



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

School of Informatics

Predicting stock prices using long short-term memory
models

End of Degree Project

Bachelor's Degree in Informatics Engineering

AUTHOR: Navarro Hawach, Anuar

Tutor: Castro Bleda, María José

ACADEMIC YEAR: 2021/2022



Abstract

Quantitative finance focuses on the mathematical models used to price securities and measure risk. In recent years, it has become a challenging aspect in today's world of valuable and better investment. This bachelor thesis makes use of long short-term memory models using historical prices to predict stock prices from three different stock indexes in order to use them in conjunction with the Markowitz model to create efficient portfolios. Various metrics have been considered to compare the performance of portfolios created using stock prices predicted and the Markowitz model with those that only used the Markowitz model and the performance of exchange-traded funds that mirror each stock index.

Keywords: prediction model; Long Short-Term Memory (LSTM); stock price prediction; neural networks; Markowitz

Resumen

Las finanzas cuantitativas se centran en los modelos matemáticos utilizados para fijar el precio de los valores y medir el riesgo. En los últimos años, se han convertido en una parte crucial a la hora de escoger mejores inversiones. Este trabajo de fin de grado hace uso de redes de gran memoria de corto plazo utilizando precios históricos para predecir los precios de las acciones de tres índices bursátiles diferentes con el fin de utilizarlos junto con el modelo de Markowitz para crear carteras eficientes. Se han considerado varias métricas para comparar el rendimiento de las carteras creadas utilizando los precios de las acciones predichas y el modelo de Markowitz con las que sólo utilizaron el modelo de Markowitz y también, con fondos indexados que reflejan cada índice bursátil.

Palabras clave: modelo de predicción; Long Short-Term Memory (LSTM); predicción de valores bursátiles; redes neuronales; modelo de Markowitz

Resum

Les finances quantitatives se centren en els models matemàtics utilitzats per a fixar el preu dels valors i mesurar el risc. En els últims anys, s'han convertit en una part crucial a l'hora de triar millors inversions. Este treball de fi de grau fa ús de xarxes de gran memòria de curt termini utilitzant preus històrics per a predir els preus de les accions de tres índexs borsaris diferents a fi d'utilitzar-los junt amb el model de Markowitz per a crear carteres eficients. S'han considerat diverses mètriques per a comparar el rendiment de les carteres creades utilitzant els preus de les accions predites i el model de Markowitz amb les que només van utilitzar el model de Markowitz i també, amb fons indexats que reflectixen cada índex borsari.

Paraules clau: model de predicció; Long Short-Term Memory (LSTM); predicció de valors borsaris; xarxes neuronals; model de Markowitz

List of Contents

List of Figures	VIII
List of Tables	X
List of Abbreviations	XI
1. Introduction	12
1.1 Objectives.....	13
1.2 Motivation.....	14
1.3 Outline of the thesis.....	15
2. State of the Art	16
2.1 History of the use of machine learning in Finance and Economics.....	16
2.2 Evolution of techniques applied in the stock market.....	19
2.2.1 Instance-based methods.....	19
2.2.2 Tree-based methods.....	20
2.2.3 Clustering methods.....	21
2.2.4 Neural networks.....	21
2.3 Long short-term memory models and their potential.....	22
3. Methodology	24
3.1 Introduction.....	24
3.2 Portfolio Optimization Model.....	24
3.3 Prices prediction.....	29
3.3.1 Step-by-step LSTM walkthrough.....	32
3.3.2 Error measurements for predicted prices.....	34
3.4 Results comparison criteria.....	35
4. Solution Development	39
4.1 Introduction.....	39
4.2 High-level solution.....	39
4.3 Data.....	41



4.4	Generate optimal portfolio	43
4.5	Price prediction models	45
4.5.1	Toolkit	47
5.	Analysis	48
5.1	Portfolio performance analysis	48
5.1.1	S&P	50
5.1.2	DAX	57
5.1.3	DJI	62
5.1.4	Cross evaluation	68
5.2	Stock price prediction models analysis	69
5.2.1	Error	70
5.2.2	Time	73
5.3	Portfolios against index exchange-traded funds	74
5.3.1	S&P	75
5.3.2	DAX	77
5.3.3	DJI	78
5.3.4	Cross evaluation	80
6.	Conclusion	82
	Appendices	84
A.1	Sustainable Development Goals	84
	Bibliography	86

List of Figures

Figure 1: Efficient Frontier (Scherer, 2002).....	26
Figure 2: Repeating module in a standard RNN containing a single layer (Olah, 2016)	30
Figure 3: Repeating module in an LSTM contains four interacting layers (Olah, 2016)	30
Figure 4: LSTM module (Olah, 2016)	31
Figure 5: Sigmoid layer (Olah, 2016)	31
Figure 6: “Forget gate layer”, a sigmoid layer (Olah, 2016)	32
Figure 7: Input gate layer (Olah, 2016)	32
Figure 8: Update cell states in LSTM module (Olah, 2016)	33
Figure 9: Output sigmoid layer (Olah, 2016).....	33
Figure 10: Efficient Frontier DAX30	44
Figure 11: All models and how they are split	49
Figure 12: Annualized return of all models with S&P stocks	50
Figure 13: Risk of all models with S&P stocks.....	51
Figure 14: Total portfolio value of all models with S&P stocks.....	54
Figure 15: Cost of all models with S&P stocks	55
Figure 16: Profit of all models with S&P stocks	56
Figure 17: Annualized return of all models with DAX stocks.....	57
Figure 18: Risk of all models with DAX stocks	58
Figure 19: Total portfolio value of all models with DAX stocks.....	60
Figure 20: Cost of all models with DAX stocks	61
Figure 21: Profit of all models with DAX stocks	62
Figure 22: Annualized return of all models with DJI stocks	63
Figure 23: Risk of all models with DJI stocks	64
Figure 24: Total portfolio value of all models with DJI stocks.....	66
Figure 25: Cost of all models with DJI stocks	67
Figure 26: Profit of all models with DJI stocks	68
Figure 27: MAE of all models	70
Figure 28: MAPE of all models.....	71
Figure 29: RSME of all models	72
Figure 30: Time taken to train and execute their predictions	73



Figure 31: Annualized return achieved by models using S&P stocks and the S&P index during 2007-2020 75

Figure 32: Risk achieved by models using S&P stocks and the S&P index during 2007-2020 76

Figure 33: Annualized return achieved by models using DAX stocks and the DAX index during 2007-2020 77

Figure 34: Risk achieved by models using DAX stocks and the DAX index during 2007-2020 78

Figure 35: Annualized return achieved by models using DJI stocks and the DJI index during 2007-2020 79

Figure 36: Risk achieved by models using DAX stocks and the DAX index during 2007-2020 80



List of Tables

Table 1: Stock prices list form Source	42
Table 2: Optimal portfolios form	44
Table 3: Annualized return and risk of all models with S&P stocks	53
Table 4: Annualized return and risk of all models with DAX stocks	59
Table 5: Annualized return and risk of all models with DJI stocks	65

List of Abbreviations

API	Application Programming Interface
ARIMA	Autoregressive Integrated Moving Average
CHAID	Chi-square Automatic Interaction Detection
DAX	Deutscher Aktien Index
DJI	Dow Jones Industrial Average
ETF	Exchange-Traded Fund
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network

1. Introduction

Business intelligence solutions are frequently used by smart investors and traders to gather, process, analyze, and report business data for informed business decisions. Modern business intelligence solutions are built on machine learning, which has helped traders and investors make trustworthy, risk-aware judgments. Investors utilize business intelligence systems capable of identifying long-term variations in the retrospective data to reduce investment risk caused by stock market uncertainty. In particular, the financial time series can be accurately predicted using innovative Long Short-Term Memory (LSTM) technology.

This bachelor thesis consists in the creation of an optimized stock portfolio using stock prices prediction in an attempt to outperform the original Markowitz model (Markowitz, 1952) and a passive management approach. To do so, the Markowitz modern portfolio theory has been used to create optimal portfolios and, it has been enhanced with new information, the predicted prices of each stock.

Stocks from three different indexes, Deutscher Aktien Index (Chen J. , 2022), Standard and Poor's (The Editors of Encyclopedia Britannica, 2022) and Dow Jones Industrial Average (Ganti, 2022) have been used to create optimal portfolios. For each index, a different portfolio has been created for each trimester, quarter, semester and year starting from 2007 until 2020 included. Also, for each timeframe and index, two different portfolios have been created, one using the original Markowitz model and, another one using the Markowitz model with the predicted prices of each stock for the next timeframe. In order to predict stock prices, a long short-term memory recurring neural network (Hochreiter & Schmidhuber, 1997) has been used.

This approach enables the possibility of comparing returns between the Markowitz model with and without stock predicted prices, as well as the comparison with other exchange-traded funds (ETFs) that mirror the established index, in order to know if using this strategy is better, or not, than a classic approach (using a simple Markowitz model without including predicted prices) or a passive management approach. This comparison has been done after running a simulation where an “investor” buys and sells all stocks

each timeframe (depending on the portfolios being used, trimester, quarter, semester or year) all through 2007-2020 to adhere to the changes of each portfolio, in order to calculate the return, after taking into account all costs and profits that carry the buying and selling of stocks.

1.1 Objectives

In an attempt to achieve the objective of this thesis, create an optimized portfolio of stocks using stock value prediction in an effort to outperform the original Markowitz model and a passive management approach, the following steps have been followed:

1. Three stock indexes have been used: Deutscher Aktien Index (Chen J. , 2022), Standard and Poor's (The Editors of Encyclopedia Britannica, 2022) and Dow Jones Industrial Average (Ganti, 2022). The purpose of this is to get all stocks that have been part of these indexes since 2005 so, the window is big enough to make enough comparisons on the analysis stage (in terms of the amount of years of data, 2005-2020 and, in terms of number of stocks). Plus, using indexes will enable the possibility of comparing the results with an exchange-traded fund (ETF) that mimics the corresponding index, which will show if this solution is better than the average performance of each index.
2. Data extraction: Collect all past prices of all stocks to be analyzed. All companies that are part, as of 12/15/2021, of the S&P500, DJI and DAX30 and up to 2005. In addition, the price of 30-year US and German bonds for the same years have also been extracted, which have been used as the risk-free rates when applying the Sharpe ratio.
3. Data normalization and cleaning: After the data had been collected, the data has been normalized and cleaned, to ensure that there are no anomalies in the data, so that the models can be successfully trained.
4. Generate efficient stock portfolios: For each of the three indexes, efficient portfolios have been generated using the Markowitz model (Markowitz, 1952) for each trimester/quarter/semester/year between 2007-2020. Four different timeframes have been used to compare more models and discuss the best possible solution. Take into account the start year is 2007 as, as it will be discussed later, certain prediction models need more than one year of past stock

prices to predict the next prices. Also, a different portfolio will be created for every timeframe starting from 2007 until 2020. For instance, in the case of an annual timeframe, a portfolio will be created for the period 2007-2008, another for 2008-2009, and so on until the period 2019-2020.

5. Apply a neural network model (LSTM) to predict stock prices: Using the data of stock prices collected, the model predicts the stock price of each stock for the next trimester, quarter, semester and year. Different models have been used, for instance, using the last month, semester or year historical prices to predict the next trimester, quarter, semester and year prices. An example is, given the 2006 stock prices of NIKE, predict the prices for the next semester, from 01-01-2007 until 01-07-2007. All models have been described fully in the 'Price prediction models' section.
6. Generate efficient stock portfolios using the predicted prices: After predicting the stock prices, repeat step 4 but now, making use of the predicted prices to create the portfolios. A different portfolio has been generated for each model.
7. Compare results: Once the portfolios have been generated with and without the predicted prices, the performance of each portfolio has been compared. To do so, a simulation has been done for each portfolio, which emulates the cost of an investor of buying/selling the stocks each trimester/quarter/semester/year, depending on the model used to generate each portfolio, to take into account the changes there are with each portfolio between timeframes. This simulation is later explained in detail in the Analysis section.

1.2 Motivation

The purpose of this chapter is to showcase the motivational factors that pushed me to complete this bachelor's thesis.

- Explore neural network possibilities in the stock market. In recent years, quantitative finance has been gaining traction due to its positive results in the stock market, in many times outperforming classic qualitative approaches. Plus, the advancement in computer power has enabled the possibility of applying advanced artificial intelligence algorithm's which were previously impossible to do so in such volatile scenarios, where quickness is crucial.

- Where finance and computer science cross. My passion in finance and computer science originally pushed me to pursue the double degree in computer science and business administration and management. Now, I wanted to mix both to use the knowledge acquired during the double degree in a single project, taking advantage of the synergy.
- Learn about machine learning. Due to not being able to select the computation branch in my computer science degree, my machine learning knowledge was very limited. I wanted to learn more about machine learning, as it is currently a hot topic that is revolutionizing many sectors, if not all.

These points describe why the project was carried out and indicate the beginning of the project's objectives.

1.3 Outline of the thesis

This bachelor thesis is divided into five main chapters:

- Chapter 1 – Introduction: Introduces the thesis with a short summary, the motivation behind it as well as its objectives.
- Chapter 2 – State of the Art: Overview of the current machine learning techniques applied to carry out a quantitative analysis of the stock market.
- Chapter 3 – Methodology: Explains in detail all the methods used in order to complete this thesis and why they were chosen. From the portfolio optimization model to the method used to predict stock prices and the results comparison criteria.
- Chapter 4 – Solution Development: Describes the workflow followed to achieve the objective of the thesis. It first showcases a high-level solution, followed by a detailed explanation of each step. Data extraction, generate optimal portfolios and price prediction models.
- Chapter 5 – Analysis: Presents the metrics and results of the investigation.
- Chapter 6 – Conclusion: Showcases the conclusions of the thesis.

2. State of the Art

2.1 History of the use of machine learning in Finance and Economics

Modern machine learning techniques have been slow to take hold in finance and economics. Nonetheless, the academics and practitioners in these fields were critical in creating the foundation for what we now call machine learning. Mathematics has been used in social and economic studies since the seventeenth century. Then, primarily in German universities, a teaching style arose that focused on the precise presentation of data in relation to public management. In the mid-eighteenth century, Gottfried Achenwall gave a lecture in this style, coining the term statistics (Meitzen, 1891). We saw a burst of economic statistics by the 19th century, some of which gave rise to statistical learning approaches. Then, at the turn of the twentieth century (Kuznets, 1941), French mathematician Louis Bachelier published *Theory of Speculation*, which is widely regarded as the first scientific work on mathematical finance (Bachelier, 1900).

"It will soon be possible for portfolio managers and financial analysts to operate a high-speed computer with the ease of the desk calculator," said Joseph Gal in the *Financial Analyst Journal* in 1966 (Gal, 1966). Machine learning code has been simplified to the point that you can develop a machine learning option pricing model that is close to state-of-the-art with free internet computing resources in less than ten lines of code. This recalls the 1970s, when Black-Scholes option values (Black & Scholes, 1973) could be quickly calculated on a pocket calculator not long after the Chicago Board Options Exchange was established.

For finance, the efficiency market hypothesis and the adaptive markets hypothesis are two hypotheses that explain why we will never identify the underlying basic links in financial markets. Neither is inherently in contradiction with the other. Professional quantitative traders have been criticizing public researchers' faulty prediction models for years. They do, however, value new ideas that emerge from the field. Alfred Cowles, a well-known scholar in the field, published one of the first critical investigations on stock

market forecasting in the journal *Econometrica* (Cowles, *Stock Market Forecasting*, 1944). In 1933, he compared the performance of 24 random portfolios with 24 financial publication portfolios during the period 1928-1932 and, the financial publications outperformed none except one (Cowles, 1933).

He returned in 1944 to recognize that one strategy had outperformed the market by roughly two percentage points over a 40-year period, with largely constant performance across all sub-periods (Cowles, *Stock Market Forecasting*, 1944). This minor victory brought some solace to active managers involved in the argument. In a letter to colleagues two decades later, he stated that he believes that knowledge of financial patterns can assist to explain observed performance (Cowles, 1960), leaving the possibility of demonstrating the success of particular funds over time.

Advanced machine learning approaches were initially applied in the industry in the mid-to-late 1980s. This approach began when traders like Edward Thorp and Richard Dennis demonstrated great success by integrating technical and statistical trading strategies (Thorp & Dennis, 2003). In 1986, labs like the Morgan Stanley ATP group were established, with experts like Nunzio Tartaglia at the helm. Adams, Harding, and Leuck founded Man AHL London a year later, in 1987. David Shaw chose to launch his own quantitative fund, DE Shaw & Co., two years after joining Morgan Stanley. In the same year, James Simons rebranded his firm Monometrics to Renaissance Technologies to reflect its new quantitative focus, while Israel Englander started Millennium Management a few months later (Show, 2021).

This explosion was no accident; we witnessed the spread of computers as a result of these methods languishing in old papers, ready to be employed. In 1981, IBM released the Personal Computer, which would go on to provide the foundation for the contemporary personal computer business (Fradkov, 2020). This advancement resulted in increased computer power, storage capacity, and data processing capability. It enabled the accumulation of massive datasets, which were then utilized by faster and more powerful computers to statistically search financial data for patterns that might be used to forecast trade signals. These companies had previously attempted and tested a lot of what we now call machine learning during this time period (Fradkov, 2020). In the late 1980s, ex-IBM employees and famous academics at Renaissance Technologies worked with neural networks, nonparametric kernel regressions, and dynamic programming methodologies (Show, 2021).

Since the 1960's decade, expert or heuristic systems were used but, they have been slowly disappearing as their expectations did not meet reality and were also too complicated to use (Cesa, 2017). Researchers have used machine learning approaches such as reinforcement learning methodologies to construct end-to-end derivatives trading enterprises in recent years (Cesa, 2017). These algorithms might be used to simulate any derivatives portfolio. The agent is presented with a market and other limitations that real-world agents experience and is then asked to hedge a position using a set of accessible securities. As a result, we've moved to a world where rules are learnt automatically from data. Furthermore, if the business process can be replicated, data from the actual world may be captured and used to automate an action. Machine learning has extended the subset of commercially feasible programmed processes by simply learning from data.

The ramifications of such developments, such as the automation of derivatives operations, might be enormous. Will these automated firms outperform exchange-traded funds in terms of hedging performance and hence proliferate? Analysts are still obliged to acquire an intuition for where their traditional models make erroneous assumptions, and they must establish a long list of tools and workarounds that might one day be rendered obsolete by well-implemented reinforcement learning tactics.

However, automation isn't limited to end-to-end reinforcement learning models; there is a push to automate all prediction issues. AutoML, a new movement in automated machine learning, is gaining traction (Hutter, Thornton, & Hoos, 2014). For supervised learning, AutoML automates feature engineering, model selection, and hyperparameter optimization.

2.2 Evolution of techniques applied in the stock market

The mathematical techniques applied in the stock market with the use of computers have evolved throughout the years, being the next methods the biggest breakthroughs.

2.2.1 Instance-based methods

Instance-based algorithms, also known as memory-based algorithms, are descended from the 1967 closest neighbor classifier (Cover & Hart, 1967). These algorithms learn from training samples and then apply what they've learned to new situations using a similarity score. This approach uses a lot of memory, but it adapts fast to new data and has a short prediction time. Each new point considers the previous labelled points that are the closest to it (the neighbors), and they vote on what the new point's label should be.

Instance-based approaches have been difficult to gain traction in economics, partially due to the discipline's aversion to nonparametric models, and partly due to issues with tiny datasets. Researchers in the University of Akorn investigated which asset flow indicators better classify bond ratings in 1990. A linear discriminant analysis, closest neighbor, and probit analysis were used. In a paper titled Nonparametric and Semiparametric Methods for Economic Research, Miguel Delegado attempted to introduce the method in 1992, arguing that nonparametric models are useful even if only at an exploratory level because "... nonparametric models make no precise assumptions about functional form." Instead, the information is left to speak for itself" (Delgado, 1992).

Both classification and regression may be done with instance-based learning. In recent years, instance-based learning has grown in popularity. In 1996, it was used to anticipate US interest rates (Barkoulas & Baum, 1996), in 1997, for FX prediction (Lisi, 1997), and in 1999 (Agnon, 1999), for commodities forecasting. The nearest neighbors aren't the only method-based instance method, of course. Support Vector Machines have also been a popular model, and Vladimir Vapkin published a workable stock prediction example within three years after it was invented in 1995 (Osuna, 1998). Learning Vector Quantization, Self-Organizing Map, and Locally Weighted Learning are a few other approaches worth mentioning.

The nonparametric regression model became particularly popular in quantitative hedge funds around this period. Rene Carmona of Renaissance Technologies first using them to simulate high-dimensional non-linear connections in 1987 (Show, 2021). Because the financial industry kept its cards close to its breast, the strategies took some time to reach financial scholars. Andrew Lo published one of the earliest kernel regression applications in finance in 2000 (Lo, 2000). This is late, considering that both Nadaraya (Nadaraya, 1964) and Watson (Watson, 1964) provided the groundwork for kernel regression in 1964. However, by 1987, we have seen kernel estimation used in the context of econometric models (Ullah, 1988).

2.2.2 Tree-based methods

A decision tree is just a decision tree that isn't connected to a regression or classification issue to monitor its learning, much like a logistic function is just a function without a classification problem (Show, 2021). In 1963, the first regression tree was created, which used an impurity metric to iteratively partition data into two groups (Loh, 2014). A decision tree is a type of classifier that divides data into groups recursively. Algorithms like information gain, chi-square, and Gini index are used to separate nodes (Quinlan, 1986).

The statistical origins of decision trees may be traced back to William Belson's 1959 publication, in which he utilized a decision tree to match and forecast population samples (Belson, 1959). Since then, many techniques and strategies have been used to build and develop supervised tree models. They've gone through the AID (Morgan, 2012), ELISEE (Cellard, 1967), or CHAID (Kass, 1980) models. Decision trees have been utilized for a variety of purposes in recent decades, including assessing and predicting business performance (Delen, 2013), identifying fraud through financial statements (Chen S. , 2016), predicting bankruptcy (Foroghi, 2011), and predicting the evolution of stock prices (Harries & Horn, 1995).

2.2.3 Clustering methods

Clustering and dimension reduction are both multivariate approaches. Both strategies maximize various criteria to create a new feature space. Clustering aims to reduce the number of samples to a smaller number of patterns by looking for similarity among characteristics, whereas dimensionality seeks to eliminate superfluous columns from the data. They're called multivariate approaches since they both modify data using several characteristics (Khan & Bano, 2018).

In 1969, W.D. Fisher demonstrated the use of clustering and aggregation in economics (Fisher., 1969). As published in a work by B.F. King in 1966 (King, 1966), industry clustering was one of the earliest advances of unsupervised learning into machine learning in finance. He uses factor analysis to describe the degree of cross-sectional dependency and build clusters of comparable enterprises based on the observed covariance matrix of a broad collection of monthly returns.

