MASTER

DATA ANALYTICS FOR BUSINESS

MASTER'S FINAL WORK

PROJECT

Machine Learning for Stocks Prediction

RAFAEL ANDRÉ CARVALHO PINHEIRO

March - 2024

MASTER

DATA ANALYTICS FOR BUSINESS

MASTER'S FINAL WORK

PROJECT

Machine Learning for Stocks Prediction

RAFAEL ANDRÉ CARVALHO PINHEIRO

SUPERVISION:
PROF. CARLOS J. COSTA

March - 2024

Quero dedicar este trabalho
Aos meus PAIS, AVÓS e IRMÃO,
Que sempre me apoiaram em tudo!

"Pra cima,
Sempre pra cima"
- BisAvó Isaura

ACRONYMS

AHC - Agglomerative Hierarchical Clustering

ANN - Artificial Neural Network

ARIMA - Autoregressive Integrated Moving Average

CNN - Convolutional Neural Network

DBSCAN - Density-Based Spatial Clustering of Applications with Noise

FFN - Feed-Forward Neural Network

LSTM - Long Short-Term Memory

MFW - Master's Final Work

MAE - Mean Absolute Error

MAPE - Mean Absolute Percentage Error

MSE - Mean Squared Error

POST-DS - Process Organization and Scheduling electing Tools for Data Science

RNN - Recurrent Neural Network

RMSE - Root Mean Squared Error

SP500 - Standard & Poor's 500 Index

ABSTRACT

The stock market is intrinsically risky and difficult to predict, as it is composed of time series that have nonlinear, complex, and dynamic behavior. Therefore, predicting these time series requires the use of machine learning algorithms capable of handling the inherent volatility and complexity of the stock market.

In this MFW, the objective is to make monthly predictions of stock returns belonging to the SP500. For this, in the first phase, a clustering algorithm - Hierarchical Clustering - will be used to find correlated stock groups, and in the second phase, a deep learning algorithm - LSTM - will be used to predict the monthly returns of the groups of stocks found in the previous phase. That is, the prediction of the return of a stock A will be made based on stocks that have monthly returns correlated with stock A. Thus, a multivariate analysis will be done so that the prediction of the return of stock A depends not only on its previous returns, but also on the previous returns of stocks that are correlated with stock A. Consequently, it will be possible to capture the dynamics of multiple time series simultaneously and take advantage of dependencies between these series to obtain better predictions.

To conclude, we explore the possibility that if most predictions in a group of correlated stocks indicate an upward movement, then all stocks belonging to that group are expected to have an upward movement. In the end, the results obtained show that the methodology used allows: (a) to obtain predictions of returns that are far from the actual returns, (b) good returns in the portfolio, (c) to reduce the impact of incorrect predictions on the portfolio.

KEYWORDS: correlated stocks; stocks clustering; stocks prediction; machine learning; deep learning; time series; monthly returns

RESUMO

O mercado de ações é intrinsecamente arriscado e difícil de prever, uma vez que é composto por séries temporais que apresentam comportamento não-linear, complexo e dinâmico. Por conseguinte, a previsão dessas séries temporais requer a utilização de algoritmos de aprendizagem automática capazes de lidar com a volatilidade e complexidade inerentes ao mercado de ações.

Neste MFW pretende-se realizar previsões mensais dos retornos das ações que pertencem ao SP500. Para tal, numa primeira fase, será utilizado um algoritmo de agrupamento - Hierarchical Clustering - para encontrar grupos de ações correlacionadas e, numa segunda fase, um algoritmo de aprendizagem profunda - LSTM - será utilizado para prever os retornos mensais dos grupos de ações encontrados na fase anterior. Ou seja, a previsão do retorno de uma ação A será feita com base em ações que têm retornos mensais correlacionados com a ação A. Deste modo, será feita uma análise multivariada, de maneira que a previsão do retorno da ação A dependa não apenas dos seus retornos anteriores, mas também dos retornos anteriores das ações que estão correlacionadas com a ação A. Consequentemente, será possível captar a dinâmica de múltiplas séries temporais simultaneamente e aproveitar as dependências entre essas séries para obter melhores previsões.

Para concluir, explora-se a possibilidade de que se num grupo de ações correlacionadas, a maioria das previsões indicar uma subida, então espera-se que todas as ações pertencentes a esse grupo tenham um movimento de subida. No final, os resultados obtidos mostram que a metodologia utilizada permite: (a) obter previsões de retornos que estão longe dos retornos reais; (b) bons retornos no portfólio; (c) reduzir o impacto de previsões incorretas no portfólio.

PALAVRAS-CHAVE: ações correlacionadas; agrupamento de ações; previsão de ações; aprendizagem automática; aprendizagem profunda; séries temporais; retornos mensais

CONTENTS

ACRONYMS i

ABSTRACT ii

RESUMO iii

LIST OF FIGURES vi

LIST OF TABLES viii

ACKNOWLEDGMENTS ix

1. INTRODUCTION 1

1.1. Context ... 1

1.2. Motivation ... 2

1.3. Objectives ... 4

1.4. Structure of the Document ... 5

2. LITERATURE REVIEW 6

2.1. Methods ... 6

2.1.1. Hierarchical Clustering ... 6

2.1.2. Long Short-Term Memory ... 8

2.2. Related Work ... 11

3. METHODOLOGY 14

3.1. Obtaining and preparing the datasets ... 14

3.2. Stocks clustering ... 17

3.2.1. Clusters creation ... 17

3.2.2. Clusters analysis ... 22

3.3. Stocks prediction ... 25

4. RESULTS AND DISCUSSION **29**

5. CONCLUSION **35**

REFERENCES **38**

APPENDIX **40**

LIST OF FIGURES

FIGURE 1 - COMPARISON BETWEEN FFN (A) AND RNN (B) 8

FIGURE 2 - AN UNROLLED RECURRENT NEURAL NETWORK 9

FIGURE 3 - LSTM UNIT/CELL 10

FIGURE 4 - COMPARISON OF DATASETS MADE BY VÁSQUEZ SÁENZ ET AL. (2023) 12

FIGURE 5 - INPUT AND OUTPUT DATASET STRUCTURE 17

FIGURE 6 - ORIGINAL CORRELATION MATRIX 19

FIGURE 7 - CLUSTERED CORRELATION MATRIX 20

FIGURE 8 - ANALYSIS OF RETURNS VS VOLATILITY 23

FIGURE 9 - SAMPLE OF THE TIME SERIES OF STOCK RETURNS FOR CLUSTER 1 (ABOVE) AND CLUSTER 2 (BELOW) 25

FIGURE 10 - ILLUSTRATION OF THE CROSS-VALIDATION PROCESS EMPLOYED 26

FIGURE 11 - TYPICAL WORKFLOW WHEN BUILDING A NEURAL NETWORK 27

FIGURE 12 - LSTM ARCHITECTURE 28

FIGURE 13 - ILLUSTRATION OF THE RULE TO BUY 31

FIGURE 14 - ILLUSTRATION OF THE RULE TO SELL 32

FIGURE 15 - EVOLUTION OF ACCUMULATED RETURNS OBTAINED IN CLUSTERS 1 AND 2 .. 32

FIGURE 16 - NEW EVOLUTION OF ACCUMULATED RETURNS OBTAINED IN CLUSTERS 1 AND 2 33

FIGURE 17 - ILLUSTRATION OF THE METHODOLOGY EMPLOYED 35

FIGURE 18 - DENDROGRAM ILLUSTRATING THE MERGING OF INDIVIDUAL STOCKS BASED ON THEIR RELATIVE DISTANCE 40

FIGURE 19 - CORRELATION MATRIX FOR CLUSTER 1 40

FIGURE 20 - CORRELATION MATRIX FOR CLUSTER 2 41

FIGURE 21 - CORRELATION MATRIX FOR CLUSTER 3 41

FIGURE 22 - OBSERVED RETURNS IN THE TEST SET FOR CLUSTER 1 41

FIGURE 23 - PREDICTED RETURNS IN THE TEST SET FOR CLUSTER 1 42

FIGURE 24 - OBSERVED RETURNS IN THE TEST SET FOR CLUSTER 2 42

FIGURE 25 - PREDICTED RETURNS IN THE TEST SET FOR CLUSTER 2 42

LIST OF TABLES

TABLE 1 - RELATED WORK 13

TABLE 2 - STOCKS DISTRIBUTION BY SECTORS AND INDUSTRIES 15

TABLE 3 - CLUSTERS OF STOCKS OBTAINED 21

TABLE 4 - CENTROIDS OF THE CLUSTERS 24

TABLE 5 - FINAL PARAMETERS CHOSEN 28

TABLE 6 - METRIC VALUES OBTAINED 30

ACKNOWLEDGMENTS

Este trabalho final de mestrado vem simbolizar não apenas o término do mestrado, mas também o fim do meu percurso académico num todo. Ao longo de todo este percurso tive a sorte de ter uma família que sempre me apoiou em tudo, em particular na minha mudança de Coimbra para Lisboa para poder realizar este mestrado. Esta mudança proporcionou-me um ano bastante diferente que ficará certamente recordado durante muitos anos. Durante todo este percurso fiz grandes amizades que ficarão para a vida e tive excelentes professores que não esquecerei. Um obrigado a todos os que fizeram parte deste percurso - família, amigos e professores - por me terem tornado na pessoa que sou hoje. Por fim, quero agradecer ao Professor Carlos Costa por me ter acompanhado e ajudado durante os últimos seis meses no desenvolvimento deste projeto.

Chapter 1

1. INTRODUCTION

1.1. Context

Stock prediction is the prediction of a time series. A time series is a sequence of observations typically measured at uniform time intervals. These observations are used to develop a model that describes the data's inherent structure. By understanding past historical patterns, the model will be able to predict future observations in the time series.

In fact, stock prediction is a challenging problem due to the difficulty and uncertainty of the stock market. In addition, stock markets are nonlinear, complex, and dynamic. Researchers have tested and analyzed several models to predict stock price movement accurately. Some have used models that assume a linear relationship between past and future observations, such as ARIMA models. However, these observations have nonlinear dependencies (Amini et al., 2021), so the use of linear models can result in incorrect predictions due to biased coefficient estimates. Other researchers have used traditional machine learning algorithms, such as k-nearest neighbors, support vector machines, random forest, and logistic regression. However, these algorithms only consider one observation to make the prediction of the next observation, which makes them unable to capture the temporal correlations existing between observations. Finally, to overcome the limitations of the models described earlier, deep learning models, in particular the RNNs, began to be used. These models can both capture nonlinear dependencies between observations, which ARIMA cannot capture, and they can also capture the temporal correlations between observations, which traditional machine learning models cannot capture. For these reasons, in the past few years, many researchers have begun to use deep learning models to make time-series predictions.

In the first phase of this MFW, a clustering algorithm will be used to create correlated groups of stocks. These groups of correlated stocks are time series that are correlated. Suppose a group k of 12 stocks was created. To predict the time series of these 12

stocks, extracting the temporal correlations within each time series and the spatial correlations among the different time series will be essential. In order to capture these two types of correlations, a deep learning algorithm called LSTM will receive a multivariate input composed of the 12 time series from group k and will return a multivariate output comprising the predictions for the 12 time series.

However, regardless of the model used, there will always be incorrect predictions. For this reason, in this MFW, an approach will be used to reduce the impact of incorrect predictions on the portfolio. This approach explores the possibility that if most predictions in a group of correlated stocks indicate an upward movement, then all stocks belonging to that group are expected to have an upward movement. In other words, if the predictions for eight stocks in group k indicate an upward movement, and the remaining four predictions indicate a downward movement, then a buy of all stocks belonging to group k will be made. The same principle applies if most predictions indicate a downward movement.

In this way, the approach used considers two assumptions: stocks belonging to the same group almost always exhibit similar behavior; most predictions are correct. If these two assumptions are verified, this approach is expected to be beneficial and result in higher returns.

1.2. Motivation

Nowadays, investing in the financial market is much easier than it was a few years ago. This is mainly due to the increasingly significant presence of brokers with online platforms that allow us, for example, to buy a stock from a smartphone. These online platforms are designed with a simple and intuitive interface so that anyone can invest easily, quickly, and from anywhere. For this reason, in recent years there has been a significant increase in new individual investors, also known as "retail investors".

As soon as these new investors begin to explore the financial market, they discover a wide range of investment options, such as forex, cryptocurrencies, indices, commodities, stocks and ETFs. The initial question that arises is: "Where am I going to invest?". Investors who choose stocks intend to obtain a piece of a company (or several). Subsequently, a second question emerges: "What strategy can I use to generate profits?".

If the investor allocates the entire portfolio to a single stock, it is considered a risky investment, as the fall in the price of that stock due to unforeseen circumstances can result in a large loss.

Hence, a well-known strategy emerges in the field of stock investment: diversification. This strategy aims to mitigate the risks associated with market volatility by distributing the portfolio across several stocks. The fundamental idea behind diversification is that a fall in the price of one stock can be offset by an increase in the price of another stock, thereby contributing to a more stable return in the long-term.

