the neural network models that had one observation generally obtained better results than the portfolios made from the predictions using the mean. It is observed that both portfolio groups presented a lower return than the index, IBEX, in the period between 2009 and 2016.

When carrying out the analysis of the behaviour of the returns obtained by the models with two input observations, *Graph 12*, it is observed: the behaviour of the returns obtained by the different models varies less than those previously analysed; In this case and contrary to the previous case, the returns remain similar in the period between 2009 and 2016; and although the final result is far from the result obtained by the averages, it is lower than that obtained by the previous models, the latter being due to the fact that the evaluation of the models in this case begins in a period prior to those of the previously analysed models.

Observing the results obtained by the latest models, *Graph 13*, it is observed: a distribution of returns higher than those trained with two observations but lower than those trained with one observation; it is observed that the returns begin to exceed those of the index after 2013 instead of 2016 as in previous years; and it is also observed that the returns of the RNA models are higher than those of the averages and also constitute the maximum returns obtained between the different structures of RNA models.

3. Conclusions

During the development of this work, the application of artificial neural networks and quadratic programming in the management of financial portfolios has been addressed. Through a careful characterization of financial time series, it was possible to understand the importance of analysing their characteristics and patterns to make more accurate forecasts.

Different configurations of artificial neural network models, made up of the combination of convolutional layers and LSTM, were evaluated, which differed in the number of historical observations they would use as inputs before making a prediction. The predictions obtained from the aforementioned models were compared with predictions obtained by using the arithmetic mean, which is one of the most commonly used indicators. As a result of the aforementioned comparison, it was obtained that the models depending on the number of observations that they used as inputs: 1, 2 or 3; they obtained an R^2 of: -0.00287, 0.0611 and 0.0179, respectively.

The predictions obtained, both with the RNA models and with the arithmetic means together with the historical behaviours were used to, through the use of quadratic programming, search for the composition of lower risk portfolios. After carrying out a portfolio management simulation, it was obtained that the portfolios made up from the predictions of the ANN models obtained at the end of the period studied, compared to those made up of the predictions using the arithmetic mean, returns: 5.63% higher , for models that used 1 observation as input; 35.67% higher for those who used 2; and 25.51% for those who used 3. In addition, it was observed that the portfolios made up of the RNA models obtained returns higher than the index, IBEX, by 40.86%, 39.78% and 60.54%, for the models that used 1, 2 and 3 observations as inputs, respectively.

The aforementioned results show that the combined use of these tools, ANN, and quadratic programming, can offer companies and organizations a significant competitive advantage in the management of their financial assets, allowing more effective decision-making, optimizing the composition of portfolios, and maximizing returns.

However, it is important to highlight that the results presented in this paper need a more in-depth study to analyse, among other aspects, the weight that the results of the predictions of the different companies have in the composition of portfolios. For this reason, this work is considered the beginning of a more exhaustive investigation in which: higher quality data must be obtained, and the use of various techniques will be contrasted, both to obtain predictions and to find the composition of the portfolio adequate.

Bibliography

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.

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/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*. Retrieved from: How To Backtest Machine Learning Models for Time Series Forecasting - MachineLearningMastery.com

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&idlng=1*.

CNMV. «Glosario Financiero: Bolsa de valores». Accedido 24 de abril de 2023. *https://cnmv.es/Portal/Inversor/Glosario.aspx?id=0&letra=B&idlng=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&idlng=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*.

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*.

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*, edited by 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 is 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*.

Nandakumar, S. (2020). «How can r-squared be negative when the correlation between prediction and truth is positive?» *Stack | overflow*. Retrieved from: *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. Retrieved from: *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*; 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. Retrieved from: *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*. 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.

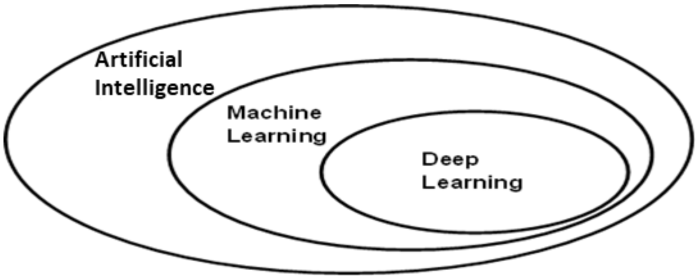
Walker, Ryan. 2014. «Solving Quadratic Program's with R's quadprog package». rwalk. 2014. *https://rwalk.xyz/solving-quadratic-progams-with-rs-quadprog-package/*.

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*.

Annex

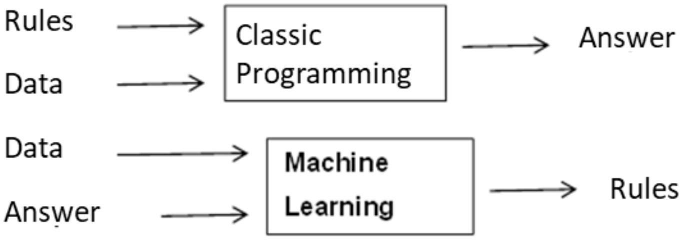
Annexo.1 Figures

Figure 1: Relation between IA-ML-DL



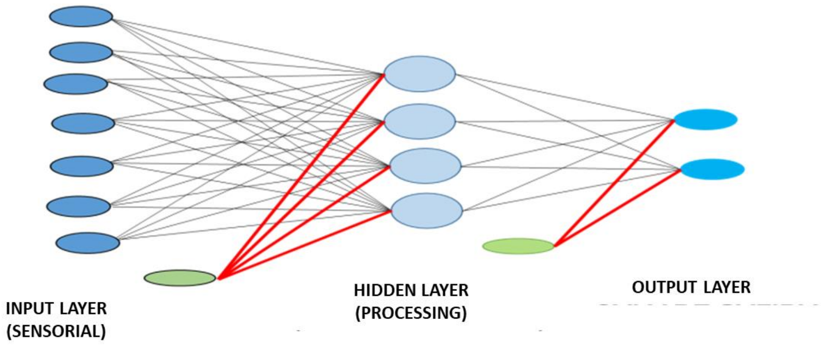
Taken from: Deep learning with R by Chollet y Allaire (2018).

Figure 2: Classic programming and machine learning



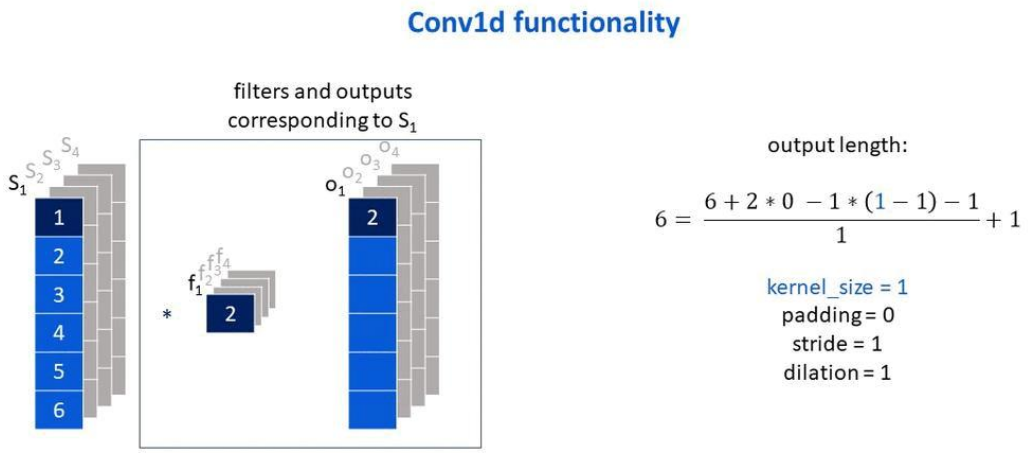
Taken from: Deep learning with R by Chollet y Allaire (2018).