With each researcher seeking to improve their clustering algorithm, the field exploded. In 1971, Cormack conducted a critical evaluation of the field and recommended that scholars begin doing comparison studies existing cluster techniques rather than just developing new ones (Cormack, 1971). Clustering has been successfully applied to portfolio management in recent years (Prado, 2016).

2.2.4 Neural networks

In the early 1990s, White sought to popularize neural nets in economics, but they did not result in significant performance increases at the time, and they did not become popular in economics (White, 1992). Before White, Kimoto & Asakawa authored an article to anticipate stock movements on the Tokyo Stock Exchange in 1990 (Kimoto & Asakawa, 1990), following a lengthy gap since Felson's studies in the 1970s. As American researchers began to catch up with Japan, a swarm of neural network articles appeared. Unlike in previous years, finance did not provide any meaningful innovation. Instead, neural architecture innovation happened, leading to new financial applications, some of which are discussed next.

Many financial applications benefit greatly from neural networks. John Hopfield invented recurrent neural networks in 1982 to aid with financial time series forecasting (Hopfield, 1982). A longer-memory solution called LSTM (Hochreiter & Schmidhuber, 1997) was created a little more than a decade later. In finance and economics, the performance of this sort of neural network is frequently compared to that of ARIMA and other mechanical time series models. These models were not widely believed in until they were demonstrated to outperform traditional models in the M4 time series competition in 2018 (Makridakis, 2018). By adding a logistic or sigmoid output function, these models may simply be converted for classification problems.

2.3 Long short-term memory models and their potential

As discussed in the previous section, long short-term memory (LSTM) networks (Hochreiter & Schmidhuber, 1997) have recently gained traction due to their positive results in predicting stock prices. In contrast with the original recurrent neural networks (RNNs) which only connect recent previous information and can't connect information as the time gap grows, LSTMs are able to remember information over a long period of time, making them better suited for predicting stock prices.

A recent study compared the pros and cons of LSTM in time series prediction, specifically in stock price prediction. In this case, the criterion of the pros and cons of the model was the mean square error between predicted value and real value. The conclusion was LSTM "can be well used in stock price forecasting" (Jiang, Tang, Chen, Wang, & Huang, 2019). Another paper published in 2020 researching forecasting stock prices with long short-term memory neural network and comparing its performance with a gated recurrent unit (GRU) neural network, showed the LSTM network had a better fitting degree and improved accuracy of the prediction results (Qiu, Wang Bin, & Zhou, 2020). A different study also compared the performance of an LSTM network and ARIMA model (a popular model used in time series prediction) in predicting stock prices and, results were very similar as the previous studies presented. In this case, "the empirical studies conducted and reported in this article show that deep learning-based algorithms such as LSTM outperform traditional-based algorithms such as ARIMA model. More specifically, the average reduction in error rates obtained by LSTM is between 84 - 87 percent when compared to ARIMA indicating the superiority of LSTM to ARIMA" (Siame-Namini & Siame

Namin, 2018). Another paper has been done to study the differences in performance between ARIMA and LSTM (Ding & Sun, 2022), showing similar results, as “the Root Mean Squared Error (RMSE) of LSTM model is lower than ARIMA, and the model based on deep learning method has stronger prediction ability on stock price index than traditional stock prediction model. This model is an effective stock prediction method.”

After the research done its clear LSTM networks have great potential and are currently one of the most promising types of neural networks for stock price forecasting. This is the reason why it has been chosen to predict stock prices in this bachelor thesis.

3. Methodology

3.1 Introduction

In this section are going to be presented the methodologies used to achieve the objectives of this bachelor thesis. First, the Markowitz model (Markowitz, 1952), used for the creation of portfolios will be presented, as well as the Sharpe ratio, used to choose one of the optimal portfolios. Secondly, the prediction model used for price prediction, Long-Short Term Memory or LSTM model (Hochreiter & Schmidhuber, 1997), followed by the method and the different criteria used to compare the results will be explained.

3.2 Portfolio Optimization Model

In order to create the optimal portfolio, Modern Portfolio Theory (Markowitz, 1952) has been applied. Harry M. Markowitz is credited with inventing new risk measurement techniques and their application to portfolio selection. He began with the concept of ordinary investors risk aversion and their desire to maximize expected return with the least amount of risk.

As a result, the Markowitz model serves as a theoretical framework for analyzing risk and return, as well as their interrelationships. He calculated risk using statistical analysis and employed mathematical programming to choose assets in a portfolio in an efficient manner. The concept of efficient portfolios arose from his theory. This is a dual model in which the objective function can be set to minimize portfolio risk for a given level of return or to maximize portfolio return for a desired level of risk.

Within a given amount of money or wealth, Markowitz created a number of portfolios based on investor risk and return preferences. Individual risk tolerance and asset preferences vary greatly. Individuals' financial resources, expenditures, and investment

needs differ. Given the preferences, the portfolio selection is a correct combination of securities rather than a simple pick of one or more stocks (Harrod, 1937).

Investors examine two dimensions of investment: risk and reward. Depending on the assumptions, the predicted return may differ. The variation of the distribution around the mean, its range, and other factors are used to calculate the risk index, which are referred to as variance and covariance in statistical terminology. Markowitz is credited with defining risk and emphasizing the need of maximizing return with the least amount of risk. This led to the development of Modern Portfolio Theory, which emphasizes the risk-reward tradeoff. If the investor wants a bigger return, he must be willing to take on more risk. However, he prefers a large return with a minimal risk, posing a tradeoff conundrum.

The selection of securities in a portfolio of assets is important. A portfolio is a collection of assets or securities. Depending on his wealth, income, and tastes, each individual investor invests in a variety of assets. The classic portfolio theory holds that assets should be chosen based on their low risk, as assessed by their standard deviation from the mean of projected returns. The more the return variability, the higher the risk.

In contrast to the Traditional Theory (Harrod, 1937), which states that standard deviation measures return variability and risk is indicated by variability, and that the best securities are those with lower variability, the modern Portfolio Theory emphasizes the need for return maximization through a combination of securities with lower total variability.

Each investment has a different risk than the others, and by properly combining securities (diversification), one can achieve a combination in which the risk of one is partially or completely offset by the risk of the other. To put it another way, the variability of each security, as well as the covariance for their returns as expressed by their inter-relationships, should be considered.

Expected returns, variance of these returns, and covariance of the returns of the stocks within the portfolio are all factors to consider while choosing a portfolio, according to Modern Portfolio Theory. If a portfolio is predicted to generate the maximum return possible for the lowest risk or a given level of risk, it is considered to be efficient.

By mixing several assets whose combined risk is lowest for a particular level of return for the same amount of investment that the investor is capable of, a set of efficient portfolios may be formed. As previously stated, Markowitz's hypothesis is founded on a variety of assumptions. These are:

1. Investors are rational and act in a way that maximizes their utility for a given amount of money or revenue.
2. Investors have unrestricted access to accurate and fair information about returns and risks.
3. Markets are efficient and swiftly and perfectly absorb information.
4. Investors are risk averse and want to reduce risk while increasing reward.
5. Expected returns and variance or standard deviation of these returns from the mean are used by investors to make judgments.
6. For a given amount of risk, investors prefer higher returns than lower returns.

Therefore, when an investor is looking for the greatest portfolio in terms of return-risk among the many options available, it must go through two processes. The first is to identify a collection of effective portfolios, creating the efficient frontier. The second step is to choose a specific portfolio from the efficient set, based on the investor's desired return, risk, or return-to-risk ratio preference.

Getting now into the practical aspect of Markowitz, if we assume an infinite number of portfolios are created, the efficient frontier will be created (Figure 1).

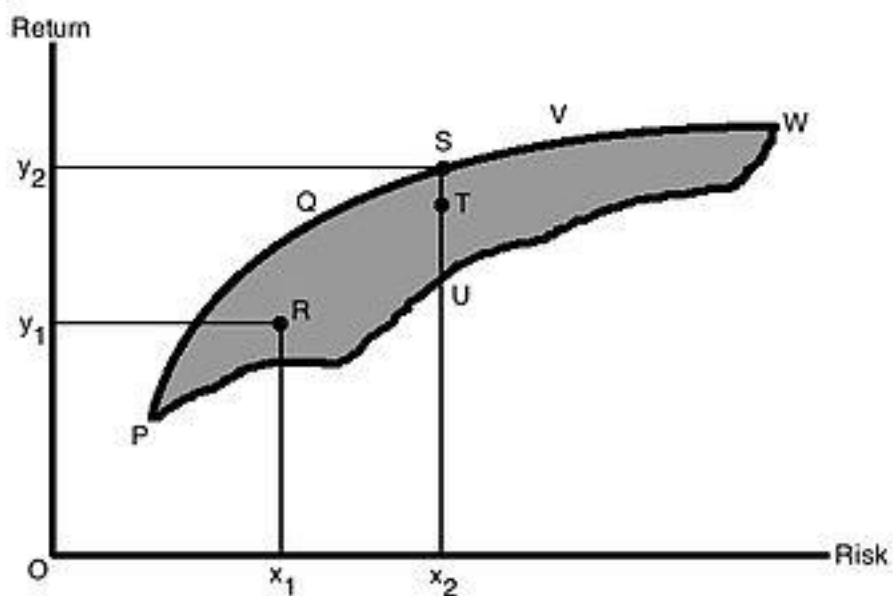


Figure 1: Efficient Frontier (Scherer, 2002)

As Markowitz assumes the investor is rational, it would want a bigger return and a lower risk (investor is risk averse). The shaded area PVWP in Figure 1 represents all of the different securities in which an investor can invest, often referred to as the feasible

region. The most efficient portfolios are those that fall on the PQVW's edge. At risk level x_2 , for example, there are three portfolios: S, T, and U. Portfolio S is known as the efficient portfolio since it has the highest return, y_2 , when compared to portfolios T and U. For a given risk level, all portfolios that lie on the PQVW boundary are efficient portfolios. This concept is usually referred to as dominance.

The Efficient Frontier is the border PQVW. All portfolios that fall below the Efficient Frontier are not good enough since the return for the risk would be lower. Portfolios to the right of the Efficient Frontier would not be suitable, as there is a higher risk associated with a given rate of return. Efficient Portfolios are all portfolios that fall within the PQVW border. The Efficient Frontier is the same for all investors, because they all desire the best potential return with the least amount of risk, and they are all risk averse.

To plot each portfolio, the expected return and risk need to be calculated. These are calculated in the following manner, following the Modern Portfolio Theory by Markowitz (Markowitz, 1952):

$$\text{- Expected portfolio return: } E(R_p) = \sum_i W_i * E(R_i) \quad (3.1)$$

Where,

W_i = Weight of asset i

R_i = Return of asset i

R_p = Return of the portfolio

$$\text{- Risk of portfolio: } \sigma_p = \sqrt{\sum_i \sum_j W_i * W_j * \sigma_{ij}} \quad (3.2)$$

Where,

σ_{ij} = Covariance between asset i and asset j

W_i = Weight of asset i

W_j = Weight of asset j

Therefore, the risk of the portfolio is the standard deviation.

Markowitz proposed a quadratic programming model to construct the efficient frontier, which can be formulated by minimizing risk subject to a set of constraints. Therefore, following its model, the objective function is to minimize the risk of the portfolio:

$$\text{MIN } \sigma_p = \sqrt{\sum_i \sum_j W_i * W_j * \sigma_{ij}} \quad (3.3)$$

Next are the constraints. The first one is the target expected return of the portfolio the investor wishes to achieve.

$$E(R_p) = \sum_i W_i * E(R_i) \quad (3.4)$$

The next constraint forces the sum of the weights to be equal to one, meaning the investor must invest everything in any of the stocks:

$$\sum_i W_i = 1 \quad (3.5)$$

Lastly, every weight must be greater than zero, meaning no short positions are permitted, only long.

$$\forall W_i \geq 0 \quad (3.6)$$

One thing to note is, no constraint has been set on the minimum or maximum number of stocks that make up the portfolio. Plus, there's no restriction on the minimum or maximum weight of each stock.

After calculating the portfolios in the efficient frontier, one of these needs to be selected. For this bachelor's thesis, the method chosen to resolve this problem has been the Sharpe Ratio (Sharpe, 1966).

The Sharpe ratio is a formula that measures a portfolio's performance by taking into consideration the portfolio's return, the return of a risk-free asset, and the portfolio's standard deviation (risk). It indicates the additional portfolio return for a particular degree of risk. The Sharpe ratio of a portfolio determines its performance. The higher the Sharpe ratio, the better. The Sharpe ratio is maximized in only one portfolio (the tangency portfolio); portfolios below and above the tangency portfolio reduce the Sharpe ratio. When the Sharpe ratio is negative, it means that a risk-free asset would outperform the portfolio under consideration.

In this case, the risk-free rate has been assumed to be the 30-year bond rates offered. For instance, when creating portfolios for the window 2007-2008, the risk-free rate is the mean of all the 30-year bond rates offered during that time. In case of the S&P500 and DJI indexes, US bonds were used, whilst German bonds were used when calculating the optimal portfolio for the DAX30 index.

To calculate the Sharpe ratio, the following formula is:

$$\text{Sharpe Ratio} = \frac{R_p - R_f}{\sigma_p} \quad (3.7)$$

Where,

R_p = Expected portfolio return

R_f = Risk-free rate

σ_p = Risk (or standard deviation) of portfolio

This bachelor's thesis uses the theory explained to construct optimal portfolios based on the last two-year historical prices of each asset in an index, in order to use the portfolios for the next trimester, quarter, semester or year. However, to achieve the highest return possible, a way to enhance the optimal portfolios created using the Modern Portfolio Theory (MPT) and the Sharpe ratio, is to predict future prices and include these in the Markowitz model at the time of creating the optimal portfolios. Therefore, the next subsection focuses on the method used to predict prices.

3.3 Prices prediction

As discussed in the previous sections, the goal is to predict stock prices for the next trimester, quarter, semester and year in order to be used in the Markowitz model in conjunction with the past prices. To achieve this, a Long Short Term Memory network (LSTM) will be used.

LSTMs are a type of Recurring Neural Network (RNN) (Rumelhart, Hinton, & Williams, 1986) that can learn long-term dependencies. Hochreiter & Schmidhuber introduced them (Hochreiter & Schmidhuber, 1997), and numerous individuals developed and

popularized them in subsequent work. They are currently frequently utilized and function exceptionally effectively on a wide range of situations, specially predicting time series.

LSTMs are specifically developed to prevent the problem of long-term dependency. They don't have to work hard to remember knowledge for lengthy periods of time; they are designed to excel on this area.

All recurrent neural networks are made up of a series of repeated neural network modules. This repeating module in ordinary RNNs will have a relatively simple structure, such as a single tanh layer, as illustrated in figure 2.

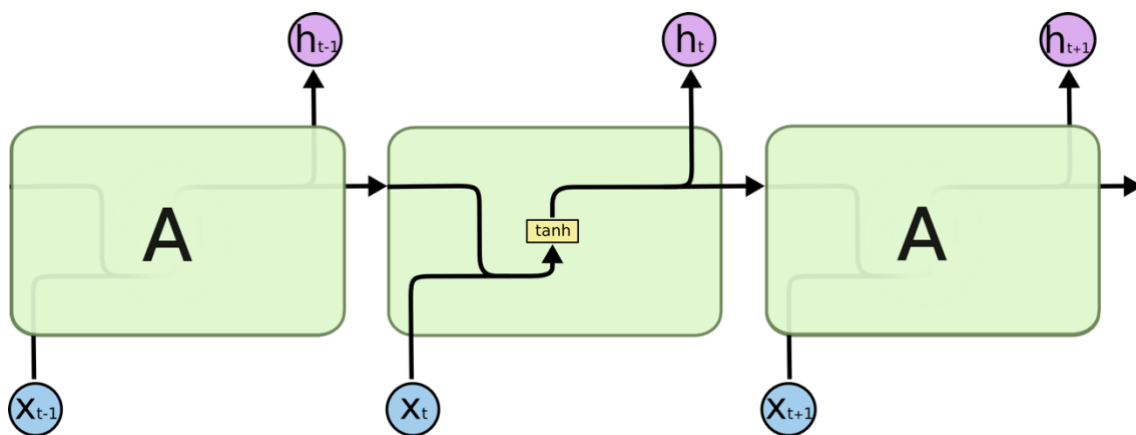


Figure 2: Repeating module in a standard RNN containing a single layer (Olah, 2016)

LSTMs have a chain-like structure as wells, but the repeating module is different. Instead of a single neural network layer, there are four, each of which interacts in a unique way.

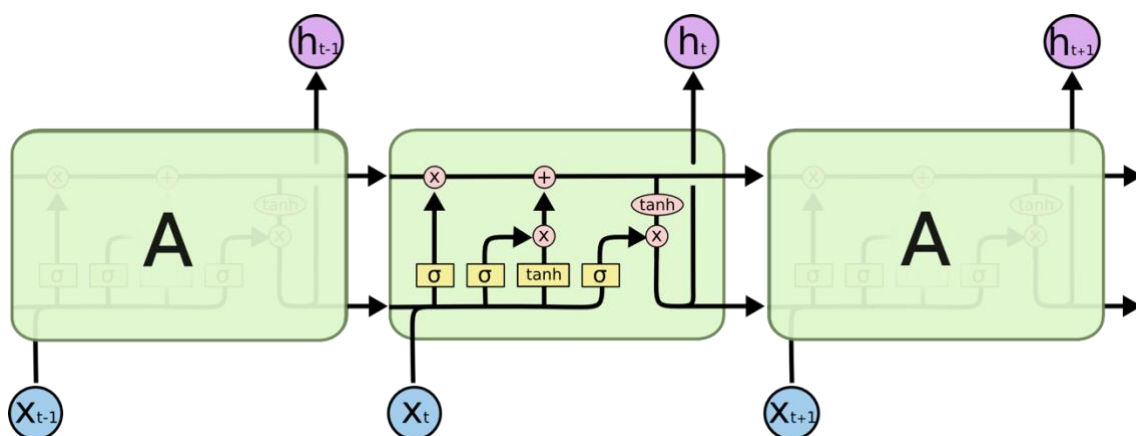


Figure 3: Repeating module in an LSTM contains four interacting layers (Olah, 2016)

Each line in figure 3 transmits a full vector from one node's output to the inputs of others. The pink circles denote pointwise operations, such as vector addition, and the yellow boxes denote learnt neural network layers. Concatenation occurs when lines merge, whereas forking occurs when a line's content is replicated and the copies are sent to various locations.

The cell state, the horizontal line going across the top of figure 3, is the key to LSTMs. It can be seen clearer in figure 4.

The condition of the cell is similar to that of a conveyor belt. With only a few linear interactions, it flows straight down the whole chain. It's incredibly easy for data to just travel over it unaltered.

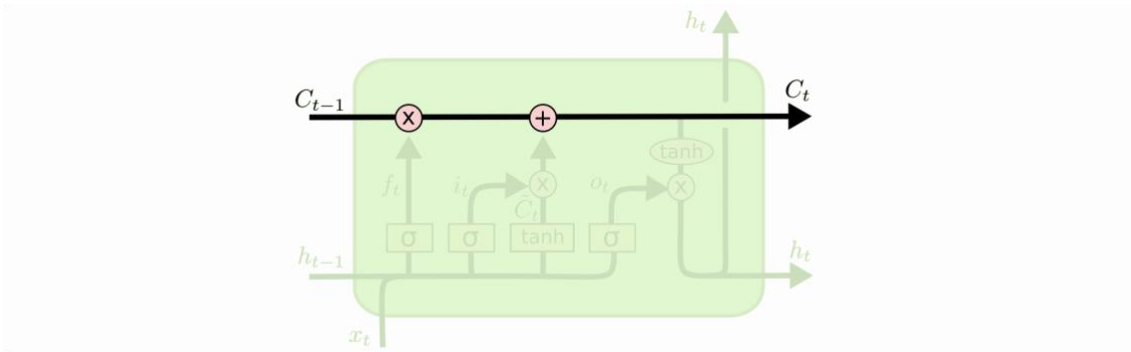


Figure 4: LSTM module (Olah, 2016)

The LSTM may delete or add information to the cell state, which is carefully controlled by structures called gates.

Gates are a mechanism to selectively allow information to pass through. A sigmoid neural net layer plus a pointwise multiplication operation makes them up (see figure 5).

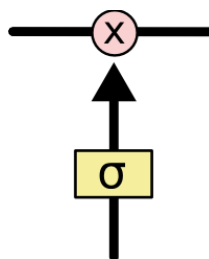
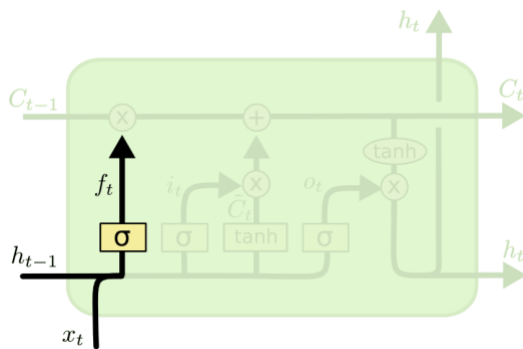


Figure 5: Sigmoid layer (Olah, 2016)

The sigmoid layer produces integers ranging from zero to one, indicating how much of each component should be allowed to pass. Zero means none goes through, whilst one let's everything through. Three of these gates are present in an LSTM to safeguard and govern the cell state.

3.3.1 Step-by-step LSTM walkthrough

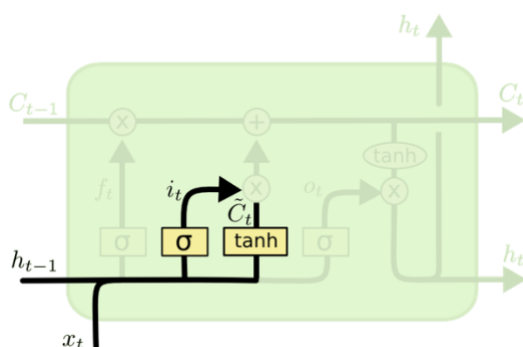
The first stage in an LSTM model is to select which information from the cell state will be discarded. The "forget gate layer," a sigmoid layer, makes this judgment. Looking at Figure 6, it examines the values in h_{t-1} and x_t and returns a number between 0 and 1 for each number in cell state C_{t-1} . A 1 stands for "totally keep", whereas 0 stands for the opposite.



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

Figure 6: "Forget gate layer", a sigmoid layer (Olah, 2016)

The next stage as seen in Figure 7 is to figure out what additional data we'll be stored in the cell state. There are two components to this. The "input gate layer," a sigmoid layer, chooses which values we'll be updated first. A tanh layer then generates a vector of new candidate values, C_t , that might be added to the state. Both will be combined in the following step to make a state update.



