Enhancing Estimation of Expected Returns in Modern Portfolio Theory through Machine Learning

Auteur : Simar, Julien
Promoteur(s) : Boniver, Fabien
Faculté : HEC-Ecole de gestion de l'Université de Liège
Diplôme : Master en sciences de gestion, à finalité spécialisée en management général (Horaire décalé)
Année académique : 2022-2023
URI/URL : http://hdl.handle.net/2268.2/18948

Enhancing Estimation of Expected Returns in Modern Portfolio Theory through Machine Learning

Supervisor :

Fabien BONIVER

Reader :

David SUETENS

Master's thesis submitted by

Julien SIMAR

to obtain the

Master's degree in general management

Academic year 2022/2023

Abstract

In the context of Modern Portfolio Theory (MPT), this study delves into the application of Machine Learning techniques to enhance the accuracy of expected return estimations. The research encompasses a comprehensive three-fold objective that traverses multiple dimensions of financial prediction and portfolio optimization.

Beginning with an exploration of Machine Learning methods commonly employed in stock return forecasting and time series forecasting domains, the study draws from an extensive literature review to spotlight key models, including Random Forest, Gradient Boosting, Long Short-Term Memory, and Transformers. This cataloging of methodologies forms a foundation for the subsequent investigation.

Building upon this framework, the study shifts its focus to assess the potential advantages of Machine Learning methodologies when compared to traditional methods for expected return estimation. By leveraging historical price data as input, the research examines the extent to which Machine Learning enhances accurate estimations of expected returns. A thorough analysis subsequently underscores the remarkable predictive capabilities of Machine Learning models, with Transformers, the architecture well known for its involvement in ChatGPT, emerging as a particularly potent contender. These findings can significantly influence financial practices, signaling a broader transformation toward data-driven decision-making. This paradigm shift is essential in an era where the complexity and scale of financial data require innovative predictive tools. The implications extend beyond individual strategies, ultimately reshaping the landscape of investment and risk management practices across the industry.

In a final facet, the study probes the financial implications inherent in enhanced expected return estimations, particularly when incorporated within the context of Modern Portfolio Theory. It is here that the research illuminates a counter intuitive insight: while better-predicted expected returns are crucial, they do not invariably translate to superior cumulative realized portfolios. The study uncovers the intricate interplay between estimation precision and the complexities of portfolio construction, highlighting the nuanced nature of investment decisions within a broader financial ecosystem.

Acknowledgments

First, I would like to thank Fabien Boniver and David Suetens for reading and reviewing this document. In particular, I want to thank Fabien Boniver for his guidance and expertise. The many discussions we had enabled me to refine my financial knowledge in terms of strategic asset allocation, which made this dissertation experience captivating.

Secondly, I'd like to thank John Pietquin, who introduced me to Modern Portfolio Theory as part of his Finance course, and Julien Hambuckers, who inspired this dissertation topic.

Finally, doing a Master's degree on a staggered timetable is a demanding but rewarding adventure, and I'd like to thank the people around me who made it possible. First and foremost, my family: you have provided me with a stable, loving and secure environment, which has enabled me to focus on my studies and my professional aspirations, and has been a constant driving force in my development. Secondly, I would especially like to thank Circé for her unfailing support and love and Melvin who showed me the value of a master's degree at HEC and introduced me to the fascinating world of stock market.

This thesis is dedicated to Harry Markowitz, the author of the Modern Portfolio Theory, who passed away in June 2023.

Contents

Chapter 1

Introduction

Although money can't buy happiness, the aspiration for more financial income remains a powerful motivator for individuals, regardless of their income level [Ahuvia, 2008]. This is justified by several factors. Firstly, money offers financial security, enabling people to meet basic needs such as food, housing and healthcare, as well as to cope with the unexpected, thereby reducing stress. Secondly, money is associated with social status, power and recognition, which can lead to a desire to be admired and respected by others. Finally, money offers greater freedom and flexibility in decision-making, allowing us to detach ourselves from the routine of everyday life in order to pursue personal passions [Furnham and Argyle, 1998].

One way to earn money is to invest. Investing means giving up a certain amount of non-essential money for a short-term consumption need in order to allocate it, for a certain period of time, to a use generating a return on investment to offset a future need. [Reilly and Brown, 2011]. This investment can be made in various assets: savings accounts, stocks, bonds, crypto-currencies, pension insurance, mutual funds, index funds, ETFs, options, real estate, etc. [Town, 2022], all of them being associated with different levels of return on investment and financial risk.

Financial risk refers to the possibility that financial decisions taken by an individual may result in financial loss [Jorion and Khoury, 1996]. It can result from a variety of factors, such as economic uncertainty, financial market fluctuations, legislative or regulatory changes, unfavorable business conditions, management errors or unforeseen events. In assets such as savings accounts or bonds, the risk is generally considered low, as the possibility of the risk materializing is virtually non-existent. The investment is therefore of interest to "high-risk-averse" investors. Conversely, "less risk-averse" investors will choose to invest their money in options or cryptocurrencies, where the risk is very high.

Nevertheless, in both cases, the objective of the investor or asset manager will be to build a portfolio by allocating resources to these different assets so that they best match his utility function, the overall satisfaction he derives from this investment [Forjan, 2019]. To achieve this utility objective, the investor will seek to build a portfolio that yields the highest return for a given risk. This is known as Strategic Asset Allocation (SAA). These objectives are, however, opposite in nature and constitute a trade-off: the highest returns are generally associated with the highest risks. In this thesis, we have chosen to study the optimization of SAA on stocks.

Historically, this SAA was based on the Modern Portfolio Theory (MPT) [Markowitz, 1952], a deterministic methodology proposed by Markowitz in 1952 to study the return-risk trade-off by selecting investments in such a way as to maximize the expected return for a given risk. Markowitz quantified the portfolio's return and risk on the basis of its mean and variance. For this reason, MPT is also sometimes referred to as Mean-Variance optimization.

MPT assumes that investors are rational and risk averse: an investor will prefer lower returns with known risks rather than higher returns with unknown risks. Markowitz showed that the contribution of an asset to the risk and return of a portfolio was more important than the risk and return of that asset. He also proved that a diversified portfolio based on asset classes with low correlation could reduce the risk of the portfolio.

This result is easy to understand by taking an example. Let's imagine that your portfolio is made up entirely of transport stocks. The COVID-19 arrives and the population, anticipating a confinement synonymous with a halt in travel, loses interest in these stocks. As supply and demand rule the financial markets, the price of these shares falls sharply. As your shares are all highly correlated, the value of your portfolio also falls sharply. Conversely, if you build up your portfolio with less correlated stocks, such as those in the transport and telecoms sectors, your portfolio will fare better. This was demonstrated during the Covid crisis, when people invested heavily in telecoms, with demand and prices rising as a result ([Ramelli and Wagner, 2020]).

That being said, to enable the MPT model to determine how much of the investor's money should be allocated to different assets, two elements are required: firstly, an estimate of the expected return of each asset, and secondly, an estimate of the covariance (proportional to the correlation) between these different assets. Once the optimization is complete, MPT generates the weights to be allocated to the various assets. These weights are then rebalanced every period (month/quarter) to adjust to the new predicted expected returns and covariances.

The simplest way to determine the expected return at time t is to average historical returns. Intuitively, this estimate only makes sense if we assume that the return has remained stable over time. However, it has been proven that expected returns are stochastic and therefore time-varying [Conrad and Kaul, 1988]. Consequently, if we consider, for example, a return that is sometimes positive and sometimes negative, the average could turn out to be very far from reality. It's easy to see that a value far from reality will underestimate or overestimate the expected return, depending on the case, ultimately resulting in either too low weighting in the portfolio, or too high, but in all cases, a poor estimate of the portfolio's return and therefore poor decision-making. This is also the case if covariance is incorrectly estimated.

Over time, increasingly sophisticated methods have been developed to better estimate these 2 parameters. Based on historical prices, the main aim of this thesis is to study different ways of predicting the expected return with Artificial Intelligence methods and to compare them with traditional methods used in the literature, in order to generate the best portfolio return for a risk given by Capital Market Line.

With these elements introduced, the main objectives of this thesis are as follows:

1. **Comprehensive Understanding of Machine Learning Methods:** The first objective is to comprehensively examine and define the Machine Learning methods utilized in the literature for the purpose of estimating expected stock returns. This involves an exploration of the various approaches, techniques, and algorithms employed in previous research.

2. **Comparative Accuracy of Expected Return Estimation:** The second objective is to assess whether the application of Machine Learning techniques based on historical prices enhances the accuracy of expected return estimation when compared to more conventional methods. Additionally, this objective aims to identify the method that produces the most accurate estimates of expected returns.

3. **Impact of Enhanced Estimation on Investment Performance:** The third objective is to investigate the potential correlation between utilizing a more accurate expected return estimator, as identified in the second objective, and the subsequent impact on achieving a higher

cumulative realized return in the medium term. This objective seeks to provide insights into the practical implications of improved estimation accuracy on investment outcomes.

Through the pursuit of these objectives, this study aims to contribute to the understanding of the application of Machine Learning in estimating expected stock returns, evaluate its effectiveness in comparison to traditional methods, and explore its potential influence on investment performance.

The structure of the report is as follows:

- The theoretical background related to Artificial Intelligence and Portfolio Optimization is presented in Chapter 2.
- The three research questions we will address, related to the above objectives, are presented in details in Chapter 3.
- The literature review required to answer the first research question is described in Chapter 4.
- The quantitative methodology used for the other research questions is detailed in Chapter 5.
- Developments related to this methodology and results are presented in Chapter 6.
- The contributions of this thesis and the guidelines for future work are listed in Chapter 7.

Chapter 2

Background

This chapter will set the scene for understanding this thesis. First, Artificial Intelligence will be demystified in section 2.1. Next, Portfolio Optimization, Modern Portfolio Theory and its possible improvements and robo-advisors will be presented in section 2.2.

2.1 Artificial Intelligence

Artificial Intelligence (AI) today occupies a central place in our rapidly evolving society. Fueled by spectacular advances in Machine Learning and Natural Language Processing, AI is now transforming sectors such as medicine, finance and manufacturing, offering powerful tools for complex data analysis and decision-making. In this thesis, AI methods will be used to predict expected returns. In order to understand how, it is necessary to introduce some context related to this field. In this section, we'll start with the history of Artificial Intelligence in section 2.1.1 in order to differentiate between the notions of Artificial Intelligence, Machine Learning and Deep Learning. Next, we'll look at the learning paradigms (section 2.1.2) associated with Machine Learning and how the best models can be selected (section 2.1.3) and evaluated (section 2.1.4).

2.1.1 Artificial Intelligence history

Artificial intelligence (AI) is an overarching concept that encompasses the development of machines capable of human-like intelligence and tasks. It's about creating systems capable of understanding, reasoning, learning, and adapting, mimicking human cognitive abilities. AI encompasses a wide spectrum of capabilities, from basic rule-based systems where everything is explicitly programmed to highly advanced systems capable of autonomous decision-making. Its history [Haenlein and Kaplan, 2019] goes back several decades and is made up of significant breakthroughs, challenges, and shifts in focus and has gone through 3 phases of evolution. At first, it was just Artificial Intelligence (AI). Then came Machine Learning (ML), which is a subset of AI, and Deep Learning (DL), which is a subset of ML. This is summarized in Figure 2.1 and discussed in more detail in the following sections.

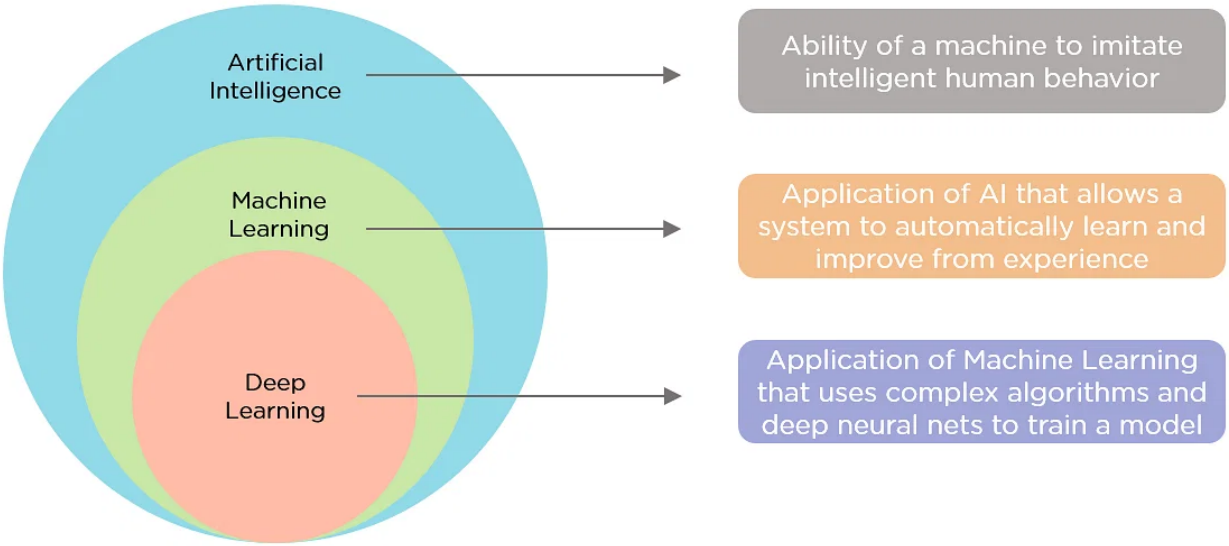

Figure 2.1: Artificial Intelligence, Machine Learning and Deep Learning [M, 2023].

Artificial Intelligence

The first AIs appeared in the 50s and 60s, with computers now capable of storing more information and becoming faster, cheaper, and more accessible. The term was introduced in 1956 by John McCarthy. Early AI efforts focused on symbolic reasoning and rule-based systems designed to emulate the decision-making capabilities of human experts in specific domains. However, the fact that these early attempts assumed that human intelligence could be formalized as a collection of rules was their main limitation, and prevented them from dealing with more complex problems. Let's take the example of a program that has to learn to recognize a type of bird on the basis of a picture of it and the thought processes of a human expert. Since birds come in a wide variety of colors, shapes, sizes and plumages, and can be observed from different angles and orientations, formalizing their recognition on the basis of rules is extremely complex. In addition, human experts often use tacit knowledge and intuition to make decisions, and are not always able to fully articulate their thought processes, making formalization into rules difficult. This limitation led to an "AI winter", when AI faced challenges due to high expectations and unmet promises, despite high spending. The British and US governments decided to suspend support for AI research in most of their universities. This period lasted from the early 70s to the early 90s.

Machine Learning

One of the lessons learned from "AI winter" was that excessive expectations had to be reduced. Researchers began to adopt a more realistic approach, focusing on specific, achievable problems rather than on the generalization of human intelligence. This period was not entirely in vain, however, as research in Machine Learning (ML) gained traction. ML is a subset of AI that enables machines to learn from data and improve their performance over time. In ML, there are no rule-based systems: ML algorithms learn patterns and insights from data to make predictions or decisions. This learning process is a fundamental aspect of ML, as it enables machines to adapt to new situations and evolving datasets. In the context of traditional Machine Learning, we find simple algorithms such as Linear Regression and Decision Trees, as well as more advanced methods such as Random Forest and Neural Networks.

The second phase began in the mid-1990s, when AI started finding practical applications in industries. A much-quoted example is Deep Blue, the chess-playing computer developed by IBM who

was able to defeat chess champion Garry Kasparov. Deep Blue was partially rule-based and partially relied on expert knowledge. Also, in the mid-90s, Ghosh and Reilly [Ghosh and Reilly, 1994] demonstrated using Neural Networks that it was possible to reduce credit card fraud detection on Visa and Mastercard cards by 20 to 40%. Finally, in the early 2000s, Amazon began using recommendation systems to improve product suggestions to customers, based on their shopping and browsing habits [Hardesty, 2019]. However, the 2000s also saw a period of disillusionment and skepticism towards Artificial Intelligence, partly due to the complexity of dealing with ever-increasing quantities of data, and the need to transform these quantities of raw, unstructured data manually.

Deep Learning

The third phase arrived in the early 2010s with the increase in computing power made possible by the use of Graphics Processing Units (GPUs). This made it possible to use ever deeper Neural Networks capable of handling raw, unstructured data without the need for manual engineering to extract relevant features. These Deep Neural Networks form what is commonly known as Deep Learning (DL). DL takes the principles of ML to a higher level by utilizing Neural Networks with multiple layers to process complex data. Deep Neural Networks have the advantage of being ever more efficient as the quantity of data evolves, unlike traditional methods. Significant advances have been made in Deep Learning, Computer Vision, Natural Language Processing and other AI applications. These include technologies such as Apple's Siri and Amazon's Alexa virtual assistants, which can interact with their owners by voice in an individualized way; Tesla's Autopilot system, which can recognize and react to traffic signs, surrounding vehicles and obstacles on the road; and, most recently, ChatGPT, which has made headlines for its ability to converse creatively in a human way, in a wide range of languages and application areas.

In this dissertation, we'll be working exclusively with Machine Learning and Deep Learning, since we want to learn from data rather than define rules about it. It is generally accepted that ML tends to perform better with little data than DL. This is because Deep Learning models, particularly Deep Neural Networks, require a large amount of data to generalize well and make accurate predictions. This gives us the situation shown in Figure 2.2.

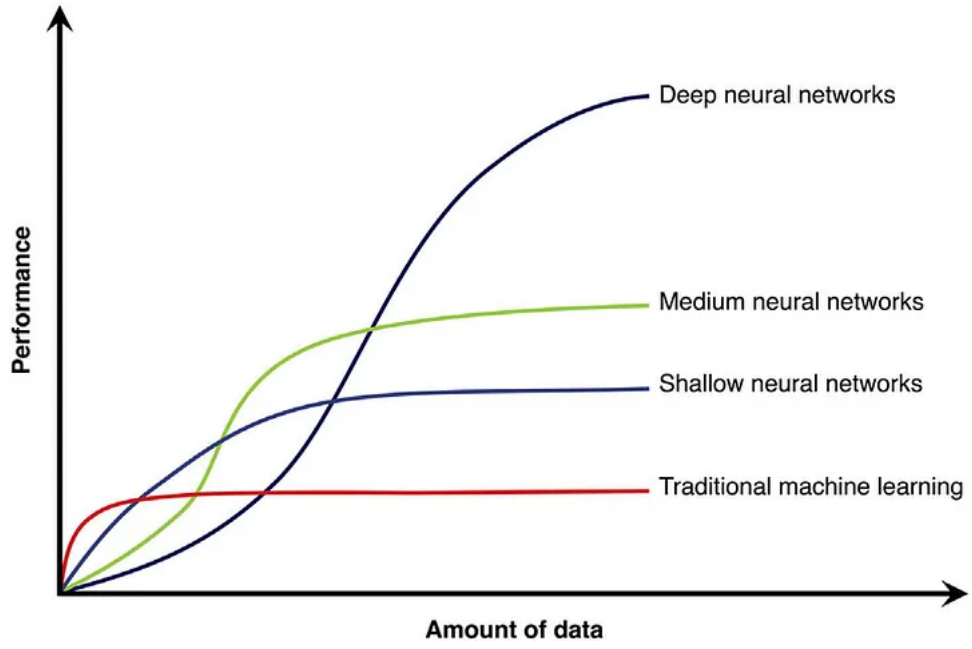

Figure 2.2: Performance of Machine Learning given amount data [Rubilmax, 2022].

2.1.2 Machine Learning Paradigms

There are 3 Machine Learning paradigms [Jo, 2021]. These offer different approaches to handling different types of task and data. The first, Supervised Learning, considers a labeled data set where inputs are associated with desired outputs. The model then learns to predict the correct outputs on the basis of the inputs provided. A financial example is the prediction of the expected stock returns. The algorithm would be given the previous realized returns of a stock as input, and the output would be the future expected return to be predicted. On the basis of a certain number of historical returns-future returns pairs, Supervised Learning is able to build a model capable of linking the two, and thus becomes capable of predicting new returns solely on the basis of old returns. This can be seen as a mathematical function connecting two axes. This is the type of paradigm we'll be using in this thesis.

The second is Unsupervised Learning, based on unlabeled data, which means there is no output data. The aim is to discover trends in the input data. A financial example is customer segmentation based on their investment behavior, which can then be used to make recommendations, offers and personalized marketing.

The final paradigm is Reinforcement Learning. In this paradigm, we have neither input nor output data. We simply have an agent taking actions in an environment in order to maximize a cumulative reward. Over time, the agent learns which actions produce the best rewards. In the financial domain, in portfolio optimization, we could imagine an agent observing the current market conditions and deciding on the basis of this how much money to allocate to each asset. After making these allocation decisions, the agent receives feedback on how well the portfolio performed, learns from this feedback and adjusts its strategy for future decisions. This last paradigm will be reviewed later. Figure 2.3 illustrates the different paradigms.

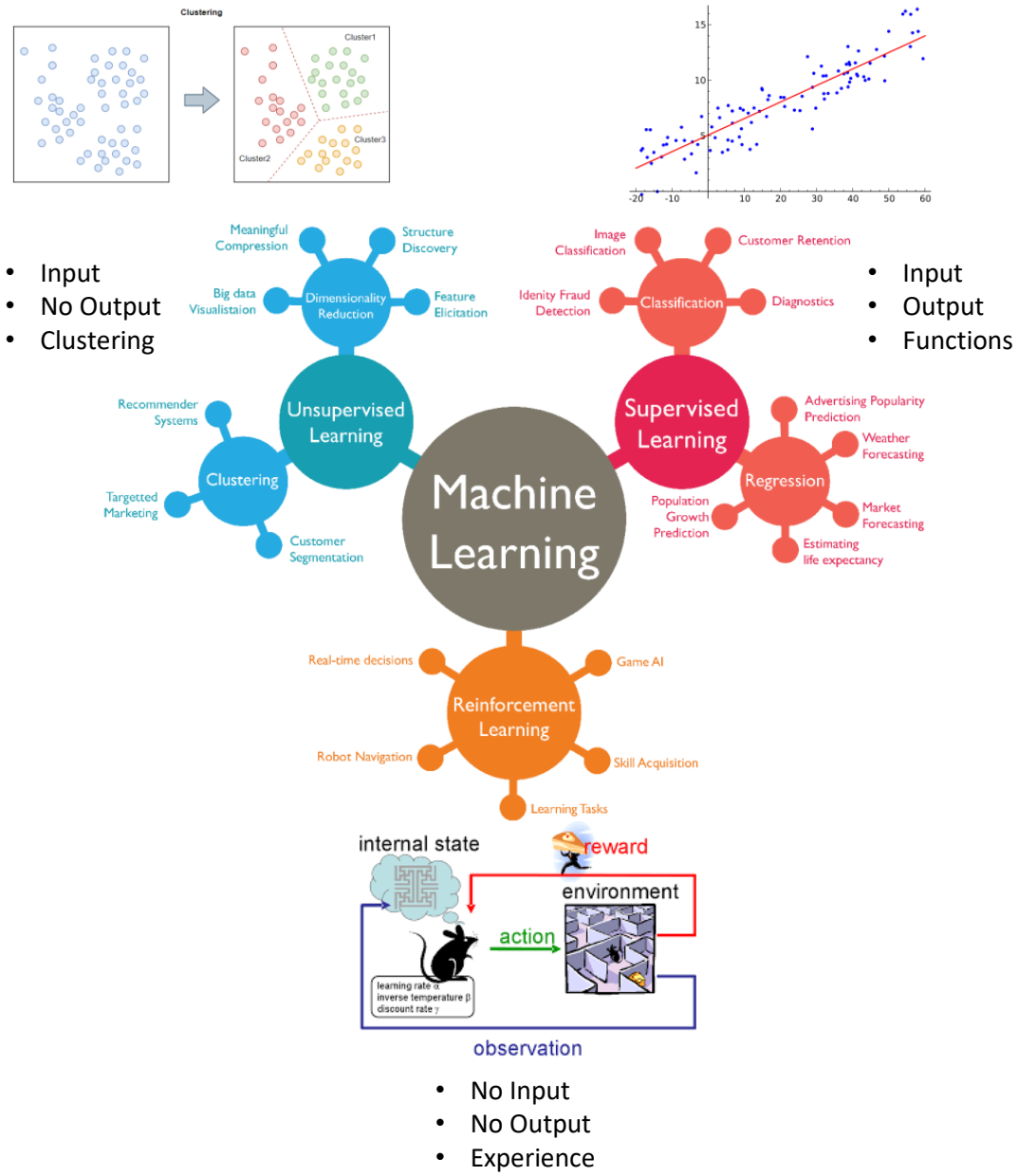

Figure 2.3: Machine Learning paradigms [Chafi, 2022], [Actors, 2018], [Gu, 2021], [Golwala, 2021].

2.1.3 Learning and model selection

In this thesis, we'll be training Supervised Learning algorithms to predict future data from historical data. Training is teaching the model to adjust to in-sample data the data it has seen, so that it can make predictions about new data it hasn't seen before. It's about learning the model that find the pattern that connects the data. When training a model, we may be tempted to believe that the more complex the model, the better it will be. This is true up to a point. Figure 2.4 illustrates this phenomenon. The loss shown is a measure of the extent to which the predictions of a Machine Learning model differ from the actual or target values: a low loss is desirable.

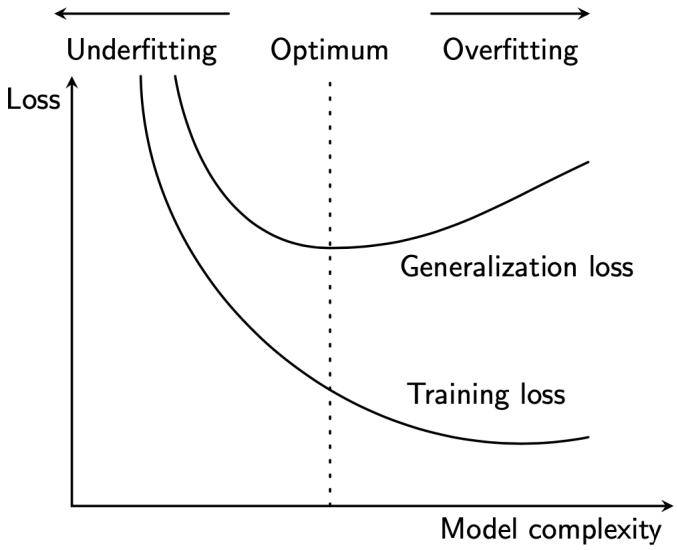

Figure 2.4: Overfitting Loss [cat, 2020].