Figure 3: Basic structure of an artificial neural network



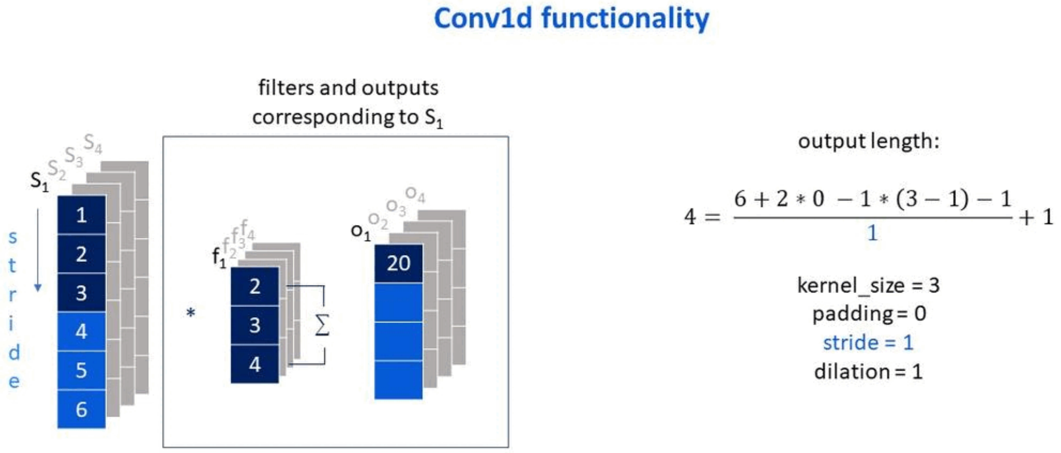
Taken from: Topic 14: neural networks by Larrañaga (2007).

Figure 4: How the size of the filter affects the output vector



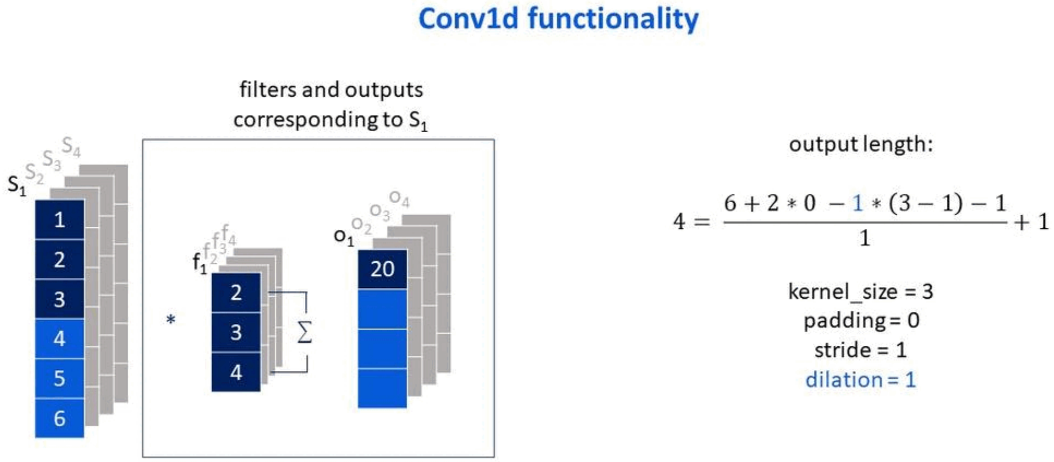
Own elaboration: Elaborated from Jing (2020). Shows how the size of the output vector changes according to the size of the filter that is used.

Figure 5: How stride affects the output vector



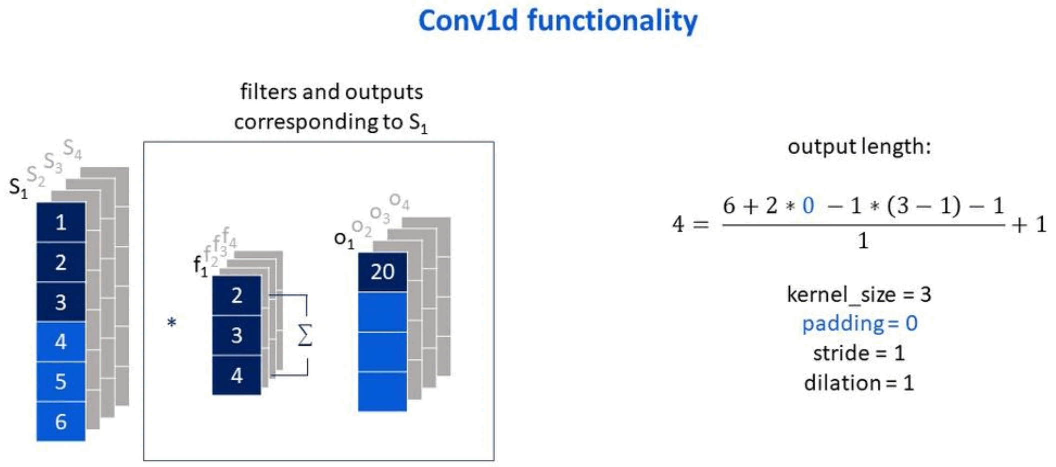
Own elaboration: Elaborated from Jing (2020). Shows how the stride parameter affects the size of the output vector.

Figure 6: How dilation affects the output vector



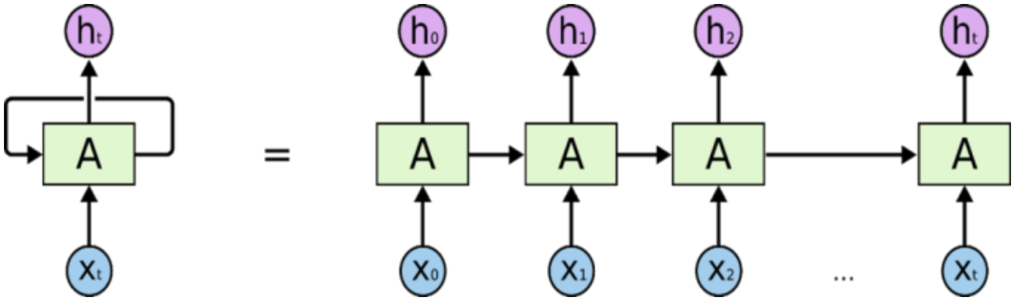
Own elaboration: Elaborated from Jing (2020). Show how the dilation parameter affects the size of the output vector.

Figure 7: How padding affects the output vector



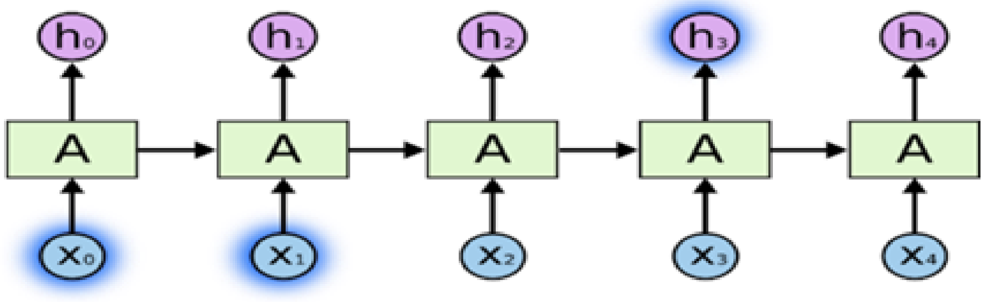
Own elaboration: Elaborated from Jing (2020). Show how the padding parameter affects the size of the output vector.

Figure 8: Deployment of the loop of a standard recurrent neural network



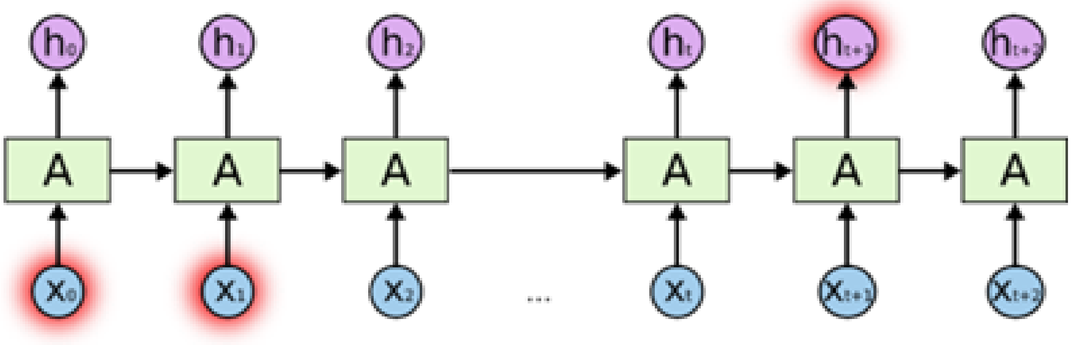
Taken from: Understanding LSTM networks, Olah (2015).