However, the benefits of diversification can be nullified if we diversify the portfolio by several stocks that are correlated, that is, by stocks that have similar movements. What happens is that if we diversify the portfolio by stocks that are very correlated, then in the case of a fall in the price of one stock, it is likely that the same will happen with the other stocks belonging to the same portfolio, causing a large loss. Thus, a good diversification strategy does not consist of just diversifying the portfolio by several stocks. For this strategy to work, it is essential to select a set of stocks that are not correlated in order to minimize the possibility of a single stock having a high negative impact. Therefore, a third question arises: "How to find stocks that are not correlated?". To answer this question, many investors use an approach that has certain characteristics of stocks as criteria, such as the sector and industry in which they operate. Based on this approach, if we buy one stock in the energy sector and another stock in the financial sector, then it means that we are diversifying the portfolio in the right way because we are investing in stocks that are not inherently correlated since they belong to different sectors. This approach assumes, therefore, that stocks belonging to the same sector are correlated, as they are stocks relating to companies with similar business activities or products, so the fluctuation of their stock prices will be influencing each other.

Nevertheless, there may be stocks from one sector that are correlated with stocks from another sector. This was verified by Yilang Lu (2018), who found, through clustering algorithms, stocks that are more correlated with stocks from another sector than with stocks belonging to his own sector. Thus, we can state that the theory of the approach described earlier is not always true, so more advanced methods, such as clustering algorithms, can be more effective in identifying stocks that are not correlated.

In this way, a correct diversification of the portfolio into stocks that are not correlated can be a highly beneficial strategy in the long-term. However, in the short-term, this strategy has the net impact of gradual and steady performance and smoother returns that never move too quickly up or down. Consequently, there may be investors who prefer to take higher risks in their investments so that they can potentially achieve greater returns in the short-term. Thus, for those investors who prefer short-term investments, a fourth question arises: "Is it beneficial to invest in uncorrelated stocks in the short-term?". Probably not as in short-term periods, the return can vary widely. Consequently, the fifth question arises: "Is it beneficial to invest in correlated stocks in the short-term?". Finding the answer to this question will be the main objective of this MFW. For this, in the first phase, a clustering algorithm - Hierarchical Clustering - will be used to find correlated stock groups, and, in the second phase, a deep learning algorithm - LSTM - will be used to predict the monthly behavior/motion of the groups of stocks found in the previous phase.

1.3. Objectives

As already mentioned, the main objective of this MFW is to answer the question: "Is it beneficial to invest in correlated stocks in the short-term?". To achieve this, we will focus on the following steps:

- through a clustering algorithm - Hierarchical Clustering - find correlated groups of stocks. Each group found will correspond to a cluster;
- through a deep learning algorithm - LSTM - make predictions of monthly returns of the stocks belonging to the clusters created;
- analyze the predictions obtained and check whether it is beneficial to invest in correlated stocks. It will be beneficial if the assumptions described at the end of section 1.1 are almost always verified.

1.4. Structure of the Document

This document is composed of five chapters to achieve the described objectives and answer the research question of this MFW. Besides this Chapter 1 - Introduction - the remaining chapters are as follows:

- **Literature Review (Chapter 2)** - this chapter is divided into two parts. The first part presents the theory underlying the algorithms that will be applied in Chapter 3: Hierarchical Clustering and LSTM. The second part provides an analysis of the current state of research in relation to the subject of interest, encompassing market data clustering, forecasting models, and forecasting models combined with clustering.

- **Methodology (Chapter 3)** - this chapter is divided into three parts. The first part describes what data will be used and how it was obtained. The Hierarchical Clustering algorithm is applied in the second part to create clusters with correlated stocks, followed by an analysis of the clusters created. In the third part, an Artificial Neural Network - LSTM - is applied to make monthly predictions of the stocks that have been grouped.

- **Results and Discussion (Chapter 4)** - in this chapter, the results obtained from the predictions of monthly returns are presented and analyzed. The answer to the research question of this MFW is found in this chapter.

- **Conclusion (Chapter 5)** - in this final chapter, a summary of the main findings and the path taken in the development of the approach used is provided. Furthermore, to conclude, the key conclusions and contributions of this MFW are presented, as well as possible aspects of improvement to be analyzed and developed in the future.

Chapter 2

2. LITERATURE REVIEW

In this chapter the objective is to make a literature review that allows to understand the theory underlying the methods/algorithms that will be used in the development of this MFW. In addition, will also be made an analysis of some articles that align with what will be done in this project and whose conclusions will allow to provide a solid basis for justifying the methodology used. Thus, through this chapter it will be possible to draw the path to be carried out in the next chapter.

2.1. Methods

In this MFW, two algorithms will be used: an unsupervised learning algorithm and a supervised learning algorithm. The unsupervised learning algorithm chosen for the creation of clusters with similar stocks was Hierarchical Clustering, and the supervised learning algorithm chosen to predict the returns of stocks that are in the clusters was Long Short-Term Memory.

As in any scientific field, before going to practice, it is essential to understand the subjacent theory. Thus, the objective of this section is to explain the theory behind the algorithms that will be applied in the next chapter.

2.1.1. Hierarchical Clustering

Hierarchical clustering is a clustering algorithm that takes into account both the distance between stocks and the distance between stock groups. Thus, the main objective of hierarchical clustering is to group elements based on the principle of maximizing intra-class similarity and minimizing inter-class similarity. That is, within the dataset, clusters are formed so that stocks that are similar are grouped together, and stocks that are very different fall into other clusters (Xu & Tian, 2015).

Compared to K-Means, this algorithm has the advantage of not requiring the specification of an initial number of clusters, as it assumes that the data can be successively grouped into increasingly different clusters. Furthermore, the result of hierarchical clustering is fixed, while in K-Means the result depends on the randomly chosen initial points.

There are two approaches of hierarchical clustering: agglomerative (bottom-up), where the data points are divided into different clusters and then aggregated as the distance decreases; divisive (top-down), where all the data points are aggregated in a single cluster and then divided in different clusters as the distance increases. We will use the most common approach - agglomerative hierarchical clustering (AHC) - that departs from the individual data points and computes a similarity matrix containing all mutual distances (Xu & Tian, 2015).

These distances can be of 3 types: distance between clusters, distance between elements, and distance between element and cluster. On the one hand, the distance between elements (stocks) can be determined, for example, using an Euclidean distance. On the other hand, the distance between clusters is done using the notion of linkage, which can be of 4 types:

- complete linkage: the largest distance between the observations of two clusters;
- single linkage: smallest distance between the observations of two clusters;
- average linkage: average distance between the observations of two clusters;
- centroid linkage: distance between the centroids of two clusters.

Once the similarity matrix (containing all mutual distances) is calculated, the AHC algorithm proceeds with N-1 steps, where N is the total number of elements (stocks). At each step, the algorithm merges the most similar clusters until no distinct clusters remain. After each merge, the similarity matrix is updated to incorporate the newly formed clusters, progressively reducing the size of the matrix.

This algorithm, combined with the application of a method called *Quasi-Diagonalisation* (a technique that reorganizes the covariance matrix so similar stocks will be placed together), will allow to create clusters with stocks that have a similar behavior, as we will see in section 3.2.1.

2.1.2. Long Short-Term Memory

Deep learning is a subfield of machine learning concerned with algorithms - Artificial Neural Networks (ANN) - which were originally inspired in the way the brain processes information. Today, deep learning models have achieved great success in long-term time series prediction tasks, where they outperform conventional linear models and machine learning algorithms. This better performance of the ANNs for time series forecasting is mainly due to their nonlinear modeling capability, which allows better capture of time series behavior, which is also nonlinear.

In fact, there are several types of ANNs with various applications in different areas. On the one hand, there are ANNs, such as Feed-Forward Neural Networks (FFN), that treat the feature vectors for each sample as independent and identically distributed. Therefore, they are considered ANNs without memory since they do not consider prior data points when evaluating the current observation. On the other hand, there are ANNs that handle sequential input, including time series, audio, and natural language text. These ANNs are called Recurrent Neural Networks and are characterized by connections with loops, allowing feedback and memory to the networks over time.

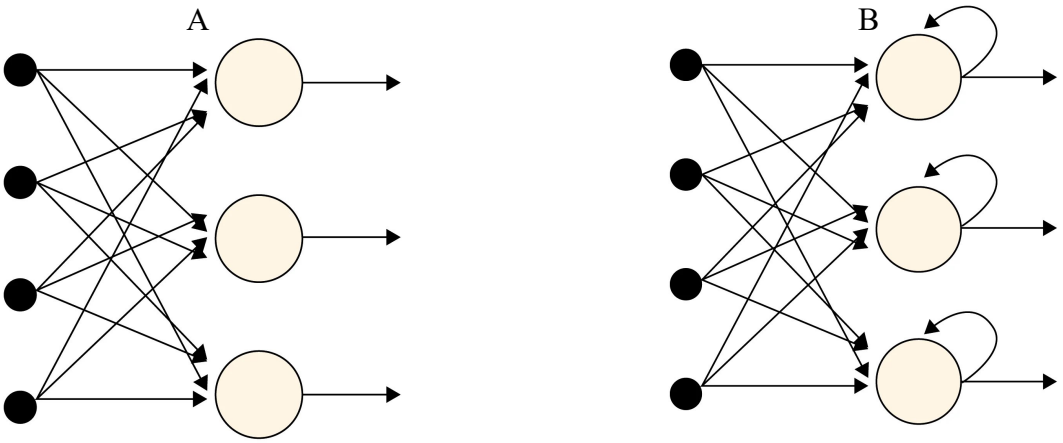

Figure 1 - Comparison between FFN (A) and RNN (B)

Figure 2 demonstrates a conventional forward RNN being extended (or unfolded/unrolled) into a complete network. It can be divided into three layers: the input layer, which receives a sequence of inputs; the hidden layer, which contains recursive edges that correspond to information persistence; and the output layer, which represents the predicted output. The symbols U, W, and V correspond to the parameters for input,

hidden and output layers, respectively, that are calculated during the training phase, and then are used to connect input, hidden and output layers.

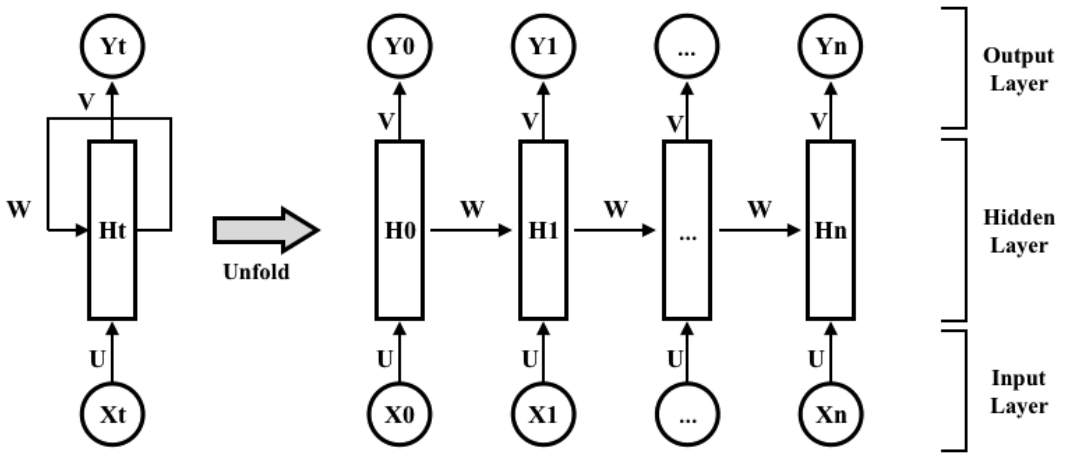

Figure 2 - An unrolled recurrent neural network

However, RNNs have two major problems (Hochreiter, 1998) associated with the repeated multiplication of gradients (derivatives of the loss with respect to the network's parameters) during backpropagation over many time steps: vanishing and exploding. The first problem, vanishing gradient, is the most common and occurs when the gradients become extremely small as they are back-propagated through the layers of the network during training. The second problem, the exploding gradient, is the opposite problem of the vanishing gradient and refers to a situation during the training where the gradients become extremely large as they are back-propagated through the layers of the network.

To overcome these problems, Long Short-Term Memory (Hochreiter & Schmid-huber, 1997) constructs the "gate" to regulate how much past information a unit maintains in its current state and when to reset or forget this information. As a result, LSTMs are able to learn dependencies over hundreds of time steps, which allows them to capture important features from inputs and store the information over a long period of time. Thus, LSTMs have achieved good results in long-term forecasting.

In general, the critical components of LSTM network architecture, which allow to regulate the flow of data into and out of the cells, consist of three gates: input, output, and forget gates (as we can see in Figure 3). The input gate controls the amount of new data that may flow into the cell. The output gate controls the amount of data is sent from the

cell to the rest of the network. The forget gate controls how much of the cell's state should be discarded in order to manage the memory of the network. In conjunction with the memory cell, these gates enable LSTM to learn and remember long-term dependencies in sequential data (Hochreiter & Schmidhuber, 1997).