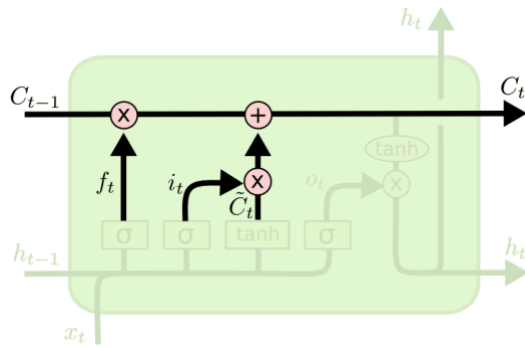
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Figure 7: Input gate layer (Olah, 2016)

Following on Figure 8, it's now time to switch from C_{t-1} to C_t . It is already known what to do because of the previous stages; now all that's necessary is follow through.

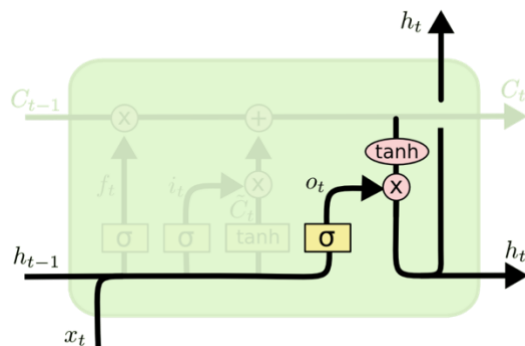
The previous state is multiplied by f_t , forgetting the items previously agreed to ignore. Then $i_t * C_t$ is added to the equation. This is the new set of candidate values, scaled by how much each state value was updated.



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

Figure 8: Update cell states in LSTM module (Olah, 2016)

Finally, as seen in Figure 9, the output must be chosen. The cell state will be the basis for this output, but it will be filtered. To begin, a sigmoid layer runs to determine which bits of the cell state will be output. The cell state is then passed through tanh (to force the values to be between -1 and 1) and multiplied by the sigmoid gate output, resulting in only the sections wanted to output.



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

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

Figure 9: Output sigmoid layer (Olah, 2016)

All in all, a certain amount of historical prices is the input of the LSTM network which, will output another certain amount of predicted prices. For the purpose of this bachelor's thesis, several models with different amounts of input historical prices and future output prices have been implemented, to compare which model works best. The specifics will be discussed in the Analysis section of this bachelor's thesis.

One thing to note is, the predictions made by the LSTM model will always have some level of error. Therefore, as these errors will be used as data for the construction of

portfolios using the Markowitz model, different measures of error should be considered to analyze how they affect the behavior of the constructed portfolios. Therefore, the next subsection addresses this issue by presenting several error measurements that have been used.

3.3.2 Error measurements for predicted prices

In order to measure the error in prediction models, many different measures exist. In this case, three different error measures have been used to calculate the error of each prediction model. These are:

1. Mean absolute error (MAE)

The mean absolute error (MAE) is a statistic that measures the difference in errors between paired observations describing the same phenomena (Sammut & Webb, 2011). Comparisons of predicted against observed or subsequent time versus starting time are examples of Y versus X. The MAE is determined as follows:

$$MAE = \frac{\sum_{i=1}^n |Y_i - X_i|}{n} \quad (3.8)$$

Where,

Y_i = Predicted value

X_i = True value

n = Size of vector

2. Mean absolute percentage error (MAPE)

In statistics, the mean absolute percentage error (MAPE) is a measure of a forecasting method's prediction accuracy (Khen, 2000). The accuracy is commonly expressed as a ratio calculated as:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{X_i - Y_i}{X_i} \right| \quad (3.9)$$

Where,

X_i = Actual value

Y_i = Forecast value

n = Size of vector

3. Root-mean-square error (RMSE)

The root-mean-square error (RMSE) is a commonly used measure of the variations between predicted and observed values (sample or population values) by a model or estimator. The square root of the second sample moment of the discrepancies between predicted and observed values, or the quadratic mean of these differences, is represented by the RMSE. When the computations are performed over the data sample that was used for estimate, these deviations are referred to as residuals, and when they are computed out-of-sample, they are referred to as errors (or prediction errors). The RMSE combines the magnitudes of prediction errors for different data points into a single measure of predictive capacity. Because RMSE is scale-dependent, it is used to evaluate forecasting errors of different models for a single dataset rather than across datasets (Shekhar, Xiong, & Zhou, 2017).

The following formula is used to calculate the RMSE:

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (Y_t - X_t)^2}{n}} \quad (3.10)$$

Where,

X_i = Actual value

Y_i = Forecast value

n = Size of vector

3.4 Results comparison criteria

Once the optimal portfolios have been created using Markowitz with and without the predicted prices, as well as for each prediction model implemented, it is time to compare the results.

1. Logarithmic return

To do so, the return of each portfolio has been calculated. This has been calculated as its logarithmic return (Miskolczi, 2017).

$$R_p = \sum_i \ln \left(\frac{P_{ti}}{P_{t-1i}} \right) * W_i \quad (3.11)$$

Where,

W_i = Weight of stock i

P_t = Price at end date of stock i

P_{t-1} = Price at start date of stock i

Being start and end date the first and last day respectively of the period the portfolio is used for.

2. Mean return

Once the return of each portfolio is calculated, the mean return of a forecast model can be calculated, which is the mean of all portfolios returns produced by the specific model being calculated. For instance, if for the period 2007-2008 a model has a return of 10% and during the period 2008-2009 the same model has a return of 20%, and so on until 2019-2020, the mean return of the model will be the mean of all those returns. Therefore, the mean return of a model is:

$$R_m = \frac{\sum_{i=1}^n R_p}{n} \quad (3.12)$$

Where,

n = Number of periods

R_p = Return of portfolio

3. Mean risk

Also, the mean risk of the model can be calculated in a similar way:

$$\sigma_m = \frac{\sigma_p}{n} \quad (3.13)$$

Where,

n = Number of periods

σ_p = Risk of portfolio

4. Costs

However, the return of a model doesn't consider commissions of selling and buying stocks, as well as the cost of opening a trading account and maintaining it. Due to its limited ability to express the true profit a person could've achieved using each model, the profit has to take into account the costs of using a broker. In this case the selected broker has been Interactive Brokers (InteractiveBrokers LLC, 2021), which has the following costs:

Annual maintenance cost = Cost of maintaining trading account annually of 91\$ (InteractiveBrokers LLC, 2021)

Transaction fee = Cost of buying/selling stocks of 0.05% per transaction (includes the market fees)

5. Total values

To apply all costs described a simulation for each model has been done, where an imaginary investor invests 100.000€ to start with. This means, for each period, the current total portfolio value and cost of the portfolio has been calculated as:

$$Total\ Portfolio\ Value_t = \sum_i \frac{P_t}{P_{t-1}} * W_i * Total\ Portfolio\ Value_{t-1} \quad (3.14)$$

Where,

P_t = Price at end date of stock i

P_{t-1} = Price at start date of stock i

W_i = Weight in current portfolio of stock i

Total Portfolio Value_{t-1} = Total Portfolio Value of the previous period (or initial investment if calculating revenue for the first period)

Then, the total cost per period is calculated as follows:

$$\begin{aligned} Total\ cost &= \frac{annual\ maintenance\ cost}{number\ of\ periods\ per\ year} \\ &+ transaction\ fee * Total\ Portfolio\ Value_{t-1} \\ &+ transaction\ fee * total\ Portfolio\ Value_t \end{aligned} \quad (3.15)$$

Known both, the current profit per period calculation is:

$$Profit = Total\ Portfolio\ Value_t - Total\ cost - Initial\ Investment \quad (3.16)$$

The profit an investor would have achieved is therefore the profit calculated for the last period. This value will be used to compare each model between each other, to decide what could be the best possible strategy for an investor in the future. One thing to note is, the profit calculated also includes the initial investment.

Lastly, in order to compare the return between models with different timeframes, returns have to be annualized.

$$\text{Annualized Return} = R_m * N \quad (3.17)$$

Where,

R_m = Return of the model

N = Number of periods in a year of the timeframe used in the specific model.

For instance, if a model uses a year as it's timeframe, N will be equal to one. However, if the timeframe used is a trimester, N will be equal to 4.

4. Solution Development

4.1 Introduction

The purpose of this section is to fully explain the process followed to achieve the objectives of this thesis. Firstly, a short high-level introduction of the process will be showcased. Then, everything regarding the data used will be discussed: source, download method used and cleaning and structuring of the data.

Next, it'll be explained how the Markowitz model (Markowitz, 1952) was applied on this case to calculate the efficient portfolios. After, all the different models used to predict prices will be discussed, breaking them down into sub-groups in order to compare their performance. Therefore, lastly, a detailed analysis of each model used will be done to compare the performance achieved between them, in terms of errors and other parameters in the prediction of prices and, the final performance, measured by the profit achieved after using each model.

After the analysis, the conclusions extracted out of it will be outlined.

4.2 High-level solution

In an effort to achieve the objective of this thesis, create an optimized portfolio of stocks using stock value prediction, the following steps have been followed:

1. Three stock indexes have been used: Deutscher Aktien Index (Chen J. , 2022), Standard and Poor's (The Editors of Encyclopedia Britannica, 2022) and Dow Jones Industrial Average (Ganti, 2022). The purpose of this is to get all stocks that have been part of these indexes since 2005 so, the window is big enough to make enough comparisons on the analysis stage (in terms of the amount of years of data, 2005-2020 and, in terms of number of stocks). Plus, using indexes will enable the possibility of comparing the results with their exchange-traded fund

- (ETF) that mimics the corresponding index, which will show if this solution is better than the average performance of each index.
2. Data extraction: Collect all past prices of all stocks to be analyzed. All companies that are part, as of 12/15/2021, of the S&P500, DJI and DAX30 and up to 2005. In addition, the price of 30-year US and German bonds for the same years have also been extracted, which have been used as the risk-free rates when applying the Sharpe ratio.
 3. Data normalization and cleaning: After the data had been collected, the data has been normalized and cleaned, to ensure that there are no anomalies in the data, so that the models can be successfully trained.
 4. Generate efficient stock portfolios: For each of the three indexes, efficient portfolios have been generated using the Markowitz model (Markowitz, 1952) for each trimester/quarter/semester/year between 2007-2020. Four different timeframes have been used to compare more models and discuss the best possible solution. Take into account the start year is 2007 as, as it will be discussed later, certain prediction models need more than one year of past stock prices to predict the next prices. Also, a different portfolio will be created for every timeframe starting from 2007 until 2020. For instance, in the case of an annual timeframe, a portfolio will be created for the period 2007-2008, another for 2008-2009, and so on until the period 2019-2020.
 5. Apply a neural network model (LSTM) to predict stock prices: Using the data of stock prices collected, the model predicts the stock price of each stock for the next trimester, quarter, semester and year. Different models have been used, for instance, using the last month, semester or year historical prices to predict the next trimester, quarter, semester and year prices. An example is, given the 2006 stock prices of NIKE, predict the prices for the next semester, from 01-01-2007 until 01-07-2007. All models have been described fully in the 'Price prediction models' section.
 6. Generate efficient stock portfolios using the predicted prices: After predicting the stock prices, repeat step 4 but now, making use of the predicted prices to create the portfolios. A different portfolio has been generated for each model.
 7. Compare results: Once the portfolios have been generated with and without the predicted prices, the performance of each portfolio has been compared. To do so, a simulation has been done for each portfolio, which emulates the cost of an investor of buying/selling the stocks each trimester/quarter/semester/year, depending on the model used to generate each portfolio, to take into account the

changes there are with each portfolio between timeframes. This simulation is later explained in detail in the Analysis section.

4.3 Data

In order to complete this thesis, a big amount of data has been collected for different purposes.

To start with, as discussed before, the stocks that have been part of three different indexes (S&P500, DJI and DAX30) from 2005-2020 have been used. To extract the name of all these stocks, a scraping method has been used.

For S&P500, the table in Wikipedia (Wikipedia, 2021a) about the index containing all companies in the index as of 2021-12-01 was scraped with a python script.

The same process was repeated to retrieve the DAX30 (Wikipedia, 2021b) and DJI (Wikipedia, 2021c) stock names.

Next are the prices of the US and German 30-year bond rates used in the calculation of the Sharpe ratio (Sharpe, 1966) as the risk-free rate of the S&P500 and DJI stocks and, DAX30 stocks respectively. The US 30-year bond rates between 2005-2020 have been collected from the US Treasury web site (U.S. DEPARTMENT OF THE TREASURY, 2022) and, downloaded in .csv format. On the other hand, the German 30-year bond rates have been extracted from Investing.com (Investing.com, 2022) and, downloaded in the same way as the US bond rates, a .csv file.

Now continuing with the stocks used, after collecting all the names of all the stocks, the closing prices of each day from 2005-2020 (both included) of all of them are needed. To get all, a public Python API was used. Specifically, the standard and most used API to do so was used, yfinance (PyPi, 2022). Getting more into detail, the method “download” was utilized:

```
data = yf.download(symbols_list, start="2005-01-01")['Close']
```

This function returns a list in the form:

Table 1: Stock prices list form Source

	Stock 1	Stock 2	Stock 3	...
Date	Closing Price	Closing Price	Closing Price	...
Date	Closing Price	Closing Price	Closing Price	...
...

After downloading all the necessary data, it is necessary to normalize and clean it. To do so, all the prices stored of each stock for each index were saved in an excel (one excel file for each index) using the same format as returned by the yfinance API. For the purpose of manipulating the data efficiently, the tool PowerBI (Microsoft, 2022) was used.

To assure only stocks with prices between 2005-2020 were included, all stocks without a closing price between these dates were eliminated. As a result, after the cleaning and normalization of the data, there were this many stocks on each index:

- DAX: 23 stocks
- DJI: 28 stocks
- S&P500: 409

For reference, this means there are 23 stocks in the DAX index at the moment which were already trading in the stock market in 2005.

Now all the data needed is normalized, it is ready to be used to generate the efficient portfolios using the Markowitz model (Markowitz, 1952), the optimal portfolio using the Sharpe ratio (Sharpe, 1966) and in the prediction of price models. All steps to apply the mentioned models are explained next.

4.4 Generate optimal portfolio

Once all prices are available, the return of each stock can be calculated, so (Investing.com, 2022) it is now possible to create the optimal portfolios based on the Markowitz model (and the Sharpe ratio (as explained before in subsection 3.2). Therefore, for each timeframe (trimester/quarter/semester/annual) between 2007-2020 an optimal portfolio needs to be generated (for now, not including any predicted prices).

To apply both methods, Markowitz and Sharpe ratio, the following steps were followed using Python:

1. Firstly, the covariance matrix is calculated to understand how assets interact between each other (parameter ij in the Markowitz model) during the specific timeframe for which the portfolio is being calculated for.
2. Next, the expected return of each asset is calculated as the percentage change between each period, where the length of the period will depend on the timeframe being calculated.
3. Once both previous steps are done, the most work intensive step is next, creating the efficient frontier. To do so, a loop is runs 10000 times and, in each iteration, the loop considers different weights for assets and calculates the return and volatility of a particular portfolio combination.
4. Lastly, after the efficient frontier has been generated, one of the portfolios needs to be selected. In order to select the optimal portfolio, the Sharpe ratio is used, as discussed before.

An example of an efficient frontier generated can be seen below (Figure 10). More specifically, it is the efficient frontier generated for the DAX30 index in the period 2005-2007. The green star indicates the portfolio selected by the Sharpe ratio, whilst the red star indicates the portfolio with the minimum volatility.

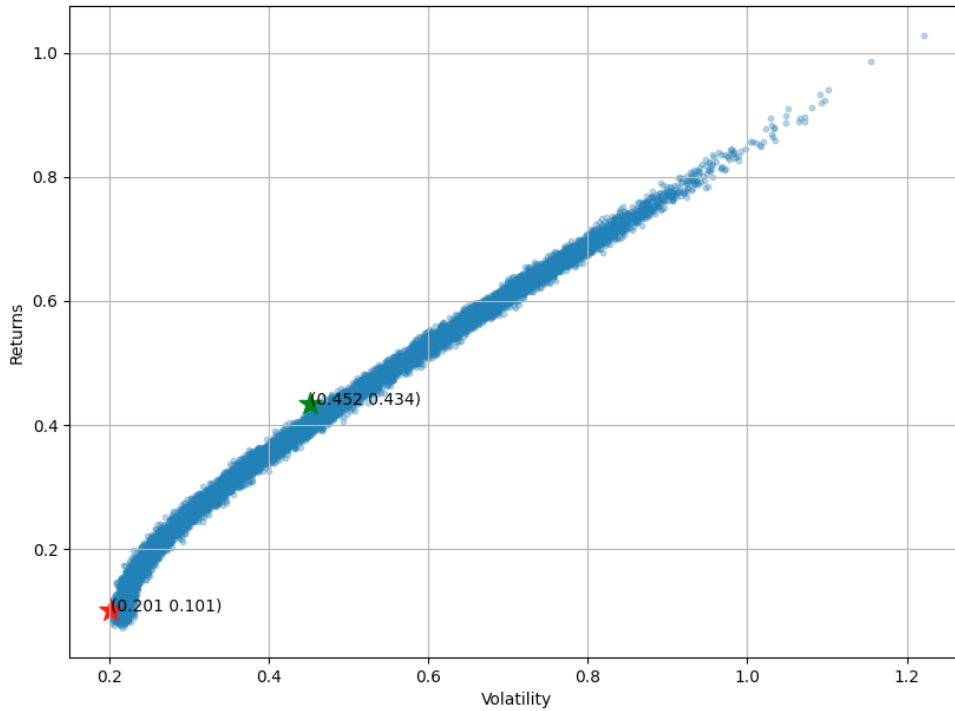


Figure 10: Efficient Frontier DAX30

This process is repeated for all timeframes (trimester, quarter, semester and annual) between 2007-2020, taking into account the prices of two years prior the specific timeframe. For instance, for the first trimester period (01/01/2007 – 01/04/2007) for the index DAX30, all prices between 01/01/2005 – 01/01/2007 will be taken into account to generate the portfolios. For the next trimester period, 01/04/2007 – 01/07/2007, all prices between 01/04/2005 – 01/04/2007 are considered.

As a result, all optimal portfolios for each timeframe and period are created. This are stored in an excel file in the following form:

Table 2: Optimal portfolios form

Date	Stock 1	Stock 2	...	Expected return	Volatility
Date 1 – Date 2	Weight stock 1	Weight stock 2	...	Expected return of portfolio	Volatility of portfolio
Date 2 – Date 3	Weight stock 1	Weight stock 2

...
-----	-----	-----	-----	-----	-----

Now all optimal portfolios have been calculated for every timeframe and period, it is time to create them using the predicted prices. But first, the prediction of the stock prices needs to be done.

4.5 Price prediction models

As discussed in the Methodology section, a Long-Short Term Memory (LSTM) neural network is used due to their known excellence in predicting time series given past values.

To achieve the best possible outcome, a series of models with certain modifications between them have been tested to compare the results between them, to find the optimal predictive model which provides the highest return.

The models have been divided first in two sections:

1. Automatic recurrence models: In this case, the number of output values of the LSTM network will exactly be the number of future prices wanted. For instance, if the number of future prices is all the prices of the next year, 250 prices, only one LSTM will be used with the number of outputs equal to 250. This means, when the LSTM is calculating the output 100 for example, it'll have to use automatically the previously predicted values [1-99] by itself.
2. Manual recurrence models: On the other hand, when using manual recurrence and wanting to predict the next 250 prices, the LSTM network will be set to only predict the next 30 prices at a time. Once the LSTM network returns the 30 prices, all these prices returned will then be used as input of another LSTM network, hence the naming 'manual recurrence'. This process will be repeated until all the 250 future prices are predicted.

The advantage of using a manual recurrence approach is the LSTM network won't be overcharged with weights, increasing its efficiency. Therefore, distinguishing between these two types of models will enable us to check not if only the manual recurrence is

more efficient but, if it's also more effective than the automatic recurrence or, if the improvement in efficiency forces a decrease in effectiveness.

Apart from the type of recurrence, models have also been divided between the number of input prices or, in other words, the number of past prices considered by the model:

1. 63 past prices (trimester)
2. 125 past prices (semester)
3. 250 past prices (annual)

And lastly, similarly to the number of input values, the number of output values also varies (the number of predicted prices):

1. 63 future prices (trimester)
2. 84 future prices (quarter)
3. 125 future prices (semester)
4. 250 future prices (annual)

Combining all, there are a total of 24 different models, but the four more models have to be added, one for each timeframe which don't use predicted prices, making a total of 28 models. All are analyzed in the Analysis section to figure out the optimal model considering different parameters.

After the prediction of the prices, optimal portfolios need to be created with the use of the predicted prices. To do so, the same process described in the previous section 'Generate optimal portfolio' must be ran but now, considering the predicted prices. Therefore, other than considering the last two-year prices, the next future trimester, quarter, semester or year prices (depending on the timeframe being calculated) need to be considered when applying the Markowitz model.

One thing to note is, every one of the 24 models has predicted different prices so, a file containing the optimal portfolios for each period has to be created in the same way as shown in Table 2.

4.5.1 Toolkit

In order to create, train and apply all models, a Python toolkit has been used. Specifically, the toolkit used has been Keras (Chollet, 2015), which is a Python-based deep learning API that runs on the TensorFlow machine learning platform. It was created with the goal of allowing for quick experimentation. Keras is defined with three words:

1. Simple - but not simplistic. Keras relieves developer cognitive load, allowing more focus on the most important aspects of the problem.
2. Flexible – Adopts the principle of progressive disclosure complexity. Simple processes should be quick and easy, whereas arbitrarily advanced workflows should be feasible via a clear path
3. Powerful. Keras provides industry-leading performance and scalability, and it is utilized by NASA, YouTube, and Waymo, among others.

5. Analysis

In this section, several analyses are presented, with the aim of showcasing:

1. Difference in mean return and absolute profit between portfolios built using each model and portfolios without the use of predicted prices
2. The performance of each model applied to predict stock prices, measured by their error and time to be trained.
3. Difference in mean return between each index exchange-traded funds and the portfolios built using predicted prices.

5.1 Portfolio performance analysis

In order to measure the performance of each portfolio four different measures will be used:

1. The average annualized return of the portfolio from 2007-2020 (formula 3.17)
2. The mean risk of the portfolios created by the model (formula 3.13)
3. The total portfolio value at the end of 2020 (formula 3.14)
4. The total cost of following the changes in the portfolio the model proposes each time a timeframe is over (formula 3.15)
5. The final profit also from 2007-2020 (formula 3.16), which assumes an 'investor' has invested 100.000€ to start with.

All measures are explained in detail in section three, methodology.