Learning takes place in 3 phases:

1. At first, the model is not sufficiently trained and has difficulty perceiving the correlation between inputs and outputs. Its ability to generalize from new examples is very poor. In this phase, training loss and generalization loss are at their highest. We speak of underfitting.
2. Then, the model is sufficiently trained. It correctly captures the dependency between input and output, both in terms of training loss (in-sample) and generalization loss (out-of-sample). The complexity is optimal.
3. Finally, the model is over-trained. It becomes excellent on in-sample data and very poor on out-of-sample data. Training loss continues to decrease, while generalization loss only increases. This is called overfitting.

Figure 2.5 visualizes these 3 situations and shows intuitively why the central model is the best model.

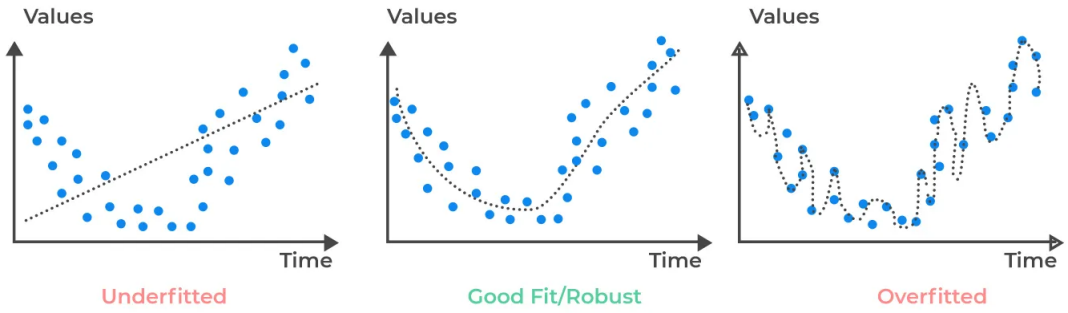

Figure 2.5: Overfitting illustration [Xie, 2022].

Several methods are available to achieve the optimum model. The classic method [Raschka, 2018] is to divide the data (input-output pairs) into three distinct sets: the training set, the validation set and the test set. The first is used to learn a model from data, based on hyperparameters - model parameters that are not learned automatically. The second is used to evaluate the performance of several models trained with different hyperparameters on data not seen during training, in order to

select the best hyperparameters (those giving the best results). A third set is used to evaluate the model's final performance on unseen data. It measures how well the model generalizes to new data.

Another method [Shrivastava, 2023] further improves generalization performance. It's called cross-validation. In cross-validation, rather than having a single training set and a single validation set, we take everything that isn't a test set and create multiple training and validation sets. Each of these training sets will be used to establish a model that will be tested on the corresponding validation sets. By taking the average of the scores obtained on the different validation sets, the best hyperparameters can be chosen.

In the case of timeseries data, such as those we'll be studying in this dissertation, a third method [Shrivastava, 2023] gives better results. This is called timeserie cross-validation. This provides a more robust assessment of model performance using the full data set. Unlike standard cross-validation, where training and test samples are chosen at random, time-series cross-validation takes into account the temporal structure of the data. In a time series, observations are ordered chronologically, and it is essential to maintain this chronology during model validation, as future observations should not be used to predict past observations. Timeseries cross-validation aims to simulate this situation by dividing time series data into training and test sets in a way that reflects the passage of time. An example of 3-split cross-validation is shown in Figure 2.6.

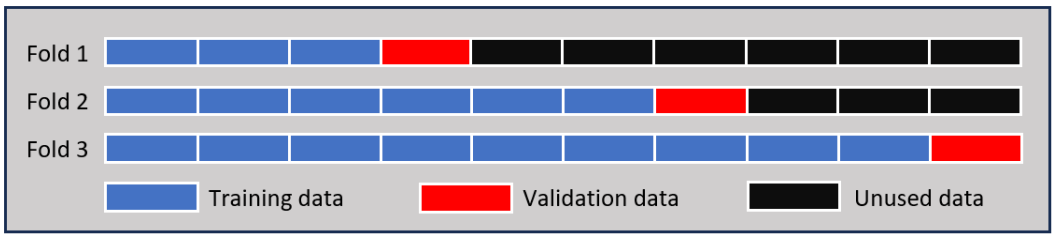

Figure 2.6: Timeserie cross-validation.

2.1.4 Model Evaluation

In this thesis, 2 error measures will be chosen [Willmott and Matsuura, 2005] to study the discrepancy between our models' predictions and actual values. Firstly, the Mean Squared Error (MSE). It is calculated by taking the average of the squared differences between each prediction x_i and the corresponding actual value y_i:

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2$$

The second measure of error is Mean Absolute Error (MAE). It is calculated by taking the average of the absolute values of the differences between each prediction x_i and the corresponding actual value y_i:

$$\frac{1}{n}\sum_{i=1}^{n}|x_i - y_i|$$

MSE puts more emphasis on large errors, as the deviations are high squared. This is useful when you want to penalize large errors. MAE treats all errors equally, whether large or small. It is useful when you don't want to overestimate the impact of outliers. By comparing models according to both MSE and MAE, it is possible to measure model performance from different angles.

2.2 Portfolio Optimization and asset allocation

Portfolio optimization is a key concept for investors of varying levels but also in investment management. It involves building an investment portfolio by allocating different proportions of it to assets of various classes, while balancing return with risk by providing the best risk-adjusted returns for a given level of risk. However, more income is generally associated with more risk [Deng et al., 2012]. In this section, we'll introduce Modern Portfolio Theory and its limitations in section 2.2.1. Then, we will see how it can be improved (section 2.2.2) and how its inputs, the expected return (section 2.2.3) and the covariance estimate (section 2.2.4) can be improved. We will also talk about alternative portfolio optimization methods (section 2.2.5) and robo-advisors (section 2.2.6).

2.2.1 Modern Portfolio Theory

Although Markowitz's groundbreaking paper dates back to 1952, the Mean-Variance (MV) framework remains the dominant model used in asset allocation and portfolio management today, despite numerous other models proposed by academics. One of the reasons for its continued popularity is its flexibility in managing real-word challenges. The MV framework allows for analytical insights and quick numerical solutions to accommodate these issues effectively [Tu and Zhou, 2011].

Mathematical background

Modern portfolio theory is a method for constructing a portfolio of assets that maximizes their expected return for a given level of risk. It assumes rational, risk-averse investors: an investor who prefers lower returns with known risks rather than higher returns with unknown risks. To understand what all this means, a number of notions need to be introduced. We study an asset i over a given period, which may be a day, a week, a month, ... Its price at the end of this period is given by $P_{i,t}$. The return on this asset corresponds to the increase or decrease in its price over the given period. Mathematically, the return $r_{i,t}$ of this asset relative to the price at the start of the period $P_{i,t-1}$ is defined by the relative price change [Brooks, 2019] $P_{i,t} - P_{i,t-1}$ relative to the price at the start of the period $P_{i,t-1}$:

$$r_{i,t} = \frac{P_{i,t} - P_{i,t-1}}{P_{i,t-1}}$$

Imagine we know all the possible returns of this asset and their associated occurring probability. From probability theory, we know that we can compute the expected value of this return, what we would expect on average as return in the long term. This can be defined as $\mu_i = E(r_{i,t})$. Risk is given by the deviation of returns from the expected return: $\sigma_i^2 = E[(r_{i,t} - \mu_i)^2]$. However, the probabilities associated with the various returns are unknown in practice, since financial markets and economic conditions are uncertain and unpredictable. We must therefore attempt to estimate them. Markowitz chooses to estimate them on the basis of timeseries of past returns over T periods of the same length:

$$\hat{\mu}_i = \frac{1}{T}\sum_{t=1}^{T} r_{i,t} \qquad\qquad \hat{\sigma}_i^2 = \frac{1}{T-1}\sum_{t=1}^{T}(r_{i,t} - \hat{\mu}_i)^2$$

Let's now construct a portfolio of n assets with respective proportions (weights) w_i. The sum of all the weights should always equal 1, which means that the entire investment is fully allocated across

the different assets. By removing the subscript t for readability, their returns, expected returns and risks are obtained using the properties of expectations and variances:

$$r_p = \frac{1}{T}\sum_{i=1}^{n} w_i r_i \qquad E(r_p) = \sum_{t=1}^{n} w_i E(r_i) \qquad \sigma_p^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j Cov(r_i, r_j)$$

where $Cov(r_i, r_j) = E[(r_i - \mu_i)(r_j - \mu_j)]$ denotes the covariance between returns r_i and r_j. Covariance can be seen as an extension of variance that considers the joint variability between two returns instead of just the spread of a single return. This formulation of expected return and portfolio risk highlights the fact that, for given weights, these ultimately depend only on the expected return of individual assets i and the covariance between them, the 2 basic elements of MPT. We have already seen that the expected return of an asset i can be estimated on the basis of historical returns. A estimator similar to that of the variance could also be found for the covariance.

Depending on the weights and estimators used, different portfolios with different $E(r_p), \sigma_p^2$ can be obtained. All of these can be represented schematically by the big dots shown in Figure 2.7.

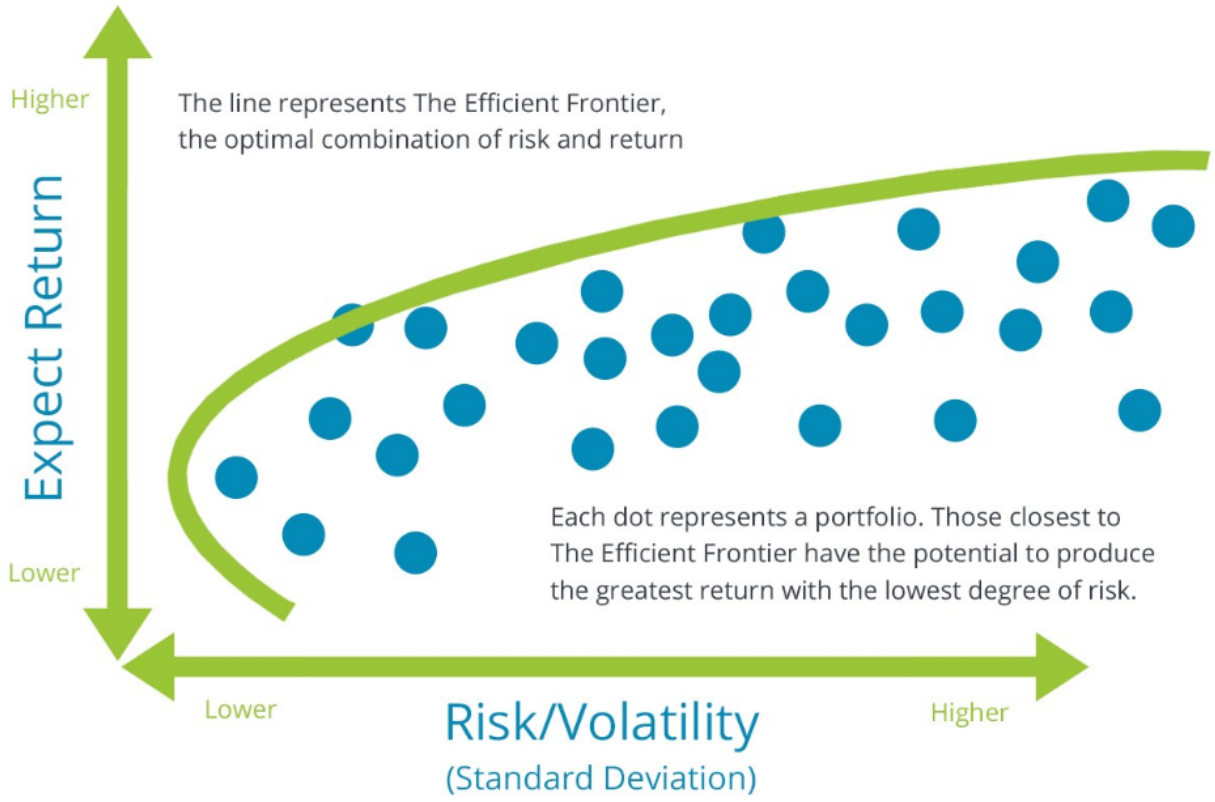

Figure 2.7: Illustration of Efficient Frontier [Goodwin, 2022].

Efficient frontier

For a given level of risk, there is a portfolio with a maximum expected return. Conversely, for a given expected return, there is a minimum risk. It's a trade-off between these 2 elements, and we

understand that there is a series of optimal portfolios, also known as efficient portfolios. There are 2 classic formulations for finding these and their weighting in relation to the various assets in practice. We can either maximize the expected return under the constraint of a given risk (which must be less than or equal to a certain value), or minimize the risk under the constraint of a given return (which must be greater than or equal to a certain value). This is why MPT is also known as Mean-Variance optimization. The solution curve of this optimization, where the efficient portfolios are located, is also shown in Figure 2.7. Its upper part is called the efficient frontier. This is the set of optimal portfolios that offer the highest expected return for a given level of risk, but also the lowest level of risk for a given expected return. According to MPT, there are no portfolios above the efficient frontier because that would imply higher expected returns without taking on more risk, which contradicts the definition of the efficient frontier.

If the portfolio contains a risk-free asset, it is possible to define a single optimal portfolio on the efficient frontier. The risk-free asset represents an investment with a guaranteed return R_f and no risk, such as a government bond or a cash deposit. The portfolio is given graphically by the point of tangency between a straight line drawn from the risk-free asset and the efficient frontier. This line is called the Capital Market Line (CML). This point represents a portfolio with no risk-free holdings and 100% of assets held in the portfolio occurring at the tangency point. A point between the tangency point and the risk-free asset will contain a positive amount of both the tangent portfolio and the risk-free asset, while a point on the half-line beyond the tangency point will contain a negative holding (a borrowing) of the risk-free asset. The situation is shown in Figure 2.8.

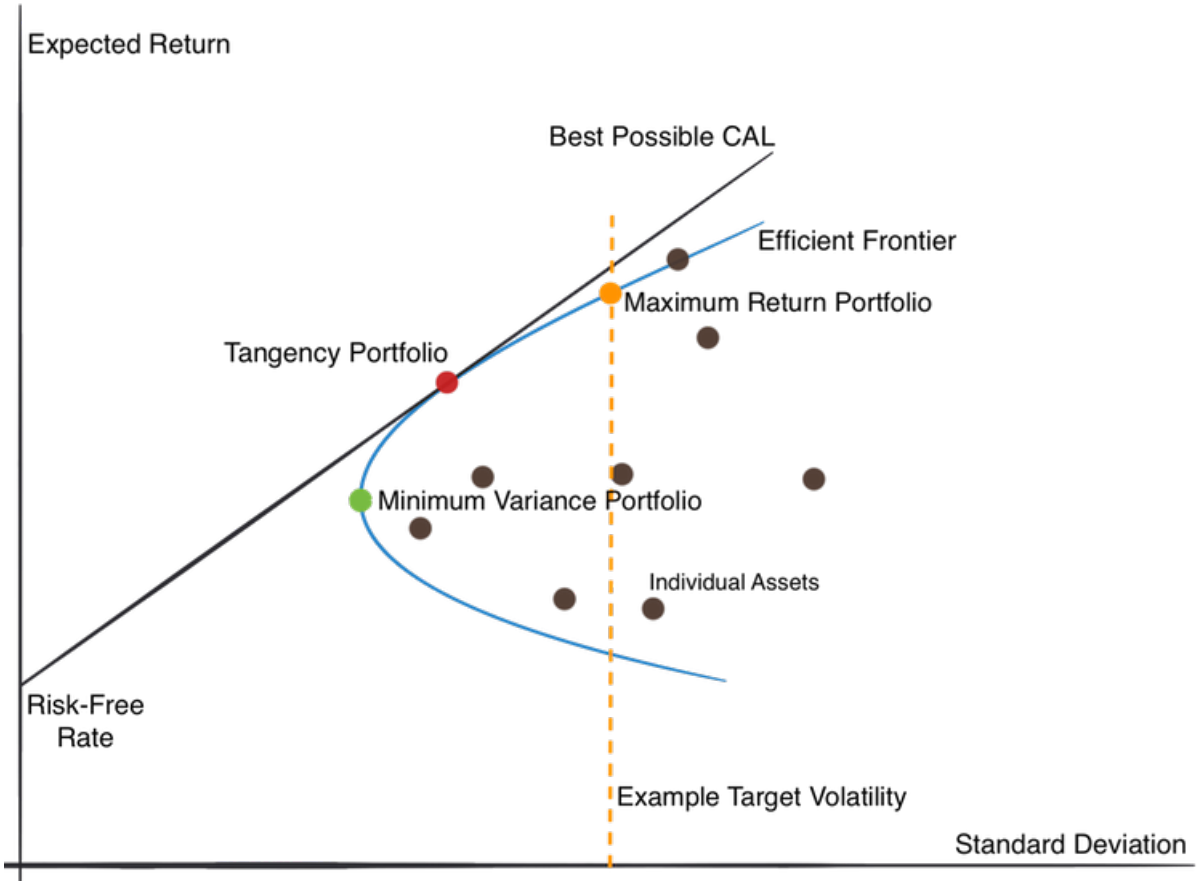

Figure 2.8: Capital Market Line and Efficient Frontier [dhanicova@gmail.com, 2022].

Mathematically, this point is obtained by taking the maximum of the Sharpe Ratio, introduced by William Sharpe in 1966 [Sharpe, 1966]. This Sharpe Ratio must be calculated for all efficient portfolios, and the one with the highest value is retained. His formula is as follows:

$$S = \frac{E(r_p) - R_f}{\sigma_p}$$

Diversification

A final important concept implied by MPT is diversification. In reality, there are two types of risk when considering asset risk: systematic risk and non-systematic risk. The former is caused by factors such as economic conditions, interest rates and geopolitical events, which affect all assets on the market. The second is linked to individual assets or classes. It can be reduced or eliminated by diversifying the portfolio [CFA, 2023]. Diversification doesn't just mean buying more stocks in the same sector. A portfolio containing many stocks from a single sector will not be as well diversified as a portfolio containing a few stocks from a few sectors. The reason is that when one company in a certain sector isn't doing well, other companies in that same sector tend to follow suit. Nor does this mean that it's enough to multiply the number of sectors, since different sectors can behave similarly. What you need to do, above all, is avoid investing in stocks with a high degree of covariance between them.

Efficient Market Hypothesis

Finally, in 1970, Eugene Fama put forward the Efficient Market Hypothesis (EMH). When market news arrives, it spreads rapidly and is promptly incorporated into market prices. As with MPT, EMH assumes that investors are rational people who buy and sell stocks based on available information.

EMH has 3 forms. In its weakest form, Fama suggests that technical analysis aimed at analyzing historical prices to predict future stock prices cannot give an investor an advantage in predicting future price movements: the price already reflects all past trading information, including historical prices and trading volumes. It was however this approach that Markowitz followed in his MPT, since he assumed that history could repeat itself and that noticeable patterns could be discerned in investor behavior by examining charts. In his semi-strong form, in addition to considering technical analysis useless for price prediction, Fama considers that fundamental analysis, which aims to analyze a company's financial information, e.g. earnings, asset values, etc., cannot enable the investor to spot stocks that are undervalued: only information private to the company can do this. In its strong form, even insider information, which is not publicly available, could not give an advantage in predicting stock prices.

Fama isn't saying that it's impossible for investors to generate gains via their strategy, or to outperform the market, but rather that they shouldn't generate consistent abnormal returns, since in his view, price series follow a random walk where price changes are random. Tomorrow's price reflects tomorrow's news, which is unpredictable. And as soon as this news is known, it is reflected in the price. Any investor building a diversified portfolio will get the same portfolio return as the experts. [Malkiel, 2003]. The 3 forms are summarized in Figure 2.9.

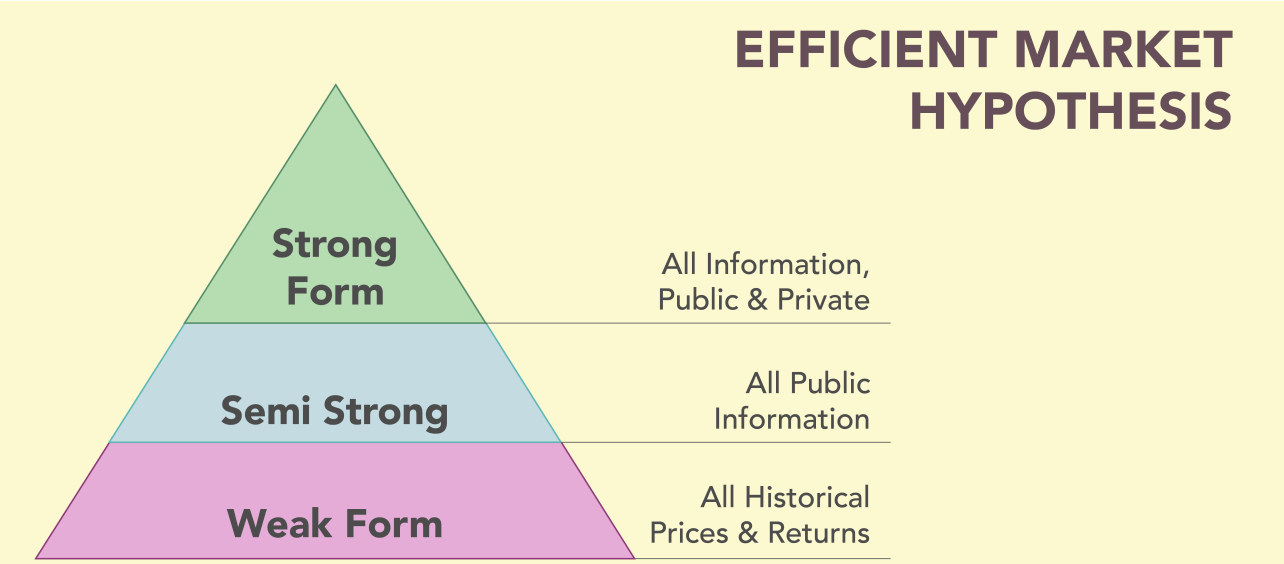

Figure 2.9: Efficient Market Hypothesis [Tejimandi, 2022].

Since what we're trying to do is akin to technical analysis, we could stop here, since even in its weakest form, Fama suggests that predicting expected returns and covariance are useless. However, more recent papers have highlighted the fact that EMH cannot fully explain market anomalies and inefficiencies such as asset bubbles or market crashes, but also the fact that some investors are capable of persistently abnormal returns, which is contrary to EMH.

One reason is that stock markets exhibit chaos [Lawrence, 1997]. Chaos is a nonlinear deterministic process that appears random and is not easy to express. But this is precisely the kind of situation where more innovative methods such as Neural Networks or Artificial Intelligence might be able to highlight these non-linear relationships and thus outperform traditional analysis. The JSE-system [Van Eyden, 1995] for example demonstrated superior performance and was able to predict market direction, so it refuted the EMH.

Similarly, behavioural finance is increasingly present and shows that investors may not always make fully rational decisions and may be influenced by emotions, cognitive biases, and herd behavior, leading to market inefficiencies. For example, the 1987 market crash was contrary to EMH because it was not based on randomly occuring information, but rather because overwhelming investors feared [Lawrence, 1997]. In this sense, Lo [Lo, 2005] suggests that there are mispricings and arbitrage opportunities that only come up from time to time and last for a limited duration . The author therefore suggests the Adaptive Market Hypothesis (AMH) instead, in which the market is efficient during certain periods and anomalous in others.

In this sense too, Timmerman [Timmermann, 2008] claims that markets possess temporary forecastability. Using U.S. stock returns, his approach makes it possible to identify short-lived periods with returns predictability. Indeed, he showed that although stocks are not predictable most of the time, there are episodes of local predictability. Lim and Brooks [Lim and Brooks, 2011] therefore suggest the adoption of rolling windows, such as those found in the moving averages we will see later, when analyzing markets in order to capture medium-term inefficiencies and re-evaluating these periodically so as to always take into account recent market trends. The EMH hypothesis could therefore be true in an ideal world with an equal distribution of information, but in real life, there are players who have this information earlier than others and can therefore outperform the market. However, it's not with a few counter-examples that we can totally refute a hypothesis.

Other limitations

The first criticism concerns the estimation error of the MPT. Although the method presented is optimal in-sample, it has been proven that the Mean-Variance portfolios generated using the estimation of the expected return and the covariance matrix previously defined lead to consequent out-sample estimation errors [Black and Litterman, 1992]. Using historical mean of returns in the estimation of expected returns is not admissible due to the multivariate nature of the problem.

A second problem is that numerically, its solution is very unstable. When the covariance matrix changes even slightly, we observe very large variations in the solution. This is because the covariance matrix must be inverted in the optimization problem, and is sometimes ill-conditioned, leading to large errors. We should also mention the error approximation linked to the fact that MPT assumes normal returns, which is rarely the case in practice.

In the next sections. We'll look at how the paradigm, its estimates and various other portfolio optimizations can be improved.

2.2.2 Modern Portfolio Theory improvements

Given the above-mentioned limitations, let's take a look at some existing methods for improving MPT.

Alternative risk measures

In MPT, risk is defined as a dispersion parameter, the variance, which assumes that returns follow a normal distribution where positive and negative returns are treated equally. However, in real life, the distribution of returns is often asymmetrical and exhibits excess kurtosis. The main problem with variance is that it treats both positive and negative returns equally. That's why, already in 1959, Markowitz [Markowitz, 1959] proposed the semi-variance risk measure, aiming to capture the variability of returns below the mean. Indeed, in practice, many rational investors prioritize the protection of underperformance in their portfolios, rather than outperformance. As a result, they seek risk measures that focus on the downside, which the semi-variance addresses by placing more weight on negative returns.

Other measures risks also exist. The Value at Risk (VaR) [Alexander et al., 2009] criterion is a widely used risk assessment benchmark . It quantifies the maximum expected loss on a target horizon with a specific confidence interval under normal market conditions. In portfolio optimization, a VaR constraint can be included to set a maximum VaR limit for the portfolio, ensuring that the VaR of the portfolio does not exceed a specific threshold (e.g., 5% VaR should not exceed $1 million next week). This allows explicit control over the downside risk of the portfolio. However, VaR has some limitations, such as being a single-point estimate and not providing information about the tail risk distribution beyond the specified confidence level. This means it may not fully capture extreme downside risks in turbulent markets.