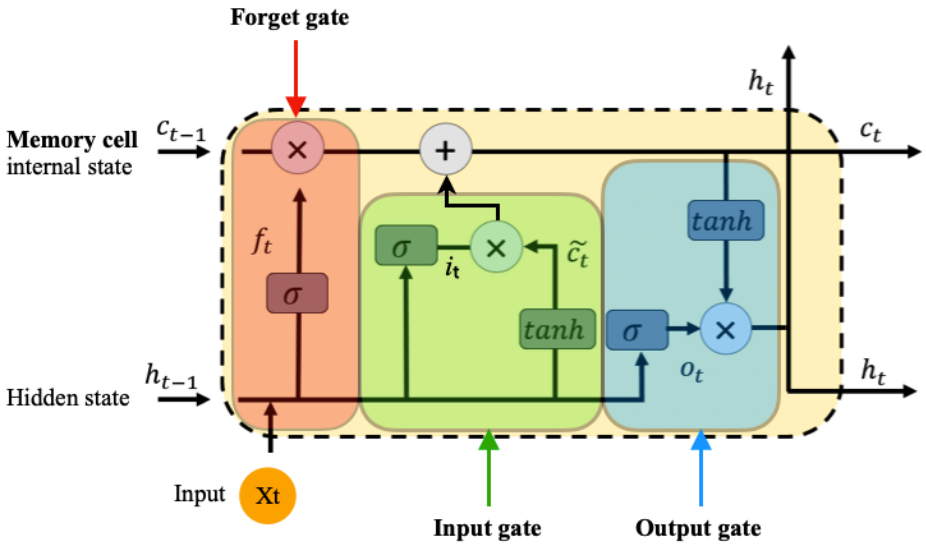

Figure 3 - LSTM unit/cell

Figure 3 shows a graphical representation of an LSTM unit, where: X_t - input data; f_t - forget gate; i_t - input gate; o_t - output gate; $\tilde{c}_t$ - cell update; c_t - cell state; h_t - hidden state. The equations of the elements present in an LSTM unit are as follows:

$$f_t = \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f) \quad [2.1]$$

$$i_t = \sigma(W_i \cdot x_t + U_i \cdot h_{t-1} + b_i) \quad [2.2]$$

$$o_t = \sigma(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o) \quad [2.3]$$

$$\tilde{c}_t = tanh(W_c \cdot x_t + U_c \cdot h_{t-1} + b_c) \quad [2.4]$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \quad [2.5]$$

$$h_t = o_t \cdot tanh(c_t) \quad [2.6]$$

An LSTM can be used in classification problems (where the output is a class) and regression problems (where the output is a numeric value). In a classification problem, the LSTM generates a probability distribution over all potential classes for each input in

a sequence - for example, the probability of a stock price going up or down in the next month. In a regression problem, the LSTM predicts a continuous value for each input in the sequence - for example, the price that a given stock will have in the next month.

2.2. Related Work

The study of stock market behavior has been a topic of debate for more than 100 years. Bachelier's seminal paper in 1900 (Bachelier, 1900) could be considered a pioneering theoretical attempt to model bond prices. Nowadays, this study has evolved into the application of complex models and algorithms that allow predictions of stock market behavior to be made. Those algorithms are used to predict even in the context of new markets, like cryptocurrencies (Aparicio et al., 2022). In particular, numerous articles have already dealt with both clustering and stock prediction algorithms. Even some of them exploit the power associated with the combination of clustering and prediction algorithms in stock analysis. These articles are the most relevant to the development of this MFW.

In Table 1 are some of the articles that contributed to the development of the methodology and approach used in this MFW. In these articles, it is observed that various clustering algorithms - k-means, DBSCAN, and Snob - combined with diverse prediction algorithms - ARIMA, LSTM, CNN, and genetic algorithm - have been used in different types of datasets. Some of these articles explore the possibility that the use of similar sets of stocks, determined through clustering, can improve the performance and results of a prediction model compared to a model that does not take into account the data derived from clustering. For example, in the study by Vásquez Sáenz et al. (2023), three types of datasets (Figure 4) are compared to determine which of them allows for better results/metrics in stock prediction using ARIMA and LSTM models. The description that this article makes of the datasets used is as follows: "In the first case, we train with only the previous values of the stock for previous time-steps. In the second case, we use the previous data of *all* stocks to train the forecast model. In the third and last case, our proposed approach uses the previous values of all stocks that are in the same cluster as Stock 1". In the end, the results obtained allowed us to conclude that LSTM models outperform ARIMA and that the dataset that allowed to obtain better predictions was the third, that is, the dataset that resulted from clustering.

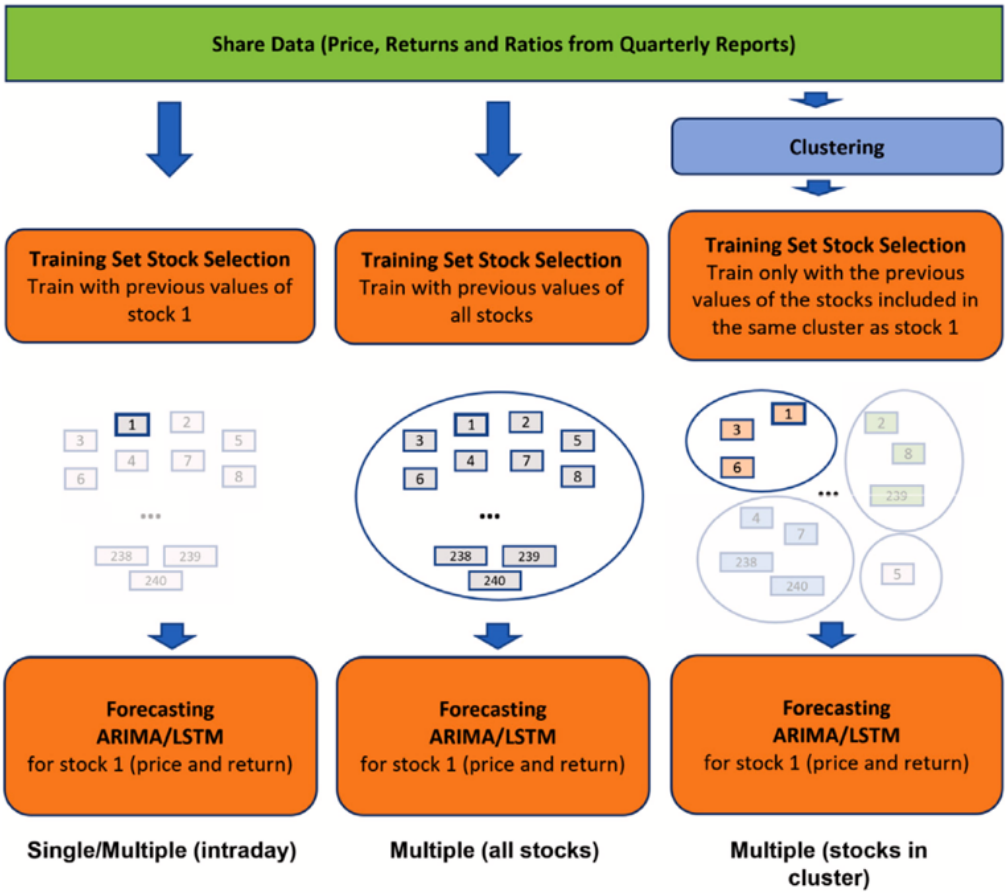

Figure 4 - Comparison of datasets made by Vásquez Sáenz et al. (2023)

Based on the articles and researchers referenced in Table 1, the main conclusions from their research that were considered in the development of the methodology of this MFW were:

- LSTM networks outperform traditional forecasting methods for time series, such as ARIMA models;
- clustering improves the predictions made by neural networks when compared to models without clustering;
- using data from unrelated stocks adds noise and deteriorates the forecasting ability of LSTM models.

These conclusions, which are properly supported and explained in the articles in Table 1, provide a solid basis for justifying the methodology used in this MFW.

Table 1 - Related Work

References	Objectives	Data & Features	Methodology	Conclusions & Results
Vásquez Sáenz et al. (2023)	Using stock clustering models to improve: a. the prediction of stock prices b. the returns of trading algorithms	Data of 240 companies from 2017 to 2022, including: a. financial ratios b. prices c. daily returns	a. cluster stocks using k-means and several alternative distance metrics b. for each cluster, train ARIMA and LSTM forecasting models to predict the daily price of each stock in the cluster c. analyze the returns of different trading algorithms	a. LSTM models outperform ARIMA b. forecasting is improved by using clustering methods c. using data from unrelated stocks adds noise and deteriorates the forecasting ability of LSTM models
André et al. (2020)	Predict stock price movements based on similar stock sets using clustering algorithms	Historical stock data and Google Trends news of 50 stocks	a. identifying similar stock sets using k-means and DBSCAN b. LSTM neural network for forecasting	Using clustering algorithms to identify stock clusters improves the accuracy, f1-score, recall, and precision of the forecasting model compared to models for a single stock
Hadavandi et al. (2010)	Develop an integrated approach for stock price forecasting	Stock price data from the IT and Airlines sectors	a. stepwise regression analysis to determine factors which have most influence on stock prices b. clustering the data of the time series of stocks using self-organized maps c. feeding this data into a genetic algorithm	Good results in terms of the accuracy of the predictions *vis-à-vis* other methods
Bandara et al. (2020)	a. develop a prediction model for forecasting time series b. improve accuracy by clustering similar time series and using LSTM networks	a. datasets consist of monthly time series b. each dataset is comprised of similar time series	a. "Snob" clustering algorithm to identify subgroups of similar time series b. using LSTM networks on subgroups of similar time series for forecasting	a. LSTM networks outperform traditional forecasting methods for time series b. clustering improve the predictions made by the neural networks, when compared to models without clustering
Long et al. (2020)	Build a deep neural network to predict stock price trend	Transaction records and public market information for each stock	a. using the transaction records to identify and cluster investment patterns b. neural networks based on CNN and Bi-directional LSTM to predict price movement	71% of accuracy, which demonstrates the effectiveness, robustness, and practicability of the model

Chapter 3

3. METHODOLOGY

The purpose of this chapter is to explain the methodology used to predict monthly stock returns. The predictions obtained will be analyzed in the next chapter.

A data science process involves a sequence of tasks and activities that must be structured and organized to successfully align with the overall project management. In order to assist good management in the development of this MFW, a process-oriented methodology called POST-DS (Costa & Aparicio, 2020) was employed. This methodology is not only supported in the processes but also in the organization, scheduling, and tools selection, which are essential components in a data science project to properly align expectations and clarify the project scope, costs, and time.

3.1. Obtaining and preparing the datasets

The initial dataset consists of 503 stocks that currently belong to the SP500 and are distributed across 11 sectors and 127 industries. Although called the SP500, the index contains 503 stocks because it includes two share classes of stock from 3 of its component companies.

In order to obtain this dataset, it was necessary to do web scraping with *Python* to the *Wikipedia* page, which is regularly updated and which, in addition to including the list of stocks that are present on the SP500, also includes other information, such as *GICS Sector*, *GICs Sub-Industry*, *Date added* and *Founded*. By grouping this dataset from the *GICS Sector* column and then by counting the number of stocks and sub-industries in each of these sectors, it was possible to obtain Table 2, where we have the number of stocks and sub-industries per sector. This table will be used in section 3.2.2, in which an analysis will be made of the clusters to be created in section 3.2.1.

Table 2 - Stocks distribution by sectors and industries

Sector Name	Number of Stocks	Number of Sub-Industries
Communication Services	22	9
Consumer Discretionary	53	19
Consumer Staples	38	12
Energy	23	5
Financials	72	13
Health Care	64	10
Industrials	78	19
Information Technology	64	12
Materials	28	10
Real Estate	31	13
Utilities	30	5

From *Yahoo Finance*, daily data was collected for stocks that are present in the initial dataset, such as date, open price, close price, adjusted close, high, low, and volume. With the collection of these data, were created six files, each consisting of a data frame of size *(t x n)*, where *t* represents the time series of the data and *n* represents the number of stocks. The timespan of the dataset ranges from 01-01-2001 to 31-12-2023. Subsequently, in these six files, it was verified the existence of some stocks whose extracted data were not complete for the period considered. The *Nans* found in these stocks is due to the fact that, for example, in 2010, these stocks did not exist. Therefore, all the *Nans* found were converted to zeros because if a stock does not exist, then its value in that period is zero.

The daily returns (r) were calculated using the file containing data relating to the adjusted closing price (ac) based on the following formula:

$$r_{t+1} = \frac{ac_{t+1}}{ac_t} - 1 \qquad [3.1]$$

The adjusted closing price was used instead of the closing price to calculate returns, as the last one doesn't take dividend payments into account, which will reduce profitability while calculating the returns. Thus, while the closing price merely refers to the

cost of stocks at the end of the day, the adjusted closing price considers other factors like dividends, which allows to give investors a more current and accurate idea of the stock's price.