Figure 9: Nearby relevant information



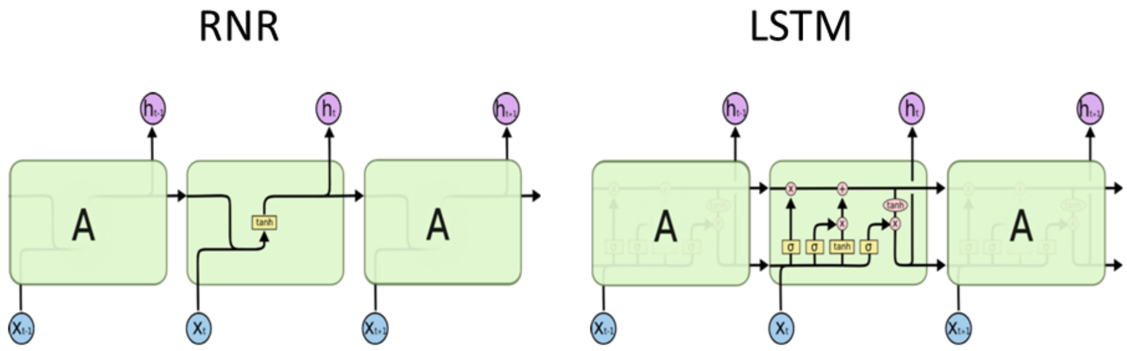
Taken from: Understanding LSTM networks, Olah (2015).

Figure 10: Distant relevant information



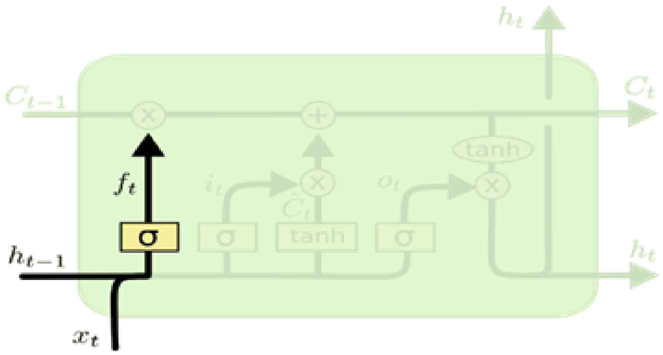
Taken from: Understanding LSTM networks, Olah (2015).

Figure 11: Difference Between Repeat Modules



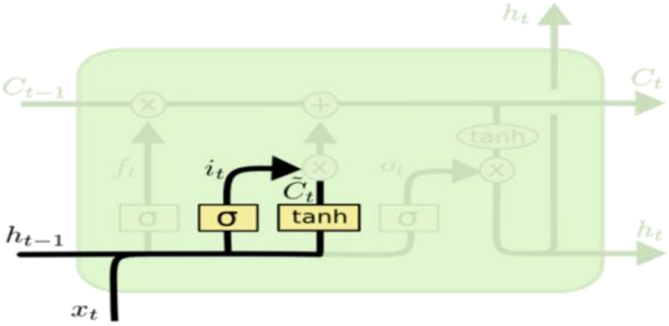
Taken from: Understanding LSTM networks, Olah (2015).

Figure 12: LSTM functionality: Representation of step 1



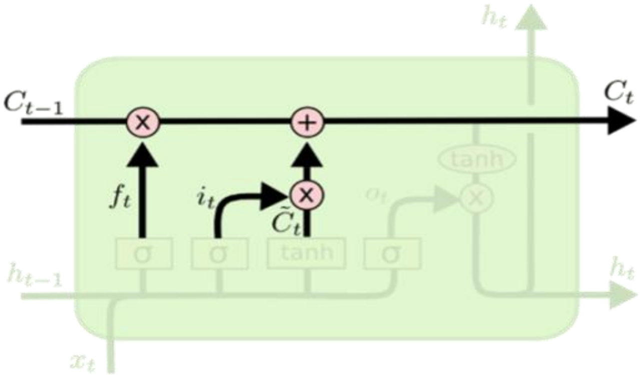
Taken from: Understanding LSTM networks, Olah (2015).

Figure 13: LSTM functionality: Representation of step 2



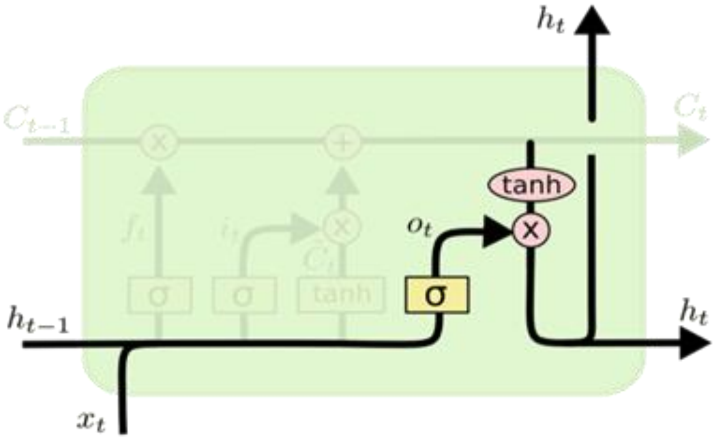
Taken from: Understanding LSTM networks, Olah (2015).

Figure 14: LSTM functionality: Representation of step 3



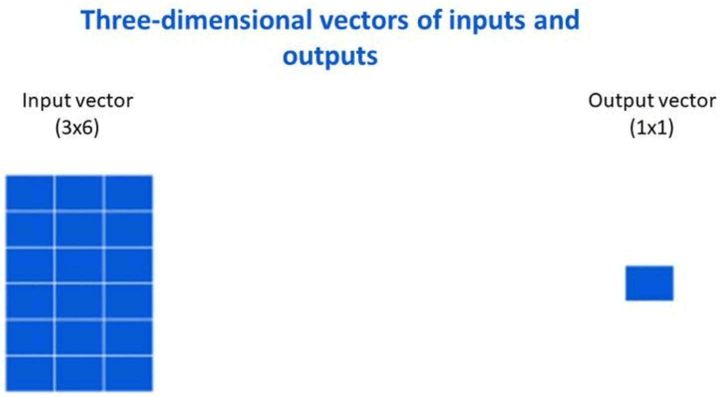
Taken from: Understanding LSTM networks, Olah (2015).

Figure 15: LSTM functionality: Representation of step 4



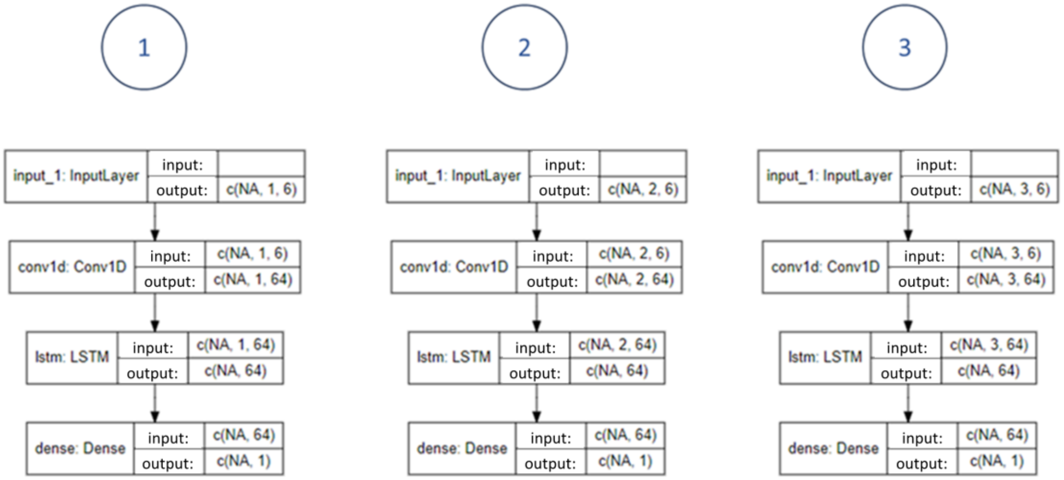
Taken from: Understanding LSTM networks, Olah (2015).