On another note, 'A-R' and 'M-R' in the following figures represent automatic and manual recurrence respectively. Also, the numbers 30, 125 and 250 represent the number of input data used to train each model. Annual, semester, quarter and trimester indicate the timeframe predicted by each model. All these are explained in detail in subsection 4.5.

In addition, before getting into detail, in order to refresh the different models that are going to be analyzed, the following graph has been done (figure 11):

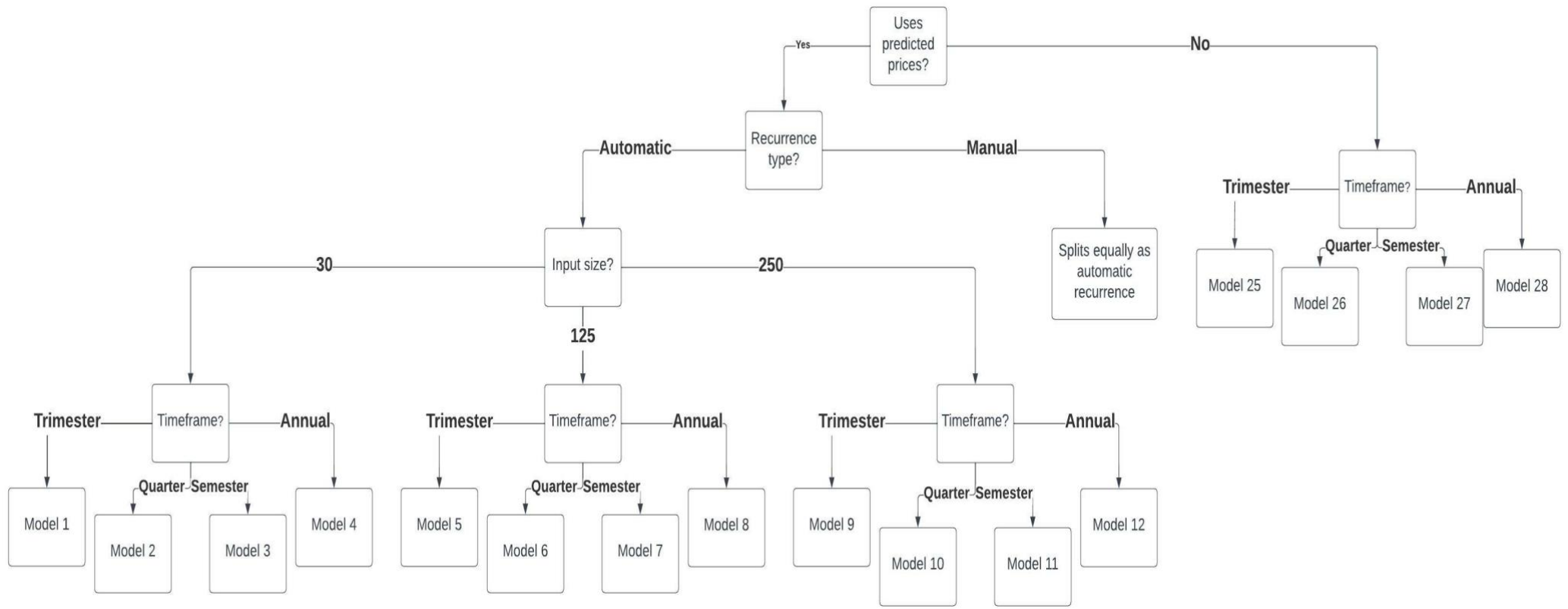


Figure 11: All models and how they are split

As seen in figure 11 and as explained in subsection 4.5, models are differentiated on several parameters. First, they differ based on whether they use predicted prices or not. After this, models that make use of predicted prices split into two different categories again those that use automatic recurrence and those that use manual recurrence. Then, they are broken down depending on the size of the input (30, 125 or 250 past prices). Lastly, irrespective of the use of predicted prices, models are split up based on the size of the window (timeframe).

5.1.1 S&P

To start with, the mean return achieved by all models by building portfolios with only S&P stocks will be analyzed.

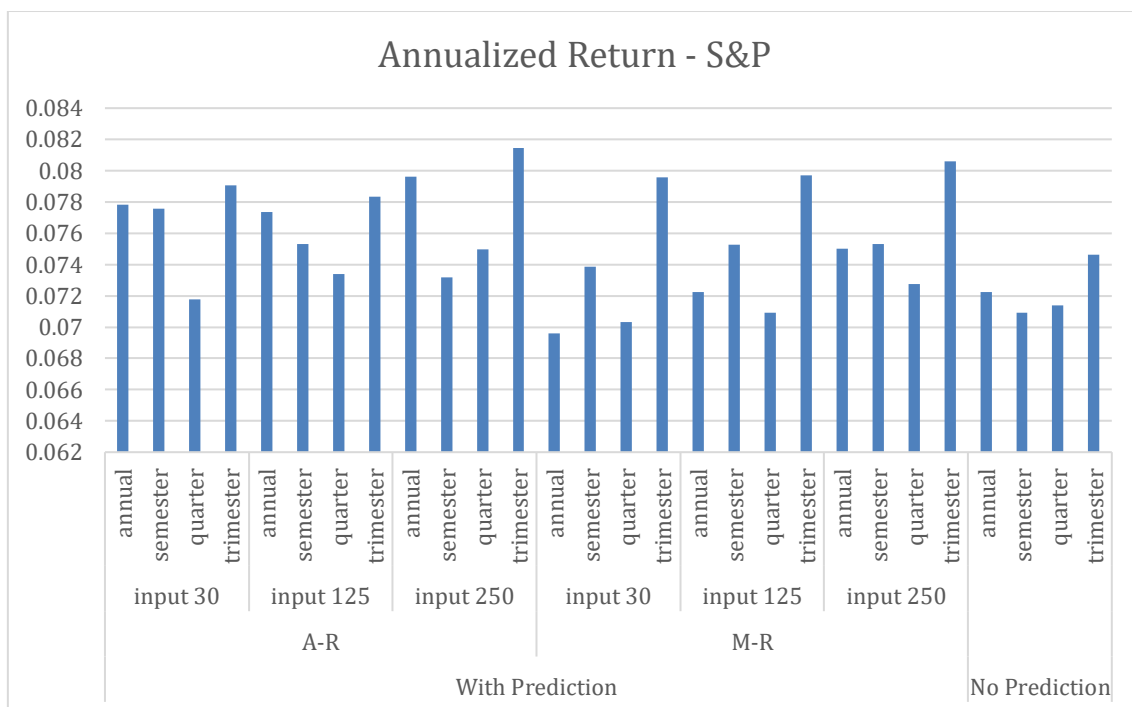


Figure 12: Annualized return of all models with S&P stocks

As seen in figure 12, there is a difference, although not much, between the returns from automatic (A-R) and manual (M-R) recurrence. Automatic recurrence models offered greater returns in general than manual recurrence models, which makes sense taking

into account the difference in prediction error they offer, as it was showcased in the previous subsection.

In addition, regarding the input size, a small trend exists where, the higher the input size, the higher the return. Considering the little difference that exists in terms of prediction error between them, it was expected this would be the case.

On the other hand, the timeframe did make a noticeable difference being trimester the best timeframe in terms of return in all type of models.

Lastly, regardless of the type of recurrence or input size, models that made use of the predicted prices to build their portfolios outperformed those that didn't, by as much as 1% increase in return.

A different parameter that can be used to measure the efficiency of each model is its risk (formula 3.13). This is shown in the following graph (figure 13).

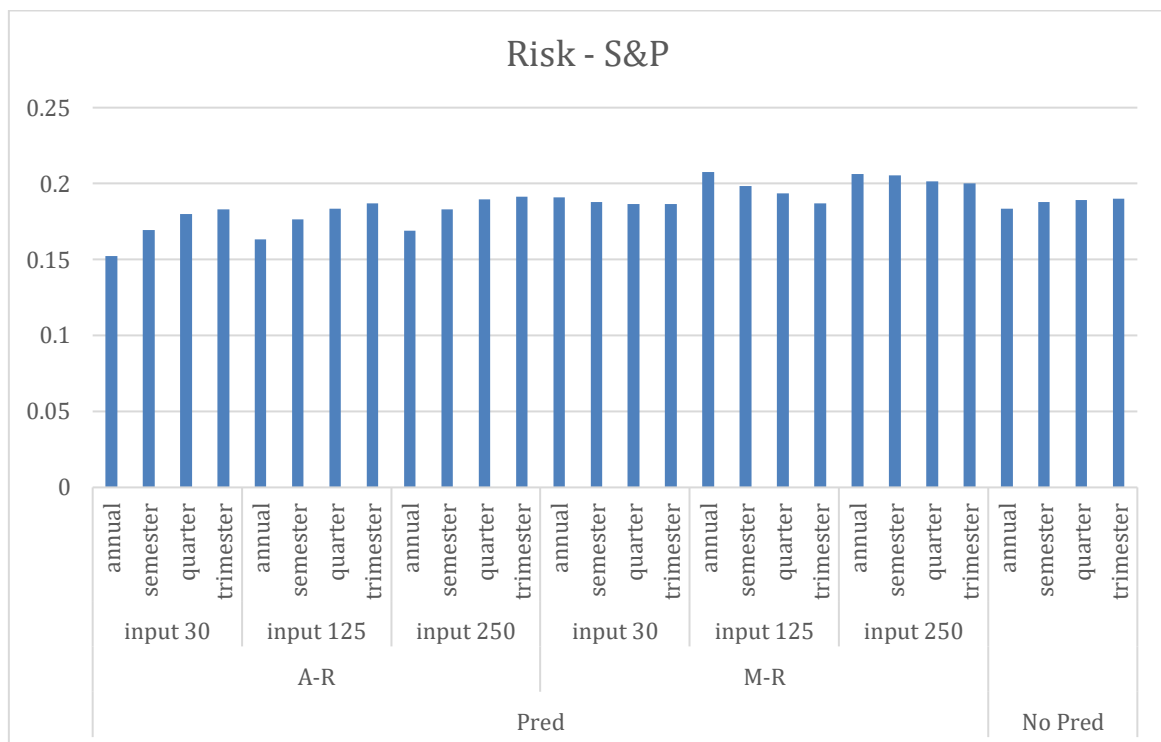


Figure 13: Risk of all models with S&P stocks

Similarly to the annualized return analysis, manual recurrence (M-R) has a worse performance as, in this case, all of its models have a greater risk than those that used automatic recurrence.

Then, regarding the input size and in contrast with the annualized return analysis, the greater the input size the greater the risk. Therefore, a priori, the smaller the input size the better in terms of risk.

On the other hand, regarding the timeframe, no clear pattern exists.

Lastly, regarding the use of predicted prices, most of the models that used automatic recurrence did outperform the risk of those models that didn't use predicted prices. Due to this, in the case they obtained a greater annualized return and a lower risk, some models would have dominated the models that didn't use predicted prices. To give a clear view of the models that did dominate, the next table has been created (table 3).

Table 3: Annualized return and risk of all models with S&P stocks

					Annualized Return	Risk
S&P	With Prediction	A-R	input 30	annual	0.07783467	0.1523261
				semester	0.038793024	0.1694183
				quarter	0.023924125	0.1798151
				trimester	0.019767161	0.1830173
			input 125	annual	0.077350471	0.1632004
				semester	0.037648508	0.1763213
				quarter	0.024465887	0.1834743
				trimester	0.01958507	0.1868375
			input 250	annual	0.079607249	0.1689503
				semester	0.036580752	0.1830743
				quarter	0.024994306	0.1897103
				trimester	0.020367186	0.1914629
		M-R	input 30	annual	0.069606884	0.1908478
				semester	0.036938019	0.1877874
				quarter	0.023439029	0.1863838
				trimester	0.019891529	0.1865941
			input 125	annual	0.072228158	0.2074429
				semester	0.03763088	0.1985469
				quarter	0.023633472	0.1936397
				trimester	0.019922058	0.1868596
			input 250	annual	0.075007367	0.2062864
				semester	0.037657623	0.2053097
				quarter	0.024245492	0.2015347
				trimester	0.020144702	0.2001865
		No Prediction		annual	0.072242166	0.1836608
				semester	0.035454903	0.1878943
				quarter	0.023793822	0.189245
				trimester	0.018656331	0.1901934

All models in bold in table 3 represent those that obtained a greater annualized return and a lower risk (domination) compared to the model that didn't use predicted prices and used the same timeframe. For instance, the first model shown which used predicted prices, automatic recurrence, an input size of 30 and an annual timeframe, is compared with the model that didn't use predicted prices and used an annual timeframe (timeframes must coincide).

As previously presented, automatic recurrence models had a better performance in both return and risk against manual recurrence models, which is reflected in table 3 (most models in bold are those that used automatic recurrence). Moreover, nearly all that used automatic recurrence are in bold, which shows a clear advantage against the models that didn't use predicted prices.

Now, jumping to the next measure, the next graph (figure 14) shows the total portfolio value (formula 3.13) achieved by every model.

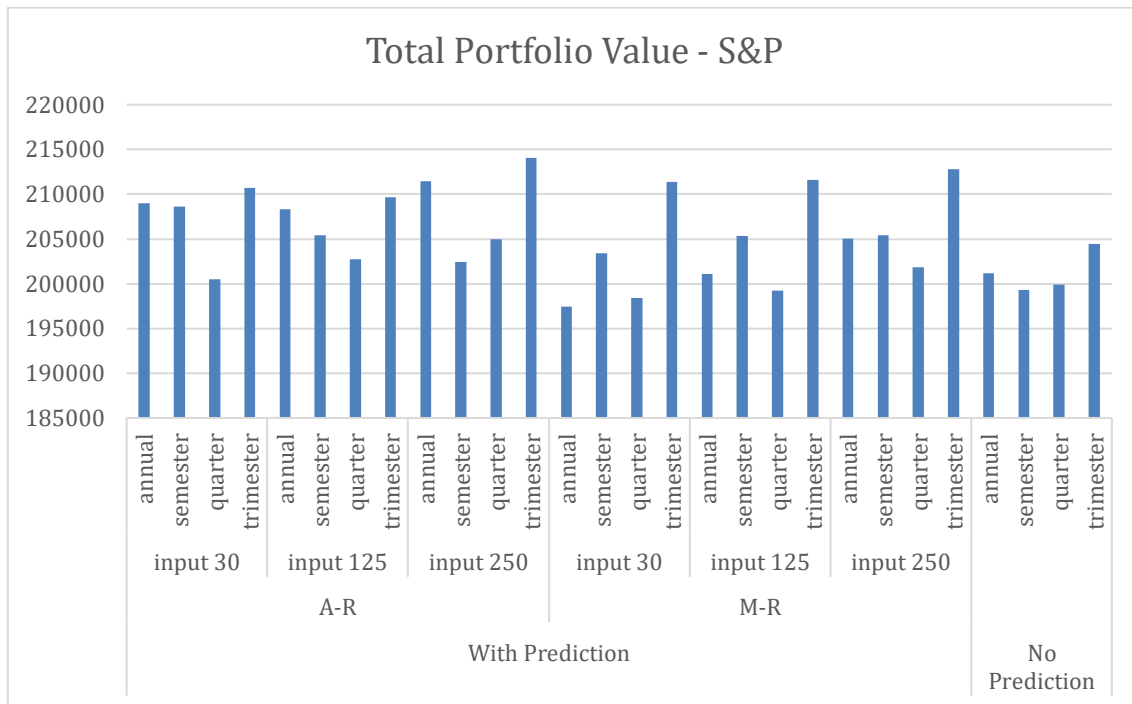


Figure 14: Total portfolio value of all models with S&P stocks

The patterns that occurred in the returns (figure 12) repeat in the case of the total portfolio values. It increases if automatic recurrence is used, a small increase exists as input size increases and the portfolio value is the highest when trimester is used as the timeframe. Plus, there is a positive difference of about 15.000€ between using and not using predicted prices.

Next, regarding the costs (formula 3.14), the results are shown in the following graph (figure 15).

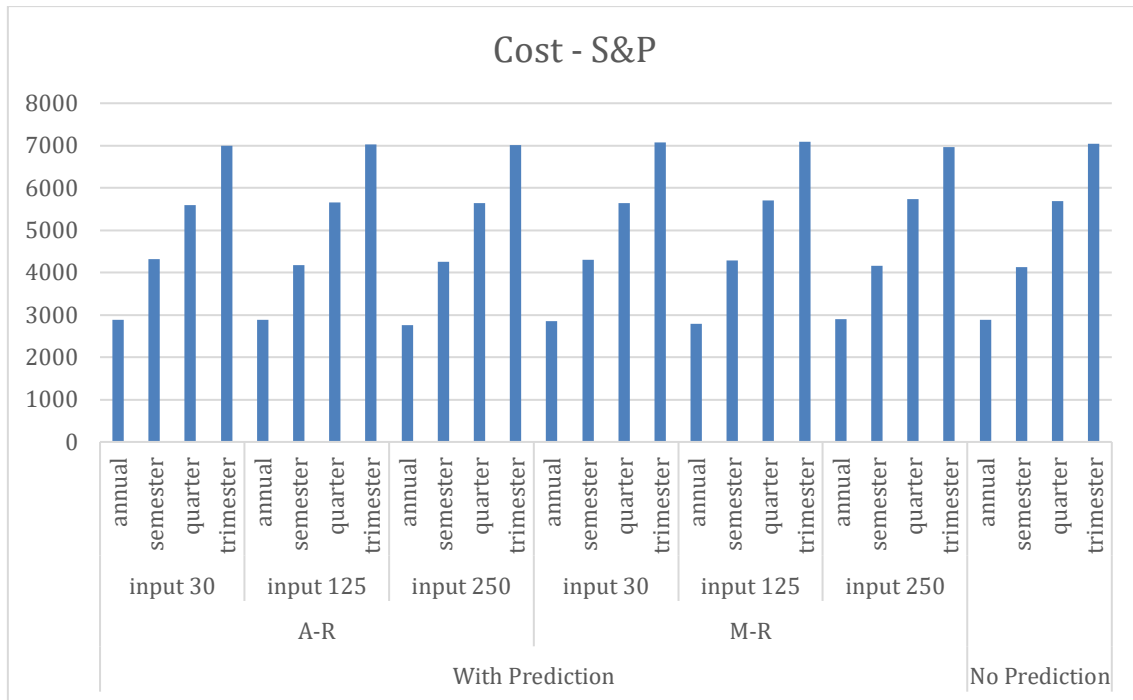


Figure 15: Cost of all models with S&P stocks

Figure 15 shows how costs increase as the timeframe decreases, due to the costs of buying and selling stocks in order to stick to the changes in the portfolio. Therefore, as there are more trimesters than quarters in a year, more quarters than semesters, and so on, the costs quickly add up. This results in an absolute difference of about 1500€ between timeframes (or roughly a 30% increase in costs). Therefore, in terms of cost, the preferred timeframe is annual, which contradicts with trimester achieving the highest annualized return.

To see the optimum model, the next graph (figure 16) showing the profit (formula 3.15) achieved by each model will be analyzed.

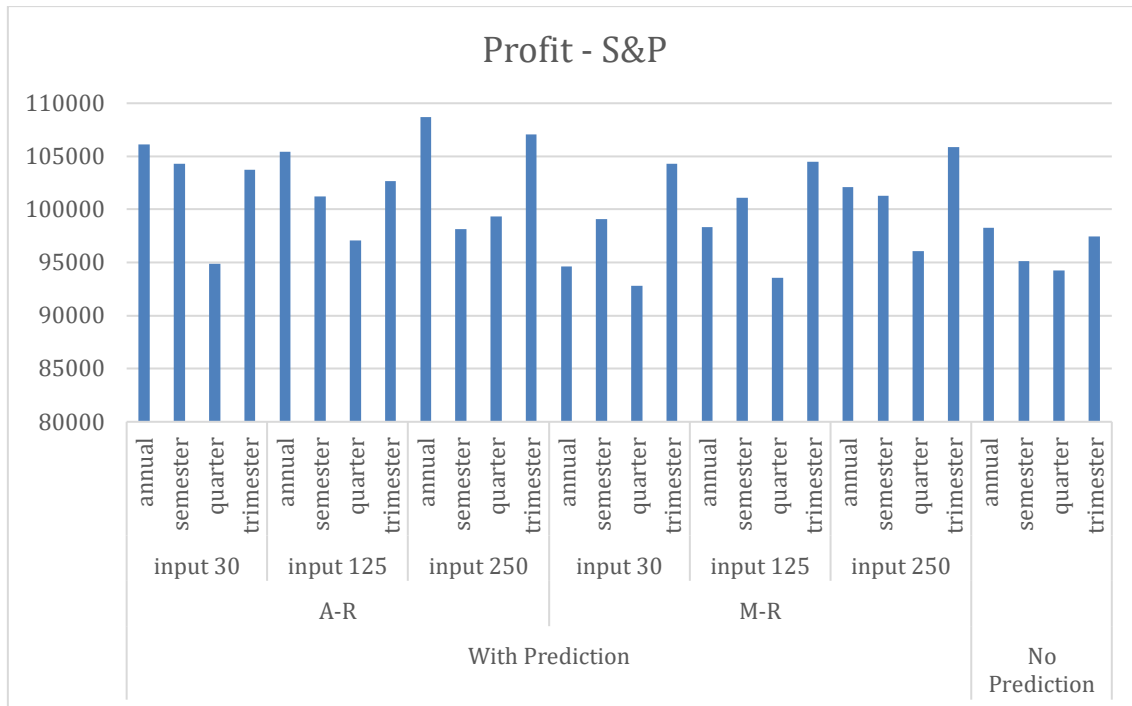


Figure 16: Profit of all models with S&P stocks

As it was just discussed, the trimester timeframe gained the highest return and total portfolio value but also the highest cost. This is all reflected in figure 16, as sometimes the annual timeframe earned the highest profit and in others, trimester did, even though it had the highest cost. However, it can be assured either annual or trimester always outperform a timeframe of a quarter or a semester.

Regarding recurrence, automatic recurrence outperformed manual recurrence as in previous measures. Input size made a small difference, increasing profit as input size increased, but not by a significant amount.

Lastly, all models which made use of predicted prices outperformed the equivalent models (in terms of the timeframe used) that didn't, by as much as 13.000€ or 13.4% increase in profit.

5.1.2 DAX

After the results of the S&P portfolios, it's the DAX portfolios created by the models turn. As in the S&P subsection, the first measure is the annualized return (formula 3.16) achieved by each model with the portfolios created for each timeframe between 2007-2020.