Since this MFW is intended to do monthly analyses, then the calculated daily returns will be converted to monthly returns. For this, daily returns belonging to the same month and year will be summed. With this conversion of daily returns to monthly returns, we now have a total of 276 observations for each stock, which will be used both in the creation of clusters (section 3.2) and for making predictions (section 3.3).

Let's consider stock A, whose monthly return we want to predict for the month $t+1$. This prediction can be made with univariate or multivariate data (time series). On the one hand, univariate data involves using the target to predict the target. For example, the returns of stock A (before the month $t+1$) can be used to predict the return of stock A in the month $t+1$. On the other hand, multivariate data involves using the target as well as another time series to predict the target. For example, predicting the return of stock A, in the month $t+1$, using the returns (before the month $t+1$) of stock A as well as the returns (before the month $t+1$) of stocks that have similar behavior with stock A. One reason why it is preferable to use multivariate time series is that each variable depends not only on its past values but also has some dependency on other variables. And this dependency is used for predicting future values. Furthermore, given that multivariate time series include more features than univariate time series, they inherently provide more information. As a result, utilizing a multivariate model is typically more advantageous for predicting the behavior of complex systems, such as stocks.

Thus, in this MFW, we will use multivariate data so that the prediction of the return of stock A in the month $t+1$ depends not only on its previous returns but also on the previous returns of stocks that are correlated with stock A. In this way, the LSTM that will be applied in section 3.3 will receive as input a matrix $(n \times w)$ with the structure shown in Figure 5 and will have as output a vector $(n \times 1)$ with the structure shown in that same figure, where $x_{i,t-w}$ is the return of a given variable i (corresponding to a stock) in month t-w, n is the number of variables/stocks and w is the window size, that is, the number of months from past used to predict the month $t+1$.

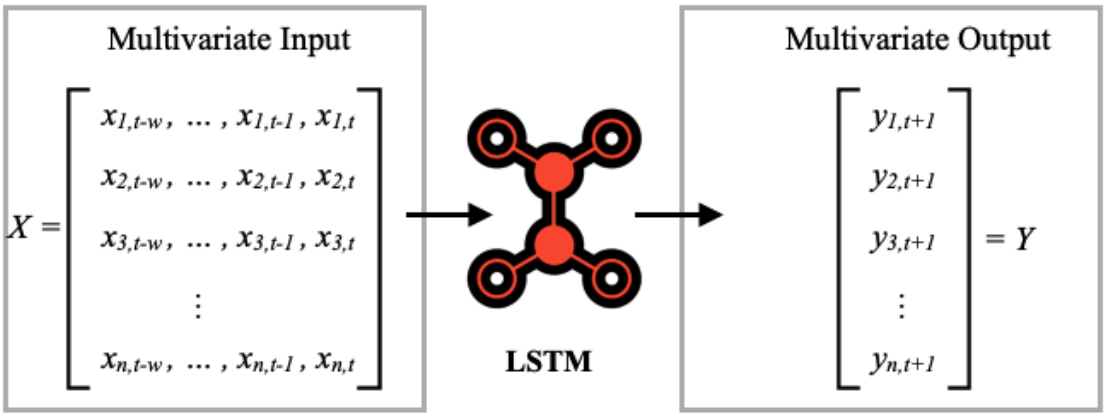

Figure 5 - Input and output dataset structure

3.2. Stocks clustering

3.2.1. Clusters creation

If we want to predict the return of stock A in the month *t+1* using a dataset composed of multivariate time series, it will be necessary to determine which stocks show similar behavior with stock A. For this, we will have to use a clustering algorithm that allows to create groups (clusters) of stocks that have similar behavior, in this case, on a monthly basis. Then, the stocks of the cluster to which stock A will belong will be used to predict the return of stock A in month *t+1*.

In order to create this dataset, the clustering algorithm that will be used is Hierarchical Clustering, which considers not only the distance between stocks but also the distance between stocks groups. In the application of this algorithm, like any other clustering algorithm, it is necessary to define a distance that allows to evaluate/measure the similarity intra-class and inter-class between the different stocks. To do this, a *Pearson's Correlation* matrix is first calculated between the different stocks, and then these correlations are transformed into distances, resulting in a matrix of distances. What is intended to happen is that:

- stocks that have a similar behavior have a high correlation and, consequently, a short distance;

- stocks that have a distinct behavior have a low correlation and, consequently, a high distance.

Let *M* be a matrix *(t x n)* of monthly stock returns, where *t* represents the time series of the data (from 01-01-2018 to 31-12-2022) and *n* represents the number of stocks. This recent period was selected to obtain groups of stocks that are currently correlated. From this *M* matrix, we will determine *Pearson's Correlations* between the different stocks for the period considered. In this way, it will be possible to obtain a new *N* matrix *(n x n)* which is composed of the *Pearson's Correlations* between the *n* stocks. The formula used to convert these correlations into distances is as follows:

$$d(x, y) = \sqrt{2 \cdot (1 - \rho_{xy})} \qquad [3.2]$$

Assume $X = \{x_1, x_2, \dots, x_n\}$ is a non-empty set, and let distance d be a function where for all $x, y, z \in X$. The distance formula 3.2 allows to satisfy the four distance principles:

(1)	$d(x, y) \geq 0$	non-negativity
(2)	$d(x, y) = d(x, y)$	symmetry
(3)	$d(x, y) = 0$, if $x = y$	identity
(4)	$d(x, y) \leq d(x, z) + d(y, z)$	triangle inequality

From this distance formula, we can verify that the distance matrix, obtained from the matrix of correlations, will have values between 0 and 2, because:

- when the correlation $\rho_{xy} = -1$, then:

$$d(x, y) = \sqrt{2 \cdot (1 - (-1))} = 2$$

- when the correlation $\rho_{xy} = 1$, then:

$$d(x, y) = \sqrt{2 \cdot (1 - 1)} = 0$$

In this way, we already have the matrices that we need to move on to the application of the Hierarchical Clustering algorithm: a matrix *(n x n)* constituted by the correlations between the *n* stocks and a matrix *(n × n)* constituted by the distances between the *n* stocks.

Initially, we can see that the stock groups are divided into small subsections in the correlation matrix, so the clustering structure is not evident.

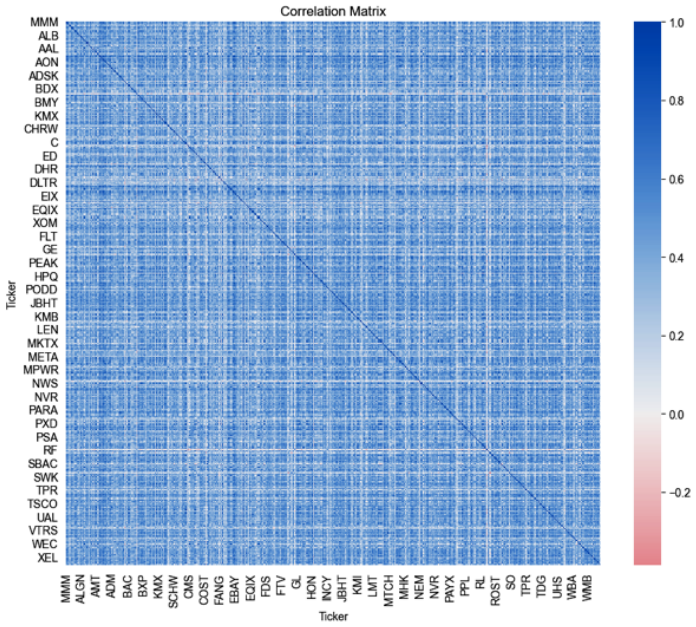

Figure 6 - Original correlation matrix

For the clustering structure to become more evident, we will use a method presented by Ledoit & Wolf (2004), called *Quasi-Diagonalisation*, in which it is possible to make a reorganization of stocks that allows to clearly show the inherent groupings. In other words, this method is a technique that reorganizes the covariance matrix so similar stocks will be placed together. This method uses a linkage matrix to visualize the resulting hierarchical clustering, which allows the distance between two clusters to be defined. In this case, we will use a single linkage matrix, in which the distances between two clusters is defined by a single element pair - those two elements which are closest to each other.

The dendrogram displayed in Figure 18, which is in the *Appendix*, shows how individual stocks and clusters of stocks merged based on their relative distance. In this

figure, we can already identify some clusters composed of stocks that are at short distances. This new stock organization shows that similar stocks stayed together and different stocks remained away. Thus, if we compare Figure 6 (related to the original matrix of correlations) with Figure 7 (related to the matrix with the reorganized stocks), we can see that in the last one, it is already possible to identify zones that correspond to clusters composed of highly correlated stocks, that is, stocks that have similar behavior in terms of monthly returns in the period considered.

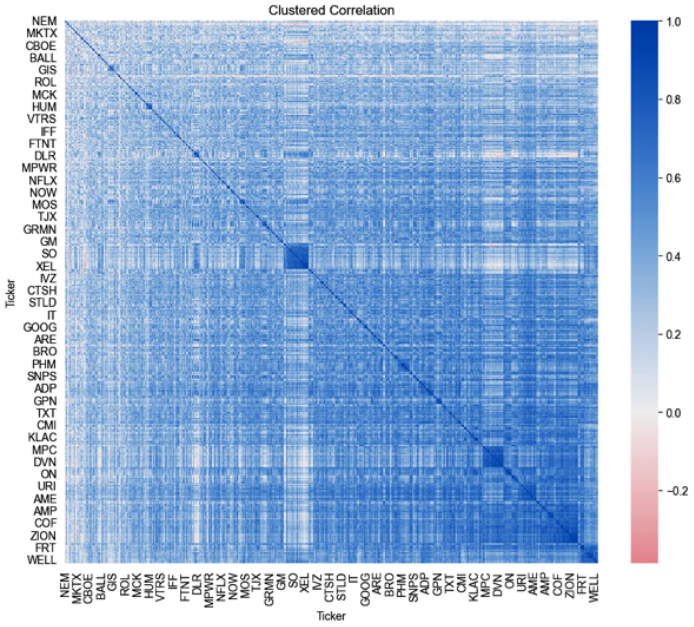

Figure 7 - Clustered correlation matrix

This reorganization of stocks resulted in the higher correlations being placed on the diagonal and the smaller correlations to be placed around that diagonal. So, if the goal was to create clusters composed of more diversified stocks, that is, less correlated stocks, then we would have to look for zones, around the diagonal, that had groups of uncorrelated stocks. In that case, we would be able to create several diversified portfolios, which could reduce risk and increase long-term return, because having a portfolio of stocks that are not correlated minimizes the possibility of a single stock having a high negative impact. On the other hand, if we have a portfolio composed of stocks that are correlated, then in the case of a fall in the price of one stock it is likely that the same will happen with the other stocks belonging to the same portfolio, causing a large loss.

However, building one or more diversified portfolios works only in long-term periods. In short-term periods, the return can vary widely. Therefore, diversification has the net impact of gradual and steady performance and smoother returns that never move too quickly up or down. Consequently, this reduced volatility reassures many investors.

Since this MFW is intended to do monthly analyses that are associated with short-term investments, then it makes no sense to create diversified portfolios. Furthermore, the creation of clusters consisting of stocks that are not correlated and, therefore, do not have similar behavior, would lead to the existence of a certain "noise" within the clusters (Vásquez Sáenz et al., 2023), which would possibly not be beneficial when applying deep learning algorithms for predicting monthly returns. Thus, if we create clusters consisting of stocks that have similar monthly behavior, we can decrease this possible "noise" that may exist within the clusters and, consequently, we can get better metrics and results in the application of the LSTM for predicting the monthly returns.

In this way, through Figure 7 we can identify on the diagonal around 3 clusters consisting of very correlated stocks, each cluster composed of 12 stocks, as we can see in Table 3.

Table 3 - Clusters of stocks obtained

Cluster 1	**Cluster 2**	**Cluster 3**
ETR	PSX	ZION
ES	HES	FITB
SO	CVX	CFG
ED	XOM	PRU
DUK	DVN	C
AEP	MRO	PNC
AEE	COP	USB
CMS	EOG	CMA
WEC	SLB	TFC
XEL	FANG	RF
DTE	PXD	JPM
LNT	HAL	BAC

This number of clusters was obtained considering two parameters:

- **P1**: minimum correlation (ρ_{min}) between stocks (the one chosen was 0.75);
- **P2**: minimum number of stocks that must be correlated to create a cluster (the one chosen was 12), considering that $\rho_{xy} \geq \rho_{min}$.

Thus, if we increase the values of the parameters *P1* and *P2,* we will decrease the number of clusters, and if we decrease the values of the parameters, we will increase the number of clusters created.

In Figures 19, 20, and 21, which are in the *Appendix*, we can visualize the matrix of correlations for each of these clusters. In fact, in these figures, it is possible to observe that there is a strong correlation between all stocks in each cluster, so we can say that the clusters created have stocks with similar behavior. We can consider outliers the stocks that remain outside these clusters, as they do not present high correlations with a significant number of stocks.

In this way, if the stock A, which was intended to be predicted at the beginning of this section, belongs to one of the clusters created, then it is already possible to predict its return in the month *t+1* through the dataset proposed in the previous section. In the case that stock A does not belong to any of the clusters created, then the prediction of its return cannot be made from a dataset of this type.