Figure 16: Input and output vector display



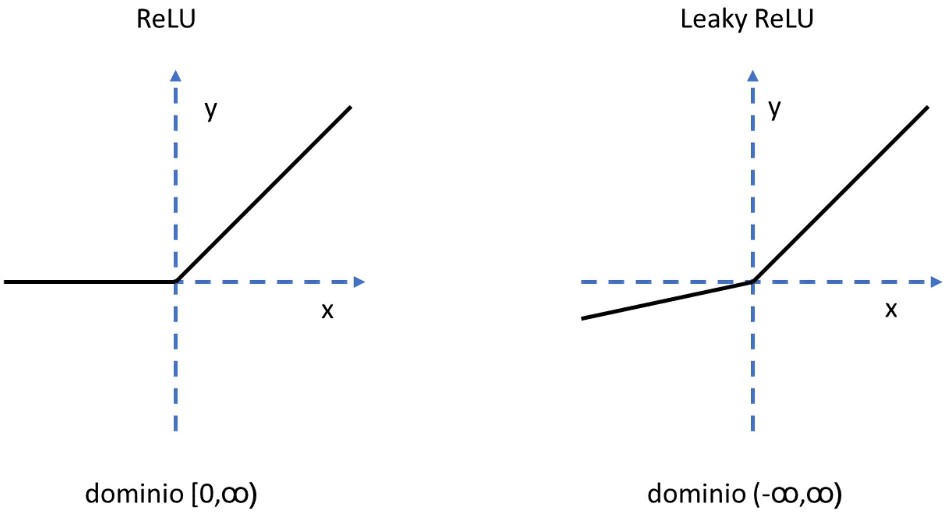
Own elaboration: Made from an image in Chollet and Allaire (2018). It shows what the three-dimensional input and output vectors for a company's data look like, if three observations are used to create the input vector.

Figure 17: Different structures depending on the different sizes of input vectors



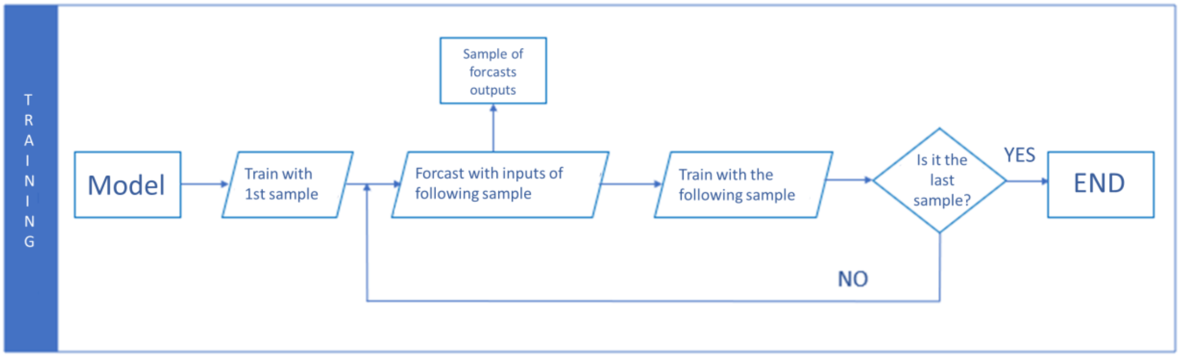
Own elaboration: Elaborated from the different models built using the keras and tensorflow packages in R and were graphed using the Diagrammer package by Iannone (2023).

Figure 18: ReLU and Leaky ReLU domain



Own elaboration: Elaborated from the images that can be seen in Rallabandi (2023).

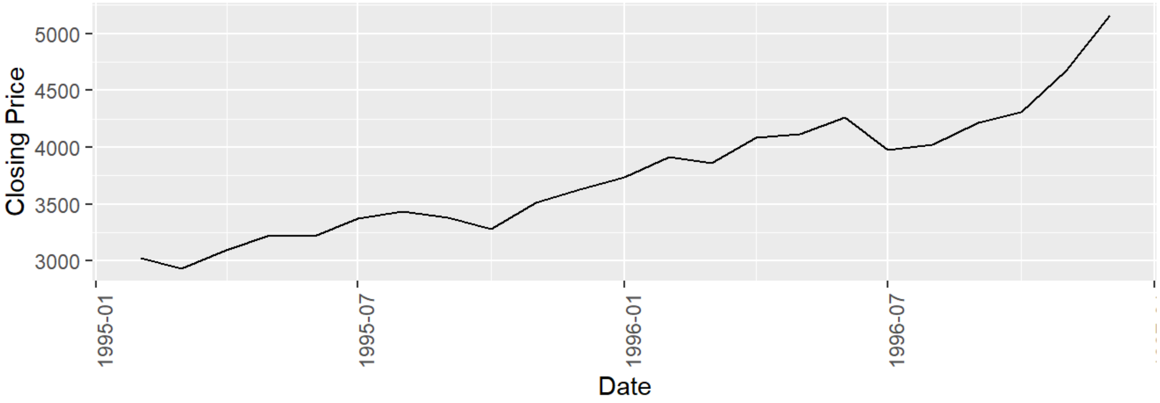
Figure 19: Flowchart of the Walk Forward Validation methodology


TRAINING
Model
Train with 1st sample
Forcast with inputs of following sample
Sample of forcasts outputs
Train with the following sample
Is it the last sample?
YES
END
NO

Own elaboration

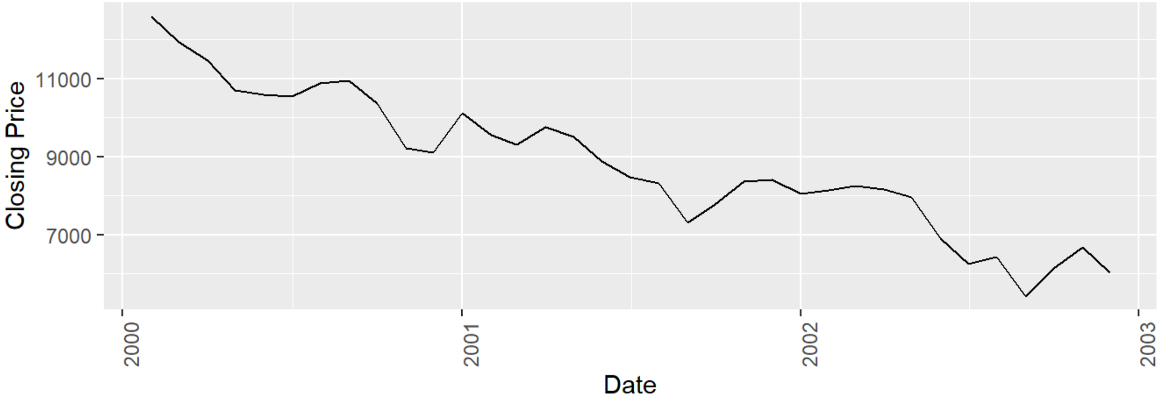
Annex.2 Graphs

Graph 1: Bullish and heteroscedastic trend



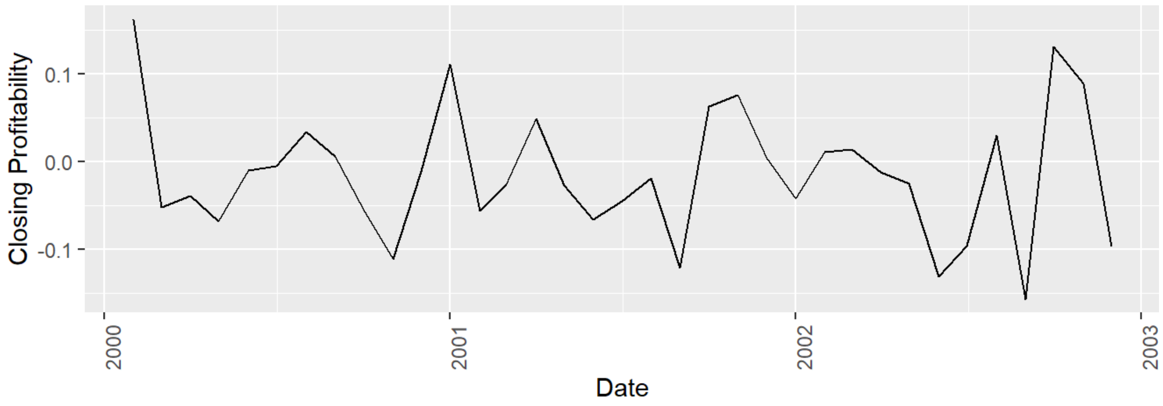
Own elaboration: By using RStudio with the IBEX historical database, obtained from https://finance.yahoo.com/, in the period between 01-1995 to 01-1997.