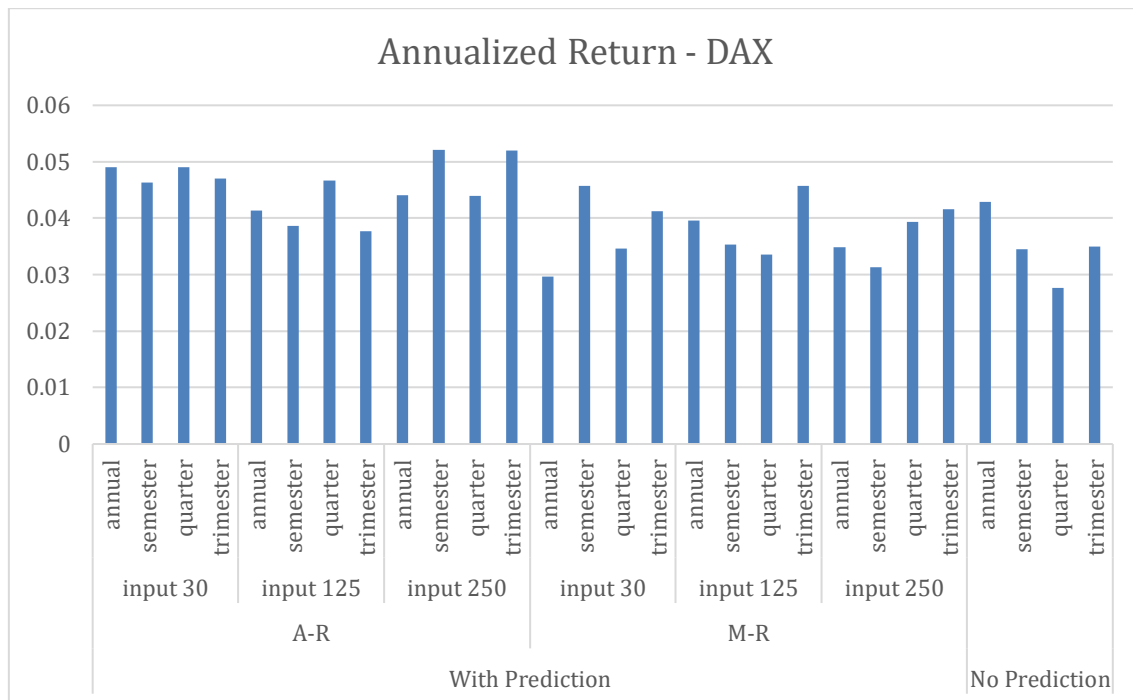


Figure 17: Annualized return of all models with DAX stocks

As seen in figure 17, there is a difference, although not much, between the returns from automatic (A-R) and manual (M-R) recurrence. Automatic recurrence models offered greater returns in general than manual recurrence models, which makes sense taking into account the difference in prediction error they offer, as it was showcased in the previous subsection.

In addition, regarding the input size, no trend can be extracted from the figure 17. The same occurs for the timeframe, none clearly outperformed others.

Lastly, regardless of the type of recurrence or input size, models that made use of the predicted prices to build their portfolios outperformed those that didn't, by as much as 1.5% increase in return.

A different parameter that can be used to measure the efficiency of each model is its risk (formula 3.13). This is shown in the following graph (figure 13).

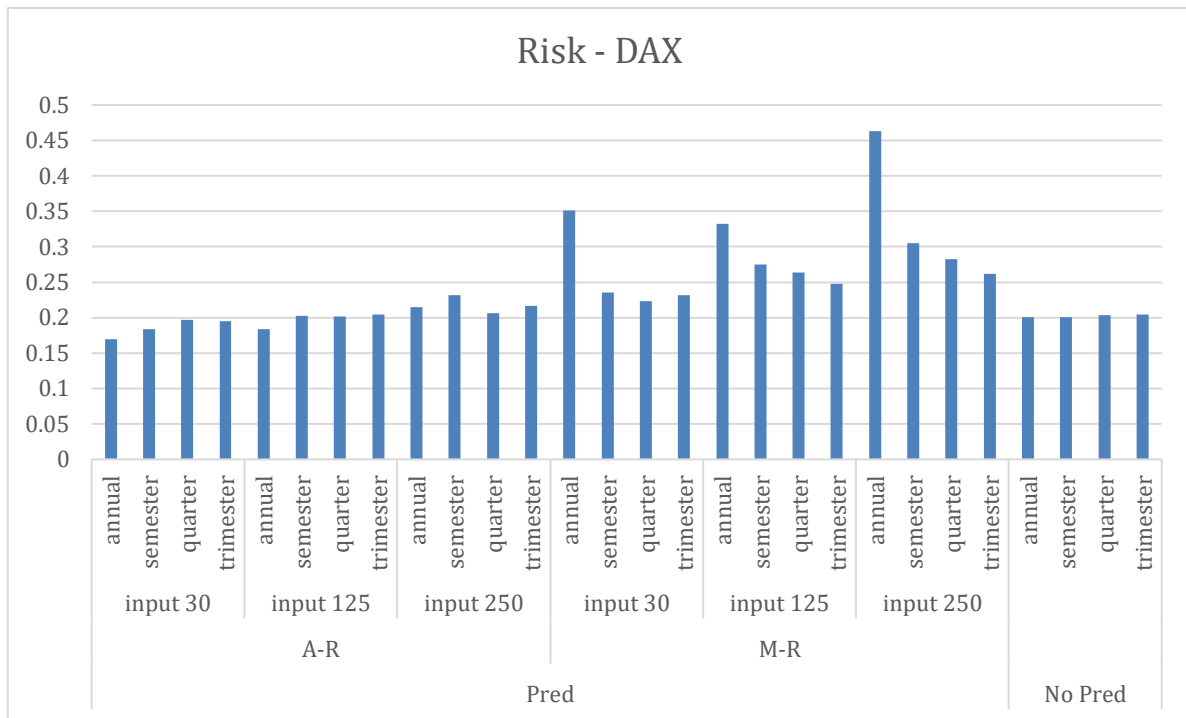


Figure 18: Risk of all models with DAX stocks

Similarly to the annualized return analysis, manual recurrence (M-R) has a worse performance as, in this case, all of its models have a greater risk than those that used automatic recurrence.

Then, regarding the input size and in contrast with the annualized return analysis, the greater the input size the greater the risk. Therefore, the smaller the input size the better in terms of risk. In addition, regarding the timeframe, no clear pattern exists.

Lastly, regarding the use of predicted prices, most of the models that used automatic recurrence did outperform the risk of those models that didn't use predicted prices. Due to this, in the case they obtained a greater annualized return and a lower risk, some models would have dominated the models that didn't use predicted prices. To give a clear view of the models that did dominate, the next table has been created (table 4).

Table 4: Annualized return and risk of all models with DAX stocks

					Annualized Return	Risk
DAX30	With Prediction	A-R	input 30	annual	0.049092265	0.1697505
				semester	0.02313339	0.1838946
				quarter	0.016337986	0.197107
				trimester	0.011747539	0.1947242
			input 125	annual	0.041389103	0.1838216
				semester	0.019332702	0.2020818
				quarter	0.015569626	0.2013507
				trimester	0.009435447	0.2040891
			input 250	annual	0.044122489	0.2143149
				semester	0.026035915	0.2315566
				quarter	0.014636457	0.2064211
				trimester	0.012990501	0.2170274
		M-R	input 30	annual	0.029676903	0.350883
				semester	0.022881851	0.235261
				quarter	0.011537603	0.2232184
				trimester	0.010311101	0.2320448
			input 125	annual	0.039606215	0.3324145
				semester	0.017672658	0.2751982
				quarter	0.011181158	0.2640484
				trimester	0.011423367	0.2474878
			input 250	annual	0.034870859	0.4630704
				semester	0.015671156	0.3051829
				quarter	0.013125764	0.2825801
				trimester	0.01039036	0.2618276
		No Prediction		annual	0.042906429	0.2006217
				semester	0.017274021	0.2002615
				quarter	0.00922284	0.2033291
				trimester	0.008749687	0.2041693

All models in bold in table 4 represent those that obtained a greater annualized return and a lower risk (domination) compared to the model that didn't use predicted prices and used the same timeframe.

As previously presented, automatic recurrence models had a better performance in both return and risk against manual recurrence models, which is reflected in table 4 (all models in bold are those that used automatic recurrence). Moreover, most that used automatic recurrence are in bold, which shows a clear advantage against the models that didn't use predicted prices.

Now, jumping to the next measure, the next graph (figure 19) shows the total portfolio value (formula 3.13) achieved by every model.

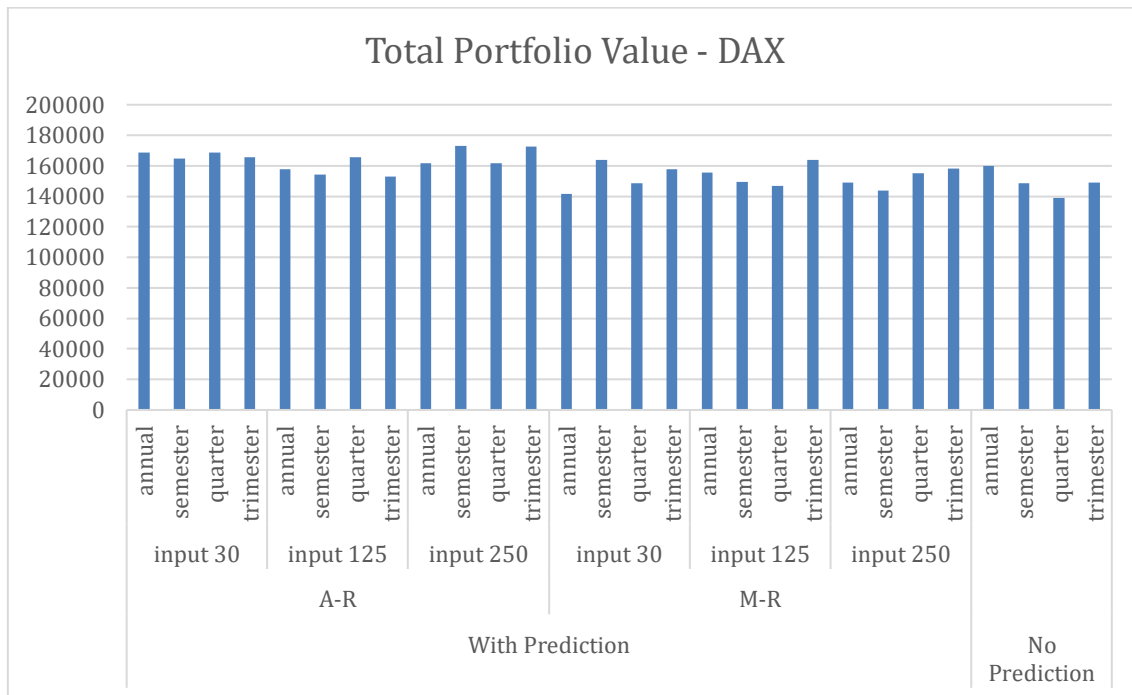


Figure 19: Total portfolio value of all models with DAX stocks

The patterns that occurred in the returns (figure 17) repeat in the case of the total portfolio values. It increases if automatic recurrence is used and there is no significant difference between input sizes as well as for the timeframe used. Plus, there is a positive difference, between using and not using predicted prices.

Next, regarding the costs (formula 3.14), the results are shown in the following graph (figure 20).

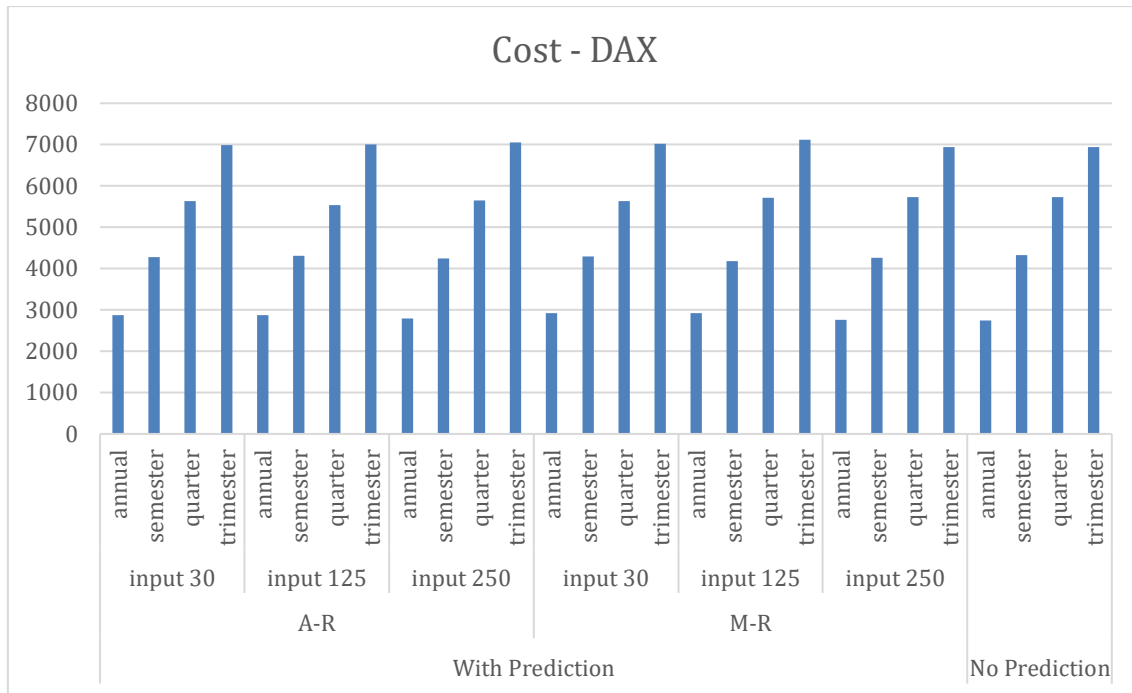


Figure 20: Cost of all models with DAX stocks

Figure 20 shows how costs increase as the timeframe decreases, due to the costs of buying and selling stocks in order to stick to the changes in the portfolio. Therefore, as there are more trimesters than quarters per year, more quarters than semesters, and so on, the costs quickly add up. This results in an absolute difference of about 1500€ between timeframes (or roughly a 20-30% increase in costs). Therefore, in terms of cost, the preferred timeframe is annual.

The next graph (figure 21) to be analyzed is the profit (formula 3.15) achieved by each model.

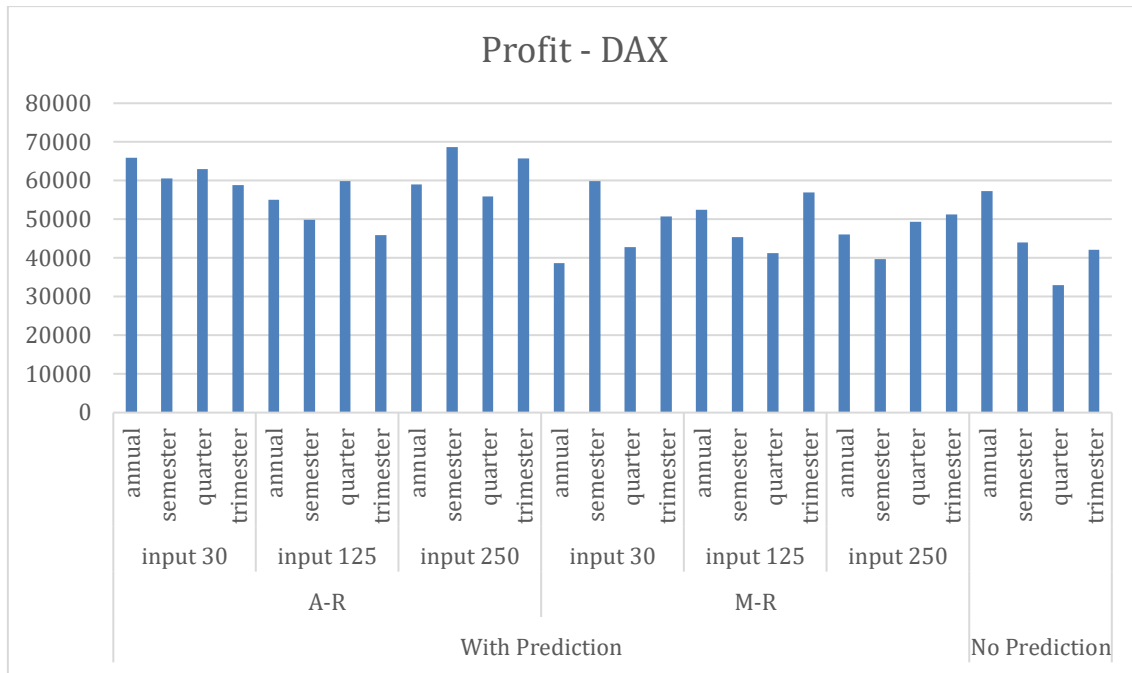


Figure 21: Profit of all models with DAX stocks

As it was previously analyzed, automatic recurrence outperforms manual recurrence as in previous measures, whilst all other variables made no difference.

Lastly, all models which made use of predicted prices outperformed the equivalent models (in terms of the timeframe used) that didn't, by as much as 25.000€ or 50% increase in profit.

5.1.3 DJI

Lastly, the analysis of the performance achieved by the different portfolios created by the models from DJI index stocks is presented.

As in the previous subsections, the first measure is the annualized return (formula 3.16) achieved by each model with the portfolios created for each timeframe between 2007-2020.

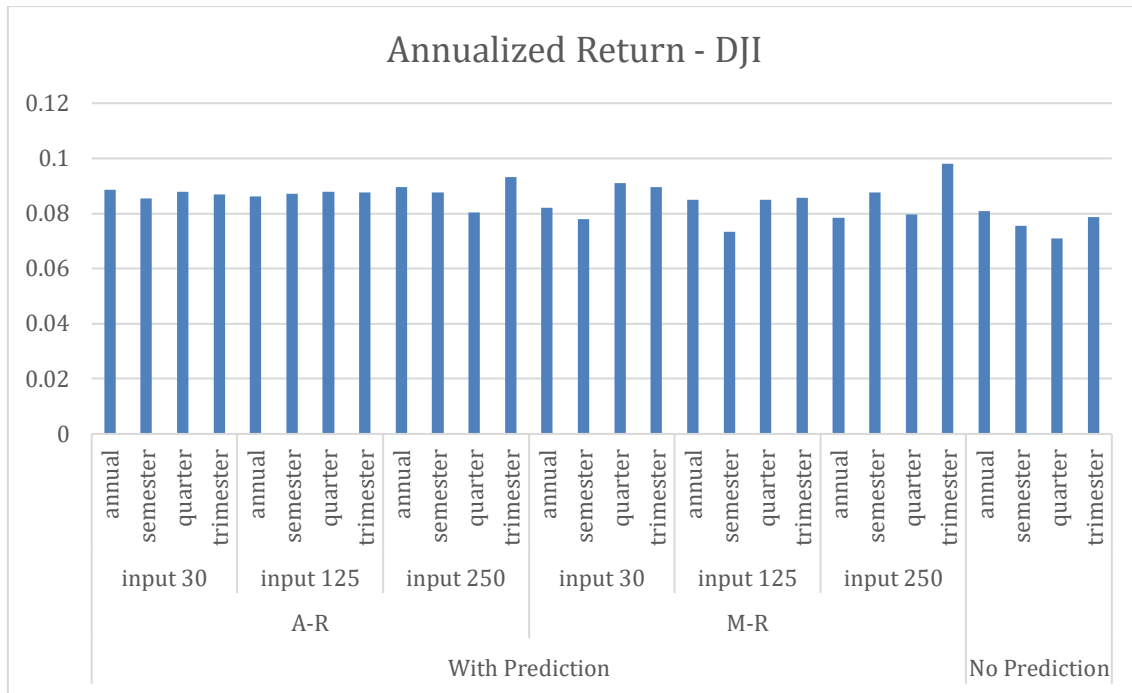


Figure 22: Annualized return of all models with DJI stocks

Similarly to S&P and DAX returns, as seen in figure 22, there is a difference between the returns from automatic and manual recurrence. Automatic recurrence models offered greater returns in general than manual recurrence models. In addition, regarding the input size, no clear trend exists. On the other hand, the timeframe did make a noticeable difference being the trimester timeframe usually the best in terms of annualized return, with a difference of nearly 2% against the other timeframes while keeping all other variables constant (input and recurrence).

Lastly, regardless of the type of recurrence or input size, models that made use of the predicted prices to build their portfolios outperformed those that didn't, although for as much as 2% increase in return in equivalent models (equivalent timeframes).

A different parameter that can be used to measure the efficiency of each model is its risk (formula 3.13). This is shown in the following graph (figure 23).

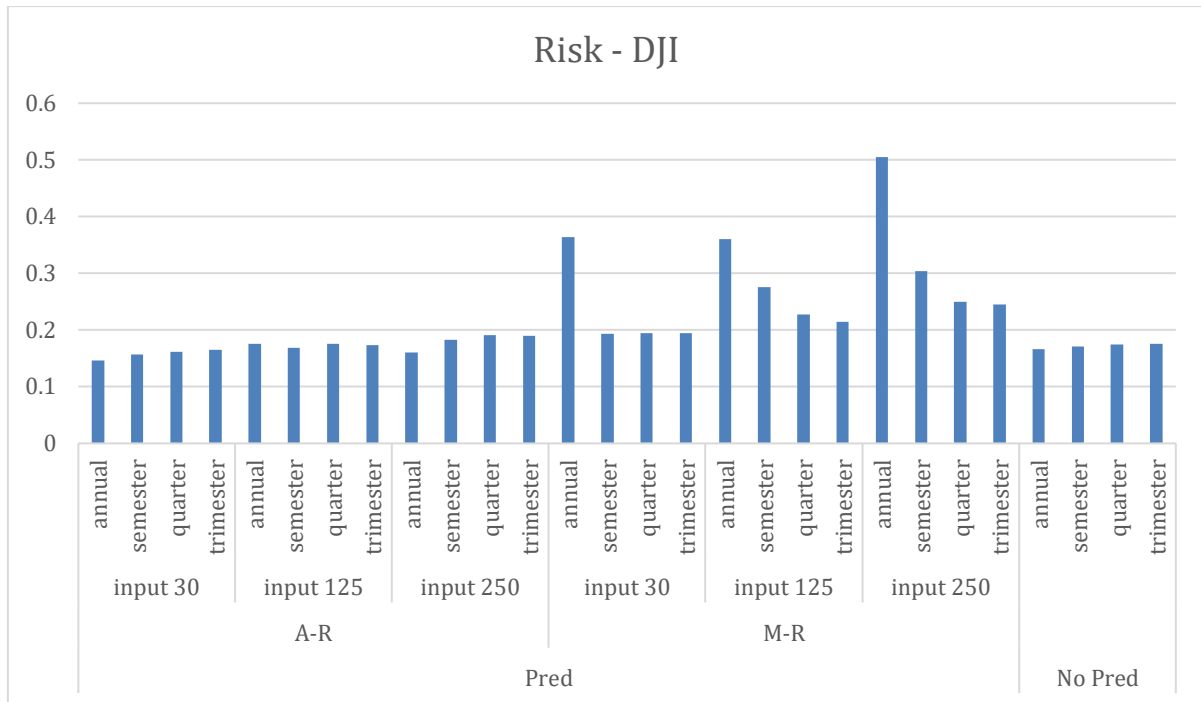


Figure 23: Risk of all models with DJI stocks

Similarly to the annualized return analysis, manual recurrence (M-R) has a worse performance as, in this case, all its models have a greater risk than those that used automatic recurrence.