3.2.2. Clusters analysis

With the clusters created, in this section we can go on to the analysis of these clusters as well as the stocks that belong to them. In this analysis we will focus on the risks of clusters and stocks.

In fact, good risk management is crucial to making wise decisions. However, this management is associated with a certain complexity since there are several methods that can be used to identify and analyze the risks involved in investing in a given stock. One of the methods used to evaluate risk is through standard deviations, which are associated with the historical volatility of a given stock. What happens is that stocks with high standard deviations are very volatile, and stocks that have low standard deviations are a little

volatile. Therefore, highly volatile stocks are associated with a high risk, whereas low volatile stocks are associated to a lower risk.

Thus, to obtain the risks of clusters and stocks, the standard deviations of all stocks that were grouped were calculated. The risk of each stock corresponds to the average of the monthly standard deviations in the period under consideration, and the risk of every cluster is the mean of the risks of the stocks they hold. Through the following figure, we can visualize the risks of the stocks and clusters on the *y-axis*, where the risks of the clusters correspond to the y of their centroid. On the *x-axis*, we have the average monthly returns of stocks and clusters, which were calculated in the same way as the average monthly standard deviations.

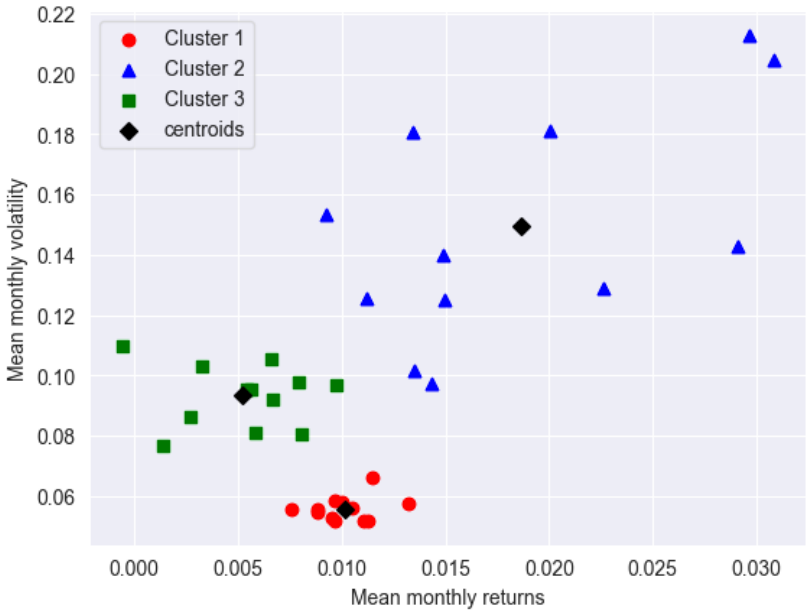

Figure 8 - Analysis of returns vs volatility

According to the variables under analysis in Figure 8, we can easily find differences between the clusters created in the previous section. We can observe that cluster 1 is the one that has the closest stocks. This cluster is the one that has stocks associated with a lower risk, and therefore, it is the cluster that has the lowest risk (5.59%). Furthermore, cluster 1 has a mean monthly return of 1.01%. Cluster 2, in turn, is the one that has the most distant stocks. In addition, this cluster has stocks that allow higher returns (1.87%), but are associated with higher risks as well (14.95%). Finally, cluster 3 is the one associ-

ated with lower returns (0.52%) and medium risks (9.34%). The values used in the analysis and comparison of *returns vs. volatility* between the clusters created are in the following table.

Table 4 - Centroids of the clusters

	Cluster 1	**Cluster 2**	**Cluster 3**
Mean monthly returns	0.010138	0.018652	0.005213
Mean monthly volatility	0.055870	0.149531	0.093362

If we compare the values of clusters 1 and 3 in Table 4, we can see that cluster 3 has a higher risk and lower return than cluster 1. Usually, higher risks are associated with higher returns, but in this case the opposite happens. In this way, for good risk management it is preferable to invest only in stocks that belong to clusters 1 and 2. Thus, in the next section we proceed with 2 clusters, where cluster 1 is associated with low- risk stocks and cluster 2 is related to high-risk stocks.

To conclude this section, it was also found that all the stocks in cluster 1 belong to the *Utilities* sector, all the stocks in cluster 2 belong to the *Energy* sector, and all the stocks in cluster 3 belong to the *Financial* sector. From Table 2 (in section 3.1) we can conclude that cluster 1 consists of 40% of the stocks belonging to the *Utilities* sector, cluster 2 is composed of around 52% of the stocks that belong to the *Energy* sector, and finally cluster 3 is made up of approximately 17% of stocks which belong to the *Financial* sector. We can also conclude that there is a strong correlation of monthly returns in stocks belonging to these sectors, so we can say that those sectors have a good clustering property. Indeed, the ideal situation for a trading strategy is that stocks in the same sector are strongly correlated while stocks in different sectors are weakly correlated or even independent. In this way, taking or removing positions in sector 1 will not influence the price of stocks that belong to sector 2.

3.3. Stocks prediction

In the prediction of a time series, stationarity is crucial as it makes it easier to extrapolate its future if its statistical properties remain constant over time. In Figure 9 is a sample of the time series (returns of stocks) that will be used to make predictions about stocks belonging to clusters 1 and 2. We can verify that these time series are stationary, as they present neither trend nor seasonality. Consequently, there is no need to apply transformations, such as Box-Cox transformations (Box & Cox, 1964), to make these time series stationary.

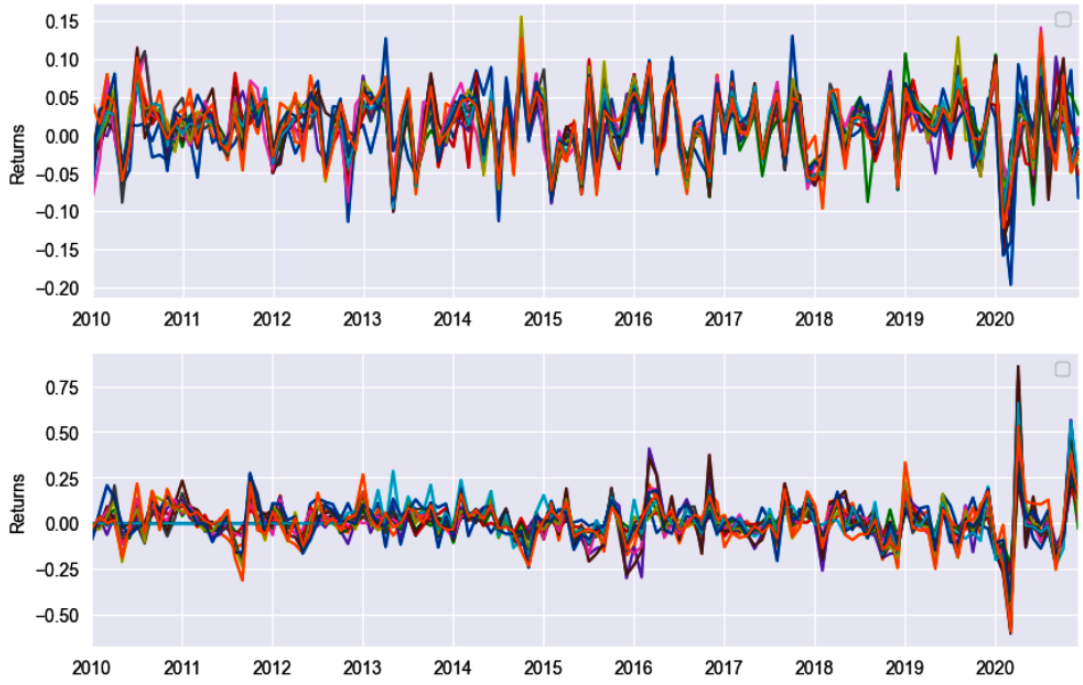

Figure 9 - Sample of the time series of stock returns for cluster 1 (above) and cluster 2 (below)

In these plots, besides the presence of stationarity, we can also observe that the time series corresponding to the stocks belonging to cluster 2 exhibits a greater variation in returns compared to the time series corresponding to the stocks belonging to cluster 1. This emphasizes the conclusions drawn in the previous section, where analyses of the obtained clusters were conducted. Additionally, we notice that the time series exhibit similar behaviors within the clusters to which they belong, allowing us to affirm that the clusters derived from the selected clustering algorithm are meaningful.

Another advantage of using these stock return time series, both as independent and dependent variables, is that the scale is always the same. Consequently, there is no need to normalize or standardize the data so that they fit within a specific range.

Let's suppose that stock A, which has been used as an example, belongs to one of the created clusters. The prediction of its return in month $t+1$ will be made using a dataset with the structure presented in section 3.1. Through this structure, the prediction of the return of stock A depends not only on its previous returns but also on the previous returns of stocks that are in the same cluster as stock A. This way, it will be possible to extract the temporal correlations within each stock and the spatial correlations among the different stocks belonging to the same cluster.

The dataset is split into two sets before the training: training and testing sets. The training set, typically 70-80% of the total data available, is used to build a model that learns from this data to identify past historical patterns. The test set, typically 20-30% of the data available, is used to provide an unbiased evaluation of the model on unseen data, i.e. on data that was not used to train the model. Next, the training set is split into k smaller subsets (folds) of approximately equal size, in this case, with $k = 6$, as shown in Figure 10. This procedure is known as cross-validation and enables the model to be trained using k-1 folds as training data while the remaining data (validation data) is used to validate and evaluate the model. This process is repeated k times, so the final performance of the model is measured through the averages of the individual errors across the k folds.

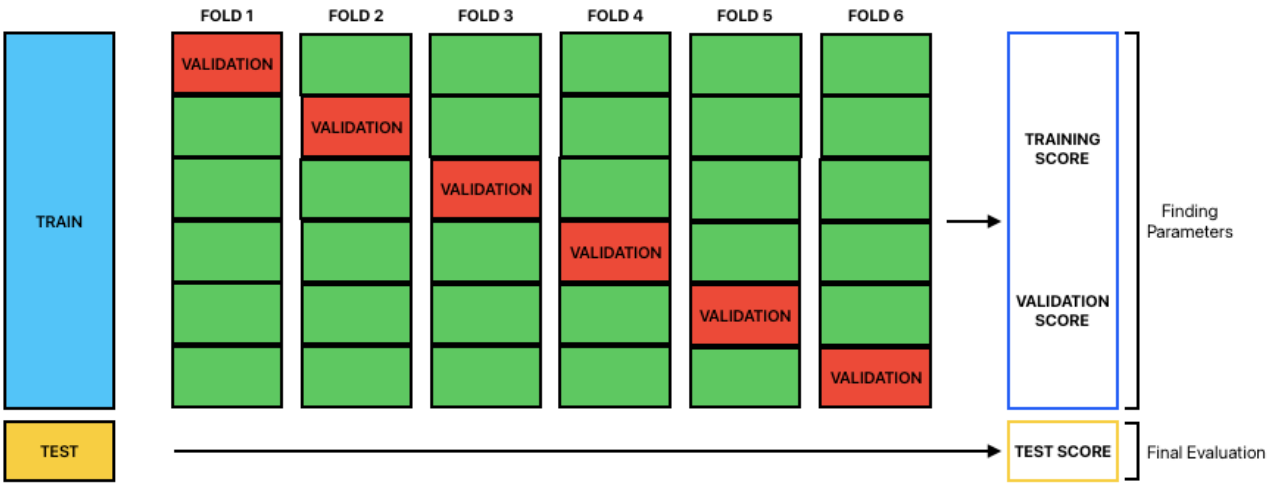

Figure 10 - Illustration of the cross-validation process employed

In the building of a neural network, various hyperparameters must be selected. These parameters encompass, for instance, the number of layers, hidden units, activation function, optimization function, learning rate, epochs, and window size. These parameters are fine-tuned through multiple experiments until the network achieves stability. Stability is reached when the training error plateaus, indicating that further adjustments yield minimal improvement. At this juncture, the network's weights are optimized, resulting in negligible changes in error. This iterative process is crucial in developing a neural network, as illustrated in Figure 11.

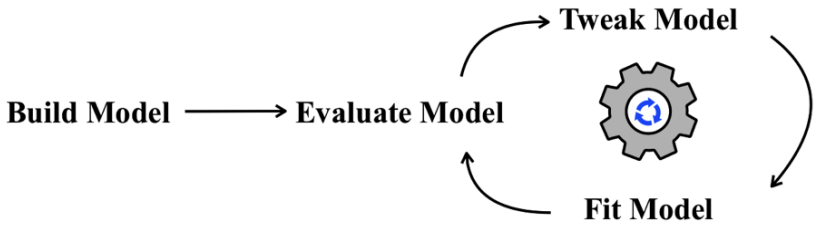

Figure 11 - Typical workflow when building a neural network

After these multiple experiments, the LSTM architecture developed for this study is illustrated in Figure 12. This structure is utilized to predict the monthly returns of stocks belonging to both cluster 1 and cluster 2. As shown, it includes five layers: an input layer, three nonlinear LSTM layers, and an output layer. The input layer consists of $N \times W$ neurons, where N represents the number of stocks and W represents the window size. The three LSTM layers are employed to capture the intricate temporal relationships within the multivariate time series data. Each LSTM layer employs dropout activation functions in their output layers (dropout = 0.1) to enhance learning by randomly deactivating a portion of output units during training, helping prevent overfitting. Finally, the output layer is a fully connected layer (or dense layer) responsible for predicting the future values of the stocks belonging to the cluster under analysis.