Graph 2: Bearish and heteroscedastic trend



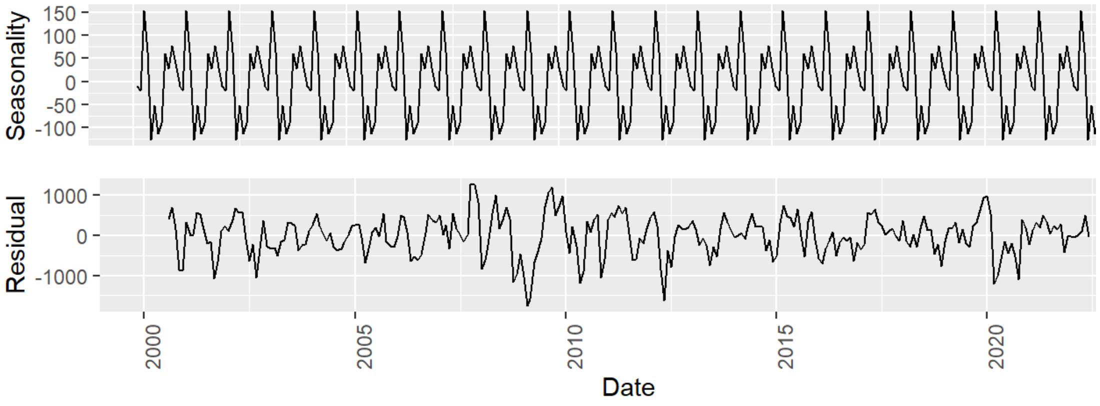
Own elaboration: Through the use of RStudio with the IBEX historical database, obtained from https://finance.yahoo.com/, in the period between 01-2000 to 01-2003.

Graph 3: No trend, homoscedastic and stationary



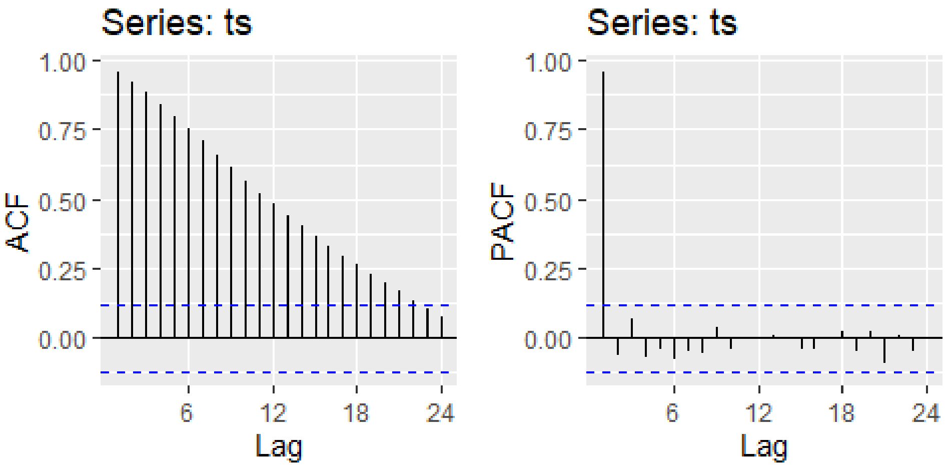
Own elaboration: By using RStudio with the IBEX historical database obtained from https://finance.yahoo.com/, in the period from 01-2000 to 01-2003, using the returns calculated from the closing price.

Graph 4: Decomposition: seasonality and error



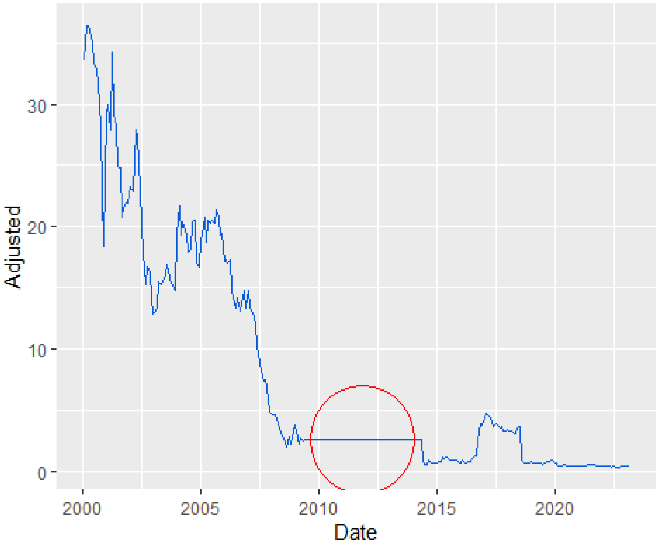
Own elaboration: By using RStudio with the IBEX historical database obtained from https://finance.yahoo.com/, decomposing the time series made up of the observations that cover the period from 01-2000 to 01-2023.

Graph 5: Correlogram



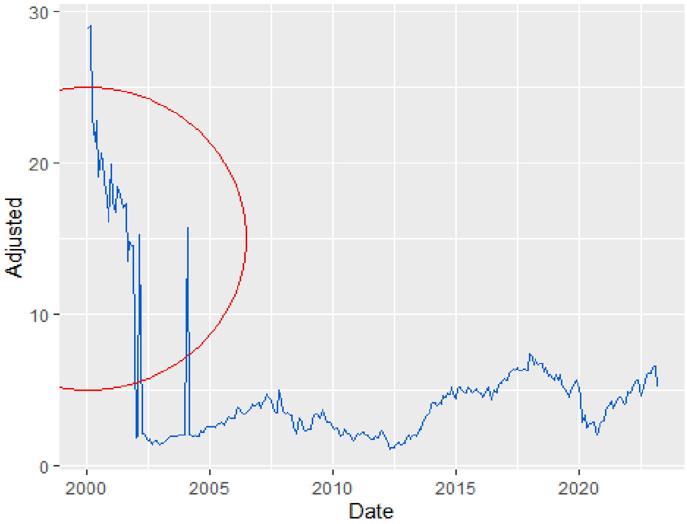
Own elaboration: By using RStudio.

Graph 6: Constant trend in the adjusted closing prices of the company “Nueva Expresión Textil S.A”



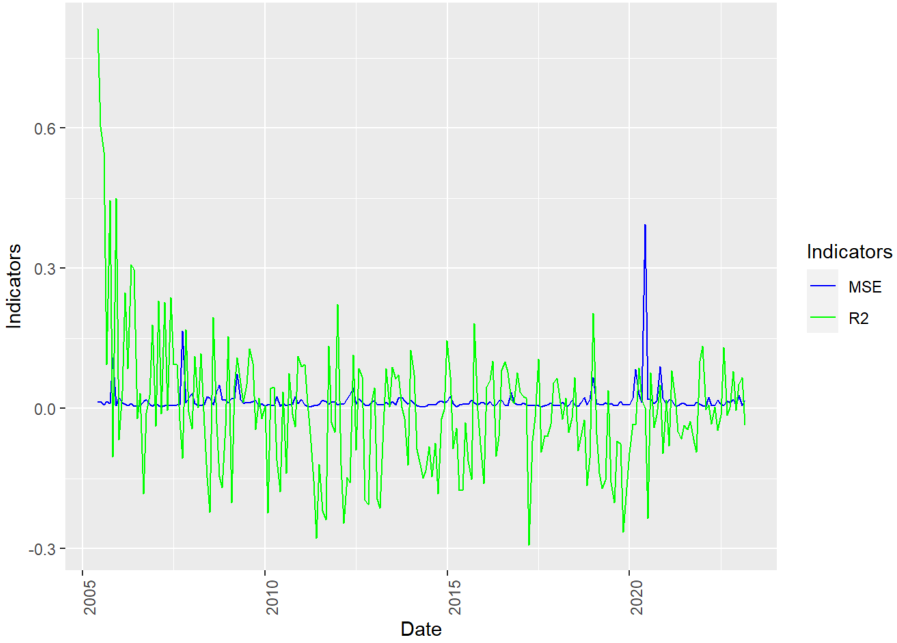
Own elaboration: Based on the data obtained from (n.d.) corresponding to the company “Nueva Expresión Textil S.A” in the period from January 31, 2000, to February 28, 2023.

Graph 7: Sudden price changes that reflect an erroneous calculation of adjusted closing prices, "BANKINTER,S.A."