Then, regarding the input size and in contrast with the annualized return analysis, the greater the input size the greater the risk. Therefore, the smaller the input size the better in terms of risk. In addition, regarding the timeframe, no clear pattern exists.

Lastly, regarding the use of predicted prices, most of the models that used automatic recurrence did outperform the risk of those models that didn't use predicted prices. Due to this, in the case they obtained a greater annualized return and a lower risk, some models would have dominated the models that didn't use predicted prices. To give a clear view of the models that did dominate, the next table has been created (table 5).

Table 5: Annualized return and risk of all models with DJI stocks

					Annualized Return	Risk
DJI	With Prediction	A-R	input 30	annual	0.088568197	0.1461364
				semester	0.042722509	0.1564152
				quarter	0.029312635	0.1613124
				trimester	0.021711505	0.165261
			input 125	annual	0.086240621	0.1754588
				semester	0.043536973	0.1682106
				quarter	0.029302315	0.175214
				trimester	0.021879221	0.1736699
			input 250	annual	0.089586331	0.1605535
				semester	0.043769301	0.1830803
				quarter	0.026798772	0.1911061
				trimester	0.023307422	0.1901311
		M-R	input 30	annual	0.081970852	0.3641321
				semester	0.038943894	0.193862
				quarter	0.030358712	0.193942
				trimester	0.022415924	0.1948734
			input 125	annual	0.084949704	0.3607711
				semester	0.036678935	0.2750653
				quarter	0.028347541	0.2279211
				trimester	0.021428676	0.2139558
			input 250	annual	0.078546344	0.5044427
				semester	0.043775319	0.3033572
				quarter	0.02654207	0.2501537
				trimester	0.02449506	0.2445389
		No Prediction		annual	0.080857806	0.1663081
				semester	0.037811364	0.1705075
				quarter	0.023639636	0.174344
				trimester	0.019643074	0.1756817

All models in bold in table 5 represent those that obtained a greater annualized return and a lower risk (domination) compared to the model that didn't use predicted prices and used the same timeframe.

As previously presented, automatic recurrence models had a better performance in both return and risk against manual recurrence models, which is reflected in table 5 (all

models in bold are those that used automatic recurrence). Moreover, most that used automatic recurrence are in bold, which shows a clear advantage against the models that didn't use predicted prices.

Now, jumping to the next measure, the next graph (figure 24) shows the total portfolio value (formula 3.13) achieved by every model.

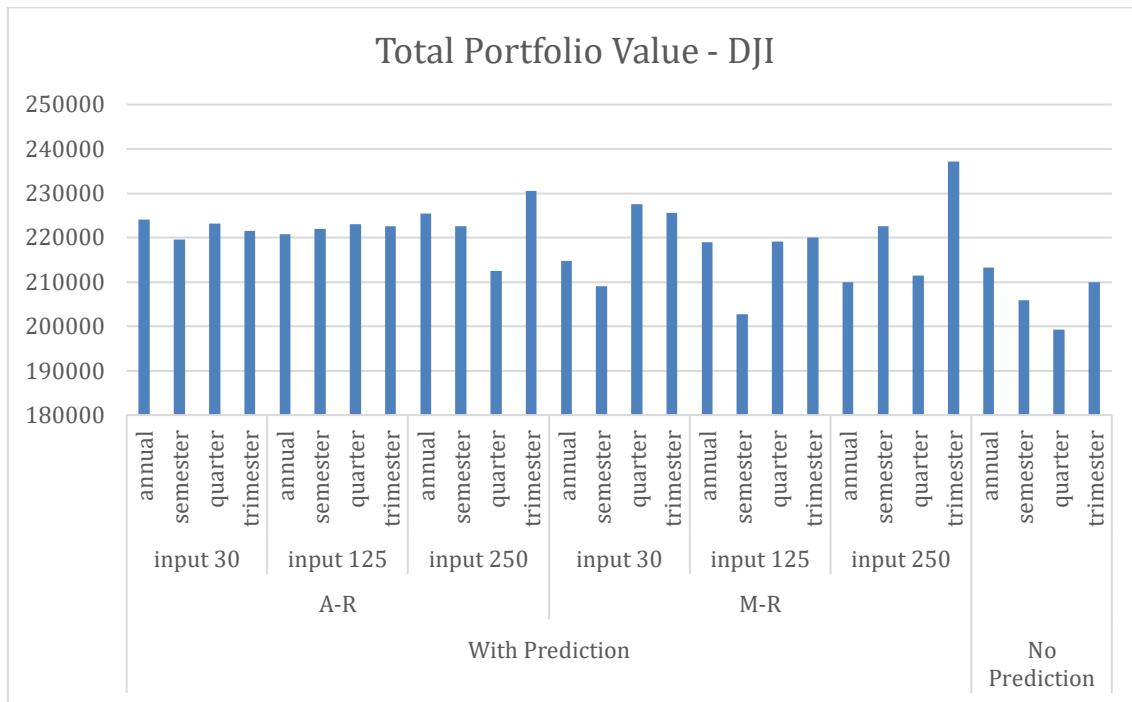


Figure 24: Total portfolio value of all models with DJI stocks

The patterns that occurred in the annualized returns (figure 22) repeat in the case of the total portfolio values. It increases if automatic recurrence is used, there is no significant difference between input sizes and the portfolio value is usually the highest when using trimester as the timeframe. Plus, there is a positive noticeable difference between using and not using predicted prices.

Next, regarding the costs (formula 3.14), the results are shown in the following graph (figure 25).

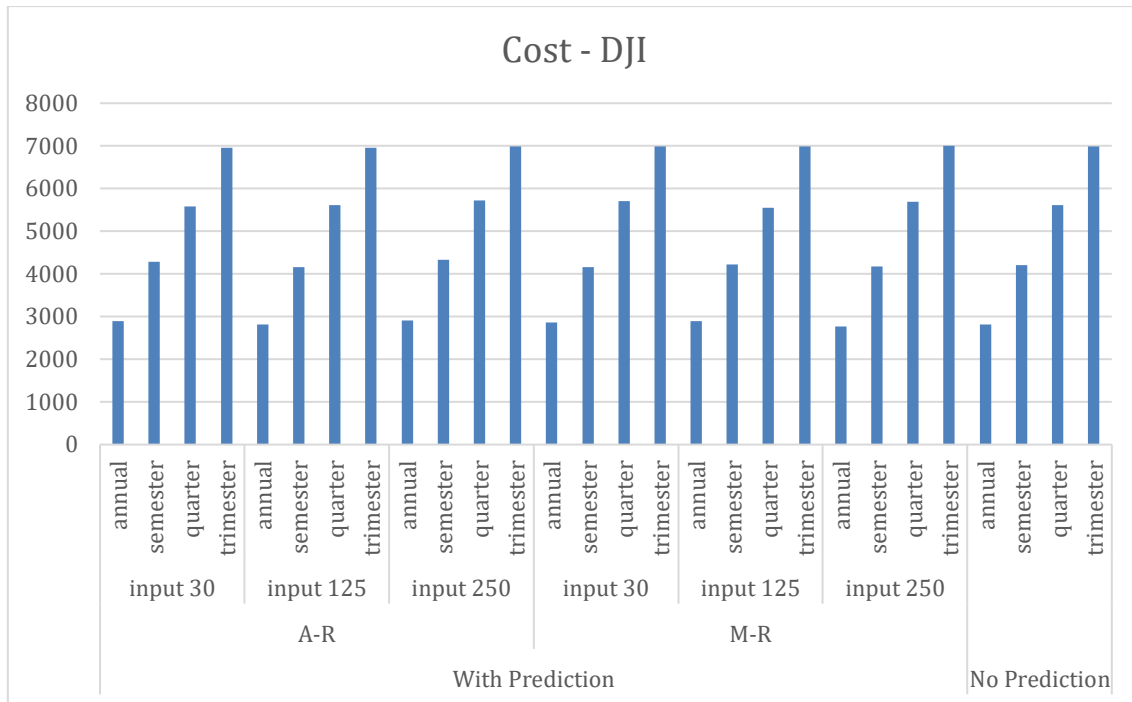


Figure 25: Cost of all models with DJI stocks

Figure 25 shows how costs increase as the timeframe decreases, due to the costs of buying and selling stocks in order to stick to the changes in the portfolio. There is an absolute difference of about 1500€ between timeframes (or roughly a 20-30% increase in costs). Therefore, in terms of cost, the preferred timeframe is annual, which contradicts with trimester achieving the highest annualized return.

To see the optimum model, the next graph (figure 26) showing the profit (formula 3.15) achieved by each model will be analyzed.

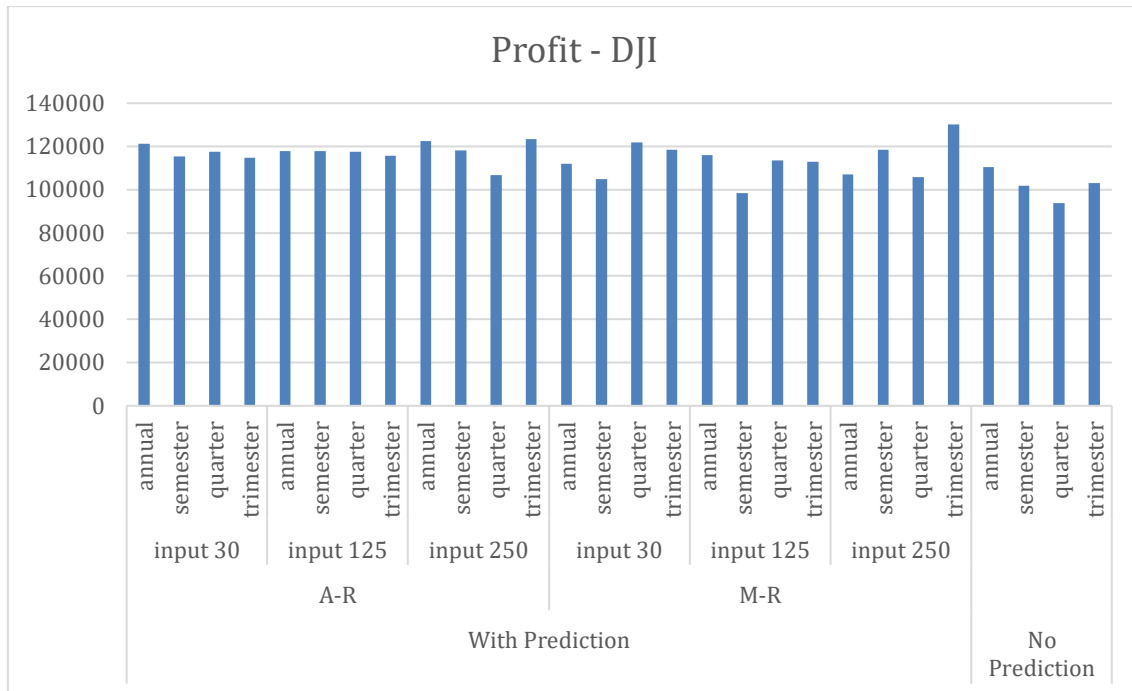


Figure 26: Profit of all models with DJI stocks

As it was just discussed, the trimester timeframe gained the highest return and total portfolio value but also the highest cost. This is all reflected in figure 18, as the timeframe with the highest profit varies, there is no clear outperformer.

Regarding recurrence, automatic recurrence outperformed manual recurrence as in previous measures. Input size made a small difference, increasing profit as input size increased, but not by a significant amount.

Lastly, all models which made use of predicted prices outperformed the equivalent models (in terms of the timeframe used) that didn't, by as much as 25.000€ or 23.8% increase in profit.

5.1.4 Cross evaluation

Now all three indexes have been analyzed, we can see some patterns between all of them.

Starting with the timeframe, it is clear the larger the timeframe, the lower the costs (due to less portfolio changes). However, the timeframe that usually gained the highest

annualized return is trimester, but not by much. As a result, no timeframe outperformed others in the amount of profit achieved. However, if costs could be reduced, this might change, and the trimester timeframe might be able to outperform the others.

Then, regarding the recurrence, automatic recurrence models outperformed manual recurrence in all three indexes, as it achieved a higher profit in all, due to the significant difference in the error when predicting prices, as it was analyzed in subsection 5.1.1. Moreover, most automatic recurrence models dominated the models that did not use predicted prices (higher annualized return and lower risk).

The input size didn't make a significant difference between models, although a small trend was seen when analyzing the results of the DAX index where, the higher the input size, the higher the return. But again, this is not enough to assume more input data equals a higher return, total portfolio value or profit.

Lastly, models that made use of predicted prices improved their performance in all three indexes. As previously shown, the profit difference was a maximum of 13.4% for S&P, 50% for DAX and 23.8% for DJI, when comparing equal models in terms of the timeframe used. Therefore, it can be concluded predicted prices had a big positive impact on the performance of the models when used to construct portfolios using the Markowitz approach, not only in terms of profit, but in absolute performance, as many times models that did use predicted prices dominated those that didn't.

5.2 Stock price prediction models analysis

In order to analyze the performance of each model used to predict stock prices, both error of predictions and time to train the model will be considered. One thing to note is, the results of this subsection are not differentiated between indexes. This means, when calculating the error of a model, all predictions made for each index are considered. The same for time, it is the total time to train and execute the model to predict values for all three indexes.

5.2.1 Error

To start with, error will be analyzed. As discussed in section three, methodology, three different error measures were used. Their results are presented next.

1. Mean absolute error (MAE)

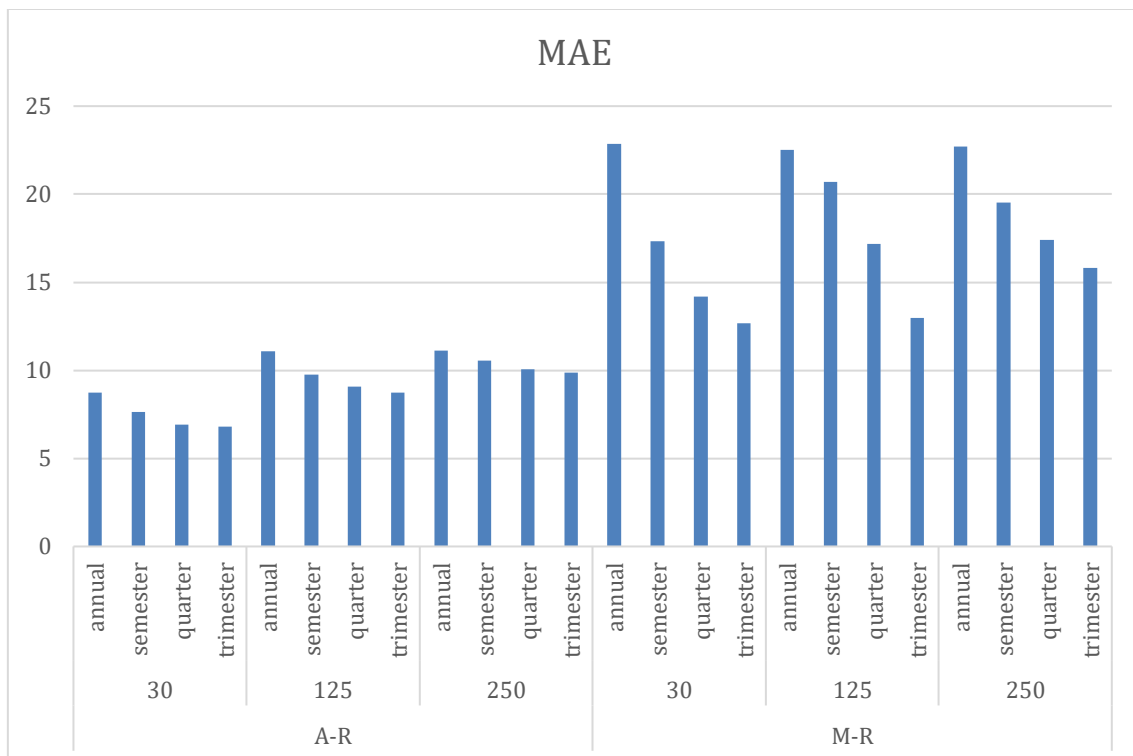


Figure 27: MAE of all models

As seen in Figure 27, automatic recurrence models have a smaller MAE than manual recurrence models.

Also, the higher the number of input data used to train, the higher the MAE in general, which could at first be counter intuitive but, the greater the amount of input data, the least the relation there'll be with the next timeframe. However, the relation between less input data equals less error will only be true until a certain amount of input data, although no tests were carried out with less input data than thirty.

Lastly, another trend can be appreciated with the timeframe predicted. The greater the timeframe is, the greater the MAE is in general, as expected.

2. Mean absolute percentage error (MAPE)

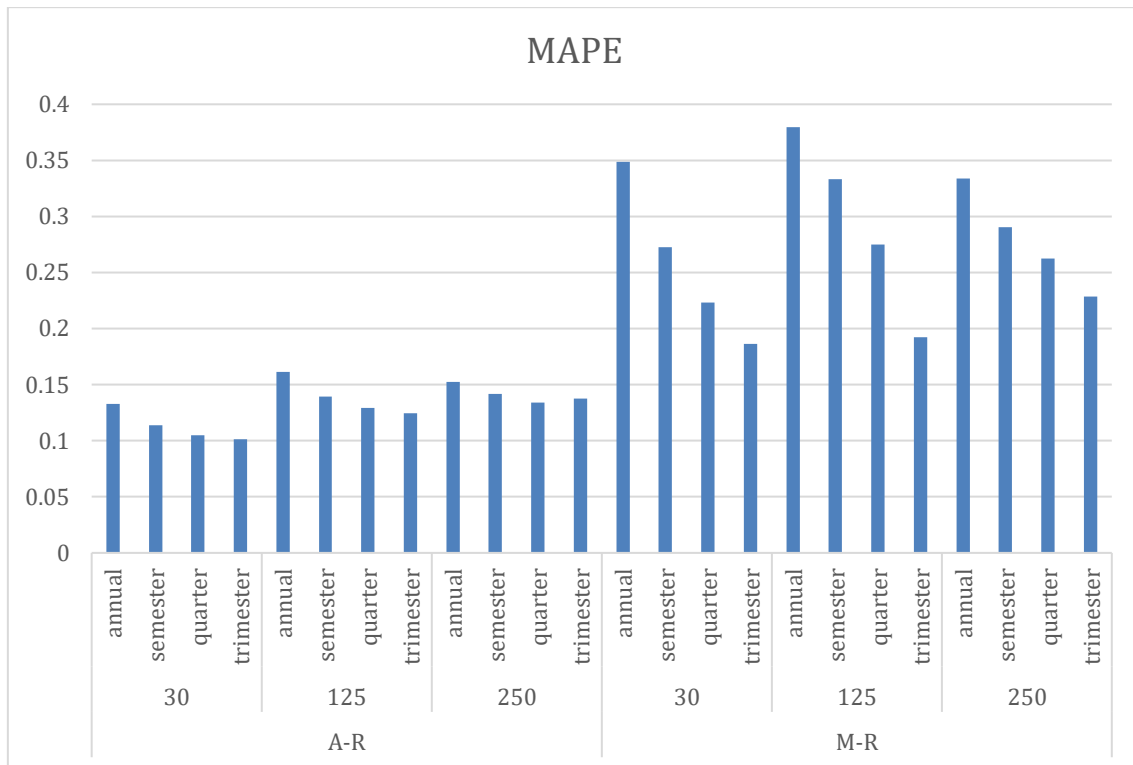


Figure 28: MAPE of all models

In the case of MAPE, as seen in figure 28, the same conclusions can be extracted as for MAE (figure 27). Automatic recurrence models have a smaller MAE than manual recurrence models. Also, the higher the number of input data used to train, the higher the MAE in general. Lastly, another trend can be appreciated with the timeframe predicted. The greater the timeframe is, the greater the MAE is in general.

3. Root-mean-square error (RMSE)

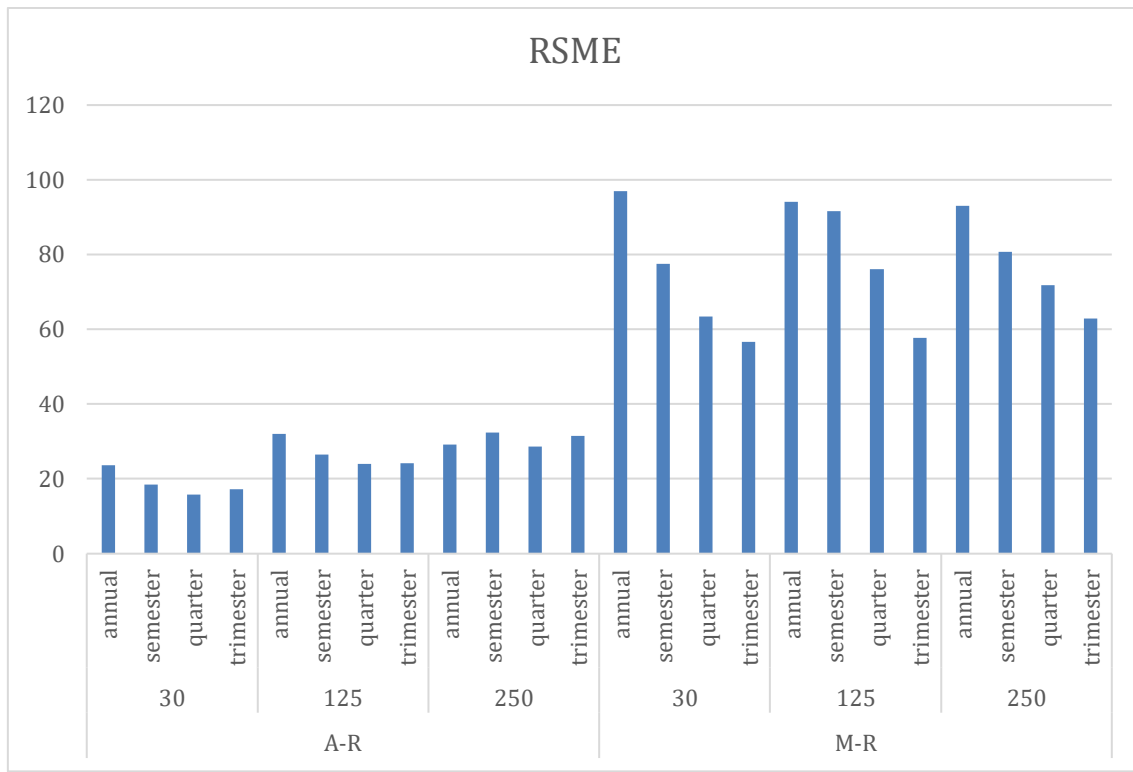


Figure 29: RSME of all models

In the case of RSME (figure 29), the same trends repeat as in MAE (figure 27) and MAPE (figure 28). However, in this case the difference between manual and automatic recurrence is more drastic.