The number of hidden layers in the architecture and the number of neurons in each layer are determined through iterative experimentations. Beginning with one hidden layer, the model is trained, and its accuracy is evaluated. Subsequently, additional hidden layers are incrementally introduced, with accuracy evaluated at each stage. Following the inclusion of three hidden layers, no significant improvement in forecasting accuracy was

observed. Table 5 summarizes the final parameters chosen for building this neural network.

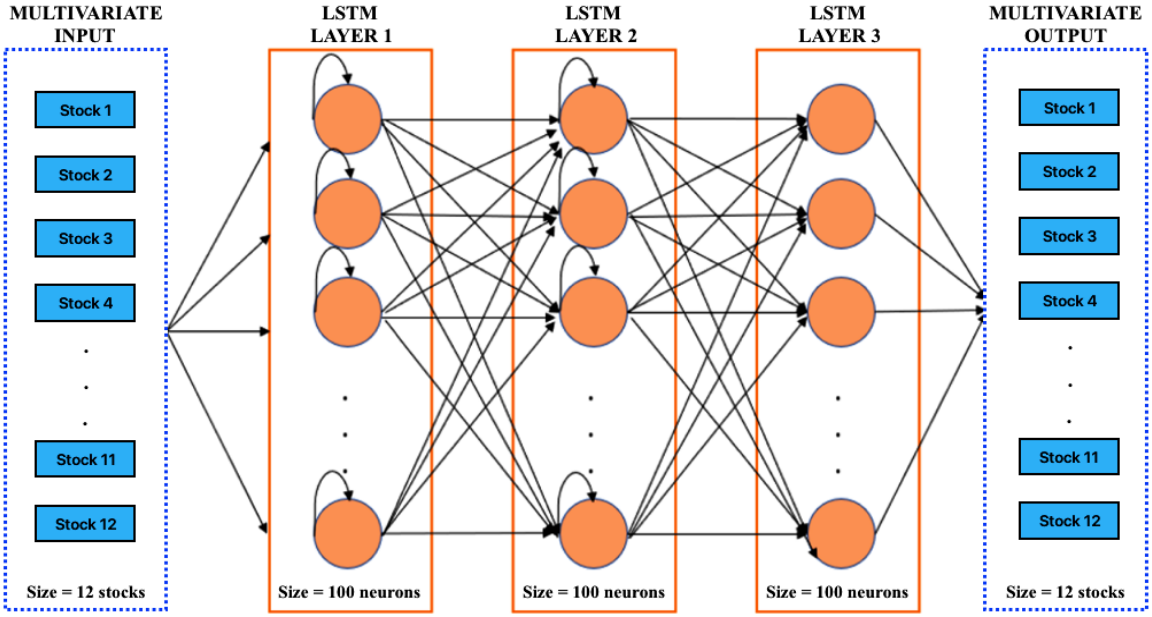

Figure 12 - LSTM architecture

Table 5 - Final parameters chosen

Parameter	**Value**
Number of epochs	100
Batch size	15
Early stopping	monitor = 'val_loss', restore_best_weights = True
Window size	12
Optimizer	RMSprop
Loss function	MSE
Activation function	tanh
Recurrent activation	sigmoid
Number of hidden layers	3
Number of neurons at each hidden layer	100
Learning rate	0.001

Chapter 4

4. RESULTS AND DISCUSSION

Three metrics will be utilized to evaluate the predictions obtained from the neural network developed in the previous chapter: MAE, RMSE, and MAPE. MAE measures the forecasting accuracy, providing insight into the deviation between predicted and measured values. RMSE computes the square root of the Mean Squared Error (MSE), which calculates the average squared deviation of predicted values, offering an overall assessment of the error magnitude. MAPE calculates the sum of individual absolute errors divided by the individual values separately, providing a measure of relative error. These metrics can be formulated as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \quad [4.1]$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \quad [4.2]$$

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad [4.3]$$

Where n represents the number of observations, y denotes the vector of actual values, and y represents the vector of predicted values.

As previously mentioned, through the utilization of cross-validation, with k = 6, the final performance of the model is measured through the averages of the individual errors, across the k folds, obtained in the predictions of the validation data. Thus, in Table 6 are the metric values obtained in each fold, along with their respective averages across the 6 folds.

Table 6 - Metric values obtained

Metrics	MAE		RMSE		MAPE	
Fold	Cluster 1	Cluster 2	Cluster 1	Cluster 2	Cluster 1	Cluster 2
1	0.0536	0.0550	0.0826	0.0755	99.17	7.46
2	0.0366	0.0650	0.0478	0.0883	9.88	22.24
3	0.0370	0.0762	0.0512	0.1083	5.25	23.36
4	0.0315	0.0589	0.0400	0.0775	4.60	25.87
5	0.0268	0.0730	0.0340	0.0982	17.95	22.79
6	0.0092	0.0633	0.0122	0.0828	0.5834	12.23
Average	0.0325	0.0653	0.0446	0.0884	22.91	18.99

In this table we can see that the calculated metrics are improving along the folds. In other words, the errors considered for analysis decrease until the last fold, indicating an improvement in the model's predictive accuracy as it progresses through the cross-validation process. In order to make it easier to interpret whether the predictions obtained are good and, consequently, if the calculated metrics are good, we will use the graphs of Figures 22, 23, 24 and 25, which are in the *Appendix*, in which we can visualize a comparison between the predicted values and the observed values of monthly returns in the test set. This test set consists of the returns from the last 24 months, the equivalent of the last two years.

In the predicted returns graphs (Figures 23 and 25), we can observe that the similarity in return movements among the stocks within each cluster is not as strong as the similarity observed in the graphs depicting the observed returns (Figures 22 and 24). Therefore, it can be inferred that the model faced challenges in capturing the spatial correlations among the different time series within each cluster. Furthermore, comparing the graphs of observed returns (Figures 22 and 24) with those of predicted returns (Figures 23 and 25), it is evident that there is not a significant similarity in return movements over the period considered in the test set. Consequently, it can be concluded that the model had difficulties in extracting the temporal correlations within each time series. Hence, the obtained metrics are not particularly appealing. Moreover, as mentioned in one of the articles referenced in section 2.2 of the literature review - Vásquez Sáenz et al. (2023) - where

one of the datasets used in LSTM has a structure similar to the dataset used in this MFW, it was concluded that the results are very far from a perfect forecast.

Nevertheless, let us verify whether the predictions obtained from this model enable us to achieve positive returns over the period considered in the test set. For this purpose, the monthly predicted returns (r) will be converted into adjusted closing prices (ac) using the following formula:

$$r_{t+1} = \frac{ac_{t+1}}{ac_t} - 1 \Leftrightarrow ac_{t+1} = (r_{t+1} + 1) \times ac_t \qquad [4.4]$$

Consequently, the buying and selling of stocks will be conducted by comparing the adjusted closing price in month *t+1*, derived from the predicted return, with the observed value of the adjusted closing price in month *t*. The rules for this buying and selling of stocks are as follows:

- If the predicted adjusted closing price (ac_{t+1}) is higher than the observed adjusted closing price (ac_t), then we have a buy signal (Figure 13);

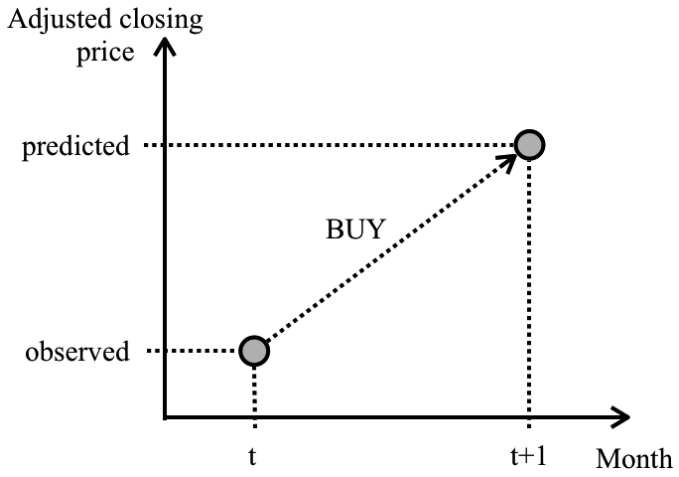

Figure 13 - Illustration of the rule to buy

- If the predicted adjusted closing price (ac_{t+1}) is lower than the observed adjusted closing price (ac_t), then we have a sell signal (Figure 14).

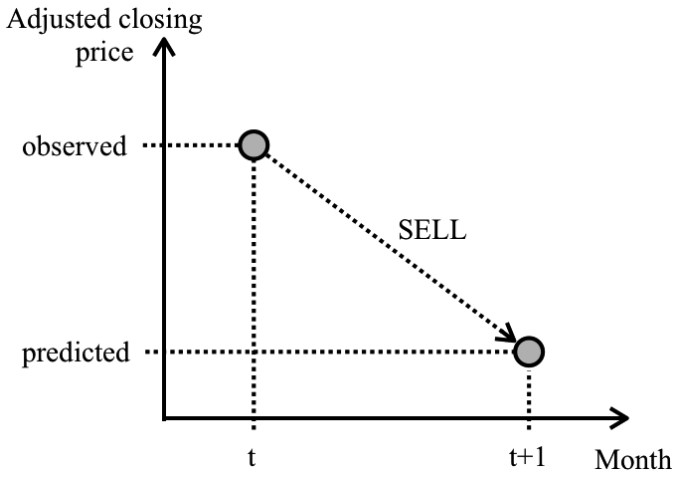

Figure 14 - Illustration of the rule to sell

This method allowed for the creation of buy and sell signals for stocks based on the obtained predictions. In a way, these signals result from the conversion of continuous values - the predicted returns by the model - into discrete values - buy or sell. It is important to note that no portfolio optimization was performed in this MFW, so all operations conducted will have an equally weighted portfolio. Thus, in Figure 15, we can verify whether the predictions obtained by the model created allow to obtain profits in the period considered in the test set. In this figure, the graph on the left shows the evolution of the returns accumulated in cluster 1, while the right graph illustrates the evolution of accumulated returns in cluster 2.

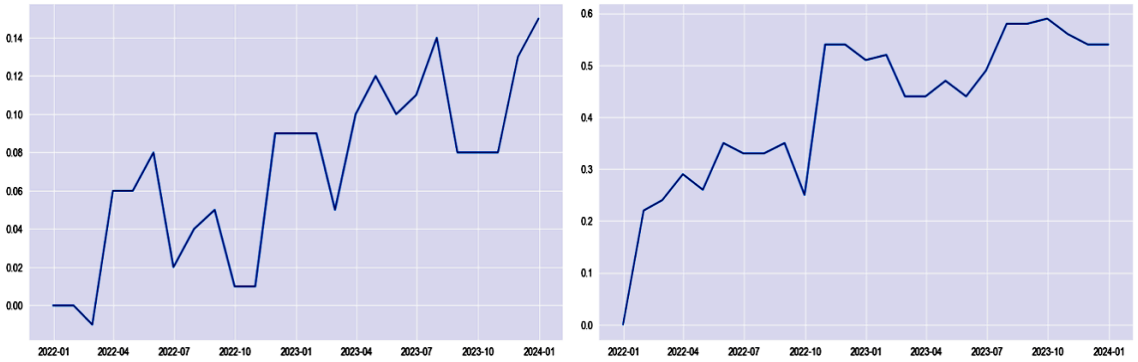

Figure 15 - Evolution of accumulated returns obtained in clusters 1 and 2

Through these graphs, we can observe the evolution of accumulated returns over the period considered in the test set. At the end of the analysis period, corresponding to

the end of the year 2023, it is evident that the final accumulated return is positive in both cluster 1 and cluster 2. More specifically, after two years, a cumulative return of 15% was achieved in cluster 1, while a cumulative return of 54% was obtained in cluster 2. The fact that a higher profit was obtained by investing in stocks from cluster 2 compared to stocks from cluster 1 aligns with the analysis conducted in section 3.2.2, where it was found that stocks belonging to cluster 2 are associated with higher returns but also higher risks.

In order to try to improve these returns, a strategy will be applied and tested that explores the possibility that if the majority of predictions in a cluster of correlated stocks indicate an upward movement, then all stocks belonging to that cluster are expected to have an upward movement. As an example and considering that each created cluster comprises 12 stocks, if the predictions for 8 stocks in a cluster k indicate an upward movement, and the remaining 4 predictions indicate a downward movement, then a buy signal will be triggered for all stocks belonging to cluster k. The same principle applies if the majority of predictions indicate a downward movement.