Because of VaR limitations, Conditional Value at Risk (CVaR) was introduced. While VaR provides a single-point estimate of the maximum loss at a specified confidence level, CVaR goes beyond and measures the expected loss exceeding VaR [Scutella and Recchia, 2013]. Similiary to VaR, CVaR can be included in portfolio optimization as a constraint.

Alternative Objective function: Risk Parity

In traditional MPT, the objective function is to maximize the portfolio expected return while minimizing its risk. In risk parity optimization, this objective function is modified to minimize the disparity between the risk contributions of the different assets in the portfolio. The concept is similar to that of a naive 1/N allocation between the various possible assets, except that it is the risk and not the wealth that is shared. Several risk measures are possible. These include variance, as in Markowitz, as well as the VaR and CVaR we've just introduced [Costa and Kwon, 2019]. It's shown in the litterature that this type of optimized portfolio outperform optimized allocation strategies such as minimum-variance and Mean-Variance efficient portfolios on a consistent basis [Chaves et al., 2011].

Input View: Black-Litterman

As we have seen, in traditional MPT, the optimization process depends on historical returns but also on asset covariance to determine the optimal portfolio. However, while these estimates are good in-sample, it performs poorly out-of-sample due to estimation errors [Black and Litterman, 1992]. Also, the relationship between expected returns and portfolio weights is complex, and the portfolio weights generated by these optimizers may appear confusing or difficult to interpret, the resulting portfolio not being aligned with the investor's subjective views or market insights.

The Black-Litterman asset allocation framework aims to address these issues by extending Modern Portfolio Theory, removing the need for input estimates of expected returns and offering the possibility of inputting investors' views and beliefs about expected returns. It was introduced by Fischer Black and Robert Litterman in a seminal paper in 1992 [Black and Litterman, 1992] and is based on Bayesian statistics, combining market equilibrium with investor views to generate a more informed set of expected returns for individual assets. On the one hand, Black-Litterman assumes that the initial expected returns are such that the equilibrium asset allocation matches the observed market allocation (assets will perform in the future as they have in the past). Several equilibrium models can be used, the most common being the Capital Asset Pricing Model (CAPM). On the other hand, injecting the investor's view helps to obtain optimal portfolio weights that will be relatively easy to understand [He and Litterman, 2002].

As shown in the original 1992 paper, these 2 improvements have significantly enhanced the MPT model. But this is also its weakness: an overly positive view of an asset leads to a greater weighting of that asset. If this view were unjustified, it could lead to major losses.

2.2.3 Improved estimation of expected return

In this section, rather than improving the MPT paradigm, we aim to improve one of these inputs: the estimation of expected returns. By improving the estimation of expected returns, investors can construct portfolios tailored to their objectives, risk tolerance and economic conditions, maximizing the chances of achieving their long-term financial goals. Three classic methods using Moving Average are discussed in the literature. As seen earlier, this also allows to capture medium-term inefficiencies and to take into account recent market trends [Lim and Brooks, 2011].

Moving Average

The mean used by Markowitz gives a general idea of the data distribution, but does not necessarily smooth out variations. What's more, it depends on just two points and is therefore a very simple estimator. The Moving Average (MA) fills this gap by proposing an average of successive sliding

windows. This smoothes out short-term fluctuations in the data, making it easier to discern general trends. The Moving Average is therefore generally a better estimator. It is often used in the literature. For example, in [Tu and Zhou, 2011] and [D'Hondt et al., 2020].

Exponentially Weighted Moving Average

The Exponentially Weighted Moving Average (EWMA) is very similar to MA, except that it assigns exponentially decreasing weights to past values. In MA, for a given window, the average always depends on just two points. The EWMA, on the other hand, assumes that these 2 points alone are not sufficient, and that the most recent values should be given greater weight, enabling us to react more quickly to recent changes in the data. Although most frequently used to estimate the future volatility of a stock [Alexander, 2008], this method is also used in the literature to estimate future stock returns [Kaul, 1988], [Lou et al., 2019].

Autoregressive Integrated Moving Average

The Autoregressive Integrated Moving Average (ARIMA) is another temporal forecasting model frequently used in statistics and econometrics [Geurts et al., 1977]. It is designed to capture trends and seasonal patterns in temporal data. We'll break down each of its terms to understand how this model works:

1. AR (Autoregressive): An autoregressive model predicts a future observation based on its past observations and a stochastic term (of random probability distribution). They are particularly useful when the time series shows trends or patterns that recur over time.
2. I (Integrated): This refers to the differentiation of temporal data. It involves taking the difference between successive observations of a time series in order to make the series stationary. A stationary series is one whose statistical properties (such as mean and variance) do not vary over time.
3. MA (Moving average): Considers previous prediction errors, also known as residuals or innovations, to adjust future predictions. This means that the model takes into account past errors and attempts to correct them in future predictions.

This model is widely used in financial forecasting. For example, in [Mondal et al., 2014] and in [Ma et al., 2021a].

2.2.4 Improved estimation of covariance

In this section, rather than improving the MPT paradigm, we aim to improve one of these inputs: the estimation of covariance. This is just as important as estimating the expected return since it plays a crucial role in determining portfolio risk and diversification. Here are two classic improvements.

Minimum Covariance Determinant

A possible improvement in covariance is provided by the Minimum Covariance Determinant (MCD) [Mondal et al., 2014], derived from robust optimization techniques. It aims to estimate the covariance matrix of a dataset by identifying a subset of observations (data points) that have the least influence on the determinant of the sample covariance matrix. Traditional estimates can be skewed

by outliers, causing overestimation of correlations and underestimation of portfolio risk. MCD helps in obtaining more accurate measures of risk and correlation, leading to better diversification strategies. The MCD estimator begins by identifying a subset of observations that are the most central to the data structure, those being relatively unaffected by outliers. It then calculates the covariance matrix using only these observations.

Shrinkage

A well-known method for improving covariance estimation is shrinkage. The idea behind shrinkage is to combine data-driven estimation with some form of a priori knowledge to obtain a more reliable estimate, especially when data is limited or noisy. Covariance matrix shrinkage is particularly relevant in the context of portfolio management, where it is crucial to correctly estimate the covariances between the returns of different financial assets. One of the best-known methods is Ledoit-Wolf shrinkage [Ledoit and Wolf, 2003], developed by Olivier Ledoit and Michael Wolf in 2003. Its primary objective is to introduce a form of regularization that reduces excessive fluctuations and estimation errors caused by a small number of observations or poorly estimated covariances. It combines two estimates of the covariance matrix: one based on observed data and one based on individual a priori variances of financial asset returns. The proportion between these two estimates is determined by a shrinkage parameter, which controls the amount of regularization applied. By introducing controlled regularization, this method can help avoid overestimation of correlations and produce more stable and realistic estimates of covariances, with a positive impact in portfolio construction. We'll be using this shrinkage in this dissertation.

2.2.5 Other optimization methods

In the previous 2 sections, we discussed improvements to the Markowitz model, and improvements to the estimates that make it up. In this section, we present a few other portfolio optimization methods found in the literature.

Robust optimization

Markowitz's portfolio optimization model assumes that estimated asset returns and co-variances are constant, even though real asset returns can vary. This leads to differences between the actual portfolio return and the optimal return calculated based on fixed estimates. To overcome this issue and create more robust portfolios, exploring alternative frameworks like robust optimization [Gregory et al., 2011] is worthwhile. Rather than assigning fixed values to the mean and covariance parameters, robust optimization start by defining uncertainty sets representing a range of possible values that these parameters can take within certain bounds. Robust optimization then focus on finding a solution that performs well under the worst case scenario within the defined uncertainty sets. It seeks to protect against extreme adverse conditions.

It's easy to see that there's a trade-off between performance and conservatism. A more conservative approach considers a wider range of uncertainties, which may lead to a more diversified and less volatile portfolio but potentially with a slightly lower expected return. Robust portfolio optimization can be more computationally complex than traditional Mean-Variance optimization because it involves solving optimization problems subject to multiple uncertainty scenarios.

Full-scale optimization

MPT encounters two sources of error: approximation error and estimation error. The first stems from the fact that returns are assumed to be normal, and the second from the fact that quadratic utility is used for investors, causing the Mean-Variance solutions to only approximate the true utility-maximizing portfolio. The estimated means, variances and covariances are therefore unlikely to match the current out-of-sample values. To overcome these limitations, full-scale optimization [Adler and Kritzman, 2006] presents an alternative approach that uses advanced search algorithms to identify the utility-maximizing portfolio within the analyzed dataset, considering any given empirical or theoretical return distribution and investor utility description. If the search algorithm is effective enough, full-scale optimization is not subject to approximation error. On the other hand, estimation error is always present, since calculations are based on the in-sample distribution of returns. Full-scale optimization thus provides the actual optimal portfolio within the analyzed dataset, whereas Mean-Variance analysis offers an approximation of the optimal portfolio within the sample.

Reinforcement Learning

In the literature, there are a growing number of approaches to portfolio optimization based on Deep Reinforcement Learning, one of the 3 Machine Learning paradigms. In this paradigm, an agent learns to make decisions by interacting with its environment, with the aim of maximizing a cumulative reward over time by taking appropriate actions based on the feedback it receives from the environment. In a chess game, for example, an agent may decide to move a particular piece in a particular way on the chessboard. Each action will bring him a certain reward, which will be greater depending on whether or not it leads to victory, a draw or defeat at the end of the game. On the basis of his decisions and those of his opponent, the agent will be able to analyze his possibilities and assess the long-term consequences of his actions on his final victory. The more games he plays, the better his strategy.

In the case of portfolio optimization [Gunjan and Bhattacharyya, 2022], instead of learning to find the best combination of moves, the agent will observe the current market conditions, such as asset prices, economic indicators, and other relevant data and decide on the basis of this how much money to allocate to each asset. After making these allocation decisions, the agent receives feedback (for example higher or lower returns) on how well the portfolio performed.. The agent learns from this feedback and adjusts its strategy for future decisions. Over time, the agent becomes better at selecting the right assets and optimizing the portfolio for improved performance.

Meta-Heuristic Models

When more constraints are added to the Markowitz problem, it may no longer be solvable by quadratic methods and meta-heuristic algorithms come into play. Meta-heuristic algorithms are nature-inspired algorithms used to tackle nonlinear problems with constraints. These algorithms perform an iterative search and strive to reach close-to-optimal solutions in each cycle. While there's no guarantee of finding the absolute optimal solution, meta-heuristic algorithms can be effective for highly constrained problems, especially because they can provide satisfactory solutions in a reasonable timeframe.

An example of meta-heuristic model is given by Ant Colony Optimization (ACO) [Gunjan and Bhattacharyya, 20
developed by Marco Dorigo in 1992. Inspired by ants' foraging behavior, the algorithm mimics how ants search for food to solve optimization problems. In ACO, artificial ants explore the solution space by building and evaluating potential solutions. They leave pheromone trails to communicate with each other, representing the quality of their solutions. When an ant finds a good solution, it

reinforces the corresponding path with more pheromones. As ants repeatedly explore and update the paths with higher pheromone levels, better solutions emerge and attract more ants. Over time, the collective behavior of the ants guides the search towards the most promising solutions. ACO has proven to be effective in solving complex optimization problems, especially those involving discrete and combinatorial spaces. Its ability to combine random exploration and learning from past successes makes it a valuable tool for finding high-quality solutions in various domains.

2.2.6 Robo-advisors

A robo-advisor is an automated online service that uses computer algorithms to provide automated portfolio management and investment advice services. They aim to make portfolio management simpler and more accessible by using technology to automate various tasks, from asset allocation to investment selection and rebalancing. Based on the client's financial situation, risk profile and investment objectives, the robo-advisor creates an investment portfolio allocating different weights to different assets. An algorithm then actively manages the portfolio, rebalancing allocations or reinvesting dividends [Brenner and Meyll, 2020]. In this context of portfolio optimization, we'll see that most robo-advisors use MPT and therefore require an estimate of the expected return, which is what interests us in this dissertation. The assets managed by robo-advisors are indexed stocks, mutual funds and exchange-traded funds (ETFs) [D'Acunto and Rossi, 2021].

History

Investing is a complex issue for many individuals, given the wide variety of financial products, their risk, compounding effects, taxes, and withdrawal strategies. For this reason, turning to financial advisors was a natural step for decades, for their expertise and specialist knowledge in finance, investment, tax planning and wealth management. They were able to provide advice based on their experience and in-depth training. Also, delegating individual allocation to advisers who manage several funds enables economies of scale, since the cost of acquiring information is shared across clients.

However, they were not without their shortcomings. While financial advisors possess the capacity to improve investment options, their effectiveness can be impeded by conflicts of interest. It was shown that advised accounts performed worse than unadvised accounts, because they traded more often, creating a cost for investors and that advisers transmitted their own biases, making the same mistakes in their investments as in those of their clients. All this lasted until the financial crisis of 2008, when many investors questioned traditional portfolio management models in search of more transparent and affordable solutions. Riding the wave of financial technology (fintech), pioneers such as startups Betterment and Wealthfront appeared on the scene.

In 2010, Wealthfront, which had started out as a mutual fund company offering high-quality asset management at a low cost with human advisors, understood what IT could do for it and shifted its business model to become an automated digital wealth management platform. At the same time, Betterment automated the process of selecting and managing investments, making investing simpler for its customers [Fisch et al., 2019]. This is how the era of robo-advisors began. Robo-advisors emerged as a solution that is not only cost-effective due to their reduced staffing requirements, but also highly efficient, capable of processing vast amounts of data within shorter timeframes. Moreover, they offer a personalized experience by incorporating individual preferences such as risk tolerance and profit objectives. The accessibility factor also comes into play, as robo-advisors are accessible to everyone and designed for user-friendliness.

These early successes attracted the attention of investors and the media, and the global rise of robo-advisors could begin. The number of users reached 2.8 million in 2015, for a total of $66 billion in

assets under management (AUM) in 2015 [Ponnaiya and Ryan, 2017]. It was around this time, after observing the growing attractiveness of robo-advisors in startups that established asset managers such as Charles Schwab, Vanguard, BlackRock, Goldman Sachs, and Merrill Lynch began developing their own robo-advisor solutions. The emergence of roboadvisors in a lower-fee environment had forced these traditional financial services to revisit their fee structure by integrating robo-advisor services, enabling them to retain their customer base and acquire a new one [Phoon and Koh, 2017]. In recent years, the growth of robo-advisors has continued unabated. In 2017, robo-advisors managed $200 billion in assets worldwide out of the $80 trillion of global assets under management, i.e. 1/400 [Fisch et al., 2019]. This ratio has continued to rise steadily in recent years. In 2022, roboadvisors were now managing $2.4 trillion, 10x more in 5 years [Statista, 2023]. Over the same period, total assets only rose to $115.1 trillion [PricewaterhouseCoopers,], increasing the proportion to 1/50, or 2%.

Clients

The clients of traditional financial advisors are generally older, with higher incomes, more wealth or more inherited wealth. Robo-advisors, on the other hand, fill a unique niche of customers who don't meet the asset under management minimums for traditional financial planning firms, who want fast, low-cost financial advice and who have generally been disappointed by past financial decisions. However, they do not cover impulsive participants with large purchase decisions or who are unaware of their budget [Fulk et al., 2018].

How it works

Most robo-advisors determine investors' risk profile using online questionnaires covering their current financial situation, investment goals, time horizon, risk tolerance, investment preferences. Eazyvest [Eazyvest, 2023], the first robo-advisor to launch in Belgium asks for example the following 6 questions:

1. How old are you?
2. How much would you invest at the beginning?
3. What is the total value of your liquid assets?
4. What is your investment horizon?
5. How much do you plan to continue to add monthly?
6. In case of market downturn, what is your yearly acceptable decline?

The main functionalities of robo-advisors are asset allocation, portfolio monitoring and portfolio rebalancing.[Phoon and Koh, 2017]. MPT is used for portfolio allocation in almost 40% of all roboadvisors [Beketov et al., 2018]. Typical rebalancing ranged from weeks to months [Parliament, 2021].

Chapter 3

Research questions

After introducing the managerial and scientific motivations of this research in section 3.1, we will talk about its main contributions in section 3.2.

3.1 Motivations

3.1.1 Managerial

More financial income is the objective of many individuals and organizations, and every investor or asset manager seeks to build a portfolio that maximizes his return for a given level of risk. The Mean-Variance framework, introduced by Markowitz and also called the Modern Portfolio Theory (MPT) helps to achieve this objective. MPT requires 2 elements: an estimate of the expected return of the different assets making up the investment universe and an estimate of the covariance between these different assets. In this dissertation, we focus on improving the prediction of expected returns with the help of Machine Learning. Artificial Intelligence revolution is underway and has proven itself in many domains: pattern recognition, speech recognition, and Natural Language Processing. It could also be beneficial here.

By improving predictions of expected returns, investors can make more informed decisions about the composition of their portfolios, and identify assets that present opportunities for higher returns. This improvement has also an impact on risk management, as more accurate predictions help to more effectively assess the levels of risk associated with each asset. In addition, better prediction of returns guides asset allocation based on new information, facilitates diversification by including more reliable assets, and enables active portfolio managers to identify undervalued opportunities. Overall, more accurate return predictions strengthen decision-making, portfolio management and the achievement of financial objectives, by judiciously balancing return and risk.

Also, in the past, wealth management was expensive due to the need to interact with finance experts. Now, everyone can have access to robo-advisors, a disruptive digital service offered by certain FinTechs and most of whom use the MPT (as seen in Context chapter). These are cheaper (less staff), more efficient (larger volumes of data can be treated in less time), personalized (as it takes individual preferences: risk tolerance, gain objective, ...) and accessible (everyone can access them and they are easy to use...). So there's an obvious interest in improving the performance of these robots.

3.1.2 Scientific

Few articles have focused on improving the MPT or robo-advisors with Artificial Intelligence. These algorithms can undoubtedly give quite interesting results also in the financial domain. Estimating the expected return of a given asset is a scientifically challenging problem, since it varies according to the supply and demand for it, which themselves evolve according to the political, economic, societal and ecological events that occur. It is therefore non-linear, dynamic, noisy and chaotic [Deboeck, 1994]. Numerous methods exist in the literature for estimating it. Until a few years ago, models were purely statistical. The simplest method, proposed by Markowitz in 1952, simply takes the asset's historical price and calculates the historical average of its value. This estimation, based on the origin and end of the time history, estimates the general trend of a time series and therefore only works when assets evolve (positively or negatively) constantly over time, which is very rare over a long period given their stochastic nature. Other classic estimators were discussed in section 2. In this dissertation, we will extend classical estimators of expected returns to state-of-the art Machine Learning algorithms, on new assets, which will constitute a significant added value.

3.2 Personal contribution

As seen previously, the literature has seen increasingly sophisticated methods replace the historical average over time, such as the Moving Average, the Exponentially Weighted Moving Average or ARIMA. Since these models are not very complex in nature, they do not capture the reality of financial market fluctuations, which can be asymmetrical and non-linear.

Today, AI is increasingly replacing these statistical models in timeseries forecasting. Unlike statistical models, which are based on simplifying assumptions, AI is capable of capturing complex, non-linear patterns in raw time series. At first, Machine Learning demonstrated its value. Then, with the rapid growth in computing power, even more advanced and powerful networks, known as Deep Learning, emerged. These AI models are still not widely used in the field of expected return prediction.

In this dissertation, we would like to predict expected returns based on different Machine Learning algorithms, and compare their performance with that of the classical methods. Most approaches in the literature call on a number of additional economic features. Here, we would like to have a purely algorithmic approach where each method is simply compared with the others on the basis of what it is able to predict from the historical price, without trying to play on these additional features. Classical approaches to expected return estimation have been defined previously. This is not the case for Machine Learning methods, which have yet to be defined. This leads to the first research question we would like to address:

Research Question 1
What is meant by Machine Learning methods from the literature for estimating expected stock returns?

For this question, we will carry out a two-stage literature review. First, we'll look at what the financial literature has to offer in terms of Machine Learning methods for stock return forecasting. Then, as the scientific literature is ahead of its time on these notions, we'll take a more general look at timeseries forecasting methods. The answer to this question will be found in the literature review, in Chapter 4.

Once this has been done, we'll compare the Machine Learning and classical methods in terms of stock return estimation, to see which is the most ***accurate***, and determine which is the best model of all those tested. By ***accurate***, as we'll see in the methodology chapter, we mean that we're going to compare the different expected returns obtained on the different stocks via the different estimation methods with the realized returns in terms of the means of the Mean Squared Error (MSE) and the Mean Absolute Error (MAE). We thus have as a research question:

Research Question 2

Is the estimation of expected return via Machine Learning based on historical prices more **accurate** than that provided by more traditional methods? Which method gives the best estimates?

At this stage, several methods (Machine Learning and classical) will have been tested and ranked in terms of their expected stock return estimates. The algorithm(s) best able to estimate expected returns will have been identified. As seen in the background chapter, at time t and for each method, these expected returns for the various stocks of our investment universe can be provided in conjunction with the covariance estimated as MPT input, in order to obtain weights for our various assets. These weights are calculated on the basis of the chosen optimization: minimization of portfolio volatility under an expected return constraint, maximization of expected return under a volatility constraint... and are then applied to the realized returns of the different assets, and a realized return of the portfolio can be calculated. A rebalancing is performed every period, i.e. in t+1, t+2, ... and the process is then repeated.

However, there's no guarantee that the best algorithms for predicting expected returns will ultimately lead to the best portfolios, any parameter in the MPT other than expected return remaining equal from one algorithm to another. Over one period of time, it would not be surprising if the best estimator did not lead to the best realized portfolio return. Indeed, the best estimator is determined by its ability to best estimate the mean realized return out-of-sample. But we could have a less good estimator that predicts more successfully than the best estimator one or more of the stocks with a very high weight in MPT. This could be for example because the expected return in question of this/these stock(s) is high and/or because the covariance of this/these stock(s) with other stocks is low. In this case, by a stroke of luck, the realized return of the portfolio would be higher. However, we believe that using a better estimator of the realized return should, in the medium term (4 years in our study), erase these effects due to chance and ultimately produce a better cumulative portfolio. In any case, this is a thesis we'd like to verify:

Research Question 3

Does a more **accurate** expected return estimator lead to a higher cumulative realized return in the medium term?

For this question, we will carry out a graphical comparison of cumulative realized returns over this 4-year period.

Chapter 4

Literature review

In a previous chapter, we introduced the Mean-Variance framework as a method for optimizing the return of a portfolio for a given risk. We saw that for this framework to work, an estimate of the expected return and the covariance between the stocks studied was necessary. We then chose to focus on the estimation of the expected return and defined several research questions along these lines. The first of these is concerned with finding Machine Learning methods for estimating expected return, and to this end we are going to carry out a two-stage qualitative study. Firstly, we will study the approaches described in the financial literature (section 4.1). We will then complement this approach with recent scientific trends (section 4.2). Indeed, AI methods appear in the scientific literature before they appear in the financial literature [Dixon et al., 2020]. Finally, we will conclude with the methods to be used for the next research question (section 4.3).

4.1 Stock Return Forecasting

We will first explore the history of stock return forecasting in section 4.1.1 and then talk about recent trends in section 4.1.2.

4.1.1 History

As we saw earlier, AI contains a range of methods, including Machine Learning, which in turn contains a range of methods, including Deep Learning, which is made up of Deep Neural Networks. Many methods from these 2 families of algorithms are used in the stock return prediction literature. The main advantage of a Machine Learning method over a conventional method is that it is able to learn the relationship between inputs and outputs.

In the 1980s, seminal work in the field of stock return prediction paved the way for the possibility of predicting stock market variations using Linear Regression models applied to time series data. Linear Regression is a method for modeling the linear relationship between two variables. It finds the best straight line (regression line) that minimizes the distance between predicted and actual values. The slope of the line represents the relationship between the variables, and the intercept is the estimated value when the independent variable is zero.

The time series studied included historical stock prices, but also economic indicators such as interest rates and inflation [Fama, 1981]. This first version of Linear Regression relied on the Ordinary Least Squares (OLS) method to determine its parameters. Over time, two improvements were made to this initial approach. Firstly, the Autoregressive Least Squares (AOLS) was introduced to take account of the possible autocorrelation between historical returns and the correlation between these returns

at different points in time. Next came Feasible Generalized Least Squares (FGLS). This improved the heteroskedasticity of returns, i.e. the non-constant variation in the variance of returns. Although Linear Regression is still used today in contexts where simplicity and interpretability of data is desired [Phan et al., 2015], it generally fails to capture the complexity involved in predicting stock returns, and a significant residual variance between predictions and reality almost always exists.

Non-linear models, more appropriate in our study where performance is sought, have therefore emerged. These explain the residual variance more satisfactorily, offering greater flexibility in capturing the complex and dynamic patterns of financial data. As early as 1988, in parallel with linear developments, a first paper [White, 1988] looked at Neural Networks to predict IBM's daily stock returns.

Neural networks are Machine Learning models inspired by the human brain, composed of layers of interconnected neurons. Each neuron performs calculations on inputs, weights them and passes them through non-linear activation functions. Neural networks can capture complex patterns in data and are used for classification, regression and other tasks. Learning is achieved by adjusting the weights of connections to minimize a loss function, using optimization algorithms such as Gradient Descent. A Neural Network consists of an input layer containing the data supplied to the algorithm, one or more hidden layers and an output layer representing the data to be predicted. The more hidden layers there are, the more weights there are and therefore the more complex the model, which is generally associated with longer computation times.

By then, this model had already proved its worth in pattern recognition and nonlinear forecasting. The author sought to establish the link between the return at time t and previous returns using a single hidden layer, the model being particularly demanding in terms of computing power. He compared his results to a Linear Regression model with OLS and showed that his model was capable of far more dynamic (and non-linear) predictive behavior than this other model, predicting peaks close to the actual peaks in some cases, which the linear model, far less volatile in its predictions, proved unable to do. However, it was precisely this lack of risk-taking on the part of the linear model that gave it, in the end, predictions that were on average (measured by a correlation index) closer to reality than those of the non-linear model, which was sometimes very wrong. In this day and age, and given the description of Deep Learning in Chapter 2, we can understand that this weakness of the 1988 model was mainly due to its lack of complexity. As a result, the model was unable to extract features from historical prices to aid its prediction.