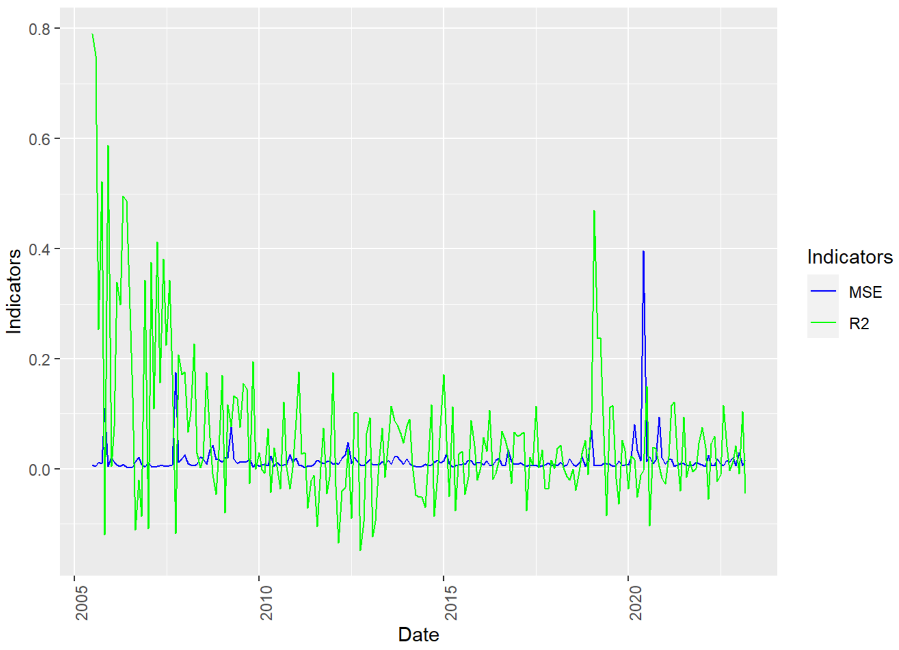
Own elaboration: Based on the data obtained from (n.d.) corresponding to the company "Nueva Expresión Textil S.A" in the period from January 31, 2000, to February 28, 2023.

Graph 8: Evolution of indicators – Entries with one observation



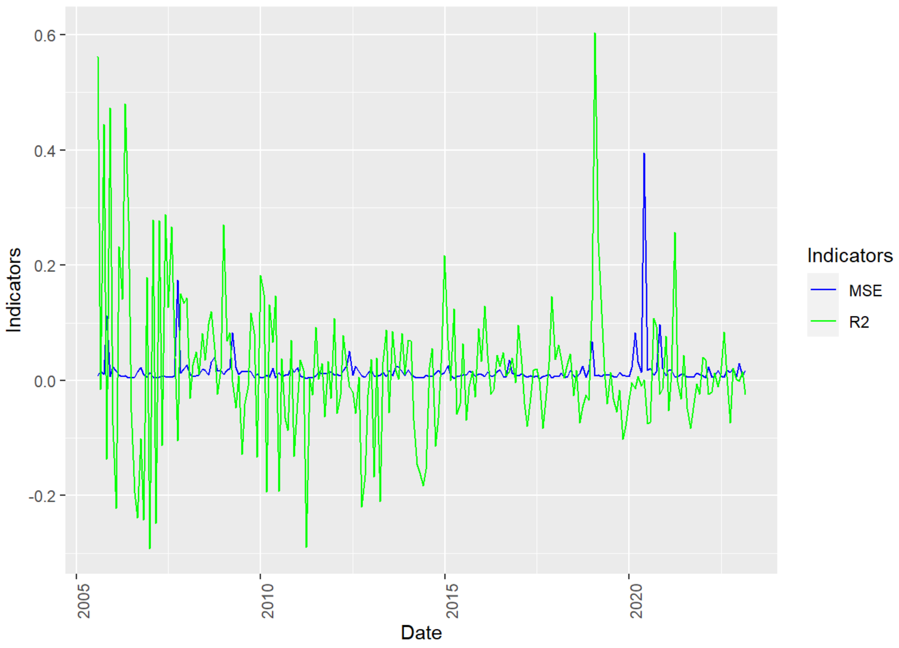
Own elaboration: Through the use of R and Rstudio.

Graph 9: Evolution of the indicators – Entries with two observations



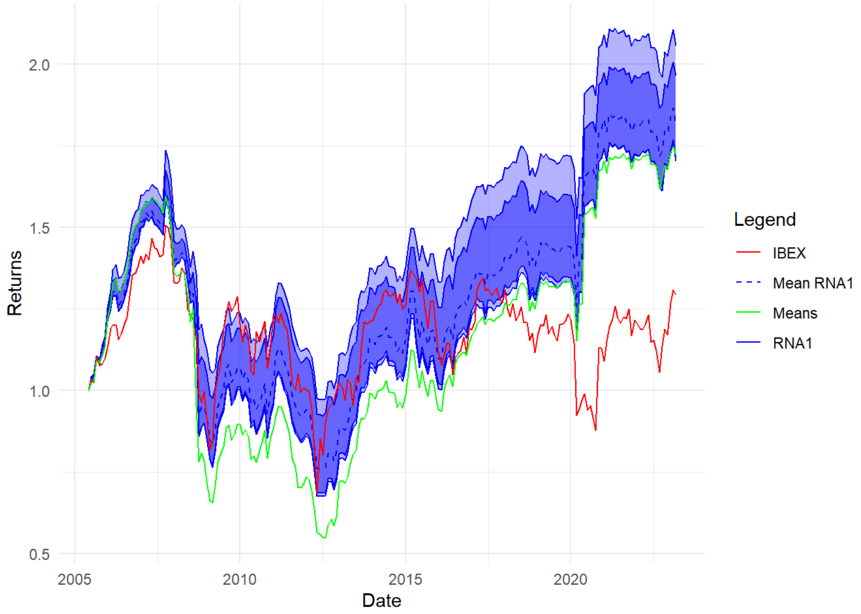
Own elaboration: Through the use of R and Rstudio.

Graph 10: Evolution of the indicators – Entries with three observations



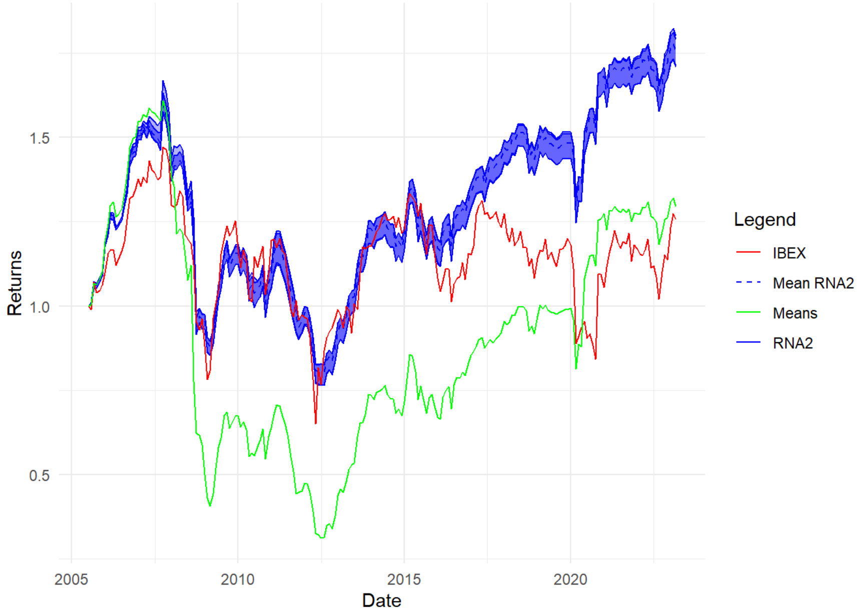
Own elaboration: Through the use of R and Rstudio.

Graph 11: Evolution of portfolios and the IBEX – Entries with one observation



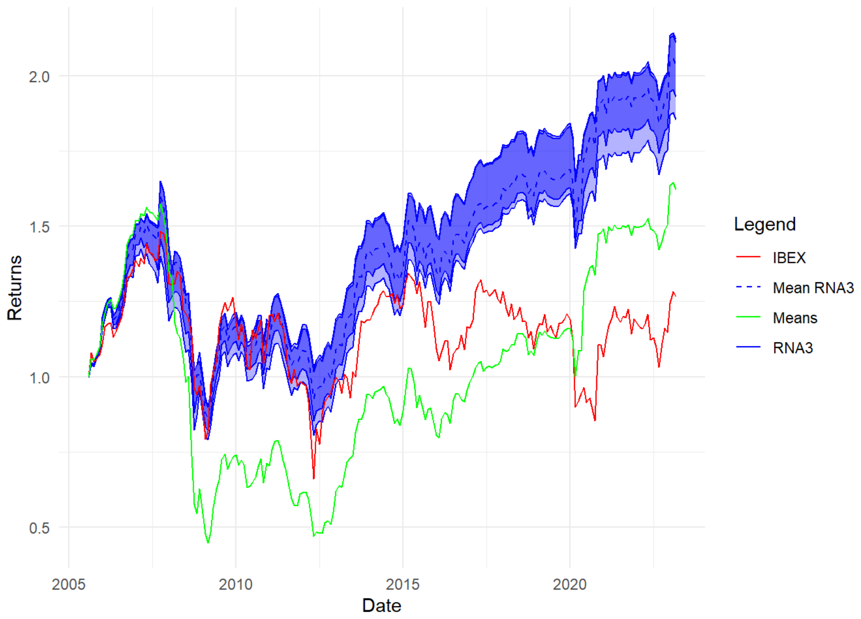
Own elaboration: Through the use of R and Rstudio.