To implement this strategy, a count of the number of buy and sell signals generated each month will be conducted. If, in a cluster, the number of sell signals for a given month exceeds the number of buy signals, then all stocks belonging to that cluster will be sold in that month. Similarly, if the number of buy signals for a given month exceeds the number of sell signals in a cluster, then all stocks belonging to that cluster will be bought in that month. Thus, in Figure 16, we can observe the evolution of accumulated returns obtained through this strategy over the period considered in the test set. In this figure, the graph on the left shows the evolution of the returns accumulated in cluster 1, while the right graph illustrates the evolution of accumulated returns in cluster 2.

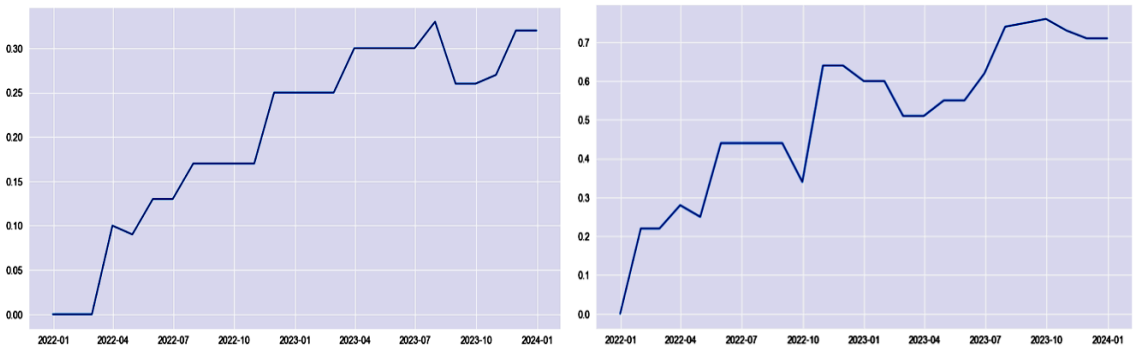

Figure 16 - New evolution of accumulated returns obtained in clusters 1 and 2

Through these graphs, we can observe that this strategy is also profitable, as it enables attaining a positive accumulated return at the end of the analysis period. Furthermore, if we compare the profits obtained through this strategy with the profits observed in Figure 15, we can conclude that the profits obtained at the end of the analysis period through this strategy are greater in both cluster 1 and cluster 2. More specifically, after two years, a cumulative return of 32% was achieved in cluster 1 - representing a 17% increase in profit compared to the previous return in cluster 1 - while a cumulative return of 70% was obtained in cluster 2 - corresponding to a 16% increase in profit compared to the previous returns in cluster 2.

In this way, we can conclude that this strategy is beneficial because it allows an increase the returns obtained initially. This is attributed to the fact that stocks belonging to the same cluster almost always exhibit similar behavior. Consequently, it is expected to be advantageous to buy all stocks of a given cluster if most predictions indicate a buy signal and to sell all stocks otherwise.

To conclude this section, it is important to answer the research question of this MFW: "Is it beneficial to invest in correlated stocks in the short-term?" On one hand, the literature review conducted it was found to be beneficial from the perspective of the forecasting ability of neural networks, as using data from uncorrelated stocks adds noise and deteriorates the forecasting ability of LSTM models. On the other hand, it was observed that stocks belonging to the same cluster almost always exhibit similar behavior, leading to an increase in returns obtained through the strategy used. Therefore, considering the returns obtained from the methodology employed in this MFW, it can be concluded that it is beneficial to invest in correlated stocks in the short-term.

Chapter 5

5. CONCLUSION

In summary, this project aimed to apply machine learning models to develop an effective strategy for achieving profits in the stock market, particularly in the stocks of the SP500 index. The methodology used is summarized in Figure 17.

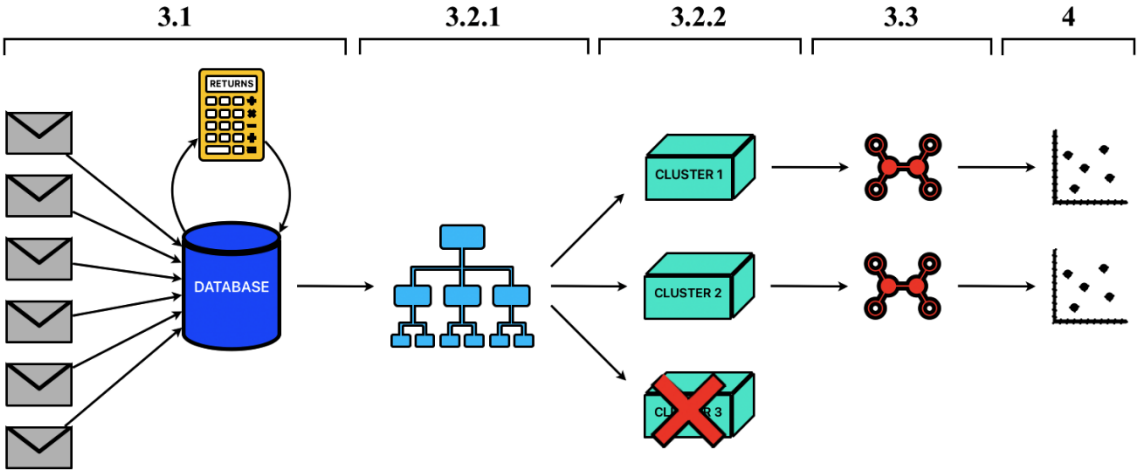

Figure 17 - Illustration of the methodology employed

Firstly, data were extracted from Yahoo Finance, and monthly returns were calculated based on adjusted close prices. Next, clusters composed of stocks exhibiting similar monthly behavior were created using the Hierarchical Clustering algorithm. An analysis of the created clusters revealed that stocks belonging to cluster 3 have higher risks and lower returns compared to stocks belonging to cluster 1. Consequently, for good risk management, it was decided to discard investment in stocks belonging to cluster 3. Therefore, we proceeded with 2 clusters, where cluster 1 was associated with low-risk stocks and cluster 2 was related to high-risk stocks. Next, in section 3.3, monthly returns of the stocks belonging to these clusters were predicted using an LSTM. In the LSTM architecture building, several hyperparameters had to be chosen. This selection was made through

multiple experiments. Finally, an analysis of the predictions that were obtained was conducted. In this analysis, it was found that the created model struggled to capture both temporal and spatial correlations. Nevertheless, both clusters achieved profits after converting the predicted continuous values into discrete values - representing monthly buy and sell signals for stocks. To conclude, a strategy was tested to mitigate the potential impact of incorrect predictions. This strategy involved buying all stocks of a given cluster if most predictions indicated a buy signal and selling all stocks otherwise. Analyzing the results obtained from this strategy, it was found that it increased profits in both clusters. This underscores the fact that stocks belonging to the same cluster almost always exhibit similar behavior. Therefore, it was possible to increase profits in both clusters. As a response to the research question posed in this MFW, we can conclude that it can be beneficial to invest in correlated stocks in the short-term.

While long-term investment may benefit from diversifying the portfolio with uncorrelated stocks, short-term investment, as we have seen, may benefit from diversifying the portfolio with correlated stocks with the aid of machine learning algorithms. In fact, if we aim for long-term investment, then in my opinion, investing in uncorrelated stocks is advisable. On the other hand, if we intend to invest in the short-term, then it makes more sense to invest in correlated stocks that exhibit similar behavior. However, this short-term investment is associated with higher returns but also with higher potential losses. Thus, these two investment alternatives depend on the investor's profile, that is, whether the investor is willing to make short-term investments associated with higher risk or prefers long-term investments associated with lower risk.

Thus, the strategy developed in this MFW has potential for investors who prefer to take higher risks in their investments, aiming to potentially achieve greater returns in the short-term. In the future, it is important to update the created clusters regularly, as the goal is to have clusters with stocks that are currently correlated rather than clusters with stocks that were correlated in the past but are no longer. Additionally, the LSTM model should be run monthly with updated data to obtain predictions for the next month.

In terms of aspects for improvement to be developed in the future, there are several avenues to explore. As demonstrated in the literature review, there are currently various strategies in which different machine learning algorithms are used to predict stock

market behavior. The strategy developed in this MFW was inspired by these reviewed articles, but exploring other articles may reveal additional interesting ideas.

There are several datasets that were exported from Yahoo Finance, in section 3.1, but were not used in this project. Perhaps the use of this data could provide more relevant information for the model to learn existing patterns in the data and, consequently, to make better predictions. There are also other types of data with potential, such as tweets and financial reports, which allow for sentiment analysis. This sentiment analysis is associated with a more fundamental perspective that would also be interesting to explore.

In addition to experimenting with other data, it would also be important to experiment with other hyperparameters and to evaluate in more detail the impact of each of them on the metrics obtained from the predictions made by the model. Despite the profit obtained in the previous chapter, it was concluded that the metrics obtained were not the best. Therefore, it is essential to re-evaluate the structure of the neural network and identify where adjustments could be made to better capture the existing temporal and spatial correlations in the data used and consequently improve the metrics obtained.

It would also be interesting to analyze the results obtained from predictions for another time-period. For example, instead of monthly predictions, make weekly or daily predictions. On the one hand, it would be possible to increase the number of observations for the algorithm to train on, which could result in a better model. On the other hand, it would be necessary to update the data and run the LSTM weekly or daily. Since this project focused on monthly analyses/predictions, it is only necessary to have this data update and to run the LSTM on the last day of each month.

Lastly, for effective risk management, it would be interesting to optimize the portfolio to achieve a good risk-return balance. In this trade-off, ratios that calculate a measure of return per unit of risk are very popular: sharpe ratio and information ratio. With the help of these ratios, it would be possible to effectively optimize the portfolio by selecting weights for a given set of stocks to minimize risk, measured as the standard deviation of returns for a given expected return. In the end, the goal is to maximize the return and minimize the standard deviation (volatility/risk).

REFERENCES

Amini, S., Hudson, R., Urquhart, A., & Wang, J. (2021). Nonlinearity everywhere: implications for empirical finance, technical analysis and value at risk. *The European Journal of Finance*, *27*, 1326 – 1349. https://doi.org/10.1080/1351847X.2021.1900888

Aparicio, J. T., Romao, M., & Costa, C. J. (2022). Predicting Bitcoin prices: The effect of interest rate, search on the internet, and energy prices. In 2022 17th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1-5). IEEE. https://doi.org/10.23919/CISTI54924.2022.9820085

Bandara, K., Bergmeir, C., & Smyl, S. (2020). Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert Systems with Applications*, *140*, 112896. https://doi.org/10.1016/j.eswa.2019.112896

Bachelier, L. (1900). Theory of Speculation: The Origins of Modern Finance. Princeton University Press.

Box, G. E., & Cox, D. R. (1964). An analysis of transformations. Journal of the Royal Statistical Society: Series B (Methodological), 26(2), 211 - 243.

Costa, C. J., & Aparicio, J. T. (2020). POST-DS: A Methodology to Boost Data Science. 2020 15th Iberian Conference on Information Systems and Technologies (CISTI). https://doi.org/10.23919/CISTI49556.2020.9140932

Hadavandi, E., Shavandi, H., & Ghanbari, A. (2010). Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting. *Knowledge-Based Systems*, *23*(8), 800 - 808. https://doi.org/10.1016/j.knosys.2010.05.004

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *06*(02), 107 - 116. https://doi.org/10.1142/S0218488598000094

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735 - 1780. https://doi.org/10.1162/neco.1997.9.8.1735

Ledoit, O., & Wolf, M. (2004). Honey, I shrunk the sample covariance matrix. *The Journal of Portfolio Management*, *30*(4), 110 – 119.

Long, J., Chen, Z., He, W., Wu, T., & Ren, J. (2020). An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in Chinese Stock Exchange Market. *Applied Soft Computing*, *91*, 106205. https://doi.org/10.1016/j.asoc.2020.106205

Lu, Y. (2018). *Application of clustering methods to trading strategies in the US equity market* (Master's thesis, Imperial College London, United Kingdom). Department of Mathematics, Imperial College London.

Oliveira, A. D., Pinto, P. F., & Colcher, S. (2020). Stocks clustering based on textual embeddings for price forecasting. *Intelligent Systems*, 665 - 678. https://doi.org/10.1007/978-3-030-61380-8_45

Vásquez Sáenz, J., Quiroga, F. M., & Bariviera, A. F. (2023). Data vs. information: Using clustering techniques to enhance stock returns forecasting. *International Review of Financial Analysis*, *88*, 102657. https://doi.org/10.1016/j.irfa.2023.102657

Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. Annals of data science, 2, 165-193.