This is why, in 1999, an approach [Qi and Maddala, 1999], proposing the input of features via a few economic variables, premade the work of a Neural Network and obtained better results with a single hidden layer too. Aware that most financial and economic variables were non-linear, the authors used 6 economics variables, namely dividend yield, one-month treasury bill rate, changes in short-term interest rate, growth rate of industrial production, inflation rate and money growth rate to predict the excess stock return S&P 500 index portfolio. It performed better in terms of Root Mean Squared Error, Mean Absolute Error, Mean Absolute Percentage Error, Pearson correlation coefficient, percentage of correct direction changes than a Linear Regression model. Subsequently, increasingly sophisticated Neural Network models were developed, and this complexity paid off in terms of performance. An example is "The use of data mining and neural networks for forecasting stock market returns" [Enke and Thawornwong, 2005], which in 2005 made this Neural Network approach even more complex by using 2 hidden layers, for greater depth. Using a total of 14 economic variables, they showed that this deeper model obtained a better Pearson correlation, Root Mean Squared Error, and correct sign of excess stock return than a more modest Neural Network.

Subsequently, these Neural Network models continued to become even more complex. As we saw in Chapter 2, the early 2010s saw an increase in available computing power, enabling deeper Neural Networks (with more layers) capable of extracting features from raw, complex data. The constant developments in Neural Networks might lead one to believe that this is the only method explored

in the financial literature in the field of stock return prediction. However, in Chapter 2, we also saw that other Machine Learning methods have been developed in parallel. In recent developments in stock return prediction, these 2 approaches are present, as we will see in the next section.

4.1.2 Recent developments

This section presents the 4 papers that represent the current state of the art in stock return prediction. Each time a method is introduced, it will also be briefly presented. In the models used, it's important to note that each method will be used to predict a single stock return at a future time t based. This is known as single-step forecasting. The models will not be trained on all stocks together, but will be stock-specific.

Deep Learning for Forecasting Stock Returns in the Cross-Section [Abe and Nakayama, 2018]

In 2018, the authors of this paper compared Machine Learning methods to Deep Learning methods on a monthly return prediction problem for MSCI Japan Index constituents. 26 years of data were used. They used as input 25 economic factors such as Earnings-to-price ratio, Dividend yield, Sales-to-price ratio... The Machine Learning methods used were Random Forest and Support Vector Regression.

Random Forest [Breiman, 2001] is an ensemble technique in which several Decision Trees are created using random samples of data and their characteristics. Each tree votes for a prediction, and the average of the predictions becomes the final prediction. Decision trees are well known to the general public for their intuitive visual representation. They make decisions by following branches of a tree based on data characteristics. Each node of the tree represents a question about a feature, and the branches correspond to possible answers. Trees are built iteratively, and the leaves (terminal nodes) give predictions. Successive decisions based on specific features correspond to decision-making processes that people can easily associate with everyday situations.

Support Vector Regression (SVR) [Drucker et al., 1996] seeks to find a function that predicts target values by maximizing the margins between predictions and actual values. Data that fall within the margins or beyond a certain tolerance are ignored to avoid over-fitting errors. The aim of SVR is to solve an optimization problem to find the weights and biases of the prediction function, while respecting the margins and tolerance.

On the Deep Learning side, the authors used shallow Neural Networks (few layers) and Deep Neural Networks (many layers). The authors showed that Deep Neural Networks outperformed both shallow Neural Networks and Machine Learning models, both in terms of rank correlation coefficient and directional accuracy, two performance metrics used to evaluate predictions.

Portfolio optimization with return prediction using Deep Learning and Machine Learning [Ma et al., 2021b]

Although Deep Neural Networks have the ability to capture complex features and non-linear relationships in data, this doesn't automatically mean that even deeper networks will always perform better. Sometimes, proposing a smarter neural layering approach can fare better.

In the case of sequential data such as temporal data, Recurrent Neural Networks (RNNs) generally fare better. Unlike conventional Neural Networks, where each neuron is a linear combination of the input, Recurrent Neural Networks incorporate contextual information from the past, enabling them to process sequential data by taking into account the relationships between successive elements.

Similarly, Convolutional Neural Networks (CNNs) are primarily designed for the analysis of 2D data such as images, but they can also be adapted to temporal data using an approach called "1D-CNN". Firstly, they are composed of convolution filters sliding along the series to extract local patterns, thus capturing important temporal features. Secondly, they also contain pooling layers that reduce data size while preserving key information. Finally, fully connected layers can be used for prediction.

In this paper, stock return prediction is evaluated on the component stocks of China securities 100 index using Random Forest (RF), Support Vector Regression (SVR), and three Deep Learning models, i.e., an LSTM-type Recurrent Neural Network (RNN), a Convolutional Neural Network (CNN) and a classical Deep Neural Network. This is compared with ARIMA, a classical statistical method introduced in Chapter 2. The aim here is to predict the next daily stock returns on the basis of previous daily stock returns. 9 years of history is used. The final models are compared using several metrics, including MSE and MAE, and it is shown that RF is the best model, followed by LSTM, ARIMA and the other AI methods, with even Deep Neural Networks coming in behind. This result shows that classical Machine Learning can sometimes outperform Neural Networks.

Artificial Intelligence Alter Egos: Who might benefit from robo-investing? [D'Hondt et al., 2020]

In this paper, the authors based themselves on the 20 years historical prices of no fewer than 683 stocks and 393 ETFs. But also on the five Fama-French monthly factors and the Welch and Goyal predictors. The aim was to make monthly stock return predictions. The models used were Moving Average, OLS, Elastic Net, RF and Deep Neural Networks.

Elastic Net [Zou and Hastie, 2005] is a regularized regression technique that combines both L1 (Lasso) and L2 (Ridge) penalization to select important features while managing multicollinearity. It optimizes a cost function by minimizing both the sum of squares of the residuals and the sum of the absolute values of the coefficients, with regularization parameters to adjust the intensity of the penalization. This makes it possible to create simpler models by removing certain features and reducing coefficients, while maintaining more stable models by managing correlations between features.

The authors showed that the 2 best models in terms of MSE were Deep Neural Networks, followed by ElasticNet, with Moving Average being the worst estimator.

Predicting Stock Market Returns with Machine Learning [Rossi, 2018]

In this paper, Alberto Rossi used a classic Machine Learning method called Boosted Regression Trees (BRT) to forecast stock returns and volatility at the monthly frequency. 40 years of history was used. Like Random Forest, boosting is also an ensemble technique in Machine Learning that aims to improve the predictive performance of a model by combining several simple Decision Trees (called "weak learners") to form a more powerful model. Using this method, he showed that his model was able to predict monthly stock returns with better coefficient of determination and directional accuracy than Linear Regression and prevailing mean.

4.2 Forecasting in general

Having presented the forecasting methods used in the financial literature, let's now turn our attention to the latest trends in regression (since forecasting is a regression problem) in Machine Learning and Deep Learning. Indeed, there is a time lag between the development and adoption of Machine Learning models in engineering settings compared to their implementation in financial literature and it's due to several reasons [Dixon et al., 2020]. Firstly, engineering literature requires rigorous

experimentation and validation leading to publication in scientific journals after peer review. While on the other hand, finance researchers often require multiple studies and real-world applications to be convinced of the models' effectiveness. It can also be argued that the limited collaborations and cross-disciplinary interactions between the Machine Learning community and the finance community can slow down the diffusion of knowledge from one field to the other. Finally, the heavy engineering ontology involved is a key barrier to finance practitioners from the quantitative disciplines such as mathematics, statistics and economics. As a result, these challenges of understanding and accessibility lead to misconceptions and limited understanding of Machine Learning capabilities.

That said, with the boom in the size of databases and the explosion in the computing power needed to process them, the recent literature on timeserie forecasting/regression is made up almost exclusively of Deep Learning. In fact, it's easy to find reviews of the methods used, and we're going to focus on 3 of them. First, let's take a look at "Time-series forecasting with deep learning: a survey" [Lim and Zohren, 2021] which presents the types of layers usually used in Neural Networks from a theoretical point of view, without associating any concrete examples. The classical Convolutional and Recurrent Neural Networks discussed above are mentioned. However, a new type of architecture not seen in the financial literature is mentioned: an Attention layer, based on the principle of Attention. When this layer is added to a Neural Network, it becomes a Transformer.

This past year, it was this type of Neural Network that created the hype for AI among the general public. On the one side, Stable Diffusion was released in August 2022 and can be credited with generating astonishing images from text sequences. On the other side, in November 2022, the release of ChatGPT, where the T is a reference to Transformer, provides a mind-blowing conversational agent. When a Recurrent Neural Network analyzes a temporal sequence, it goes through it element by element, maintaining an internal memory to retain previously processed elements of the sequence. Rather than storing information about the entire sequence, Attention weights the elements of the sequence according to their relevance to the element currently being analyzed.

The other 2 method reviews are more practical, highlighting the datasets used and focusing specifically on Deep Learning for renewable energy forecasting and Deep Learning for wind and solar energy forecasting respectively. The first review [Alkhayat and Mehmood, 2021] mentions CNNs and RNNs once again, as well as Stacked auto-encoder and Deep belief network, two methods renowned for their excellence in feature extraction. In this sense, they can be seen as analogous to CNNs. The second [Wang et al., 2019], two years more recent, mentions similar networks but places greater emphasis on combinations of networks, what it calls hybrid/ensemble networks. It shows that this is the type of network most used in the literature since 2018, while also being mentioned more and more, as can be seen in Figure 4.1.

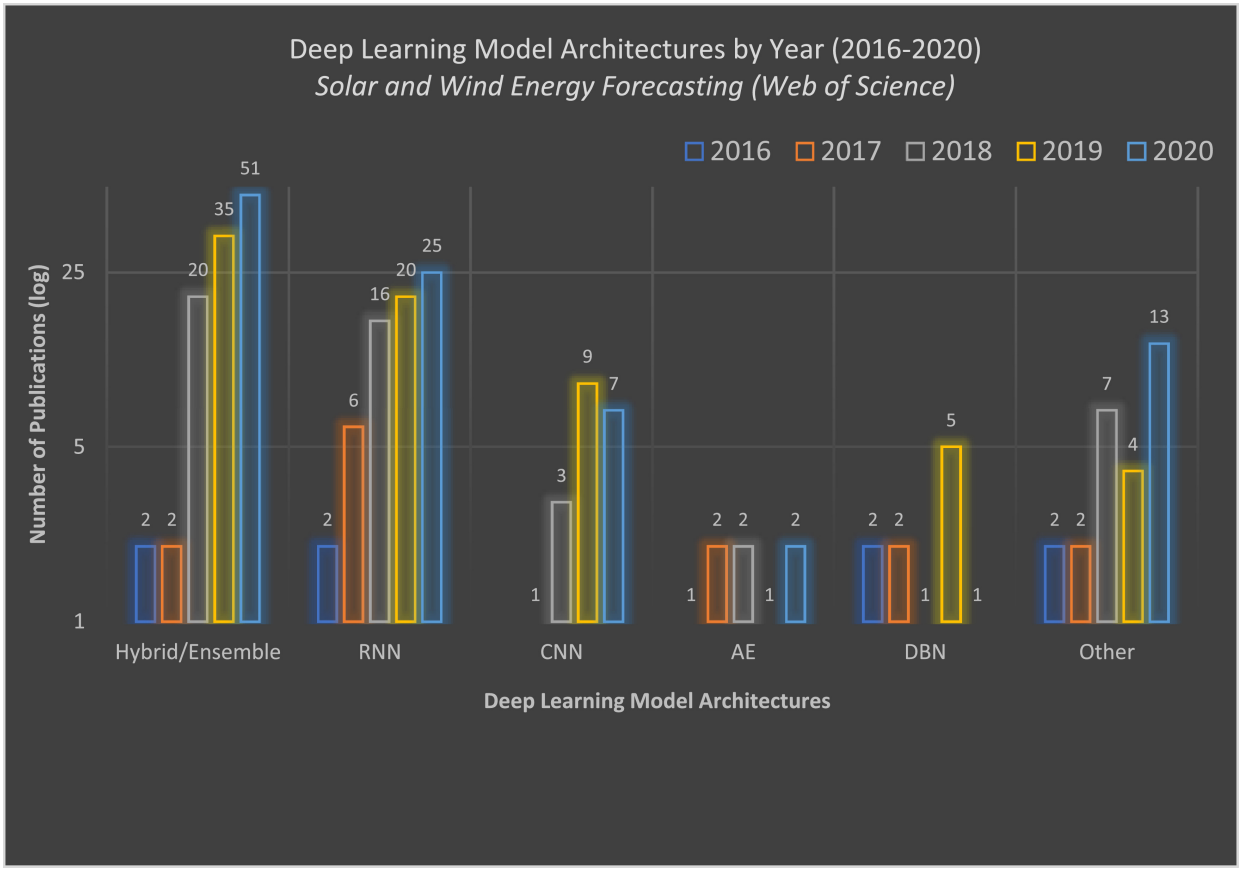

Figure 4.1: Number of publications by Deep Learning Model Architectures.

The second most widely used network type, RNNs, is also on the increase. In fact, of the 45 papers citing forecasting methods for predicting wind energy, 31 use at least one RNN layer. For solar energy prediction, this number rises to 17 out of 22 papers.

It might therefore be tempting to consider only Neural Networks in this study, but that was without counting one type of algorithm which is an exception: tree-based models, particularly when incorporated into an ensemble method as seen in Chapter 2. In "Why do tree-based models still outperform Deep Learning on tabular data?" [Grinsztajn et al., 2022], the authors used 19 datasets from OpenML, a dataset-sharing platform [Vanschoren et al., 2014] to study the possibilities of regression based on tabular data (several features/covariates by timestep) linked to various domains such as bike sharing demand, house sales or temperatures. They showed that a RandomForest method and two boosting methods, GradientBoosting and XGBoost, achieved a better coefficient of determination than Deep Neural Networks such as FTtransformer and SAINT, the two transformers making up SOTA, all in a much smaller number of iterations.

This feature is not restricted to tabular data, and also works on data without covariates where a simple timeseries is given, i.e. as in our case where we rely solely on the history of closes to determine the timeseries of historical returns. In "Do We Really Need Deep Learning Models for Time Series Forecasting?" [Elsayed et al., 2021], a rolling forecast approach is used in which the timeseries is split into smaller overlapping windows in order to predict the value of the timeseries after this window (single-step forecasting). The RMSEs obtained by a boosting model, XGboost, are among the two best models out of six on three of the four datasets studied, despite the fact that an LSTM and two transformer models, two models making up the SOTA in Deep Learning, are part of the benchmark. ARIMA is also mentioned, but is in the bottom two models out of three of the four datasets studied, proving once again its obsolete nature.

4.3 Answer to the first research question

In this chapter, we first saw how historically predominant Neural Networks have been for stock return prediction. This architecture is indeed capable of learning complex, non-linear relationships in the data, making them well suited to forecasting in environments where the relationships between variables are dynamic and non-trivial. We have also seen that certain models, such as LSTMs, which offer smarter layers of neurons than conventional Neural Networks, fare better than the latter on temporal data. We also saw that more traditional Machine Learning methods such as ensemble Random Forest and Gradient Boosting were sometimes the best-performing models, depending on the metrics used.

Next, we looked at forecasting in general and saw that the current trend is almost exclusively linked to Deep Learning. Recurrent networks such as LSTMs are still in vogue, and are in fact the most widely used individual model in the literature, even if the trend is towards ensemble models. We've also seen that a certain type of network is currently fuelling most of the AI-related hype: transformers, based on the Attention principle. To our knowledge, this model has not yet been used in the finance forecasting literature. We have also seen that, in specific cases, such as using rolling forecasts, RF and boosting methods can still perform better than Deep Learning models.

As we have to make choices and it's not possible to explore everything, from all the methods we've seen, we'll make the choice to keep RF and GB as Machine Learning methods and LSTMs and TFs as Deep Learning methods. RF is chosen because it is almost always mentioned in papers as a Machine Learning baseline. GB, on the other hand, is only rarely mentioned in the financial literature and, as far as we know, has not yet been compared with Deep Learning models. And yet, according to the timeserie forecasting literature, it's worth the effort. Secondly, LSTMs are the most widely used individual Deep Learning method in timeserie forecasting, and were the best-performing architecture for time-series data before the arrival of Transformers. So they are an obvious choice. Finally, TFs will be chosen because they are linked to the current trend in Deep Learning and are not yet included in the financial literature. They thus represent a strong added value in this dissertation. It should be noted that, unlike many approaches in the literature, we will only use historical prices and not various economic features to predict future returns, in order to compare the algorithms in a simpler way and thus limit computation time.

Chapter 5

Quantitative methodology

To answer the 2 other research questions defined above, we will carry out a quantitative study. To do this, we need historical price data and, more specifically, historical returns data. Section 5.1 covers the various steps involved in obtaining these historical returns. Once these have been obtained, section 5.2 then illustrates how future expected stock returns can be determined on the basis of them. Finally, based on the future expected stock returns, section 5.3 will show how our research questions can be answered.

5.1 Data

In this chapter, in section 5.1.1, we'll define the programming environment and language. This will condition our future choices in terms of data sources and analysis. Then, in section 5.1.2, we'll determine from which sources to retrieve our data. Once we've done that, in section 5.1.3, we'll define the type of data we need: how frequent is it? What time history do we need? How much stock do we need? Finally, we'll be able to select our stocks and calculate their returns, taking care to avoid certain biases in section 5.1.4. This section will also present the main statistical properties of the returns of the stocks making up our investment universe.

5.1.1 Processing

In the past, computer development was carried out on personal computers. However, programming languages have evolved and are now so sophisticated that they pose a technological barrier to use. If we take the example of the Python language and data science, there are a huge number of libraries to install, all of which have versioning constraints with each other, and it's not easy to find your way around.

To solve this problem, development is now increasingly carried out in cloud computing [Kim and Henke, 2021] environments offered by Google, Amazon or Microsoft, where it is possible to access numerous pre-installed libraries in a few clicks, in user-friendly interfaces such as Jupyter notebooks. Cloud computing also means access to computing power and RAM far superior to that of a personal computer, both of which are essential to learning models such as those proposed in Artificial Intelligence. In this dissertation, we've chosen to use the cloud environment offered by the online platform Kaggle, which provides free access to its GPU computing power for 30h/week [Becker, 2019], more than sufficient for the purposes of this dissertation. The programming language used will be Python. In the past years, this language gained significant popularity due to its versatility, ease of use, and a rich ecosystem of libraries and tools that are well-suited for various data science tasks. It's now the preferred language for data science.

5.1.2 Sources

To answer our research questions, we need historical price data for several stocks, and more precisely the Adj Close, the daily closing value taking dividends and splits into account. As the dissertation is being developed in Python, several APIs (Application Programming Interface) are possible for obtaining this data. To carry out our analyses, we'll need 20 years' worth of data on 20 historical stock prices (this will be explained later). Since one year is made up of approximately 250 trading days, this represents approximately 250x20x20 points to be retrieved, or 100,000 points. The 2 most important criteria for choosing which API to use are therefore the number of free requests we can send and the complexity of the API (based on the author's appreciation). These and their main characteristics are listed in the Table 5.1 below:

Source	Free Python API Limitations	Python API complexity
MorningStar	500 requests/months	Medium
Eikon	No limit/month but 3000 points/request	Medium
Bloomberg	Not free	High
Google Finance	500 requests/day	Low
Yahoo Finance	2000 requests/h	Low

Table 5.1: Python Finance APIs.

Looking at this table, we might be tempted to choose between Eikon and Yahoo Finance, which have the 2 best practical features combined. However, the fact that Eikon limits the number of points per query is rather restrictive, as it means that you have to segment the queries at the beginning and reassemble them at the end, which adds a constraint from a development point of view. We will thus choose Yahoo Finance.

5.1.3 Desired properties

In this section, we will define the precise type of data we need to answer our research questions. As we have seen from the various authors in the literature, more or less substantial data histories, ranging from 9 years to 40 years, are required to train algorithms for predicting stock returns on a daily/weekly/monthly basis on one to several hundred stocks.

Authors with higher frequency use fewer stocks and/or less history. Conversely, authors using a longer history use fewer stocks and/or a lower frequency, and authors using more stocks use a lower frequency and/or a shorter history. So, for a given computing power, a trade-off must be found between the length of the history we need for each stock, the frequency of this history and the number of stocks making up our universe of assets. Frequency is probably the most impactful of the 3. When we go from quarterly to monthly frequency, the amount of data is tripled. It then quadruples when we move to a weekly frequency and quintuples when we move to a daily frequency. For the other 2 terms, the granularity is much finer and we can afford to add one stock at a time or one year at a time, without significantly impacting the amount of data to be processed.

Frequency

The literature mostly uses a monthly frequency in stock return prediction, for several reasons. One the one hand, choosing a monthly frequency allows a trade-off between having a large number of data for analysis and minimizing the impact of random fluctuations [Helms et al., 2022] since these generally have a normal distribution [Pham et al., 2020]. On the other hand, most authors use

macroeconomic data [Jiang et al., 2019] in addition to historical prices to make their predictions. These indicators are generally available on a monthly basis.

Since we're focusing on historical prices, the second argument doesn't hold in our case, and we might be tempted to aim for a different frequency. However, in our case, the first argument particularly resonates. Indeed, AI models can capture non-linearity in the data via a greater number of parameters, and are therefore better the more data there is. However, this large number of parameters is also a handicap, as calculation time is considerably increased. Preliminary tests with 20 years of history at a monthly frequency showed that for the most sophyticated AI model, around 30 minutes were needed to learn the best combination of hyperparameters for a given stock. For 20 stocks, we arrive at 10h of training. Then, there's the calculation time for the other methods, and the portfolio optimization time for each method... Given the effectiveness of Artificial Intelligence on a large amount of data, using a weekly or daily frequency would probably lead to slightly better results, but for a training time that would be multiplied by 4 and 20 respectively. As the research question is aimed at comparing classical and Artificial Intelligence methods, and not at obtaining the best possible models, we feel that the planned training time is sufficiently long.

While this second argument is particularly valid, in order to compare ourselves more easily with the results of other authors, and for ease of transition to MPT by rebalancing every month after predicting expected stock returns each month, we'll also be choosing a monthly frequency.

History size

For the history size, 20 years (2003-2022) seemed to us to be a good compromise, relevant and in line with the literature. Taking a longer timeframe would mean taking into account economic, geopolitical and technological conditions that do not reflect current market conditions, making present-day decision-making complex: Investors need up-to-date information to make informed decisions. Also, while historical data can provide insights into trends, past performance is not indicative of future result. Moreover, too much data would make learning even more complex. Taking a shorter history would mean not having enough data to understand the non-linearity present in the data, as we saw in chapter 2. Another factor in favor of the 20-year timeframe is that it covers a period of 5 years before the 2008 crisis, the crisis in question and then the post-crisis period. This means that our models will learn to account for and predict such extreme events, which are crucial for risk management and portfolio optimization. It should also be noted that the out-of-sample period (as we will see later) covers the period before/during/after the Covid crisis, and thus the economic bubble of 2022.

Number of assets

Finally, we have chosen an investment universe made up of 20 stocks. This may not seem much in comparison with the literature, which systematically looks at around a hundred stocks. However, the other 2 elements of the trade-off imply a fairly substantial amount of data, given our modest computing power. Since our aim is to produce a proof-of-concept for a dissertation, and not to create a robo-advisor covering all the stocks making up the real asset universe, this number seemed sufficient to us.

5.1.4 Selection

To carry out our analysis, we need to select data from all existing stocks. Selecting data always induces bias. Let's take a look at the main biases involved in data selection in our case, and see how we can mitigate them. Bias examples are taken from [Nikolopoulou, 2022].

1. First of all, sample bias. This consists of selecting a subset of data that is not representative of the entire market for analysis, and therefore rejecting part of the population because of the unavailability of data.

2. Secondly, time period bias may arise if the chosen data period is not representative of the stock market's typical behavior.

3. Also, survivor bias. This is a bias that occurs when only surviving entities are considered in a study, while failures are ignored. This bias leads to an incomplete or skewed representation of the overall population, as it only includes the data of entities that have "survived" a certain period.

4. Finally, the look-ahead bias. This bias arises when information not available at the time of analysis is used.

The first 2 biases cannot be eliminated, and we'll have to take them into account. For the sample bias, the need to collect 20 years of historical data means that we will have to reject all companies with less than 20 years of existence. However, among large caps, there are many companies whose IPO dates back less than 20 years. Alphabet (2004), Meta (2012) and Tesla (2010), for example, are all now in the top 10 of the Standard & Poor's 500 (S&P500), a stock market index tracking the stock performance of 500 of the largest companies listed on stock exchanges in the United States. Their influence on the markets, particularly in the tech sector, is non-negligible. By not selecting these companies, we ignore significant players and miss part of the current trend towards them.

Likewise, with regard to time period bias, selecting 20 years of data means that our analysis will only be valid for those 20 years, and not for an earlier history, nor for predicting the future. Also, as we shall see later, our out-of-sample analysis (test set) will cover the period from January 2019 to December 2022. Although it's interesting to have a test set covering the Covid period to see how the algorithms behave in a crisis, the results would probably not have been the same over another period, although we'll never know.

However, the other 2 biases can be mitigated. For the survivor bias, we will only select large-cap companies. Indeed, small-cap companies are more likely to bankrupt for several reasons [Investopedia, 2023] On the one hand, they experience greater volatility. On the other hand, they have less financial stability than large-cap companies, and may also have liquidity issues. We will also be using point-in-time data. This is a method whereby analysis is performed at a certain point-in-time, regardless of subsequent developments or outcomes. The entire population is therefore covered, whether surviving entities or not. This 2nd method also eliminates the look-ahead bias [Perspectives, 2020].

To select large-cap companies, S&P500 is an ideal dataset. To further reduce survivor bias, we've chosen to select the top 20 most-weighted companies in the S&P500 (a sign that their capitalization is among the largest) with at least 20 years of data available (before 2023) on Yahoo Finance (whose Initial Public Offering = IPO dates back at least 20 years). What's more, we're not going to take the ranking based on the index at the start of 2023, but rather that of early 2019 [CodingFinance, 2018] (the start of the Out-of-sample period, as we'll see later), the S&P500 being revised periodically according to changes in large-cap capitalizations. As a result, we won't know how companies fared afterwards (whether they survived or not).