Graph 12: Evolution of the portfolios and the IBEX – Entries with two observations



Own elaboration: Through the use of R and Rstudio.

Graph 13: Evolution of the portfolios and the IBEX – Entries with three observations



Own elaboration: Through the use of R and Rstudio.

Annex.3 Tables

Table 1: Price data structure

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

Own elaboration: Through the use of RStudio with the historical database of “INDITEX”, obtained from https://finance.yahoo.com/, in the period between 05-24-2001 to 05-31-2001.

Table 2: List of listed companies

NAME	TICKERS	SECTOR-SUBSECTOR	MARKET	INDEX	SELECTED
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 - Aerospacial	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

NAME	TICKERS	SECTOR-SUBSECTOR	MARKET	INDEX	SELECTED
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
ATRYS 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		

NAME	TICKERS	SECTOR-SUBSECTOR	MARKET	INDEX	SELECTED
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

NAME	TICKERS	SECTOR-SUBSECTOR	MARKET	INDEX	SELECTED
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

NAME	TICKERS	SECTOR-SUBSECTOR	MARKET	INDEX	SELECTED
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
GRENERGY 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

NAME	TICKERS	SECTOR-SUBSECTOR	MARKET	INDEX	SELECTED
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

NAME	TICKERS	SECTOR-SUBSECTOR	MARKET	INDEX	SELECTED
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

NAME	TICKERS	SECTOR-SUBSECTOR	MARKET	INDEX	SELECTED
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

NAME	TICKERS	SECTOR-SUBSECTOR	MARKET	INDEX	SELECTED
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

Obtained from: The information displayed on the official site of the Spanish Stock Exchanges and Markets, “Empresas Cotizadas”.

Table 3: Amounts of samples used to train the models

Inputs	Total.samples
1	17347
2	17244
3	17141

Own elaboration.

Table 4: Best and best companies according to the results obtained from the calculations of the indicators

1			2			3		
TICKER	R^2	MSE	TICKER	R^2	MSE	TICKER	R^2	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

Own elaboration.

Annex.4 Codes

The code used to conduct the procedure described in the development of the work is presented below.

Data

Data Collection

The first thing that was done was to load the table of companies.

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

Then the ticks of the companies were extracted.

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

Once the ticks of the companies were stored in the ticks variable, we proceeded to download the data corresponding to said companies from Yahoo Finance using the quantmod package from Ryan and 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

}
```

With the objective of conducting an exploratory analysis of the data, it was decided to conduct a visual evaluation of the historical data of the adjusted price for what was executed:

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

})
```

After the visual analysis executed with the previous code fragment, the existence of constant prices was detected, as well as erroneous calculations in the adjusted price corresponding to the first years of some series. In order to eliminate these irregularities, only those observations after January 2005 were selected.

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

  })
```

In order to determine if the data that had been imported had missing values, the following code was executed:

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

emp_con_na <- which(na_values > 0)
```

In order to solve the problem regarding the incorrect recording of the data, it was decided to eliminate those that did not present price variations in more than 10 observations. For which, the returns were first computed by executing the following code, through which the series with missing values were also eliminated.

```
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)

  })
```

Once the returns were computed, those series that presented 0 returns in more than 10 observations were eliminated, for which the following code was executed.

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

returns_emps3 <- returns_emps2[zero_values<10]
```

Indicators

Below is the code used during the process described in the *indicators* sub-heading of chapter 2.

First, the IBEX data were downloaded, the returns of the adjusted price of the same were computed and the values after January 2005 were selected.

```
R
#Importing 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
# Calculating profitability and selecting observations after
# January 2005.
IBEXsel <- IBEXsel |>
  mutate(Return_I = Delt(Adjusted)[,1]) |>
  na.omit() |>
  filter(Date >= "2005-01-31") |>
  select(Date, Return_I)
```

Then the values of the IBEX returns were added to the tables of the returns of the shares of the selected companies, and the variables listed below were computed and added to each of the tables:

- Company volatility
- Index volatility
- Correlation between the profitability of the company and the index
- The Beta between the company and the index

```
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
      )
  })
```

Vectors

Below is the code used during the process described in the sub-heading *vectors* of the heading modeling in Chapter 2.

The first step conducted for the execution of the process explained in the sub-section in question was to create a function that allowed obtaining the consecutive samples for each series used. The function exposed below, as already mentioned, allows obtaining the consecutive samples of a series, for which the parameters mentioned in the sub-heading are used, number of input observations and number of output observations, as well as a parameter conditional with which it is indicated if the vector to be created is input or output.

```
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)
}
```

Below is the code used to create the input vectors corresponding to each of the series. For which two functions were first created, one for the inputs and the other for the outputs.

```
R
# Function that will be used to create the three-dimensional inputs
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)
}

# Function to be used to create the three-dimensional outputs
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)

}
```

Then the lists of three-dimensional vectors of inputs and outputs per company were created, executing the following code another two times with the aim of creating the lists vecs3d2e and vecs3d3e that correspond to those cases in which 2 and 3 inputs were selected.

```
R
# The time horizon is defined
ht <- 1

# The input observations are defined
oe <- 1
```

```
R

# 3d input vectors and 2d output vectors are created for input size 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))
  tibblex <- tibble(
    Date = dates,
    ID = id,
    inputs = inps,
    outputs = outs
  )
  vecs3d1e[[names(returns_indc)[i]]] <- tibblex
}
```

Modelling and training

The code used during the process described in the different sub-sections of the *Modelling and training* section is presented below.

Modelling

For the creation of the models, the first step to execute is to obtain the information of the vectors for which the model is going to be built, which was done by executing the following code:

```
R

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

```
R
# Define parameters
n_ob_pas <- inputsinfo[2]
n_variables <- inputsinfo[3]
n_ob_fut <- outputsinfo[2]
```

Then the structure of the models was constituted with the aspects described in *2.5.1 Modelling*.

```
R
# Input layer
inp <- layer_input(
  shape = c(NULL,n_ob_pas,n_variables))

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

# Output layer
out <- lstm |>
  layer_dense(
    n_ob_fut*1)

# Join the layers to constitute the model
model <- keras_model(inp, out)
# Setting Learning parameters
model |>
  compile(loss = "mse", optimizer = optimizer_sgd(0.0005))
```

ⓘ **Note**

You can find untrained models in the data folder of the repository where this work is located. The models were saved using the hdf5 extension and under the names model1e, model2e and model3e

Training

The first step is to define the function to use for training the models. This function was built with the goal of using the training method described in *2.5.2 Training*. As a result, this function will return a list that will contain the predictions obtained and the model after having been trained and will take as main inputs the tibble called data constituted in the first step that is exposed in this annex's section Modelling and the model as well. of other arguments that allow the use of the function with some main inputs that are not used in the present work.

```R
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)
  }


  # Iteration that will be executed for each unique value in the variable
  # that defines the data sequence. For this reason, it is of vital
  # importance that the data in tibble x be ordered by the sequence
  # variable whose name is passed to seq_var_name

    for (i in 1:length(seq_val)) {
    val_seq <- seq_val[i]

    # Extract inputs and outputs corresponding to the period in the
    # current sequence variable
    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
    # Use inputs to get forecasts for all periods in the sequence
    # variable except for the first
    if(i > 1){
      pred <- modelo |>
        predict(inputs, verbose = 3)
      predictions <- rbind(predictions, pred)
    }

    # Train the model
    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)
}
```

Once the function was created, the predictions were obtained using the following code:

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

predicciones1e <- resultados$predicciones
```

ⓘ **Note**

You can find trained models in the `data` folder of the repository where this work is located. The models were saved using the `hdf5` extension and under the names `model1etd`, `model2etd` and `model3etd`.

As explained in *2.6.1 Predictions* , in addition to the predictions obtained by the models, predictions obtained from the use of the arithmetic mean were computed, to compare with those obtained with the models. To compute these predictions, the following function was created:

```
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)
}
```

Once the function was created, the predictions were obtained using the following code:

R

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

> **Nota**
>
> In addition to what was previously stated, two functions getconfig and plot_modelk were created, in the .Rprofile file of the repository in which this work is found, which allow graphing the structure of the models by using the package Diagrammer by Iannone(2023), as seen in the *Figure 17*. The code to use would be:
>
> R
>
> ```
> # The functions are created to graph the structures used in this work.
> model |>
>   getconfig() |>
>   plot_modelk() |>
>   grViz()
> ```

The procedure exposed in the sections Modelling and Training of this annex was repeated to build the 10 models made from each group of three-dimensional vectors, replacing the call to vecs3d1e with in the first code exposed. vecs3d2e and vecs3d3e, depending on the group of three-dimensional vectors used.

Result

The code used during the process described in the different sub-sections of the *Result* section is presented below.

Predictions

The analysis exposed in *2.6.1 Predictions* was carried out from graphs (see *Graph 8*, *Graph 9* and *Graph 10*), which show the values of the MSE and R^2 indicators for each of the tested structures.

The first step to obtain these graphs was to compute the indicators, for each period of time, for each of the predictions obtained from the different models built with each structure. This is done using the following code.

R

```
# Extract actual outputs
salidas <- data |>
```

```
  filter(
    Date > data$Date[1]
  ) |>
  select(outputs) |>
  pull()
salidas <- salidas[,,1]

# Compute MSE and R2 indicators
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),
  )
```

The different indicators computed for each of the 10 models trained with each of the structures were stored in a list called list_indicadores. This is done using the following code:

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

Once this is done, a list is obtained that contains 10 data frames (indicadores1,…,indicadores10), which in turn contain the values of those of MSE and $R^2$of the predictions obtained by RNA models for each of the companies grouped by date. So, then the graph was built by using the following code.

```
R
# Group the information of the different constructions in a single
# data frame
indi_graf_data <- do.call(cbind,list_indicadores)

# Obtain the average results, for each period of time, using the
# different constructions
indi_graf_data |>
  rowwise() |>
  mutate(
    Date = `indicadores1.Date`,
    meanmse = mean(c_across(contains("mse"))),
    meanr2 = mean(c_across(contains("r2")))
    ) |>
  select(
    Date, meanmse,meanr2
  )|>
  # Graph
  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")
```

In addition to the graphs, the *Table 4* was also used in the analysis of the results, which contains the companies that obtained the best and worst indicators for each structure. To obtain these data, the following code was used:

```
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)
```

Like the indicators computed by date, to save the indicators computed by company, a list called list_indic_emp was created. After having stored the 10 indicator data frames per company in the list, the companies with the best and worst results were extracted using the following code:

```
R
# Group the information of the different constructions in a single
# data frame
ind_emp_t <- do.call(rbind, list_indic_emp)

# Compute the average R2 and MSE by company
ind_emp_t <- ind_emp_t |>
  group_by(ID) |>
  summarize(
    r2 = mean(r2),
    mse= mean(mse)) |>
  ungroup() |>
  arrange(desc(r2))

# Obtain the 10 companies with the best and worst indicators
mejores10 <- head(ind_emp_t,10)

peores10 <- tail(ind_emp_t,10)
```

And using the above variables and using the rbind() and cbind functions was how the *Table 4* was created.

Portfolio composition

This section explains how the analysis of the comparison of the results obtained by the different portfolios was carried out (see *Gráfica 11*, *Gráfica 12* and *Gráfica 13*). For this, it is first necessary to obtain the composition of the portfolios, by date, from the predictions obtained by using the arithmetic means and the RNA models.

To calculate the composition of the portfolios, the R package quadprog by Berwin A. Turlach R port by Andreas Weingessel <Andreas.Weingessel@ci.tuwien.ac.at> Fortran contributions from Cleve Moler dpodi/LINPACK) (2019) was used. Below is the code used to find the composition of the portfolios from the predictions of the mean::

```
R
# A data frame was created in which all the information was stored:
#   - IBEX values, as a reference index
#   - Values of the predictions, both those obtained by the RNA
#     model and by the arithmetic means

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)

# From the data frame DATA were created:
#    - A data frame whose columns are the actual data of each of
#      the companies for each of the time periods for which
#      predictions were obtained.
#    - A data frame whose columns are the data obtained by using
#      the arithmetic means of each of the companies for each of the
#      time periods for which predictions were obtained.

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
# The data frame was created in which the composition of the
# portfolios was stored for each of the periods for which
# the prediction was obtained.
weightsm <- data.frame()

# Iteration by which the composition of the portfolios is found

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){

    # The data frame is created that includes the data to be used
    # to find the composition of the portfolio, this is created by
    # the actual data to date and the forecast for the next period.
    datamQP <- pvtReal |>
      filter(Date < unique(data$Date)[-1][i]) |>
      rbind(pvtMeans |>
              filter(Date == unique(data$Date)[-1][i])
      )

    # Eliminate those companies that do not have actual or
    # forecast data
    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
    }

    # Extract the forecasts
    returnm <- carteram[dim(carteram)[1], -1] |>
      as.matrix() |>
      t()

    # Calculate the covariance matrix
    covmm <- cov(carteram[, -1], use = "complete.obs")
    npcovmm <- nearPD(covmm)$mat |>
      as.matrix()
    # Extract the number of companies
    n <- ncol(npcovmm)
R
```

```
    # Find the composition of the portfolio
    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{}
    }

    # Save portfolio composition
    names(qp_outm) <- names(carteram[, -1])
    weightsm <- bind_rows(weightsm, qp_outm)
  }

  setTxtProgressBar(pb,i)
}

close(pb)

# Replace actual weights and observations with missing values with zero
pvtReal[is.na(pvtReal)] <- 0
weightsm[is.na(weightsm)] <- 0
```

Then, to find the profitability of the portfolio, the compositions were multiplied by the real returns, it was assumed that one was invested in the first period and a cumulative sum was made throughout the values to obtain the behavior of the profitability throughout the period. time.

```
R
# Finding the returns of the portfolios formed from the predictions
# of the arithmetic mean

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
)
```

The same steps that were conducted to find the behaviour of the profitability of the portfolios from the arithmetic means were conducted to find the behaviour from the predictions obtained by the RNA model as seen in the code below.

```
R
# From the DATA data frame, a data frame was created whose columns
# are the data obtained by using the RNA model of each of the
# companies for each of the time periods for which predictions
# were obtained.

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

# The data frame was created in which the composition of the portfolios
# was stored for each of the periods for which the prediction
# was obtained.
weightse <- data.frame()

# Iteration by which the composition of the portfolios is found

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

```
# The data frame is created that includes the data to be used
# to find the composition of the portfolio, this is created by
# the actual data to date and the forecast for the next period.
dataeQP <- pvtReal |>
  filter(Date < unique(data$Date)[-1][i]) |>
  rbind(pvtRNA |>
          filter(Date == unique(data$Date)[-1][-1][i])
        )
# Eliminate those companies that do not have actual or
# forecast data
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
}

# Extract forecasts
returne <- carterae[dim(carterae)[1], -1] |>
  as.matrix() |>
  t()

# Calculate the covariance matrix
covme <- cov(carterae[, -1], use = "complete.obs")
npcovme <- nearPD(covme)$mat |>
  as.matrix()
# Extract the number of companies
n <- ncol(npcovme)

# Find the composition of the portfolio
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
    # Save portfolio composition
    names(qp_oute) <- names(carterae[, -1])
    weightse <- bind_rows(weightse, qp_oute)
  }

  setTxtProgressBar(pb,i)
}

close(pb)

# Replace weights with missing values with zero
weightse[is.na(weightse)] <- 0
```

Then, to find the profitability of the portfolio, the compositions were multiplied by the real returns, it was assumed that one was invested in the first period and a cumulative sum was made throughout the values to obtain the behaviour of the profitability throughout the period time.

```
R
# Finding the returns of the portfolios formed from the predictions
# of the RNA model

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
)
```

Then, as with the indicators, a list_ret_RNA list was created in which the data frames of the different models built with each of the structures were stored. Then the following code was executed to obtain the graph.

```
R
# Finding the behaviour of the returns of the IBEX for the period

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")) +
```

```r
  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()
```