5.2.2 Time

Another indicator used in order to differentiate between models is time taken to train and make predictions. The next graph (figure 30) shows the time taken in hours to train and execute each model.

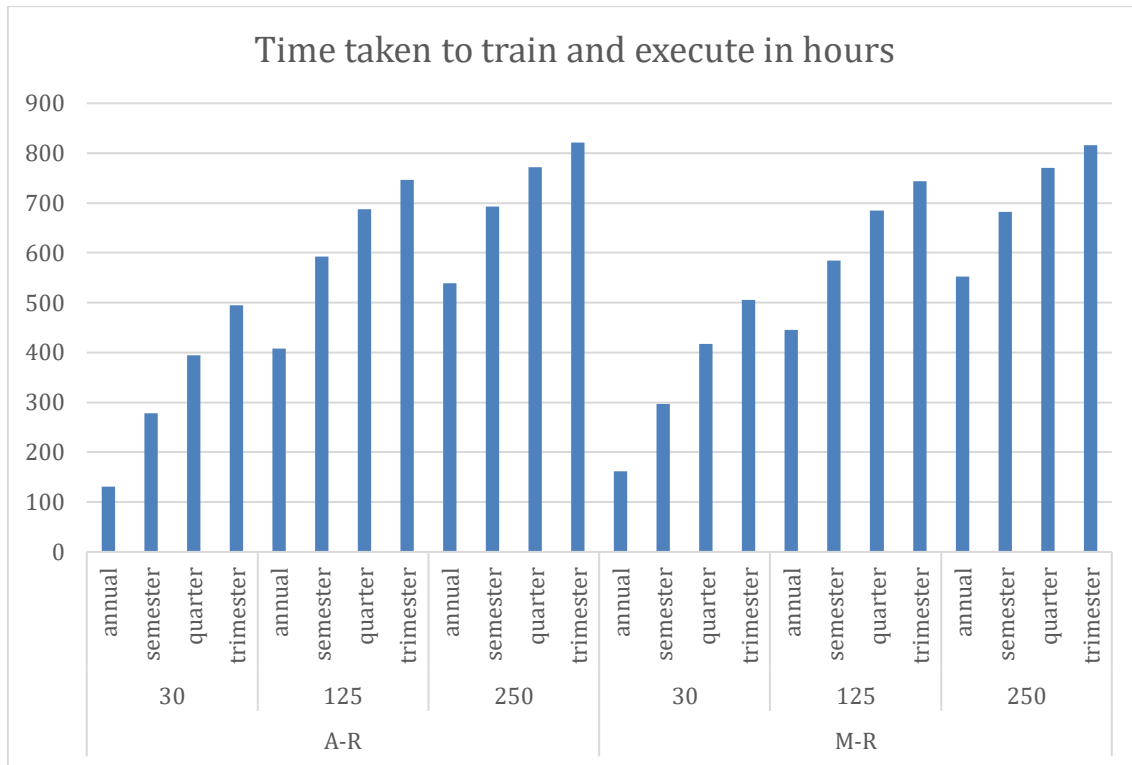


Figure 30: Time taken to train and execute their predictions

As seen in figure 30, no significant difference exists between manual and automatic recurrence in time to train and execute the models. However, a trend does exist for input data and the timeframe predicted. In the case of the size of the input data, the more input data, the higher the time to train the model. This seems intuitive as the model needs to consider more datapoints. On the other hand, as the timeframe increases, the time decreases.

To sum up, even though manual recurrence was included because it was thought the time to train would decrease, it was not the case. Plus, given the error drastically decreases using all three measures (MAE, MAPE and RSME), it can be concluded

automatic recurrence is preferred. In the case of the amount of input data, the error increases the more input data there is, as well as the time to train. This means the optimal input data, based on our results, is thirty. Lastly, regarding the timeframe predicted, the error of the predictions increases as the timeframe increases but, the time taken to train and execute decreases. Therefore, depending on what's valued the highest, error or time, a timeframe or another is preferred, but a cost of opportunity will always exist.

However, the performance of the models doesn't reduce to error and time. Now, the most important indicators of the portfolios created by each model are analyzed, their average return between 2007-2020 and their profit.

5.3 Portfolios against index exchange-traded funds

After the different models have been compared between each other, it is time to compare them with the annualized return of each index as, this type of investment could be easily accessible for an investor by investing in an exchange-traded fund that mimics each index.

5.3.1 S&P

During the 2007-2020 period, the S&P500 index achieved an annualized return of 8.58% (Webster, 2021). In order to compare this return with the annualized return of the models, the following graph (figure 31) has been done.

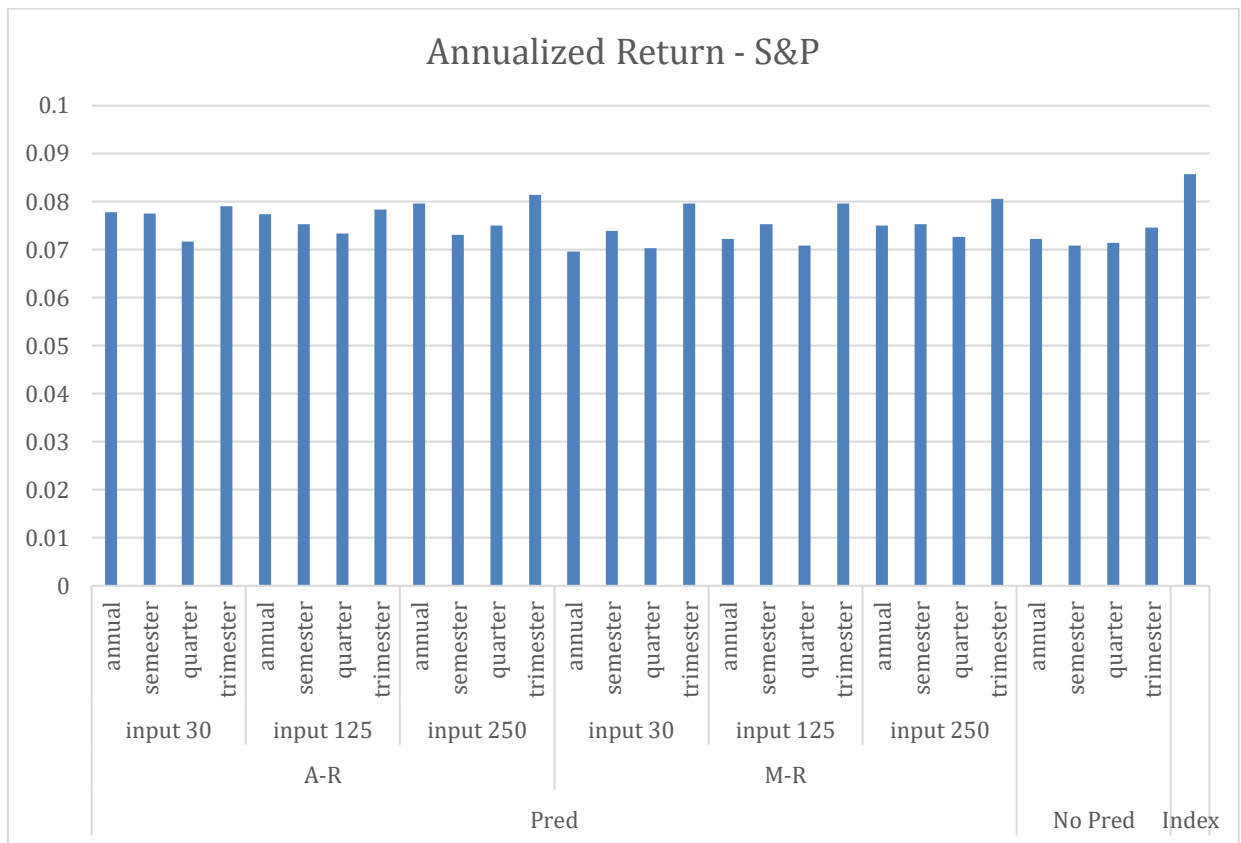


Figure 31: Annualized return achieved by models using S&P stocks and the S&P index during 2007-2020

As seen in figure 31, the S&P index achieved a higher return than what the models were able to achieve. The best model, which timeframe is trimester, input is 250, used automatic recurrence and used predicted prices, was able to return 8.2%. Therefore, given the S&P index returned 8.58% during the same period, it beat the best model by 0.38%. Even though it is not much, this means we can't assure applying the model using predicted prices is better than buying an exchange-traded fund that mimics the S&P index.

Also, regarding the risk (formula 3.2, standard deviation in this case), the S&P index during the 2007-2020 had a risk associated of 15.9% (Curvo, 2022). This is reflected in the next graph (figure 32) to easily compare with the risk of the models created in this thesis.

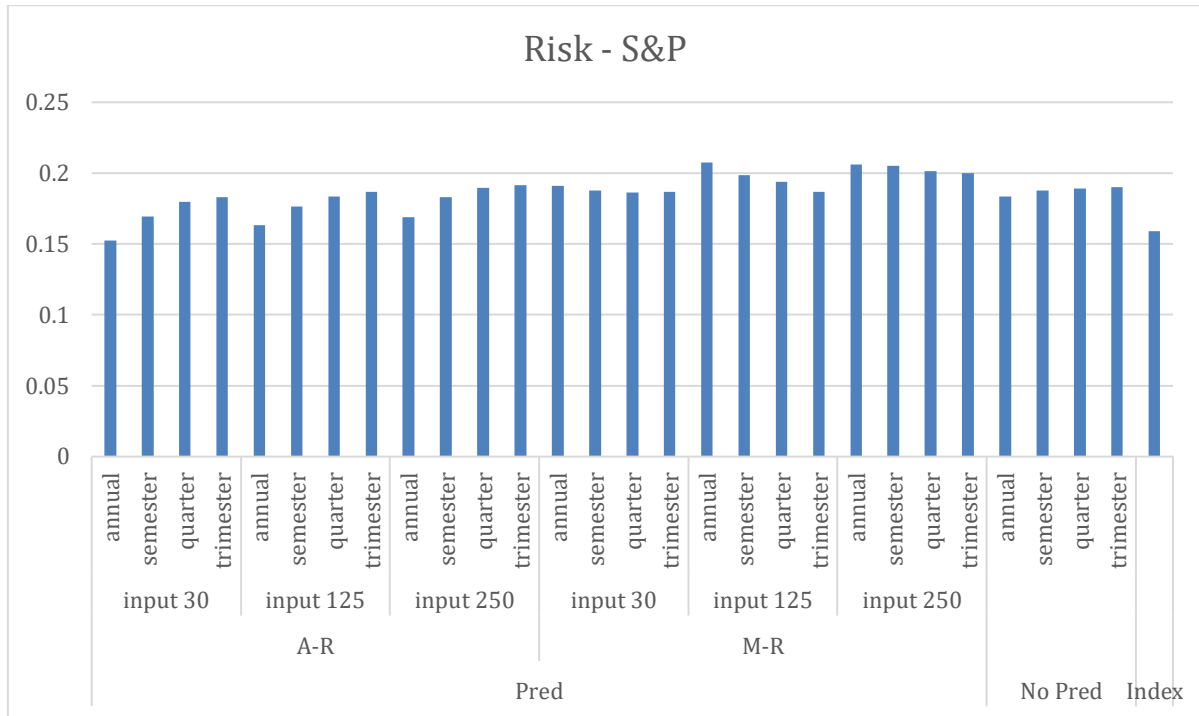


Figure 32: Risk achieved by models using S&P stocks and the S&P index during 2007-2020

As seen in figure 32, the risk of the index is lower than all model's risk. This means, the index dominated all models (higher return and lower risk), showing it was clearly superior.

5.3.2 DAX

Moving on to DAX, its return during the period 2007-2020 was 7.46% (1stock1, 2020). Again, to compare with the annualized returns achieved by the models that only used DAX stocks, the following graph has been done (figure 33).

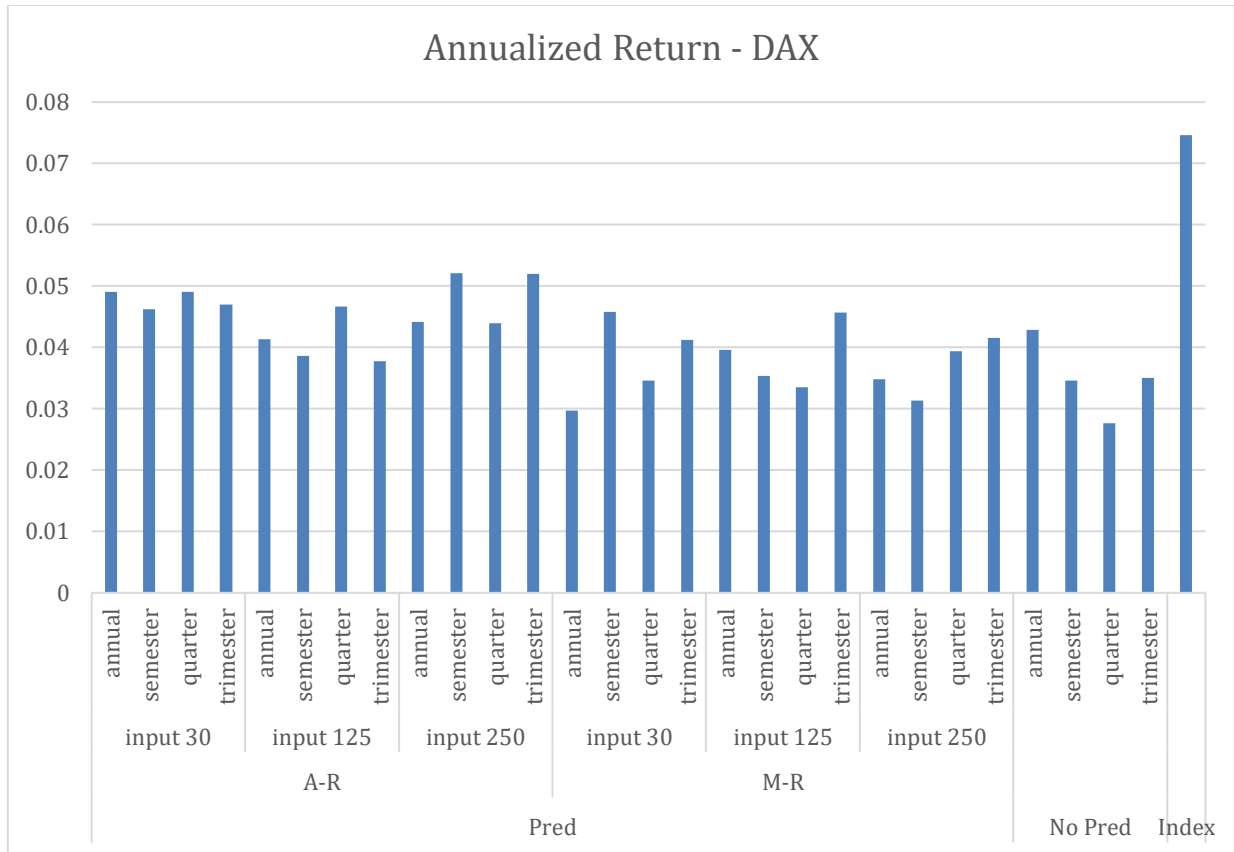


Figure 33: Annualized return achieved by models using DAX stocks and the DAX index during 2007-2020

As seen in figure 33, the DAX index achieved a much higher return than what the models were able to achieve. The best model, which timeframe is trimester, input is 250, used automatic recurrence and used predicted prices, was able to return 5.2%. Therefore, given the DAX index returned 7.46% during the same period, it beat the best model by 2.26%. In this case, the difference is noticeable.

Next, regarding the risk (formula 3.2, standard deviation in this case), the DAX index during the 2007-2020 had a risk associated of 20.11% (Morningstar, 2022). This is reflected in the next graph (figure 34) to easily compare with the risk of the models created in this thesis.

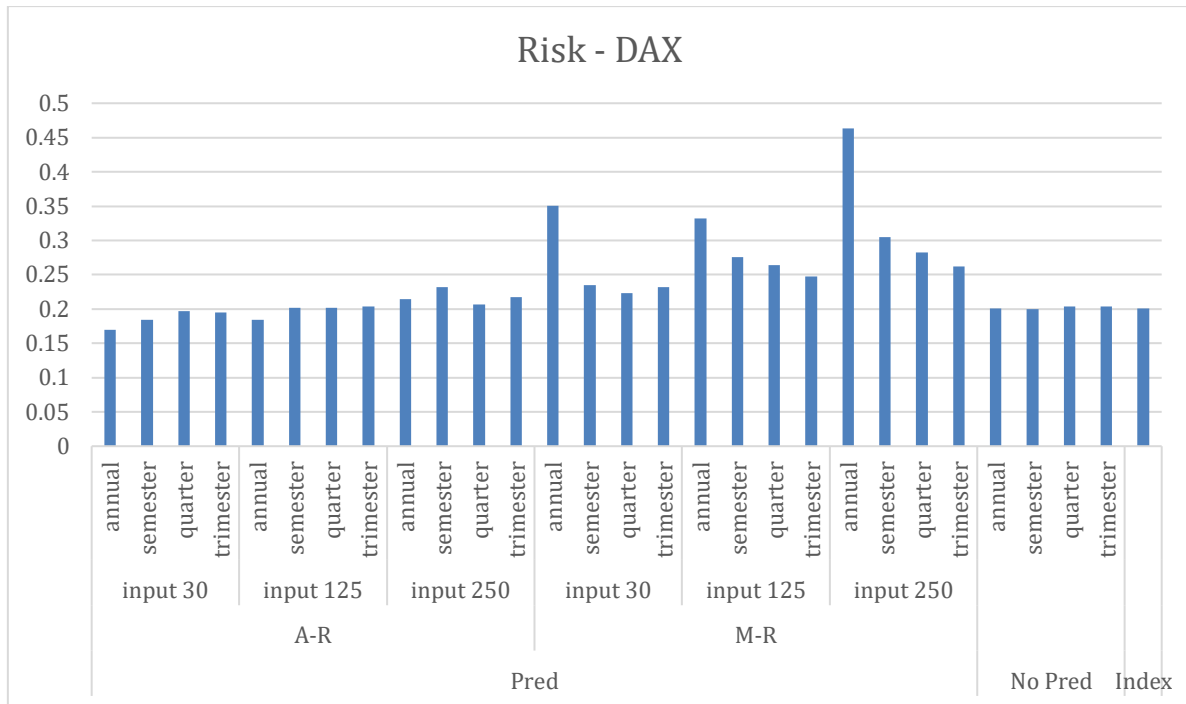


Figure 34: Risk achieved by models using DAX stocks and the DAX index during 2007-2020

As seen in figure 34, the risk of the index is lower than most model's risk and, for the ones that is not, the difference is not significant, whilst the return (figure 33) difference is significant. Therefore, it can be assumed the performance of the index is superior in this case too.

5.3.3 DJI

Lastly, regarding the DJI index, its return during the period 2007-2020 was 8.47% (1stock1, 2020). As with the previous indexes, to compare with the annualized returns achieved by the models that only used DJI stocks, the following graph has been done (figure 35).

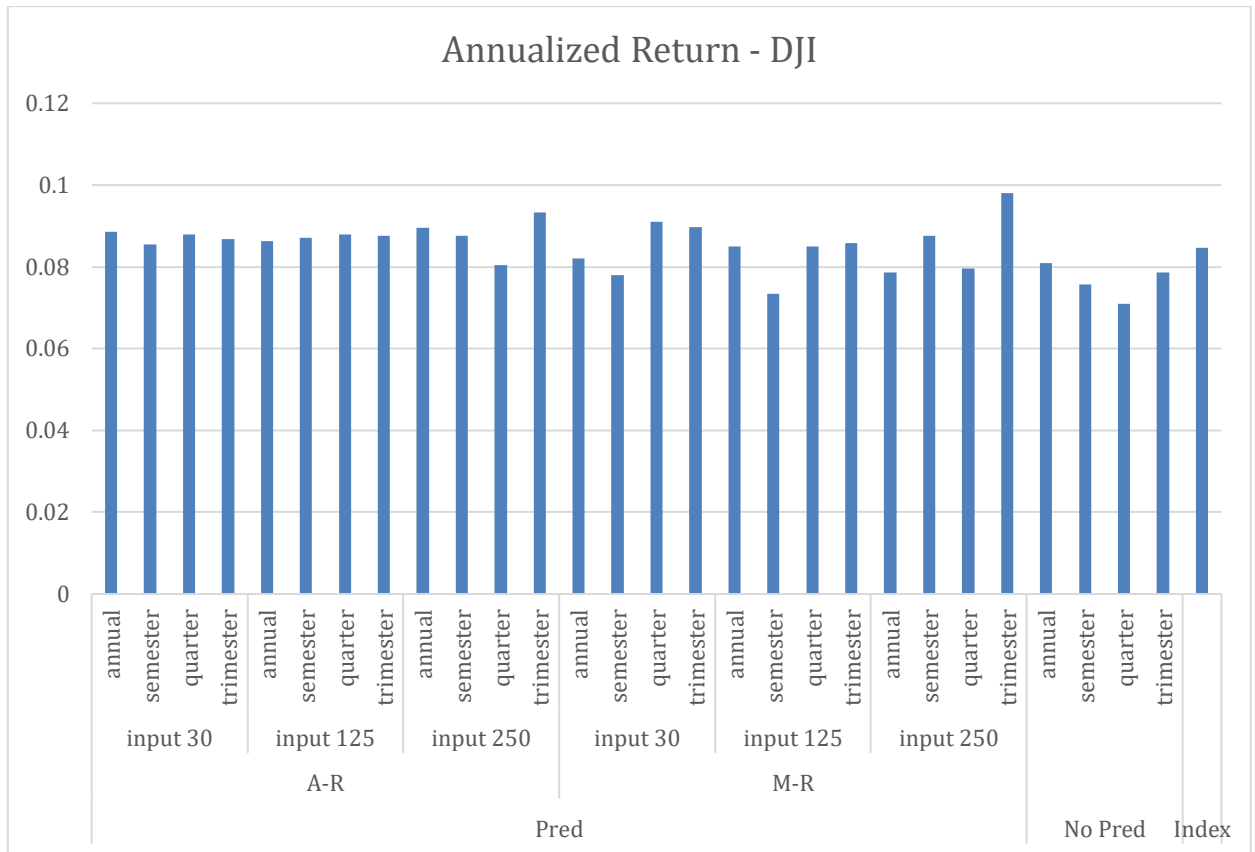


Figure 35: Annualized return achieved by models using DJI stocks and the DJI index during 2007-2020

As seen in figure 35, the DJI index achieved a lower return than what most models were able to achieve. The best model, which in this case has the timeframe is trimester, input 250, used manual recurrence and used predicted prices, was able to return 9.8%. Therefore, given the DJI index returned 8.47% during the same period, the best model beat it's performance by 1.33%, which is a significant amount.