To obtain these monthly returns from daily prices, we choose to define a month's return as the difference between the stock price on the first Wednesday of the month and the stock price on the first Wednesday of the previous month, the whole divided by the second stock price. Indeed, this day of the week is chosen to avoid stock market fluctuations at the start of the week and as the weekend approaches [Wolff and Echterling, 2020]. Here is the process summarized in Fig 5.1 for a single stock:

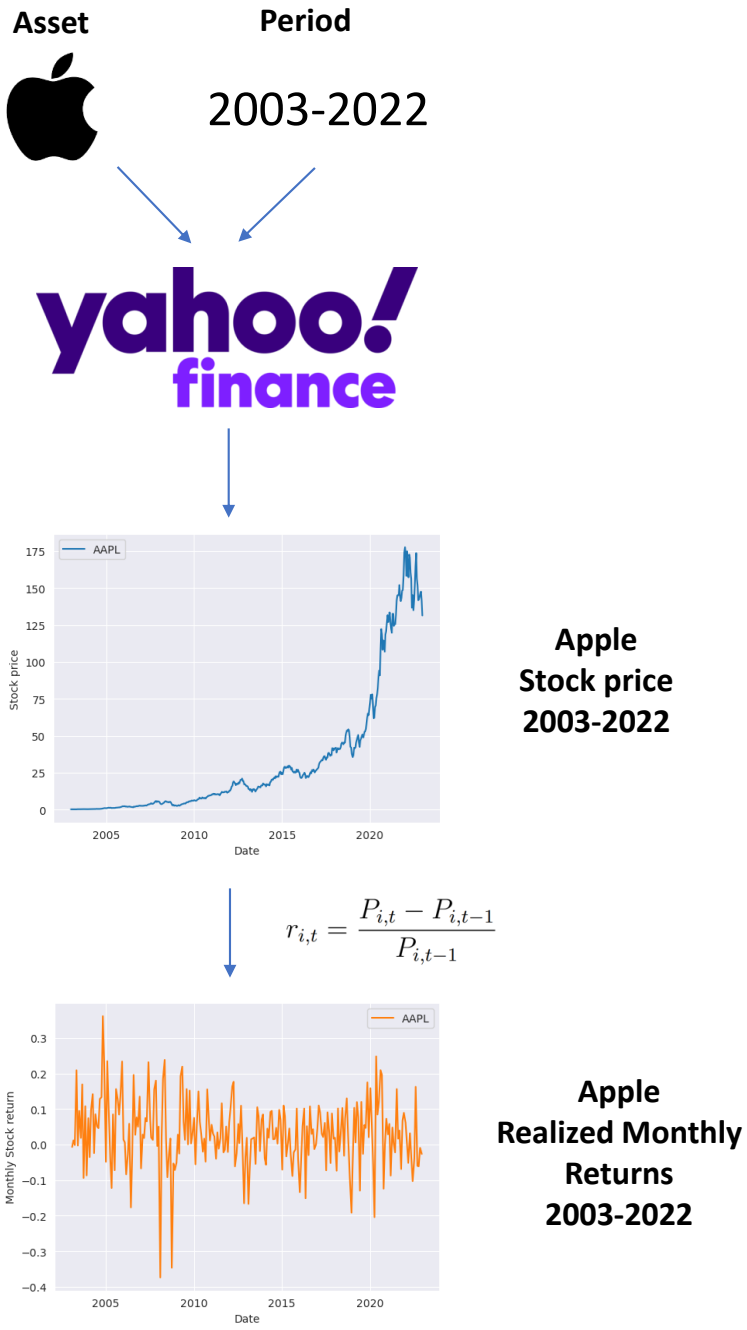

Figure 5.1: Process to create Realized Monthly Returns for a given stock and a given period.

In Table 5.2, here are the stocks we've chosen in the S&P500, along with some of their statistical properties:

Company	Ticker	Mean(%)	Median(%)	Volatility(%)	MDD(%)
Microsoft corp	MSFT	1.326	1.278	6.418	-53.309
Apple Inc	AAPL	3.197	2.791	9.556	-53.29
Amazon.com Inc	AMZN	2.401	3.284	10.371	-54.018
Berkshire Hathaway Inc Cl B	BRK-B	0.923	0.45	5.298	-50.051
Johnson + Johnson	JNJ	0.809	1.027	4.342	-30.194
JPMorgan Chase + Co	JPM	1.264	1.23	8.635	-60.859
Exxon Mobil Corp	XOM	0.926	0.738	6.276	-57.906
Pfizer Inc	PFE	0.724	0.598	6.131	-59.361
Unitedhealth Group Inc	UNH	1.74	1.894	7.465	-71.258
Verizon Communications Inc	VZ	0.522	0.214	5.045	-33.21
Procter + Gamble Co/The	PG	0.85	0.931	4.328	-33.928
Bank of America Corp	BAC	1.059	0.593	13.572	-92.463
Intel Corporation	INTC	0.727	0.849	7.569	-58.66
Chevron Corp	CVX	1.23	0.847	6.849	-42.631
AT&T	T	0.73	0.803	5.904	-39.709
Wells Frago & Company	WFC	0.967	0.977	10.321	-72.924
Merck + Co. Inc	MRK	0.817	0.927	6.349	-58.556
Cisco Systems Inc	CSCO	0.923	0.714	7.422	-53.435
Home Depot Inc	HD	1.564	1.163	6.817	-51.868
Coca Cola Co/The	KO	0.813	0.88	4.892	-35.162

Table 5.2: Stocks, Tickers and the Mean, Median, Volatility, Maximum Drawdown (MDD) of their realized monthly return.

From the above table, we can see that AAPL and AMZN are the 2 best-performing stocks over the last 2 years: the former in terms of mean and the latter in terms of median. However, these 2 stocks are in the top4 in terms of volatility. Conversely, stocks such as JNJ and PG have the lowest volatility but average realized returns among the 8 lowest. This is in line with what we said earlier: taking more risk generally pays off in terms of returns, and vice versa. BAC is the most volatile stock and has the highest Maximum Drawdown (MDD), following the 2008 financial crisis [Lee, 2022].

5.2 Prediction

Once we've transformed our daily historical prices into monthly returns, we can get to the heart of the matter. For each stock, the idea is to determine an input vector used to predict an output vector. A classic approach in the literature [Abe and Nakayama, 2018], [D'Hondt et al., 2020] is to consider a sliding window (of tunable size) containing the previous monthly returns over a certain number of months in order to predict the next monthly return, the window then being shifted by one month successively until we reach the end of the time horizon (this is similar to Rolling-sample Markowitz). Following preliminary tests, a window size of 24 months proved to be the optimum size for gaining sufficient insights into trends, and at the same time to get sufficiently recent insights to be relevant.

Once this is done, we have our inputs and outputs. All that remains is to separate these into an in-sample set (Training set) for training Machine Learning models to learn historical return trends, and an out-of-sample set (Test set) for measuring performance on unseen data. For non Machine Learning models, only the second set will be used. In practice, rolling windows with an output (expected monthly return to predict) from January 2019 onwards will be part of this out-of-sample, which corresponds to a 78%/22% split, not far from the 80%/20% proposed in the literature [Detective, 2021].

This means that our methods will predict expected returns from January 2019 to December 2022, and that these will be compared with the realized returns actually observed. The dataset creation process is illustrated in Figure 5.2.

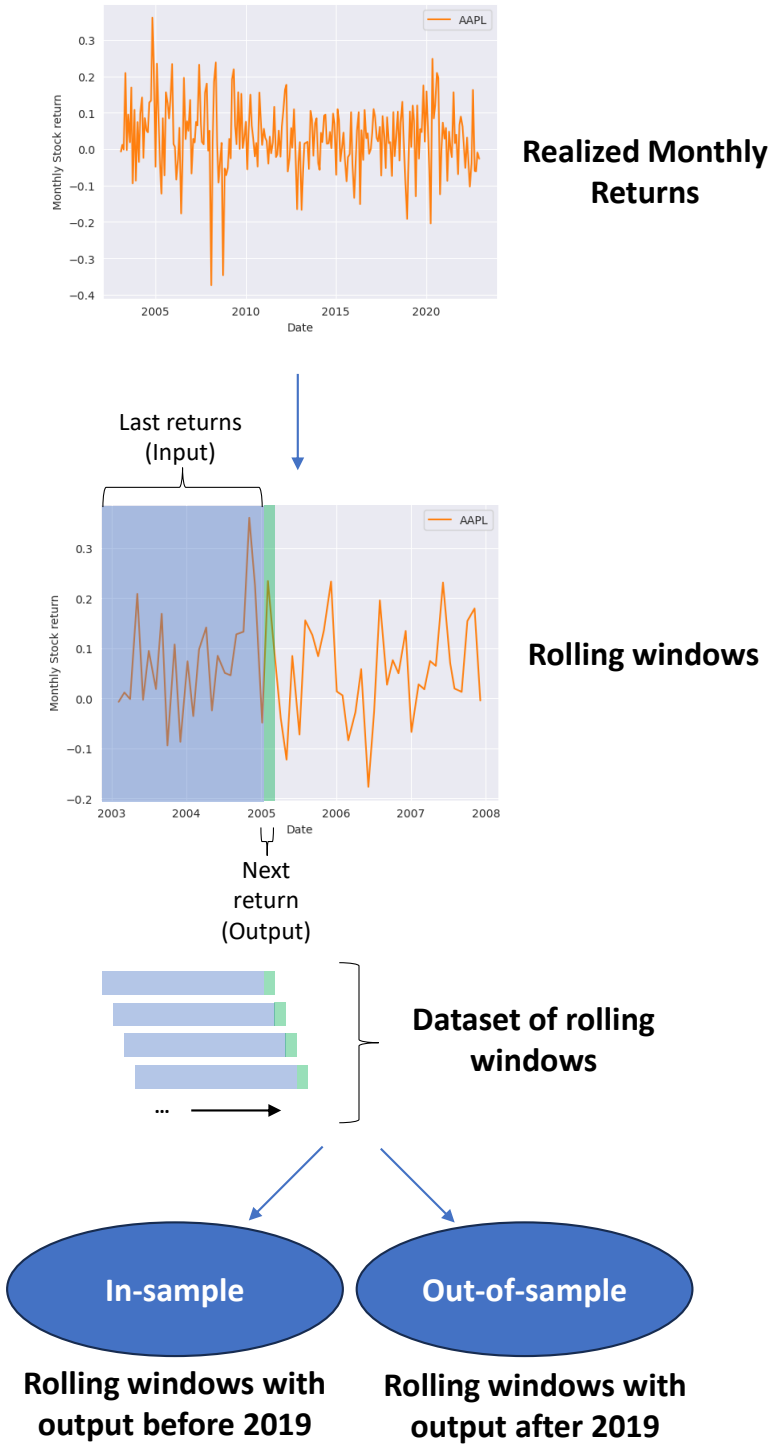

Figure 5.2: Process to create Dataset.

Here are the stocks we've chosen in the S&P500, along with some of their statistical properties. First for the training set, in Table 5.3:

Company	Ticker	Mean(%)	Median(%)	Volatility(%)	MDD(%)
Microsoft corp	MSFT	1.148	1.278	6.204	-53.309
Apple Inc	AAPL	3.233	2.744	9.575	-53.29
Amazon.com Inc	AMZN	2.864	3.568	10.579	-54.018
Berkshire Hathaway Inc Cl B	BRK-B	0.887	0.366	4.982	-50.051
Johnson + Johnson	JNJ	0.821	1.062	4.008	-30.194
JPMorgan Chase + Co	JPM	1.285	1.126	8.675	-60.859
Exxon Mobil Corp	XOM	0.745	0.718	4.786	-33.933
Pfizer Inc	PFE	0.649	0.619	5.424	-59.361
Unitedhealth Group Inc	UNH	1.711	2.041	7.479	-71.258
Verizon Communications Inc	VZ	0.758	0.498	5.15	-33.21
Procter + Gamble Co/The	PG	0.731	0.75	4.101	-33.928
Bank of America Corp	BAC	0.999	0.506	14.354	-92.463
Intel Corporation	INTC	1.03	1.338	7.261	-58.66
Chevron Corp	CVX	1.103	1.143	5.724	-37.59
AT&T	T	0.754	0.803	5.478	-39.709
Wells Frago & Company	WFC	1.068	1.233	10.275	-72.924
Merck + Co. Inc	MRK	0.706	0.851	6.454	-58.556
Cisco Systems Inc	CSCO	1.009	0.722	7.345	-53.435
Home Depot Inc	HD	1.5	1.163	6.404	-51.868
Coca Cola Co/The	KO	0.762	0.788	4.428	-35.162

Table 5.3: Stocks, Tickers and the Mean, Median, Volatility, Maximum Drawdown (MDD) of their realized monthly return (2003-2018).

Then for the test set, in Table 5.4:

Company	Ticker	Mean(%)	Median(%)	Volatility(%)	MDD(%)
Microsoft corp	MSFT	2.035	1.198	7.232	-33.55
Apple Inc	AAPL	3.055	3.012	9.578	-24.877
Amazon.com Inc	AMZN	0.56	0.066	9.372	-52.14
Berkshire Hathaway Inc Cl B	BRK-B	1.07	1.221	6.464	-24.61
Johnson + Johnson	JNJ	0.759	0.724	5.524	-15.818
JPMorgan Chase + Co	JPM	1.183	1.811	8.561	-38.687
Exxon Mobil Corp	XOM	1.644	1.201	10.306	-55.113
Pfizer Inc	PFE	1.024	-1.191	8.443	-26.38
Unitedhealth Group Inc	UNH	1.856	1.341	7.485	-19.45
Verizon Communications Inc	VZ	-0.418	-0.761	4.53	-33.21
Procter + Gamble Co/The	PG	1.323	1.011	5.153	-19.935
Bank of America Corp	BAC	1.299	0.997	9.989	-43.105
Intel Corporation	INTC	-0.478	-0.668	8.665	-56.704
Chevron Corp	CVX	1.734	0.493	10.235	-42.631
AT&T	T	0.637	0.791	7.431	-32.178
Wells Frago & Company	WFC	0.567	-0.441	10.602	-57.486
Merck + Co. Inc	MRK	1.259	2.106	5.957	-16.058
Cisco Systems Inc	CSCO	0.58	0.254	7.791	-32.519
Home Depot Inc	HD	1.818	0.936	8.333	-29.503
Coca Cola Co/The	KO	1.018	1.945	6.472	-27.886

Table 5.4: Stocks, Tickers and the Mean, Median, Volatility, Maximum Drawdown (MDD) of their realized monthly return (2019-2022).

Let's move on to the methods we're going to study. As seen in the context, there are various classical methods. In this dissertation, we will choose to use the Moving and the Exponentially Weighted Moving Average. Regarding AI methods, based on the literature review, we have decided to select Random Forest and Gradient Boosting as Machine Learning methods, and LSTM and Transformers as Deep Learning methods. These different AI methods will be presented progressively in next chapter. Given the Input-Output separation presented above, it is implied that our methods work in a rolling fashion.

5.3 Research questions answers

To answer research question 2, for each method, for each stock, we can then simply compare realized returns with expected return predictions over the 4 years of the test set, month by month (48 values), using Mean Squared Error (MSE) and Mean Absolute Error (MAE) measure. A summary table will complete the analysis by showing the MSE/MAE ranking of each model for each stock. The process is shown graphically in Figure 5.3.

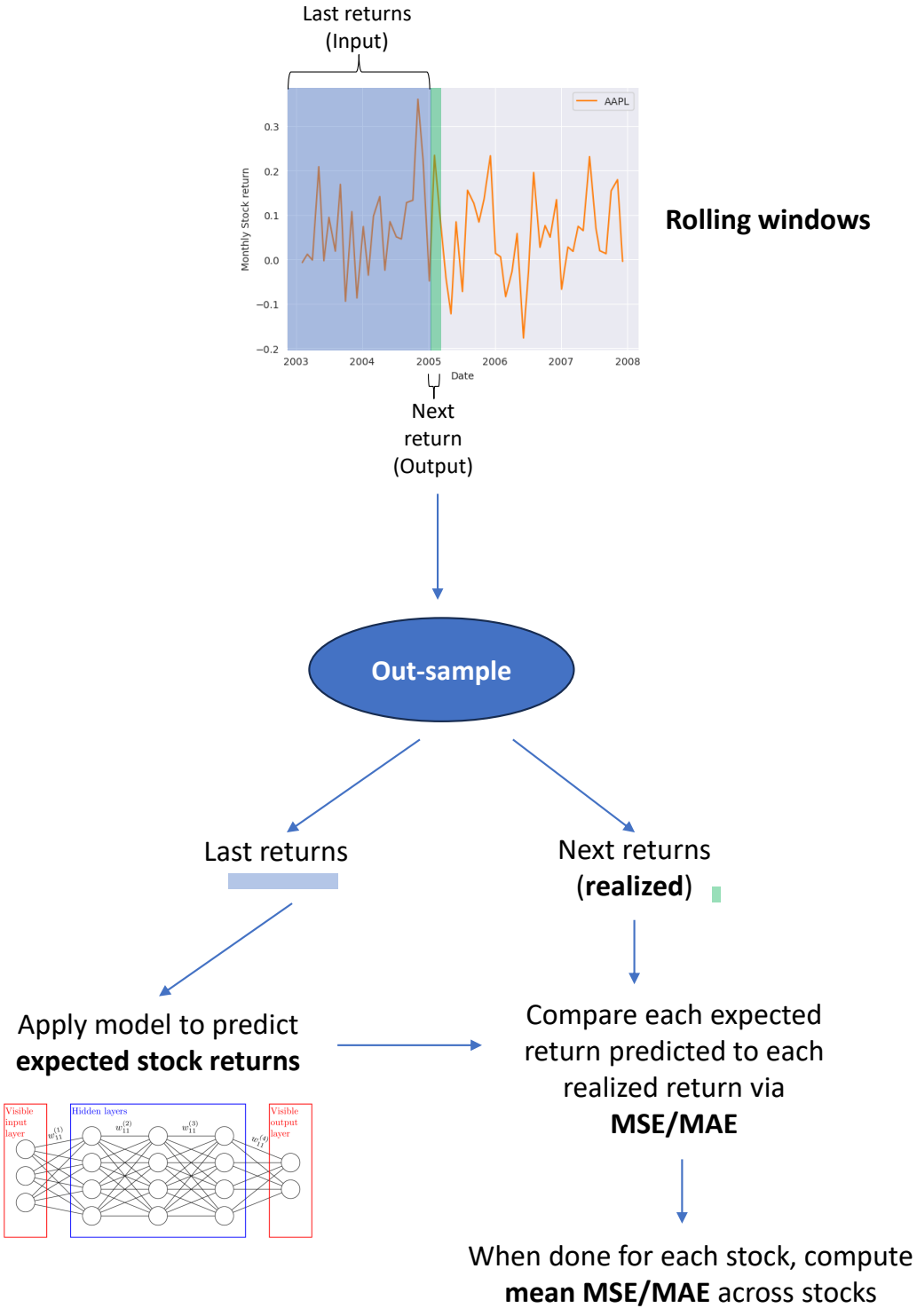

Figure 5.3: Research Question 2 Methodology.

Finally, to answer research question 3, for each method, for each month of the test set, the expected return for each stock will be taken and injected into the MPT to determine the weights maximizing the Sharpe ratio of a portfolio made up of these stocks. These weights will then be applied to the realized stocks returns over the various months, to calculate the portfolio's realized return each month. A graph showing the cumulative realized return of the portfolios constructed will complete the analysis. The process is shown graphically in Figure 5.4.

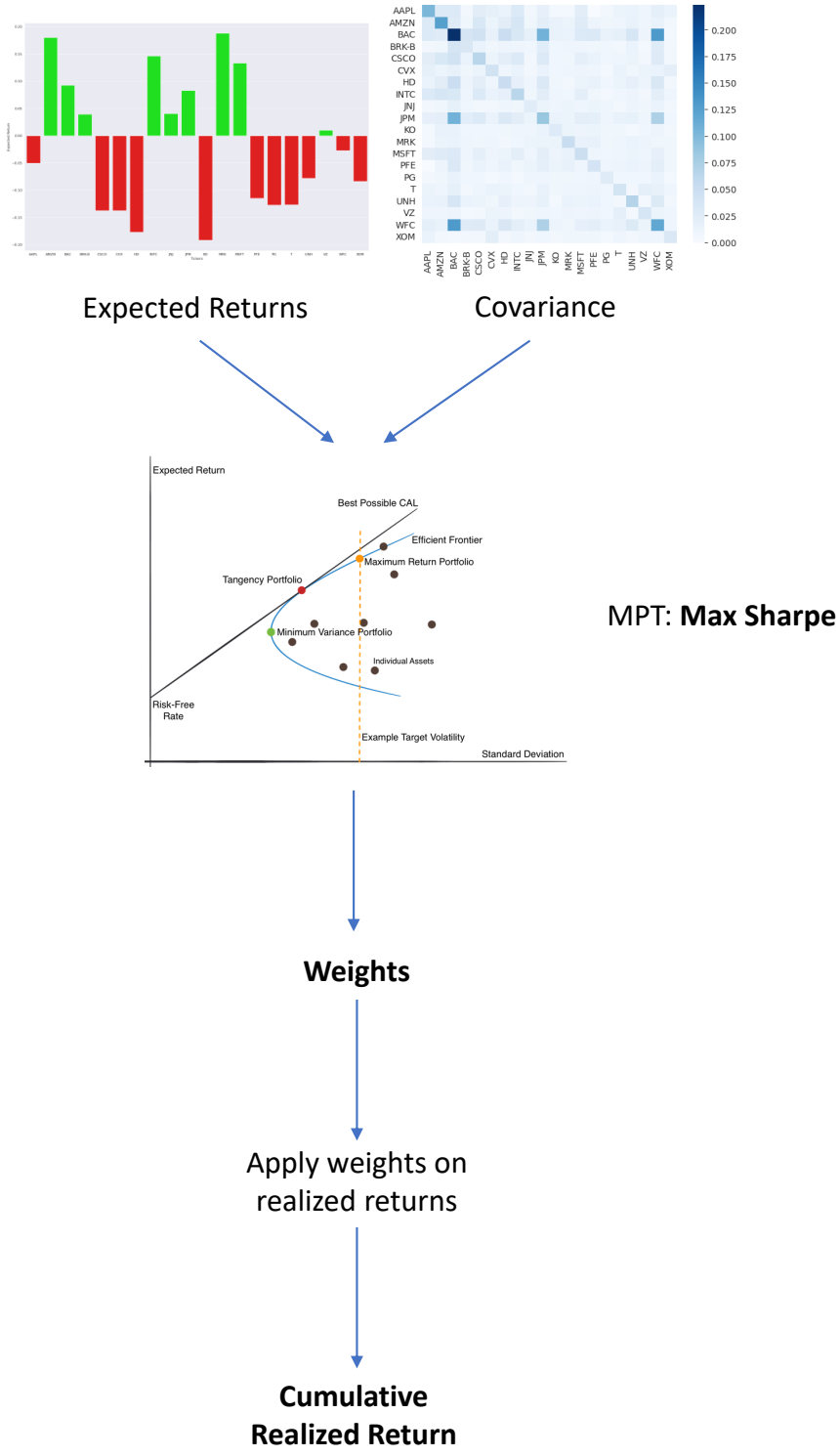

Figure 5.4: Research Question 3 Methodology.

Chapter 6

Developments and Results

In Chapter 2, we presented 2 classical methods for predicting expected stock returns. In Chapter 4, we explored the literature and chose Random Forest, Gradient Boosting, Long Short-Term Memory and Transformers as the 4 methods representing the state of the art in Machine Learning and Deep Learning for predicting stock returns. In this section, we will take a closer look at these different methods before using them as part of the methodology defined in Chapter 5. First, we'll look at Machine Learning (section 6.1) and then Deep Learning (section 6.2) methods, in each case versus classical methods. We believe that this dual comparison is advantageous. At the end of the chapter, we'll answer the last 2 research questions in section 6.3 and 6.4.

These different models will take the place of the "Apply model to predict expected stock returns" box in Figure 5.3, with the rest of the methodology remaining the same. For a given model, once these hyperparameters have been presented, they will be automatically tuned via *RandomizedSearchCV* of *scikit-learn*. [Pedregosa et al., 2011]. *PyPortfolioOpt* [Martin, 2021] will be used to perform MPT.

6.1 Machine Learning methods

As we saw earlier, Random Forest and Gradient Boosting are the Machine Learning methods widely used in forecasting. They are actually two ensemble methods: an approach that combines several individual models to improve generalizability/robustness, but from two different families:

- Random Forest (RF) is a bagging method: several independent models are trained on data samples drawn with replacement from the training set. RF uses Decision Trees as individual models and outputs the mean of their predictions. The variance of RF is then lower than that of the individual Decision Trees.
- Gradient Boosting (GB) is a boosting method: several models are also combined, but each new model seeks to correct the errors of the previous models. The final model is obtained by weighting the predictions of the individual models according to their performance. The aim is therefore to combine several weak models to produce a powerful whole.

Both models use Decision Trees as individual models. This is a widely used Machine Learning model for decision making. A Decision Tree is a tree representation in which each internal node represents a decision based on a particular feature, and each leaf represents a class or output value. The following example, shown in Figure 6.1, makes it easy to understand the risks of preventing a heart attack.