APPENDIX

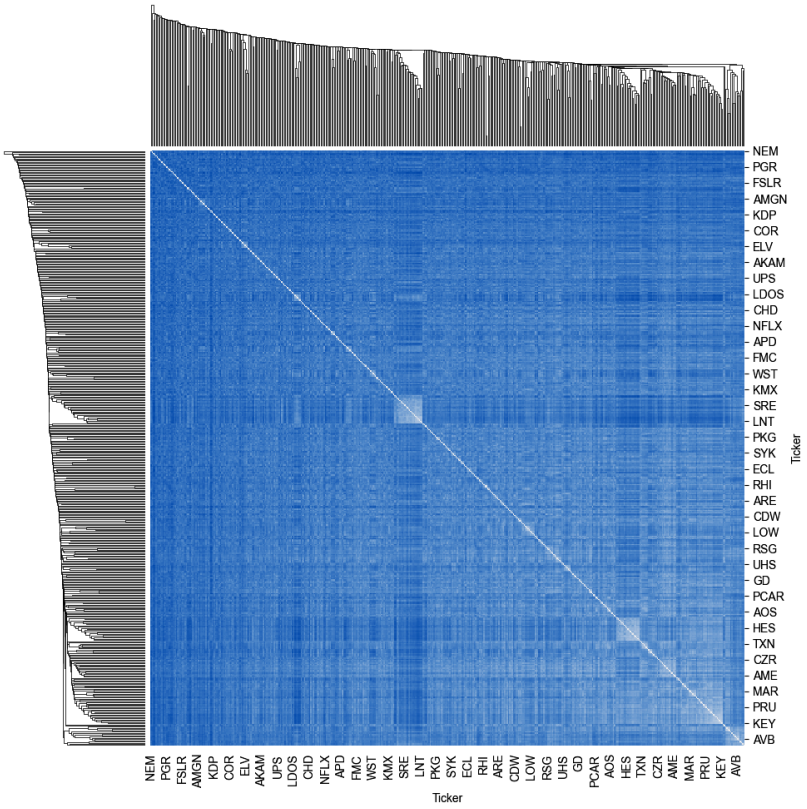

Figure 18 - Dendrogram illustrating the merging of individual stocks based on their relative distance

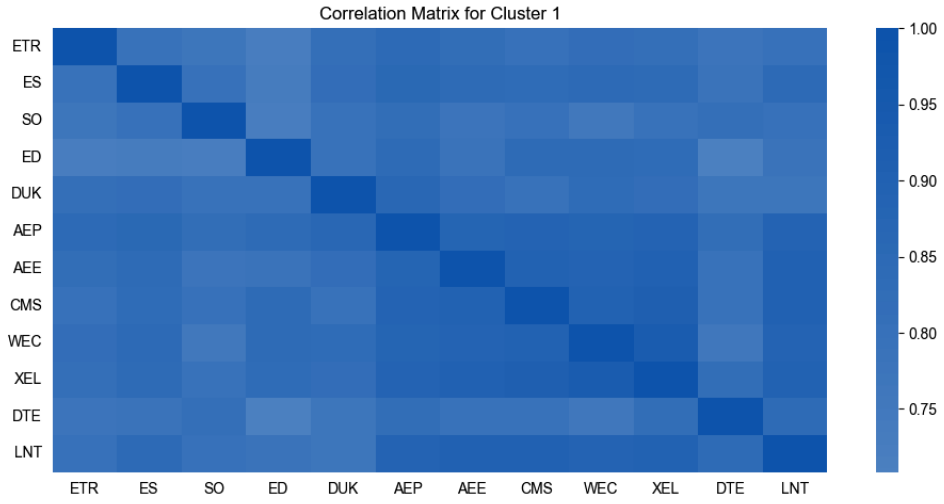

Figure 19 - Correlation matrix for cluster 1

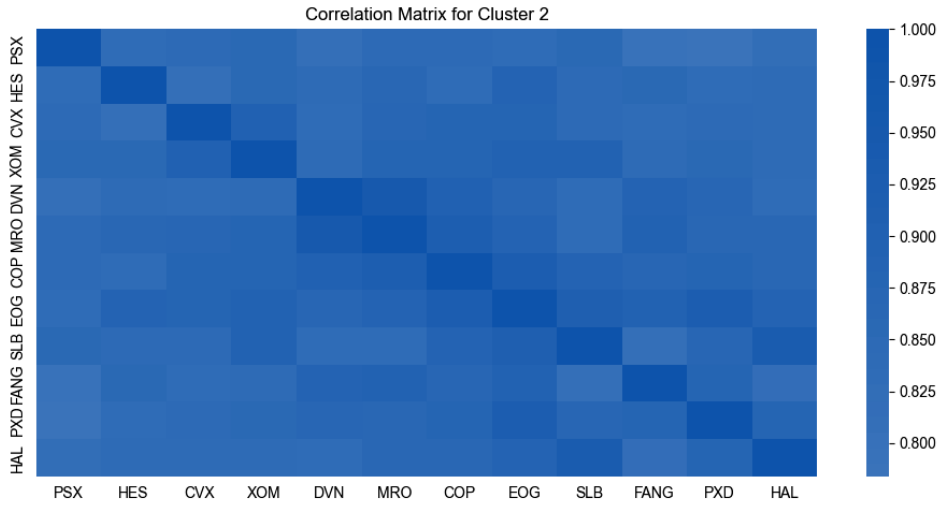

Figure 20 - Correlation matrix for cluster 2

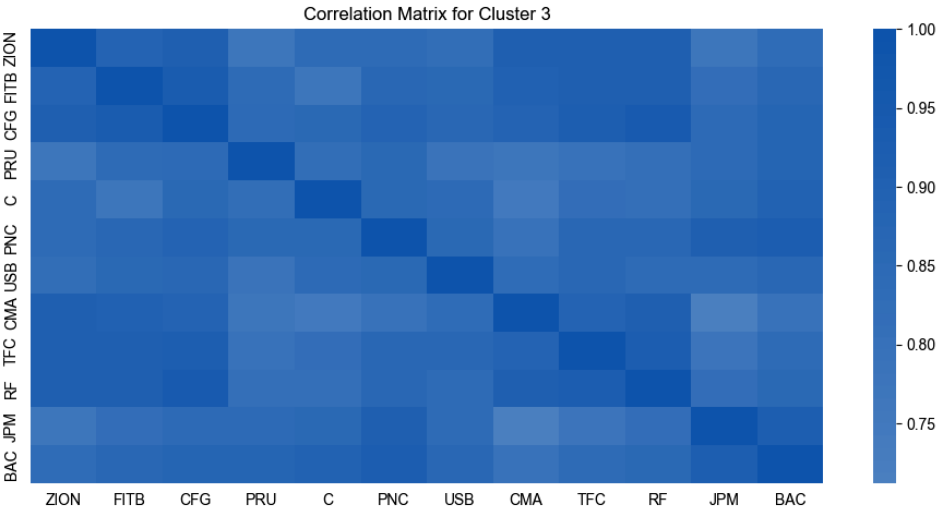

Figure 21 - Correlation matrix for cluster 3

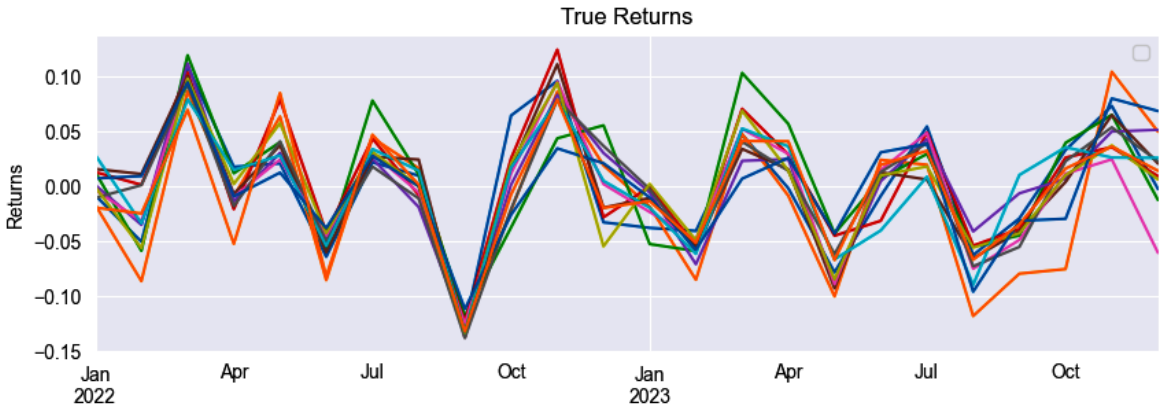

Figure 22 - Observed returns in the test set for cluster 1

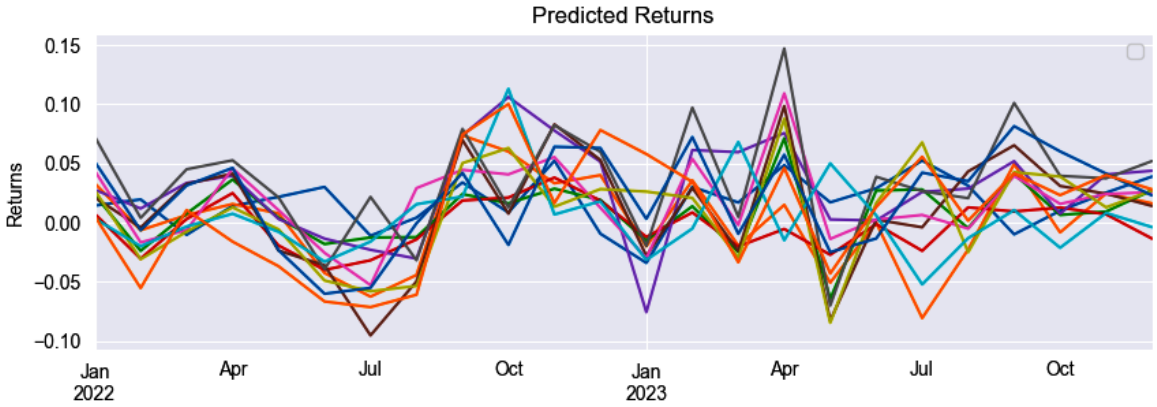

Figure 23 - Predicted returns in the test set for cluster 1

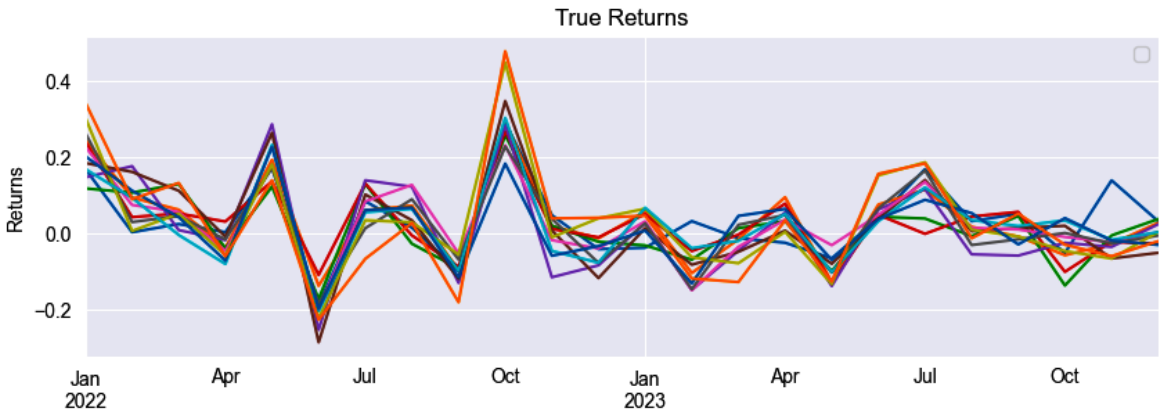

Figure 24 - Observed returns in the test set for cluster 2

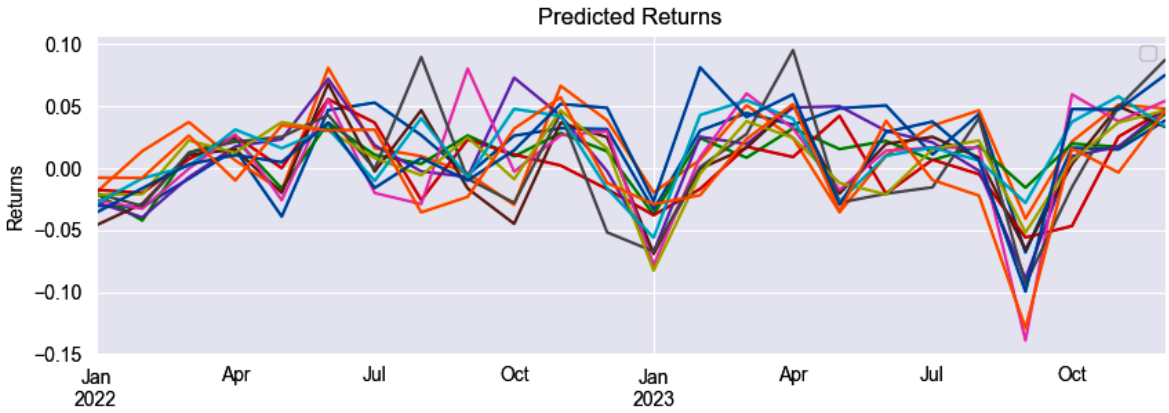

Figure 25 - Predicted returns in the test set for cluster 2