Regarding the models that did not use predicted prices, all their returns were lower than the DJI return.

Next, regarding the risk (formula 3.2, standard deviation in this case), the DAX index during the 2007-2020 had a risk associated of 16.23% (Morningstar, 2022). This is reflected in the next graph (figure 36) to easily compare with the risk of the models created in this thesis.

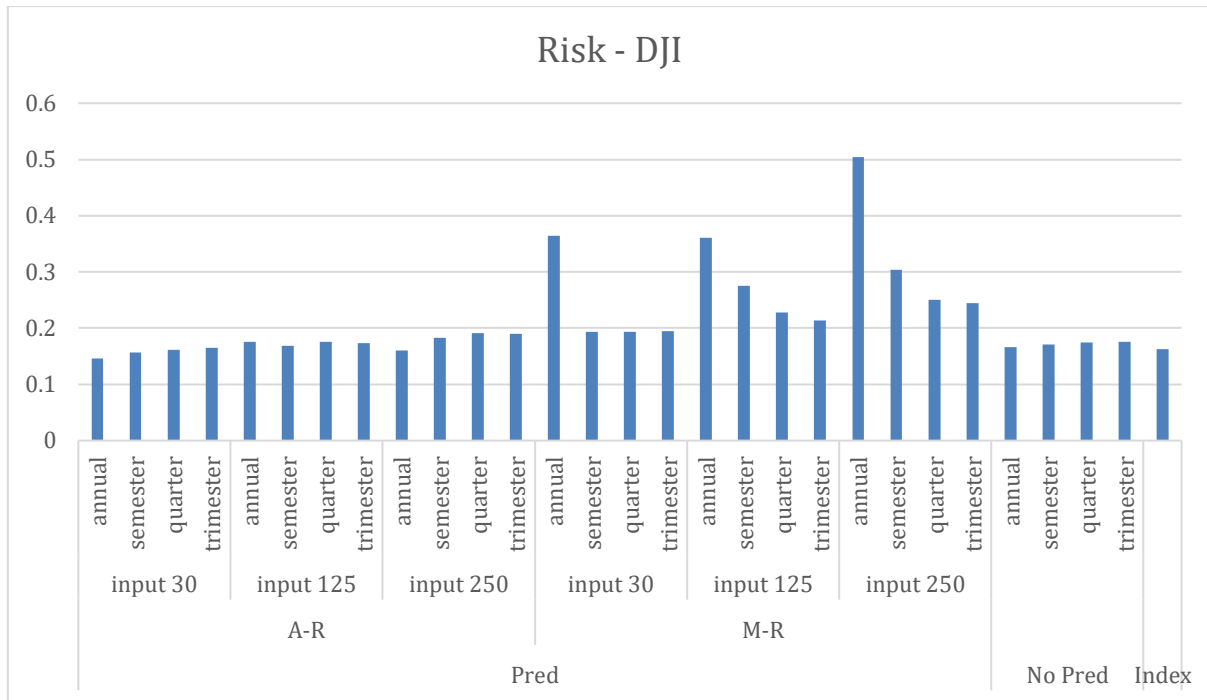


Figure 36: Risk achieved by models using DAX stocks and the DAX index during 2007-2020

As seen in figure 36, the risk of the index is similar to most model's risk. Therefore, given its risk is similar but, its annualized return is less, the models would be a better choice in this case.

5.3.4 Cross evaluation

After analyzing all three indexes separately, some conclusions can be extracted. Regarding the performance of the models that did use predicted prices against that of the indexes, the return of the S&P index was a bit over the best model, but not a significant amount, although given the risk of the index was significantly less, it can be assumed it is a better choice. Then, regarding the DAX index, it clearly outperformed the best model in terms of annualized return, whilst risk was similar. Lastly, the return of the DJI index was clearly surpassed by the return of the best model, whilst their risk was similar. Therefore, to sum up, no assumption can be made saying the models were either inferior or superior to the indexes considering all three cases.

However, one clear pattern does exist. All three indexes achieved a higher annualized return than the models that didn't use predicted prices. Plus, the risk associated to the index was in general less than the models without predicted prices risk. This shows



applying the Markowitz model by itself to build the portfolios is not enough to outperform the indexes returns.

All this indicates that, probably, the bottleneck that is keeping the models from outperforming the indexes might be using the Markowitz model to build the portfolios. Other more modern solutions to build portfolios could be explored in order to test this.

6. Conclusion

In order to achieve the objective of this bachelor thesis, create an optimized portfolio of stocks using stock value prediction in an effort to outperform the original Markowitz model and a passive management approach, twenty-four different stock price predictive long short-term memory models have been created and used. These vary depending on the timeframe being predicted, the number of past prices considered to predict the next timeframe and the recurrence type. They have been applied to three different stock indexes, S&P, DJI and DAX to predict prices and then, used in the Markowitz model to create efficient portfolios for each timeframe.

The portfolios created using predicted prices have been compared with the portfolios created using the Markowitz model without predicted prices and with the performance of each index exchange-traded fund, to compare with a passive management approach.

First, regarding the comparison between portfolios created with and without predicted prices, as seen in the analysis section, portfolios that did use predicted prices improved their performance (measured in return) in all three indexes. Specifically, 13.4% more for S&P, 50% more for DAX and 23.8% more for DJI, an average of 29% increase in return. Therefore, it can be concluded predicted prices had a big positive impact on the performance of the models when used to construct portfolios using the Markowitz approach.

Secondly, with respect to index exchange-traded funds, results were mixed. The return of the S&P index was slightly over the return of the portfolios created by the models that used predicted prices, although given the risk of the index was significantly less it can be assumed it is a better choice. Then, regarding the DAX index, it clearly outperformed the portfolios created in terms of annualized return, whilst risk was similar. Lastly, the return of the DJI index was clearly surpassed by the return of the portfolios created, whilst their risk was similar. Therefore, to sum up, no assumption can be made saying the portfolios created using predicted prices in conjunction with the Markowitz model were either inferior or superior to the indexes considering all three cases. However, one clear pattern does exist. All three indexes achieved a higher annualized return than the

models that didn't use predicted prices. Plus, the risk associated to the index was in general less than the models without predicted prices risk. This shows applying the Markowitz model by itself to build the portfolios is not enough to outperform the indexes returns.

All this indicates that, probably, the bottleneck that is keeping the models from outperforming the indexes might be using the Markowitz model to build the portfolios. Other more modern solutions to build portfolios could be explored in order to test this.

All in all, these conclusions are limited to the three indexes used. Therefore, in order to improve the reliability, a higher amount as well as more diverse indexes could be considered, such as Asian indexes.

Appendices

A.1 Sustainable Development Goals

Degree to which the work is related to the Sustainable Development Goals (SDGs).

Sustainable Development Goals	Alto	Medio	Bajo	No Procede
SDG 1. No poverty.			x	
SDG 2. Zero hunger.			x	
SDG 3. Good health and well-being.			x	
SDG 4. Quality education.			x	
SDG 5. Gender equality.				x
SDG 6. Clean water and sanitation.				x
SDG 7. Affordable and clean energy.				x
SDG 8. Decent work and economic growth.	x			
SDG 9. Industry, innovation and infrastructure.		x		
SDG 10. Reduced inequalities.			x	
SDG 11. Sustainable cities and communities.				x
SDG 12. Responsible consumption and production.				x
SDG 13. Climate action.				x
SDG 14. Life below water.				x
SDG 15. Life on land.				x
SDG 16. Peace, justice and strong institutions.				x
SDG 17. Partnerships for the goals.				x



This bachelor thesis relates to the sustainable development goal number 8, decent work and economic growth. This goal promotes sustained inclusive and sustainable growth, full and productive employment and decent work for all. Specifically, this goal relates to the objective of outperforming the current passive management approaches provided. If this objective is achieved, banks and other investment funds will be able to provide higher return portfolios, increasing the profits gained by clients. Furthermore, money will be invested in companies with a higher expected growth, enabling them to fulfill their expectations. This means, more job opportunities will arise, improving the life standards of living.

Bibliography

1stock1. (2020). *1stock1*. Obtenido de DAX Index (Germany):
https://www.1stock1.com/1stock1_758.htm

1stock1. (2020). *1stock1*. Obtenido de DJIA Index:
https://www.1stock1.com/1stock1_139.htm

Agnon, Y. (1999). Nonparametric, nonlinear, short-term forecasting: theory and evidence for nonlinearities in the commodity markets. *Economics Letters*, 65(3), 293-299. Obtenido de [https://doi.org/10.1016/S0165-1765\(99\)00156-1](https://doi.org/10.1016/S0165-1765(99)00156-1)

Bachelier, L. (1900). THE THEORY OF SPECULATION. *Annales scientifiques de l'École Normale Supérieure*, 17(3), 21-86. Obtenido de <https://www.investmenttheory.org/uploads/3/4/8/2/34825752/emhbachelier.pdf>

Barkoulas, J., & Baum, C. (1996). Nearest-Neighbor Forecasts of U.S. Interest Rates. *Boston College Working Papers in Economics*, 313-420. Obtenido de <http://ideas.repec.org/p/boc/bocoec/313.html>

Belson, W. (1959). Matching and Prediction on the Principle of Biological Classification. *Journal of the Royal Statistical Society*, 8(2), 66-75. Obtenido de <https://doi.org/10.2307/2985543>

Black, F., & Scholes, M. (1973). The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81(3), 637-654. Obtenido de <https://www.journals.uchicago.edu/doi/10.1086/260062>

Cellard, J. (1967). Le programme ELISEE. *Revue Metra*, 1(1), 503-520.

Cesa, M. (2017). A brief history of quantitative finance. *Probability, Uncertainty and Quantitative Risk*, 2(6), 10-24. Obtenido de <https://doi.org/10.1186/s41546-017-0018-3>

Chen, J. (2022). *Investopedia*. Obtenido de Investopedia:
<https://www.investopedia.com/terms/d/dax.asp>

Chen, S. (2016). Detection of fraudulent financial statements using the hybrid data mining approach. *SpringerPlus*, 3(1), 50-62. Obtenido de <https://doi.org/10.1186/s40064-016-1707-6>

Chollet, F. (2015). *Keras*. Obtenido de Keras: <https://keras.io/getting-started/faq/>

Cormack, R. (1971). A Review of Classification. *Journal of the Royal Statistical Society.*, 134(3), 321-367. Obtenido de <https://doi.org/10.2307/2344237>

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 21-27. Obtenido de 10.1109/TIT.1967.1053964

Cowles, A. (1933). CAN STOCK MARKET FORECASTERS FORECAST? *Econometric Society*, 4(3), 20-26. Obtenido de <https://cowles.yale.edu/sites/default/files/files/pub/misc/cowles-forecasters33.pdf>

Cowles, A. (1944). Stock Market Forecasting. *Econometrica*, 12(3), 206-214. Obtenido de <https://doi.org/10.2307/1905433>

Cowles, A. (1960). A Revision of Previous Conclusions Regarding Stock Price Behavior. *Econometrica*, 28(4), 909-915. Obtenido de <https://doi.org/10.2307/1907573>

Curvo. (2022). *Backtest*. Obtenido de Backtest: <https://backtest.curvo.eu/portfolio/s-p-500--NolgygZACgBARABgSANMUBJAokgQnXAWQCUEAOAdIQEYBdeoA?config=%7B%22periodStart%22%3A%222007-02%22%2C%22periodEnd%22%3A%222020-12%22%7D#returns>

Delen, D. (2013). Measuring firm performance using financial ratios: A decision tree approach. *Expert Systems with Applications*, 40(10), 3970-3983. Obtenido de <https://doi.org/10.1016/j.eswa.2013.01.012>

Delgado, M. (1992). Nonparametric and semiparametric methods for economic research. *Journal of Economic Surveys*, 6(3), 201-249. Obtenido de doi:10.1111/j.1467-6419.1992.tb00151.x

Ding, Y., & Sun, N. (2022). Research on Shanghai Stock Exchange 50 Index Forecast Based on Deep Learning. *Mathematical Problems in Engineering*, 22(1), 1-9. Obtenido de <https://doi.org/10.1155/2022/1367920>

Fisher., W. D. (1969). Clustering and Aggregation in Economics. *The Johns Hopkins Press*, 12(1), 22-53.

Foroghi, D. (2011). Applying decision tree to predict bankruptcy. *IEEE International Conference on Computer Science and Automation Engineering*, 165-169. Obtenido de 10.1109/CSAE.2011.5952826

Fradkov, A. L. (2020). Early History of Machine Learning. *IFAC-PapersOnLine*, 53(2), 1385-1390. Obtenido de <https://doi.org/10.1016/j.ifacol.2020.12.1888>.

Gal, J. (1966). Man-Machine Interactive Systems and Their Application to Financial Analysis. *Financial Analysts Journal*, 22(3), 11-22. Obtenido de <https://doi.org/10.2469/faj.v22.n3.126>

Ganti, A. (2022). *Investopedia*. Obtenido de Dow Jones Industrial Average (DJIA): <https://www.investopedia.com/terms/d/djia.asp>

Harries, M., & Horn, K. (1995). Detecting Concept Drift in Financial Time Series Prediction using Symbolic Machine Learning. *World Scientific Publishing*, 91-98. Obtenido de <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.51.5260&rep=rep1&type=pdf>

Harrod, R. F. (1937). Mr. Keynes and Traditional Theory. *Econometrica*, 5(1), 74-86. Obtenido de <https://doi.org/10.2307/1905402>

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-term Memory. *Neural Computation*, 9(8), 1735-1780. Obtenido de <https://doi.org/10.1162/neco.1997.9.8.1735>

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *PNAS*, 79(8), 2554-2558. Obtenido de <https://doi.org/10.1073/pnas.79.8.2554>

Hutter, F., Thornton, C., & Hoos, H. (2014). AutoML. *AutoML 2014 @ ICML* (págs. 50-62). ICML. Obtenido de <http://icml2014.automl.org/>

InteractiveBrokers LLC. (2021). *InteractiveBrokers*. Obtenido de <https://www.interactivebrokers.ie/en/index.php?f=38209>

Investing.com. (1 de 4 de 2022). *German 30-Year Bond Yield*. Obtenido de investing.com: <https://www.investing.com/rates-bonds/germany-30-year-bond-yield-historical-data>

Jiang, Q., Tang, C., Chen, C., Wang, X., & Huang, Q. (2019). Stock Price Forecast Based on LSTM Neural Network. *Proceedings of the Twelfth International Conference on Management Science and Engineering Management*. ICMSEM. Obtenido de https://doi.org/10.1007/978-3-319-93351-1_32

Kass, G. (1980). An Exploratory Technique for Investigating Large Quantities of Categorical Data. *Applied Statistics*, 29(2), 119-127. Obtenido de <https://doi.org/10.2307/2986296>

Khan, N., & Bano, S. (2018). A Survey of Data Clustering Methods. *International Journal of Advanced Science and Technology*, 113(1), 133-142. Obtenido de DOI:10.14257/ijast.2018.113.14

Khen, C. (2000). MEAN ABSOLUTE PERCENTAGE ERROR (MAPE). *Encyclopedia of Production and Manufacturing Management*, 32-34. Obtenido de https://doi.org/10.1007/1-4020-0612-8_580

Kimoto, T., & Asakawa, K. (1990). Stock market prediction system with modular neural networks. *International Joint Conference on Neural Networks*. Obtenido de 10.1109/ijcnn.1990.137535

King, B. (1966). Market and Industry Factors in Stock Price Behavior. *The Journal of Business*, 39(1), 139-190. Obtenido de <http://www.jstor.org/stable/2351742>

Kuznets, S. (1941). Statistics and Economic History. *The Journal of Economic History*, 1(1), 26-41. Obtenido de <https://www.jstor.org/stable/2113393>

Lisi, F. (1997). Is a random walk the best exchange rate predictor? *International Journal of Forecasting*, 13(2), 255-267. Obtenido de [https://doi.org/10.1016/S0169-2070\(97\)00001-0](https://doi.org/10.1016/S0169-2070(97)00001-0)

Lo, A. (2000). Foundations of Technical Analysis: Computational Algorithms, Statistical Inference, and Empirical Implementation. *NBER Working Paper*, 3(14), 34-94. Obtenido de https://papers.ssrn.com/sol3/papers.cfm?abstract_id=228099

Loh, W.-Y. (2014). *A Brief History of Classification and Regression Trees*. Obtenido de Washtat: https://washstat.org/presentations/20150604/loh_slides.pdf

Makridakis, S. (2018). The M4 Competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4), 802-808. Obtenido de <https://doi.org/10.1016/j.ijforecast.2018.06.001>

Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance*, 7(1), 77-91. Obtenido de The Journal of Finance: <https://doi.org/10.2307/2975974>

Meitzen, A. (1891). Part I. The History of Statistics. *The American Academy of Political Science*, 1(2), 13-100. Obtenido de <https://doi.org/10.1177%2F000271629100100203>

Microsoft. (5 de 2 de 2022). *PowerBI*. Obtenido de Microsoft: <https://powerbi.microsoft.com/es-es/>

Miskolczi, P. (2017). Note on simple and logarithmic return. *Applied Studies in Agribusiness and Commerce*, 11(1-2), 127-136. Obtenido de DOI:10.19041/APSTRACT/2017/1-2/16

Morgan, J. (2012). Problems in the Analysis of Survey Data, and a Proposal. *Journal of the American Statistical Association*, 3(1), 415-434. Obtenido de <https://doi.org/10.1080/01621459.1963.10500855>

Morningstar. (2022). *Morningstar*. Obtenido de DAX Risk: <https://www.morningstar.com/etfs/xnas/dax/risk>

Morningstar. (2022). *Morningstar*. Obtenido de DJI Risk: <https://www.morningstar.com/indexes/dji/dji/risk>

Nadaraya, E. (1964). On Estimating Regression. *Teor. Veroyatnost. i Primenen*, 9(1), 157-159. Obtenido de <https://doi.org/10.1137/1109020>

Olah, C. (27 de August de 2016). *Colah's blog*. Obtenido de Understanding LSTM Networks: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Osuna, E. (1998). Support Vector Machines: Training and Applications. *Journal of Economics*, 28(2), 290-501. Obtenido de <http://hdl.handle.net/1721.1/9925>

Prado, M. L. (2016). Building Diversified Portfolios that Outperform Out-of-Sample. *Journal of Portfolio Management*, 30-61. Obtenido de <https://doi.org/10.3905/jpm.2016.42.4.059>

PyPi. (30 de 1 de 2022). *yfinance*. Obtenido de PyPi: <https://pypi.org/project/yfinance/>

Qiu, J., Wang Bin, & Zhou, C. (2020). Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PLoS ONE*, 15(1). Obtenido de <https://doi.org/10.1371/journal.pone.0227222>

Quinlan, J. (1986). Induction of Decision Trees. En J. Quinlan, *Machine Learning* (págs. 81-106). Boston: KluwerAcademicPublisher.

Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning Internal Representations by Error Propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, 1, 318-362. Obtenido de <https://doi.org/10.1016/B978-1-4832-1446-7.50035-2>

Sammut, C., & Webb, G. (2011). Mean Absolute Error. *Encyclopedia of Machine Learning*, 20-26. Obtenido de https://doi.org/10.1007/978-0-387-30164-8_525

Scherer, B. (2002). Portfolio Resampling: Review and Critique. *Financial Analysts Journal*, 58(6), 98-109. Obtenido de <https://doi.org/10.2469/faj.v58.n6.2489>

Sharpe, W. F. (1966). Mutual Fund Performance. *The Journal of Business*, 39(1), 119-138. Obtenido de <http://www.jstor.org/stable/2351741>

Shekhar, S., Xiong, H., & Zhou, X. (2017). Root-Mean-Square Error. *Encyclopedia of GIS*, 143-165. Obtenido de https://doi.org/10.1007/978-3-319-17885-1_101126

Show, D. (2021). *ml-quant*. Obtenido de History of Machine Learning in Finance and Economics: https://blog.ml-quant.com/p/history-of-machine-learning-in-finance?s=r#_ftn82

Siami-Namini, S., & Siami Namin, A. (2018). Forecasting Economics and Financial Time Series: ARIMA vs. LSTM. *ArXiv*. Obtenido de <https://doi.org/10.48550/arXiv.1803.06386>

The Editors of Encyclopedia Britannica. (2022). *Britannica*. Obtenido de S&P 500: <https://www.britannica.com/topic/SandP-500>

Throp, E., & Dennis, R. (2003). A Perspective on Quantitative Finance: Models for Beating the Market. En E. Throp, *Quantitative Finance Review* (págs. 34-50).

U.S. DEPARTMENT OF THE TREASURY. (1 de 4 de 2022). *Daily Treasury Par Yield Curve Rates*. Obtenido de U.S. DEPARTMENT OF THE TREASURY: https://home.treasury.gov/resource-center/data-chart-center/interest-rates/TextView?type=daily_treasury_yield_curve&field_tdr_date_value=all&data=yieldAll

Ullah, A. (1988). Nonparametric estimation and hypothesis testing in econometric models. *Empirical Economics*, 13(1), 223-249. Obtenido de <https://doi.org/10.1007/BF01972450>

Watson, G. (1964). Smooth Regression Analysis. *The Indian Journal of Statistics*, 359-372. Obtenido de <https://pub.uni-bielefeld.de/record/2715511>

Webster, I. (2021). *Official Data*. Obtenido de S&P 500: <https://www.officialdata.org/us/stocks/s-p-500/2007?amount=100&endYear=2020#citation>

White, H. (1992). *Artificial Neural Networks: Approximation and Learning Theory*. Blackwell Pub.

Wikipedia. (1 de 12 de 2021a). *List of S&P 500 companies*. Obtenido de Wikipedia: https://en.wikipedia.org/wiki/List_of_S%26P_500_companies

Wikipedia. (17 de 8 de 2021c). *Dow Jones Industrial Average*. Obtenido de Wikipedia: https://en.wikipedia.org/wiki/Dow_Jones_Industrial_Average

Wikipedia. (20 de 09 de 2021b). *Wikipedia*. Obtenido de DAX: <https://en.wikipedia.org/wiki/DAX>