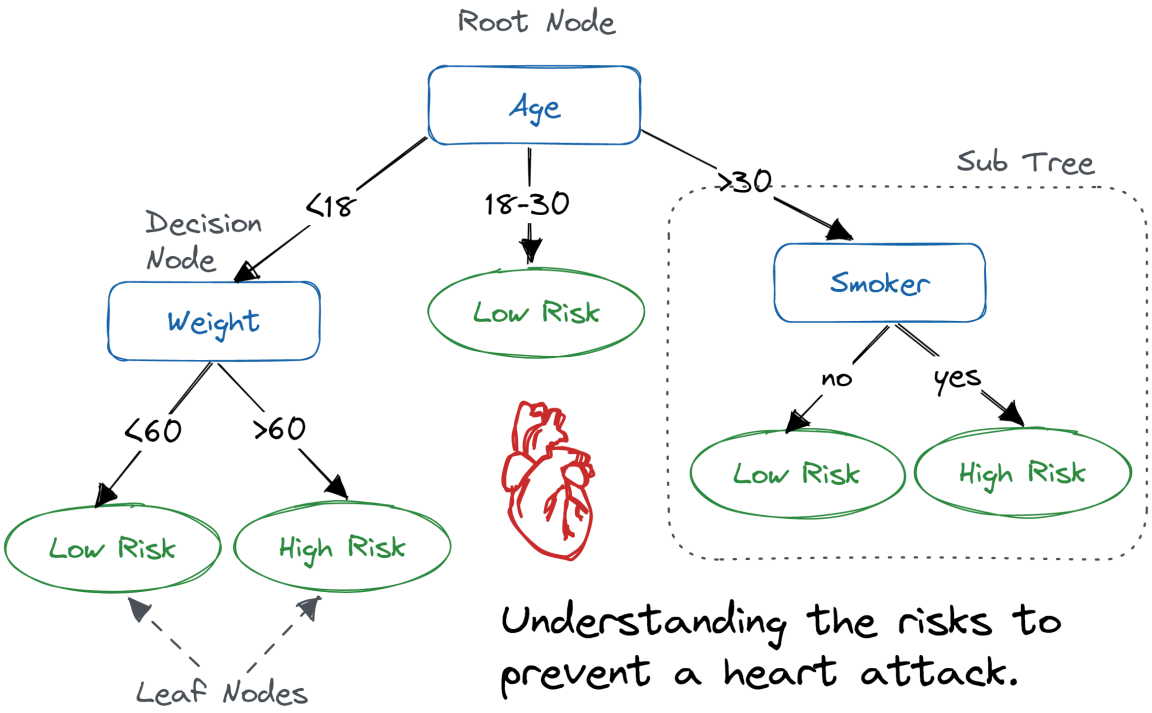

Figure 6.1: Decision Tree to understand the risks to prevent a heart attack [Navlani, 2023].

In this example, the DT was built on the basis of medical knowledge. In Machine Learning, it will be built automatically on the basis of timeseries characteristics linked to historical returns, in order to obtain future returns. Now that we've introduced the Decision Tree, let's take a closer look at these models and see how they can be used.

6.1.1 Random Forest

Principles

The RF principle is as follows: [Geurts, 2017]:

1. Input (historical returns) and output (future returns) pairs are randomly drawn from the training set with replacement. After each draw, these examples are put back into the training set so that they can be selected again.
2. For each split, a Decision Tree is trained to establish the link between inputs and outputs, which is then learned. In learning, for each split, rather than choosing all the input features, we choose the best split on a random subset of them.
3. For a given input, the output predicted by RF is the average of the returns predicted by all the trees.

Decision trees have a high variance and are highly subject to overfit. In the first 2 steps, we therefore use 2 sources of randomness to reduce the drill variance. An illustration of the process is shown in Figure 6.2.

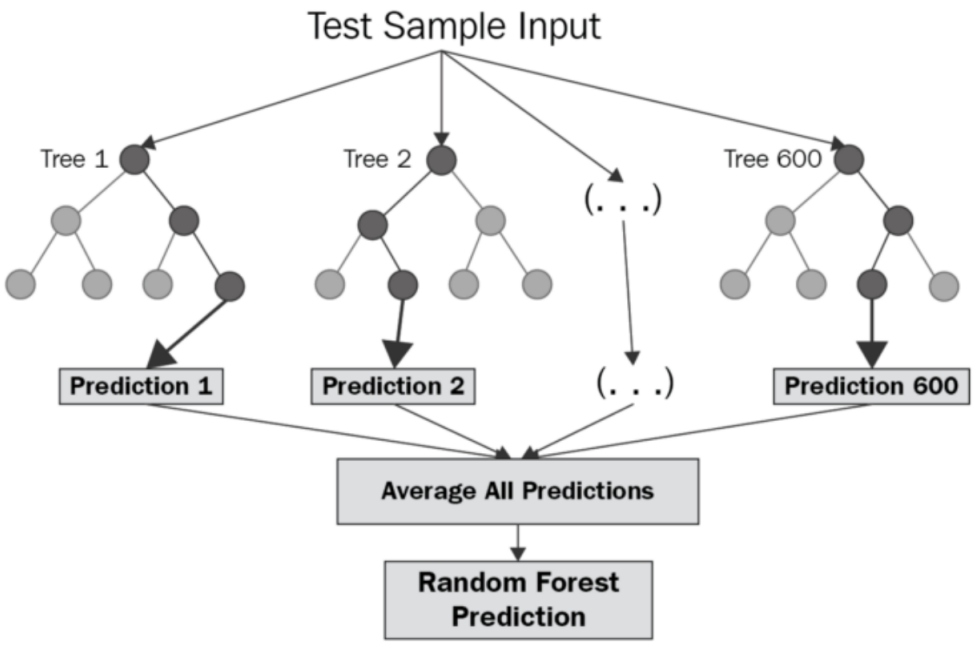

Figure 6.2: Random Forest process [Keboola, 2020].

Implementation

Since our task is a regression problem, we'll be using RandomForestRegressor from *scikit-learn*. [Pedregosa et al., 2011]. The *scikit-learn* implementation has 19 hyperparameters that can be tuned. As more and more hyperparameters with different values are used, the total number of combinations to be tested increases rapidly, making it impossible to test everything. So we had to make some choices. We used the guidelines of [scikit learn, 2020] and [Meinert, 2019] listing parameters that may influence the results. We finally chose to tune the parameters present in Table 6.1. Their intuition is also presented.

Parameters	**Definition**	**Intuition**
n_estimators	Number of Decision Trees to be built in the ensemble	Increasing the number of trees improves model performance, but also increases computing time. However, the results no longer increase significantly after a certain number of trees. Too few trees leads to underfitting, but as described in the original paper, Random Forest don't overfit with increasing number of trees [Breiman, 2001].
max_features	Maximum number of features to consider when searching for the best division for each tree node	Reducing the number of features reduces overfitting but increases underfitting.
max_depth	Maximum depth that each tree can reach	Limiting depth prevents trees from overfitting the data, but a depth that is too shallow won't capture complex relationships in the data (underfitting).

Table 6.1: Main hyperparameters Random Forest.

6.1.2 Boosting methods

Principles

The GB principle is as follows [Geurts, 2017]:

1. The GB generally starts by training a shallow Decision Tree, hardly better than random guessing. This model will contain errors, corresponding to the difference between predicted and actual values.
2. In each subsequent iteration, a new weak learner is trained on the same dataset, with the aim of minimizing the errors of the previous step. Its contribution is controlled by a learning rate
3. The predictions from all the weak learners are combined in a weighted sum, where the weights are determined based on the performance of each model. Models that perform better get higher weights.

An illustration of the process is shown in Figure 6.3.

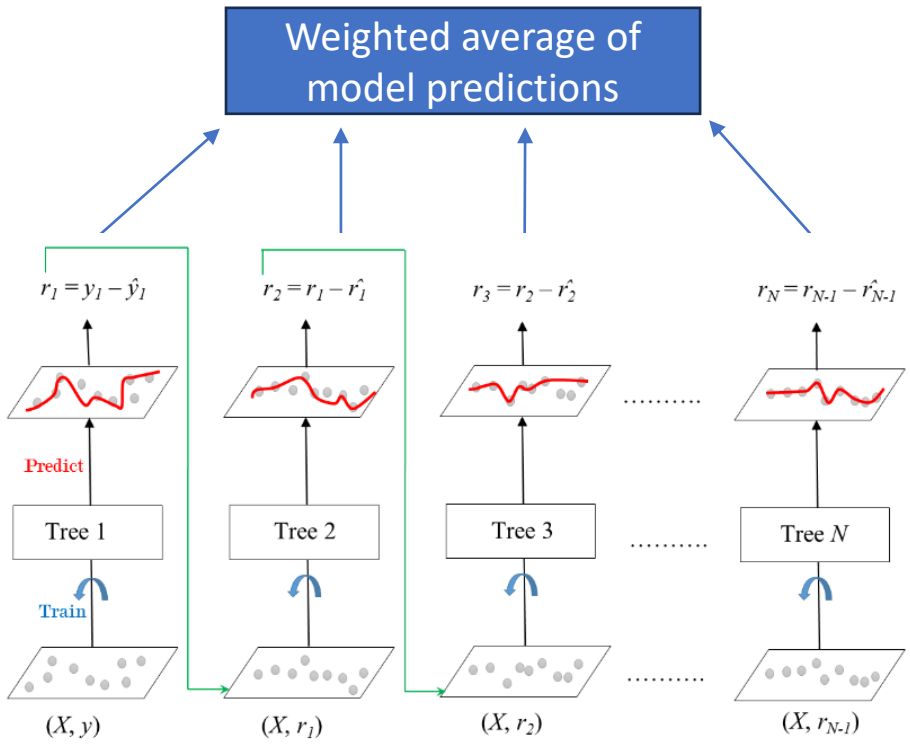

Figure 6.3: Gradient Boosting process [Datacamp, 2023].

Implementation

Once again, we've chosen the *scikit-learn* implementation. This time, we'll be using the developer guidelines *scikit-learn* [Inria, 2022]. Finally, we've chosen to tune the parameters present in Table 6.2. Their intuition is also presented.

Parameters	Definition	Intuition
max_depth	Maximum depth that each tree can reach	Unlike RF, we want the max depth to be low because we want weak learners. However, a value that is too low is still associated with underfitting, while a value that is too high is associated with overfitting.
n_estimators	Number of Decision Trees to be built in the set	Lowering max depth means each tree might need more iterations to capture complex patterns effectively. As a result, increasing the number of estimators can help compensate for the reduction in the depth of individual trees. A value that is too low, however, remains linked to underfitting, while a value that is too high remains linked to overfitting.
learning_rate	Contribution of each weak learner	A low learning rate slows down the learning process, which means that the model needs more iterations to fit the training data accurately. This can lead to underfitting. A high learning rate speeds up the learning process, allowing the model to quickly adapt to the training data. However, this can also make the model more prone to overfitting.

Table 6.2: Main hyperparameters Gradient Boosting.

6.1.3 Results

The MSEs per stock for the 2 classical methods and the 2 ML methods chosen are shown in Table 6.3, while the MAEs are shown in Table 6.4. For each stock, green highlights the lowest MSE/MAE, while red highlights the highest MSE/MAE.

Company	Ticker	MA	EWMA	RF	GB
Microsoft corp	MSFT	0.0052	0.00523	0.00541	0.00515
Apple Inc	AAPL	0.00945	0.00946	0.01016	0.00907
Amazon.com Inc	AMZN	0.00883	0.00876	0.00846	0.00871
Berkshire Hathaway Inc Cl B	BRK-B	0.00434	0.00436	0.00398	0.00414
Johnson + Johnson	JNJ	0.00318	0.00321	0.00311	0.00295
JPMorgan Chase + Co	JPM	0.00738	0.0075	0.00801	0.00739
Exxon Mobil Corp	XOM	0.01075	0.01071	0.01052	0.0106
Pfizer Inc	PFE	0.00741	0.00747	0.0071	0.00749
Unitedhealth Group Inc	UNH	0.00578	0.00584	0.00527	0.00538
Verizon Communications Inc	VZ	0.00208	0.00208	0.0022	0.00215
Procter + Gamble Co/The	PG	0.00276	0.00278	0.0026	0.00264
Bank of America Corp	BAC	0.01019	0.0103	0.01031	0.00987
Intel Corporation	INTC	0.00737	0.00744	0.00751	0.00751
Chevron Corp	CVX	0.01065	0.01073	0.01015	0.0102
AT&T	T	0.00562	0.00572	0.00587	0.00541
Wells Frago & Company	WFC	0.01174	0.01173	0.01218	0.01071
Merck + Co. Inc	MRK	0.00367	0.00368	0.00395	0.00333
Cisco Systems Inc	CSCO	0.0064	0.00642	0.00635	0.00616
Home Depot Inc	HD	0.00713	0.00721	0.00653	0.00688
Coca Cola Co/The	KO	0.00431	0.00436	0.00412	0.0041

Table 6.3: Mean Squared Error (MSE) of the two classical and two Machine Learning methods.

Company	Ticker	MA	EWMA	RF	GB
Microsoft corp	MSFT	0.0613	0.06126	0.05962	0.05912
Apple Inc	AAPL	0.07776	0.07732	0.08209	0.07612
Amazon.com Inc	AMZN	0.07302	0.07261	0.07204	0.07401
Berkshire Hathaway Inc Cl B	BRK-B	0.05018	0.05027	0.04877	0.04993
Johnson + Johnson	JNJ	0.04441	0.04468	0.04281	0.04257
JPMorgan Chase + Co	JPM	0.06044	0.06139	0.06183	0.05963
Exxon Mobil Corp	XOM	0.07843	0.07842	0.08028	0.08035
Pfizer Inc	PFE	0.06684	0.0673	0.06504	0.06809
Unitedhealth Group Inc	UNH	0.05858	0.05897	0.05513	0.05579
Verizon Communications Inc	VZ	0.03527	0.03531	0.03706	0.03677
Procter + Gamble Co/The	PG	0.03932	0.03913	0.03934	0.03822
Bank of America Corp	BAC	0.079	0.07906	0.07938	0.07748
Intel Corporation	INTC	0.06865	0.06891	0.07053	0.07044
Chevron Corp	CVX	0.06915	0.06921	0.07141	0.07021
AT&T	T	0.05402	0.05426	0.05599	0.05369
Wells Frago & Company	WFC	0.07953	0.07898	0.08055	0.07752
Merck + Co. Inc	MRK	0.04597	0.04908	0.05087	0.04749
Cisco Systems Inc	CSCO	0.06632	0.06638	0.06584	0.0652
Home Depot Inc	HD	0.06118	0.06148	0.05885	0.06037
Coca Cola Co/The	KO	0.04567	0.04597	0.04604	0.04432

Table 6.4: Mean Absolute Error (MAE) of the two classical and two Machine Learning methods.

First of all, it's interesting to note that for a given stock, regardless of the Table viewed, the estimates are of the same order of magnitude, whatever the estimator used. This result highlights the fact that some stocks are easier to predict, due to their low volatility. Indeed, by comparing the MSE/MAE of the stocks with the volatilities of the various stocks (See Table 5.2), we can see that there is a relationship of proportionality between error and volatility. For example, the error on low-volatility stocks such as JNJ, VZ, PG and MRK is much lower than that on highly volatile stocks such as XOM, BAC, CVX and WFC.

That said, out of the 20 stocks, the MSE error measure is the lowest for the ML methods for 17 of the 20 stocks: 8 times thanks to RF and 9 times thanks to GB. However, RF seems to be rather versatile in its predictions, since it corresponds 9 times to the worst estimate of the expected return, i.e. the most often of the 4 methods analyzed here. GB, on the other hand, is only 2x in last place and only 7x behind a classical estimator (although this is not highlighted in the Table).

In terms of MAE, another measure of error, ML methods continue to dominate, albeit at a slightly lower level. The best estimates are obtained only 15 times out of 20 by ML methods: 10 times by Gradient Boosting and 5 times by RF. RF achieves the worst estimate in half the cases, which again reflects its volatile nature. GB is again very often ahead of the classical estimators.

If we cross-tabulate the 2 tables, we can see that INTC and VZ are, according to each table, the best predicted by a classical method. Since these are the only 2 stocks to go from a positive to a negative mean between training (Table 5.3) and test set (Table 5.4), we could hypothesize that the ML models didn't have enough examples in training to be able to correctly predict downward trends. Indeed, our methodology implies that the model remains the same from training to test set, so it was unable to correct itself with the new trend. ML methods are therefore disadvantaged in this particular case. Conversely, the 2 classical methods, thanks to their Moving Average, capture these trends and can adjust their predictions in the short term.

To conclude this first comparison, let's look at the MSE and MAE means of our different methods, which are presented in Table 6.5.

Metrics	MA	EWMA	RF	GB
Mean MSE	0.00671	0.00675	0.00669	0.00649
Mean MAE	0.06091	0.061	0.06117	0.06037

Table 6.5: Mean MSE and MAE of all methods (except DL) across stocks.

Once again, GB is the best estimator, both in terms of MSE and MAE, confirming its superiority over conventional methods. RF, 2nd in terms of MSE, is last in terms of MAE, no doubt because its poor estimates drag down its good estimates and therefore the overall MAE.

6.2 Deep Learning methods

In recent years, as we saw earlier, Deep Learning and its Deep Neural Networks have shown themselves capable of providing optimal solutions for a variety of challenges in image recognition, speech recognition, and Natural Language Processing. These algorithms draw inspiration from biological Neural Networks found within the brains of animals, enabling them to learn. This section introduces the fundamental characteristics of a Neural Network.

6.2.1 General operating principles

A classic Neural Network is the Multi-Layer Perceptron (MLP) [Rumelhart et al., 1986], also known as the fully connected feedforward Network. It is shown in the Figure 6.4.

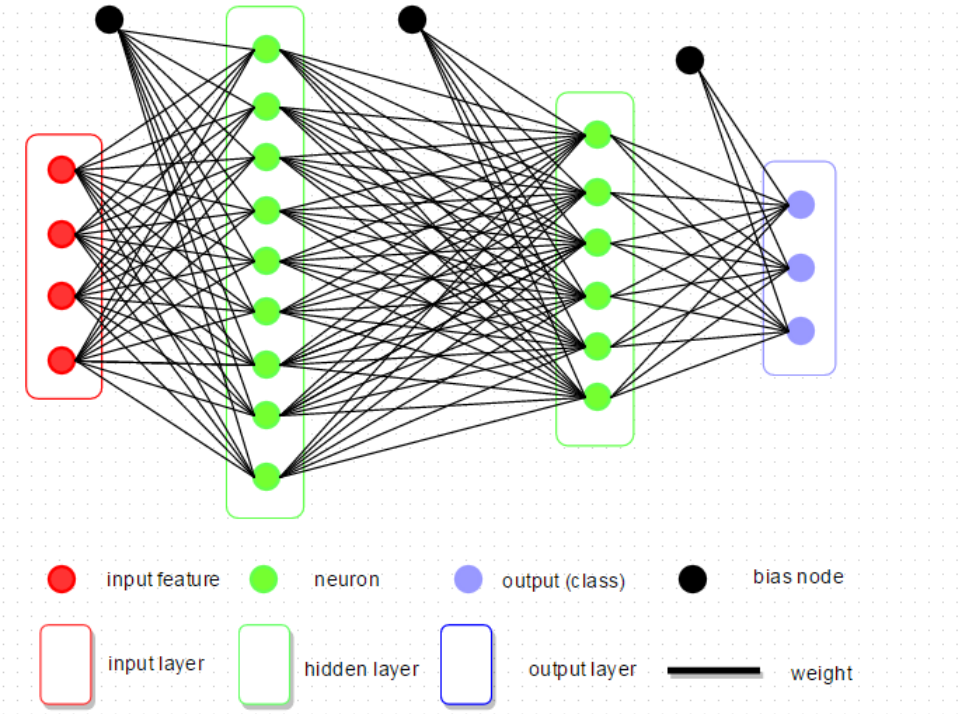

Figure 6.4: Neural networks example with 2 hidden layers [Zhao, 2016].

It consists of an input layer, an output layer and one or more hidden layers in between. The more hidden layers, the deeper the Neural Network. Each layer has a certain number of neurons. The number of neurons in the input layer corresponds to the number of input variables, while the number of neurons in the output layer corresponds to the number of output variables. In our case, the number of input neurons will be equal to the size of the window containing the previous returns used to determine the return at the next time. The output layer will consist of a single neuron corresponding to this return. In the simplest hidden layer, called the Dense Layer, each neuron is a linear combination (via weights) of the neurons in the previous layer, to which an activation function is applied, adding non-linearity. The result is coupled with a bias term that adjusts it. This is also the case for the output layer. Training the Neural Network involves adjusting weights to minimize the difference between predicted and actual outcomes, improving the network's ability to generalize and make accurate predictions. This is achieved using Gradient Descent [Hill, 2018], a first-order iterative optimization algorithm for finding the local minimum of a differentiable function.

In this thesis, we will use Adaptive moment estimation (Adam) [Kingma and Ba, 2014], which is an extension of the Stochastic Gradient Descent widely used in the literature. Unlike Gradient Descent, it is straightforward to implement and computationally efficient. What's more, its hyperparameters are intuitive and require little tuning. Neural networks consisting solely of Dense layers are the simplest and the first to have existed. Since then, more sophisticated and higher-performance layers have emerged. In this dissertation, we'll be using the Long Short-Term Memory layer (LSTM) and the MultiHeadAttention layer, characteristic of transformers, in addition to the Dense layers. We'll also need a GlobalAveragePooling1D layer to resize the output of the Attention layer.

6.2.2 Long Short-Term Memory Layer

Principles

The Long Short-Term Memory (LSTM) layer is what's known as a recurring layer. The idea behind a Recurrent layer is to use the sequentiality present in a timeseries, such as that of historical returns, by storing in memory the important information linked to this sequentiality. Conventional Recurrent layers suffer from the vanishing gradient problem, making learning impossible when many hidden layers are added. LSTMs were therefore introduced in 1997 [Hochreiter and Schmidhuber, 1997] with the aim of learning long-term dependencies across many number of time steps and solving the problem of vanishing gradient.

Implementation

Deep Learning models are much heavier, so their implementations are often parallelized, calling on GPUs. This is not possible with the *scikit-learn* library, where only CPUs are used. So we're going to use the *Tensorflow* library [Abadi et al., 2015] library with the *Keras* sub-library to carry out our implementation. Although LSTMs solve the vanishing gradient problem, it's generally advisable not to stack too many layers of LSTMs. So we'll limit ourselves to a maximum of 2 layers. Once again, we'll have to chose hyperparameters and for this we will use the guidelines of [Eckhardt, 2021].

Table 6.6 shows the main hyperparameters for LSTMs:

Parameters	Definition	Intuition
LSTM width	Number of neurons per layer. Not necessarily the same in each layer	The higher the number of neurons, the more complexity can be captured in the data. A low number is therefore linked to underfitting, and a high number to overfitting.
Dropout	Temporary and random deletion of neurons in training to avoid overfitting. The Recurrent version is an extension to Recurrent networks	Prevent the network from becoming overly reliant on specific neurons and features. If the dropout is low, there is a risk of overfitting and conversely, if the dropout is too high, there is a risk of underfitting.

Table 6.6: Main hyperparameters Long Short-Term Memory layers.

We'll be using 1 or 2 layers of LSTM, the appreciation being left to *RandomizedSearchCV* choice. For Neural Networks, we will also tune the learning rate. This is the step size at which the model's parameters are updated during Gradient Descent (loss minimization). If the learning rate is low, this means that the parameters are not updated very often, and that the loss can easily fall into a local minimum during minimization. If the learning rate is too high, the optimization may oscillate around an optimal solution without ever reaching it.

6.2.3 Transformers

Principles

Transformers (TF) layers are designed to efficiently capture long-range relationships in sequential data, such as time series. Unlike an LSTM, which processes one element of a sequence at a time and keeps track of previous elements, transformers analyze all the data together to calculate an Attention score that quantifies their mutual relevance. Attention makes it easier to capture long-term

dependencies between elements in a sequence by assigning weights to each element according to its relevance to the current task. We'll use MultiHeadAttention from the original 2017 paper, "Attention is All You Need" [Vaswani et al., 2017] which uses several independent Attentions. These are called Attention heads. Their combination enables the model to learn richer, more diverse representations.

Implementation

We will use the MultiHeadAttention layer from *Tensorflow*. Note also that the Attention layer is not generally used on its own. We will use the Attention after 1 or 2 layers of LSTM, the appreciation being left to *RandomizedSearchCV* choice. But as the Attention principle is present, global Neural Network will be called TF.

The output of the MultiHeadAttention layer is too large for our output. We will need to add a pooling layer, the GlobalAveragePooling1D layer, which downsamples the representation input while preserving the essential information contained in the sequence.

Table 6.7 shows the main hyperparameters to play with in TFs [Adaloglou, 2021]:

Parameters	**Definition**	**Intuition**
Number of heads	Number of independent Attention heads in the MultiHeadAttention layer.	Each Attention head processes the input in a different way, capturing different aspects of the relationships between the elements in the sequence. More heads enable the model to learn richer and more complex Attention representations, thanks to their combination. If the number of heads is too low, underfitting may occur, and conversely, overfitting.
Head size	Size of subspaces in which information is managed by each head.	A smaller head size may allow richer information sharing between heads and avoid excessive growth in the total number of parameters. On the other hand, a larger head size could enable the model to learn more complex and specific relationships. If the size is too small, underfitting may occur, and conversely, overfitting.
Head dropout	Random deactivation of certain neurons as in classic dropout	A low dropout is linked to overfitting, and a high dropout to underfitting.

Table 6.7: Main hyperparameters Transformers layers.

6.2.4 Results

In the same way as we compared classic methods to ML methods, let's compare classic methods to DL methods. MSEs are present in Table 6.8, while MAEs are present in Table 6.9.

Company	Ticker	MA	EWMA	LSTM	TF
Microsoft corp	MSFT	0.0052	0.00523	0.00526	0.00522
Apple Inc	AAPL	0.00945	0.00946	0.00917	0.00899
Amazon.com Inc	AMZN	0.00883	0.00876	0.00953	0.00942
Berkshire Hathaway Inc Cl B	BRK-B	0.00434	0.00436	0.00396	0.00407
Johnson + Johnson	JNJ	0.00318	0.00321	0.003	0.00297
JPMorgan Chase + Co	JPM	0.00738	0.0075	0.00692	0.00716
Exxon Mobil Corp	XOM	0.01075	0.01071	0.01073	0.0104
Pfizer Inc	PFE	0.00741	0.00747	0.00708	0.007
Unitedhealth Group Inc	UNH	0.00578	0.00584	0.00517	0.0055
Verizon Communications Inc	VZ	0.00208	0.00208	0.00206	0.00215
Procter + Gamble Co/The	PG	0.00276	0.00278	0.00266	0.00267
Bank of America Corp	BAC	0.01019	0.0103	0.00973	0.00964
Intel Corporation	INTC	0.00737	0.00744	0.00755	0.00761
Chevron Corp	CVX	0.01065	0.01073	0.00975	0.01025
AT&T	T	0.00562	0.00572	0.00523	0.00541
Wells Frago & Company	WFC	0.01174	0.01173	0.01248	0.01106
Merck + Co. Inc	MRK	0.00367	0.00368	0.00339	0.00354
Cisco Systems Inc	CSCO	0.0064	0.00642	0.00599	0.00574
Home Depot Inc	HD	0.00713	0.00721	0.0065	0.00694
Coca Cola Co/The	KO	0.00431	0.00436	0.00385	0.00413

Table 6.8: Mean Squared Error (MSE) of the two classical and two Deep Learning methods.

Company	Ticker	MA	EWMA	LSTM	TF
Microsoft corp	MSFT	0.0613	0.06126	0.05973	0.05957
Apple Inc	AAPL	0.07776	0.07732	0.07628	0.07611
Amazon.com Inc	AMZN	0.07302	0.07261	0.077	0.07712
Berkshire Hathaway Inc Cl B	BRK-B	0.05018	0.05027	0.04897	0.04938
Johnson + Johnson	JNJ	0.04441	0.04468	0.04295	0.04287
JPMorgan Chase + Co	JPM	0.06044	0.06139	0.05895	0.05862
Exxon Mobil Corp	XOM	0.07843	0.07842	0.08225	0.08007
Pfizer Inc	PFE	0.06684	0.0673	0.06487	0.06434
Unitedhealth Group Inc	UNH	0.05858	0.05897	0.05435	0.05654
Verizon Communications Inc	VZ	0.03527	0.03531	0.03586	0.03661
Procter + Gamble Co/The	PG	0.03932	0.03913	0.03786	0.03822
Bank of America Corp	BAC	0.079	0.07906	0.0769	0.07636
Intel Corporation	INTC	0.06865	0.06891	0.07047	0.07095
Chevron Corp	CVX	0.06915	0.06921	0.07115	0.07117
AT&T	T	0.05402	0.05426	0.0526	0.05294
Wells Frago & Company	WFC	0.07953	0.07898	0.08451	0.07816
Merck + Co. Inc	MRK	0.04597	0.04908	0.0457	0.04638
Cisco Systems Inc	CSCO	0.06632	0.06638	0.06451	0.06333
Home Depot Inc	HD	0.06118	0.06148	0.05963	0.06024
Coca Cola Co/The	KO	0.04567	0.04597	0.04347	0.04527

Table 6.9: Mean Absolute Error (MAE) of the two classical and two Deep Learning methods.

First of all, the MSE error measure is the best for the DL methods in 17 of the 20 cases: 10x thanks to LSTM and 7x thanks to TF. We also note that in 14 of these 17 cases, when TF or LSTM is first, the other follows. The worst estimate is obtained in three quarters of cases by a classical method, 14x via EWMA, whose MSEs are often close to that of MA. In the MAE table, the best estimate is obtained in 15 out of 20 cases by a DL method, 7x by LSTM and 8x by TF. Here, therefore, TF dominates. We can assume that this dominance is linked to the fact that TF generally generates larger errors than LSTM, the latter being squared in the MSE and thus justifying its poorer performance in the former case. In 13 of these 15 cases, LSTM and TF share the top 2 places.

By crossing the 2 tables, we can see that the best INTC estimator is once again given by a classical method, probably for the same reason as the hypothesis formulated previously. This is also still the case in MAE for VZ, but this time LSTM has succeeded very slightly in outperforming the classical estimators in MSE.

To conclude this second comparison, let's take a look at the MSE and MAE means of our different methods, which can be found in Table 6.10.

Metrics	MA	EWMA	LSTM	TF
Mean MSE	0.00671	0.00675	0.0065	0.00649
Mean MAE	0.06091	0.061	0.0604	0.06021

Table 6.10: Mean MSE and MAE of all methods (except ML) across stocks.

The dominance of DL methods is confirmed by the fact that the 2 best averages in both MSE and MAE are obtained by LSTM and TF.

6.3 Answer to the second research question

In the previous sections, we compared classical methods with ML and DL methods respectively. Random Forest was sometimes much better than the classical methods, and even better than GB, and sometimes worse than the classical methods. The other 3 methods, on the other hand, performed well in terms of MSE and MAE, both individually and on average. We'll now compare all the methods to answer the second research question. Tables 6.11 and 6.12 summarize this process:

Company	Ticker	MA	EWMA	RF	GB	LSTM	TF
Microsoft corp	MSFT	0.0052	0.00523	0.00541	0.00515	0.00526	0.00522
Apple Inc	AAPL	0.00945	0.00946	0.01016	0.00907	0.00917	0.00899
Amazon.com Inc	AMZN	0.00883	0.00876	0.00846	0.00871	0.00953	0.00942
Berkshire Hathaway Inc Cl B	BRK-B	0.00434	0.00436	0.00398	0.00414	0.00396	0.00407
Johnson + Johnson	JNJ	0.00318	0.00321	0.00311	0.00295	0.003	0.00297
JPMorgan Chase + Co	JPM	0.00738	0.0075	0.00801	0.00739	0.00692	0.00716
Exxon Mobil Corp	XOM	0.01075	0.01071	0.01052	0.0106	0.01073	0.0104
Pfizer Inc	PFE	0.00741	0.00747	0.0071	0.00749	0.00708	0.007
Unitedhealth Group Inc	UNH	0.00578	0.00584	0.00527	0.00538	0.00517	0.0055
Verizon Communications Inc	VZ	0.00208	0.00208	0.0022	0.00215	0.00206	0.00215
Procter + Gamble Co/The	PG	0.00276	0.00278	0.0026	0.00264	0.00266	0.00267
Bank of America Corp	BAC	0.01019	0.0103	0.01031	0.00987	0.00973	0.00964
Intel Corporation	INTC	0.00737	0.00744	0.00751	0.00751	0.00755	0.00761
Chevron Corp	CVX	0.01065	0.01073	0.01015	0.0102	0.00975	0.01025
AT&T	T	0.00562	0.00572	0.00587	0.00541	0.00523	0.00541
Wells Frago & Company	WFC	0.01174	0.01173	0.01218	0.01071	0.01248	0.01106
Merck + Co. Inc	MRK	0.00367	0.00368	0.00395	0.00333	0.00339	0.00354
Cisco Systems Inc	CSCO	0.0064	0.00642	0.00635	0.00616	0.00599	0.00574
Home Depot Inc	HD	0.00713	0.00721	0.00653	0.00688	0.0065	0.00694
Coca Cola Co/The	KO	0.00431	0.00436	0.00412	0.0041	0.00385	0.00413

Table 6.11: Mean Squared Error (MSE) of all the methods.

Company	Ticker	MA	EWMA	RF	GB	LSTM	TF
Microsoft corp	MSFT	0.0613	0.06126	0.05962	0.05912	0.05973	0.05957
Apple Inc	AAPL	0.07776	0.07732	0.08209	0.07612	0.07628	0.07611
Amazon.com Inc	AMZN	0.07302	0.07261	0.07204	0.07401	0.077	0.07712
Berkshire Hathaway Inc Cl B	BRK-B	0.05018	0.05027	0.04877	0.04993	0.04897	0.04938
Johnson + Johnson	JNJ	0.04441	0.04468	0.04281	0.04257	0.04295	0.04287
JPMorgan Chase + Co	JPM	0.06044	0.06139	0.06183	0.05963	0.05895	0.05862
Exxon Mobil Corp	XOM	0.07843	0.07842	0.08028	0.08035	0.08225	0.08007
Pfizer Inc	PFE	0.06684	0.0673	0.06504	0.06809	0.06487	0.06434
Unitedhealth Group Inc	UNH	0.05858	0.05897	0.05513	0.05579	0.05435	0.05654
Verizon Communications Inc	VZ	0.03527	0.03531	0.03706	0.03677	0.03586	0.03661
Procter + Gamble Co/The	PG	0.03932	0.03913	0.03934	0.03822	0.03786	0.03822
Bank of America Corp	BAC	0.079	0.07906	0.07938	0.07748	0.0769	0.07636
Intel Corporation	INTC	0.06865	0.06891	0.07053	0.07044	0.07047	0.07095
Chevron Corp	CVX	0.06915	0.06921	0.07141	0.07021	0.07115	0.07117
AT&T	T	0.05402	0.05426	0.05599	0.05369	0.0526	0.05294
Wells Frago & Company	WFC	0.07953	0.07898	0.08055	0.07752	0.08451	0.07816
Merck + Co. Inc	MRK	0.04597	0.04908	0.05087	0.04749	0.0457	0.04638
Cisco Systems Inc	CSCO	0.06632	0.06638	0.06584	0.0652	0.06451	0.06333
Home Depot Inc	HD	0.06118	0.06148	0.05885	0.06037	0.05963	0.06024
Coca Cola Co/The	KO	0.04567	0.04597	0.04604	0.04432	0.04347	0.04527

Table 6.12: Mean Absolute Error (MAE) of all the methods.

We can see that in terms of MSE, AI methods obtain the best estimate in 19 cases out of 20: 2x thanks to RF, 4x thanks to GB, 8x thanks to LSTM and 5x thanks to TF. EWMA is the worst estimator in 8 cases out of 20, and RF is the worst estimator in 7 cases out of 20. In terms of MAE, AI methods generate the best estimates in 16 cases out of 20: 3x thanks to RF, 3x thanks to GB, 5x thanks to LSTM and 5x thanks to TF. The fact that here, as before, the number of first-place estimates is lower for AI methods in terms of MAE reflects the fact that conventional methods generally make larger errors on estimates, as these are squared.

To conclude this last comparison, let's take a look at the means MSE and MAE of our different methods, as shown in Table 6.13.

Metrics	**MA**	**EWMA**	**RF**	**GB**	**LSTM**	**TF**
Mean MSE	0.00671	0.00675	0.00669	0.00649	0.0065	0.00649
Mean MAE	0.06091	0.061	0.06117	0.06037	0.0604	0.06021

Table 6.13: Mean MSE and MAE of all methods across stocks.

This last table shows once again the dominance of AI methods (excluding RF) over traditional methods. Our research question asked whether the estimation of expected returns via Machine Learning based on historical prices was more **accurate** than that provided by more traditional methods. It also aimed to establish which method gives the best estimates.

We found that estimates generated by Machine Learning (including Deep Learning) techniques were not always better than those generated by conventional methods. However, the best estimates per stock were almost always obtained via these methods, and they were almost always more accurate (in terms of MSE and MAE) than classical methods with the exception of Random Forest, which sometimes performs very well, sometimes falls well short of expectations. No AI method, either in terms of MSE or MAE, proved capable of challenging INTC's MA estimate. We believe this is because the out-of-sample returns for this stock are very different from the in-sample data. Since ML models have a fixed model trained between 2003 and 2019, data that are too different out-of-sample could not be correctly predicted. Conversely, models such as MA and EWMA are capable of adapting to new short-term trends in out-of-sample data. Figure 6.5 illustrates this point.

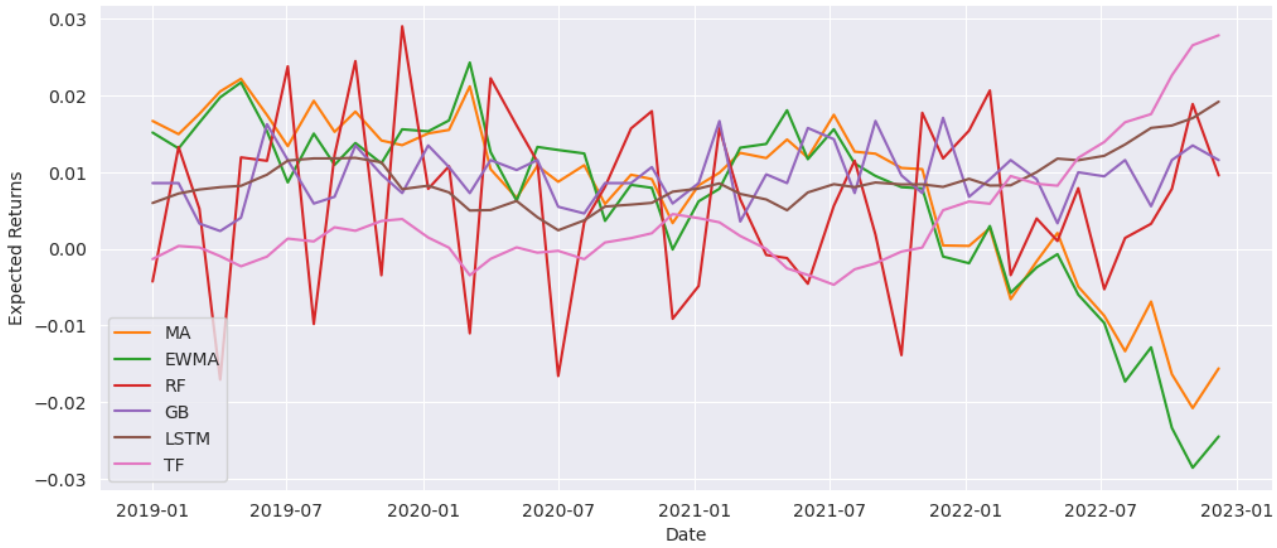

Figure 6.5: INTC expected returns for all the methods (2019-2022).

We can see that most estimators (except RF) maintain a certain course during the out-of-sample period. Then, in early 2022, MA and EWMA are the only ones to fall. When we compare this trend with realized returns, we can see why it's linked to the fall in INTC's realized returns, as shown in Figure 6.6.

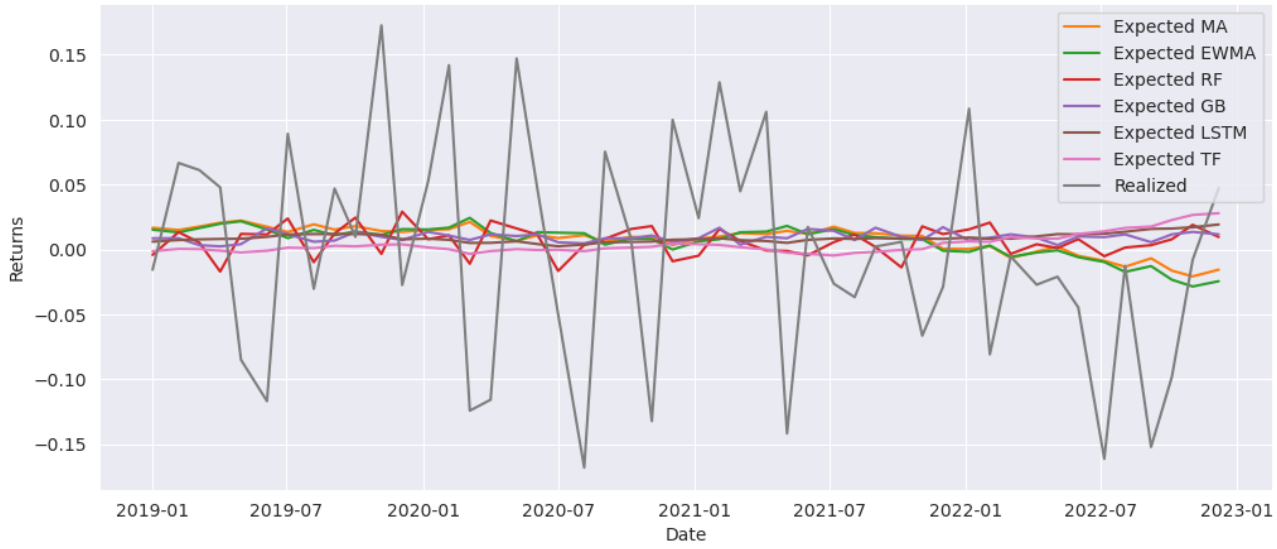

Figure 6.6: INTC expected returns for all the methods and realized returns (2019-2022).

Indeed, from mid-2021 onwards, whereas they were on average positive until then, almost all INTC's realized returns become negative. These 2 Figures also highlight the fact that although our estimators are close in terms of MSE and EWMA, they do not behave in a similar way, in particular for MA and EWMA that almost always have the same scores. We also see that RF is highly volatile, as expected.

Finally, although our estimates with AI have improved, we can see that there's still some way to go before we're able to predict expected returns on a par with realized returns. As we'll see in the next section, this won't stop us from generating better portfolios than before with conventional

methods. The various succesful approaches (GB, LSTM and TF) do indeed offer a kind of improved rolling average, but one that is very smooth and low-risk. The reasoning is based on INTC, but the Figure is also representative of predictions on other stocks: similar reasoning could be established for other stocks.

Let's move on to the question of the best model, which is more complex to deal with. 3 models could lay claim to this title: GB, LSTM and TF. LSTM is the best estimator for the largest number of stocks in terms of MSE. However, GB and TF have the best average MSE. In terms of MAE, LSTM and TF are the best estimators for the greatest number of times. But TF has the best average MAE. What's more, in each of the comparisons carried out, the 3 models are neck-and-neck: domination is not wide.

An additional element may help to settle the matter: execution time. While the classic models take a handful of seconds, GB takes 30min of training, RF 2h and LSTM and TF 6h. One might be tempted to declare GB the clear winner... However, given that classic robo-advisor rebalancing takes place on a monthly or even quarterly basis, 6h is still acceptable and should not tip the balance.

So, although these 3 models could be declared the best, we'll choose TF as the best. It has the best combination of features with regard to the criteria presented above, notably the best MSE (albeit tied with GB) and the best MAE.

Let's now compare these results with the literature, more specifically with papers comparing several models aimed at predicting stock returns. In "Portfolio optimization with return prediction using Deep Learning and Machine Learning" [Ma et al., 2021b], the authors obtained an average MSE across stocks of 0.00081 for their best RF model and 0.00089 for their best LSTM model. The respective MAEs were 0.0185 and 0.0202. These were their 2 best models (against 4 other models, classical or not) and the RF model was therefore better than their LSTM model. In "Artificial Intelligence Alter Egos: Who might benefit from robo-investing?" [D'Hondt et al., 2020], the average MSE across stocks of a Neural Network was 0.01, while that of RF was 0.0114, suggesting the dominance of Neural Networks in this estimation task vis-à-vis the 4 estimators (classical or not) tested. In "Deep Learning for Forecasting Stock Returns in the Cross-Section" [Abe and Nakayama, 2018], Neural Networks performed better than the SVR and RF algorithms tested.

Since the stocks (and therefore their variances) are different, comparing our orders of magnitude in terms of results to theirs makes no sense, since as we saw at the start of this chapter, some stocks are easier to predict than others. However, we also found that Neural Networks were better than conventional estimators for stock return estimation. Moreover, we also find that these Neural Networks (both LSTM and TF) are a better estimator than RF, both in terms of MSE and MAE. This is in line with the last two articles mentioned.

6.4 Answer to the third research question

In the previous section, we determined that GB, LSTM and TF were 3 models above the others in terms of accurate forecasting. We finally chose TF as the best model. But is TF the model with the best cumulative realized return over the period 2019-2022? Let's find out. As described in the methodology section, for every method and every month within the test set, the expected returns for individual stocks will be utilized as inputs in the Modern Portfolio Theory (MPT). These MPTs will then output the weights maximizing the Sharpe ratio of a portfolio composed of these particular stocks. Subsequently, these calculated weights will be employed to the realized returns of the stocks across the different months, thereby computing the realized return of the portfolio on a monthly basis. A rebalancing will then take place every month on the basis of the new predicted expected returns.

Applying this methodology to all the methods and all the months making up the 2019-2023 period, we obtain the realized returns for each month. These returns can then be used to determine the cumulative realized return, the total change in the investment price over the period 2019-2023. In addition to the 6 models seen above, 2 other strategies will be represented. Firstly, a 1/N allocation, where each of the 20 stocks is present with the same proportion in the portfolio, i.e. 5%. Secondly, a passive strategy consisting of investing solely in the S&P500, an ETF presented earlier. The result is shown in Figure 6.7.

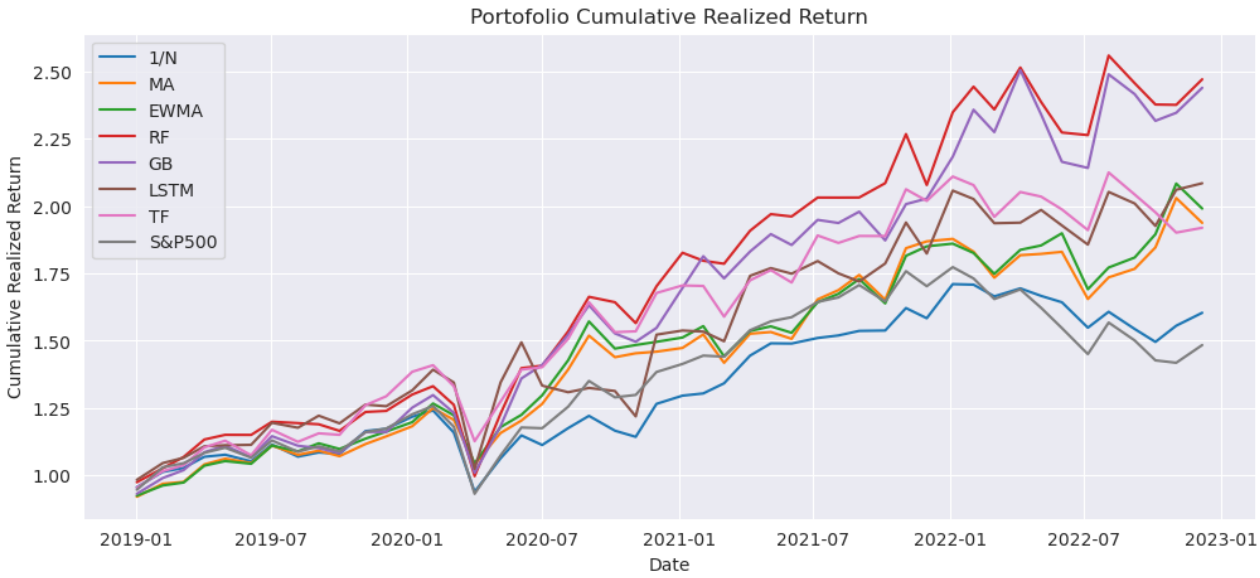

Figure 6.7: Portfolio Cumulative Realized Return.

Up until April 2020 and the start of the Covid Crisis, we can see that no single strategy really stood out. But as we move from March to April 2020, we can see that all stocks have fallen. As the realized returns of our stocks were all negative that month, this result could not be avoided. The strategy that fared best at that point was the one linked to an estimate of expected returns given by TF. Thereafter, until early 2022, our 3 best expected return models from the previous section - GB, LSTM and TF - remained neck and neck, while being constantly outperformed by RF. It was at this point that the two DL methods, LSTM and TF, began to stagnate at around an cumulative realized return of 2 throughout 2022, eventually being overtaken by MA and EWMA, corresponding to an annualized return of 19%.

The date of early 2022 is not insignificant. After the Covid Crisis, the prices of a wide range of assets such as equities, bonds, real estate, commodities and other asset classes rose excessively and simultaneously to create a speculative bubble. It wasn't until early 2022 that this speculative bubble finally burst [Finance, 2023], with asset prices plummeting and investors suffering financial losses.

LSTM and TF felt the effect of this bubble, although they fared rather well, remaining stable. Indeed, most of the MDDs in our test set (see Table 5.4), some of which exceeded 50%, took place in the year 2022. So we can see that, thanks to MPT diversification, our models were able to avoid the least profitable investments. Some of our models, GB and RF, even continued to rise in 2022. GB even ended up catching up with RF, and the two finished 2022 at around 2.5 cumulative expected return, i.e. an annualized return of 25%. The 2 worst-performing strategies were the simplest, generating a realized expected return of around 1.5 after 4 years. Annualized, this represents a return of 11%.

Institutions might be tempted to keep a capital cost as a safety buffer to cover potential losses arising from risky activities or adverse economic events. If we decide to allocate 30% of our portfolio to this capital cost, investing it at a risk-free rate (2%/year), we obtain the cumulative realized return shown in Figure 6.8.

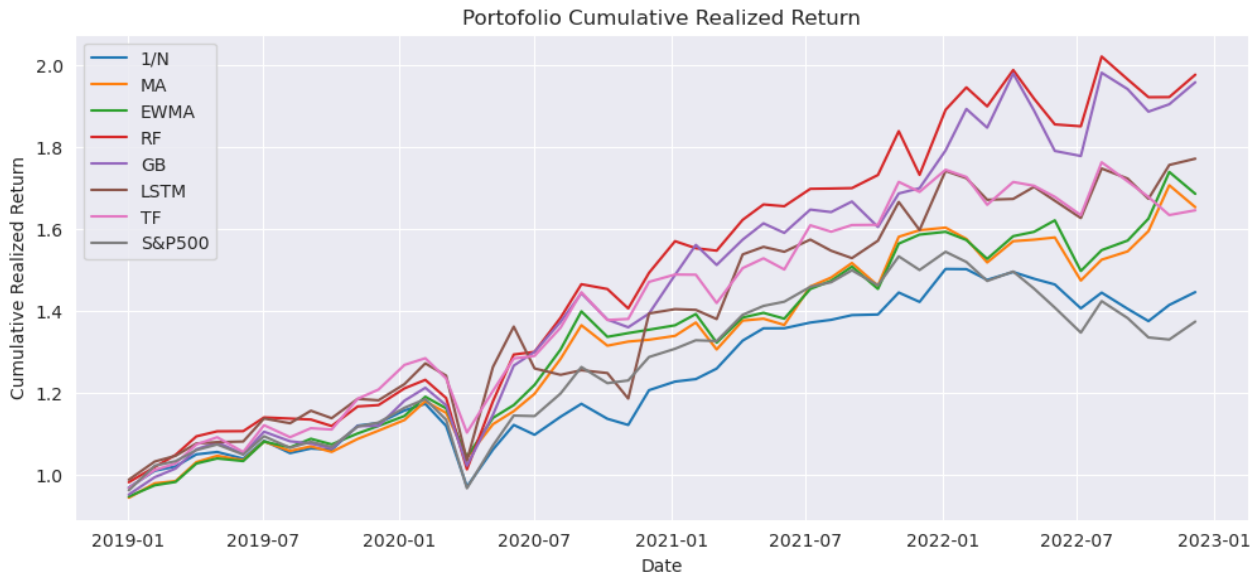

Figure 6.8: Portfolio Cumulative Realized Return with capital cost.

We can see that this solution reduces each of our declines in cumulative realized return. However, it also reduces each of our increases, since almost all are at a higher rate than that of the risk-free asset. The portfolios realized after 4 years are therefore all significantly worse than before. But this solution will allow institutions to maintain a safety buffer if they so wish.

In the previous section, we saw that RF was far from being the best estimator. However, in the end, it is the one giving the best realized expected return, albeit very close to GB. We'll try to understand why by looking at the average weights assigned by our methods to the various assets, juxtaposed with the average realized returns over this period. The weights in question can be found in Table 6.14.

Company	Ticker	MA(%)	EWMA(%)	RF(%)	GB(%)	LSTM(%)	TF(%)
Microsoft corp	MSFT	29.24	27.18	3.21	1.96	0	0
Apple Inc	AAPL	14.13	13.74	10.1	21.87	3.13	28.89
Amazon.com Inc	AMZN	2.63	1.91	10.65	16.12	10.27	14.62
Berkshire Hathaway Inc Cl B	BRK-B	1.82	2.31	7.12	6.69	0	0
Johnson + Johnson	JNJ	0.32	0.28	8.89	5.84	0	0
JPMorgan Chase + Co	JPM	0.06	0.48	3.18	1.32	0.31	6.86
Exxon Mobil Corp	XOM	11.8	13.2	2.6	3.2	0.47	3.04
Pfizer Inc	PFE	5.1	5.43	6.42	7.9	4.38	0
Unitedhealth Group Inc	UNH	6.08	5.36	4.24	6.2	4.03	1.48
Verizon Communications Inc	VZ	1.18	0.78	4.55	3.41	0	0
Procter + Gamble Co/The	PG	17.9	18.47	3.77	1.6	11.86	0
Bank of America Corp	BAC	0	0	1.13	0	0	0
Intel Corporation	INTC	0.15	0.11	0.6	0.41	0	0.28
Chevron Corp	CVX	0.41	0.69	6.25	1.78	15.79	0.93
AT&T	T	0.04	0.2	4.31	1.55	5.15	0
Wells Frago & Company	WFC	0.03	0	5.21	3.64	11.28	3.08
Merck + Co. Inc	MRK	2.31	2.74	5.22	6.76	10.71	34.07
Cisco Systems Inc	CSCO	2.76	2.83	3.27	5.01	4.21	0
Home Depot Inc	HD	3.64	3.7	4.44	1.87	15.15	6.74
Coca Cola Co/The	KO	0.39	0.58	4.83	2.87	3.26	0

Table 6.14: Mean weights of all the methods (2019-2022).

The weights show that the key to RF and GB's success probably lies in diversification. Indeed, they are the only stocks to have a weighting of at least 1% in 19 and 18 stocks respectively. For MA, EWMA, LSTM and TF, these numbers are 12, 11, 12, 8 stocks respectively. As shown by Markowitz, diversification enables risk to be spread across different assets that react differently to economic events and market fluctuations. If one asset performs poorly, other assets in the portfolio can compensate for these losses, helping to reduce overall portfolio volatility. This limits potential losses if an asset underperforms. Diversification can also lead to more stable and consistent returns over the long term. On the previous chart, we saw that these 2 methods, which are diversified, were the only ones to continue to grow in terms of realized cumulative return in 2022, i.e. during the bubble burst period, when many assets fell. This is probably where their success lies.

But, why did these 2 methods lead MPT to create more diversified portfolios than the others, while only the expected returns changed? This is probably linked to the nature of the expected returns predicted by these different methods. Table 6.15 gives an idea of the mean and volatility of the monthly expected returns associated with the different methods.

Metrics	MA	EWMA	RF	GB	LSTM	TF
Mean Expected Return	1.292	1.271	0.986	1.121	1.219	1.199
Volatility Expected Return	0.788	0.76	0.588	0.589	0.783	0.885

Table 6.15: Mean and Volatility of expected returns for all the methods (2019-2022).

We can see that the means in the case of RF and GB are lower than in the other cases, and that volatility relative to this average is much lower. Intuitively, low volatility may be associated with greater diversification. When certain methods generate expected returns with higher volatility, there is generally a greater difference between the highest and lowest expected returns. In contrast, for

methods with low-volatility expected returns, the differences between expected returns are less pronounced. Maintaining an equivalent level of risk (an assumption that is often realistic, especially if the covariance matrix is dominated by small elements, as in our case: see Figure 5.4), it is likely that a greater weight will be allocated to the higher expected returns than to the lower ones in the first case. This differs from the second case, where the expected returns are all close together, leading to a more uniformly diversified portfolio, as observed in RF and GB. It is important to note, however, that simple diversification is not enough: otherwise the 1/N allocation would have achieved better results. It is by combining more sophisticated methods of estimating expected returns and MPT that we have been able to achieve smarter diversification.

One final point to note is that each of our methods almost consistently achieved a better cumulative realized return than the S&P500, a passive strategy. In "Artificial Intelligence Alter Egos: Who might benefit from robo-investing?" [D'Hondt et al., 2020], the authors fell far short of this ETF, despite equivalent use of MPT. The most likely explanation for this phenomenon is the probable lack of diversification on their part. The article considered investment universes based on a minimum of 2 stocks, whereas ours contains 20. With only 2 stocks, we understand that the risk is very high.

Let's thus answer our last research question. No, better expected return estimates do not necessarily lead to better cumulative expected returns. Random Forest, which was very often the worst model in terms of estimation, and 4th on average, is the one that finally obtained the best cumulative realized return after 4 years. We believe that this result is due to the fact that its estimates are all very close, leading in fine to a more diversified portfolio in terms of number of stocks, and so the weights are not so far apart. Perhaps our out-of-sample time horizon wasn't long enough for the strategy with the best TF estimate to win.

Chapter 7

Conclusion and guidelines for future work

This study has delved into the application of Machine Learning techniques for estimating expected returns within the context of Modern Portfolio Theory. Our tripartite objective encompassed several facets. Firstly, the endeavor sought to catalog the most widely employed Machine Learning methods in the realm of stock return forecasting, and more broadly, in time series forecasting. The comprehensive literature review in Chapter 4 unveiled notable models, including Random Forest, Gradient Boosting, Long Short-Term Memory, and Transformers.

Then, in Chapter 6, the second aim aimed to ascertain whether the application of Machine Learning techniques rooted in historical prices augments the accuracy of expected return estimation when compared to conventional methods. To address this inquiry, we delineated an investment universe composed of the top 20 constituents of the S&P 500 with a minimum 20-year history. The goal was to contrast the monthly stock returns predicted by the various methods under scrutiny—namely, the aforementioned four methods alongside two classical methods from literature: Moving Average and Exponentially Weighted Moving Average. The comparison was conducted in terms of Mean Squared Error and Mean Absolute Error. Our findings consistently illuminated that in nearly all instances, Machine Learning methods outperformed traditional estimations of future stock returns. Despite Gradient Boosting, Long Short-Term Memory, and Transformers closely vying in terms of MSE and MAE, Transformers ultimately emerged as the optimal model.

The third objective entailed investigating whether improved estimations of expected returns across diverse stocks inherently translated to superior cumulative realized returns of portfolios constructed from these estimations. Random Forest, the fourth model among the six explored in the second research question, surfaced as the superior choice for addressing this aspect. This shows that better predicted expected returns do not necessarily lead to better cumulative realized portfolios. By closely analyzing the attributes of expected returns associated with Random Forest and the portfolio weights derived from these estimations, we hypothesized that the relatively low volatility between Random Forest's estimations likely facilitated the construction of portfolios yielding enhanced cumulative realized returns.

The implications of these findings bear significant financial importance. The application of Machine Learning techniques exhibits the potential to reshape investment strategies fundamentally. By providing more accurate estimations of expected returns, investors can make more informed decisions when constructing portfolios. This heightened precision contributes to improved risk management as well as the optimization of the risk-return trade-off, a central tenet of Modern Portfolio Theory. As a result, the financial industry can expect to witness an evolution in investment practices, wherein the integration of Machine Learning methodologies augments the capability to navigate intricate market dynamics with greater confidence.

Furthermore, the study's revelation that Machine Learning models consistently outperform tradi-

tional methods in terms of estimating stock returns further underscores the need for adapting to modern approaches. The financial sector has long been driven by data-driven decision-making, and as the complexity of financial markets increases, harnessing advanced predictive tools becomes imperative. By leveraging these Machine Learning techniques, financial professionals can gain a competitive edge in identifying investment opportunities and tailoring strategies to better align with market trends.

From a broader perspective, the findings have the potential to impact asset management companies, financial institutions, and individual investors alike. The integration of Machine Learning methodologies may lead to more efficient allocation of resources, potentially unlocking higher returns while mitigating risks. Additionally, the implementation of these advanced techniques could foster innovation in financial product offerings, enabling the design of investment solutions that are tailored to investors' unique risk appetites and financial goals.

To finish, here are the guidelines for future work:

- Our study relied solely on historical stock prices as input data. However, the literature suggests the common use of various economic indicators. It would be intriguing to replicate the analysis by incorporating a broader set of input variables. Random Forest and Gradient Boosting, particularly adept at tabular data [Grinsztajn et al., 2022], might unveil even stronger performance under these conditions.

- The optimization of portfolios in our study resulted in substantial variations in the allocation weights of different stocks. In practical scenarios, transaction costs play a pivotal role, potentially undermining the cumulative realized returns we have obtained. Integrating these transaction costs into the cost objective would provide a more realistic evaluation of portfolio performance.

- Our analysis is confined to stocks with over 20 years of historical data, excluding several of today's major capitalization stocks. This limitation introduces a sample (youth) bias [Nikolopoulou, 2022]. Exploring methodologies to mitigate this limitation and incorporate more contemporary stocks could yield more comprehensive insights.

- We have seen that the best model for a given stock is not always the same. It would be interesting to see the effect of averaging the contributions of the different models (as the ensemble methods we have seen already do) or to consider that certain models are better for certain stocks and combine their predictions in the construction of optimal portfolios and therefore cumulative realized returns.

- The complexity of most of our Deep Learning models was relatively modest. Given the rapid advancement of GPUs, future research could venture into constructing increasingly intricate models to extract more nuanced patterns from Deep Neural Networks. As GPUs continue to evolve, capitalizing on their capabilities can lead to the development of more powerful and accurate predictive models.

Bibliography

[Abadi et al., 2015] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[Abe and Nakayama, 2018] Abe, M. and Nakayama, H. (2018). Deep learning for forecasting stock returns in the cross-section. In *Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part I 22*, pages 273–284. Springer.

[Actors, 2018] Actors, L. (2018). Machine Learning Algorithms: Which one to choose for your problem. *Learning Actors*.

[Adaloglou, 2021] Adaloglou, N. (2021). Why multi-head self attention works: Math, intuitions and 10+1 hidden Insights | AI Summer.

[Adler and Kritzman, 2006] Adler, T. and Kritzman, M. (2006). Mean–variance versus full-scale optimisation: in and out of sample. *Journal of Asset Management*, 7(5):302–311.

[Ahuvia, 2008] Ahuvia, A. (2008). If money doesn't make us happy, why do we act as if it does? *Journal of economic psychology*, 29(4):491–507.

[Alexander, 2008] Alexander, C. (2008). *Market Risk Analysis. Volume II. Practical Financial Econometrics*.

[Alexander et al., 2009] Alexander, C. et al. (2009). *Market risk analysis Volume IV: Value-at-Risk models*. John Wiley & Sons Chichester.

[Alkhayat and Mehmood, 2021] Alkhayat, G. and Mehmood, R. (2021). A review and taxonomy of wind and solar energy forecasting methods based on deep learning. *Energy and AI*, 4:100060.

[Becker, 2019] Becker, D. (2019). Weekly Maximum GPU usage | Kaggle.

[Beketov et al., 2018] Beketov, M., Lehmann, K., and Wittke, M. (2018). Robo advisors: quantitative methods inside the robots. *Journal of Asset Management*, 19(6):363–370.

[Black and Litterman, 1992] Black, F. and Litterman, R. (1992). Global portfolio optimization. *Financial analysts journal*, 48(5):28–43.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.

[Brenner and Meyll, 2020] Brenner, L. and Meyll, T. (2020). Robo-advisors: a substitute for human financial advice? *Journal of Behavioral and Experimental Finance*, 25:100275.

[Brooks, 2019] Brooks, C. (2019). *Introductory econometrics for finance*. Cambridge university press.

[cat, 2020] cat, S. (2020). Making an loss vs model complexity graph using tikz. https://tex.stackexchange.com/questions/532077/making-an-loss-vs-model-complexity-graph-using-tikz. Accessed on 2023-08-16.

[CFA, 2023] CFA (2023). Systematic and non-systematic risks. *AnalystPrep | CFA® Exam Study Notes*.

[Chafi, 2022] Chafi, R. (2022). Qu'est ce que le machine learning ? - Redouane Chafi - Medium.

[Chaves et al., 2011] Chaves, D., Hsu, J., Li, F., and Shakernia, O. (2011). Risk parity portfolio vs. other asset allocation heuristic portfolios. *Journal of Investing*, 20(1):108.

[CodingFinance, 2018] CodingFinance (2018). SP500 2018.

[Conrad and Kaul, 1988] Conrad, J. and Kaul, G. (1988). Time-variation in expected returns. *Journal of business*, pages 409–425.

[Costa and Kwon, 2019] Costa, G. and Kwon, R. H. (2019). Risk parity portfolio optimization under a markov regime-switching framework. *Quantitative Finance*, 19(3):453–471.

[D'Acunto and Rossi, 2021] D'Acunto, F. and Rossi, A. G. (2021). *Robo-Advising*.

[Datacamp, 2023] Datacamp (2023). Boosting - Gradient boosting (GB).

[Deboeck, 1994] Deboeck, G. J. (1994). *Trading on the edge: neural, genetic, and fuzzy systems for chaotic financial markets*, volume 39. John Wiley & Sons.

[Deng et al., 2012] Deng, G.-F., Lin, W.-T., and Lo, C.-C. (2012). Markowitz-based portfolio selection with cardinality constraints using improved particle swarm optimization. *Expert systems with applications*, 39(4):4558–4566.

[Detective, 2021] Detective, D. (2021). The 80/20 split intuition and an alternative split method.

[dhanicova@gmail.com, 2022] dhanicova@gmail.com (2022). Markowitz Model - QuantPedia.

[Dixon et al., 2020] Dixon, M. F., Halperin, I., and Bilokon, P. (2020). *Machine learning in finance*, volume 1170. Springer.

[Drucker et al., 1996] Drucker, H., Burges, C. J., Kaufman, L., Smola, A., and Vapnik, V. (1996). Support vector regression machines. *Advances in neural information processing systems*, 9.

[D'Hondt et al., 2020] D'Hondt, C., De Winne, R., Ghysels, E., and Raymond, S. (2020). Artificial intelligence alter egos: Who might benefit from robo-investing? *Journal of Empirical Finance*, 59:278–299.

[Eazyvest, 2023] Eazyvest (2023). Quels sont les meilleurs robo-advisors de Belgique en 2023?

[Eckhardt, 2021] Eckhardt, K. (2021). Choosing the right hyperparameters for a simple LSTM using Keras.

[Elsayed et al., 2021] Elsayed, S., Thyssens, D., Rashed, A., Jomaa, H. S., and Schmidt-Thieme, L. (2021). Do we really need deep learning models for time series forecasting? *arXiv preprint arXiv:2101.02118*.

[Enke and Thawornwong, 2005] Enke, D. and Thawornwong, S. (2005). The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with applications*, 29(4):927–940.

[Fama, 1981] Fama, E. F. (1981). Stock returns, real activity, inflation, and money. *The American economic review*, 71(4):545–565.

[Finance, 2023] Finance, I. (2023). Is the tech bubble finally bursting in 2022? *International Finance*.

[Fisch et al., 2019] Fisch, J. E., Laboure, M., and Turner, J. A. (2019). *The emergence of the Robo-Advisor*.

[Forjan, 2019] Forjan, J. (2019). Optimal Portfolios. *AnalystPrep | CFA® Exam Study Notes*.

[Fulk et al., 2018] Fulk, M., Watkins, K., Kruger, M., et al. (2018). Who uses robo-advisory services, and who does not? *Financial Services Review*, 27(2):173–188.

[Furnham and Argyle, 1998] Furnham, A. and Argyle, M. (1998). *The psychology of money*. Psychology Press.

[Geurts et al., 1977] Geurts, M. D., Box, G. E. P., and Jenkins, G. M. (1977). Time Series Analysis: Forecasting and control. *Journal of Marketing Research*, 14(2):269.

[Geurts, 2017] Geurts, P. (2017). Ensemble methods and feature selection. `http://www.montefiore.ulg.ac.be/~lwh/AIA/ensembles-21-11-2017.pdf`. Accessed on 2023-08-16.

[Ghosh and Reilly, 1994] Ghosh, S. and Reilly, D. L. (1994). Credit card fraud detection with a neural-network. In *System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on*, volume 3, pages 621–630. IEEE.

[Golwala, 2021] Golwala, Z. (2021). Machine Learning algorithms and basics of artificial Intelligence.

[Goodwin, 2022] Goodwin, PhD, K. (2022). Modern Portfolio Theory: What it is amp; how it works. *PropertyMetrics*.

[Gregory et al., 2011] Gregory, C., Darby-Dowman, K., and Mitra, G. (2011). Robust optimization and portfolio selection: the cost of robustness. *European Journal of Operational Research*, 212(2):417–428.

[Grinsztajn et al., 2022] Grinsztajn, L., Oyallon, E., and Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? *Advances in Neural Information Processing Systems*, 35:507–520.

[Gu, 2021] Gu, M. (2021). In depth linear regression - towards data science.

[Gunjan and Bhattacharyya, 2022] Gunjan, A. and Bhattacharyya, S. (2022). A brief review of portfolio optimization techniques. *Artificial Intelligence Review*, 56(5):3847–3886.

[Haenlein and Kaplan, 2019] Haenlein, M. and Kaplan, A. (2019). A brief history of artificial intelligence: On the past, present, and future of artificial intelligence. *California management review*, 61(4):5–14.

[Hardesty, 2019] Hardesty, L. (2019). The history of amazon's recommendation algorithm. *Amazon Science*, 22.

[He and Litterman, 2002] He, G. and Litterman, R. B. (2002). The intuition behind Black-Litterman model portfolios. *Social Science Research Network*.

[Helms et al., 2022] Helms, N., Hölscher, R., and Nelde, M. (2022). Automated investment management: Comparing the design and performance of international robo-managers. *European Financial Management*, 28(4):1028–1078.

[Hill, 2018] Hill, T. (2018). Part 2 – gradient descent and backpropagation. `https://machinelearning.tobiashill.se/part-2-gradient-descent-and-backpropagation/`. Accessed on 2023-08-16.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Inria, 2022] Inria (2022). Hyperparameter tuning — Scikit-learn course.

[Investopedia, 2023] Investopedia (2023). Small cap stocks.

[Jiang et al., 2019] Jiang, F., Lee, J., Martin, X., and Zhou, G. (2019). Manager sentiment and stock returns. *Journal of Financial Economics*, 132(1):126–149.

[Jo, 2021] Jo, T. (2021). Machine learning foundations. *Supervised, Unsupervised, and Advanced Learning. Cham: Springer International Publishing*.

[Jorion and Khoury, 1996] Jorion, P. and Khoury, S. (1996). Financial risk management. *Cambridge/-Massachusetts*.

[Kaul, 1988] Kaul, G. (1988). Time-Variation in expected returns. *The Journal of Business*, 61(4):409.

[Keboola, 2020] Keboola (2020). The Ultimate guide to random forest regression.

[Kim and Henke, 2021] Kim, B. and Henke, G. (2021). Easy-to-use cloud computing for teaching data science. *Journal of Statistics and Data Science Education*, 29(sup1):S103–S111.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Lawrence, 1997] Lawrence, R. (1997). Using neural networks to forecast stock market prices. *University of Manitoba*, 333(2006):2013.

[Ledoit and Wolf, 2003] Ledoit, O. and Wolf, M. (2003). Honey, I shrunk the sample covariance matrix. *Social Science Research Network*.

[Lee, 2022] Lee, J. (2022). How Bank of America achieved a massive comeback from the brink of collapse.

[Lim and Zohren, 2021] Lim, B. and Zohren, S. (2021). Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209.

[Lim and Brooks, 2011] Lim, K.-P. and Brooks, R. (2011). The evolution of stock market efficiency over time: A survey of the empirical literature. *Journal of economic surveys*, 25(1):69–108.

[Lo, 2005] Lo, A. W. (2005). Reconciling efficient markets with behavioral finance: the adaptive markets hypothesis. *Journal of investment consulting*, 7(2):21–44.

[Lou et al., 2019] Lou, D., Polk, C., and Skouras, S. (2019). A tug of war: Overnight versus intraday expected returns. *Journal of Financial Economics*, 134(1):192–213.

[M, 2023] M, S. (2023). AI vs Machine learning vs Deep Learning: Know the differences. *Simplilearn.com*.

[Ma et al., 2021a] Ma, Y., Han, R., and Wang, W. (2021a). Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems With Applications*, 165:113973.

[Ma et al., 2021b] Ma, Y., Han, R., and Wang, W. (2021b). Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications*, 165:113973.

[Malkiel, 2003] Malkiel, B. G. (2003). The efficient market hypothesis and its critics. *Journal of economic perspectives*, 17(1):59–82.

[Markowitz, 1952] Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, 7(1):77–91.

[Markowitz, 1959] Markowitz, H. (1959). The semi-variance. *Chapter IX, Portfolio Selection. Efficient Diversification of Investments*, pages 188–201.

[Martin, 2021] Martin, R. A. (2021). Pyportfolioopt: portfolio optimization in python. *Journal of Open Source Software*, 6(61):3066.

[Meinert, 2019] Meinert, R. (2019). Optimizing hyperparameters in random forest classification. `https://towardsdatascience.com/optimizing-hyperparameters-in-random-forest-classification-ec7741f9d` Accessed on 2020-06-05.

[Mondal et al., 2014] Mondal, P., Shit, L., and Goswami, S. (2014). Study of Effectiveness of Time Series Modeling (ARIMA) in forecasting stock Prices. *International journal of computer science, engineering and applications*, 4(2):13–29.

[Navlani, 2023] Navlani, A. (2023). Decision Tree Classification in Python Tutorial.

[Nikolopoulou, 2022] Nikolopoulou, K. (2022). What is survivorship bias? | Definition 038; Examples. *Scribbr*.

[Parliament, 2021] Parliament, E. (2021). Robo-advisors: How do they fit in the existing EU regulatory framework, in particular with regard to investor protection?

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

[Perspectives, 2020] Perspectives, R. (2020). Why is point-in-time data crucial in backtesting? | Refinitiv perspectives.

[Pham et al., 2020] Pham, C. D. et al. (2020). Is estimating the capital asset pricing model using monthly and short-horizon data a good choice? *Heliyon*, 6(7).

[Phan et al., 2015] Phan, D. H. B., Sharma, S. S., and Narayan, P. K. (2015). Stock return forecasting: Some new evidence. *International Review of Financial Analysis*, 40:38–51.

[Phoon and Koh, 2017] Phoon, K. and Koh, F. (2017). Robo-Advisors and wealth management. *The Journal of Alternative Investments*, 20(3):79–94.

[Ponnaiya and Ryan, 2017] Ponnaiya, S. and Ryan, K. (2017). Robo-advisors: The rise of automated financial advice. *Ipsos.[Google Scholar]*.

[PricewaterhouseCoopers,] PricewaterhouseCoopers. One in six asset and wealth management companies will be swallowed up or fall by the wayside in the next five years: PWC Global Asset Wealth Management Survey.

[Qi and Maddala, 1999] Qi, M. and Maddala, G. (1999). Economic factors and the stock market: a new perspective. *Journal of forecasting*, 18(3):151–166.

[Ramelli and Wagner, 2020] Ramelli, S. and Wagner, A. (2020). What the stock market tells us about the consequences of covid-19. *Mitigating the COVID Economic Crisis: Act Fast and Do Whatever*, 63.

[Raschka, 2018] Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*.

[Reilly and Brown, 2011] Reilly, F. K. and Brown, K. C. (2011). *Investment Analysis and Portfolio Management*. Cengage Learning.

[Rossi, 2018] Rossi, A. G. (2018). Predicting stock market returns with machine learning. *Georgetown University*.

[Rubilmax, 2022] Rubilmax (2022). Introduction au contrastive learning : une forme d'apprentissage auto supervisé.

[Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

[scikit learn, 2020] scikit learn (2020). Parameter tuning guidelines. `https://scikit-learn.org/stable/modules/ensemble.html#random-forest-parameters`. Accessed on 2020-06-05.

[Scutella and Recchia, 2013] Scutella, M. G. and Recchia, R. (2013). Robust portfolio asset allocation and risk measures. *Annals of Operations Research*, 204(1):145–169.

[Sharpe, 1966] Sharpe, W. F. (1966). Mutual fund performance. *The Journal of business*, 39(1):119–138.

[Shrivastava, 2023] Shrivastava, S. (2023). Cross Validation in time Series - Soumya Shrivastava - Medium.

[Statista, 2023] Statista (2023). AUM of Robo-Advisors Worldwide 2018-2027.

[Tejimandi, 2022] Tejimandi (2022). Efficient Market Hypothesis: A Unique Market perspective. *Teji mandi*.

[Timmermann, 2008] Timmermann, A. (2008). Elusive return predictability. *International Journal of Forecasting*, 24(1):1–18.

[Town, 2022] Town, P. (2022). 15 Types of Investments: What Will Make You the Most Money?

[Tu and Zhou, 2011] Tu, J. and Zhou, G. (2011). Markowitz meets talmud: A combination of sophisticated and naive diversification strategies. *Journal of Financial Economics*, 99(1):204–215.

[Van Eyden, 1995] Van Eyden, R. J. (1995). The application of neural networks in the forecasting of share prices.

[Vanschoren et al., 2014] Vanschoren, J., Van Rijn, J. N., Bischl, B., and Torgo, L. (2014). Openml: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[Wang et al., 2019] Wang, H., Lei, Z., Zhang, X., Zhou, B., and Peng, J. (2019). A review of deep learning for renewable energy forecasting. *Energy Conversion and Management*, 198:111799.

[White, 1988] White, H. (1988). Economic prediction using neural networks: The case of ibm daily stock returns. In *ICNN*, volume 2, pages 451–458.

[Willmott and Matsuura, 2005] Willmott, C. J. and Matsuura, K. (2005). Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1):79–82.

[Wolff and Echterling, 2020] Wolff, D. and Echterling, F. (2020). Stock picking with machine learning. *Available at SSRN 3607845*.

[Xie, 2022] Xie, K. (2022). Review Machine Learning with Me: Overfitting vs. Underfitting.

[Zhao, 2016] Zhao, P. (2016). R for deep learning (i): Build fully connected neural network from scratch. https://www.r-bloggers.com/r-for-deep-learning-i-build-fully-connected-neural-network-from-scra Accessed on 2023-08-16.

[Zou and Hastie, 2005] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320.